

Пакеты прикладных программ



**Кафедра «Вычислительные системы и
информационная безопасность»**

Учебно-методический комплекс дисциплины

Составитель

Цветкова Ольга Леонидовна

ОГЛАВЛЕНИЕ

Рабочая программа	5
Учебно-методические рекомендации для практических занятий	27
Цели и задачи дисциплины «Пакеты прикладных программ»	28
Описание последовательности изучения УМК.....	28
Рекомендации изучения отдельных тем курса:.....	29
Особенности самостоятельного изучения курса. Советы для подготовки к рейтинговому контролю и экзамену	30
Распределение времени для самостоятельной работы студентов Самостоятельная работа студентов	31
Рекомендации по работе с литературой.....	32
Курс Лекций	35
Лекция № 1 Пакеты прикладных программ	36
Лекция № 2 Ознакомление с пакетом Mathcad	47
Лекция № 3 Mathcad: Выражения, стандартные функции и функции, определяемые пользователем.....	52
Лекция № 4 Mathcad: Создание графиков	66
Лекция № 5 Mathcad: Задание и обработка массивов. Символьные вычисления	92
Лекция № 6 Mathcad: Дифференцирование и интегрирование. Решение уравнений и их систем.....	111
Лекция № 7 Mathcad: Элементы программирования.....	123
Лекция № 8 Mathcad: Функции для решения задач интерполяции и аппроксимации ...	129
Лекция № 9 Mathcad: Функции для решения задач сглаживания и экстраполяции (предсказания)	142
Лекция № 10 Ознакомление с пакетом Matlab. Выражения	147
Лекция № 11 Matlab: Работа с матрицами.....	162
Лекция № 12 Matlab: Элементы графики	169
Лекция № 13 Matlab: Символьные вычисления.....	177
Лекция № 14 Matlab: Дифференцирование и интегрирование	182
Лекция № 15 Matlab: Типовые средства программирования	185

Лекция № 16 Matlab: Составление функций пользователя.....	194
Лекция № 17 Matlab: Решение уравнений и их систем.....	198
Лекция № 18 Matlab: Функции для решения задач интерполяции и аппроксимации	208
Лекция № 19 Matlab: Создание пользовательского интерфейса	214
Лекция № 20 Пакеты расширения Matlab	215
Лекция № 21 Matlab: Simulink.....	236
Рекомендации для организации самостоятельной работы по дисциплине.....	246
Лабораторная работа Ознакомление с пакетом Mathcad.....	247
Лабораторная работа Mathcad: Создание графиков	263
Лабораторная работа Mathcad: Задание и обработка массивов	299
Лабораторная работа Mathcad: Символьные вычисления.....	306
Лабораторная работа Mathcad: Операторы суммы Σ и произведения Π . Дифференцирование и интегрирование.....	323
Лабораторная работа Mathcad: Решение уравнений и их систем	335
Лабораторная работа Mathcad: Решение дифференциальных уравнений и их систем	344
Лабораторная работа Mathcad: Элементы программирования	353
Лабораторная работа Mathcad: Функции для решения задач интерполяции и аппроксимации.....	363
Лабораторная работа Mathcad: Функции для решения задач сглаживания и экстраполяции (предсказания)	378
Лабораторная работа Ознакомление с пакетом Matlab. Операции с матрицами	384
Лабораторная работа Matlab: Элементы графики	404
Лабораторная работа Matlab: Символьные вычисления.....	415
Лабораторная работа Matlab: Дифференцирование и интегрирование	424
Лабораторная работа Matlab: Типовые средства программирования.....	433
Лабораторная работа Matlab: Составление функций пользователя	446
Лабораторная работа Matlab: Решение уравнений и их систем	452
Лабораторная работа Matlab: Решение дифференциальных уравнений и их систем..	458
Лабораторная работа Matlab: Функции для решения задач интерполяции	467

Лабораторная работа Matlab: Функции для решения задач аппроксимации	472
Лабораторная работа Matlab: Создание пользовательского интерфейса	476
Лабораторная работа Matlab: Simulink.....	477
Итоговые вопросы по дисциплине	489
Вопросы по дисциплине	490
Вывод списка пакетов расширения	491
Обзор пакетов расширения Matlab.....	491
Контрольно-измерительные материалы по дисциплине.....	492

Рабочая программа

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)

Факультет Энергетика и системы коммуникаций

Кафедра Вычислительные системы и информационная безопасность

УТВЕРЖДАЮ
Проректор по МР
Н.Н. Шумская
«26» 011 2012 г
Рег. № 367

РАБОЧАЯ ПРОГРАММА

По дисциплине Б2.В.ОД.3. Пакеты прикладных программ

По направлению 090900 Информационная безопасность

Форма и срок освоения ООП очная нормативный

Общая трудоемкость – 4 (з.е.)

Всего учебных часов – 144 час.

ФОРМЫ КОНТРОЛЯ	СЕМЕСТР			
	Нормативный срок		Сокращенный срок	
	очная	заочная	очная	заочная
Экзамен	4			
Зачет				
КР	4			
КП				

Адреса электронной версии программы _____

Ростов-на-Дону
20 12

Лист согласования

Рабочая программа по дисциплине «Пакеты прикладных программ» составлена в соответствии с требованиями основной образовательной программы, сформированной на основе Федерального государственного образовательного стандарта высшего профессионального образования по направлению подготовки 090900 Информационная безопасность.

Дисциплина относится к циклу обязательных дисциплин вариативной части (Б2).

Рабочая программа составлена коллективом авторов к.т.н., доцент Цветкова О.Л.,
Ганжур М.А. и рассмотрена на заседании кафедры

«Вычислительные системы и информационная безопасность»

Протокол № 4 от « 19 » ноября 2012 г

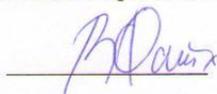
Зав. кафедрой «Вычислительные
системы и информационная безопасность»



В.А.Фатхи

Одобрена методическим советом направления «Информационная безопасность»

Председатель совета



В.А.Фатхи

«19» 11 20 12 г.

Руководитель ЦНМО и ТОП



В.В. Юр'ева

«22» 11 20 12 г.



Раздел 1. Общие положения

1.1 Цели и задачи дисциплины, ее место в учебном процессе.

Целью изучения дисциплины «Пакеты прикладных программ» является подготовка бакалавров к деятельности, связанной с использованием пакетов прикладных программ, разработкой, анализом вычислительной сложности и применением алгоритмов. При изучении данного курса у студентов формируются знания, и навыки, необходимые для решения профессиональных задач.

Задачи дисциплины:

- 1) изучение пакетов прикладных для проведения расчетов;
- 2) изучение основных математических методов и алгоритмов для проведения расчетов;
- 3) привитие навыков выбора методов для и проведения расчетов,
- 4) выработка навыков использования математических методов при решении практических задач профессиональной сферы.

1.2 Связь с предшествующими и последующими дисциплинами (модулями, практиками, научно-исследовательской работой (НИР)).

Для успешного усвоения данной дисциплины необходимо, чтобы студент владел знаниями, умениями и навыками, сформированными в процессе изучения дисциплин:

«Технологии и методы программирования» - знать основы выбора и разработки алгоритмов, методы оценки вычислительной сложности.

«Вычислительная математика» - использовать численные методы для решения профессиональных задач.

Дисциплина «Пакеты прикладных программ» является предшествующей для изучения следующих дисциплин: «Криптографические методы защиты информации», «Надежность автоматизированных систем», «Управление системами», «Обработка экспериментальных данных», при прохождении производственной практики, а также при разработке курсовых работ итоговой государственной аттестации, и выполнения научно-исследовательской работы.

Раздел 2. Компетенции обучающегося, формируемые в результате освоения дисциплины и планируемые результаты обучения.

Студенты, завершившие изучение дисциплины «Пакеты прикладных программ», должны обладать следующими вузовскими (обобщенными) компетенциями (ВК):

- общекультурными компетенциями (ВОК) – способность к обобщению, анализу, восприятию информации, постановке цели и выбору путей ее достижения, владеть культурой мышления;
- профессиональными компетенциями (ВПК) – Способность применять программные средства и реализовывать алгоритмы для решения профессиональных задач.

ВОК	Знать	Уметь	Владеть
Уровень 1	Основные законы естественнонаучных дисциплин	Формулировать цели решения задач	Способность к обобщению и анализу
Уровень 2	Основные методы анализа информации	Формулировать цели и выбирать пути ее достижения	Способность к обобщению, анализу, восприятию информации
Уровень 3	Методы анализа информации	Формулировать цели и выбирать эффективные пути ее достижения	Способность к обобщению, анализу, восприятию информации, владеть культурой мышления
Уровень 1	Основные функции и инструменты пакетов прикладных программ	применять программные средства системного, прикладного и специального назначения для задач с очевидным решением	способностью к программной реализации алгоритмов решения типовых задач с очевидным решением
Уровень 2	Основные функции и инструменты пакетов прикладных программ	применять программные средства системного, прикладного и специального назначения и использовать инструментальные средства и системы программирования для решения профессиональных задач	способностью к программной реализации алгоритмов решения типовых задач
Уровень 3	Основные функции и инструменты пакетов прикладных программ, их назначения, преимущество и недостатки	Эффективно применять программные средства системного, прикладного и специального назначения и использовать инструментальные средства и системы программирования для решения профессиональных задач	способностью к программной реализации алгоритмов решения типовых задач для обеспечения информационной безопасности

Таблица соответствия ВК и компетенций по ФГОС

ВК	Код направления	Перечень компетенций направления (ОК или ПК)
ВОК	090900	Способность к обобщению, анализу, восприятию информации, постановке цели и выбору путей ее достижения, владеть культурой мышления (ОК-8).
ВПК	090900	Способность применять программные средства системного, прикладного и специального назначения (ПК-15); Способность использовать инструментальные средства и системы программирования для решения профессиональных задач (ПК-16);

Способность к программной реализации алгоритмов решения типовых задач обеспечения информационной безопасности (ПК-17).

Раздел 3 Структура и содержание дисциплины

3.1 Тематический план дисциплины

Раздел (название)	Тема, литература	Содержание
1. Введение и ознакомление с прикладным программным пакетом Mathcad	1.1 Пакеты прикладных программ [6.1.1, 6.1.4, 6.2.1]	Структура и основные компоненты ППП Основные группы пакетов прикладных программ Эволюция ППП
	1.2 Ознакомление с пакетом Mathcad [6.1.1, 6.1.2, 6.3.1, 6.3.2]	Основные возможности пакета Mathcad Сравнительная характеристика пакета Mathcad с аналогичными системами Расширение функциональности пакета Mathcad Взаимодействие Mathcad с другими программами Структура Mathcad-документа и состав панели инструментов Math (Математика)
	1.3 Mathcad: Выражения, стандартные функции и функции, определяемые пользователем [6.1.1, 6.1.2, 6.3.1, 6.3.2]	Выражения и стандартные функции Определение переменных (присвоение значений переменным) Размерности физических величин Функции, определяемые пользователем Запись логических выражений. Функция if Переменные диапазона (ранжированные переменные)
	1.4 Mathcad: Создание графиков [6.1.1, 6.1.2, 6.3.1, 6.3.2]	Состав панели инструментов Math → Graph (Математика → График) Построение X-Y-графиков Построение графиков в полярной системе координат Форматирование двумерных

		<p>графиков</p> <p>Построение трехмерных графиков</p> <p>Форматирование трехмерных графиков</p> <p>Создание анимации</p>
2. Прикладной программный пакет Mathcad	2.1 Mathcad: Задание и обработка массивов. Символьные вычисления [6.1.1, 6.1.2, 6.3.1, 6.3.2]	<p>Задание индексированной переменной</p> <p>Операторы суммы Σ и произведения Π</p> <p>Задание и обработка массивов</p> <p>Символьные вычисления</p>
	2.2 Mathcad: Дифференцирование и интегрирование. Решение уравнений и их систем [6.1.1, 6.1.2, 6.3.1, 6.3.2]	<p>Дифференцирование и интегрирование</p> <p>Решение уравнений и их систем</p> <p>Решение дифференциальных уравнений и их систем</p>
	2.3 Mathcad: Элементы программирования [6.1.1, 6.1.2, 6.3.1, 6.3.2]	Элементы программирования
	2.4 Mathcad: Функции для решения задач интерполяции и аппроксимации [6.1.1, 6.1.2, 6.3.1, 6.3.2]	<p>Функции для решения задач интерполяции</p> <p>Функции для решения задач аппроксимации</p>
	2.5 Mathcad: Функции для решения задач сглаживания и экстраполяции (предсказания) [6.1.1, 6.1.2, 6.3.1, 6.3.2]	<p>Функции для решения задач сглаживания</p> <p>Функции для решения задач экстраполяции (предсказания)</p>
3. Прикладной программный пакет Matlab	3.1 Ознакомление с пакетом Matlab. Выражения [6.1.3, 6.1.4, 6.1.5, 6.3.3, 6.3.4]	<p>Основные возможности пакета Matlab</p> <p>Наборы инструментов пакета Matlab</p> <p>Структура и рабочие окна пакета Matlab</p> <p>Работа в командном режиме</p> <p>Основные элементы языка программирования Matlab</p>
	3.2 Matlab: Работа с матрицами [6.1.3, 6.1.4, 6.1.5, 6.3.3, 6.3.4]	<p>Ввод матриц</p> <p>Индексы элементов матриц</p>

		Матричные функции линейной алгебры Массивы
3.3 Matlab: Элементы графики [6.1.3, 6.1.4, 6.1.5, 6.3.3, 6.3.4]		Создание графика Окна изображений. Добавление кривых на существующий график. Подграфики Управление осями. Подписи к осям и заголовки Трехмерные графики
3.4 Matlab: Символьные вычисления [6.1.3, 6.1.4, 6.1.5, 6.3.3, 6.3.4]		Создание символьных переменных и выражений Символьные операции с выражениями
3.5 Matlab: Дифференцирование и интегрирование [6.1.3, 6.1.4, 6.1.5, 6.3.3, 6.3.4]		Дифференцирование Интегрирование
3.6 Matlab: Типовые средства программирования [6.1.3, 6.1.4, 6.1.5, 6.3.3, 6.3.4]		Назначение языка программирования Matlab Управляющие структуры Векторизация циклов
3.7 Matlab: Составление функций пользователя [6.1.3, 6.1.4, 6.1.5, 6.3.3, 6.3.4]		Функции, задаваемые пользователем Понятие о файлах-функциях Глобальные переменные
3.8 Matlab: Решение уравнений и их систем [6.1.3, 6.1.4, 6.1.5, 6.3.3, 6.3.4]		Решение уравнений и их систем Решение дифференциальных уравнений и их систем
<i>3.9 Matlab: Функции для решения задач интерполяции и аппроксимации [6.1.3, 6.1.4, 6.1.5, 6.3.3, 6.3.4]</i>		Функции для решения задач интерполяции Функции для решения задач аппроксимации
3.10 Matlab: Создание пользовательского интерфейса [6.1.3, 6.1.4, 6.1.5, 6.3.3, 6.3.4]		Создание пользовательского интерфейса

	3.11 Пакеты расширения Matlab [6.1.3, 6.1.4, 6.1.5, 6.3.3, 6.3.4]	<i>Вывод списка пакетов расширения</i> <i>Обзор пакетов расширения Matlab</i>
	3.12 Matlab: Simulink [6.1.3, 6.1.4, 6.1.5, 6.3.3, 6.3.4]	Назначение пакета Simulink Библиотека компонентов пакета Simulink

Распределение бюджета времени по видам занятий

3.2 Лекционные занятия

№ рейтинго вого блока	№ темы	Объем времени, час			
		норм. срок обучения		сокращ. (ускорен.)	
		очная	заочная	очная	заочная
1	1.1, 1.2	1			
	1.3, 1.4	1			
	2.1, 2.2	1			
	2.3	1			
	2.4	1			
	2.5	1			
2	3.1, 3.2	2			
	3.3, 3.4	1			
	3.5, 3.6	1			
	3.7	1			
	3.8	1			
	3.9	1			
	3.10	1			
	3.11	1			
3.12	1				
Итого		16			

3.3 Лабораторные занятия

№ рейтинг ового блока	Тема и содержание лабораторного занятия	№ темы из раздела 3	Объем времени, час			
			норм. срок обучения		сокращ. (ускорен.)	
			очная	заочная	очная	заочная
1	2	3	4	5	6	7
1	Ознакомление с пакетом Mathcad. Создание графиков	1.2, 1.3, 1.4	2			
	Mathcad: Задание и обработка массивов	2.1	2			
	Mathcad: Символьные вычисления. Операторы суммы Σ и произведения Π . Дифференцирование и интегрирование	2.1, 2.2	2			
	Mathcad: Решение уравнений и их систем	2.2	2			
	Mathcad: Элементы программирования	2.3	2			
	Mathcad: Функции для решения задач интерполяции, аппроксимации, сглаживания и экстраполяции (предсказания)	2.4, 2.5	4			
2	Ознакомление с пакетом Matlab. Операции с матрицами	3.1, 3.2	2			
	Matlab: Элементы графики	3.3	2			
	Matlab: Символьные вычисления	3.4	2			
	Matlab: Дифференцирование и	3.5	2			

	интегрирование					
	Matlab: Типовые средства программирования. Составление функций пользователя	3.6, 3.7	4			
	Matlab: Решение уравнений и их систем	3.8	2			
	Matlab: Функции для решения задач интерполяции и аппроксимации	3.9	2			
	Matlab: Создание пользовательского интерфейса	3.10	2			
	Matlab: Simulink	3.11, 3.12	2			
Итого			34			

3.4 Самостоятельная работа студентов

№ п.п.	Вид самостоятельной работы	Объем времени, час				Рекомендуемая литература
		норм.		сокр.		
		очная	заочная	очная	заочная	
1	2	3	4	5	6	7
1	Усвоение текущего материала	4				[6.1.1, 6.1.2, 6.1.3, 6.1.4]
2	Подготовка к экзамену	36				[6.1.1, 6.1.2, 6.1.3, 6.1.4, 6.1.5, 6.3.1, 6.3.2, 6.3.3, 6.3.4, 6.2.1]
3	Курсовая работа	50				[6.1.1, 6.1.2, 6.1.3, 6.1.4, 6.5.1]
4	Подготовка к лабораторным занятиям	4				[6.4.1, 6.4.2,]
		94				

3.5 Курсовая работа, ее характеристика (цели и задачи), примерная тематика

Тема курсовой работы «Компьютерное моделирование процессов».

Цель курсовой работы: развитие у студентов способностей применять программные средства прикладного назначения и системы программирования для решения математических задач.

Тематика: курсовая работа заключается в освоение методов составления, написания и отладки программ для решения математических задач с помощью пакетов MathCAD и MatLab. В процессе выполнения курсовой работы, необходимо используя пакеты прикладных программ MathCAD и MatLab выполнить задания в соответствии с вариантом.

3.6. Распределение баллов за текущую работу

Вид текущей учебной работы	Количество баллов
Тестовый контроль	10
Выполнение лабораторных работ	40
Выполнение курсовой работы	50
Итого:	100

Раздел 4 Образовательные технологии

Реализация программы предусматривает использование образовательных технологий, направленных на формирование элементов компетенций, в обеспечении которых участвует дисциплина «Пакеты прикладных программ». В процессе обучения реализуется лекционно-лабораторная система обучения, и используются следующие образовательные технологии:

- В ходе проведения лекционных занятий используется классическая методика с разбиением на 3 части: вводная, основная, заключительная. Во время лекций выделяется время для активных и интерактивных форм проведения занятий.

- В ходе проведения лабораторных занятий студенты проводят исследования на компьютерной технике по индивидуальному заданию. Используются активные и интерактивные технологии.

- В ходе выполнения курсовой работы студенты изучают численные методы с использованием пакетов прикладных программ.

В процессе реализации указанных технологий выполняются следующие условия

- Чтение лекций с использованием проекционной техники – 25 %
- Проведение интерактивных занятий — 50 %
- Занятия, проводимые в активных формах — 10 %

Раздел 5. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации и самоконтроля по итогам освоения дисциплины

Рабочая программа дисциплины обеспечена фондом оценочных средств для проведения текущего контроля, промежуточного контроля, экзаменов. Фонд включает:

- задания для текущего контроля, в том числе в тестовой форме;
- вопросы к экзаменам, в соответствии с рейтинговыми блоками;
- критерии для оценки достижения результатов освоения дисциплины в целом по каждому виду работ.

5.1 Перечень вопросов для подготовки к промежуточным аттестациям и экзаменам

4 семестр

1 рейтинг

Раздел 1. Основные понятия дисциплины и типы данных

1. Принципы организации пакетов прикладных программ
2. Основные группы пакетов прикладных программ
3. Особенности программного обеспечения технологических процессов
4. Простые арифметические вычисления Mathcad
5. Стандартные функции Mathcad
6. Построение графиков функций Mathcad
7. Операции с векторами и матрицами Mathcad
8. Символьные вычисления Mathcad

Раздел 2. Понятие алгоритма и структурное программирование.

1. Интегрирование и дифференцирование Mathcad
2. Решение алгебраических уравнений Mathcad
3. Mathcad: Решение дифференциальных уравнений и их систем
4. Mathcad: Элементы программирования
5. Функции для решения задач интерполяции Mathcad
6. Функции для решения задач аппроксимации Mathcad
7. Функции для решения задач сглаживания Mathcad
8. Функции для решения задач экстраполяции (предсказания) Mathcad

2 рейтинг

Раздел 3. Сложные типы данных

1. Ознакомление с пакетом Matlab
2. Работа в командном режиме Matlab
3. Выражения в Matlab
4. Ввод матриц в Matlab

5. Индексы элементов матриц в Matlab
6. Линейная алгебра. Массивы. Многомерные данные в Matlab
7. Создание графика в Matlab
8. Окна изображений. Добавление кривых на существующий график в Matlab
9. Управление осями. Подписи к осям и заголовки в Matlab
10. Трехмерные графики в Matlab
11. Создание символьных переменных и выражений в Matlab
12. Упрощение символьных выражений в Matlab
13. Интегрирование в Matlab
14. Дифференцирование в Matlab
15. Решение алгебраических уравнений в Matlab
16. Matlab: Структуры управления потоками в Matlab
17. Понятие о файлах-функциях в Matlab
18. Глобальные переменные в Matlab
19. Векторизация в Matlab
20. Matlab: Решение систем дифференциальных уравнений
21. Функции для решения задач интерполяции в Matlab
22. Функции для решения задач аппроксимации в Matlab
23. Назначение пакета Simulink и интегрированных пакетов в Matlab
24. Библиотека компонентов пакета Simulink в Matlab

5.2 Комплект тестовых заданий

Комплекс практических заданий реализуется в пакетах прикладных программ MATHCAD и MATLAB

Задание 1.

Задать функцию:

$$f = y^2 \cdot t^3 + x \cdot t^2 + x^2 y \cdot t - y$$

Переменные x, y задаются произвольно.

Построить график функции f при изменении параметра t .

Задание 2.

Определить вектор-столбец из трех произвольных числовых элементов. Затем прибавить к нему вектор такой же размерности с другими произвольно выбранными элементами. Выполнить то же для вычитания и умножения. Применить ко второму вектору оператор транспонирования.

Создать матрицу размером 5x5 и заполнить ее произвольными элементами. Удалить строку и столбец матрицы. Затем добавить строку/столбец. Определить вектор a как один из столбцов этой матрицы.

Задание 3.

Привести подобные слагаемые в выражении:
 $(1 + x^2)y - 2xy + 1 + x^2 + z^2 + xz$ относительно каждой из переменных x, y, z .

Определить коэффициенты полинома относительно каждой из переменных x , y , z .

Задание 4.

Найти интеграл: $\int_0^1 (x^2 + 10x^3) dx$

Задание 5.

Для функции $(1 + x^2)y - 2xy + (1 + x^2)^2$ найти первую и вторую частные производные по x и y .

Задание 6.

Решить уравнение: $5x^4 - 3x^3 + 10x^2 - 15x + 8 = 0$ при помощи встроенной функции `polyroots(v)`.

Задание 7.

Решить систему уравнений при помощи встроенной функции `Find`:

$$\begin{cases} x^3 + 6y^2 = 0 \\ 2x + 9y = 0 \end{cases}$$

Задание 8.

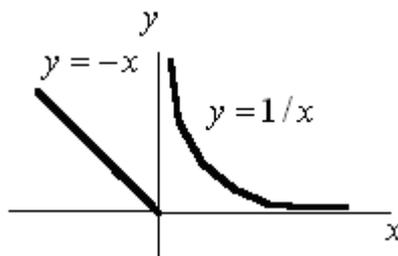
Решить ОДУ первого порядка (при помощи блока `Given/Odesolve`), задав $t1=10$, $M=100$:

$$3y'(t) - 5y(t) + 2y(t)^2 = 0, \quad y(0) = 0,1$$

Построить график решения уравнения.

Задание 9.

Для функции, определенной графически, вычислить значения функции y для всех значений аргумента x на заданном интервале $[x_n, x_k]$ с заданным шагом Δx . Результат вывести на экран в виде графика.



5.3 Технические средства обучения и контроля, использование ЭВМ

1. Компьютерные классы, оснащенные необходимым программным обеспечением.
2. Система тестирования «ЦДО-тест», разработка ДГТУ.
3. Сайт центра дистанционного обучения <http://de.dstu.edu.ru>

5.4 Уровни и критерии итоговой оценки результатов освоения

Уровни		Критерии выполнения заданий ОС	Итогов ый семестровы й балл	Итоговая оценка
Недостаточны й		Имеет представление о содержании дисциплины, но не знает основные определения, не способен выполнить задание с очевидным решением, не знает основных функций пакетов прикладных задач.	Менее 41	Неудовлетв орительно
Базовый		Знает основные определения, знает основные функции пакетов прикладных программ и может их использовать при выполнении задания с очевидным решением, способен использовать инструментальные средства и программировать несложные задачи.	41-60	Удовлетвор ительно
Повы шенны й	ПУ 1	Знает и понимает основные определения, функции пакетов прикладных программ и может их использовать при выполнении задания, в котором нет явно указанных способов решения. Способен использовать инструментальные средства и программировать задачи: собрать и провести анализ исходных данных для проектирования подсистем, может реализовать базовые алгоритмы, провести сравнительный анализ алгоритмов по критерию вычислительной сложности.	61-80	Хорошо
	ПУ 2 (пов ышен ный)	Знает и понимает основные определения, функции пакетов прикладных программ и может их использовать при выполнении задания, в котором нет явно указанных способов решения. Способен использовать инструментальные средства и программировать задачи: способен собрать и провести	81-100	Отлично

		<p>анализ исходных данных, способен провести целенаправленный поиск в различных источниках информации необходимых исходных данных для проектирования подсистем, может реализовать базовые алгоритмы, провести сравнительный анализ алгоритмов по критерию вычислительной сложности, умеет модифицировать базовые алгоритмы для решения конкретных задач, умеет разрабатывать новые эффективные алгоритмы. Может обосновано выбирать более эффективный пакет прикладных программ.</p>		
--	--	--	--	--

Раздел 6 Литература

Карта методического обеспечения дисциплины

№	Автор	Название	Издательство	Гриф издания	Год издания	Кол-во в библиотеке	Ссылка на электронный ресурс	Доступность
1	2	3	4	5	6	7	8	9
6.1 Основная литература								
6.1.1	Кудрявцев Е.М.	Mathcad 11: Полное руководство по русской версии	М.: «ДМК Пресс»		2009		http://e.lanbook.com	С любой точки доступа для авторизованных пользователей
6.1.2	Очков В.Ф.	Mathcad12 для студентов инженеров	СПб.: БХВ-Петербург		2005	2		
6.1.3	Дьяконов В.П.	MATLAB 7.*/R2006/R2007: Самоучитель	М.: «ДМК Пресс»		2009		http://e.lanbook.com	С любой точки доступа для авторизованных пользователей
6.1.4	Сизиков В.С.	Обратные прикладные задачи и MatLab	СПб.: «Лань»		2011		http://e.lanbook.com	С любой точки доступа для авторизованных пользователей
6.1.5	Шампайн Л.Ф.,	Решение обыкновенны	СПб.: «Лань	Допущено	200		http://lanbook.com	С любой точки

	Гладвел И., Томпсон С.	х дифференциальных уравнений с использованием MATLAB	»	УМО	9		com	доступа для авторизированных пользователей
6.2 Периодическая литература								
6.2.1	Периодический журнал	Компоненты и технологии	СПб.: «Файн стрит»				http://e.lanbook.com	С любой точки доступа для авторизированных пользователей
6.3 Дополнительная литература								
6.3.1	Охорзин В.А.	Прикладная математика в системе MATHCAD	СПб.: «Лань»	Допущено УМО	2009		http://e.lanbook.com	С любой точки доступа для авторизированных пользователей
6.3.2	Кудрявцев Е.М	Справочник по Mathcad 11	М.: «ДМК Пресс»		2009		http://e.lanbook.com	С любой точки доступа для авторизированных пользователей
6.3.3	Ануфриев И. Е	.Самоучитель Matlab 5.3/6.x	СПб. : БХВ-Петербург		2003	20		
6.3.4	Сизиков В.С.	Обратные прикладные задачи и MatLab. + CD	СПб.: «Лань»	Рекомендовано УМО вузов	2011		http://lanbook.com	С любой точки доступа для авториз

				РФ				ированных пользователей
6.4 Практические (семинарские) и (или) лабораторные занятия								
6.4.1	Сост.: О.Л. Цветкова	Пакеты прикладных программ для научно-технических расчетов Методические указания к выполнению лабораторных работ	Ростов н/Д: ДГТУ		201 3	10 (на кафедре)		
6.4.2	Сост.: О.Л. Цветкова, М.А. Ганжур	Комплекс методических указаний по дисциплине Пакеты прикладных программ	Ростов н/Д: ДГТУ		201 2		http://de.dstu.edu.ru	С любой точки доступа для авторизированных пользователей
6.5 Курсовая работа								
6.5.1	Сост.: О.Л. Цветкова	Методические указания к выполнению курсовой работы «Компьютерное моделирование процессов»	Ростов н/Д: ДГТУ		201 3	10 (на кафедре)		
6.6 Контрольные работы								
6.7 Программно-информационное обеспечение дисциплины, Интернет-ресурсы								
6.7.1 Пакет прикладных программ MathCad								
6.7.2 Пакет прикладных программ MatLab								
6.7.3 Официальный сайт программного продукта: MathCad — www.ptc.com/product/mathcad								

Раздел 7. Материально-техническое обеспечение дисциплины (приборы, установки, стенды и т.д.)

7.1 Помещения для проведения лекционных и лабораторных занятий укомплектованы необходимой специализированной мебелью и техническими средствами для представления учебной информации студентам, включая проекционное оборудование. Кафедра располагает компьютерным классом с компьютерами, оснащенными операционной системой Windows XP и пакетами прикладных программ MathCad и MatLab.

Экспертное заключение

Совета направления «090900 – Информационная безопасность» по рабочей программе дисциплины «Пакеты прикладных программ», предусмотренной учебным планом подготовки бакалавров по направлению 090900 «Информационная безопасность».

Рассмотрев структуру, содержание и качество оформления рабочей программы по дисциплине совет отмечает:

– рабочая программа по содержанию соответствует Федеральному государственному образовательному стандарту высшего профессионального образования в части выполнения требований, предъявляемых к уровню профессиональной квалификации выпускников по соответствующему циклу дисциплин и по самой дисциплине, а так же требованиям «Положения об основной образовательной программе высшего профессионального образования»;

– соотношение объемов основных разделов выбраны логично в целесообразных пропорциях;

– бюджет времени, отводимый на различные виды аудиторных занятий, согласуется с бюджетом времени, выделяемого для выполнения самостоятельной работы;

– объем и количество видов самостоятельной работы обоснованы, соответствуют фактическим трудозатратам на их выполнение;

– достаточную полноту информационно-методического обеспечения (подготовлен электронный конспект лекций, разработаны методические материалы для самостоятельной проработки отдельных разделов);

– разработанные оценочные средства для контроля и самоконтроля позволяют оценить уровень освоения дисциплины.

Рекомендации Совета:

подготовить заявку для регистрации конспекта лекций как электронного ресурса, издать учебное пособие.

На основании выше изложенного совет направления предлагает утвердить рабочую программу по дисциплине «Пакеты прикладных программ», представленную на экспертизу.

Председатель совета


В.А. Фатхи

Члены экспертной группы:


А.И. Зотов


В.Н. Герасименко

**Учебно-методические
рекомендации для практических
занятий**

Цели и задачи дисциплины «Пакеты прикладных программ»

Целью изучения дисциплины «Пакеты прикладных программ» является подготовка бакалавров к деятельности, связанной с использованием пакетов прикладных программ, разработкой, анализом вычислительной сложности и применением алгоритмов. При изучении данного курса у студентов формируются знания, и навыки, необходимые для решения профессиональных задач.

Задачи дисциплины:

- 1) изучение пакетов прикладных для проведения расчетов;
- 2) изучение основных математических методов и алгоритмов для проведения расчетов;
- 3) привитие навыков выбора методов для и проведения расчетов,
- 4) выработка навыков использования математических методов при решении практических задач профессиональной сферы.

Описание последовательности изучения УМК

Учебно-методический комплекс (УМК) призван помочь студентам в организации самостоятельной работы по освоению курса «Пакеты прикладных программ». Дисциплина «Пакеты прикладных программ» входит в федеральный компонент к циклу обязательных дисциплин вариативной части и изучается в 4 семестре. В течение семестра предлагается тест текущего контроля знаний, завершающим этапом изучения дисциплины является сдача экзамена.

Комплекс содержит рабочую программу дисциплины, составленную в строгом соответствии с учебным планом по специальности и государственным образовательным стандартом ВПО. Учебно-методические материалы по подготовке лекционных и лабораторных занятий в УМК представлены отдельно по каждому разделу курса «Пакеты прикладных программ» в соответствии с программой дисциплины и последовательностью изучения.

В каждом разделе даны:

1) учебно-методические материалы лекционного курса, включающие подробный план лекций по каждой изучаемой теме, вопросы и задания для самоконтроля, список основной и дополнительной литературы, с указанием конкретных разделов;

2) учебно-методические материалы по подготовке лабораторных занятий, содержащие планы проведения занятий с указанием последовательности рассматриваемых тем, задания для самостоятельной работы, краткие теоретические и учебно-методические материалы по теме, систему заданий для самопроверки. Выполнение заданий даст возможность студентам глубже усвоить теоретический материал, применить полученные знания на практике, выработать прочные умения и навыки использования технологии программирования при решении практических задач профессиональной сферы, в том числе и в кооперации с коллегами.

В комплексе представлены также контрольные тесты по всем разделам курса «Пакеты прикладных программ», которые позволят проверить уровень усвоения

изученного материала. Задания для самопроверки сопровождаются теоретическими справками и методическими рекомендациями.

Рекомендации изучения отдельных тем курса:

Материал курса разбит на 3 раздела.

При изучении первого раздела «Введение и ознакомление с прикладным программным пакетом Mathcad» следует обратить внимание на принципы организации пакетов прикладных программ; основные группы пакетов прикладных программ; особенности программного обеспечения технологических процессов. В пакете прикладных программ Mathcad изучить простые арифметические вычисления, стандартные функции, построение графиков функций.

Для исследования базовых возможности пакета прикладных программ Mathcad предусмотрены лабораторные работы, на которых изучаются простые арифметические вычисления, стандартные функции, построение графиков функций, операции с векторами и матрицами, символьные вычисления. Каждый студент выполняет свой вариант задания.

При изучении второго раздела «Прикладной программный пакет Mathcad» следует обратить внимание на: способы выполнения операций интегрирования и дифференцирования в пакете Mathcad, решения алгебраических и дифференциальных уравнений и их систем, использование функций для решения задач интерполяции, аппроксимации, сглаживания, экстраполяции (предсказания).

Для изучения второго раздела предусмотрены лабораторные работы, на которых изучаются все рассматриваемые во втором разделе темы. Каждый студент выполняет свой вариант задания.

При изучении третьего раздела «Прикладной программный пакет Matlab» необходимо ознакомиться с: пакетом прикладных программ Matlab, способами работы с матрицами, элементами графики, символьными вычислениями, способами реализации операций интегрирования и дифференцирования, решения алгебраических уравнений и систем уравнений, функциями для решения задач интерполяции и аппроксимации, средой Simulink.

Для изучения третьего раздела предусмотрены лабораторные работы, на которых изучаются прикладной программный пакет Matlab и среда Simulink. Каждый студент выполняет свой вариант задания.

Также предусмотрена курсовая работа по теме «Численное решение уравнений и систем уравнений». Целью курсовой работы является освоение методов составления, написания и отладки программ для решения математических задач с использованием пакетов MathCAD и MatLab. В процессе выполнения курсовой работы, необходимо используя пакеты прикладных программ MathCAD и MatLab выполнить задания в соответствии с вариантом: написать программу для решения системы линейных уравнений методами простых итераций и Зейделя.

Особенности самостоятельного изучения курса. Советы для подготовки к рейтинговому контролю и экзамену

Прежде чем приступить к выполнению заданий для самоконтроля, студентам необходимо изучить материал лекций и сопоставить его с трактовками, предлагаемыми в рекомендуемой по каждой теме литературе.

На дневном отделении в качестве основных элементов учебного процесса выступают проблемно-ориентированные лекции с объяснением и иллюстрированием ключевых понятий связанных с пакетами прикладных программ. Лабораторные занятия организованы в компьютерных классах по выполнению конкретных практических задач. Ход и уровень усвоения материала студентами очного отделения оцениваются с помощью рейтинговой шкалы. Максимальная сумма, которую можно набрать, успешно выполнив все тесты (включая итоговый), составляет 150 баллов.

Распределение баллов за текущую работу

Вид текущей учебной работы	Количество баллов
Тестовый контроль	10
Выполнение лабораторных работ	40
Выполнение курсовой работы	50
Экзамен	50
Итого	150

Для студентов особое значение приобретает самостоятельная проработка материала курса по учебникам и пособиям, в том числе и кафедральным.

Общий список учебной, учебно-методической и научной литературы представлен в рабочей программе, курсе лекций и методических рекомендациях по каждой тематике.

Следует учитывать тот факт, что отводимые на изучение курса «Пакеты прикладных программ» часы не позволяют в полной мере охватить все разделы, отведенные к данному курсу. Более подробно, некоторые разделы курса «Пакеты прикладных программ» предлагается студентам изучать индивидуально. Индивидуальная работа студента предполагает самостоятельное составление конспекта и изучение рекомендуемой литературы. Работу следует начинать с прочтения материала с целью уяснения его содержания, основной идеи, выделения выводов и аргументов автора. Конспектировать рекомендуется лишь при повторном чтении.

Особое внимание следует обращать на определение основных понятий дисциплины. Студент должен подробно разбирать приведенные примеры на лекциях и лабораторных работах и уметь строить аналогичные примеры самостоятельно. Это является одним из важных условий усвоения дисциплины.

Все пропущенные лабораторные занятия подлежат отработке. Форма отработки — сдача задания по теме. В рамках самостоятельной работы организуется проведение консультаций, на которых будет осуществляться обсуждение прошедших заданий, а также прием отработки пропущенных занятий и лабораторных работ. На консультациях же вы сможете получить ответы на любые вопросы, возникшие по ходу освоения курса в целом и по выполнению заданий.

Распределение времени для самостоятельной работы студентов

Самостоятельная работа студентов

№ п.п.	Вид самостоятельной работы	Объем времени, час				Рекомендуемая литература
		норм.		сокр.		
		очная	заочная	очная	заочная	
1	2	3	4	5	6	7
1	Усвоение текущего материала	4				[6.1.1, 6.1.2, 6.1.3, 6.1.4]
2	Подготовка к экзамену	36				[6.1.1, 6.1.2, 6.1.3, 6.1.4, 6.1.5, 6.3.1, 6.3.2, 6.3.3, 6.3.4, 6.2.1]
3	Курсовая работа	50				[6.1.1, 6.1.2, 6.1.3, 6.1.4, 6.5.1]
4	Подготовка к лабораторным занятиям	4				[6.4.1, 6.4.2,]
		94				

Для контроля самостоятельной работы студентов используется экзамен. Экзамен проводится в устной форме, вопросы по подготовке к экзамену составлены с учетом всех тем курса в соответствии с программой. Экзаменационный билет содержит два теоретических вопроса и один практический, связанный с использованием одного из изучаемых пакетов прикладных программ.

Рекомендации по работе с литературой

При изучении дисциплины особое внимание следует обратить на следующие литературные источники:

№	Автор	Название	Издательство	Гриф издания	Год издания	Кол-во в библиотеке	Ссылка на электронный ресурс	Доступ
1	2	3	4	5	6	7	8	9
6.1 Основная литература								
6.1.1	Очков В.Ф.	Mathcad12 для студентов инженеров	СПб.: БХВ-Петербург		2005	2		
6.1.2	Поршнева С.В.	MATLAB 7. Основы работы и программирования	М.: Бино		2006	3		
6.1.3	Кудрявцев Е.М.	Mathcad 11: Полное руководство по русской версии	М.: "ДМК Пресс"		2009		http://elibrary.ru/entry.asp?id=1172	с ПК локальной сети ДГТУ
6.1.4	Шампайн Л.Ф., Гладвел И., Томпсон С.	Решение обыкновенных дифференциальных уравнений с использованием MATLAB	СПб.: "Лань"	Допущено МО	2009		http://elibrary.ru/entry.asp?id=436	с ПК локальной сети ДГТУ
6.1.5	Дьяконов В.П.	MATLAB 7.*/R2006/R2007: Самоучитель	М.: "ДМК Пресс"		2009		http://elibrary.ru/entry.asp?id=1172	с ПК локальной сети ДГТУ

Информационные технологии в менеджменте

							_cid=25&pl1_id=1178	
6.2 Периодическая литература								
6.2.1	Периодический журнал	Компоненты и технологии	СПб.: "Файн стрит "				http://e.lanbook.com/journal/element.php?pl10_cid=227&pl10_id=2026	
6.3 Дополнительная литература								
6.3.1	Ануфриев И. Е	.Самоучитель Matlab 5.3/6.x	СПб. : БХВ-Петербург		2003	20		
6.3.2	Охорзин В.А.	Прикладная математика в системе MATHCAD	СПб.: "Лань "	Допущено МО	2009		http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=294	с ПК локальной сети ДГТУ
6.3.3	Кудрявцев Е.М	Справочник по Mathcad 11	М.: "ДМК Пресс "		2009		http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=1173	с ПК локальной сети ДГТУ
6.3.4	Сизиков В.С.	Обратные прикладные задачи и MatLab. + CD	СПб.: "Лань "	Рекомендовано УМО вузов РФ	2011		http://lanbook.com/books/element.php?pl1_cid=51&pl1_id=718	с ПК локальной сети ДГТУ

Информационные технологии в менеджменте

6.4 Практические (семинарские) и (или) лабораторные занятия								
6.4.1	Цветкова О.Л.	Пакеты прикладных программ для научно-технических расчетов Методические указания к выполнению лабораторных работ	Ростов н/Д: РГАСХМ		2010		-	с ПК локальной сети кафедры «ВС и ИБ»
6.4.2	Цветкова О.Л., Ганжур М.А.	Комплекс методических указаний по дисциплине Пакеты прикладных программ	Ростов н/Д: ДГТУ		2012		http://de.dstu.edu.ru	Логин: student Пароль: cdodstu
6.5 Курсовая работа								
6.5.1	Ханова А.А.	Численное решение уравнений и систем уравнений	Астрахань, АГТУ		2001		http://window.edu.ru/resource/943/53943	
6.7 Программно-информационное обеспечение дисциплины, Интернет-ресурсы								
6.7.1 Пакет прикладных программ MathCad								
6.7.2 Пакет прикладных программ MatLab								
6.7.3 Официальный сайт программного продукта: MatLab - www.mathworks.com								
6.7.4 Официальный сайт программного продукта: MathCad - www.ptc.com/product/mathcad								

КУРС ЛЕКЦИЙ



Лекция № 1

Пакеты прикладных программ

1. Структура и основные компоненты ППП
2. Основные группы пакетов прикладных программ
3. Эволюция ППП

1. Структура и основные компоненты ППП

Пакеты прикладных программ (ППП) — комплекс взаимосвязанных программ, предназначенных для решения определенного круга задач из различных проблемных областей, снижающих трудоемкость и повышающих эффективность работы пользователя.

ППП создаются для решения наиболее массовых научно-технических, инженерных, экономических и многих других классов задач. Суть ППП состоит в максимальном упрощении интерфейса с ЭВМ пользователя.

На текущем этапе развития информационных технологий именно ППП являются наиболее востребованным видом прикладного ПО. Это связано со следующими *особенностями ППП*:

- *Ориентация на решение класса задач.* Одной из главных особенностей является ориентация ППП не на отдельную задачу, а на некоторый класс задач, в том числе и специфичных, из определенной предметной области. Так, например, офисные пакеты ориентированы на офисную деятельность, одна из задач которой — подготовка документов (в общем случае включающих не только текстовую информацию, но и таблицы, диаграммы, изображения). Следовательно, офисный пакет должен реализовывать функции обработки текста, представлять средства обработки табличной информации, средства построения диаграмм разного вида и первичные средства редактирования растровой и векторной графики.

- *Наличие языковых средств.* Другой особенностью ППП является наличие в его составе специализированных языковых средств, позволяющих расширить число задач, решаемых пакетом или адаптировать пакет под конкретные нужды. Пакет может представлять поддержку нескольких входных языков, которые используются для формализации исходной задачи, описания алгоритма решения и начальных данных, организации доступа к внешним источникам данных, разработки программных модулей, описания модели предметной области, управления процессом решения в диалоговом режиме и других целей. Примерами входных языков ППП являются VBA в пакете MS Office, AutoLISP/VisualLISP в Autodesk AutoCAD, StarBasic в OpenOffice.org

- *Единообразие работы с компонентами пакета.* Еще одна особенность ППП состоит в наличии специальных системных средств, обеспечивавших унифицированную работу с компонентами. К их числу относятся специализированные банки данных, средства информационного обеспечения, средства взаимодействия пакета с операционной системой, типовой пользовательский интерфейс и т.п.

Обобщенную внутреннюю структуру ППП можно представить в виде трех взаимосвязанных элементов (рис. 1):

Информационные технологии в менеджменте

1. входной язык (макроязык, язык управления) — представляет средство общения пользователя с пакетом;
2. предметное обеспечение (функциональное наполнение) — реализует особенности конкретной предметной области;
3. системное обеспечение (системное наполнение) — представляет низкоуровневые средства, например, доступ к функциям операционной системы.

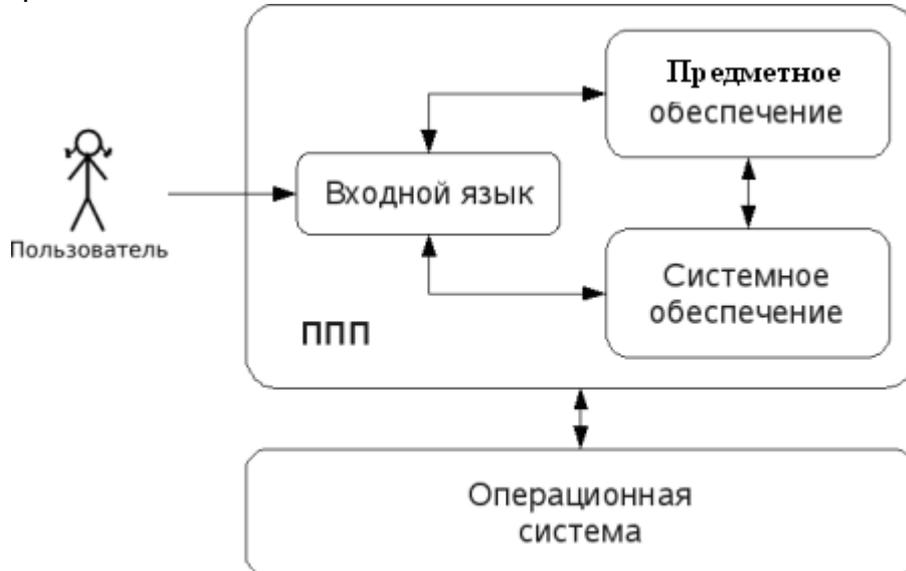


Рис. 1. Структура ППП

Входной язык — основной инструмент при работе пользователя с пакетом прикладных программ. В качестве входного языка могут использоваться как универсальные (Pascal, Basic и т.п.), так и специализированные, проблемно-ориентированные языки программирования (Cobol — для бизнес-приложений, Lisp — списочные структуры данных, Fortran и MathLAB — математические задачи и т.п.).

Развитый пакет может обладать несколькими входными языками, предназначенными для выполнения различных функций в рамках решаемого класса задач. Так, например, в пакете OpenOffice.org поддерживаются языки StarBasic, Python, JavaScript и Java. StarBasic является основным входным языком, предназначенным для автоматизации работы с пакетом, для этого языка имеется интегрированная среда разработки и встроенный отладчик. Скрипты на языках Python и JavaScript загружаются и исполняются из внешних файлов. На Java (через SDK и функции API OpenOffice) можно создавать модули расширения и полнофункциональные приложения-компоненты.

Входные языки отражают объем и качество предоставляемых пакетом возможностей, а также удобство их использования. Таким образом, именно входной язык является основным показателем возможностей ППП. Однако, в современных пакетах обращение пользователя к языковым средствам обычно происходит косвенно, через графический интерфейс.

Предметное обеспечение отражает особенности решаемого класса задач из конкретной предметной области и включает:

- программные модули, реализующие алгоритмы (или их отдельные фрагменты) прикладных задач;
- средства сборки программ из отдельных модулей.

Информационные технологии в менеджменте

Наиболее распространено в настоящее время оформление программных модулей в виде библиотек, подключаемых статически или динамически. В зависимости от использованного разработчиками подхода к проектированию и реализации ППП такие библиотеки содержат встроенные классы и описания их интерфейсов (при использовании объектно-ориентированного программирования). При использовании парадигмы структурного программирования в библиотечных модулях содержатся процедуры и функции, предназначенные для решения некоторых самостоятельных задач. В обоих случаях библиотеки связаны с другими модулями пакета лишь входной и выходной информацией.

Системное обеспечение представляет собой совокупность низкоуровневых средств (программы, файлы, таблицы и т.д.), обеспечивающих определенную дисциплину работы пользователя при решении прикладных задач и формирующих окружение пакета. К системному обеспечению ППП относят следующие компоненты:

- монитор — программа, управляющая взаимодействием всех компонентов ППП;
- транслятор(ы) с входных языков — для ППП характерно использование интерпретируемых языков;
- средства доступа к данным — драйверы баз данных и/или компоненты, представляющие доступ через унифицированные интерфейсы (ODBC, JDBC, ADO, BDE и т.п.);
- информационно-справочный модуль — предоставляет функции поддержки, среди которых информационные сообщения, встроенная справочная системы и т.п.
- различные служебные программы, выполняющие низкоуровневые операции (автосохранение, синхронизация совместно используемых файлов и т.д.).

Приведенная логическая структура ППП достаточна условна и в конкретном ППП может отсутствовать четкое разделение программ на предметное и системное обеспечение. Например, программа планирования вычислений, относящаяся к прикладному обеспечению, может одновременно выполнять и ряд служебных функций (информационное обеспечение, связь с операционной системой и т.п.).

Кроме того, одни и те же программы в одном пакете могут относиться к предметному обеспечению, а в другом — к системному. Так, программы построения диаграмм в рамках специализированного пакета машинной графики естественно отнести к предметному обеспечению. Однако те же программы следует считать вспомогательными и относящимися к системному обеспечению, например, в пакете решения вычислительных задач.

2. Основные группы пакетов прикладных программ

ППП можно условно классифицировать по четырем основным группам:

- 1) общего назначения;
- 2) проблемно-ориентированные;
- 3) расширяющие функции ОС;
- 4) интегрированные.

ППП общего назначения — ориентированы на широкий круг пользователей в различных проблемных областях, позволяя автоматизировать наиболее часто используемые виды работ. К пакетам такого типа относятся обработка текстовой информации, деловой графики, электронных таблиц и т. д. При этом следует иметь в виду, что функции пакета одного типа могут перекрывать функ-

ции пакета другого типа. Однако каждый из этих пакетов все-таки в большей мере акцентирован на обработку информации определенного вида.

Проблемно-ориентированные ППП имеют относительно узкое применение, используя особые методы представления и обработки информации. Это пакеты для экспертных систем, математические пакеты, статистические, издательские системы, управления производством, работы бухгалтерии, обработки фотографий, финансовой деятельности, управления персоналом (кадрового учета), управления материальными запасами (учета материалов), учета и управления транспортными средствами и т. п.

Расширяющие функции ОС пакеты обеспечивают, в частности, сопряжение ЭВМ с унифицированными приборными интерфейсами, научными приборами и установками.

Интегрированные ППП объединяют основные функции ряда пакетов, как правило, общего назначения. В таких пакетах используются не только единый формат файлов, но и методы доступа к файлам отдельных пакетов, т. е. обеспечивается полная информационная совместимость составляющих совокупность пакетов. Современные интегрированные пакеты для массового применения включают такие основные функции, как текстовый редактор, электронные таблицы, деловую графику, системы управления базами данных и телекоммуникацию. Типичным средством данного класса является известный пакет MS Office.

Рассмотрим более подробно группы ППП.

1. Обработка текстовой информации различного типа и назначения составляет одну из наиболее массовых групп задач на ЭВМ. Пакеты данной группы являются одними из наиболее часто используемых на ПК и позволяют проводить набор основных процедур работы с текстовой информацией:

- создание и корректировка текстовых документов;
- компоновка текстовой и графической информации;
- форматирование выходных документов, предварительный их просмотр;
- синтаксический контроль и т. д.

Среди пакетов этой группы можно выделить три подгруппы:

- общего назначения (например, MS Word);
- специального назначения (например, News Master — осуществляет редактирование деловой и коммерческой информации, Easy Flow — используется для документирования технической документации);
- издательские системы (например, Page Maker, Quark XPress и др.).

2. Графические пакеты — предназначены для обработки различного рода и назначения графической и иной изобразительной информации (рисование, создание и использование библиотек рисунков различного назначения, черчение в двух и трех измерениях, создание иллюстраций и т. д.). Среди средств этой группы можно проводить довольно разветвленную классификацию, учитывая бурное развитие и многоаспектность использования графической информации. Наиболее доступными и популярными являются пакеты: Paint, Power Point, Freelance Graphics for Windows и целый ряд других, позволяющих проводить весьма разнообразную обработку графической информации для различных приложений. Профессионалам из редакций газет и журналов или рекламных агентств требуются более мощные редакторы типа Corel Draw или Adobe PhotoShop. Для автоматизации проектно-конструкторских работ фактически стандартом является

Информационные технологии в менеджменте

пакет AutoCAD, представляющий собой мощную систему автоматизированного проектирования (САПР) для создания и редактирования сложных графических объектов, черчения, в первую очередь промышленных приложений. Отметим, что освоение этих средств требует от пользователя большого трудолюбия и настойчивости.

3. Обработка информации табличного типа (электронные таблицы).

Электронная таблица — компьютерный эквивалент обычной двумерной таблице, в клетках (ячейках) которой записаны данные различных типов: тексты, даты, формулы, числа и другая информация. Электронная таблица — одна из самых распространенных и мощных информационных технологий для профессиональной работы с данными. Для управления электронной таблицей созданы специальные программные продукты — *табличные процессоры*. Идея создания электронной таблицы возникла у студента Гарвардского университета (США) Дэна Бриклина в 1979 году. Выполняя скучные вычисления экономического характера с помощью бухгалтерской книги, он и его друг Боб Франкстон, который разбирался в программировании, разработали первую программу электронной таблицы, названную ими VisiCalc. Среди средств данной группы в настоящее время хорошо зарекомендовали себя такие пакеты, как Super Calc, Quattro Pro, Excel, Lotus и др. В современных электронных таблицах Excel обеспечиваются такие основные операции, как:

- хранение в памяти и просмотр на экране дисплея таблиц большой размерности;
- размещение в клетках таблицы информации и операции (в том числе математические) над содержанием клеток;
- графическое отображение информации и многие другие.

Большинство электронных таблиц имеют средства создания графиков и диаграмм, средства их редактирования и включения их в нужное место. Кроме того, в них имеется большое число встроенных функций — математических, статистических, имеется возможность решать задачи условной и безусловной, дискретной и непрерывной оптимизации. Сервисные функции табличного процессора Excel позволяют проверять орфографию текста, осуществлять защиту данных от чтения или записи.

4. Системы управления базами данных (СУБД). СУБД позволяют хранить различного рода информацию, обеспечивая доступ к ней как локально, так и в сетевых режимах. В настоящее время данный класс средств достаточно многочислен, и только для IBM-совместимых ПК существует около 50 известных пакетов общего назначения, настольных СУБД, среди которых можно отметить такие, как FoxPro, Paradox, Access. Для управления большими базами данных и их эффективной обработки на серверах баз данных устанавливаются SQL Server, SQL Base, Oracle Server, Informix, SyBase и другие программные продукты.

5. Специальные математические пакеты (средства численных и символьных вычислений) ориентированы на достаточно массовое применение и обеспечивают решение задач в математике, технике, статистике и других областях. Одной из основных областей применения компьютеров и поныне являются математические и научно-технические расчеты. Само по себе появление компьютеров не упрощало математические расчеты, а лишь позволяло резко повысить скорость их выполнения и сложность решаемых задач. Пользователям компьютеров, прежде чем начинать такие расчеты, необходимо было изучить сами компьютеры, языки программирования и довольно сложные методы вычислений, применить и представить под свои цели программы для решения расчетных задач

на языках программирования. Поневоле инженеру, физику, химику и математику приходилось становиться программистом.

Такая практика изменилась лишь после появления интегрированных математических программных систем для научно-технических расчетов, наиболее распространенные из них: MatLAB, MathCAD, Mathematica, Maple, Statistika, пакет оптимизации QSB и т. д.

Большое количество подобных разработок свидетельствует о значительном интересе к ним во всем мире. Математики, физики, ученые, инженеры из других смежных отраслей науки давно мечтали о математически ориентированном языке программирования для записи алгоритмов решения математических и научно-технических задач в наиболее удобной, компактной и доступной для понимания форме. Для этого они пытались приспособить различные языки программирования высокого уровня. Но их попытки так и не увенчались успехом — программы на этих языках, увы, ничем не напоминали привычные математические и физические символы и формулы, с которыми все привыкли работать и с помощью которых описываются решения математических задач.

Математические пакеты такого рода являются системами, в которых описание решения математических задач задается с помощью привычных математических формул и знаков. Такой же вид имеют и результаты вычислений.

Как правило, они имеют удобный пользовательский интерфейс (совокупность средств общения с пользователем). У этих систем, как правило, имеются и эффективные средства типовой научной графики, они просты в применении и интуитивно понятны. Одним словом, эти системы ориентированы на массового пользователя, от ученика начальных классов до академика.

Особый интерес представляют встроенные в систему электронные книги, содержащие справки и примеры применения системы по ряду разделов математики, механики, физики, электротехники. Справки содержат математические формулы и иллюстрации.

По ряду пакетов этой группы — MathCAD, Mathematica, Maple, Statistika — издан целый ряд книг различного уровня и назначения, рассчитанных на различный круг пользователей. В этих книгах приведены достаточно подробные описания этих популярных пакетов, особенности их эксплуатации и применения для решения различного рода задач математического характера.

Заметим, что математические пакеты — ценный инструмент, но лишь в квалифицированных руках, и он не избавляет, а, наоборот, усиливает необходимость изучения математики. Действительно, освоение математических пакетов может создать у пользователя иллюзию освоения самой математики. Однако следует помнить, что инструмент не заменяет компетентность. Никакие красочные меню не освобождают пользователя от понимания сути математических методов, реализованных в таких пакетах.

6. Экспертные системы (ЭС). В настоящее время в обращении находится свыше 6000 ЭС и инструментальных средств для их построения, постоянно расширяется их рынок и ведутся интенсивные работы по совершенствованию их архитектуры и возможностям. При этом наибольший интерес представляют так называемые «оболочки» и «программные обстановки».

Под «*оболочками*» понимают «пустые» версии существующих экспертных систем, т.е. готовые экспертные системы без базы знаний. Достоинство оболочек в том, что они вообще не требуют работы программистов для создания готовой экспертной системы. Требуются только специалисты в предметной области и инженеры по знаниям для заполнения базы знаний. Примером таких оболочек

Информационные технологии в менеджменте

могут служить EMYCIN (Empty MYCIN — пустой MYCIN) и VP Expert, которые представляет собой незаполненные экспертные системы.

Программные обстановки для разработки экспертных систем представляют собой высокоуровневые системы поддержки автоматизированной разработки и создания проблемно-ориентированных ЭС. Они, как правило, содержат богатый спектр языков представления знаний. Разработчик экспертной системы, пользуясь набором разнообразных средств организации вывода в ЭС, может сконструировать свою проблемно-независимую оболочку. Наиболее известной программной обстановкой является система KEE. Система имеет большие возможности, обладает средствами тестирования и отладки знаний, прекрасным пользовательским интерфейсом и графикой. Система Nexpert Object является мощным средством разработки ЭС, способным конкурировать с такими системами, как Gold Work и KEE, и имеющим подобную организацию базы знаний. Система работает в среде MS Windows, имеет интерфейс с языками программирования, хорошие возможности работы с базами данных.

7. Коммуникационные ППП предназначены для организации взаимодействия пользователя с удаленными абонентами или информационными ресурсами сети. Так, в условиях развития глобальной информационной сети Internet, идеология которой сейчас используется в локальных сетях, появился класс программного обеспечения — браузеры, средства создания WWW страниц и др.

8. Пакеты, реализующие CASE (Computer Aided Software Engineering) -технологии применяются при создании сложных информационных систем, обычно требующих коллективной разработки проекта, в котором участвуют различные специалисты: системные аналитики, проектировщики и программисты.

Под CASE-технологией понимается совокупность средств автоматизации разработки информационной системы, включающей в себя методологию анализа предметной области, проектирования, программирования и эксплуатации информационных систем.

Инструментальные средства CASE-технологии применяются на всех этапах жизненного цикла системы (от анализа и проектирования до внедрения и сопровождения), значительно упрощая решение возникающих задач.

CASE-технология позволяет отделить проектирование информационной системы от собственно программирования и отладки: разработчик системы занимается проектированием на более высоком уровне, не отвлекаясь на детали. Это позволяет не допускать ошибок уже на стадии проектирования и получить более совершенные программные продукты.

Коллективная работа над проектом предполагает обмен информацией, контроль выполнения задач, отслеживание изменений и версий, планирование, взаимодействие и управление. Фундаментом реализации подобных функций чаще всего служит общая база данных проекта, которую обычно называют *репозитарием*.

По существу, репозитарий — это информационный архив, где хранятся сведения о процессах, данных и связях объектов в разрабатываемом приложении.

В различных CASE-технологиях репозитарий реализуется по-разному и может содержать описания и модели данных, а также правила их обработки. Репозитарий является важнейшим компонентом набора инструментальных средств CASE и служит источником информации, необходимой для автоматизации построения проектируемой системы и генераций приложений.

9. Интегрированные ППП — набор нескольких программных продуктов, функционально дополняющих друг друга, поддерживающих единые информационные технологии, реализованные на общей вычислительной и операционной платформе. Организационно такие пакеты представляют собой либо логическое объединение пакетов под управлением некоторого координирующего их работу модуля, либо выполняются в виде интегрированной системы функций пакетов общего назначения.

На заре эры персональных компьютеров каждое из приложений представляло собой независимую программу. Обычно пользователь покупал программу обработки текстов у одной компании, программу обработки электронных таблиц — у другой, а систему управления базой данных — у третьей. Вероятность того, что эти приложения имеют одинаковый интерфейс и функционируют на базе одинаковых принципов, была низкой, что затрудняло изучение каждой новой программы. Причина такой обособленности приложений отчасти заключалась в самой однозадачной операционной системе, в которой она работала, — MS DOS.

Современные процессоры и операционные системы обладают возможностями реальной многозадачности — они позволяют создавать среду, в которой одновременно могут функционировать несколько приложений, совместно используя микропроцессор. Многозадачность позволила создавать пакеты, которые состоят из программ, имеющих одинаковый интерфейс, и основаны на одних и тех же принципах работы.

Современные интегрированные пакеты для массового применения включают следующие основные функции:

- текстовый редактор;
- электронные таблицы;
- деловую графику;
- СУБД;
- телекоммуникацию.

Компоненты интегрированных пакетов могут работать изолированно друг от друга, но основные достоинства интегрированных пакетов проявляются при их разумном сочетании друг с другом. Пользователи интегрированных пакетов имеют унифицированный для различных компонентов интерфейс, тем самым обеспечивается относительная легкость процесса их освоения. Типичный пример интегрированного пакета — MS Office, который включает в себя:

- Word — приложение для обработки текстов;
- Excel — программа обработки электронных таблиц;
- Access — программа обработки баз данных;
- Outlook — комбинация программ обработки электронной почты и менеджмента личной информации;
- PowerPoint, позволяющий создавать презентации для выставок, продаж, семинаров или для случаев, когда необходимо представлять информацию группе людей с применением графики, текста и диаграмм.

Имеются, конечно, и другие типы прикладных программ; например, имеется большое количество пакетов, поддерживающих автоматизированные рабочие места (АРМ) самого различного назначения: для научных работников, банковских служащих, бухгалтеров, сотрудников налоговой службы, управления персоналом и т. д.

Информационные технологии в менеджменте

Так, интегрированный пакет Scientific Work Place (SWP) — «рабочее место ученого» удачно объединяет несколько полезных инструментов:

- подобный Word текстовый редактор;
- текстовый редактор научных документов с возможностью набора математических зависимостей в привычной для ученого форме;
- подмножество пакета Maple V, позволяющее выполнять аналитические преобразования и численные расчеты.

На рынке программных продуктов имеется масса и других программных средств, в частности, мультимедиа, офисных ППП: органайзеров (планировщиков), программ-переводчиков, средств проверки орфографии и распознавания текста и т. п.

3. Эволюция ППП

Первые пакеты прикладных программ представляли собой простые тематические подборки программ для решения отдельных задач в той или иной предметной области, обращение к ним выполнялось с помощью средств оболочки ОС или из других программ. Современный пакет является сложной программной системой, включающей специализированные системные и языковые средства. В относительно короткой истории развития вычислительных ППП можно выделить *4 основных поколения* (класса) пакетов. Каждый из этих классов характеризуется определенными особенностями входящих в состав ППП компонентов — входных языков, предметного и системного обеспечения.

Первое поколение

В качестве входных языков ППП первого поколения использовались универсальные языки программирования (Фортран, Алгол-60 и т.п.) или языки управления заданиями соответствующих операционных систем. **Проблемная ориентация входных языков достигалась за счет соответствующей мнемоники в идентификаторах.** Составление заданий на таком языке практически не отличалось от написания программ на алгоритмическом языке. Предметное обеспечение первых ППП, как правило, было организовано в форме библиотек программ, т.е. в виде наборов (пакетов) независимых программ на некотором базовом языке программирования (отсюда впервые возник и сам термин «пакет»). Такие ППП иногда называют *пакетами библиотечного типа*, или *пакетами простой структуры*.

В качестве системного обеспечения пакетов первого поколения обычно использовались штатные компоненты программного обеспечения ЭВМ: компиляторы с алгоритмических языков, редакторы текстов, средства организации библиотек программ, архивные системы и т.д. Эти пакеты не требовали сколько-нибудь развитой системной поддержки, и для их функционирования вполне хватало указанных системных средств общего назначения. В большинстве случаев разработчиками таких пакетов были прикладные программисты, которые пытались приспособить универсальные языки программирования к своим нуждам.

Второе поколение

Разработка ППП второго поколения осуществлялась уже с участием системных программистов. Это привело к появлению специализированных входных языков на базе универсальных языков программирования. **Проблемная ориентация таких языков достигалась не только за счет использования определенной мнемоники, но также применением соответствующих языковых конструкций, которые упрощали формулировку задачи и делали ее более наглядной.** Транслятор с такого языка представлял собой препроцессор (чаще

Информационные технологии в менеджменте

всего макропроцессор) к транслятору соответствующего алгоритмического языка. В качестве модулей в пакетах этого класса стали использоваться не только программные единицы (т.е. законченные программы на том или ином языке программирования), но и такие объекты, как последовательность операторов языка программирования, совокупность данных, схема счета и др.

Существенные изменения претерпели также принципы организации системного обеспечения ППП. В достаточно развитых пакетах второго поколения уже можно выделить элементы системного обеспечения, характерные для современных пакетов: монитор, трансляторы с входных языков, специализированные банки данных, средства описания модели предметной области и планирования вычислений и др.

Третье поколение

Третий этап развития ППП характеризуется появлением самостоятельных входных языков, ориентированных на пользователей-непрограммистов. Особое внимание в таких ППП уделяется системным компонентам, обеспечивающим простоту и удобство. Это достигается главным образом за счет специализации входных языков и включения в состав пакета средств автоматизированного планирования вычислений.

Интероперабельность ППП

Расширение сферы применения вычислительной техники привело к появлению разнообразных цифровых устройств (от настольных систем и нетбуков, до смартфонов и бытовой техники) на различных аппаратно-программных платформах. Эта ситуация, с одной стороны, обострила вопрос, возникший еще у разработчиков вычислительных систем первого-второго поколений: как обеспечить работу программ при смене платформы? А с другой — создала проблему для конечных пользователей: как использовать необходимые им ППП, созданные для другой системы и при этом не менять платформу? То есть как, например, заставить работать приложения для андроид под управлением iOS или как запустить MS Office под Linux'ом.

Решением проблемы обеспечения *интероперабельности* (переносимости) ПО является концепция *открытых систем*. Такая система разрабатывается с использованием *открытых стандартов и спецификаций*. Эта концепция все чаще используется разработчиками прикладного программного обеспечения и существенно облегчает жизнь пользователям. Примерами такого ПО являются офисный пакет LibreOffice, веб-браузер Firefox и всевозможные веб-сервисы.

Четвертое поколение

Четвертый этап характеризуется созданием ППП, эксплуатируемых в интерактивном режиме работы. Основным преимуществом диалогового взаимодействия с ЭВМ является возможность активной обратной связи с пользователем в процессе постановки задачи, ее решения и анализа полученных результатов. Появление и интенсивное развитие различных форм диалогового общения обусловлено, прежде всего, прогрессом в области технических средств (графическая подсистема ЭВМ и средства мультимедиа, сетевые средства).

Развитие аппаратного обеспечения повлекло за собой создание разнообразных программных средств поддержки диалогового режима работы (диалоговые операционные системы, диалоговые пакеты программ различного назначения и т. д.). Прикладная система состоит из *диалогового монитора* — набора универсальных программ, обеспечивающих ведение диалога и обмен данными, и базы знаний об области. Информация о структуре, целях и форма диалога задает сценарий, в соответствии с которым монитор управляет ходом

Информационные технологии в менеджменте

диалога. Носителями процедурных знаний о предметной области являются прикладные модули, реализующие функции собственной системы.

Таким образом, создание прикладной системы сводится к настройке диалогового монитора на конкретный диалог, путем заполнения базы знаний. При этом программировать в традиционном смысле этого слова приходится лишь прикладные модули, знания о диалоге вводятся в систему с помощью набора соответствующих средств — редактора сценариев. Логично требовать, чтобы редактор сценариев также представлял собой диалоговую программу, отвечающую рассмотренным выше требованиям. Благодаря готовому универсальному монитору программист может сосредоточиться на решении чисто прикладных задач, выделение же знаний о диалоге в сценарий обеспечивает в значительной степени необходимая гибкость программного продукта.

Большое внимание в настоящее время уделяется проблеме создания «интеллектуальных ППП». Такой пакет позволяет конечному пользователю лишь сформулировать свою задачу в содержательных терминах, не указывая алгоритма ее решения. Синтез решения и сборка целевой программы производятся автоматически. При этом детали вычислений скрыты от пользователя, и компьютер становится интеллектуальным партнером человека, способным понимать его задачи. Предметное обеспечение подобного ППП представляет собой некоторую базу знаний, содержащую как процедурные, так и описательные знания. Такой способ решения иногда называют *концептуальным программированием*, характерными особенностями которого является программирование в терминах предметной области использование ЭВМ уже на этапе постановки задач, автоматический синтез программ решения задачи, накопление знаний о решаемых задачах в базе знаний.

Лекция № 2

Ознакомление с пакетом Mathcad

1. Основные возможности пакета Mathcad
2. Сравнительная характеристика пакета Mathcad с аналогичными системами
3. Расширение функциональности пакета Mathcad
4. Взаимодействие Mathcad с другими программами
5. Структура Mathcad-документа и состав панели инструментов Math (Математика)

1. Основные возможности пакета Mathcad

Mathcad — система визуальных математических расчетов. Основная идея Mathcad состоит в том, что вычисляемые выражения записываются в визуальной форме, максимально приближенной к математической записи.

В самом общем смысле Mathcad можно рассматривать как программный пакет, объединяющий в себе: мощную вычислительную среду; гибкий полнофункциональный текстовый процессор.

Mathcad был задуман и первоначально написан Алленом Раздовом из Массачусетского технологического института (MIT), соучредителем компании Mathsoft, которая с 2006 года является частью корпорации PTC (Parametric Technology Corporation).

Mathcad имеет интуитивный и простой для использования интерфейс пользователя. Для ввода формул и данных можно использовать как клавиатуру, так и специальные панели инструментов.

Некоторые из математических возможностей Mathcad (версии до 13.1 включительно) основаны на подмножестве системы компьютерной алгебры Maple (MKM, Maple Kernel Mathsoft). Начиная с 14 версии — использует символьное ядро MuPAD.

Работа осуществляется в пределах рабочего листа, на котором уравнения и выражения отображаются графически, в противовес текстовой записи в языках программирования. При создании документов-приложений используется принцип WYSIWYG (*What You See Is What You Get* — «что видишь, то и получаешь»).

Возможности системы:

1. *Числовые расчеты* со скалярами, матрицами и векторами (матрицами из одного столбца). Возможны расчеты с использованием комплексных чисел.

2. *Аналитические преобразования:* интегрирование, дифференцирование, вычисление пределов, сумм и произведений рядов, упрощение, преобразования Лапласа и Фурье и др.

3. *Определение законов вычисления элементов матриц*, что позволяет реализовать итерационные вычисления, в том числе по рекуррентным формулам.

4. Работа со *стандартными функциями:* интерполяция, экстраполяция, численное интегрирование, матричные функции и др.

5. *Определение своих функций.*

6. *Построение двумерных и трехмерных графиков* различных видов.

7. *Решение систем* линейных и нелинейных уравнений.

8. *Решение оптимизационных задач* вида: найти значения переменных, при которых функция принимает минимальное или максимальное значение.

9. *Решение дифференциальных уравнений* (обыкновенные дифференциальные уравнения и системы уравнений; уравнения Пуассона и Лапласа).

10. *Элементы программирования.*

2. Сравнительная характеристика пакета Mathcad с аналогичными системами

По назначению.

Mathcad относится к [системам компьютерной алгебры](#), то есть средств [автоматизации](#) математических расчетов. В этом классе [программного обеспечения](#) существует много аналогов различной направленности и принципа построения. Наиболее часто Mathcad сравнивают с такими программными комплексами, как [Maple](#), [Mathematica](#), [MATLAB](#), а также с их аналогами [MuPAD](#), [Scilab](#), [Maxima](#) и др. Впрочем, объективное сравнение осложняется в связи с разным назначением программ и идеологией их использования.

Система Maple, например, предназначена главным образом для выполнения аналитических (символьных) вычислений и имеет для этого один из самых мощных в своем классе арсенал специализированных процедур и функций (более 3000). Такая комплектация для большинства пользователей, которые сталкиваются с необходимостью выполнения математических расчетов среднего уровня сложности, является избыточной. Возможности [Maple](#) ориентированы на пользователей — профессиональных [математиков](#); решения задач в среде Maple требуют не только умения оперировать какой-либо функцией, но и знания методов решения, в нее заложенных: во многих встроенных функциях Maple фигурирует аргумент, задающий метод решения.

То же самое можно сказать и о [Mathematica](#). Это одна из самых мощных систем, имеет чрезвычайно большую функциональную наполненность (есть даже синтезирование [звука](#)). [Mathematica](#) обладает высокой скоростью вычислений, но требует изучения довольно необычного [языка программирования](#).

Разработчики Mathcad сделали ставку на расширение системы в соответствии с потребностями пользователя. Для этого назначены дополнительные библиотеки и пакеты расширения, которые можно приобрести отдельно и которые имеют дополнительные функции, встраиваемые в систему при установке, а также электронные книги с описанием методов решения специфических задач, с примерами действующих алгоритмов и документов, которые можно использовать непосредственно в собственных расчетах. Кроме того, в случае необходимости и при условии наличия навыков программирования в C, есть возможность создания собственных функций и их прикрепления к ядру системы через механизм DLL.

Mathcad, в отличие от Maple, изначально создавался для численного решения математических задач, он ориентирован на решение задач именно прикладной, а не теоретической математики, когда нужно получить результат без углубления в математическую суть задачи. Впрочем, для тех, кому нужны символьные вычисления и предназначено интегрированное ядро Maple (с версии 14 — MuPAD). Особенно это полезно, когда речь идет о создании документов образовательного назначения, когда необходимо продемонстрировать построение математической модели, исходя из физической картины процесса или явления. Символьное ядро Mathcad, в отличие от оригинального Maple (MuPAD), искусственно ограничено (доступно около 300 функций), но этого в большинстве случаев вполне достаточно для решения задач инженерного характера.

Информационные технологии в менеджменте

Более того, опытные пользователи Mathcad обнаружили, что в версиях до 13 включительно есть возможность не слишком сложным способом задействовать почти весь функциональный арсенал ядра Maple (так называемые «недокументированные возможности»), что приближает вычислительную мощность Mathcad к Maple.

По типу интерфейса.

Основное отличие Mathcad от аналогичных программ — это графический, а не текстовый режим ввода выражений. Для набора команд, функций, формул можно использовать как клавиатуру, так и кнопки на многочисленных специальных панелях инструментов. В любом случае — формулы будут иметь привычный, аналогичный книжному, вид. То есть особой подготовки для набора формул не нужно. Вычисления с введенными формулами осуществляются по желанию пользователя или мгновенно, одновременно с набором, либо по команде. Обычные формулы вычисляются слева направо и сверху вниз (подобно чтению текста). Любые переменные, формулы, параметры можно изменять, наблюдая воочию соответствующие изменения результата. Это дает возможность организации действительности интерактивных вычислительных документов.

В других программах (Maple, MuPAD, Mathematica) вычисления осуществляются в режиме программного [интерпретатора](#), который трансформирует в формулы, введенные в виде текста команды. Maple своим интерфейсом ориентирован на тех пользователей, кто уже имеет навыки программирования в среде традиционных языков с введением сложных формул в текстовом режиме. Для пользования Mathcad можно вообще не быть знакомым с [программированием](#) в том или ином виде.

Mathcad задумывался как средство программирования без программирования, но, если возникает такая потребность — Mathcad имеет довольно простые для усвоения инструменты программирования, позволяющие, впрочем, строить весьма сложные алгоритмы, к чему прибегают, когда встроенных средств решения задачи не хватает, а также когда необходимо выполнять серийные расчеты.

Отдельно следует отметить возможность использования в расчетах Mathcad величин с размерностями, причем можно выбрать [систему единиц](#): [СИ](#), [СГС](#), [МКС](#), [английскую](#), или построить собственную. Результаты вычислений, разумеется, также получают соответствующую размерность. Пользу от такой возможности трудно переоценить, поскольку значительно упрощается отслеживание ошибок в расчетах, особенно в физических и инженерных.

По механизму визуализации данных (графикам).

В среде Mathcad фактически нет [графиков функций](#) в математическом понимании термина, а есть визуализация данных, находящихся в векторах и матрицах (то есть осуществляется построение как линий, так и поверхностей по точкам с [интерполяцией](#)), хотя пользователь может об этом и не знать, поскольку у него есть возможность использования непосредственно функций одной или двух переменных для построения графиков или поверхностей соответственно. Так или иначе, механизм визуализации Mathcad значительно уступает таковому у Maple, где достаточно иметь только вид функции, чтобы построить график или поверхность любого уровня сложности. По сравнению с Maple, графика Mathcad имеет еще такие недостатки, как: невозможность построения поверхностей, заданных параметрически, с непрямоугольной областью определения двух параметров; создание и форматирование графиков только через меню, что ограничивает возможности программного управления параметрами графики.

Информационные технологии в менеджменте

Однако следует помнить об основной области применения Mathcad — для задач инженерного характера и создания учебных интерактивных документов возможностей визуализации вполне достаточно. Опытные пользователи Mathcad демонстрируют возможность визуализации сложнейших математических конструкций, но объективно это уже выходит за рамки назначения пакета.

3. Расширение функциональности пакета Mathcad

Возможно дополнение Mathcad новыми возможностями с помощью специализированных пакетов расширений и библиотек, которые пополняют систему дополнительными функциями и константами для решения специализированных задач:

- *Пакет для анализа данных (Data Analysis Extension Pack)* — обеспечивает Mathcad необходимыми инструментами для анализа данных.
- *Пакет для обработки сигналов (Signal Processing Extension Pack)* — содержит более 70 встроенных функций для аналоговой и цифровой обработки сигналов, анализа и представления результатов в графическом виде.
- *Пакет для обработки изображений (Image Processing Extension Pack)* — обеспечивает Mathcad необходимыми инструментами для обработки изображений, анализа и визуализации.
- *Пакет для работы с функциями волнового преобразования (Wavelets Extension Pack)* — содержит большой набор дополнительных вейвлет-функций, которые можно добавить в библиотеку встроенных функций базового модуля Mathcad Professional. Пакет предоставляет возможность применить новый подход к анализу сигналов и изображений, статистической оценки сигналов, анализа сжатия данных, а также специальных численных методов. Функциональность включает одно- и двумерные вейвлеты, дискретные вейвлет-преобразования, мультианализ разрешения и многое другое. Пакет объединяет более 60 функций ключевых вейвлетов. Включены ортогональные и биортогональные семейства вейвлетов, среди прочего — вейвлет Хаара, вейвлет Добеши, симлет, койфлет и В-сплайны. Пакет также содержит обширную диалоговую документацию по основным принципам вейвлетов, приложения, примеры и таблицы ссылок.
- *Библиотека строительства (Civil Engineering Library)* — включает справочник англ. Roark's Formulas for Stress and Strain (Формулы Роарка для расчета напряжений и деформаций), настраиваемые шаблоны для строительного проектирования и примеры тепловых расчетов.
- *Электротехническая библиотека (Electrical Engineering Library)* — содержит стандартные вычислительные процедуры, формулы и справочные таблицы, используемые в электротехнике. Текстовые пояснения и примеры облегчают работу с библиотекой — каждый заголовок имеет гиперссылку на оглавление и указатель, и его можно найти в системе поиска.
- *Библиотека машиностроения (Mechanical Engineering Library)* — включает справочник *Roark's Formulas for Stress and Strain* (Формулы Роарка для расчета напряжений и деформаций), содержащий более пяти тысяч формул, вычислительные процедуры из справочника McGraw-Hill и метод конечных элементов. Текстовые пояснения, поисковая система и примеры облегчают работу. В состав библиотеки включена электронная книга Дэвида Пинтура «Введение в метод конечных элементов».

4. Взаимодействие Mathcad с другими программами

Mathcad интегрируется с программами [SmartSketch](#), [VisSim/ Comm PE, Pro/ENGINEER](#).

Приложение SmartSketch позволяет [инженерам](#), [дизайнерам](#), [архитекторам](#), [чертежникам](#), [системным](#) и [сетевым администраторам](#) работать с точными чертежами и графиками.

VisSim/Comm PE — это Windows-приложение для моделирования аналоговых, цифровых или смешанных систем сообщения на сигнальном или физическом уровне.

Также в документах-программах Mathcad есть возможность *вставки модулей (component)* других приложений для расширения возможностей визуализации, анализа данных, выполнение специфических вычислений.

Для расширенной визуализации данных предназначен компонент [Axum Graph](#). Для работы с табличными данными — [Microsoft Excel](#).

Компоненты [Data Acquisition](#), [ODBC](#) Input позволяют пользоваться внешними [базами данных](#).

Предлагаются также бесплатные модули (add-in) для интеграции Mathcad с программами Excel, [AutoCAD](#).

Для статистического анализа предназначен компонент [Axum S-PLUS Script](#).

Значительное расширение возможностей пакета достигается при интеграции со сверхмощным приложением MATLAB.

5. Структура Mathcad-документа и состав панели инструментов Math (Математика)

При запуске системы Mathcad появляется основное окно, которое содержит три панели (стандартную, форматирования текста, математическую) и рабочую область с автоматически созданным листом вычислений. Включать и отключать панели можно из верхнего меню в разделе **View** → **Toolbars (Вид** → **Панели инструментов)**.

MathCAD-документ представляет собой совокупность областей типа: Формула, График, Текст. Каждая область имеет форму прямоугольника и может размещаться в любом месте рабочего листа. Для позиционирования областей используется курсор (красного цвета), имеющий крестообразную форму. В текстовой области курсор имеет форму вертикальной черты красного цвета, а в области типа формула и график — форму уголка синего цвета, называемый *следом курсора формул*. Перемещение курсора выполняется с помощью клавиш управления курсором или перемещением указателя мыши с последующим однократным щелчком ее левой кнопки. Область, которая является *активной* или *текущей* (создается, редактируется), выделяется прямоугольной рамкой. Рамка исчезает, как только курсор выводится из области.

Вставка текста и рисунков. Чтобы *вставить текст*, необходимо щелкнуть мышью в той точке листа, где необходим текст, и выполнить команду верхнего меню **Insert** → **Text Region (Вставка** → **Текстовый регион)**. Появится рамка, в которую можно ввести текст. При этом иногда возникает необходимость сменить шрифт для правильного отображения русских букв — это можно сделать с помощью списка шрифтов на панели форматирования.

Вставку рисунка проще всего осуществить через буфер обмена. Предварительно необходимо поместить изображение в буфер (например, с

Информационные технологии в менеджменте

помощью графического редактора Paint), затем вернуться в Mathcad и выполнить пункт верхнего меню **Edit** → **Paste** (**Правка** → **Вставить**).

Mathcad может одновременно работать с несколькими листами. Создание нового листа осуществляется из верхнего меню — **File** → **New** (**Файл** → **Новый**). Загрузить уже имеющийся лист — **File** → **Open** (**Файл** → **Открыть**). Сохранить созданный лист — **File** → **Save** (**Файл** → **Сохранить**). Переключение между загруженными листами — комбинация клавиш Ctrl+F6 или на панели инструментов **Окно**.

На панели **Math** (**Математика**) находятся кнопки, при нажатии на которые открываются дополнительные панели с шаблонами ввода различных выражений:

Кнопка	Панель инструментов
	Калькулятор (Calculator): арифметические операторы.
	График (Graph): двух- и трехмерные типы графиков и графические инструменты.
	Матрица (Matrix): матричные и векторные операторы.
	Выражения (Evaluation): знак равенства при вычислениях и в определении.
	Математический анализ (Calculus): производные, интегралы, пределы, а также итерируемые суммы и произведения.
	Булева алгебра (Boolean): операторы сравнения и логические операторы для булевых выражений.
	Программирование (Programming): программные конструкции.
	Греческие символы (Greek): буквы греческого алфавита.
	Символика (Symbolic): Символьные ключевые слова и модификаторы.

Лекция № 3

Mathcad: Выражения, стандартные функции и функции, определяемые пользователем

1. **Выражения и стандартные функции**
2. **Определение переменных (присвоение значений переменным)**
3. **Размерности физических величин**
4. **Функции, определяемые пользователем**
5. **Запись логических выражений. Функция if**
6. **Переменные диапазона (ранжированные переменные)**

1. Выражения и стандартные функции

Выражения в Mathcad представляют собой совокупность имен переменных, чисел, функций, соединенных знаками арифметических и/или логических

операторов. Скобки в записи выражения позволяют изменить принятый по умолчанию порядок выполнения действий, а также служат для указания аргументов функций.

Любая формула может быть записана в любом месте листа. Необходимо щелкнуть мышью в предполагаемой точке ввода формулы, там должен появиться указатель ввода — крест. Во время ввода или редактирования формулы Mathcad обводит ее рамкой (рис. 1).

Рис. 1. Формула на листе Mathcad в рамке ввода

Смена уровня ввода формул. Наиболее частая ошибка заключается в том, что в процессе ввода формулы пользователь неосторожно выходит из указанной рамки ввода (при этом рамка исчезает и в рабочей области появляется указатель ввода — крест) и все равно продолжает ввод. При этом на листе вместо одной обычно оказываются две формулы, каждая из которых — лишь часть требуемой. Проблема в том, что визуально эти две формулы могут находиться близко друг к другу и выглядеть как единое целое, но это не так. Единственный способ проверить целостность формулы — щелкнуть по ней мышью. Если все правильно, то в рамке ввода окажется вся формула.

Например, необходимо ввести выражение вида $\frac{a+b}{2}$. После ввода последовательности «**a+b**» формула приобретет вид $\frac{a+b}{}$. Курсор подчеркивает символ «**b**», поэтому если сразу ввести символ деления и затем цифру, то будет

введено выражение $a + \frac{b}{2}$. Поэтому после ввода «**a+b**» следует перевести курсор на более низкий уровень, нажав клавишу «пробел». Тогда выражение принимает вид $\frac{a+b}{}$, и если далее ввести символ деления и цифру, то будет получена требуемая формула.

На рис. 2 приведены примеры аналитического и числового вычислений. Левая часть, включая знаки вычисления, вводится пользователем, правая часть — результат, вычисленный системой Mathcad.

Рис. 2. Фрагмент листа Mathcad с вычисленными выражениями

При наборе выражения можно последовательно отменять произведенные действия, нажимая Ctrl+Z. Однако следует учесть, что в младших версиях Mathcad (до Mathcad 11) отмена возможна только до выхода из рамки ввода. Закончить ввод формулы можно, либо нажав Enter или Tab, либо щелкнув мышью где-нибудь вне рамки ввода формулы.

Если при вводе выражения произошла какая-либо ошибка, то ошибочный фрагмент будет автоматически выделен красным цветом.

Информационные технологии в менеджменте

Для *удаления формулы* достаточно выделить ее (например, щелкнув мышью) и нажать Ctrl+D или Delete. Для *перемещения формулы* необходимо выделить ее и, ухватившись мышью за появившуюся рамку ввода, переместить. Можно одновременно перемещать блок формул, который необходимо предварительно выделить методом протягивания.

Основные операторы, которые используются для записи формул, перечислены в [таблицах 1.4 и 1.5](#).

Таблица 1.4. Арифметические операторы

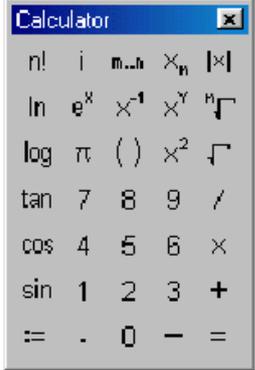
Операция	Запись	Клавиша (комбинация клавиш)	Панель инструментов Math
Сложение, вычитание	$x + y, x - y$	+, -	
Умножение	$x \cdot y$	*	
Деление	x / y	/	
Возведение в степень	x^y	^	
Факториал	$x!$!	
Квадратный корень	\sqrt{x}	\	
Корень n-й степени	$\sqrt[n]{x}$	[Ctrl+\]	
Модуль числа	$ x $		
Скобки	(x)	' (апостроф)	

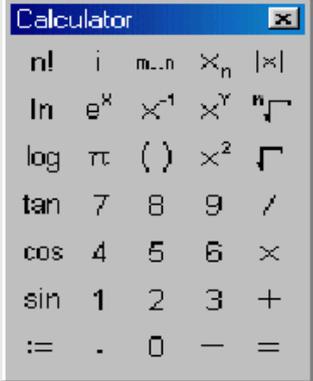
Таблица 1.5. Логические операторы и операторы сравнения

Операция	Запись	Клавиша (комбинация клавиш)	Панель инструментов Math
Равно	$x = y$	Ctrl+ =	
Не равно	$x \neq y$	Ctrl+ ≠	
Логическое отрицание	$\neg x$	Ctrl+Shift+1	
Логическое и	$x \wedge y$	Ctrl+Shift+7	
Логическое или	$x \vee y$	Ctrl+Shift+6	
Меньше	$x < y$	<	
Больше	$x > y$	>	
Меньше или равно	$x \leq y$	Ctrl+ ≤	
Больше или равно	$x \geq y$	Ctrl+ ≥	

Оператор, для которого требуется ввести один операнд, называют *унарным* (например, корень квадратный, модуль, факториал). Если же оператор служит для обработки двух операндов, то его называют *бинарным* (например, сложение, вычитание, умножение, деление).

В [таблицах 1.6 и 1.7](#) перечислены некоторые функции Mathcad.

Таблица 1.6. Основные встроенные математические функции

Название функции	Запись в MathCAD	Панель инструментов Math
Тригонометрические: sin, cos, tg, ctg, sec, cosec	sin(x), cos(x), tan(x), cot(x), sec(x), csc(x)	
Обратные тригонометрические: arcsin, arcos, arctg, arcctg, arcsec, arcsc	asin(x), acos(x), atan(x), acot(x), asec(x), acsc(x)	
Экспонента, e^x	e^x , exp(x)	
Натуральный логарифм, ln	ln(x)	
Десятичный логарифм, lg	log(x)	

log(X,b) — логарифм от X по основанию b .

sinh(X), cosh(X), tanh(X), coth(X), sech(X), csch(X) — гиперболические синус, косинус, тангенс, котангенс, секанс, косеканс от X , аргументы указываются в радианах.

asin(z), acos(z), atan(z), acot(z), asec(z), acsc(z) — соответственно арксинус, арккосинус, арктангенс, арккотангенс, арксеканс, арккосеканс от z .

Функции для работы с комплексными числами:

Re(Z), Im(Z) — соответственно вещественная и мнимая части комплексного числа Z ;

arg(z) — аргумент комплексного числа z (в радианах).

Элементы статистического анализа данных:

gmean(G1,G2,G3...) — среднее геометрическое аргументов;

mean(G1,G2,G3...) — среднее арифметическое аргументов;

var(G1,G2,G3...) — дисперсия;

stdev(G1,G2,G3...) — среднеквадратичное отклонение.

Таблица 1.7. Прочие функции MathCAD

Назначение функции	Запись в MathCAD	Пример
Наибольшее целое, меньшее или равное x	floor(x)	floor(5.6)=5, floor(-3.56)=-4
Наименьшее целое, большее или равное x	ceil(x)	ceil(5.6)=6, ceil(-3.56)=-3
Угол в радианах между осью OX и радиус-вектором (x, y), значение угла лежит в промежутке $[0,2\pi]$	angle(x,y)	angle(1,1)=0.785
Остаток от целочисленного деления m на n	mod(m,n)	mod(45,7)=3

Ввод функции может быть выполнен с помощью алфавитно-цифровой клавиатуры, или (наиболее распространенных) — по нажатию соответствующей кнопки панели инструментов **Calculator**, или с помощью мастера функций (выводится полный список функций, объединенных по категориям). Последний режим ввода активизируется по команде вставки функции в документ

Insert → **Function** (Вставка → Функция), когда выбор требуемой функции осуществляется в диалоговом окне (рис. 1.5).

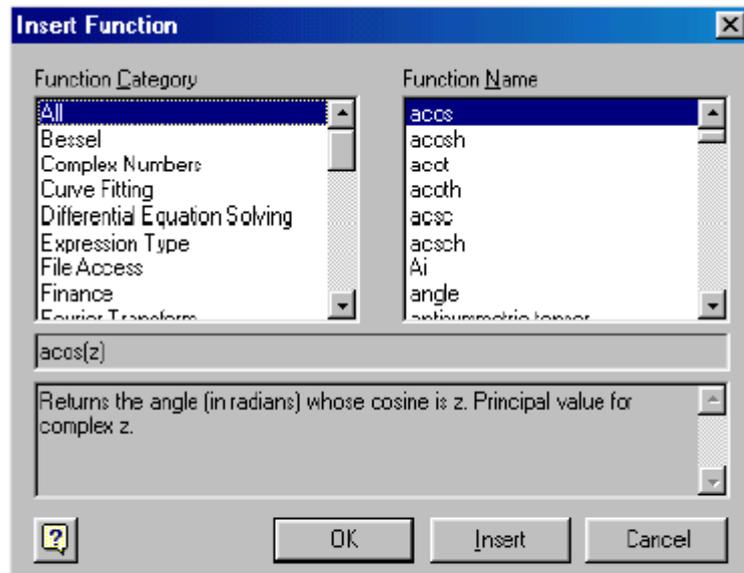


Рис. 1.5. Диалоговое окно вставки функции

2. Определение переменных (присвоение значений переменным)

Как и в языках программирования, в Mathcad различают:

- локальные переменные,
- глобальные переменные.

В обоих случаях определение переменной означает присваивание ей значения. В качестве оператора присваивания используется обозначение $:=$ для локального и \equiv для глобального присваивания.

Для ввода операторов используют символы на панели инструментов **Evaluation**. Слева от оператора должна находиться определяемая переменная. Справа — выражение, имеющее определенное значение, которое присваивается переменной. Например, с помощью следующего оператора $x := 5.6$ локальной переменной x присваивается значение 5,6. Глобальная переменная определяется аналогично. Например, $x \equiv 3.56$.

Если ввести оператор без задания имени справа, то на экран в зависимости от вида оператора присваивания будет выведена одна из приведенных ниже конструкций:

$$\blacksquare := \blacksquare \qquad \blacksquare \equiv \blacksquare$$

Теперь останется только заполнить знакоместа слева и справа оператора, указанные черными квадратиками.

Различие между локальными и глобальными переменными заключается в процессе, посредством которого Mathcad производит вычисления в документе. Mathcad читает документ **слева направо и сверху вниз** несколько раз. Во время «первого чтения» вычисляются значения глобальных переменных, в последующем чтении, когда происходит вычисление выражения, все переменные анализируются и интерпретируются. Таким образом, глобальные переменные могут использоваться в любом месте документа, в том числе слева от их определения или над ним. Для локальных переменных должно соблюдаться правило логической последовательности — определение переменной слева на одной

высоте или выше вычисляемого выражения, в котором используется данная переменная.

Вычисление результата. После ввода выражения и знака «=» справа будет показан его результат. Таким же образом можно выводить на экран значения переменных, вектора, матрицы, таблицы значений функций.

Значения переменных можно изменить в любой момент, отредактировав соответствующую формулу. При этом обычно автоматически пересчитываются все формулы, которые прямо или опосредованно зависят от данной переменной. Если же этого не происходит, то необходимо включить автовычисление с помощью верхнего меню Mathcad (пункт **Math** → **Automatic calculation** или **Математика** → **Автовычисление**).

3. Размерности физических величин

Работа в среде Mathcad — эта третья ступень в использовании вычислительной техники при решении физико-математических, инженерно-технических, а также учебных (школьных и вузовских) задач. Две предыдущие ступени — это работа с машинными кодами (на ассемблере, например) и с языками программирования (BASIC, Pascal, C, fortran и т.д.).

Использование вычислительной техники в научно-технических расчетах сыграло с этими расчетами злую шутку: из расчетов были безжалостно выброшены размерности физических величин и единицы их измерения — метры, килограммы, секунды... Ручное решение физической задачи (школьной или вузовской задачи по физике, если говорить конкретнее), как правило, требовало и требует оперирования сугубо размерными величинами. Автоматизация таких расчетов — написание программ для компьютера исключает из задачи ее «физику»: переменные программы хранят только числовые величины, а соответствующие им единицы измерений программист должен «держать в уме». Из-за этого при переводе расчета на язык ЭВМ необходимо придерживаться строгого правила — все физические величины должны быть в одной системе единиц, кроме того, они должны быть без множителей мили, мега и т.д. Это жесткое правило вызывало и вызывает ряд неудобств, основные из которых следующие:

1. Международная система (СИ) хоть и широко распространена в мире, но не является и никогда не будет единственной. Базирование программы на какой-либо одной системе величин мешает естественному процессу глобального обмена идеями, получившему новое ускорение в эпоху Internet.

2. В процессе создания программы большую роль играет ее *отладка*, а основной инструмент отладки — это вывод на дисплей промежуточных результатов, анализ которых позволяет локализовать и устранить ошибку, если она была допущена при подборе формул и/или при написании самой программы. А здесь важно не только вывести значение какой-либо физической величины с правильным форматом, но и выразить ее в *нужных* единицах *нужной* системы измерения.

Система СИ при всех ее достоинствах внедрялась и у нас и во всем остальном мире в виде «подарочного набора». Часть единиц измерения (метры, секунды) используются без каких-либо проблем, другие же («нагрузка») так и не прижились в качестве доминирующих единиц. В теплоэнергетике, например, давление пара в котле чаще всего измеряют и выражают в атмосферах, а давление в конденсаторе — в миллиметрах ртутного столба. «Узаконенная» стандартная единица измерения давления — *паскаль* (ньютон на квадратный метр) оказалась крайне

Информационные технологии в менеджменте

неудобной. Трудно припомнить научно-техническую область, где паскали применялись бы в чистом виде — без масштабирующих множителей (бары, килопаскали, мегапаскали и др.). Дело в том, что «живая», вернее «прижившаяся» единица измерения, связана с «жизнью» — с физическим явлением или физическим опытом: атмосфера, как следует из самого названия, — это давление воздуха на уровне моря (*примерное давление*), а миллиметры ртутного столба напоминают нам об экспериментах Э.Торричели (в среде Mathcad torr — это и есть миллиметр ртутного столба).

3. В создаваемые программы приходится вставлять формулы, выведенные не только в результате теоретического анализа проблемы, но и полученные после статистической обработки экспериментальных данных. Коэффициенты таких формул, как правило, жестко связаны с той или иной системой единиц измерения и нередко требуют пересчета для использования в программе. Это может вносить дополнительные погрешности.

Один из путей решения проблемы единиц — это отказ от размерных физических величин и переход к использованию безразмерных чисел, хранящих количественную оценку тех или иных физических явлений, — число Рейнольдса, число Нуссельта, число Шмидта и т.д. Еще более радикальный путь — это переход от числовых характеристик к качественным оценкам физических явлений и технологических процессов. Можно сказать, что давление в котле равно стольким атмосферам, а можно сказать, что давление *нормальное (низкое, высокое и т.д.)* и построить систему управления энергоблоком, основанную на подобных *лингвистических (экспертных)* оценках и опирающуюся на *теорию нечетких множеств*.

Исключение единиц измерения физических величин из расчетов на компьютере сказалось и на работе в среде Mathcad. Многие неопытные пользователи примерно так начинают формировать Mathcad-документ

Давление P := 20 МПа

записывая размерность введенной величины в виде комментария, а не в виде *множителя у числовой константы* и отключая тем самым размерности физических величин из дальнейших расчетов.

Механизм работы с единицами измерений физических величин позволяет в среде Mathcad:

- вводить исходные данные в нужной системе измерений и в нужных единицах измерений;
- вести контроль размерностей в формулах, по которым проводятся расчеты (не складывать килограммы с метрами, грубо говоря);
- выводить рассчитанные данные в нужной системе измерений и в нужных единицах измерений;

Создание размерной переменной. Чтобы создать размерную переменную, определяющую, например, силу тока в **10 А**, введите выражение, присваивающее переменной *i* значение **10: i:=10**, и затем символ умножения **<*>**, а потом букву **"А"**. Поскольку все символы, обозначающие единицы измерения, зарезервированы и имеют предустановленные значения (связанные с размерностью), то литера **А** будет распознана Mathcad как Ампер:

$$I := 10 \cdot A$$

$$U := 110 \cdot V$$

$$R := \frac{U}{I} \quad R = 11 \Omega$$

Если ранее вы переопределили переменную **A**, присвоив ей какое-либо значение, то восприниматься как единица силы тока она уже не будет.

Вставить единицу измерения можно и по-другому, не вручную, а при помощи средств Mathcad. Для этого выберите команду **Insert** → **Unit** (**Вставка** → **Единица**), либо нажмите кнопку с изображением мерного стакана  на стандартной панели инструментов.

В диалоговом окне **Insert Unit** (**Вставить единицы**) три поля и три кнопки (рис. 1). В первом поле пользователь выбирает нужную физическую величину (**Dimension**: Force — сила, Frequency — частота, Inductance — индуктивность, Luminosity — освещенности и т.д.), а во втором — единицу измерения (**Unit**). В третьем поле показана система измерений, которую пользователь также вправе менять. Кнопка **OK** вставляет выбранную единицу и убирает с дисплея окно **Insert Unit**, кнопка **Insert** — вставляет единицу и оставляет окно на дисплее для дальнейшей работы с ним, а кнопка **Cancel** выполняет третью комбинацию действий – убирает окно и не вставляет единицу.

Выбрать используемую систему измерений можно в диалоговом окне **Math** → **Option** → **Unit System** (**Математика** → **Параметры** → **Система единиц**).

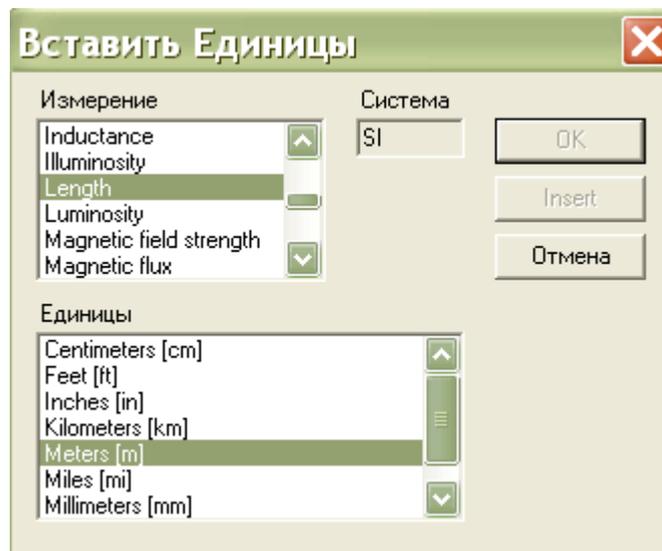


Рис. 1. Вставка единиц измерения размерной величины

Над размерными переменными можно производить любые разумные с физической точки зрения расчеты. Работая с размерными переменными Mathcad постоянно контролирует корректность расчетов. Например, нельзя складывать переменные разной размерности, в противном случае будет получено сообщение об ошибке «The units in this expression do not match» (Размерности в этом выражении не совпадают). Тем не менее, позволяется складывать, например, амперы с килоамперами и величины, размерность которых выражена в разных системах измерения (например, СИ и СГС).

Можно включить автоматический перевод единиц измерения в более простые единицы. Для этого необходимо открыть диалоговое окно **Result Format** (**Формат результата**) на вкладку, посвященную размерностям, с помощью

Информационные технологии в менеджменте

команды **Format** → **Result** → **Unit** **Display**
(Формат → **Результат** → **Отображение размерности**). Установите в ней флажок **Simplify units when possible** (**Упрощать единицы, когда возможно**).

Таблица 1

Встроенные переменные Mathcad, к которым «прикрепили» единицы измерения физических величин

<i>Физическая величина</i>	<i>Единицы измерения (предопределенная переменная Mathcad)</i>
активность (activity)	Bq (беккерель)
время (time)	day (день), hr (час), min, s, sec и yr (год)
вязкость динамическая (dynamic viscosity)	poise (пуаз)
вязкость кинематическая (cinematic viscosity)	stokes (стокс)
давление (pressure)	atm (атмосфера физическая), in_Hg (дюймы ртутного столба), Pa, psi (фунт силы на квадратный дюйм) и torr (мм ртутного столба)
длина (length)	cm, ft (фут), in (дюйм), km, m, mi (миля), mm и yd (ярд)
доза (dose)	Gy (грей) и Sv (зиберт)
емкость (capacitance)	F, farad, μF, nF, pF и statfarad
заряд (charge)	C, coul (кулон) и statcoul
индуктивность (inductance)	H, henry, μH, mH и stathenry
индукция магнитного поля (magnetic flux density)	gauss, stattersla, T и tesla
количество вещества (substance)	mole (моль)
магнитный поток (magnetic flux)	statweber, Wb и weber
масса (mass)	gm, kg, lb (фунт), mg, oz (унция), slug (пуд), ton (тонна британская) и tonne (тонна метрическая)
мощность (power)	hp (лошадиная сила), kW, W и watt
напряженность магнитного поля (magnetic field strange)	Oe и oersted (эрстед)
объем (volume)	fl_oz («жидкая» унция), gal (галлон), L, liter и mL
освещенность (illuminosity)	lx (люкс)
площадь (area)	acre (акр) и hectare
потенциал (potential)	kV, KV, mV, statvolt, V и volt
проводимость (conductance)	mho, S, siemens и statsiemens
сила (force)	dyne, kgf, lbf, N и newton
сила света (luminosity)	Cd (кандела) и Im (люмен)
скорость (velocity)	kph (км в час) и mph (мили в час)
сопротивление (resistance)	KΩ, MΩ, ohm, statohm и Ω
температура (temperature)	K и R
ток (current)	A, amp, KA, μA, mA и statamp
угол (angular)	deg, rad и str (стерадиан)
ускорение (acceleration)	g (ускорение свободного падения)
частота (frequency)	GHz, Hz, kHz, KHz и MHz
энергия (energy)	BTU (британская тепловая единица), cal, erg, J,

4. Функции, определяемые пользователем

Важнейшим объектом в математике является *функция*. Функция характеризуется своим именем и аргументами (список параметров) и однозначно ставит в соответствие значениям аргументов значение функции. Имя функции — это идентификатор, удовлетворяющий требованиям MathCAD (смотри правила создания имени в разделе «Переменные»). Список параметров представляет список переменных, разделенных запятыми, и заключенный в круглые скобки. Формат функции:

Имя_функции(переменная_1, переменная_2, ...)

Определение функции (локальное или глобальное) производится так же, как и определение переменной:

Имя_функции(переменная_1, переменная_2, ...):=определяющее_выражение

При глобальном определении функции следует использовать символ \equiv .

Поскольку аргументы функции представляют собой *формальные переменные*, значения которым присваиваются непосредственно при вычислении функции, то их не нужно определять заранее.

Примеры.

1. Функция расстояния между точками (x_1, y_1) и (x_2, y_2) :

$$r(x_1, y_1, x_2, y_2) := \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

2. Функция общего сопротивления участка электрической цепи с параллельно соединенными сопротивлениями:

$$R_P(R_1, R_2) := \frac{R_1 \cdot R_2}{R_1 + R_2}$$

3. Функция округления числа x до заданного количества N знаков после запятой:

$$\text{Round}(x, N) := \text{floor}(x \cdot 10^N + 0.5) \cdot 10^{-N}$$

4. Функция для вычисления тангенса (с именем, которое вам известно еще из школьного курса математики):

$$\text{tg}(x) := \tan(x)$$

Если выражение, определяющее функцию, содержит переменные, которые не входят в список ее аргументов, то они должны быть предварительно заданы. Например,

$$a := 4.5$$

$$f(x) := a \cdot x^2 + \cos(x)$$

К определенной пользователем функции можно выполнить обращение, указав ее *фактические аргументы*, которыми могут быть соответствующие им по типу константы, заданные до обращения к функции переменные, произвольные выражения. Например,

Информационные технологии в менеджменте

$$r(0, 2, 4, -5) = 8.062$$

$$RP(2, 5) = 1.429$$

$$\text{Round}(r(0, 0, -3, 7), 2) = 7.62$$

$$a := 2 \quad b := 4 \quad R := r(0, 0, a, b) \quad R = 4.472$$

Аргументом функции может быть и имя функции (встроенной или определенной пользователем), например:

$$g(x, f) := 2 \cdot x + f(x)$$

$$s := g(3, \sin) \quad s = 6.141$$

5. Запись логических выражений. Функция if

Часто при проведении вычислений требуется установить выполнение некоторого условия или условий для заданных величин. К условиям, например, относятся простые или двойные неравенства, их совокупности или системы.

В Mathcad выделяют **простые** и **составные условия** (логические выражения). Простое условие имеет следующую структуру:

$$\text{Выражение}_1 \text{ Оператор}_\text{сравнения} \text{ Выражение}_2$$

Например,

$$x \geq 5 \quad (x + y) > x^2 \quad \sin(x) \neq a \quad (a + b) = 12$$

Операторы сравнения и варианты их ввода приведены в [таблице 1.5](#).

Составные условия содержат простые условия, соединенные логическими операторами ([см. таблица 1.5](#)). Например, двойное неравенство $-4 < x \leq 56$ можно записать следующим образом:

$$(x > -4) \wedge (x \leq 56)$$

а совокупность неравенств $-4 < x \leq 7, 12 < x \leq 34$:

$$(x > -4) \wedge (x \leq 7) \vee (x \leq 34) \wedge (x > 12)$$

При записи условий следует помнить, что существует определенный порядок выполнения операций — слева направо в соответствии с приоритетом операций:

- Сначала вычисляются значения числовых выражений,
- Затем устанавливаются результаты простых условий (выполняются операции сравнения),

— Далее в случае составного условия выполняются логические операции в последовательности: отрицание, умножение, сложение. Изменить такой, принятый по умолчанию порядок выполнения логических операций, можно, используя в записи логического выражения скобки. При этом выражение, содержащееся в самых внутренних скобках, вычисляется первым.

Например,

Информационные технологии в менеджменте

$$x := 2 \quad y := 2$$

$$(x > 1) \vee (y < 1) \wedge (x^2 + y^2 \leq 4) = 1$$

$$[(x > 1) \vee (y < 1)] \wedge (x^2 + y^2 \leq 4) = 0$$

Здесь в первом выражении сначала выполняется операция умножения \wedge , а затем — сложения \vee , во втором — наоборот, сначала — сложения \vee , а затем — умножения \wedge .

Поскольку результатом оценивания условия является значение 1, соответствующее значению ИСТИНА, или 0, соответствующее значению ЛОЖЬ, и, более того, любое отличное от нуля числовое значение соответствует в Mathcad логическому значению ИСТИНА, то вместо логических операторов умножения \wedge и сложения \vee могут быть использованы арифметические операторы умножения и сложения соответственно. Например,

$$x := 4 \quad y := -1$$

$$x > 7 = 0$$

$$x < 10 = 1$$

$$(x > 7) \wedge (y < 0) = 0$$

$$(x > 7) \cdot (y < 0) = 0$$

$$(x > 7) \vee (y < 0) = 1$$

$$(x > 7) + (y < 0) = 1$$

А для ранее записанных с помощью логических операторов составных условий можно предложить и такой вариант записи:

Пример 2.1. Составьте логическое выражение, которое принимает значение ИСТИНА только в случае, когда точка с заданными координатами (x,y) принадлежит заштрихованной области, показанной на рис. 2.1.

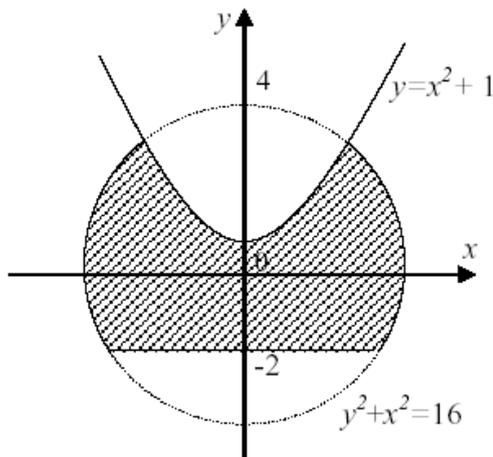


Рис. 2.1

Решение: Заштрихованная область – это решение системы неравенств:

$$\begin{cases} x^2 + y^2 \leq 16, \\ y \geq -2, \\ y \leq x^2 + 1, \end{cases}$$

которую, используя операции логического умножения, можно записать в виде следующего логического выражения:

$$(x^2 + y^2 \leq 16) \wedge (y \geq -2) \wedge (y \leq x^2 + 1)$$

Информационные технологии в менеджменте

Выясним, какое значение принимает логическое выражение для заданных x и y , подготовив MathCAD-документ следующего содержания:

$$\begin{aligned} x &:= 0 & y &:= 0 \\ (x^2 + y^2 \leq 16) \wedge (y \geq -2) \wedge (y \leq x^2 + 1) &= 1 \\ \\ x &:= -2 & y &:= -3.5 \\ (x^2 + y^2 \leq 16) \wedge (y \geq -2) \wedge (y \leq x^2 + 1) &= 0 \end{aligned}$$

Логические выражения используются для задания аргумента-условия встроенной функции **if**. Функция имеет следующий формат:

$$\mathbf{if}(\text{условие}, \text{выражение}_1, \text{выражение}_2)$$

Функция **if** принимает значение выражения₁, если условие является истинным, иначе значения выражения₂.

Пример 2.2. Требуется для заданного значения переменной x найти значение функции

$$f(x) = \begin{cases} \cos(x), & x < 0, \\ x+1, & 0 \leq x < 10, \\ \operatorname{tg}(x), & x \geq 10 \end{cases}$$

Решение: MathCAD-документ может быть составлен следующим образом:

$$\begin{aligned} a &:= 7 \\ f(x) &:= \mathbf{if}(x < 0, \cos(x), \mathbf{if}(x < 10, x + 1, \tan(x))) \\ f(a) &= 8 \end{aligned}$$

Пример 2.3. Задавая значения двух сопротивлений R_1 и R_2 и тип их соединения \emptyset – последовательное, 1 – параллельное), найдите общее сопротивление участка электрической цепи.

Решение: Возможный вариант оформления MathCAD-документа:

$$\begin{aligned} R_1 &:= 2 & R_2 &:= 4 & \text{type} &:= 1 \\ R &:= \mathbf{if}\left(\text{type} = 0, R_1 + R_2, \frac{R_1 \cdot R_2}{R_1 + R_2}\right) \\ R &= 1.333 \end{aligned}$$

6. Переменные диапазона (ранжированные переменные)

Переменная диапазона (или **ранжированная переменная**) — это переменная, которая принимает конечное множество значений из заданного диапазона при каждом обращении к ней.

С помощью переменной диапазона можно многократно вычислять одно и то же выражение для различных значений переменной. Результаты вычислений могут быть выведены в виде таблицы или графика.

Переменная диапазона задается с помощью оператора присваивания следующим образом:

$$\text{Имя_переменной} := \text{Выражение}_1 .. \text{Выражение}_K$$

или

Имя_переменной := Выражение_1, Выражение_2 .. Выражение_K

Здесь символ .. (две точки) указывает на задание диапазона и вводится нажатием клавиши [;] (точка с запятой) или нажатием кнопки **m..n** панели инструментов **Math, Calculator** (см. таблица 1.4), а остальные элементы имеют следующий смысл:

- Выражение_1 – первое значение переменной,
- Выражение_2 – второе значение переменной,
- Выражение_K – последнее значение переменной.

Разность $\text{Выражение}_2 - \text{Выражение}_1$ задает шаг изменения переменной. Если Выражение_2 не указано (первая форма задания переменной), то шаг по умолчанию считается равным 1. Таким образом, множество значений переменной диапазона образуют арифметическую прогрессию, все члены которой принадлежат промежутку:

$$[\text{Значение Выражения}_1; \text{Значение Выражения}_K]$$

Все параметры в выражениях должны быть определены до задания переменной диапазона. Например,

$$x0 := -12.4 \quad hx := 0.1 \quad xk := 14.3$$

$$x := x0, x0 + hx .. xk$$

Если значение Выражения_K окажется меньше значения Выражения_1 , а шаг изменения значения переменной диапазона отрицательным, например, $x := 6, 4 .. 18$, то будет выдано следующее сообщение об ошибке:

Can't understand the way this range variable is defined

Пример 2.4. Вычислите значения функции $y = 4x^3 - 2x + 5$ для значений x , изменяющихся от -1.15 до 1 с шагом $0,3$.

Решение: MathCAD-документ может быть оформлен следующим образом:

$$f(x) := 4x^3 - 2x + 5$$

$$x0 := -1.15 \quad xk := 1 \quad h := 0.3$$

$$x := x0, x0 + h .. xk \quad x = \quad f(x) =$$

-1.15	1.217
-0.85	4.244
-0.55	5.434
-0.25	5.438
0.05	4.9
0.35	4.471
0.65	4.799
0.95	6.53

Обратите внимание на то, что последнее значение, которое принимает переменная, может не совпадать с последним, указанным при задании диапазона. Это имеет место тогда, когда последний член арифметической прогрессии не совпадает с границей диапазона.

Если выражение содержит более двух переменных диапазона, то при выводе его значений в виде таблицы сначала все свои значения примет первая в записи выражения. Проиллюстрируем это следующим примером.

Информационные технологии в менеджменте

Пример 2.5. Вычислите значения функции $f(x,y)=x^2 + y$ для значений x , изменяющихся от 1 до 1,5 с шагом 0,2, и y , изменяющихся на промежутке от -1 до 0 с шагом 0,3.

Решение:

$x := 1, 1.2, 1.5$		$y := -1, -0.7, 0$		$f(x,y) := x^2 + y$																										
$x =$	$y =$	$f(x,y) =$	$f(x,-1) =$	$f(x,-0.7) :$																										
<table border="1" style="border-collapse: collapse; width: 30px; height: 30px; text-align: center;"><tr><td>1</td></tr><tr><td>1.2</td></tr><tr><td>1.4</td></tr></table>	1	1.2	1.4	<table border="1" style="border-collapse: collapse; width: 30px; height: 30px; text-align: center;"><tr><td>-1</td></tr><tr><td>-0.7</td></tr><tr><td>-0.4</td></tr><tr><td>-0.1</td></tr></table>	-1	-0.7	-0.4	-0.1	<table border="1" style="border-collapse: collapse; width: 30px; height: 30px; text-align: center;"><tr><td>0</td></tr><tr><td>0.44</td></tr><tr><td>0.96</td></tr><tr><td>0.3</td></tr><tr><td>0.74</td></tr><tr><td>1.26</td></tr><tr><td>0.6</td></tr><tr><td>1.04</td></tr><tr><td>1.56</td></tr><tr><td>0.9</td></tr><tr><td>1.34</td></tr><tr><td>1.86</td></tr></table>	0	0.44	0.96	0.3	0.74	1.26	0.6	1.04	1.56	0.9	1.34	1.86	<table border="1" style="border-collapse: collapse; width: 30px; height: 30px; text-align: center;"><tr><td>0</td></tr><tr><td>0.44</td></tr><tr><td>0.96</td></tr></table>	0	0.44	0.96	<table border="1" style="border-collapse: collapse; width: 30px; height: 30px; text-align: center;"><tr><td>0.3</td></tr><tr><td>0.74</td></tr><tr><td>1.26</td></tr></table>	0.3	0.74	1.26	
1																														
1.2																														
1.4																														
-1																														
-0.7																														
-0.4																														
-0.1																														
0																														
0.44																														
0.96																														
0.3																														
0.74																														
1.26																														
0.6																														
1.04																														
1.56																														
0.9																														
1.34																														
1.86																														
0																														
0.44																														
0.96																														
0.3																														
0.74																														
1.26																														
			$f(x,-0.4) =$	$f(x,-0.1) :$																										
			<table border="1" style="border-collapse: collapse; width: 30px; height: 30px; text-align: center;"><tr><td>0.6</td></tr><tr><td>1.04</td></tr><tr><td>1.56</td></tr></table>	0.6	1.04	1.56	<table border="1" style="border-collapse: collapse; width: 30px; height: 30px; text-align: center;"><tr><td>0.9</td></tr><tr><td>1.34</td></tr><tr><td>1.86</td></tr></table>	0.9	1.34	1.86																				
0.6																														
1.04																														
1.56																														
0.9																														
1.34																														
1.86																														

Сопоставьте таблицы вывода значений функции $f(x,y)$ с фиксированными значениями второго аргумента (в последовательности -1, -0.7, -0.4, -0.1) с таблицей вывода значений функции $f(x,y)$.

Лекция № 4

Mathcad: Создание графиков

1. **Состав панели инструментов Math → Graph (Математика → График)**
2. **Построение X-Y-графиков**
3. **Построение графиков в полярной системе координат**
4. **Форматирование двумерных графиков**
5. **Построение трехмерных графиков**
6. **Форматирование трехмерных графиков**
7. **Создание анимации**

1. Состав панели инструментов Math → Graph (Математика → График)

В системе Mathcad предусмотрена возможность вставки в документ графиков различного типа. Полный их список можно узнать из подменю команды **Insert → Graph (Вставка → График)** (рис. 3.1).

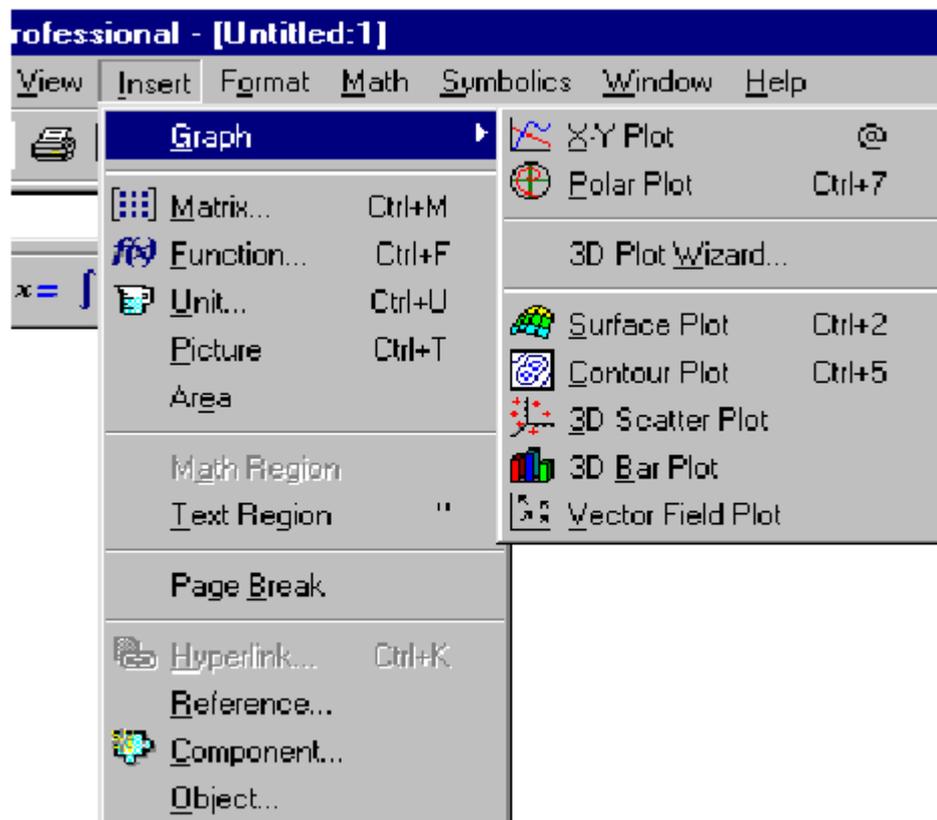


Рис. 3.1. Раскрытое подменю **Graph**

Все типы графиков можно разбить на две большие группы:

Двумерные графики	Трёхмерные графики
<ul style="list-style-type: none"> ▪ X–Y Plot (Декартов график) ▪ Polar Plot (Полярный график) 	<ul style="list-style-type: none"> ▪ Surface Plot (График поверхности) ▪ Contour Plot (Контурный график, линии уровня поверхности) ▪ 3D Scatter Plot (Точечный график) ▪ 3D Bar Plot (Трёхмерная гистограмма) ▪ Vector Field Plot (Векторное поле)

Деление графиков на типы несколько условно, так как, управляя установками многочисленных параметров, можно создавать комбинации различных типов графиков, а также новые типы. Например, двумерная гистограмма распределения (столбиковая диаграмма) или ступенчатый график являются разновидностями простого X–Y-графика (рис. 3.2, 3.3).

Большинство параметров графического процессора, необходимых для построения графиков, по умолчанию задаются автоматически. Поэтому для начального построения графика достаточно выбрать его тип. Графики могут иметь различные размеры и перемещаться в окне редактирования документа точно так же, как и любая другая область Mathcad-документа.

Информационные технологии в менеджменте

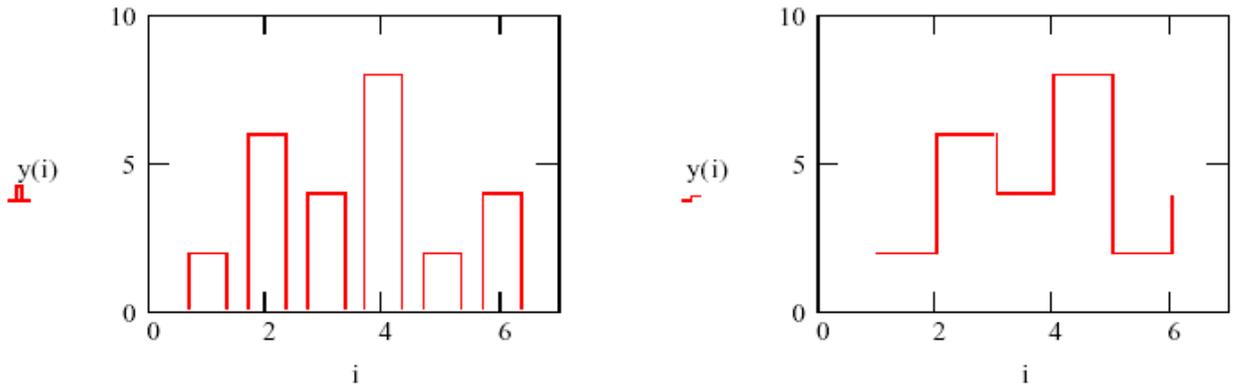


Рис. 3.2. Примеры столбиковой диаграммы и ступенчатого графика

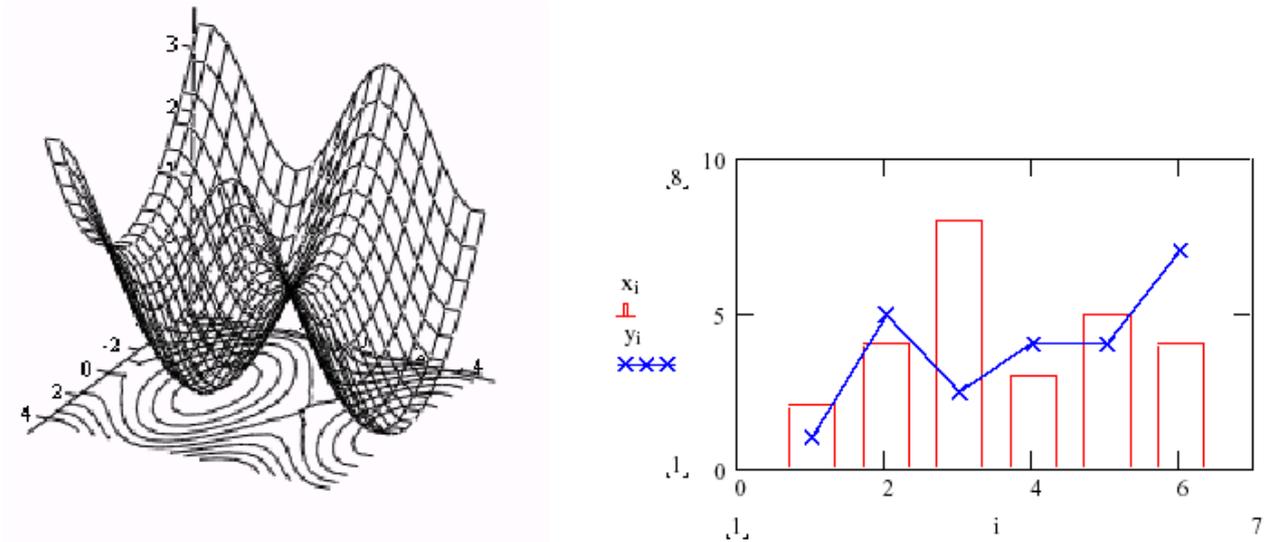


Рис. 3.3. Примеры комбинированных графиков

Вывести шаблон можно любым из следующих способов:

1. Выбором из меню команды **Insert** → **Graph** (**Вставка** → **График**) (рис. 3.1) соответствующего пункта.
2. Нажатием соответствующей кнопки на панели инструментов **Math** → **Graph** (**Математика** → **График**) (рис. 3.4).



Рис. 3.4. Панель инструментов **Graph**

3. Нажатием сочетания двух клавиш.

Рассмотрим более подробно команды меню **Math** → **Graph**:

 **X-Y Plot (Декартов график)** клавиша @. Служит для построения графика функции $y=f(x)$ в виде связанных друг с другом пар координат (x_i, y_i) при заданном промежутке изменения для i .

 **Polar Plot (Полярный график)** клавиши Ctrl+7. Служит для построения графика функции $r(\varphi)$, заданной в полярных координатах, где полярный радиус r зависит от полярного угла φ .

 **Surface Plot (График поверхности)** клавиши Ctrl+2. Служит для представления функции $z=f(x, y)$ в виде поверхности в трехмерном пространстве. При этом должны быть заданы векторы значений x_i и y_j , а также определена матрица вида $A_{ij} = f(x_i, y_j)$. Имя матрицы A указывается при заполнении рамки-шаблона.

 **Contour Plot (Карта линий уровня)**. Строит диаграмму линий уровня функции вида $z=f(x, y)$, т. е. отображает точки, в которых данная функция принимает фиксированное значение $z=const$.

 **3D Bar Plot (3D Столбковая гистограмма)**. Служит для представления матрицы значений A_{ij} или отображения значений функции $z=f(x, y)$ в виде трехмерной столбчатой диаграммы.

 **3D Scatter Plot (3D Точечный график)**. Служит для точечного представления матрицы значений A_{ij} или отображения значений функции $z=f(x, y)$ в заданных точках. Эта команда может также использоваться для построения пространственных кривых. В этом случае при заполнении рамки-шаблона можно задать три координаты отдельными векторами одинаковой размерности в виде (X, Y, Z) .

 **Vector Field Plot (Векторное поле)**. Служит для представления двумерных векторных полей $V=(V_x, V_y)$. При этом компоненты векторного поля V_x и V_y должны быть представлены в виде матриц. При помощи этой команды можно построить поле градиента функции $f(x, y)$.

2. Построение X-Y-графиков

Незаполненный шаблон графика (рис. 3.5) представляет собой пустой прямоугольник с двумя знакозаполнителями (знакоместами) в виде темных маленьких квадратов, расположенных на осях X и Y. Ось X (горизонтальная) — ось абсцисс точек графика $y=f(x)$, а ось Y (вертикальная) — ось ординат точек графика. На рис. 3.5 курсор ввода размещен на знакозаполнителе оси X. Переход с одного знакоместа на другое выполняется перемещением указателя мыши с последующим щелчком левой кнопкой или нажатием клавиши **TAB**.

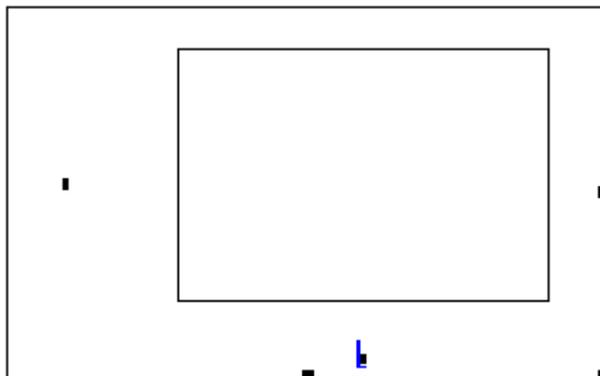


Рис. 3.5. Шаблон двумерного графика

Mathcad строит график линии по точкам, последовательно соединяя соседние точки отрезком. Их координаты определяются выражениями, указанными в соответствующих заполнителях осей. Таким образом, график линии — это ломаная. Mathcad предусматривает два способа построения графика функции одной переменной $y = f(x)$:

1. *Без задания переменной диапазона (быстрое построение графика).* При этом пределы изменения независимой переменной x автоматически задаются от -10 до 10 , но с учетом ее допустимых значений (так, например, для функции $\ln(x-3)$ будут установлены пределы от 3 до 10).

2. *Заданием переменной диапазона (ранжированной переменной).* Пользователь сам указывает границы изменения переменной x , учитывая допустимое множество ее значений, причем область задания переменной в документе должна располагаться строго левее или выше графической области.

Если график построен первым из перечисленных способов (*без задания переменной диапазона*), то автоматически установленное масштабирование может оказаться не совсем удачным, например, числовые значения меток осей не будут представлены неокругленными десятичными числами (как это хотелось бы), или не совсем ясно, каково поведение функции на отдельных промежутках изменения переменной x . Однако для первичного графического представления функции предпочтительнее автоматическое масштабирование. А затем, анализируя график, выбирается более подходящий масштаб.

Пример. Построить график функции $\cos(x)$:

$$x := -2\pi, -2\pi + 0.001.. 2\pi$$

$$2\pi = 6.283$$

$$-2\pi = -6.283$$

$x =$

-6.283
-5.283
-4.283
-3.283
-2.283
-1.283
-0.283
0.717
1.717
2.717
3.717
4.717
5.717

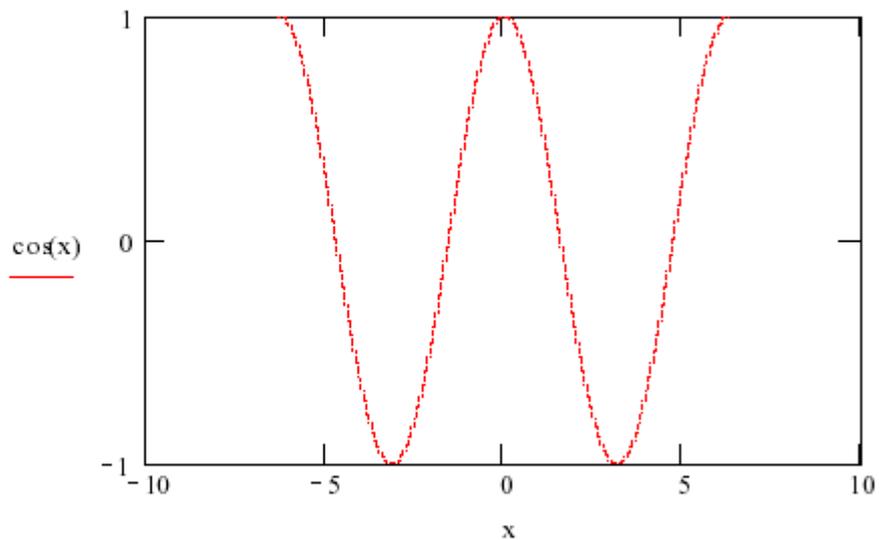


Рис. 3.8.

Как видно из [рис. 3.8](#), график был построен только в пределах заданного диапазона изменения переменной x (от -6.283 до 6.283), а автоматически установленные границы на оси X им не соответствуют. Если графическую область сделать активной (переместите курсор на область графика), то вы увидите ([рис. 3.9](#)) числовые значения концов диапазона задания переменной x и границы диапазона изменения выражения на оси Y .

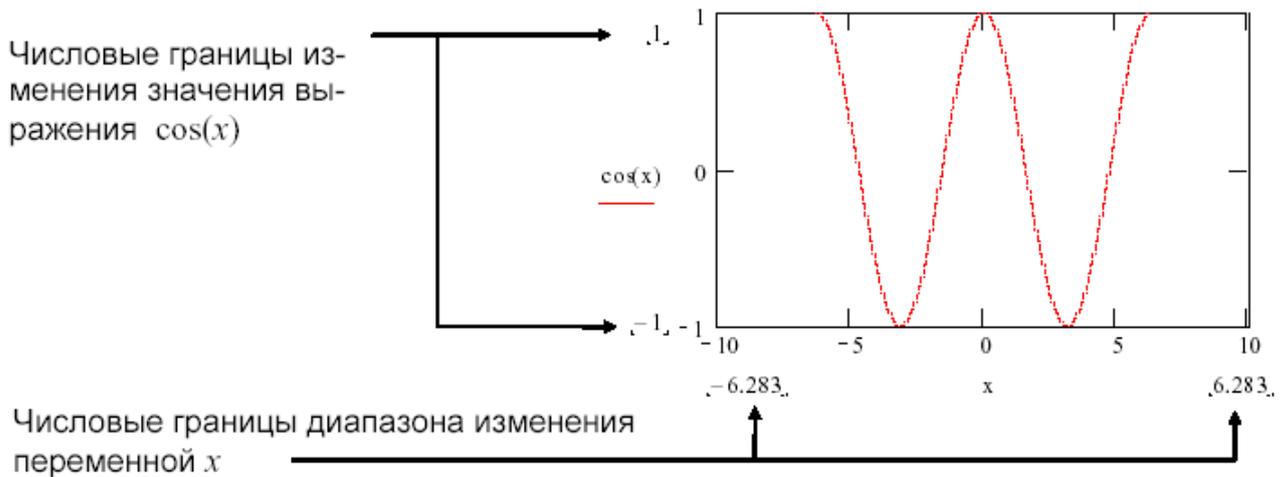


Рис. 3.9

Вы можете построить график на другом промежутке задания переменной x . Для этого достаточно отредактировать область задания переменной диапазона x .

Сделав активной область задания переменной диапазона x , отредактируйте ее следующим образом:

$$x := -7.5, -7.49, 12.1$$

После такого изменения график автоматически обновляется (рис. 3.10).

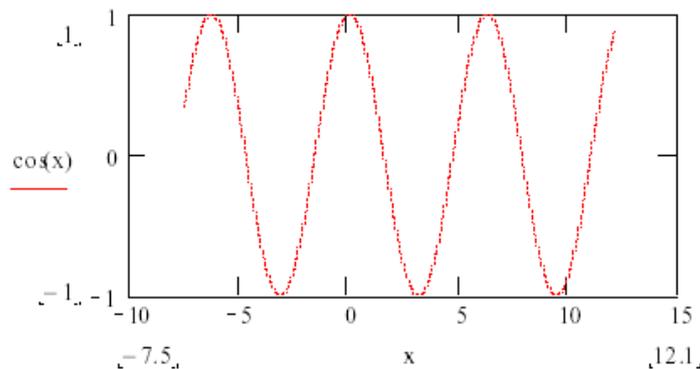


Рис. 3.10

Построение нескольких графиков в одной системе координат. В одной системе координат можно строить графики нескольких функций. Для этого необходимо ввести определяющие их выражения, заполняя соответствующее знакоместо на оси Y . Элементы списка выражений разделяются запятыми. Ввод разделителя-запятой приводит к появлению очередного знакоместа для его заполнения (рис. 3.15). После заполнения необходимых знакомест и выхода из области можно увидеть результат построения. Графики различных функций будут выведены линиями, отличающимися цветом и/или стилем, что зависит от настройки графических параметров. Если какое-либо знакоместо в списке выражений оказалось незаполненным, то будет выведено сообщение об ошибке.

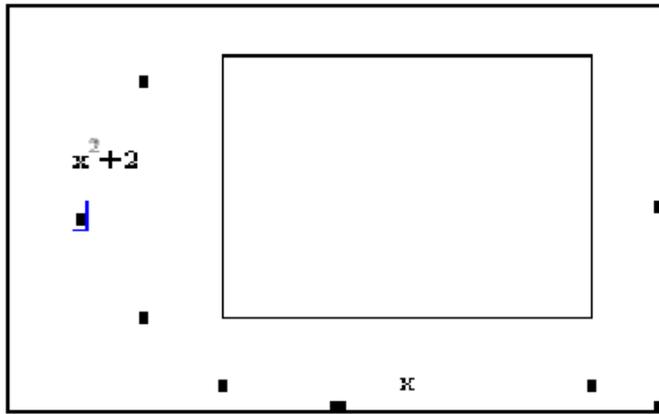


Рис. 3.15. Ввод выражений на оси Y

Пример.

В одной графической области построим графики следующих функций:

$$y = \frac{x \cdot \sin(x)}{5}, \quad y = \frac{5 \cdot x^2 - 15}{x^2 + 5}, \quad y = 0.$$

Если все функции зависят от одной переменной x и рассматриваются на одном и том же промежутке ее задания, то при заполнении знакоместа на оси X достаточно ввести имя переменной один раз.

Возможный вариант оформления MathCAD-документа показан на рис. 3.16.

$x := -20, -19.99, 20$

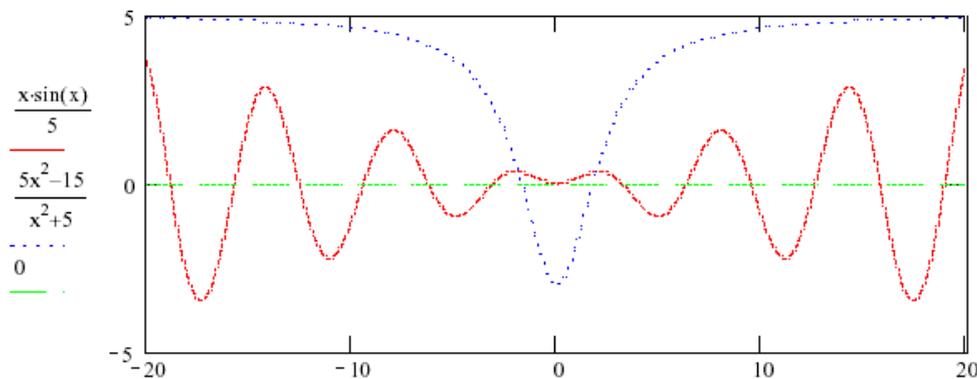


Рис. 3.16

Если выражение, определяющее функцию, является достаточно громоздким для ввода его в области построения графика или его необходимо неоднократно использовать в процессе решения задачи, то целесообразнее задать его с помощью функции пользователя. Если выражений несколько, то им должны соответствовать функции с разными именами.

Например, предыдущее задание можно выполнить так, как показано на рис. 3.17.

$$x := -20, -19.99..20$$

$$f(x) := \frac{x \cdot \sin(x)}{5} \quad g(x) := \frac{5x^2 - 15}{x^2 + 5}$$

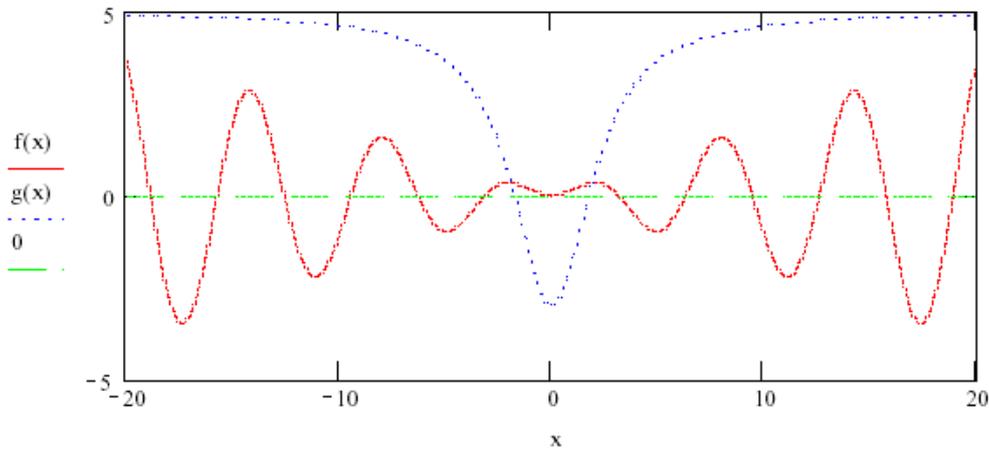


Рис. 3.17

Пример.

Исследуем изменение периода функции $y = \sin(\omega \cdot x)$ в зависимости от частоты ω . Зададим значения ω равными 1, 2 и 3 (рис. 3.18).

$$x := 0, 0.01..2\pi$$

$$f(x) := \sin(x)$$

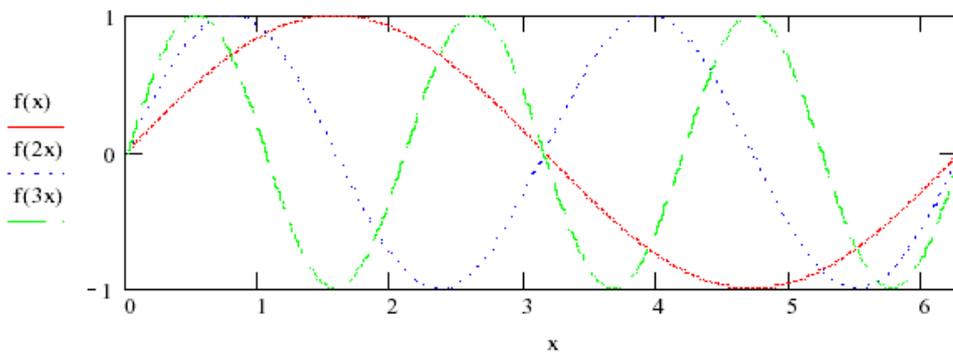


Рис. 3.18

Пример. Исследуем функцию на непрерывность, построив ее график:

$$y = \begin{cases} x^2, & 0 \leq x < 1, \\ 2, & 1 \leq x < 4, \\ 6\cos(7x), & 4 \leq x \leq 6. \end{cases}$$

Так как данная функция на разных промежутках задается различными выражениями (является кусочной), то ее график можно получить, построив отдельно графики трех выражений, рассматриваемых на соответствующих интервалах. Для чего удобно задать три различные переменные диапазона, например x , t , v (рис. 3.20), и ввести их списком при заполнении знакоместа на оси X. Элементы списка разделяются запятыми. Ввод запятой приводит к появлению нового знакоместа для его заполнения очередным элементом списка.

Информационные технологии в менеджменте

На рис. 3.19 показан фрагмент графической области при вводе переменных на оси X.

Каждому выражению из списка на оси Y соответствует элемент списка на оси X. Если какому-либо выражению на оси Y нет соответствующего выражения на оси X, то по умолчанию ему будет поставлен в соответствие последний элемент из списка на оси X.

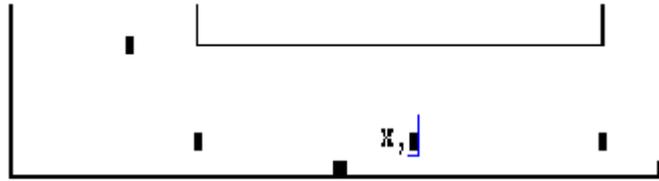


Рис. 3.19

$x := 0, 0.1..1$

$t := 1, 1.1..4$

$v := 4, 4.1..6$

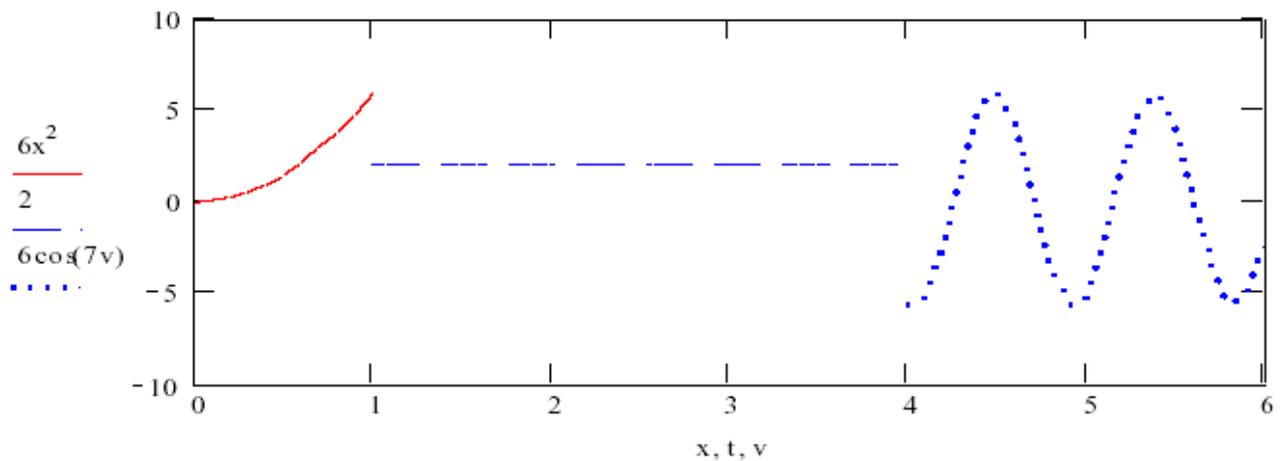


Рис. 3.20

Анализируя построенный график (рис. 3.20), можно сделать вывод о наличии двух точек разрыва функции $x_1=1$ и $x_2=4$ на заданном интервале $[0;6]$.

Рассматриваемую функцию на интервале $[0; 6]$ можно задать с помощью одного выражения таким образом:

$$y(x) := \text{if}(x < 1, 6x^2, \text{if}(x \geq 4, 6 \cos(7x), 2))$$

или

$$y(x) := 6x^2 \cdot (x \geq 0) \cdot (x < 1) + 2 \cdot (x \geq 1) \cdot (x < 4) + 6 \cos(7x) \cdot (x \geq 4) \cdot (x < 6)$$

Поскольку Mathcad строит график, последовательно соединяя соседние точки, то, выполнив построения так, как показано на рис. 3.21, увидим график в виде непрерывной (без разрывов) линии.

Информационные технологии в менеджменте

$$y(x) := 6x^2 \cdot (x \geq 0) \cdot (x < 1) + 2 \cdot (x \geq 1) \cdot (x < 4) + 6 \cos(7x) \cdot (x \geq 4) \cdot (x < 6)$$

$$x := 0, 0.001..6$$

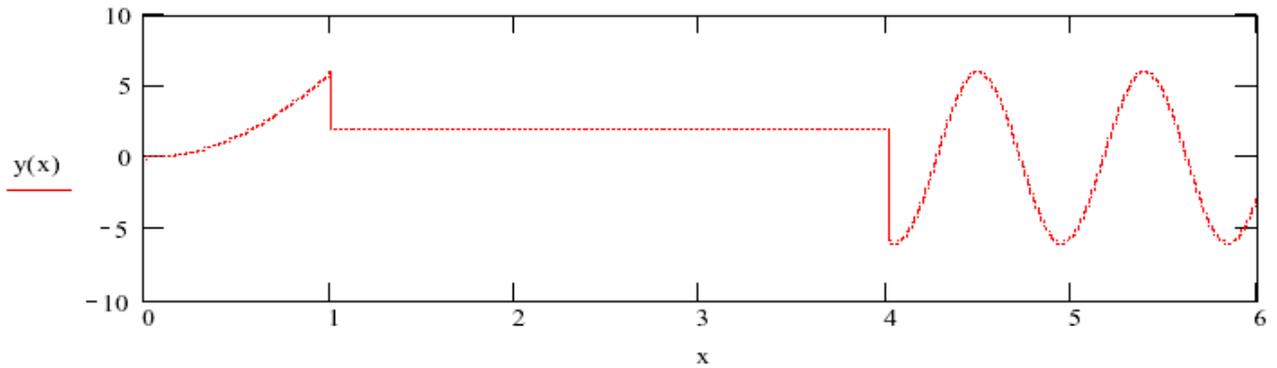


Рис. 3.21

Устранить недостаток такого построения можно, задав три переменные диапазона и строя график одной и той же функции $y(x)$, но на разных промежутках задания ее аргумента (рис. 3.22).

$$y(x) := 6x^2 \cdot (x < 1) + 2 \cdot (x \geq 1) \cdot (x < 4) + 6 \cos(7x) \cdot (x \geq 4)$$

$$x := 0, 0.001..0.99 \quad t := 1, 1.01..3.99 \quad v := 4, 4.01..6$$

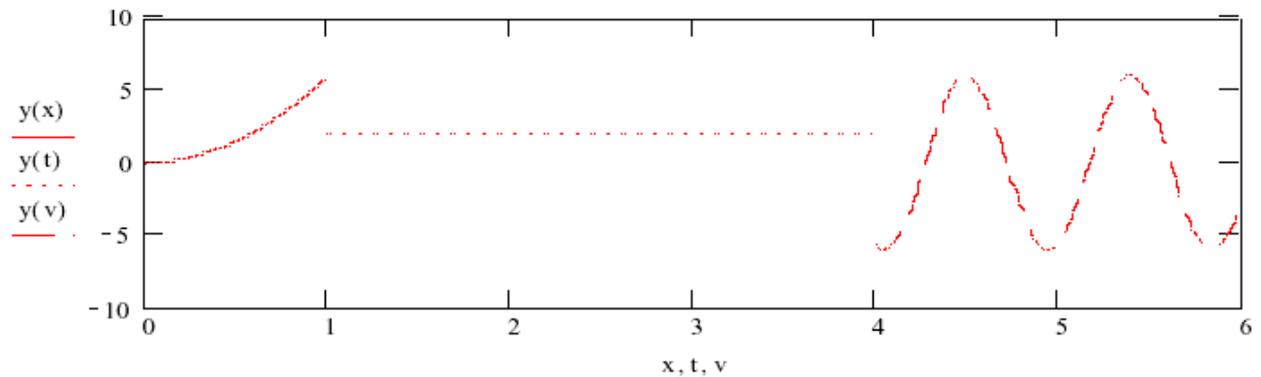


Рис. 3.22

Построение графиков параметрически заданных кривых. Графики параметрически заданных кривых: $x = x(t)$, $y = y(t)$ строятся на шаблоне двумерного графика. В этом случае при заполнении знакомест на осях X и Y вводятся выражения, задающие соответствующие координаты x и y точек кривой. И так же, как и при построении графиков функций, аргумент (параметр) t может быть заранее не задан (быстрое построение графика) или задан как переменная диапазона.

В одной системе координат могут быть построены графики различных кривых. Примеры построения кривых показаны на рис. 3.23—3.26.

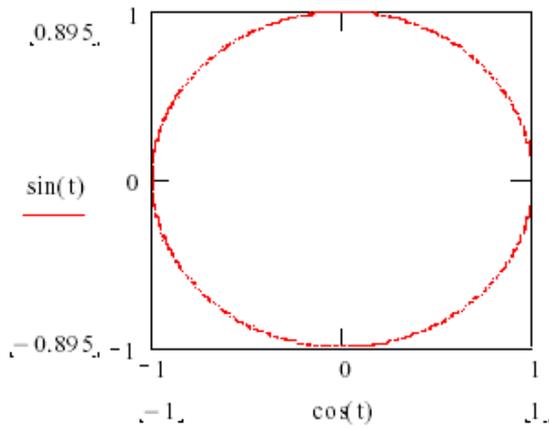
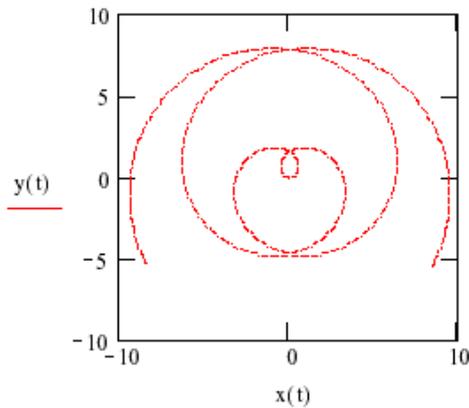


Рис. 3.23. График параметрически заданной окружности радиуса $R=1$: $x=\cos(t)$, $y=\sin(t)$

$$x(t) := t \cdot \cos(t) \quad y(t) := t \cdot \sin(t)$$



$$t := 0, 0.01..5\pi$$

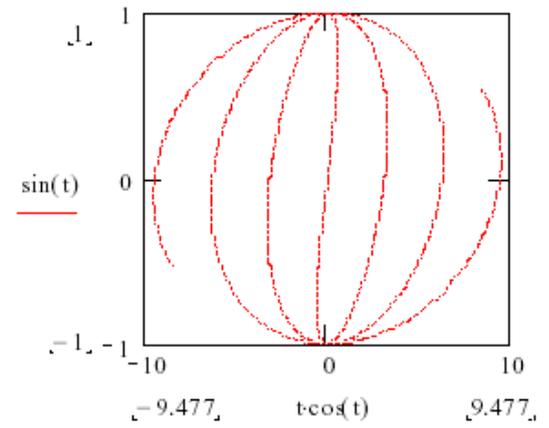
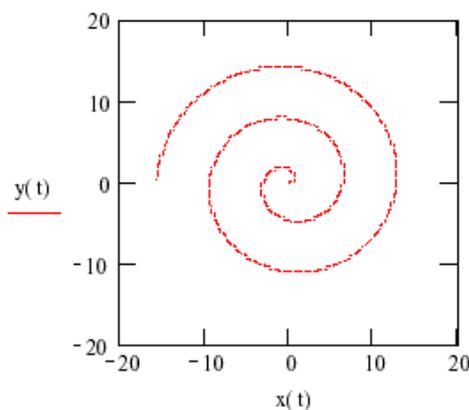


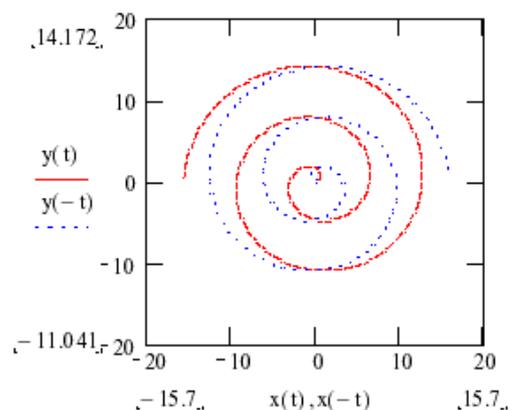
Рис. 3.24. График кривой $x=t\cos(t)$, $y=\sin(t)$

Рис. 3.25. Графики параметрически заданной кривой.

Первый построен без задания переменной t ; второй и третий – с указанием границ ее изменения.

На втором графике построена одна спираль кривой (при положительных значениях параметра t).

На третьем – две спирали, построенные отдельно при положительных и отрицательных значениях параметра



Заметим, что графики любых функций можно рассматривать как графики параметрически заданных кривых, для которых роль параметра t играет независимая переменная x .

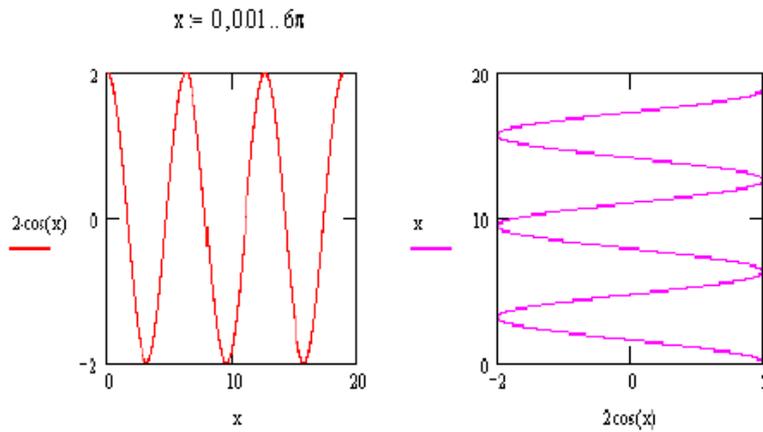


Рис. 3.26. Изменение графика функции $y=2\cos x$ (первый график) в результате перестановки выражений, заданных на осях X и Y (поворот графика на угол 90^0)

3. Построение графиков в полярной системе координат

В полярной системе координат любая точка определяется двумя координатами R и φ , где R — полярный радиус; φ — полярный угол точки, $R=R(\varphi)$.

Вывести шаблон графика в полярной системе координат можно одним из трех способов:

1. Выберите из меню команды **Insert** → **Graph** (рис. 3.1) пункт **Polar Plot** (**Полярный график**).

2. Или нажмите комбинацию клавиш **Ctrl+7**.

3. Или нажмите кнопку  на панели **Graph** (рис. 3.4).

Незаполненный шаблон графика представляет собой окружность с двумя значащими для ввода данных в виде темных маленьких прямоугольников (рис. 3.38).

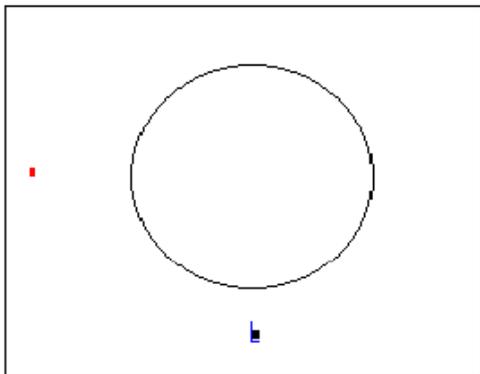


Рис. 3.38. Шаблон полярного графика с сообщением о необходимости заполнения знакомест ввода

Есть две возможности построения таких графиков:

— задавая пределы изменения переменной φ (причем переменная может иметь и другое имя), рис. 3.39;

— без определения диапазона изменения переменной (быстрое построение), рис. 3.40.

Информационные технологии в менеджменте

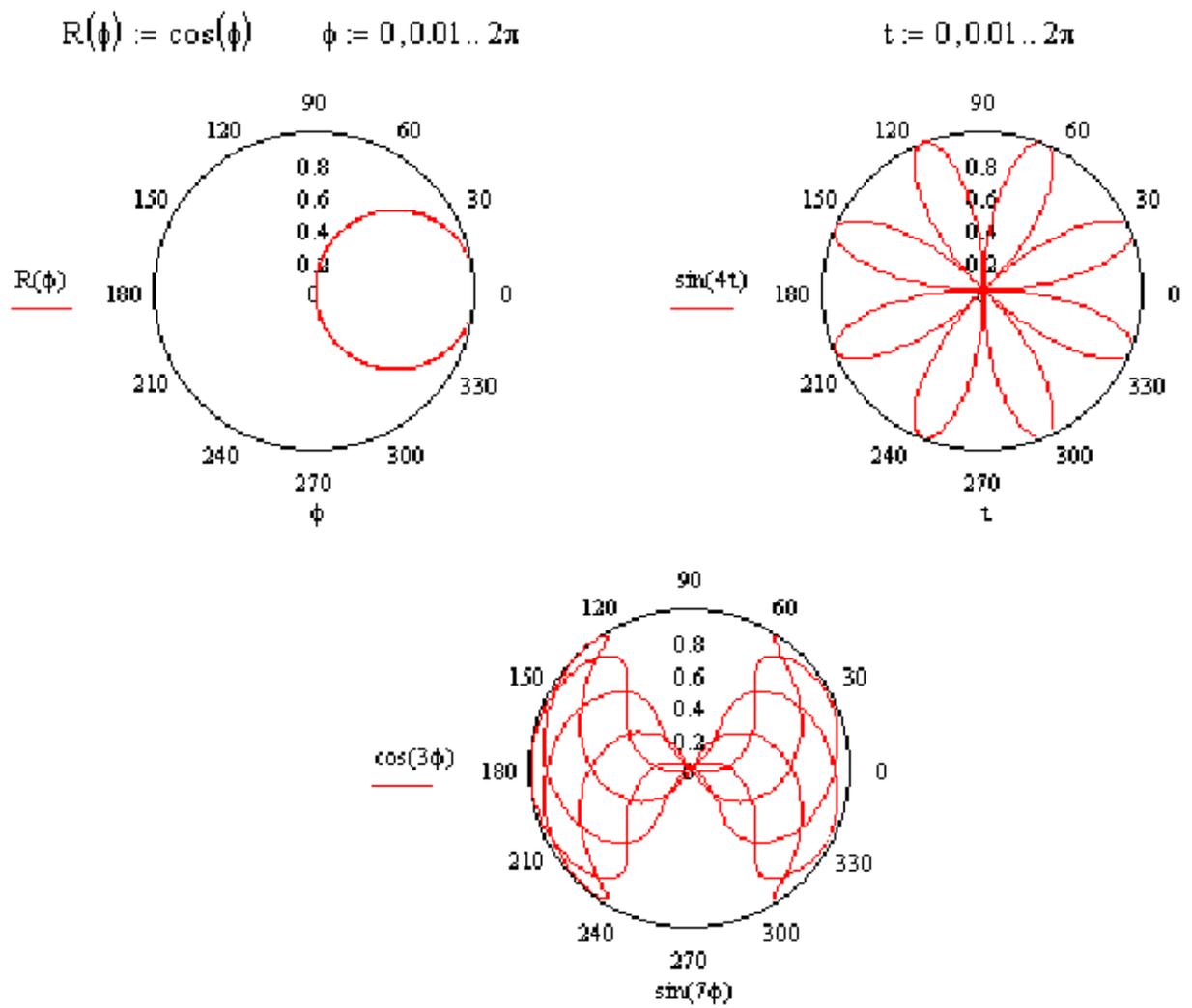


Рис. 3.39. Построение графиков с определением переменной

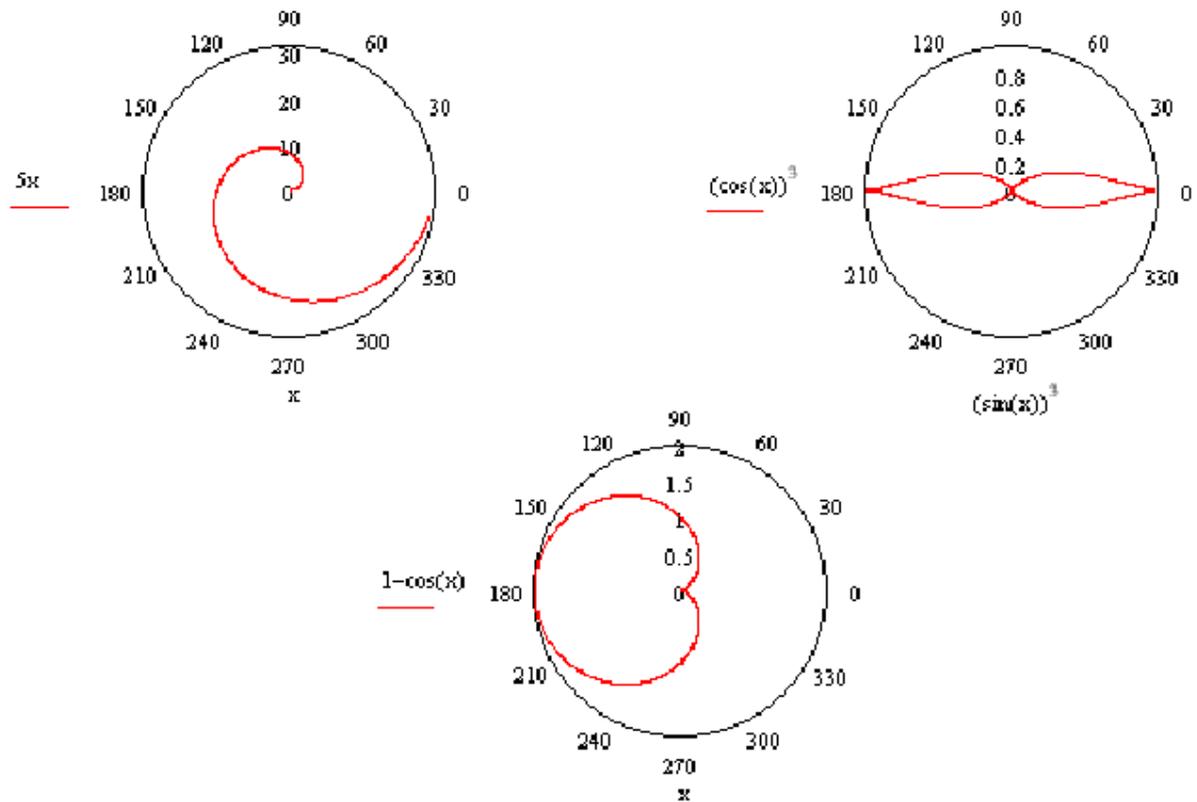


Рис. 3.40. Построение графиков без указания границ переменной (быстрое построение)

Аналогично двумерным графикам в полярной системе можно строить графики нескольких функций или кривых (рис. 3.41).

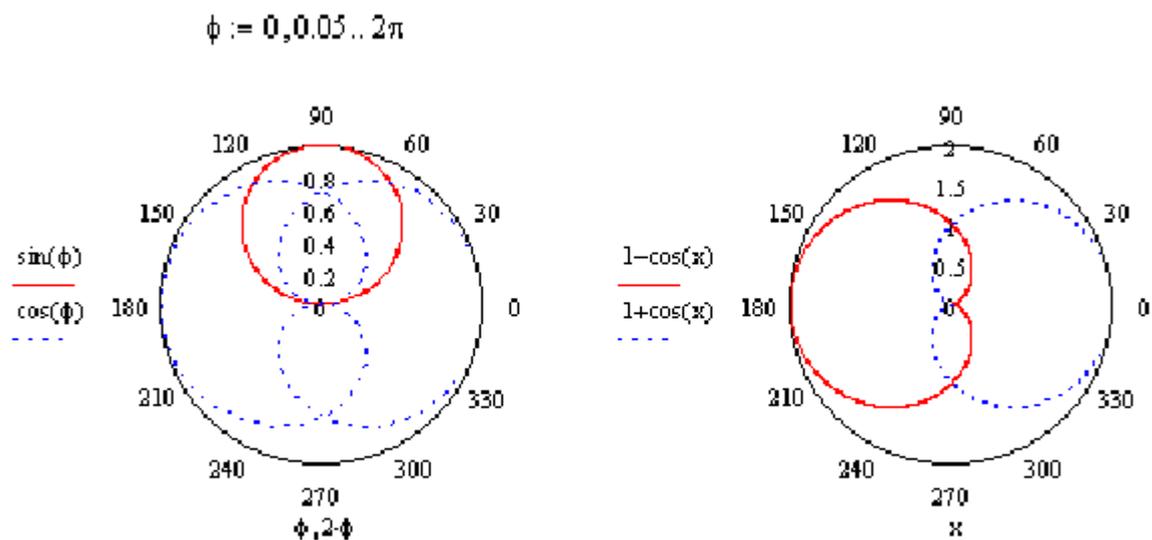


Рис. 3.41. Построение нескольких графиков на одном шаблоне

4. Форматирование двумерных графиков

Для вывода окна форматирования двумерного графика достаточно поместить указатель мыши в область графика и дважды щелкнуть левой кнопкой мыши. В окне документа появится окно форматирования (рис. 4).

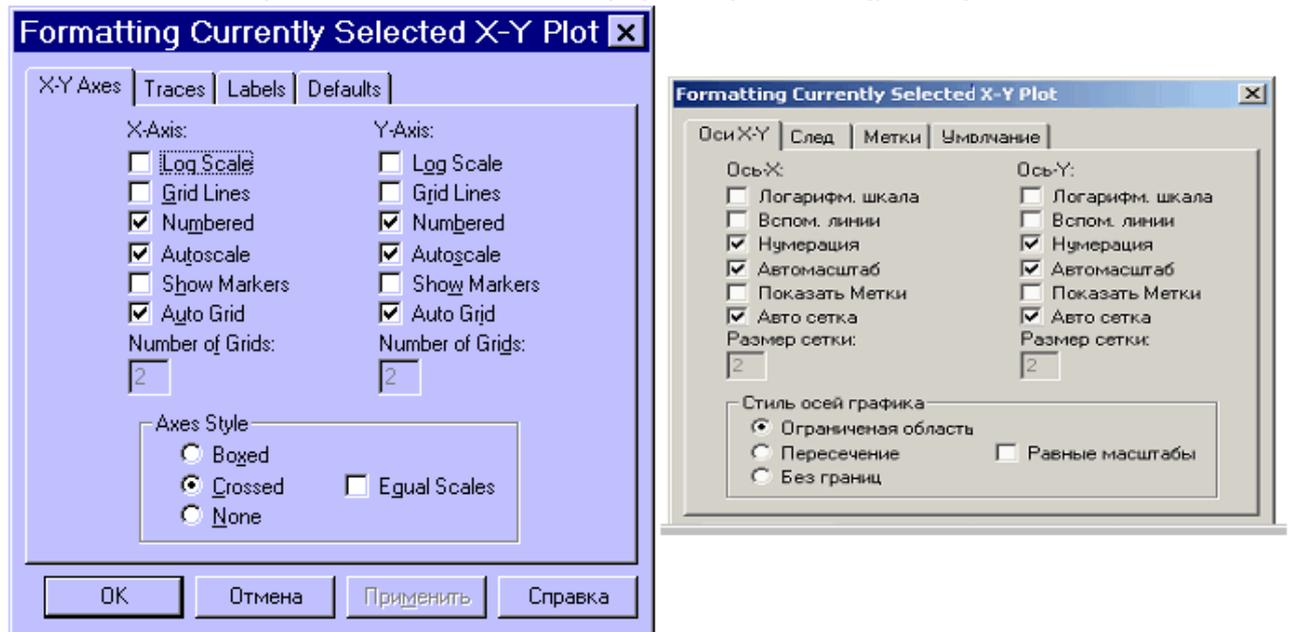


Рис. 4.

Оно имеет ряд вкладок:

- **X-Y Axes (Оси X-Y)**;
- **Traces (Трассировки)**;
- **Labels (Метки)**;
- **Defaults (Умолчание)**.

Вкладка становится активной, если установить на ее имя указатель мыши и щелкнуть левой кнопкой.

Первая из вкладок **X-Y Axes (Оси X-Y)** позволяет форматировать оси координат:

- **Log Scale (Логарифмическая шкала)** — задает логарифмические оси, в этом случае границы графика должны задаваться положительными числами;
- **Grid Lines (Вспомогательные линии)** — задает отображение сетки;
- **Numbered (Нумерация)** — задает отображение подписи к маркировкам на осях;

— **Autoscale (Автомасштаб)** — задает автоматическое нахождение подходящих границ для осей. Но если вы сами зададите в соответствующих ячейках минимальные и максимальные значения X_{min} , X_{max} , U_{min} , U_{max} , именно эти значения будут использоваться для определения границ графика;

— **Show Markers (Показать метки)** — если установить эту опцию, то в графической области появятся четыре дополнительные ячейки для создания красных линий маркировки, соответствующих двум специальным значениям x и двум специальным значениям y ;

— **Auto Grid (Авто сетка)** — при установке этой опции число линий сетки определяет Mathcad.

— **Axis Style (Стиль осей графика)** — группа кнопок этой области позволяет выбрать следующие варианты представления осей: **Boxed (Ограниченная область)**, **Crossed (Пересечение)** — оси пересекаются в точке

Информационные технологии в менеджменте

с координатами (0; 0), **None (Без границ)**. Флажок **Equal Scales (Равные масштабы)** позволяет задать одинаковый масштаб для обеих осей.

Форматирование оси графика можно произвести, выполнив на ней двойной щелчок.

Для изменения типа линий графиков необходимо активизировать вкладку **Traces (Трассировки)** (рис. 5):

— **Legend Lable (Легенда)** — каждой кривой можно поставить в соответствие некоторый текст, называемый легендой. Легенда отображается в нижней части графической области, а рядом с каждой легендой отображается тип линии соответствующей кривой;

— **Symbol (Символ)** — позволяет выбрать символ для каждой точки кривой (плюс, крестик, кружок и др.);

— **Line (Линия)** — можно выбрать один из следующих типов линий: solid (сплошная), dash (штриховая), dot (точечная) или dadot (штрихпунктирная). Это поле списка доступно в случае, если в поле **Type (Тип)** выбран элемент lines;

— **Color (Цвет)** — задается цвет представления кривой на экране;

— **Type (Тип)** — позволяет выбрать один из видов графика: в виде линий, в виде точек и т. п.;

— **Weight (Вес)** — позволяет задавать толщину линий графика.

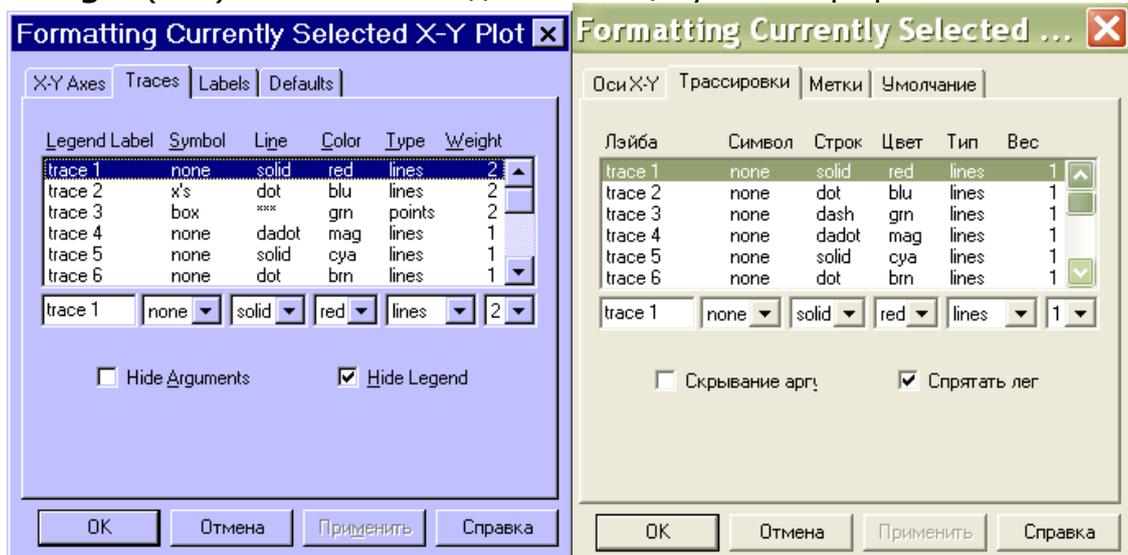


Рис. 5.

В нижней части вкладки **Traces** расположены опции:

— **Hide Arguments (Скрыть аргументы)** — эта опция по умолчанию отключена. В этом случае под именем функции рядом с осью ординат указывается текущий тип линий. Если установить данную опцию, указание типа линий исчезнет;

— **Hide Legend (Скрыть легенду)** — по умолчанию легенда не отображается. Если вы хотите отобразить под графиком текст легенды, его необходимо перед этим ввести в поле **Legend Lable (Легенда)** и подтвердить ввод, выполнив щелчок на кнопке **Применить**.

Вкладка **Labels (Метки)** позволяет ввести заголовок графика и подписи для осей (рис. 6).

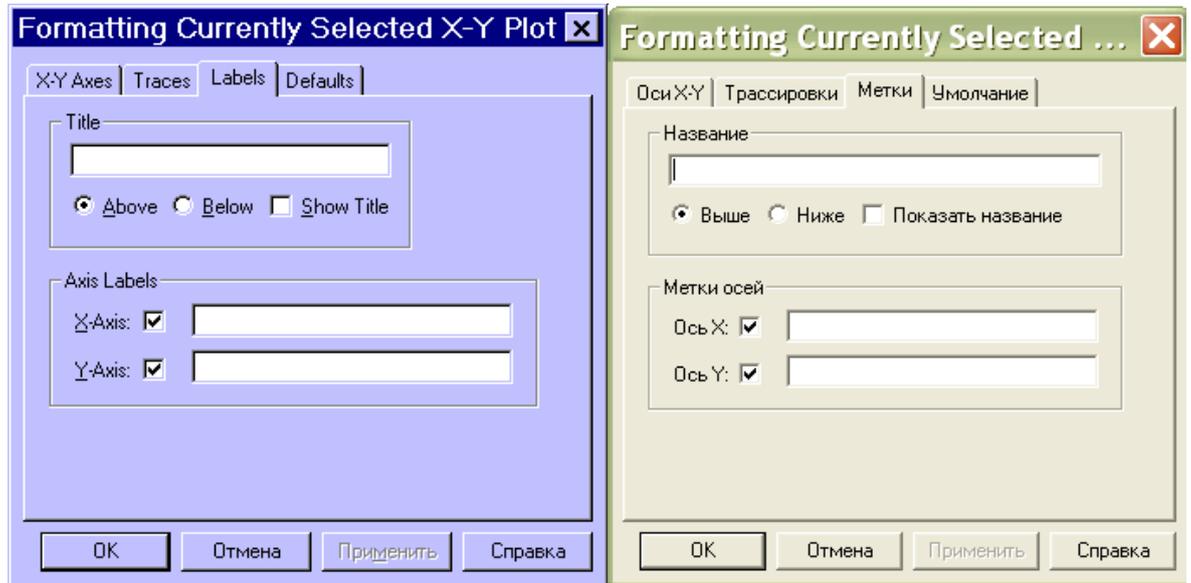


Рис. 6.

В меню **Format** → **Graph** (**Формат** → **График**) содержится команда **Zoom** (**Изменение масштаба**). При помощи этой команды можно увеличить фрагмент графика, предварительно выделив его протаскиванием мышки с нажатой левой клавишей. После отпускания клавиши координаты углов выделенной области будут отображены в полях окна **X-Y Zoom** (рис. 7). При помощи кнопки **Zoom** (**Масштаб +**) фрагмент можно увеличить, при помощи кнопки **Unzoom** (**Масштаб –**) отменить выделение фрагмента, а при помощи кнопки **Full View** (**Обзор**) — восстановить первоначальный вид графика. Если вы увеличили фрагмент графика, то при щелчке на кнопке **OK** в документе будет отображаться только этот фрагмент.

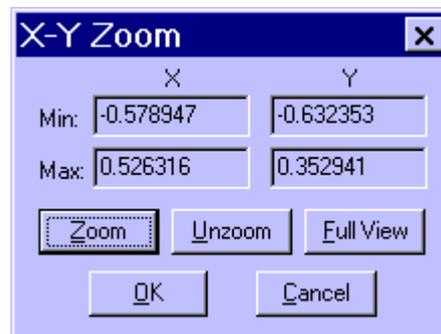


Рис. 7.

5. Построение трехмерных графиков

В Mathcad возможно построение следующих видов трехмерных графиков:

- График трехмерной поверхности (Surface Plot), [рис. 7.1a](#);
- График линий уровня (Contour Plot), [рис. 7.1б](#);
- Трехмерная столбчатая диаграмма/гистограмма (3D Bar Plot), [рис. 7.1в](#);
- Трехмерное множество точек (3D Scatter Plot), [рис. 7.1г](#);
- Векторное поле (Vector Field Plot).

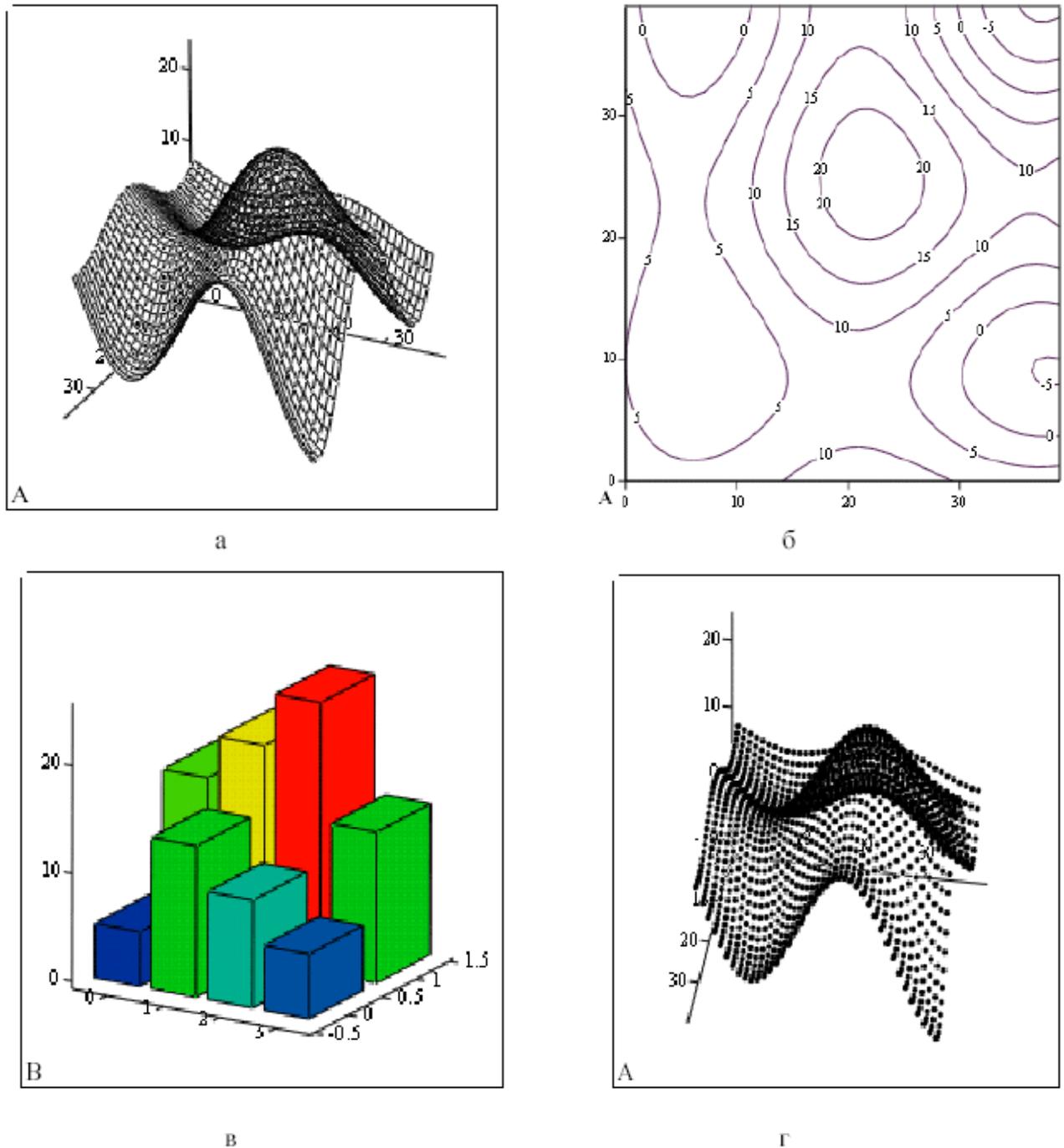


Рис. 7.1. Виды трехмерных графиков

Чтобы создать трехмерный график, требуется нажать кнопку с изображением любого из типов трехмерных графиков на панели инструментов **Graph** или выполнить команду вставки графика **Insert** → **Graph** (**Вставка** → **График**) или воспользоваться комбинацией клавиш **Ctrl+2** (для Surface Plot) или **Ctrl+5** (для Contour Plot).

В результате на рабочем листе будет выведена пустая область графика с тремя осями (рис.7.3) и единственным знакоместом в нижнем левом углу. В это знакоместо следует ввести или имя функции двух переменных (не указывая список аргументов) для быстрого построения трехмерного графика, либо имя матричной переменной, которая задаст распределение данных на плоскости. Понятно, что функция или матрица к этому моменту должна быть определена.

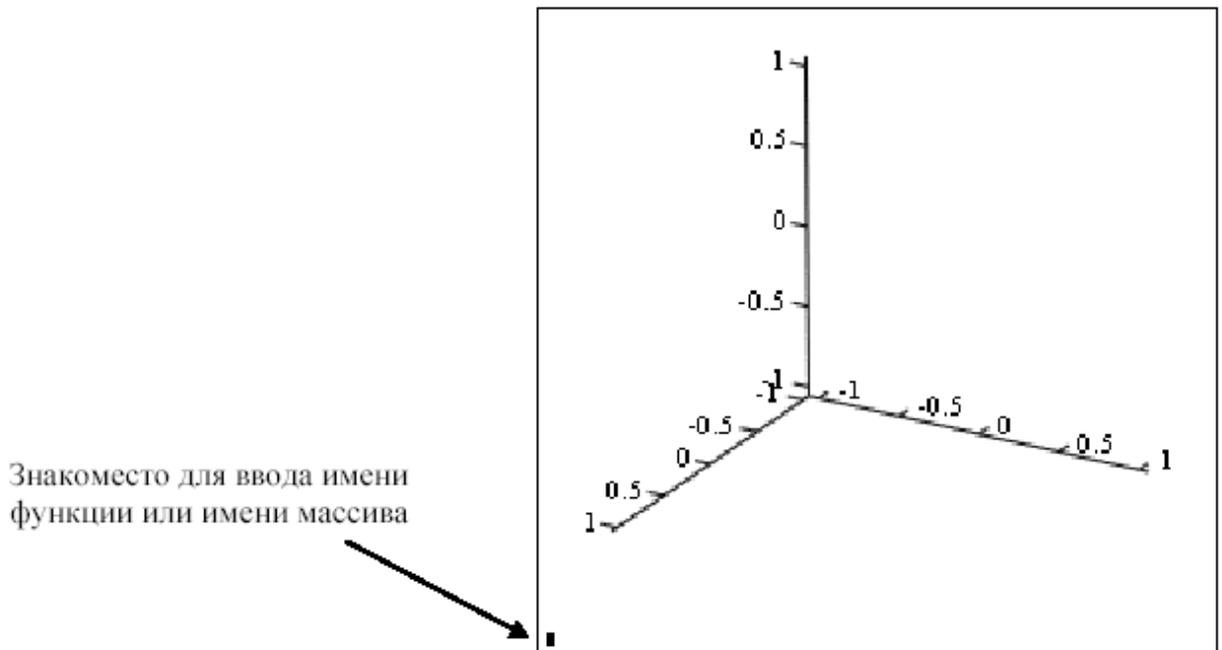


Рис. 7.3. Шаблон для построения трехмерного графика

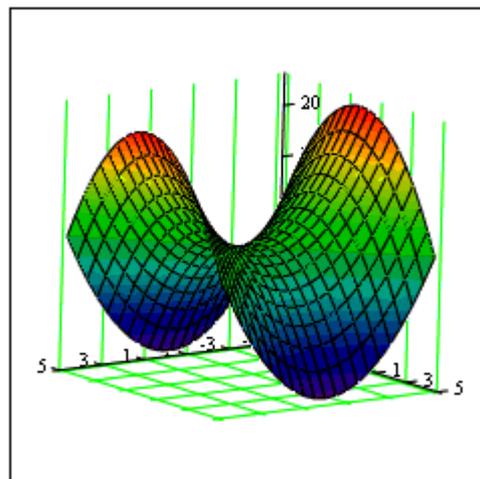
Замечание. При построении трехмерного графика можно воспользоваться услугами мастера трехмерной графики, сделав выбор пункта **3D Plot Wizard (Мастер 3D графиков)** из меню команды **Insert → Graph (Вставка → График)**.

Рассмотрим несколько примеров построения трехмерных графиков.

Для построения **графика функции $z=f(x, y)$ в виде поверхности в декартовой системе координат** можно воспользоваться двумя способами:

1. *Быстрое построение графика.* Необходимо определить функцию $f(x, y)$ и на панели **Graph** выбрать **Surface Plot (График поверхности)**. В появившейся графической области под осями на месте шаблона для ввода надо указать имя (без аргументов) функции. Независимые переменные x и y принимают значения из промежутка $[-5; 5]$ (рис. 8).

$$g(x, y) := x^2 - y^2$$



g

Рис. 8

Информационные технологии в менеджменте

При необходимости этот промежуток может быть уменьшен или увеличен. Для этого необходимо дважды щелкнуть левой кнопкой мыши по выделенному графику и в появившемся окне **3D Plot Format (Формат 3D графика)** на вкладке **QuickPlot Data** можно установить другие параметры изменения независимых переменных x и y (рис. 9).

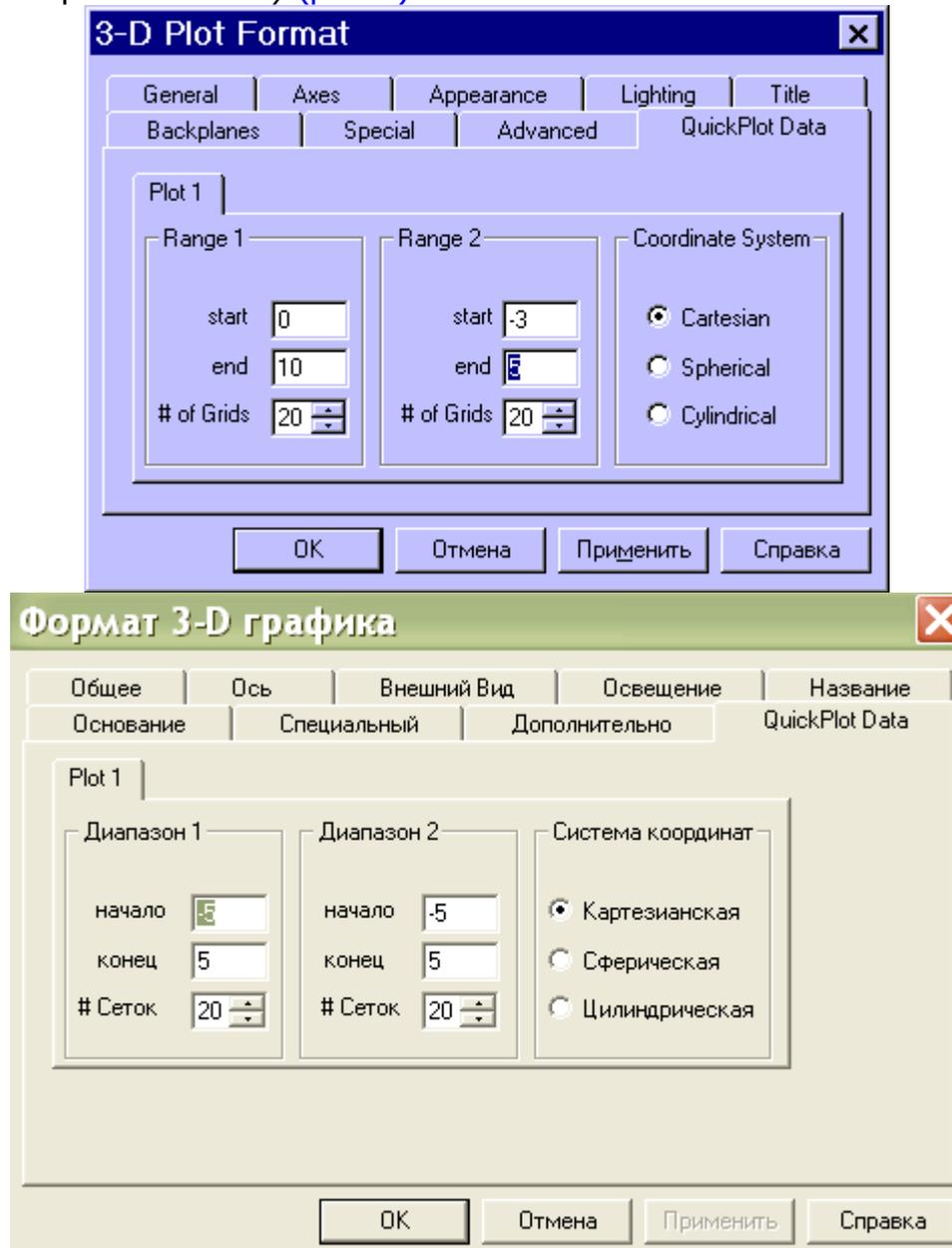


Рис. 9.

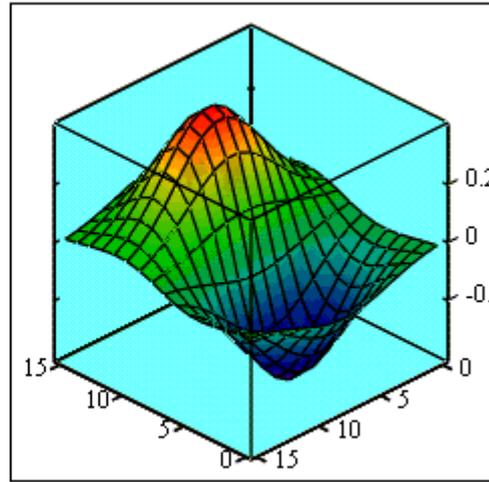
2. *С помощью заданной матрицы.* Для построения графика поверхности в определенной области изменения независимых переменных или с конкретным шагом их изменения необходимо сначала задать узловые точки x_i и y_j , в которых будут определяться значения функции (индексы i и j будут указывать координаты узла на сетке). После (а можно и до) этого надо определить функцию $f(x, y)$, график которой хотите построить. После этого необходимо сформировать матрицу значений функции в виде: $A_{i,j} = f(x_i, y_j)$ (рис. 10).

Теперь после выполнения команды **Graph** → **Surface Plot** в появившейся графической области достаточно ввести имя матрицы (без индексов).

```

N := 15
i := 0..N      j := 0..N
x_min := -1.5  x_max := 1.5
y_min := -1.5  y_max := 1.5

x_i := x_min + (i/N) * (x_max - x_min)
y_j := y_min + (j/N) * (y_max - y_min)
f(x,y) := sin(x) * exp(-x^2 - y^2)
A_{i,j} := f(x_i, y_j)
    
```



A

Рис. 10.

Также для построения графика поверхности в определенной области изменения независимых переменных или с конкретным шагом их изменения можно использовать функцию:

M := CreateMesh(f, xn, xk, yn, yk, s1, s2),

где f — функция, определяющая поверхность; x_n, x_k, y_n, y_k — начальные и конечные значения независимых переменных x и y ; s_1, s_2 — размерность сетки.

После выполнения команды **Graph** → **Surface Plot** в появившейся графической области вводится имя переменной (в данном случае M).

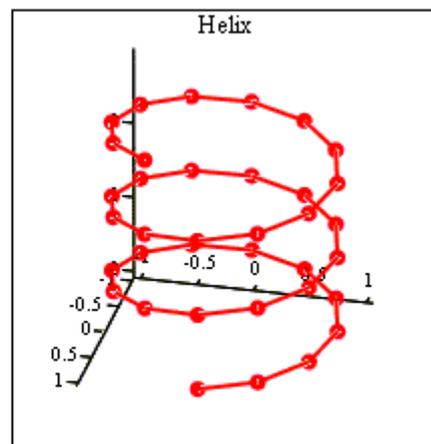
Построение графика кривой в пространстве. Трехмерные точечные графики можно использовать для построения изображения пространственных кривых. Пространственные кривые задаются, как правило, в виде $(x(t), y(t), z(t))$, где t представляет собой непрерывный действительный параметр (рис. 11).

Поскольку при построении трехмерной точечной диаграммы Mathcad позволяет отображать на графике только отдельные точки и соединяющие их линии, необходимо сначала определить три вектора координат — x_i, y_i, z_i . Пространственная кривая создается командой **Graph** → **3D Scatter Plot**.

```

N := 36      Число значений параметров
i := 0..N - 1  Индексная переменная i

x_i := cos(i/N * 6 * pi)  Задание векторов
                          координат x, y и z,
                          которые являются
                          функциями от
y_i := sin(i/N * 6 * pi)  индекса i. Данные
                          векторы координат
                          описывают винтовую
z_i := i/N * 3            линию.
    
```



(x, y, z)

Рис. 11.

6. Форматирование трехмерных графиков

Если вас не устраивает внешний вид созданного трехмерного графика, вы можете изменить его, выполнив команду **Format**→**Graph**→**3D Plot** или выполнив двойной щелчок мышкой на графической области. В результате на экране появится диалоговое окно **3D Plot Format**, позволяющее изменять параметры отображения графика.

Диалоговое окно **3D Plot Format** содержит несколько вкладок (рис. 12).

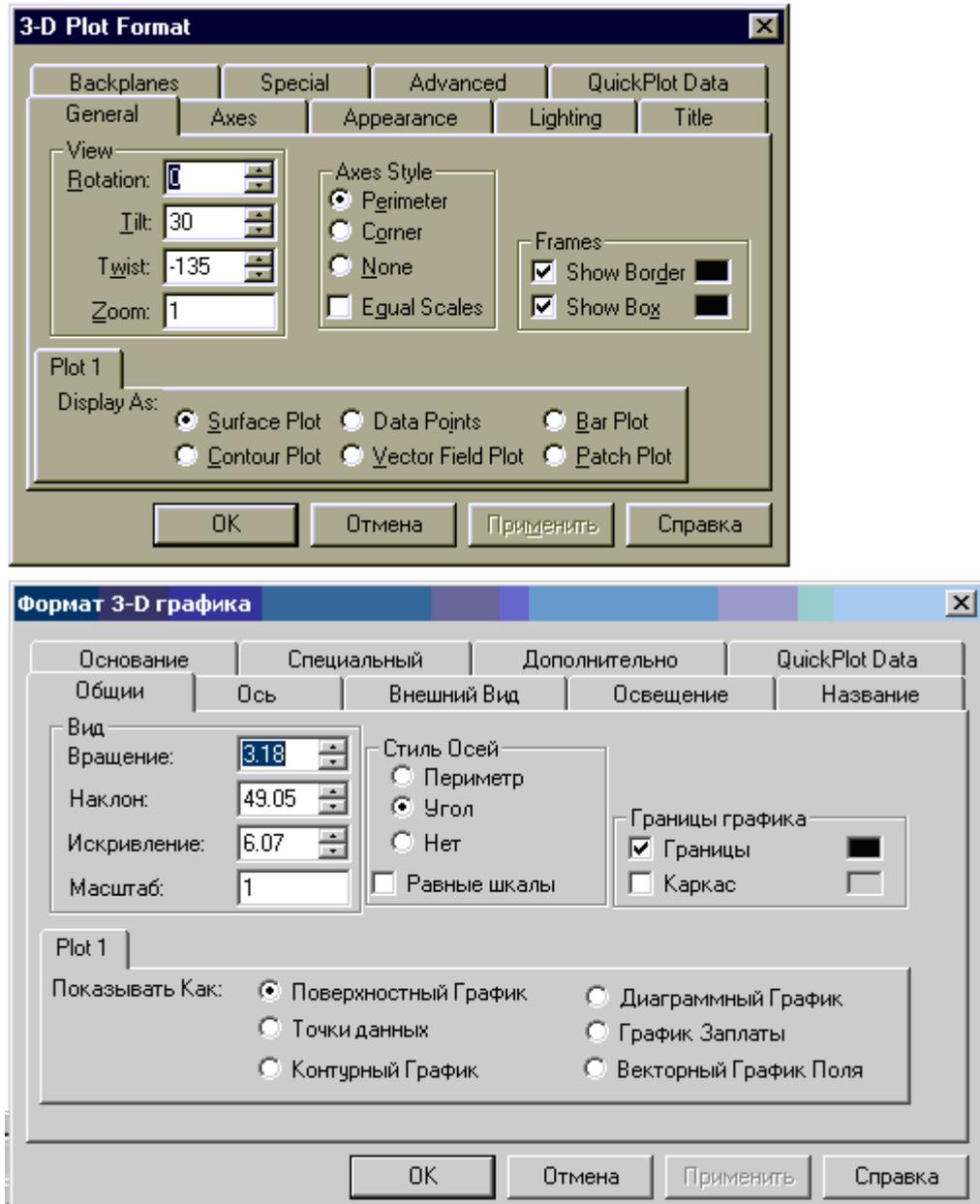


Рис. 12.

На вкладке **General (Общее)**:

— в области **View (Вид)** можно задать направление взгляда наблюдателя на трехмерный график. Значение в поле **Rotation** определяет угол поворота вокруг оси Z в плоскости $X-Y$. Значение в поле **Tilt** задает угол наклона линии взгляда к плоскости $X-Y$. Поле **Zoom** позволяет увеличить (уменьшить) графическое изображение в число раз, равное цифре, указанной в поле;

— в области **Axis Style (Стиль осей)** задать вид осей, выбрав селекторную кнопку **Perimetr (Периметр)** или **Corner (Угол)**. В первом случае оси всегда

Информационные технологии в менеджменте

находятся на переднем плане. При выборе кнопки **Corner** точка пересечения осей Ox и Oy задается элементом $A_{0,0}$ матрицы A ;

— в области **Frames (Границы графика)** опция **Show box (Каркас)** предназначена для отображения вокруг графика куба с прозрачными гранями, а опция **Show border (Границы)** позволяет заключить график в прямоугольную рамку;

— в области **Plot 1 (Plot 2...) Display as: (График 1 Показывать как:)** имеются селекторные кнопки для представления графика в других видах (контурный, точечный, векторное поле и др.);

Элементы вкладки **Axes (Ось)** позволяют изменять внешний вид осей координат (рис. 13).

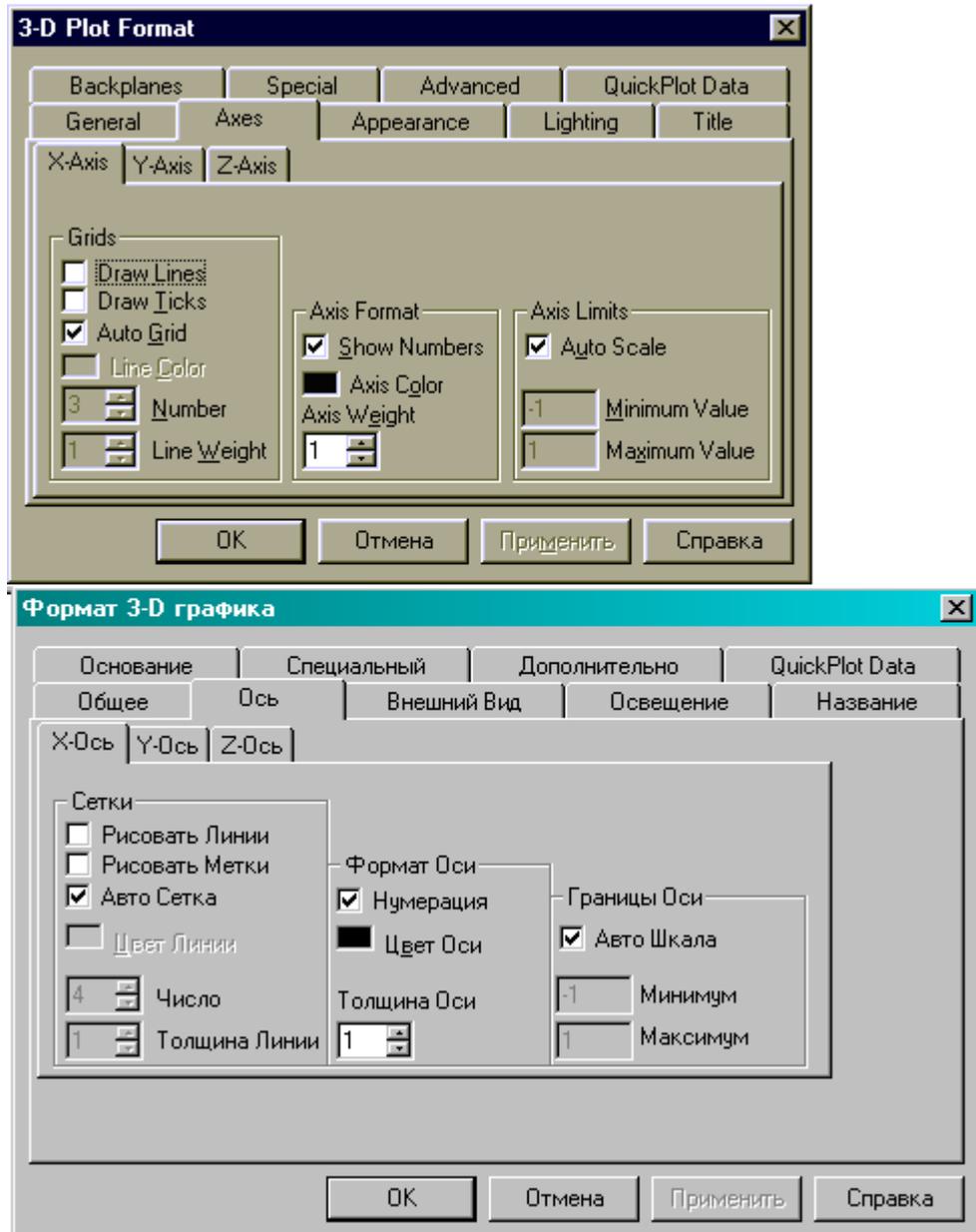


Рис. 13.

Посредством опций области **Grids (Сетки)** можно отобразить на графике линии, описываемые уравнениями $x, y, z = \text{const}$.

Если установлены опции **Show Numbers (Нумерация)**, отображаются метки на осях и подписи к ним.

Информационные технологии в менеджменте

При этом рядом с осями Ox и Oy указываются не значения узловых точек x_i, y_j , а значения индексов i и j , в то время как ось Oz размечается в соответствии с промежутком, которому принадлежат элементы матрицы значений $A_{i,j}$.

Если установлена опция **Auto Grid (Авто сетка)**, программа самостоятельно задает расстояние между соседними отметками на осях. Вы можете сами указать число линий сетки, если отключите указанную опцию.

Если установлена опция **Auto Scale (Авто шкала)**, то Mathcad сам определяет границы построения графика и масштабы по осям. Можно отключить данную опцию и для каждой оси самостоятельно задать пределы изменения переменных в полях **Minimum Value (Минимум)** и **Maximum Value (Максимум)**.

Вкладка **Appearance (Внешний вид)** позволяет изменять для каждого графика вид и цвет заливки поверхности (область **Fill Options**); вид, цвет и толщину дополнительных линий на графике (область **Line Options**); наносить на график точки данных (опция **Draw Points** области **Point Options**), менять их вид, размер и цвет.

Вкладка **Lighting (Освещение)** при включении опции **Enable Lighting (Включить освещение)** позволяет выбрать цветовую схему для освещения, установить несколько источников света, выбрав для них цвет освещения и определив его направление.

Вкладка **Backplanes (Основание)** позволяет изменить внешний вид плоскостей, ограничивающих область построения: цвет, нанесение сетки, определение ее цвета и толщины, прорисовка границ плоскостей.

На вкладке **Special (Специальный)** можно изменять параметры построения, специфичные для различных типов графиков.

Вкладка **Advanced (Дополнительно)** позволяет установить параметры печати и изменить цветовую схему для окрашивания поверхности графика, а также указать направление смены окраски (вдоль оси Ox, Oy или Oz). Включение опции **Enable Fog (Наличие тумана)** делает график нечетким, слегка размытым (полупрозрачным). При включении опции **Perspective (Перспектива)** появляется возможность указать в соответствующем поле расстояние до наблюдателя.

7. Создание анимации

Во многих случаях самый зрелищный способ представления результатов математических расчетов — это анимация. Mathcad позволяет создавать анимационные ролики и сохранять их в видеофайлах.

Основной принцип анимации в Mathcad — покадровая анимация. Ролик анимации — это просто последовательность кадров, представляющих собой некоторый участок документа, который выделяется пользователем. Расчеты производятся обособленно для каждого кадра, причем формулы и графики, которые в нем содержатся, должны быть функцией от номера кадра. Номер кадра задается системной переменной **FRAME**. По умолчанию, если не включен режим подготовки анимации, **FRAME=0**.

Последовательность действий для создания ролика анимации:

— ввести в документ необходимые выражения и графики, в которых участвует переменная номера кадра **FRAME**;

— подготовить часть документа, которая будет анимацией, таким образом, чтобы она находилась в поле зрения на экране. Выбрать панель инструментов **Tools / Animation (Вид / Анимация)**;

Информационные технологии в менеджменте

— в диалоговом окне **Animate (Анимация)** задать номер первого кадра в поле **From (От)**, номер последнего кадра в поле **To (До)** и скорость анимации в поле **At (Скорость)** в кадрах в секунду (рис. 1);

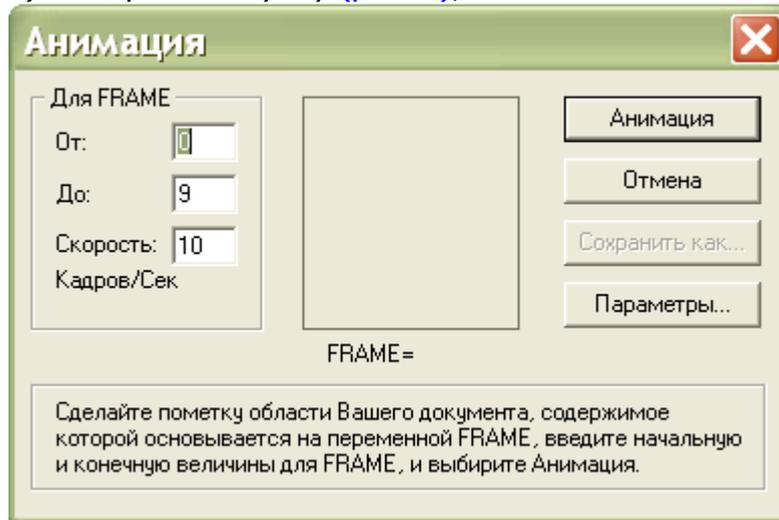


Рис. 1.

— выделить протаскиванием указателя мыши при нажатой левой кнопке мыши область в документе, которая станет роликом анимации;

— в диалоговом окне **Animate (Анимация)** нажать кнопку **Animate (Анимация)**. После этого в окошке диалогового окна **Animate (Анимация)** будут появляться результаты расчетов выделенной области, сопровождающиеся выводом текущего значения переменной **FRAME**. По окончании этого процесса на экране появится окно проигрывателя анимации (рис. 2);

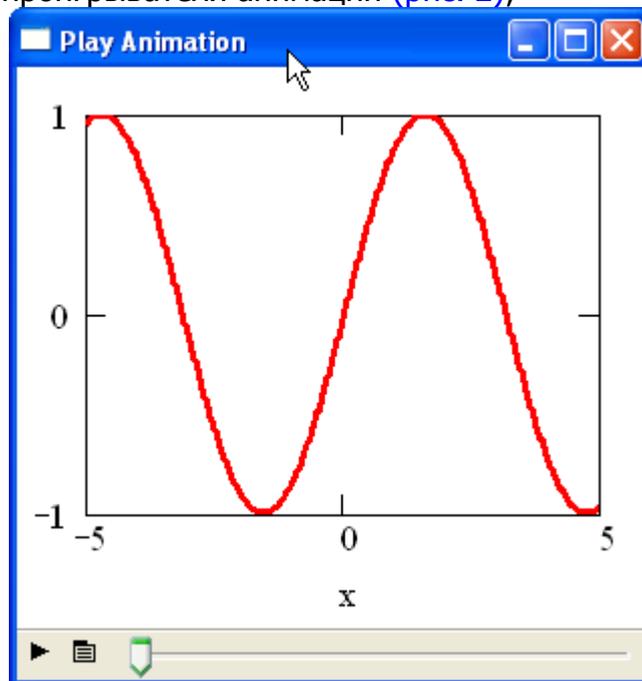


Рис. 2.

— запустить просмотр анимации в проигрывателе нажатием кнопки воспроизведения в левом нижнем углу окна проигрывателя;

— в случае если вид анимации Вас устраивает, сохраните ее в виде видеофайла, нажав кнопку **Save As (Сохранить как)** в диалоговом окне **Animate (Анимация)**. В появившемся диалоговом окне **Save Animation (Сохранить анимацию)** обычным для Windows способом укажите имя файла и его расположение на диске.

Информационные технологии в менеджменте

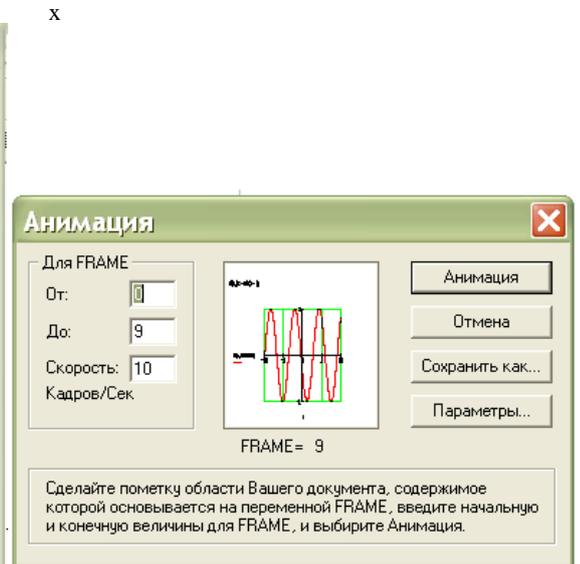
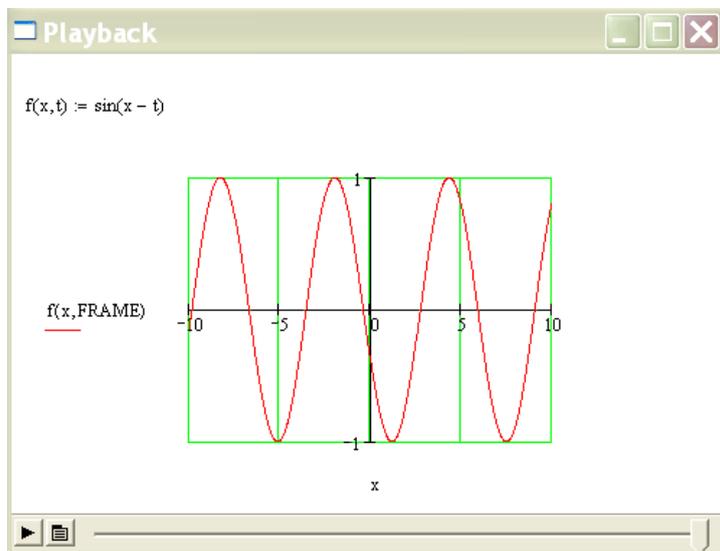
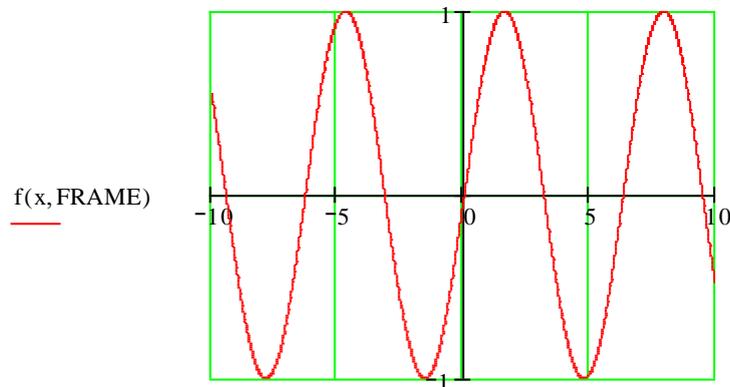
Сохраненный видеофайл можно использовать за пределами Mathcad. Таким образом, запуская видеофайлы, можно устроить красочную презентацию результатов работы.

При создании файлов анимации допускается выбирать программу видеосжатия (кодек) и качество компрессии. Делается это с помощью кнопки **Options (Параметры)** в диалоговом окне **Animate (Анимация)**.

Пример 1. Создание ролика анимации, демонстрирующего перемещение гармонической бегущей волны. При этом каждый момент времени будет задаваться переменной **FRAME**.

Подготовка сводится к определению функции $f(x,t) := \sin(x-t)$ и созданию ее графика $f(x, \text{FRAME})$.

$$f(x, t) := \sin(x - t)$$



Лекция № 5

Mathcad: Задание и обработка массивов. Символьные вычисления

1. Задание индексированной переменной
2. Операторы суммы Σ и произведения Π
3. Задание и обработка массивов
4. Символьные вычисления

1. Задание индексированной переменной

В Mathcad для задания и обработки числовых последовательностей используется индексированная переменная, обозначение которой содержит ее имя с указанием индекса (или индексов). Индекс в обозначении является самостоятельной переменной или выражением.

Для записи в документе индексированной переменной используется символ квадратной скобки []. Индексированная переменная может содержать не более двух индексов:

Введите	Чтобы увидеть
$x[j]$	x_j
$y[j-1]$	y_{j-1}
$M[i,j]$	$M_{i,j}$

Индексированная переменная задается с помощью оператора присваивания:

Индексированная_переменная := выражение/список_значений

Понятно, что все переменные, входящие в индексное выражение, должны быть определены, например, как переменные диапазона:

$$i := 1..5 \quad y_i := i^2 \cdot \sin\left(\frac{\pi}{i}\right)$$

$$i := 1..3 \quad j := 1..4 \quad M_{i,j} := i + \sqrt{j}$$

Если значения переменной задаются списком, то его элементы при вводе разделяются запятыми. При этом вводимые значения формируют таблицу, называемую **таблицей ввода**:

$$i := 1..4 \quad x_i :=$$

23.5
7.4
4
-4.6

Чтобы выполнить вставку пропущенных значений в таблице, следует выделить следом курсора тот элемент, после или до которого (в зависимости от вида следа) должны быть введены значения, и ввести символ-разделитель «запятая», после чего заполнить отведенное знакоместо. Если добавляемых значений несколько, то ввод каждого (кроме последнего) оканчивайте вводом запятой. Удаление элементов таблицы осуществляется точно так же, как и любой фрагмент формулы.

Информационные технологии в менеджменте

Значения индексированной переменной можно вычислять и с помощью итерационных формул, зная одно или несколько ее значений:

$$i := 1..5 \quad p := 0.15 \quad x_0 := 12 \quad x_i := x_{i-1} \cdot (1 + p)$$

$$N := 10 \quad f_0 := 0 \quad f_1 := 1 \quad i := 2..N \quad f_i := f_{i-1} + f_{i-2}$$

Чтобы получить значения индексированной переменной в виде таблицы, используйте оператор результата =.

Пример 5.1. Для N видов продукции известны плановый и фактический объемы выпуска. Найдите отклонение фактического объема от планового.

Решение:

$N := 5$	$i := 1..N$	$D_i := P_i - F_i$	$D_i =$
$P_i :=$	$F_i :=$		
100	101		-1
120	119		1
13.5	13		0.5
94	96		-2
450	460		-10

Пример 5.2. Пусть D_i, M_i, Y_i – соответственно день, месяц и год поступления на работу i-го рабочего, $i=1,2,\dots,N$. Определите стаж N рабочих (в полных годах) на заданную дату, задаваемую днем, номером месяца и годом.

Решение:

Определение трудового стажа

$N := 5$ Количество рабочих
 $i := 1..N$

Даты поступления на работу: $D_i :=$ $M_i :=$ $Y_i :=$

5	7	1993
14	10	1989
17	5	1985
2	3	1980
12	12	1979

$$\text{Day}(d, m) := 100 \cdot m + d$$

Определение стажа на дату: $Do := 18$ $Mo := 3$ $Yo := 2002$

$$S_i := Yo - Y_i - \text{if}(\text{Day}(D_i, M_i) > \text{Day}(Do, Mo), 1, 0)$$

$S_i =$

8
12
16
22
22

2. Операторы суммы Σ и произведения Π

Вычисление сумм и произведений в пакете Mathcad осуществляется с использованием панели **Calculus (Калькуляция)** на панели инструментов **Math (Математика)**.

Операция	Запись	Клавиша или комбинация клавиш
Сумма (индекс суммирования должен быть определен заранее)	$\sum x_i$	Shift+4 (символ \$)
Сумма (значения индекса суммирования задаются как элементы оператора)	$\sum_{i=1}^N x_i$	Ctrl+Shift+4
Произведение (индекс произведения должен быть определен заранее)	$\prod x_i$	Shift+3 (символ #)
Произведение (значения индекса произведения задаются как элементы оператора)	$\prod_{i=1}^N x_i$	Ctrl+Shift+3

Ввод оператора приводит к выводу на экран соответствующего шаблона с пустыми знаками, которые следует заполнить требуемыми выражениями:

$$\sum, \quad \sum_{i=}, \quad \prod, \quad \prod_{i=}$$

Индекс в операторах является простой переменной и может принимать только неотрицательные целые значения. Выражения, стоящие под знаками суммы или произведения, сами в свою очередь могут содержать названные операторы. Например,

$$\sum_{i=1}^{10} \left(\cos(2 \cdot j) + \prod_{j=1}^i j + \sum_{j=i}^N j^2 \right)$$

Использование операторов суммы и произведения, для которых указывается только имя переменной индекса, требует предварительного ее задания как переменной диапазона с произвольным целым шагом.

Пример 5.5. Для заданного N найдите сумму:

$$\frac{1}{2 \cdot 3} + \frac{2}{3 \cdot 4} + \dots + \frac{N}{(N+1) \cdot (N+2)}$$

Решение:

$$N := 100$$

$$S := \sum_{n=1}^N \frac{n}{(n+1) \cdot (n+2)} \quad S = 3.217$$

Пример 5.6. Для числовой последовательности, состоящей из N членов, n -е значение которой задается выражением $\sin(2n+1)$, найти отношение суммы членов с четными номерами к сумме членов с нечетными номерами. Нумерация членов начинается с единицы.

Решение:

$$N := 20$$

$$i := 2, 4..N \quad j := 1, 3..N$$

$$k := \frac{\sum_{i=2,4..N} \sin(2 \cdot i + 1)}{\sum_{j=1,3..N} \sin(2 \cdot j + 1)} \quad k = -1.011$$

Второй вид операторов суммы и произведения содержит задание диапазона изменения переменной индекса, шаг изменения значений которой по умолчанию равен 1. С использованием оператора суммы второго вида решение может быть оформлено следующим образом:

$$N := 20$$

$$\frac{\sum_{i=1}^N \sin(2 \cdot i + 1) \cdot (\text{mod}(i, 2) = 0)}{\sum_{i=1}^N \sin(2 \cdot i + 1) \cdot (\text{mod}(i, 2) = 1)} = -1.011$$

Пример 5.7. Найти значение выражения $b_1 + b_2 s_1 + b_3 s_1 s_2 + \dots + b_N s_1 s_2 \dots s_{N-1}$, где $b_i = i^2 \cdot \sin(i)$ и $s_i = \cos(2i)$.

Решение:

$$N := 10$$

$$i := 1..N$$

$$b_i := i^2 \cdot \sin(i) \quad s_i := \cos(2 \cdot i)$$

$$b_1 + \sum_{i=2}^N \left(b_i \cdot \prod_{j=1}^{i-1} s_j \right) = -1.791$$

3. Задание и обработка массивов

Общее название для векторов и матриц в Mathcad — *массивы*.

Задание массивов. Существуют три способа задания массивов:

1. Путем заполнения пустых полей. Для этого необходимо нажать Ctrl+M (или выбрать пункт меню **Insert** → **Matrix (Insert** → **Matrix)** или панель инструментов **Math** → **Matrix (Математика** → **Матрица)**) и в появившемся диалоговом окне выбрать размерности матрицы (если одна из них равна 1 — вектора) (рис. 5.2). После нажатия кнопки «ОК» в рабочем листе Mathcad появится изображение матрицы заданной размерности с пустыми полями для числовых значений, которые далее необходимо заполнить вручную.

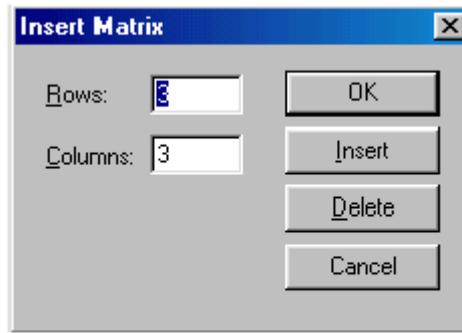


Рис. 5.2. Диалоговое окно **Insert Matrix** (Вставка массива)

Изменение размерностей матриц производится с помощью того же диалогового окна (кнопки **Insert** и **Delete**), в котором указывается количество добавляемых или удаляемых строк и столбцов. Предварительно в матрице курсор устанавливают в позицию добавления или удаления.

В Mathcad вектор-строка задается как двумерный массив, состоящий из одной строки.

После закрытия окна по нажатию кнопки ОК в документ будет вставлен шаблон массива заданной размерности с незаполненными знаками. Ниже в таблице приведены шаблоны, соответствующие указанной в клетках первой строки размерности:

3 x 1	3 x 3	1 x 3
$\begin{pmatrix} \blacksquare \\ \blacksquare \\ \blacksquare \end{pmatrix}$	$\begin{pmatrix} \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \end{pmatrix}$	$(\blacksquare \ \blacksquare \ \blacksquare)$

Шаблон следует заполнить заданными значениями для элементов массива, перемещая курсор по знакам. Заполнение может выполняться в любом порядке, редактирование введенных значений подчиняется тем же правилам, что и редактирование формул. Знакоместа могут быть заполнены не обязательно числами, но и произвольными выражениями, допустимыми в Mathcad, например:

$$\begin{pmatrix} 3.6 & a + b & -12 \\ \sqrt{a} & 0 & \frac{\sin(x)}{4} \\ 1.5 & a^2 + 4 & 17 \end{pmatrix}$$

Массиву можно присвоить имя с помощью оператора присваивания:

$$\text{Имя} := \text{Массив}$$

Например,

$$X := \begin{pmatrix} 12 \\ -4 \\ 5 \end{pmatrix} \quad A := \begin{pmatrix} 1 & -2 & 0 \\ 23 & 2.5 & 1 \\ 12 & -1 & 1 \end{pmatrix} \quad y := (12 \ -2 \ 1)$$

Обращение к элементам массива осуществляется по имени с указанием индекса (для вектора-столбца) или индексов (для матрицы). Элемент матрицы

Информационные технологии в менеджменте

идентифицируется парой индексов, которые разделяются запятыми. Первый индекс определяет номер строки матрицы, второй – номер ее столбца, на пересечении которых находится элемент. Например $A_{i,j}$, где i — номер строки, j — номер столбца. Ввод индексов осуществляется при помощи клавиши «[» или используя знак X_n на панели инструментов **Matrix (Матрица)**.

Понятно, что элементы массива могут быть заданы с помощью индексированной переменной.

Наименьшее значение номеров строки и столбца определяет встроенная переменная **ORIGIN**. Если она имеет значение 0, то нумерация строк и столбцов в массиве начинается с 0. Установить желаемый способ нумерации можно, изменив значение переменной **ORIGIN** с помощью оператора присваивания.

По умолчанию нумерация столбцов и строк начинается с 0.

Часто размерность массива очень велика и он становится очень громоздким для отображения, поэтому Mathcad отображает матрицы и вектора с размерностями больше девяти в виде таблиц с полосами прокрутки.

2. Используя ранжированные переменные (когда имеется явная зависимость между элементами массива и их индексами).

Пример.

$i := 0..10$ $j := 0..10$

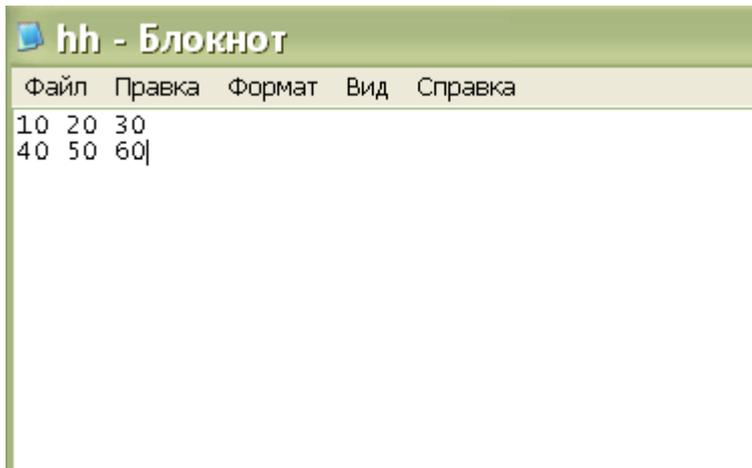
$$A_{i,j} := i^2 - j \cdot \sin(i)$$

A =

	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	1	0.159	-0.683	-1.524	-2.366	-3.207	-4.049	-4.89	-5.732	-6.573	-7.415
2	4	3.091	2.181	1.272	0.363	-0.546	-1.456	-2.365	-3.274	-4.184	-5.093
3	9	8.859	8.718	8.577	8.436	8.294	8.153	8.012	7.871	7.73	7.589
4	16	16.757	17.514	18.27	19.027	19.784	20.541	21.298	22.054	22.811	23.568
5	25	25.959	26.918	27.877	28.836	29.795	30.754	31.712	32.671	33.63	34.589
6	36	36.279	36.559	36.838	37.118	37.397	37.676	37.956	38.235	38.515	38.794
7	49	48.343	47.686	47.029	46.372	45.715	45.058	44.401	43.744	43.087	42.43
8	64	63.011	62.021	61.032	60.043	59.053	58.064	57.074	56.085	55.096	54.106
9	81	80.588	80.176	79.764	79.352	78.939	78.527	78.115	77.703	77.291	76.879
10	100	100.544	101.088	101.632	102.176	102.72	103.264	103.808	104.352	104.896	105.44

3. Считывая элементы массива из файлов данных. Для этого необходимо на панели инструментов **Вставка** → **Компонент** выбрать программу, из которой требуется получить данные (например, Excel, Matlab). Также здесь можно выбрать операцию **Чтение или запись файла**.

Пример. В файле hh.txt, созданном в Блокноте содержатся данные вида (пробелом отделяются элементы, а переход на новую строка определяет новую строку матрицы):



Чтобы иметь возможность использовать эти данные в Mathcad необходимо выбрать: **Вставка** → **Компонент** → **Чтение или запись файла**, затем указать какую операцию выполнять (чтение или запись) и путь к файлу с данными.

Mathcad осуществит вставку фрагмента на лист:

s :=  E:\hh.txt

$$s = \begin{pmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \end{pmatrix}$$

Или можно воспользоваться функциями:

x := READPRN("hh.txt")

$$x = \begin{pmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \end{pmatrix}$$

— для чтения данных из файла;

WRITEPRN("hh1.txt") := x 1(— для записи данных в файл.

Обработка массивов. Над массивами в Mathcad могут выполняться бинарные арифметические операции: сложение, вычитание, умножение массивов при условии согласования размерностей операндов, а также умножение массива на скалярное выражение. Перечень унарных операций дан в [таблице 5.1](#).

Таблица 5.1. Унарные операции над массивами

Операция	Запись	Комбинация клавиш	Панель инструментов
Транспонирование матрицы	A^T	Ctrl+1	Matrix
Обращение матрицы	A^{-1}	^	Matrix
Определитель матрицы, длина вектора	$ A $		Matrix
Столбец матрицы	$A^{<n>}$	Ctrl+6	Matrix
Элемент вектора	a_n	[Matrix
Элемент матрицы	$A_{i,j}$	[Matrix

Информационные технологии в менеджменте

Пример 5.16. Даны две матрицы A и B:

$$A = \begin{pmatrix} 2 & -3 & 2 \\ 1 & 2 & 4 \\ -1 & 5 & 2 \end{pmatrix} \quad B = \begin{pmatrix} -1 & 4 & 0 \\ -2 & 3 & 1 \\ 2 & 0 & 4 \end{pmatrix}$$

Выполните над матрицами следующие действия: $3(A+B)(2B^{-1}-A)+A^2$

Решение:

$$A := \begin{pmatrix} 2 & -3 & 2 \\ 1 & 2 & 4 \\ -1 & 5 & 2 \end{pmatrix} \quad B := \begin{pmatrix} -1 & 4 & 0 \\ -2 & 3 & 1 \\ 2 & 0 & 4 \end{pmatrix}$$

$$3 \cdot (A + B) \cdot (2 \cdot B^{-1} - A) + A^2 = \begin{pmatrix} -1.857 & -29.857 & -30.786 \\ 7.714 & -85.286 & -60.429 \\ 3.571 & -85.429 & -71.643 \end{pmatrix}$$

Таблица 1

Функции для оперирования с массивами в пакете Mathcad

<i>Имя функции</i>	<i>Возвращаемое значение</i>
rows(A)	количество строк в массиве A
cols(A)	количество столбцов в массиве A
length(v)	длина вектора v
last(v)	индекс последнего элемента вектора v
max(A)	максимальный элемент массива A
min(A)	минимальный элемент массива A
tr(A)	след матрицы A (сумма диагональных элементов)
rank(A)	ранг матрицы A
norme(A)	евклидова норма матрицы A, то есть корень квадратный из суммы квадратов всех элементов
mean(A)	среднее значение элементов массива A
sort(v)	упорядочивание (сортировка) элементов вектора v по возрастанию
csort(A,n)	сортировка матрицы A по элементам n-го столбца (перестановка строк при упорядочении элементов указанного столбца в возрастающем порядке)
rsort(A,n)	сортировка матрицы A по элементам n-й строки (перестановка столбцов при упорядочении элементов указанной строки в возрастающем порядке)
reverse(A)	перестановка строк массива A в обратном порядке, так что результатом обращения к функциям reverse(sort(v)) будет вектор, элементы которого отсортированы в убывающем порядке
polyroots(v)	поиск корней полинома с коэффициентами, заданными вектором-столбцом v. Коэффициенты перечисляются в направлении увеличения степени неизвестной, начиная с нулевой
augment(A,B)	создание массива, полученного добавлением к массиву A

	справа массива B (склеивание массивов с одинаковым количеством строк). Список аргументов может содержать и более двух массивов
identity(n)	единичная матрица размерности $n \times n$
diag(v)	диагональная матрица с элементами вектора v на диагонали
matrix(m,n,f)	матрица размером $m \times n$, значения элементов матрицы определяются f — функцией $f(i,j)$ от двух переменных (номера строки и номера столбца). Эта функция должна быть предварительно определена пользователем. Пример: <pre>f(i,j) := i + j i := 0..2 j := 0..3 M := matrix(3,4,f) R_{i,j} := i + j M = $\begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \end{pmatrix}$ R = $\begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \end{pmatrix}$</pre>
eigenvals(A)	вектор, элементы которого являются собственными числами матрицы A
eigenvecs(A)	матрица, состоящая из нормализованных собственных векторов матрицы A

4. Символьные вычисления

Системы компьютерной математики снабжаются специальным процессором для выполнения аналитических (символьных) вычислений. Его основой является ядро, хранящее всю совокупность формул и формульных преобразований. Чем больше этих формул в ядре, тем надежней работа символьного процессора и тем вероятней, что поставленная задача будет решена.

Ядро символьного процессора Mathcad — несколько упрощенный вариант известной системы символьной математики Maple фирмы Waterloo Maple Software, у которой фирма MathSoft (разработчик системы Mathcad) приобрела лицензию на его применение в системе Mathcad. Благодаря этому Mathcad стала системой символьной математики и перешла в разряд универсальных систем компьютерной математики. Правда, доступ к ядру осуществляется лишь частично, так как все-таки основное назначение Mathcad — это производить численные расчеты.

Символьные вычисления осуществляются с помощью команд меню **Symbolics** на панели **Math** (рис. 1). В табл. 8.4 приведены функции основных символьных преобразований.

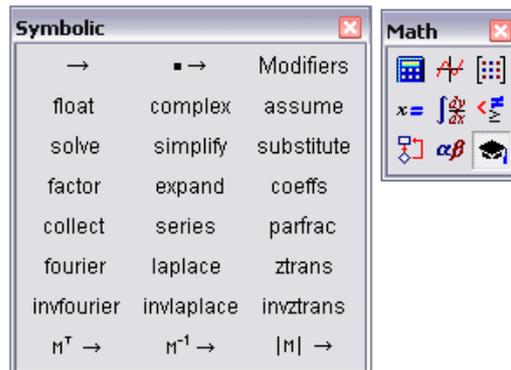


Рис. 1. Математическая панель для символьных вычислений

Информационные технологии в менеджменте

Таблица 8.4. Описание директив панели **Symbolic**

Назначение	Шаблон
Символьное вычисление выражения	■ →
Символьное вычисление выражения с указанием ключевого слова операции	■ ■ →
Преобразование в формат чисел с плавающей точкой	■ float, ■ →
Представление в комплексной форме	■ complex →
Решение уравнений, неравенств и их систем	■ solve, ■ →
Упрощение выражения	■ simplify →
Замена в исходном выражении переменной или подвыражения заданным выражением	■ substitute, ■ = ■ →
Разложение выражения на простые множители	■ factor, ■ →
Разложение выражения по степеням	■ expand, ■ →
Возвращение вектора с коэффициентами полинома	■ coeffs, ■ →
Разложение выражений с приведением подобных	■ collect, ■ →
Разложение в ряд по заданным переменным	■ series, ■, ■ →
Разложение на элементарные дроби	■ convert, parfrac, ■ →
Транспонирование матрицы	$M^T \rightarrow$
Построение обратной матрицы	$M^{-1} \rightarrow$
Вычисление определителя матрицы	■ →

<i>Вид преобразования</i>	<i>Пункты меню Symbolics</i>
Преобразование Лапласа	Laplace
Обратное преобразование Лапласа	Invlaplace
Преобразование Фурье	Fourier
Обратное преобразование Фурье	Invfourier

Для ряда операций следует указать переменную, относительно которой выполняется та или иная символьная операция.

В результате преобразований могут появляться специальные функции — как встроенные в систему (функции Бесселя, гамма-функция, интеграл вероятности и др.), так и ряд функций, дополнительно определенных при загрузке символьного процессора (интегральные синус и косинус, интегралы Френеля, эллиптические интегралы и др.). Последние нельзя использовать при создании математических выражений.

Пример 8.24 (Использование оператора символьного вывода).

$$2 + \sqrt{56} - \frac{4}{5} \rightarrow \frac{6}{5} + 2 \cdot \sqrt{14}$$

$$\sqrt{144} + 4\sqrt{24} + 5\sqrt{96} \rightarrow 12 + 28 \cdot \sqrt{6}$$

$$\sin(2\pi) + \cos\left(12 \cdot \frac{\pi}{5}\right) \rightarrow \frac{1}{4} \cdot \sqrt{3} - \frac{1}{4}$$

$$\begin{pmatrix} 1 & 3 \\ a & 2 \end{pmatrix} \cdot \begin{pmatrix} b & a \\ 1 & b \end{pmatrix} \rightarrow \begin{pmatrix} b+3 & a+3 \cdot b \\ a \cdot b+2 & a^2+2 \cdot b \end{pmatrix}$$

$$\frac{d^2}{dx^2}(1 + a \cdot \cos(x)) \rightarrow -a \cdot \cos(x)$$

$$\int \cos(x) dx \rightarrow \sin(x)$$

$$\lim_{x \rightarrow 0} (1 + a \cdot x)^{\frac{1}{x}} \rightarrow \exp(a)$$

Символьная операция **Simplify (Упрощение выражения)** — одна из самых важных. Эта операция позволяет упрощать математические выражения, содержащие алгебраические и тригонометрические функции, а также выражения со степенными многочленами (полиномами). Упрощение означает замену более сложных фрагментов выражений на более простые. Приоритет тут отдается простоте функций. К примеру, функция **tan(x)** считается более сложной, чем функции **sin(x)** и **cos(x)**. Поэтому **tan(x)** упрощается так, что получает представление через соотношение этих функций.

Эта команда открывает широкие возможности для упрощения сложных и плохо упорядоченных алгебраических выражений. На [рис. 2](#) даны примеры применения операции **Simplify**.

$$\begin{array}{ll} \sin(x)^2 + \cos(x)^2 & 1 \\ \frac{x^2 - 3 \cdot x - 4}{x - 4} + 2 \cdot x - 5 & 3 \cdot x - 4 \\ \frac{d}{dx} \sin(3 \cdot x) & 3 \cdot \cos(3 \cdot x) \\ \int_a^b e^{-t} dt & -\exp(-b) + \exp(-a) \end{array}$$

Рис. 2.

Система Mathcad содержит встроенную функцию для вычисления значений определенных интегралов приближенным численным методом. Ею целесообразно пользоваться, когда нужно просто получить значение определенного интеграла в виде числа.

Однако команда **Simplify** применительно к вычислениям определенных интегралов делает гораздо больше — она ищет аналитическое выражение для интеграла. Более того, она способна делать это и при вычислении кратных интегралов, пределы которых — функции. Наглядный пример этому продемонстрирован на рис. 3.

$$\int_0^1 \int_0^{\sqrt{1+x^2}} \frac{1}{(x^2 + y^2 + 1)} dy dx = \frac{-1}{4} \cdot \ln(\sqrt{2} - 1) \cdot \pi$$

$$\int_0^{\frac{\pi}{2}} \int_0^{\frac{\pi}{4}} \int_0^{4 \cdot \cos(\psi)} \rho^2 \sin(\psi) d\rho d\psi d\theta = 2 \cdot \pi$$

Рис. 3.

Операцию **Simplify** можно использовать и для вычисления сумм и произведений символьных последовательностей. Результат операции, как и следовало ожидать, получается в символьной форме (если она существует) (рис. 4).

$$\sum_i j^3 = \frac{1}{4} \cdot j^4 - \frac{1}{2} \cdot j^3 + \frac{1}{4} \cdot j^2$$

$$\prod_k \frac{k^2}{(k-1)} = \Gamma(k) \cdot (k-1)$$

Рис. 4.

Пример 8.33 (Упрощение выражения).

$$23 + \frac{\frac{1}{2} + \frac{3}{4} - \frac{12}{49}}{45} + \frac{1}{2} - \frac{12}{36} \text{ simplify} \rightarrow \frac{204527}{8820}$$

$$\left(\frac{a-3}{a^2-3a+9} - \frac{6a-18}{a^3+27} \right) \cdot \left[\frac{5(a-3)^2}{2a^3+54} \right]^{-1} \text{ simplify} \rightarrow \frac{2}{5}$$

$$\left(1 + \frac{3x+x^2}{3+x} \right) \cdot \left(\frac{1}{x+1} - \frac{x}{1+2x+x^2} \right) \text{ simplify} \rightarrow \frac{1}{(x+1)}$$

Информационные технологии в менеджменте

$$\frac{d}{dx} (x \cdot \sqrt{1+x^2}) \text{ simplify} \rightarrow \frac{(1+2 \cdot x^2)}{(1+x^2)^{\frac{3}{2}}}$$

$$\frac{d}{dx} \frac{d}{dx} \left(\frac{x}{\sqrt{a^2-x^2}} \right) \text{ simplify} \rightarrow 3 \cdot x \cdot \frac{a^2}{(a^2-x^2)^{\frac{5}{2}}}$$

$$\int \frac{1}{1-\cos(x)} dx \text{ simplify} \rightarrow \frac{-1}{\tan\left(\frac{1}{2} \cdot x\right)}$$

$$\int_0^2 \int_a^b (a \cdot y - x^2) dx dy \text{ simplify} \rightarrow 2 \cdot a \cdot b - \frac{2}{3} \cdot b^3 - 2 \cdot a^2 + \frac{2}{3} \cdot a^3$$

$$\sum_{n=1}^4 2 \frac{n}{n^2+1} \text{ simplify} \rightarrow \frac{244}{85}$$

$$\prod_{n=1}^5 \frac{n}{n^3+2} \text{ simplify} \rightarrow \frac{2}{121539}$$

$$\sum_{i=0}^n a^i \text{ simplify} \rightarrow \frac{[a^{(n+1)} - 1]}{(a-1)}$$

$$\sum_{n=0}^{\infty} a^n \text{ simplify} \rightarrow \frac{-1}{(a-1)}$$

$$\sum_{n=0}^{\infty} \frac{x^n}{2^n \cdot n!} \text{ simplify} \rightarrow \exp\left(\frac{1}{2} \cdot x\right)$$

Действие операции **Expand (Раскрытие скобок)** в известном смысле противоположно действию операции **Simplify**. Подвергаемое преобразованию выражение расширяется с использованием известных (и введенных в символьное ядро) соотношений, например алгебраических разложений многочленов, произведений углов и т. д. Разумеется, расширение происходит только в том случае, когда его результат однозначно возможен. Иначе нельзя считать, что действие этой операции противоположно действию операции **Simplify**. К примеру, операция **Simplify** преобразует сумму квадратов синуса и косинуса в 1, тогда как обратное преобразование многозначно и потому в общем виде невыполнимо.

При преобразовании выражений операция **Expand** старается более простые функции представить через более сложные, свести алгебраические выражения, представленные в сжатом виде, к выражениям в развернутом виде и т. д. Примеры действия операции **Expand** даны на [рис. 5](#).

Информационные технологии в менеджменте

$$\begin{array}{ll} \sin(5 \cdot x) & 16 \cdot \sin(x) \cdot \cos(x)^4 - 12 \cdot \sin(x) \cdot \cos(x)^2 + \sin(x) \\ (x - y)^2 \cdot (x + y) & x^3 - x^2 \cdot y - x \cdot y^2 + y^3 \\ (a + b)^n & (a + b)^n \\ \prod_n \frac{1}{n^2} & \frac{1}{\Gamma(n)^2} \end{array}$$

Рис. 5.

Последний пример показывает, что результатом операции может быть специальная математическая функция, которая считается более сложным выражением, чем порождающее ее выражение. С виду, однако, выражения со специальными математическими функциями обычно выглядят гораздо проще, чем исходные выражения.

Пример 8.28 (Раскрытие скобок). Второе знакоместо шаблона можно не заполнять, удалив его.

$$\begin{array}{l} (a + b)^3 \text{ expand} \rightarrow b^3 + 3 \cdot b^2 \cdot a + 3 \cdot b \cdot a^2 + a^3 \\ \sin(4x) \text{ expand} \rightarrow 8 \cdot \sin(x) \cdot \cos(x)^3 - 4 \cdot \sin(x) \cdot \cos(x) \\ \sin(4x) \text{ expand, x} \rightarrow 8 \cdot \sin(x) \cdot \cos(x)^3 - 4 \cdot \sin(x) \cdot \cos(x) \end{array}$$

Отличия в результатах выполнения преобразования с заполнением второго знакоместа шаблона или без заполнения шаблона можно увидеть на следующих примерах:

$$\begin{array}{l} \left(\frac{1}{x+1} - x^2\right) \cdot (x+1) \text{ expand, x} \rightarrow \frac{1}{(x+1)} \cdot x + \frac{1}{(x+1)} - x^3 - x^2 \\ \left(\frac{1}{x+1} - x^2\right) \cdot (x+1) \text{ expand} \rightarrow -x^3 - x^2 + 1 \\ \left(\frac{1}{x+1} - x^2\right) \cdot \left(1 + x + \frac{1}{x^2}\right) \text{ expand, x} \rightarrow \frac{1}{(x+1)} \cdot x + \frac{1}{(x+1)} + \frac{1}{(x+1) \cdot x^2} - x^3 - x^2 - 1 \\ \left(\frac{1}{x+1} - x^2\right) \cdot \left(1 + x + \frac{1}{x^2}\right) \text{ expand} \rightarrow \frac{(1 - x^6 - 2 \cdot x^5 - x^4)}{(x^3 + x^2)} \end{array}$$

Пример 8.25 (Разложение на элементарные дроби).

$$\frac{x^3 + 1}{x^3 - 5x^2 + 6x} \text{ convert, parfrac, x} \rightarrow 1 + \frac{1}{6 \cdot x} - \frac{9}{2 \cdot (x - 2)} + \frac{28}{3 \cdot (x - 3)}$$

$$\left(\frac{x}{x^2 - 3x + 2}\right)^2 \text{ convert, parfrac, x} \rightarrow \frac{1}{(x - 1)^2} + \frac{4}{(x - 1)} + \frac{4}{(x - 2)^2} - \frac{4}{(x - 2)}$$

$$\frac{a + x}{(x - a) \cdot (x + b)} \text{ convert, parfrac, a} \rightarrow \frac{-1}{(x + b)} + 2 \cdot \frac{x}{(x - a) \cdot (x + b)}$$

$$\frac{a + x}{(x - a) \cdot (x + b)} \text{ convert, parfrac, x} \rightarrow 2 \cdot \frac{a}{(a + b) \cdot (x - a)} - \frac{(-b + a)}{(a + b) \cdot (x + b)}$$

Операция **Coeffs (Определение вектора коэффициентов полинома)** применяется, если заданное выражение — полином (степенной многочлен) или может быть представлено таковым относительно выделенной переменной. Результатом операции является вектор с коэффициентами полинома (рис. 8).

$$a \cdot x^3 + b \cdot x^2 + c \cdot x + d \quad \begin{pmatrix} d \\ c \\ b \\ a \end{pmatrix} \quad \begin{array}{l} \text{относительно} \\ \text{переменной } x \end{array}$$

$$a \cdot x^3 + b \cdot x^2 + c \cdot x + d \quad \begin{pmatrix} a \cdot x^3 + b \cdot x^2 + d \\ x \end{pmatrix} \quad \begin{array}{l} \text{относительно} \\ \text{переменной } c \end{array}$$

Рис. 8.

Пример 8.26 (Определение вектора коэффициентов полинома).

$$x^3 - 3 \cdot x^2 + x - 5 \text{ coeffs, x} \rightarrow \begin{pmatrix} -5 \\ 1 \\ -3 \\ 1 \end{pmatrix}$$

$$\sin(x)^3 - 2 \cdot \sin(x) + 5 \text{ coeffs, sin(x)} \rightarrow \begin{pmatrix} 5 \\ -2 \\ 0 \\ 1 \end{pmatrix}$$

$$x^3 - 2 \cdot a \cdot x^2 + (a + b) \cdot x - 7 \text{ coeffs, x} \rightarrow \begin{pmatrix} -7 \\ b + a \\ -2 \cdot a \\ 1 \end{pmatrix}$$

Операция **Factor (Разложение на множители)** используется для факторизации — разложения выражений или чисел на простые множители. Она способствует выявлению математической сущности выражений; к примеру, наглядно выявляет представление полинома через его действительные корни, а в

Информационные технологии в менеджменте

том случае, когда разложение части полинома содержит комплексно-сопряженные корни, порождающее их выражение представляется квадратичным трехчленом. Примеры действия этой операции даны на рис. 6.

$$\begin{array}{ll} x^3 - 6x^2 + 11x - 6 & (x-1) \cdot (x-2) \cdot (x-3) \\ x^3 + 11x - 6x^2 - 6 & x^3 + 11x - 6 \cdot (x-1) \cdot (x+1) \\ x^3 - 6x^2 + 21x - 52 & (x-4) \cdot (x^2 - 2x + 13) \\ 123456 & (2)^6 \cdot (3) \cdot (643) \end{array}$$

Рис. 6.

В большинстве случаев (но не всегда) операция факторизации ведет к упрощению выражений. Термин «*факторизация*» не является общепризнанным в отечественной математической литературе, но мы его оставляем в связи с созвучностью с англоязычным именем этой операции.

Пример 8.27 (Разложение на множители). При заполнении шаблона следует удалить второе знакоместо.

$$7739550 \text{ factor} \rightarrow 2 \cdot 3^5 \cdot 5^2 \cdot 7^2 \cdot 13$$

$$x^2 - 5x + 6 \text{ factor} \rightarrow (x-2) \cdot (x-3)$$

$$(b^2 \cdot a^4 - 1) \cdot (4 \cos(x)^2 - 9) \text{ factor} \rightarrow (b \cdot a^2 - 1) \cdot (b \cdot a^2 + 1) \cdot (2 \cdot \cos(x) - 3) \cdot (2 \cdot \cos(x) + 3)$$

$$\frac{x+1}{x^2-4} \text{ factor} \rightarrow \frac{(x+1)}{(x-2) \cdot (x+2)}$$

$$\frac{1}{x^2-4} + \frac{x+1}{x^2-6x+8} \text{ factor} \rightarrow \frac{(x^2+4 \cdot x-2)}{(x-2) \cdot (x-4) \cdot (x+2)}$$

Операция **Collect (Приведение подобных слагаемых)** обеспечивает замену указанного выражения выражением, скомплектованным по базису указанной переменной, если такое представление возможно (рис. 7). В противном случае появляется окно с сообщением о невозможности комплектования по указанному базису.

$$\begin{array}{lll} (a+b+c)^2 & a^2 + (2 \cdot b + 2 \cdot c) \cdot a + (b+c)^2 & \text{по } a \\ (a+b+c)^2 & c^2 + (2 \cdot a + 2 \cdot b) \cdot c + (a+b)^2 & \text{по } c \\ (y-x) \cdot (z+y) & (y-x) \cdot z + (y-x) \cdot y & \text{по } z \\ (y-x) \cdot (z+y) & y^2 + (-x+z) \cdot y - xz & \text{по } y \end{array}$$

Рис. 7.

Эта команда особенно удобна, когда заданное выражение есть функция ряда переменных и нужно представить его в виде функции заданной переменной имеющей вид степенного многочлена. При этом другие переменные входят в сомножители указанной переменной, представленной в порядке уменьшения ее степени.

Пример 8.29 (Приведение подобных слагаемых).

$$5x^2 - [(x + 1) \cdot x - 4] \cdot x \text{ collect, } x \rightarrow -x^3 + 4 \cdot x^2 + 4 \cdot x$$

$$x^2 \cdot (y + 1) - (3x^2 \cdot y) - x - 6 \text{ collect, } x \rightarrow (-2 \cdot y + 1) \cdot x^2 - x - 6$$

$$x^2 \cdot (y + 1) - (3x^2 \cdot y) - x - 6 \text{ collect, } y \rightarrow -2 \cdot x^2 \cdot y + x^2 - x - 6$$

$$\sin(x) \cdot x + (\cos(x) \cdot \sin(x) + 3x^2 + 4) \cdot x \text{ collect, } x \rightarrow 3 \cdot x^3 + (\sin(x) + \sin(x) \cdot \cos(x) + 4) \cdot x$$

$$\sin(x) \cdot x + (\cos(x) \cdot \sin(x) + 3x^2 + 4) \cdot x \text{ collect, } \sin(x) \rightarrow (x + \cos(x) \cdot x) \cdot \sin(x) + (3 \cdot x^2 + 4) \cdot x$$

Пример 8.30 (Подстановка в выражение).

$$x^2 - 5x + 6 \text{ substitute, } x = u^2 \rightarrow u^4 - 5 \cdot u^2 + 6$$

$$a + \frac{S}{a - v} \text{ substitute, } a = v \cdot t \rightarrow v \cdot t + \frac{S}{(v \cdot t - v)}$$

$$\sin(x)^2 - (a \cdot \sin(x) + b) \text{ substitute, } \sin(x) = u \rightarrow u^2 - a \cdot u - b$$

$$\frac{1}{x + 1} - x^2 \text{ substitute, } x = 4 \rightarrow \frac{-79}{5}$$

Замечание. В записи команды символ = соответствует логическому оператору сравнения.

В выражении можно выполнить сразу же несколько подстановок вводом нужного количества двойных маркеров ■ = ■, разделенных запятыми:

$$a + b + \frac{a}{b + a} \text{ substitute, } a = 2, b = 4 \rightarrow \frac{19}{3}$$

или одновременным использованием нескольких команд:

$$a + b + \frac{a}{b + a} \left| \begin{array}{l} \text{substitute, } a = 2 \\ \text{substitute, } b = 4 \end{array} \right. \rightarrow \frac{19}{3}$$

✓ Ввод нескольких команд выполняется следующим образом:

- 1) ввести первую команду и охватить ее справа уголком курсора;
- 2) выбрать на панели очередную команду;
- 3) продолжить ввод в случае использования более двух команд одновременно.

Информационные технологии в менеджменте

Пример 8.31 (Задание формата представления).

$$2 + \sqrt{56} - \frac{4}{5} \text{ float,5} \rightarrow 8.6834$$

$$\sqrt{144} + 4\sqrt{24} + 5\sqrt{96} \text{ float,7} \rightarrow 80.58572$$

$$\sin(2\pi) + \cos\left(12 \frac{\pi}{5}\right) \text{ float,4} \rightarrow .3090$$

$$\sqrt{-124} \text{ complex} \rightarrow 2 \cdot i \cdot \sqrt{31}$$

$$23 + \sqrt{-6} \text{ complex} \rightarrow 23 + i \cdot \sqrt{6}$$

Пример 8.32 (Разложение в степенной ряд). Для команды **series** двойные маркеры вводятся: имя (или значение) переменной, относительно которой выполняется разложение в степенной ряд. Последний маркер содержит число, задающее порядок ошибки. Но последний маркер может быть исключен, тогда в этом случае команда **series** выводит ряд с точностью до членов шестого порядка. Если в первом маркере после названия команды указано только имя переменной, то это воспринимается командой как $x=0$:

$$e^x \text{ series, x, 5} \rightarrow 1 + x + \frac{1}{2} \cdot x^2 + \frac{1}{6} \cdot x^3 + \frac{1}{24} \cdot x^4$$

$$e^x \text{ series, x=0} \rightarrow 1 + x + \frac{1}{2} \cdot x^2 + \frac{1}{6} \cdot x^3 + \frac{1}{24} \cdot x^4 + \frac{1}{120} \cdot x^5$$

$$\sin(x) \text{ series, x, 6} \rightarrow x - \frac{1}{6} \cdot x^3 + \frac{1}{120} \cdot x^5$$

$$e^x \text{ series, x=1, 4} \rightarrow \exp(1) + \exp(1) \cdot (x-1) + \frac{1}{2} \cdot \exp(1) \cdot (x-1)^2 + \frac{1}{6} \cdot \exp(1) \cdot (x-1)^3$$

$$(e^x - x \cdot a) \text{ series, x, 4} \rightarrow 1 + (1-a) \cdot x + \frac{1}{2} \cdot x^2 + \frac{1}{6} \cdot x^3$$

В первый маркер после названия команды можно вводить через запятые несколько переменных (возможно, с указанием их значений) при разложении в ряд функции многих переменных в окрестности заданной точки:

$$e^{xy} \text{ series, x, y} \rightarrow 1 + x \cdot y + \frac{1}{2} \cdot x^2 \cdot y^2$$

$$e^{xy} \text{ series, x=1, y=0, 4} \rightarrow 1 + y + (x-1) \cdot y + \frac{1}{2} \cdot y^2 + (x-1) \cdot y^2 + \frac{1}{6} \cdot y^3$$

$$(x^2 \cdot y + y) \text{ series, x=1, y=1, 3} \rightarrow -2 + 2 \cdot y + 2 \cdot x + 2 \cdot (x-1) \cdot (y-1) + (x-1)^2$$

Пример 8.34 (Решение уравнений, неравенств и их систем).

$$(x^3 - 5 \cdot x^2 + 10 \cdot x - 6) = 0 \text{ solve, } x \rightarrow \begin{pmatrix} 1 \\ 2 + i \cdot \sqrt{2} \\ 2 - i \cdot \sqrt{2} \end{pmatrix}$$

Если надо найти решение уравнения вида $f(x)=0$, то в записи команды можно не указывать правую часть уравнения:

$$x^3 - 5 \cdot x^2 + 10 \cdot x - 6 \text{ solve, } x \rightarrow \begin{pmatrix} 1 \\ 2 + i \cdot \sqrt{2} \\ 2 - i \cdot \sqrt{2} \end{pmatrix}$$

Указав уравнение и условие $\text{Im}(x)=0$ как элементы матрицы-столбца, можно найти только вещественные корни уравнения:

$$\begin{pmatrix} x^3 - 5 \cdot x^2 + 10 \cdot x - 6 \\ \text{Im}(x) = 0 \end{pmatrix} \text{ solve, } x \rightarrow 1 \quad \left[\begin{pmatrix} x^3 - 5 \cdot x^2 + 10 \cdot x - 6 \\ \text{Im}(x) = 0 \end{pmatrix} = 0 \right] \text{ solve, } x \rightarrow 1$$

Решение системы уравнений можно оформить следующим образом:

$$\begin{pmatrix} x + y = 3 \\ x^3 + y^3 = 9 \end{pmatrix} \text{ solve, } x, y \rightarrow \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \quad \begin{pmatrix} x + y = 3 \\ x^3 + y^3 = 9 \end{pmatrix} \text{ solve, } \begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

Если среди решений системы есть комплексные, например:

$$\begin{pmatrix} \frac{x+y}{x-y} + \frac{x-y}{x+y} = 5.2 \\ x \cdot y = 6 \end{pmatrix} \text{ solve, } x, y \rightarrow \begin{pmatrix} 3 \cdot i & -2 \cdot i \\ -3 \cdot i & 2 \cdot i \\ 3 & 2 \\ -3 & -2 \end{pmatrix}$$

то выделить только вещественные пары можно так:

$$\begin{pmatrix} \frac{x+y}{x-y} + \frac{x-y}{x+y} = 5.2 \\ x \cdot y = 6 \\ \text{Im}(x) = 0 \\ \text{Im}(y) = 0 \end{pmatrix} \text{ solve, } \begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} 3 & 2 \\ -3 & -2 \end{pmatrix}$$

Можно решить уравнение при заданном условии, например:

$$\left[\begin{pmatrix} \frac{x^2 - 1}{(x - 1)^2} + \frac{x^3 - 8}{(x - 2)^2} = x \\ x > 1 \end{pmatrix} \right] \text{ solve, } x \rightarrow \frac{6}{5}$$

Пример решения неравенства:

$$\frac{2x^2 - 5x - 7}{x - 5} \geq 3 \text{ solve, } x \rightarrow \begin{pmatrix} (x \leq -2) \\ 5 < x \end{pmatrix}$$

Пример 8.35 (Символьные преобразования над функциями пользователя).

$$f(x) := \frac{x^2 + 2x - 3}{x \cdot (x^2 - 4x + 4)}$$

$$f(x) = 0 \text{ solve, } x \rightarrow \begin{pmatrix} -3 \\ 1 \end{pmatrix}$$

$$f(x) \text{ convert, parfrac, } x \rightarrow \frac{-3}{4 \cdot x} + \frac{5}{2 \cdot (x - 2)^2} + \frac{7}{4 \cdot (x - 2)}$$

$$f\left(\frac{1}{3}\right) \rightarrow \frac{-12}{5} \quad f\left(\frac{1}{3}\right) \text{ float, } 5 \rightarrow -2.4000$$

$$f(z) \text{ factor} \rightarrow (z + 3) \cdot \frac{(z - 1)}{z \cdot (z - 2)^2}$$

$$\frac{d}{dx} f(x) \rightarrow \frac{(2 \cdot x + 2)}{x \cdot (x^2 - 4 \cdot x + 4)} - \frac{(x^2 + 2 \cdot x - 3)}{x^2 \cdot (x^2 - 4 \cdot x + 4)} - \frac{(x^2 + 2 \cdot x - 3)}{x \cdot (x^2 - 4 \cdot x + 4)^2} \cdot (2 \cdot x - 4)$$

$$\frac{d}{dx} f(x) \text{ simplify} \rightarrow \frac{-(x^3 + 6 \cdot x^2 - 9 \cdot x + 6)}{(x - 2) \cdot (x^2 - 4 \cdot x + 4) \cdot x^2}$$

$$\frac{d}{dx} f(x) \left| \begin{array}{l} \text{solve, } x \\ \text{float, } 3 \end{array} \right. \rightarrow \begin{pmatrix} -7.34 \\ .67 - .606 \cdot i \\ .67 + .606 \cdot i \end{pmatrix}$$

Лекция № 6

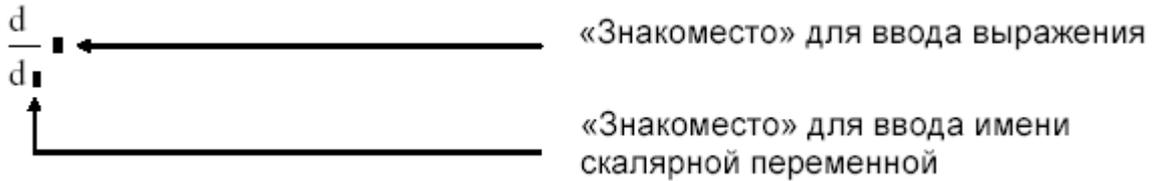
Mathcad: Дифференцирование и интегрирование. Решение уравнений и их систем

1. Дифференцирование и интегрирование
 2. Решение уравнений и их систем
 3. Решение дифференциальных уравнений и их систем
1. Дифференцирование и интегрирование

Интегрирование и дифференцирование функций в пакете Mathcad осуществляется с использованием панели **Calculus (Калькуляция)** на панели инструментов **Math (Математика)**.

Дифференцирование. Для вычисления производной первого порядка служит оператор дифференцирования, ввод которого можно выполнить нажатием

кнопки  панели **Calculus** или комбинацией двух клавиш **Shift** и **?**. Шаблон оператора имеет вид:



Например, найдем производную функции $y=x^2$ в точке $x=2$:

$$y(x) := x^2 \quad x := 2 \quad \frac{d}{dx} y(x) = 4$$

Чтобы вычислить производную порядка выше первого, можно последовательно применить несколько раз оператор дифференцирования. Например,

$$y(x) := x^3 + 2x \quad x := 2 \quad \frac{d}{dx} \frac{d}{dx} y(x) = 12 \quad \frac{d}{dx} \frac{d}{dx} \frac{d}{dx} y(x) = 6$$

Однако лучше воспользоваться оператором дифференцирования n -го порядка ($0 \leq n \leq 5$), вводимого нажатием кнопки  панели инструментов **Calculus** или комбинацией клавиш **Ctrl+Shift+?**. Шаблон оператора имеет вид:



Заметим, что при заполнении любого из двух знакомест для ввода порядка производной автоматически заполняется второе знакоместо. Теперь предыдущие вычисления можно оформить так:

$$y(x) := x^3 + 2x \quad x := 2 \quad \frac{d^2}{dx^2} y(x) = 12 \quad \frac{d^3}{dx^3} y(x) = 6$$

Однако если требуется вычислить производную порядка выше 5, то придется последовательно применить несколько раз оператор дифференцирования n -го порядка.

Например, найдем производную 7-го порядка функции $e^x \cos x$ в точке $x=0.1\pi$:

$$x := 0.1\pi \quad \frac{d^5}{dx^5} \frac{d^2}{dx^2} (e^x \cos(x)) = 13.801$$

Интегрирование. В Mathcad встроена также возможность вычисления определенных интегралов. Ввод оператора определенного интеграла выполняется

нажатием кнопки  панели инструментов **Calculus** или комбинацией клавиш **Shift+7** (символ **&**). Шаблон оператора имеет вид:



2. Решение уравнений и их систем

Численное решение уравнения. Многие уравнения, например трансцендентные, не имеют аналитических решений. Однако корни таких уравнений могут быть найдены численными методами с некоторой заранее заданной погрешностью. В Mathcad для этой цели запрограммирован *метод хорд*. Для поиска корней уравнений вида $y(x)=0$ служит встроенная функция **root**. Она возвращает найденное с заданной точностью значение неизвестной x , обращающее функцию $y(x)$ в 0 (*корень функции*). Функция **root** имеет следующий формат:

root(y(x),x)

Таким образом, **root** — это функция двух аргументов. Перед обращением к ней, как и в случае любой другой функции, ее аргументы должны быть уже определены. Первый аргумент **y(x)** — это сама функция, корень которой мы ищем, а **x** — скалярная переменная, относительно которой ищется корень (или решается уравнение). Переменной **x** перед обращением к функции должно быть присвоено некоторое значение — приближенное значение корня, которое можно определить графически (причем для поиска комплексного корня должно быть задано приближенное комплексное значение корня).

Чтобы получить искомый корень, необходимо, чтобы между приближенным и точным значениями корня не лежали точки перегиба функции.

Пример 4.1. Решить уравнение $x + \sin(x) = 2$.

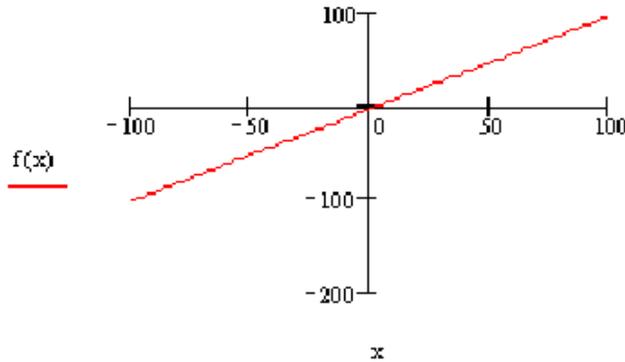
Решение:

1. Определим функцию, корни которой мы ищем:

$$f(x) := x + \sin(x) - 2$$

2. Построим предварительный (на достаточно большом промежутке) график функции, чтобы определить наличие и расположение корней:

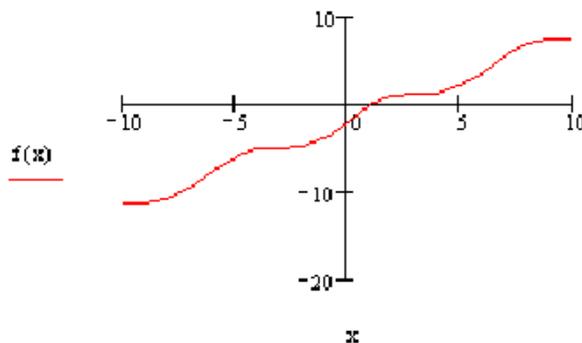
$$x := -100..100$$



Из графика видно, что корень (или корни) функции лежат в окрестности точки $x=0$. Этот же факт можно было установить, проанализировав поведение функции на бесконечности.

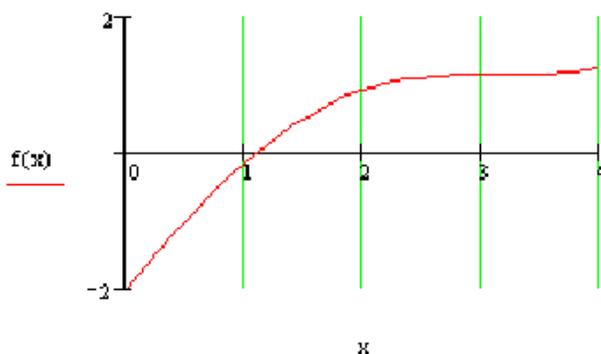
3. Уменьшим промежуток задания переменной x :

$$x := -10..10$$



Теперь видно, что корень только один, и мы можем уточнить его расположение следующим образом:

$$x := 0,0.1..4$$



Из графика видно, что корень лежит внутри промежутка $[1; 2]$. Примем приближенное значение $x=1.5$ и определим корень с большей точностью:

$$x := 1.5 \quad \text{root}(f(x), x) = 1.106$$

Если требуется определить корень с еще большей точностью, то надо переопределить значение встроенной переменной **TOL** задающей точность вычислений, а затем снова воспользоваться функцией **root**. При этом потребуются также изменить значение формата результата в диалоговом окне команды **Format** → **Result** (**Формат** → **Результат**).

$$\text{TOL} := 0.0001 \quad \text{root}(f(x), x) = 1.10606$$

$$\text{TOL} := 0.001$$

Вернитесь к точности, установленной по умолчанию:

Начиная с версии Mathcad 8, допустим еще один способ обращения к функции **root**, когда не требуется задавать начальное приближение для корня. Функция в этом случае имеет следующий формат:

$$\text{root}(y(x), x, a, b)$$

Здесь a и b — начало и конец интервала, на котором находится корень. Причем значения $f(a)$ и $f(b)$ должны иметь разный знак. Решение уравнения из примера можно оформить следующим образом:

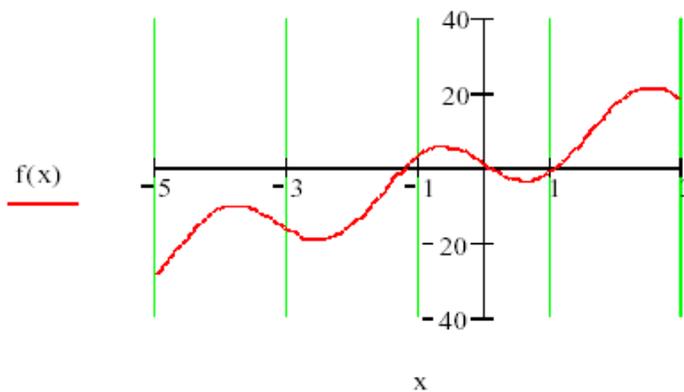
$$\text{root}(f(x), x, -10, 10) = 1.106$$

Если функция имеет несколько корней, то можно определить новую функцию, например $r(x)$, обращаясь к которой, можно найти корень, задавая его начальное приближение в качестве аргумента.

Пример 4.2. Найдите все корни уравнения $5x - 8\sin(2x) + 1 = 0$.

Решение: Определив функцию $f(x)$, построим ее график:

$$f(x) := 5x - 8\sin(2x) + 1 \quad x := -5, -4.9..3$$



Из графика видно, что функция имеет три корня, поэтому удобнее определить новую функцию $r(x)$:

$$r(x) := \text{root}(f(x), x)$$

Теперь найдем значения функции $r(x)$ для трех начальных приближений x_1 , x_2 и x_3 , определенных по графику:

$$\begin{aligned} x_1 &:= -1 & x_2 &:= 0 & x_3 &:= 1 \\ r(x_1) &= -1.224 & r(x_2) &= 0.092 & r(x_3) &= 1.098 \end{aligned}$$

Нахождение корней полинома. Для нахождения корней выражения, имеющего вид:

$$v_n x^n + \dots + v_2 x^2 + v_1 x + v_0,$$

Информационные технологии в менеджменте

лучше использовать функцию **polyroots**, нежели **root**. В отличие от функции **root**, функция **polyroots** не требует начального приближения. Кроме того, функция **polyroots** возвращает сразу все корни, как вещественные, так и комплексные.

Функция **polyroots** имеет следующий формат:

polyroots(v),

где $v = \begin{pmatrix} v_0 \\ v_1 \\ \dots \\ v_n \end{pmatrix}$ — вектор-столбец коэффициентов полинома, перечисленных в

направлении увеличения степени неизвестной, начиная с нулевой.

Функция **polyroots** всегда возвращает значения корней полинома, найденные численно. Чтобы найти корни аналитически (символьно), можно использовать функцию **Solve** на панели **Math** → **Symbolics** (**Математика** → **Символика**).

Численное решение системы уравнений. Для решения систем уравнений и неравенств (можно решить и одно уравнение/неравенство) используется конструкция, которая называется «**решающий блок**». Как и функция **root**, решающий блок использует для поиска корней итерационные методы, поэтому необходимо задавать начальные приближения для значений переменных, относительно которых решается система.

Решающий блок состоит из нескольких компонент, следующих на листе в строго определенном порядке:

1. Начальное приближение (присваивание начальных значений искомым переменным).

2. Директива **Given**, которую необходимо набрать с клавиатуры.

3. Уравнения, которые необходимо решить. Уравнения вводятся в обычной математической форме, но вместо простого знака равенства используется оператор логического равенства (вводится путем нажатия Ctrl+= или с панели инструментов **Math** → **Boolean** (**Математика** → **Булево**)).

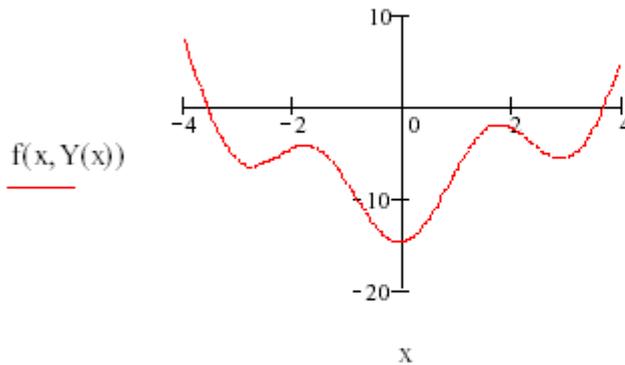
4. Обращение к функции **Find**. Аргументами функции являются имена переменных, относительно которых решается система. Функция **Find** возвращает вектор-столбец значений, где первый элемент соответствует первой переменной в списке аргументов, второй элемент — второй переменной и т. д.

Пример 4.4. Найти решение системы уравнений:

$$\begin{cases} x^2 + y^2 = 15, \\ y = 3 \sin x + 0.2 \end{cases}$$

Решение:

$$Y(x) := 3 \sin(x) + 0.2 \quad f(x, y) := x^2 + y^2 - 15 \quad x := -4, -3.9..4$$



$$\begin{aligned} &x := -4 \quad y := Y(x) \\ \text{Given} \quad &x^2 + y^2 - 15 = 0 \quad y = 3 \sin(x) + 0.2 \quad \text{Find}(x, y) = \begin{pmatrix} -3.581 \\ 1.476 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} &x := 4 \quad y := Y(x) \\ \text{Given} \quad &x^2 + y^2 - 15 = 0 \quad y = 3 \sin(x) + 0.2 \quad \text{Find}(x, y) = \begin{pmatrix} 3.656 \\ -1.277 \end{pmatrix} \end{aligned}$$

В Mathcad существует возможность *одновременно найти несколько решений*. Для этого система уравнений и начальные приближения должны быть переписаны в векторной форме (рис. 2). Каждая переменная будет вектором, содержащим столько компонент, сколько решений находится. Допустим, что в уравнении присутствует выражение $x \cdot x$. Если $x = (x_1; x_2)$ — вектор, то $x \cdot x = x_1 \cdot x_1 + x_2 \cdot x_2$. Нам же необходим результат поэлементного перемножения $(x_1 \cdot x_1; x_2 \cdot x_2)$. Для этого существует специальная операция, записываемая как $\overline{(x \cdot x)}$ (знак $\overrightarrow{f(M)}$ на панели инструментов **Math** → **Matrix (Математика)** → **Матрица**)).

$$\begin{aligned} &x := \begin{pmatrix} 0 \\ 2 \end{pmatrix} \quad y := \begin{pmatrix} 0 \\ 2 \end{pmatrix} \\ \text{Given} \\ &\overrightarrow{(x \cdot x)} - x = 2 \\ &x - 0.5y = 0 \\ &\begin{pmatrix} X \\ Y \end{pmatrix} := \text{Find}(x, y) \\ &X = \begin{pmatrix} -1 \\ 2 \end{pmatrix} \quad Y = \begin{pmatrix} -2 \\ 4 \end{pmatrix} \end{aligned}$$

Рис. 2. Пример одновременного поиска нескольких решений

Изменения коснулись и части получения результата. В данном случае функция **Find** вернет вектор из двух элементов, которые мы обозначили как X и Y . Каждый из этих элементов есть вектор значений x или y для решений. Соответственно первое решение — $(-1; -2)$; второе решение — $(2; 4)$.

Аналитическое решение уравнений и их систем. Аналитическое решение уравнений и их систем может быть получено с помощью функции **Solve** на панели **Math** → **Symbolics (Математика)** → **Символика**).

Пример 1. Аналитическое решение уравнения:

$$a \cdot x = x^2 + b \text{ solve, } x \rightarrow \begin{bmatrix} \frac{1}{2} \cdot a + \frac{1}{2} \cdot (a^2 - 4 \cdot b)^{\frac{1}{2}} \\ \frac{1}{2} \cdot a - \frac{1}{2} \cdot (a^2 - 4 \cdot b)^{\frac{1}{2}} \end{bmatrix}$$

Пример 2. Аналитическое решение системы уравнений:

$$\begin{cases} a \cdot y = x + b \\ c \cdot x = y - d \end{cases} \text{ solve, } x, y \rightarrow \begin{bmatrix} \frac{-(d \cdot a - b)}{(c \cdot a - 1)} & \frac{(c \cdot b - d)}{(c \cdot a - 1)} \end{bmatrix}$$

Для получения аналитического решения можно также применить **решающий блок Given/Find**. При этом уже нет необходимости указывать начальные приближения, а после функции **Find** используется оператор аналитического вычисления «→» вместо оператора числового вычисления «=».

Пример 3.

$$\begin{array}{l} \text{Given} \\ \\ y = \cos(\omega \cdot x + t) \\ \\ y = \sin(\omega \cdot x - t) \\ \\ \text{Find}(x, y) \rightarrow \begin{pmatrix} \frac{1}{4} \cdot \frac{\pi}{\omega} \\ \cos\left(\frac{1}{4} \cdot \pi + t\right) \end{pmatrix} \end{array}$$

3. Решение дифференциальных уравнений и их систем

Вычислительный процессор Mathcad может работать с дифференциальными уравнениями, представленными только в форме Коши, т. е. в виде:

$$y'(t) = f(y(t), t).$$

Задача Коши — одна из основных задач теории дифференциальных уравнений (обыкновенных и с частными производными); состоит в нахождении решения (интеграла) дифференциального уравнения, удовлетворяющего начальным условиям.

Задача Коши обычно возникает при анализе процессов, определяемых дифференциальным законом эволюции и начальным состоянием (математическим выражением которых и являются уравнение и начальное условие). Этим мотивируется терминология и выбор обозначений: начальные данные задаются при $t = 0$, а решение отыскивается при $t > 0$.

От краевых задач задача Коши отличается тем, что область, в которой должно быть определено искомое решение, здесь заранее не указывается. Тем не менее, задачу Коши можно рассматривать как одну из краевых задач.

Информационные технологии в менеджменте

Основные вопросы, которые связаны с задачей Коши, таковы:

1. Существует ли (хотя бы локально) решение задачи Коши?
2. Если решение существует, то какова область его существования?
3. Является ли решение единственным?
4. Если решение единственно, то будет ли оно корректным, то есть непрерывным (в каком-либо смысле) относительно начальных данных?

У пользователя имеется выбор — либо использовать вычислительный блок **Given/Odesolve**, либо встроенные функции.

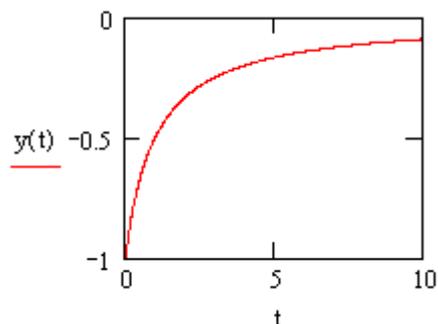
Для решения дифференциальных уравнений при помощи блока Given/Odesolve:

- ввести ключевое слово **Given**;
- ввести дифференциальное уравнение в форме Коши, записанное с помощью логических операторов;
- ввести начальное условие в форме $y(t_0) = b$;
- решить уравнение с помощью функции **Odesolve(t, t₁, step)**, где t — переменная дифференцирования в дифференциальном уравнении; t_1 — правая граница интервала, на котором производится поиск решения; step — необязательный параметр, определяющий количество шагов метода Рунге-Кутты, который применяется при решении. Чем больше step, тем больше точность решения, тем больше времени будет затрачено на его поиск. По умолчанию в методе применяется фиксированный шаг (Fixed). Для замены его на адаптивный (Adaptive) необходимо вызвать контекстное меню над функцией **Odesolve**.

При использовании блока **Given/Odesolve** для решения **дифференциального уравнения высшего порядка** необходимо кроме начального значения самой функции, задать начальные значения для всех ее производных, входящих в уравнение, причем производные должны быть записаны при помощи обычного обозначения производной (один, два и т. д. штриха справа от имени функции, Ctrl+F7).

Пример 1. Решение дифференциального уравнения первого порядка $y' = y^2$ при помощи блока **Given/Odesolve**:

```
Given
d
—
dt y(t) = y(t)2
y(0) = -1
y := Odesolve(t,10)
```



Пример 2. Решение дифференциального уравнения 3-го порядка при помощи блока **Given/Odesolve**:

$$y'''(t) + 4y''(t) + 3y'(t) + y(t) = 0;$$

$$y(0) = 50, y'(0) = 10, y''(0) = 0.$$

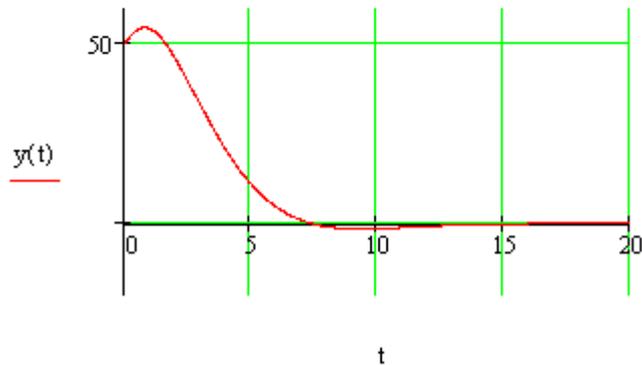
Информационные технологии в менеджменте

Given

$$\frac{d^3}{dt^3}y(t) + 4 \cdot \frac{d^2}{dt^2}y(t) + 3 \cdot \frac{d^1}{dt}y(t) + y(t) = 0$$

$$y(0) = 50 \quad y'(0) = 10 \quad y''(0) = 0$$

`y := odesolve(t,20)`



Для решения дифференциального уравнения или системы дифференциальных уравнений при помощи встроенных функций необходимо:

- переменной y_0 присвоить начальное значение либо вектор начальных значений в случае решения системы;

- функции $D(t, y)$ присвоить функцию из правой части уравнения в форме Коши (либо вектор-функцию при решении системы);

- решить уравнение (или систему) при помощи одной из функций:

rkfixed (y_0, t_0, t_1, M, D) — метод Рунге-Кутты с фиксированным шагом,

rkadapt (y_0, t_0, t_1, M, D) — метод Рунге-Кутты с переменным шагом,

bulstoer (y_0, t_0, t_1, M, D) — метод Булирша-Штера,

где y_0 — вектор начальных значений в точке t_0 ; t_0 — начальная точка расчета; t_1 — конечная точка расчета; M — количество точек интегрирования (число шагов); D — векторная функция размера $n \times 1$ (где n — порядок решаемого дифференциального уравнения или количество уравнений 1-го порядка в системе уравнений) двух аргументов — скалярного t и векторного y (для одного дифференциального уравнения — просто функция).

Результатом будет матрица, 0-й столбец которой представляет собой значения аргумента t , 1-ый — соответствующие значения искомой функции; 2-ой — значения первой производной искомой функции и т.д. до $n - 1$ производной. Количество строк матрицы равно числу шагов M .

Пример 3. Решение дифференциального уравнения первого порядка $y' = y^2$ при помощи встроенной функции **rkfixed**:

Информационные технологии в менеджменте

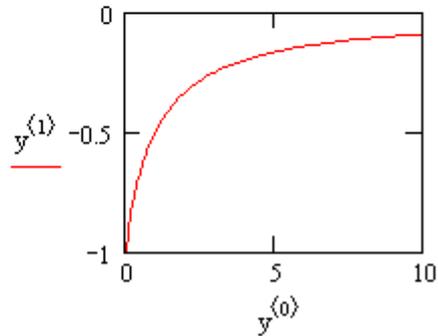
$$y_0 := -1$$

$$D(t,y) := y^2$$

$$M := 100$$

$$y := \text{rkfixed}(y_0, 0, 10, M, D)$$

	0	1
0	0	-1
1	0.1	-0.909
2	0.2	-0.833
3	0.3	-0.769
4	0.4	-0.714
5	0.5	-0.667
6	0.6	-0.625



Пример 4. Решение дифференциального уравнения 3-го порядка при помощи встроенной функции **rkfixed**:

$$y'''(t) + 4y''(t) + 3y'(t) + y(t) = 0;$$

$$y(0) = 50, y'(0) = 10, y''(0) = 0.$$

При этом необходимо преобразовать дифференциальное уравнение третьего порядка в систему трех дифференциальных уравнений первого порядка (переход к переменным состояния). Для этого введем так называемые *переменные состояния* y_0, y_1, y_2, y_3 :

$$y_0 = y, y_1 = y', y_2 = y'', y_3 = y''''.$$

Тогда можно записать:

$$\begin{cases} y_0' = y_1 = y_1 \\ y_1' = y_2 = y_2 \\ y_2' = y_3 = y_3 = -4y_2 - 3y_1 - y_0 \end{cases}$$

Информационные технологии в менеджменте

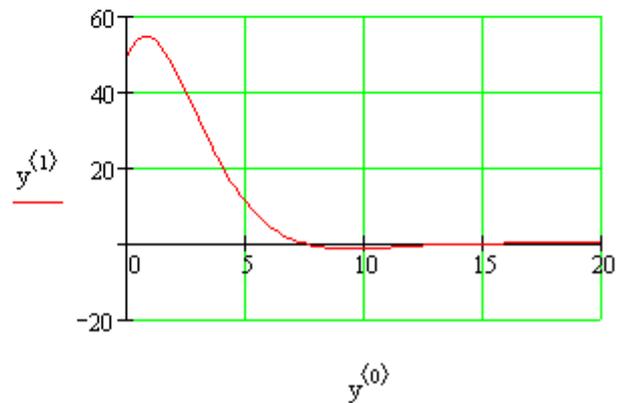
$$y^0 := \begin{pmatrix} 50 \\ 10 \\ 0 \end{pmatrix}$$

$$D(t, y) := \begin{pmatrix} y_1 \\ y_2 \\ -y_0 - 3 \cdot y_1 - 4 \cdot y_2 \end{pmatrix}$$

$$M := 100$$

$$y := \text{rkfixed}(y^0, 0, 20, M, D)$$

	0	1	2	3
0	0	50	10	0
1	0.2	51.914	8.747	-10.923
2	0.4	53.41	6.053	-15.177
3	0.6	54.308	2.892	-16.009
4	0.8	54.57	-0.244	-15.149
5	1	54.229	-3.119	-13.514
6	1.2	53.347	-5.632	-11.585
7	1.4	52.002	-7.751	-9.614
8	1.6	50.273	-9.484	-7.729



Пример 5. Решение системы двух дифференциальных уравнений при помощи встроенной функции **rkfixed**:

$$\begin{cases} \frac{dy_0}{dt} = y_1; \\ \frac{dy_1}{dt} = -y_0 - 0,1y_1; \end{cases}$$

$$y_0(0) = 0,1, \quad y_1(0) = 0.$$

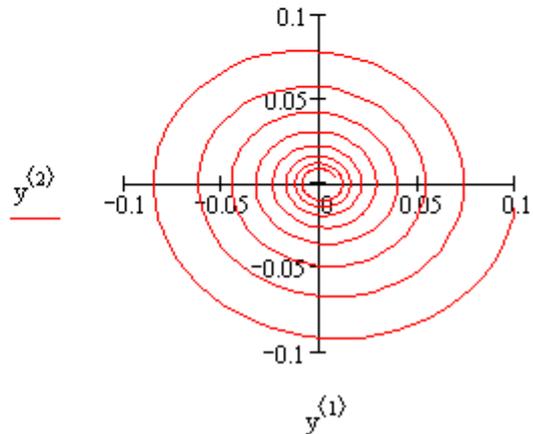
$$D(t, y) := \begin{pmatrix} y_1 \\ -y_0 - 0.1 \cdot y_1 \end{pmatrix}$$

$$y_0 := \begin{pmatrix} 0.1 \\ 0 \end{pmatrix}$$

$$M := 500$$

$$y := \text{rkfixed}(y_0, 0, 50, M, D)$$

	0	1	2
0	0	0.1	0
1	0.1	0.1	$-9.934 \cdot 10^{-3}$
2	0.2	0.098	-0.02
3	0.3	0.096	-0.029
4	0.4	0.092	-0.038
5	0.5	0.088	-0.047
6	0.6	0.083	-0.055
7	0.7	0.077	-0.062



Лекция № 7

Mathcad: Элементы программирования

На одном листе Mathcad могут определяться один или несколько программных блоков. Обычно их используют при разработке функций, которые осуществляют какую-либо сложную обработку данных.

В программном блоке можно читать значения переменных, определенных в Mathcad до этого блока. Однако изменить значения этих переменных внутри программного блока невозможно. Все переменные, которым присваиваются значения внутри программного блока, будут локальными переменными, которые недоступны вне блока.

Программный блок представляет собой группу операторов присваивания и управляющих операторов. Необходимо обратить особое внимание, что все ключевые слова (например, **if**) в этих операторах обязательно вводятся с помощью панели **Programming (Программирование)** (рис. 1) включаемой кнопкой  на панели **Math**. Их ввод с клавиатуры — ошибка.

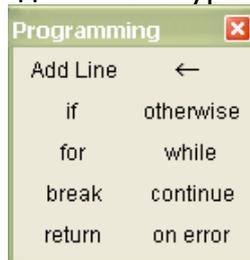


Рис. 1.

В табл. 1 приведены управляющие операторы панели **Programming**.

Таблица 1

Управляющие операторы панели Programming

Оператор Mathcad	Комментарий
Add line	Команда вычерчивает вертикальную линию, ограничивающую область программы.
$A \leftarrow B$	Присваивание переменной A значения B .
или <pre> оператор1 if условие if условие оператор1 оператор2 .. </pre>	Условный оператор (оператор ветвления) if . <i>Условием</i> может быть любое логическое выражение (логические операторы находятся на панели Boolean). <i>Условие</i> должно стоять после if , а оператор, который выполняется, если выполнено заданное условие, — перед if .
Пример: <pre> оператор1 if условие оператор2 otherwise f(x) := -x if x < 0 x otherwise f(-3) = 3 f(4) = 4 </pre>	Полный оператор ветвления. Otherwise — оператор, который должен быть исполнен, если условие оператора if не выполняется.
Пример: <pre> for инд ∈ нач..кон оператор Sum(n) := s ← 0 for i ∈ 1..n s ← s + i Sum(4) = 10 </pre>	Оператор цикла с параметром for . За ключевым словом for следует переменная-счетчик, которая принимает значения от начального до конечного с шагом, равным единице. Внутренние операторы цикла сдвинуты немного вправо.
Пример: <pre> while условие оператор N(x, f, f_x) := while f(x) > 10⁻⁶ x ← x - f(x)/f_x(x) N(2, sin, cos) = 3.142 </pre>	Оператор цикла с предусловием while . Внутренние операторы цикла будут исполняться до тех пор, пока будет истинным <i>условие</i> , следующее за ключевым словом while .
Пример: <pre> continue if x ≥ 10 </pre>	Оператор continue служит для преждевременного завершения текущей итерации цикла; сам цикл при этом продолжается.
Пример: <pre> break if i ≥ 10 </pre>	Оператор break служит для преждевременного завершения цикла, чтобы, например, избежать зацикливания или слишком продолжительных вычислений.
Пример: <pre> return y </pre>	Оператор преждевременного завершения программы; указанное в ячейке значение будет возвращено.
Пример: <pre> expr1 on error expr2 </pre>	Если при вычислении выражения $expr2$ возникла ошибка, вычисляется выражение $expr1$.

Информационные технологии в менеджменте

Пример 1. Функция, возвращающая -1 , 0 или 1 в зависимости от знака аргумента:

$$\text{Sign}(x) := \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$

$$\text{Sign}(2) = 1$$

Пример 2.

$$f(x) := \begin{cases} \text{"negative"} & \text{if } x < 0 \\ \text{"positive"} & \text{if } x > 0 \\ \text{"zero"} & \text{otherwise} \end{cases}$$

$$f(-1) = \text{"negative"}$$

$$f(1) = \text{"positive"}$$

$$f(0) = \text{"zero"}$$

Пример 3. Оператор цикла **for** с ранжированной переменной:

$$x := \begin{cases} z \leftarrow 0 \\ \text{for } i \in 0..5 \\ z \leftarrow z + i \end{cases}$$

$$x = 15$$

Пример 4. Оператор цикла **for** с вектором:

$$x := \begin{cases} z \leftarrow 0 \\ \text{for } i \in (1 \ 2 \ 3) \\ z \leftarrow z + i \end{cases}$$

$$x = 6$$

Пример 5. Оператор цикла **while**:

$$x := \begin{cases} z \leftarrow 0 \\ \text{while } z < 10 \\ z \leftarrow z + 1 \end{cases}$$

$$x = 10$$

Пример 6. Оператор **break** внутри цикла **for**:

$$x := \begin{cases} z \leftarrow 0 \\ \text{for } i \in 0..5 \\ \quad z \leftarrow z + i \\ \quad \text{break if } i = 2 \end{cases}$$

$$x = 3$$

Пример 7. Оператор **break** внутри цикла **while**:

$$x := \begin{cases} z \leftarrow 0 \\ \text{while } z < 10 \\ \quad \begin{cases} z \leftarrow z + 1 \\ \text{break if } z > 5 \end{cases} \end{cases}$$

$x = 6$

Пример 8. Оператор **continue** в конце цикла **while**:

$$x := \begin{cases} z \leftarrow 0 \\ \text{while } z < 10 \\ \quad \begin{cases} z \leftarrow z + 1 \\ \text{continue} \end{cases} \end{cases}$$

$x = 10$

Пример 9. Оператор **continue** в конце цикла **for**:

$$x := \begin{cases} z \leftarrow 0 \\ \text{for } i \in (1 \ 2 \ 3) \\ \quad \begin{cases} z \leftarrow z + i \\ \text{continue} \end{cases} \end{cases}$$

$x = 6$

Пример 10. Возврат значения обозначен явно в последней строке программы:

$$f(x) := \begin{cases} y \leftarrow x^2 \\ z \leftarrow y + 1 \\ z \end{cases}$$

$f(2) = 5$

Вместе с тем, можно прервать выполнение программы в любой ее точке (например, с помощью условного оператора) и выдать некоторое значение, применив оператор **return**. В этом случае при выполнении указанного условия (пример 11) значение, введенное в местозаполнитель после **return**, возвращается в качестве результата, а никакой другой код больше не выполняется.

Пример 11. Применение оператора **return**:

$$f(x) := \begin{cases} z \leftarrow x^2 \\ \text{return "zero" if } x = 0 \\ z \end{cases}$$

$f(-1) = 1$ $f(0) = \text{"zero"}$

$f(2) = 4$ $f(6) = 36$

Программирование в MathCAD позволяет осуществлять дополнительную **обработку ошибок**. Если пользователь предполагает, что выполнение кода в каком-либо месте программного модуля способно вызвать ошибку (например, деление на ноль), то эту ошибку можно перехватить с помощью оператора **on error**.

Информационные технологии в менеджменте

Пример 12. Перехват ошибки деления на ноль:

$$f(n) := \begin{cases} z \leftarrow n \\ \text{"user error: can't divide by zero"} \quad \text{on error } \frac{1}{z} \end{cases}$$

$$f(-2) = -0.5$$

$$f(0) = \text{"user error: can't divide by zero"}$$

Оператор перехвата ошибок удобно применять в комбинации со встроенной функцией **error(S)**. Она приводит к генерации ошибки в обычной для MathCAD форме с сообщением **S**. Пример усовершенствования листинга [примера 12](#) для такого стиля обработки ошибки деления на ноль:

$$f(n) := \begin{cases} z \leftarrow n \\ \text{error("user error: can't divide by zero")} \quad \text{on error } \frac{1}{z} \end{cases}$$

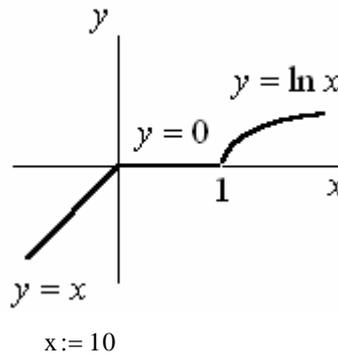
$$f(-2) = -0.5$$

$$f(0) = \blacksquare$$



Обратите внимание, что сделанные изменения свелись к помещению текста сообщения об ошибке в аргумент функции **error**.

Пример 13. Напишите программу для вычисления значения функции, заданной графически, с использованием операторов ветвления **if**.



$$y := \begin{cases} y \leftarrow x & \text{if } x < 0 \\ y \leftarrow 0 & \text{if } 0 \leq x \leq 1 \\ y \leftarrow \ln(x) & \text{if } x > 1 \\ y \end{cases}$$

$$y = 2.303$$

Информационные технологии в менеджменте

Пример 14. Для функции, заданной графически, вычислите значения функции y для всех значений аргумента x на заданном интервале $[x_H, x_K]$ с заданным шагом d_x .

```


$$\begin{pmatrix} x \\ y \end{pmatrix} := \begin{array}{l} dx \leftarrow 0.05 \\ x_n \leftarrow -2 \\ x_k \leftarrow 2 \\ i \leftarrow 0 \\ \text{for } xx \in x_n, x_n + dx.. x_k \\ \quad \begin{array}{l} y_i \leftarrow xx \text{ if } xx < 0 \\ y_i \leftarrow 0 \text{ if } 0 \leq xx \leq 1 \\ y_i \leftarrow \ln(xx) \text{ if } xx > 1 \\ x_i \leftarrow xx \\ i \leftarrow i + 1 \end{array} \\ \end{array} \begin{pmatrix} x \\ y \end{pmatrix}$$


```

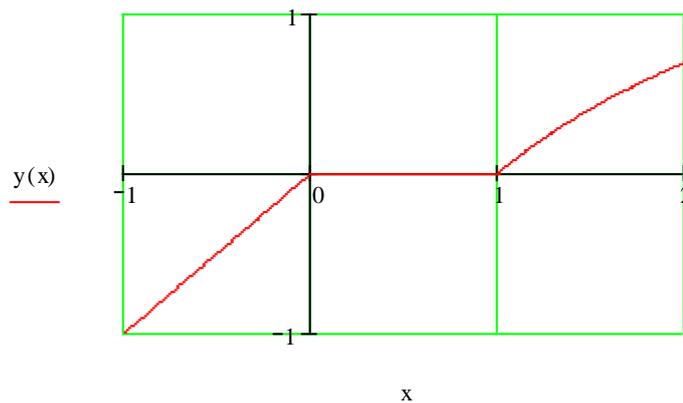
Или более просто:

```
x := -1, -0.99..2
```

```

y(x) := 
$$\begin{array}{l} y \leftarrow x \text{ if } x < 0 \\ y \leftarrow 0 \text{ if } 0 \leq x \leq 1 \\ y \leftarrow \ln(x) \text{ if } x > 1 \\ y \end{array}$$


```



Лекция № 8

Mathcad: Функции для решения задач интерполяции и аппроксимации

1. **Функции для решения задач интерполяции**
2. **Функции для решения задач аппроксимации**

1. Функции для решения задач интерполяции**Одномерная интерполяция**

Интерполяция является одним из распространенных методов обработки массивов данных, заданных своими узловыми значениями в табличной форме, и предполагает нахождение значений исследуемых функций между узловыми точками.

В Mathcad применяются два вида интерполяции:

— линейная — соседние точки соединяются отрезками прямых (линейным сплайном);

— нелинейная сплайн-интерполяция — соседние точки соединяются отрезками кубического сплайна (от [англ.](#) spline — планка, рейка).

Суть любого метода интерполяции заключается в разбиении отрезка, на котором исследуется функция на n отрезков. На каждом из отрезков вычисляются значения функции и строится кривая, соединяющая две точки на отрезке.

В случае линейной интерполяции такой кривой является прямая, а в случае кубической — парабола.

Линейная интерполяция позволяет построить «грубую» интерполяцию, а кубическая более точную, т. к. кубическая интерполяция не только содержит в себе формулу линейной интерполяции $P(x) = d + cx$, но к ней добавляется уточнение $P(x) = d + cx + bx^2 + ax^3$.

Линейная интерполяция. Пусть вектор X содержит координаты абсцисс заданных точек (абсциссы должны идти по возрастанию), а вектор Y — координаты ординат точек. При этом векторы X и Y должны быть одинаковой размерности.

Чтобы предсказать значение функции, например, в точках 2,5 и 1,2 достаточно набрать:

Linterp(X,Y,2.5)=

Аналогично:

Linterp(X,Y,1.2)=

Можно также построить график интерполяционной кривой (в случае линейной интерполяции это ломаная).

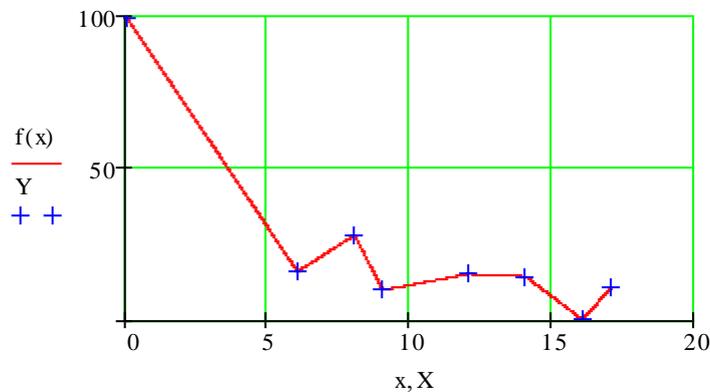
Пример 1. Реализация линейной интерполяции:

Информационные технологии в менеджменте

$$X := \begin{pmatrix} 0 \\ 6 \\ 8 \\ 9 \\ 12 \\ 14 \\ 16 \\ 17 \end{pmatrix} \quad Y := \begin{pmatrix} 100 \\ 16 \\ 28 \\ 9.9 \\ 15.2 \\ 14.4 \\ 0.16 \\ 10.7 \end{pmatrix}$$

$$x := X_0, X_0 + 0.01.. X_{\text{last}(X)} \quad X_{\text{last}(X)} = 17$$

$$f(x) := \text{linterp}(X, Y, x)$$



Кубическая сплайн-интерполяция. Кубическая интерполяция проводит кривую через заданные точки таким образом, чтобы первые и вторые производные кривой были непрерывны в каждой точке. Это обеспечивает гладкость интерполяционной кривой.

Для решения задач одномерной кубической интерполяции предусмотрены функции **lspline**, **pspline**, **cspline**, **interp**. При этом функции **lspline**, **pspline**, **cspline** имеют вспомогательное значение и осуществляют предварительную подготовку к решению задачи в соответствии с выбранным методом интерполяции данных, а **interp** использует эти предварительно полученные результаты для определения значений функции во всем диапазоне изменения аргумента.

Функции **lspline**, **pspline**, **cspline** возвращают вектор D , размерностью $(n + 3) \times 1$, который в качестве первых трех элементов содержит служебные параметры, определяющие особенности функций **lspline**, **pspline**, **cspline**. Остальными n элементами вектора D служат вторые производные в узловых точках.

Рассмотрим функции более подробно:

lspline(X,Y) — вычисляет вектор D с линейным приближением в конечных точках;

pspline(X,Y) — вычисляет вектор D с параболическим приближением в конечных точках;

cspline(X,Y) — вычисляет вектор D с полиномиально-кубическим приближением в конечных точках;

Информационные технологии в менеджменте

interp(D,X,Y,x) — вычисляет значения функции $f(x)$ с использованием кубических сплайнов для произвольного значения аргумента x по определенному одной из перечисленных выше функций вектору D .

Векторы X и Y размерности $n \times 1$ задают совокупность исходных узловых точек. Вектор X должен быть ранжирован в порядке возрастания значений его элементов.

Особенностью функций **lspline**, **pspline**, **cspline** служит то, что состыковка сплайнов осуществляется в узловых точках; между узлами функция $f(x)$ вычисляется по кубическому полиному.

Пример 2. Реализация кубической интерполяции:

$Dl := lspline(X, Y)$

$Dp := pspline(X, Y)$

$Dc := cspline(X, Y)$

$Dl =$

	0
0	0
1	3
2	0
3	0
4	11.427
5	-31.413
6	21.026
7	-5.865
8	-8.715
9	20.565
10	0

$Dp =$

	0
0	0
1	3
2	1
3	8.212
4	8.212
5	-30.328
6	20.943
7	-6.006
8	-7.886
9	17.39
10	17.39

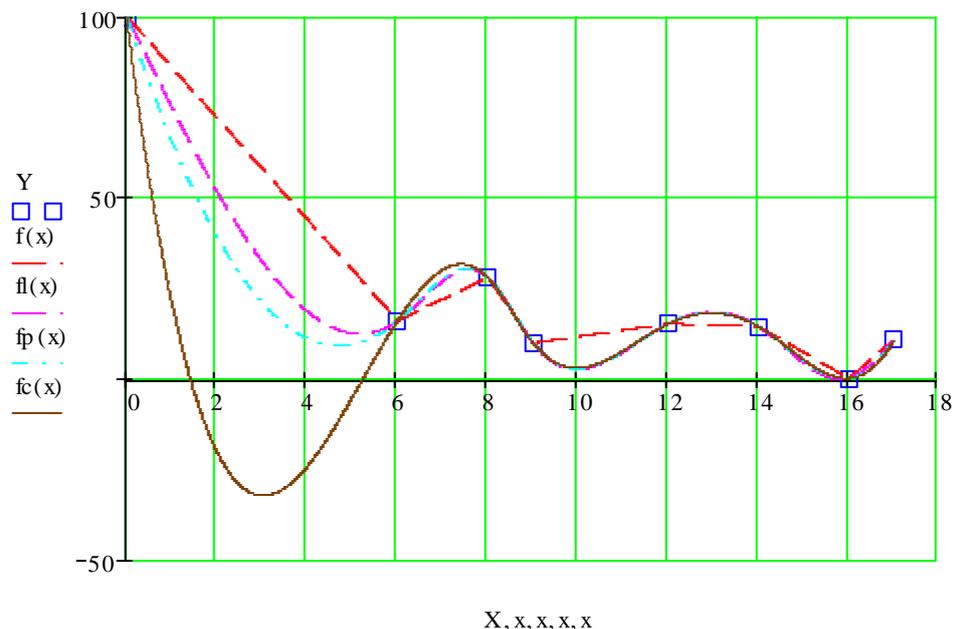
$Dc =$

	0
0	0
1	3
2	2
3	47.174
4	-7.05
5	-25.124
6	20.245
7	-5.878
8	-7.476
9	15.623
10	27.173

$fl(x) := interp(Dl, X, Y, x)$

$fp(x) := interp(Dp, X, Y, x)$

$fc(x) := interp(Dc, X, Y, x)$



Двумерная кубическая сплайн-интерполяция

Функции **lspline**, **pspline**, **cspline**, **interp** могут быть использованы также и для осуществления двумерной кубической сплайн-интерполяции поверхностей $z = F(x, y)$.

Для двумерного случая вызов функции осуществляется следующим образом:

cspline(XY,Z) — возвращает вектор D вторых производных функции в узловых точках, заданных значениями элементов двухстолбцовой матрицы XY и значениями элементов матрицы Z размерности $n \times n$.

Элементы векторов X и Y должны быть ранжированы по возрастанию их значений. Двухстолбцовая матрица XY может быть создана соединением векторов X и Y с использованием функции **augment**:

$$X := \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \end{pmatrix} \quad Y := \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \end{pmatrix}$$

$$XY := \text{augment}(X, Y)$$

$$XY = \begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 2 & 2 \\ 3 & 3 \end{pmatrix}$$

interp(D,XY,Z,v) — вычисляет $z = F(x, y)$ на всем диапазоне изменения значений x и y . Вектор v служит для рассматриваемого двумерного случая векторным аргументом, т. е. вектором с двумя компонентами x и y : $v = \begin{pmatrix} x \\ y \end{pmatrix}$.

Пример 3. Реализация двумерной интерполяции:

Информационные технологии в менеджменте

$$X := \begin{pmatrix} -5 \\ -4 \\ -3 \\ 0 \\ 4 \\ 5 \\ 6 \\ 7 \end{pmatrix} \quad Y := \begin{pmatrix} -5 \\ -4 \\ -3 \\ 0 \\ 4 \\ 5 \\ 6 \\ 7 \end{pmatrix}$$

$$XY := \text{augment}(X, Y) \quad XY = \begin{pmatrix} -5 & -5 \\ -4 & -4 \\ -3 & -3 \\ 0 & 0 \\ 4 & 4 \\ 5 & 5 \\ 6 & 6 \\ 7 & 7 \end{pmatrix}$$

$$Z := \begin{cases} \text{for } i \in 0..\text{length}(X) - 1 \\ \quad \text{for } j \in 0..\text{length}(Y) - 1 \\ \quad \quad Z_{i,j} \leftarrow \text{md}(X_i) + \text{md}(Y_j) \\ Z \end{cases} \quad Z = \begin{pmatrix} -0.973 & -4.326 & -4.637 & -3.552 & 0.132 & -4.347 & 3.145 & -1.845 \\ -2.088 & -5.213 & -6.372 & -2.157 & -0.33 & -0.93 & 3.976 & 3.237 \\ -1.406 & -5.114 & -3.686 & -1.374 & 1.143 & -0.959 & 1.827 & -1.072 \\ -3.499 & -0.566 & -1.28 & 0 & 0.765 & 0.778 & 1.678 & 5.053 \\ -3.681 & 0.363 & 2.149 & 1.887 & 5.756 & 3.937 & 6.363 & 4.118 \\ -1.559 & -0.081 & -0.393 & 1.43 & 6.574 & 1.272 & 5.295 & 3.562 \\ 1.796 & -0.165 & 0.886 & 3.772 & 4.934 & 3.358 & 1.152 & 11.851 \\ -0.462 & 2.59 & 1.62 & 4.373 & 3.511 & 4.724 & 7.059 & 9.528 \end{pmatrix}$$

$$D := \text{cspline}(XY, Z)$$

$$D = \begin{array}{|c|c|} \hline & 0 \\ \hline 0 & 0 \\ \hline 1 & 3 \\ \hline 2 & 2 \\ \hline 3 & 4.979 \\ \hline 4 & 1.798 \\ \hline 5 & -1.383 \\ \hline 6 & 0.328 \\ \hline 7 & 0.866 \\ \hline 8 & 3.035 \\ \hline 9 & -5.613 \\ \hline \end{array}$$

Информационные технологии в менеджменте

$$x := X_0, X_0 + 0.01.. X_{\text{last}(X)} \quad y := Y_0, Y_0 + 0.01.. Y_{\text{last}(Y)}$$

$$C(x, y) := \text{interp} \left[D, XY, Z, \begin{pmatrix} x \\ y \end{pmatrix} \right]$$

$$M := \text{CreateMesh} (C, \text{min}(X), \text{max}(X), \text{min}(Y), \text{max}(Y), 40, 40)$$

Для построения исходных опорных точек в пространстве необходимо преобразовать векторы X и Y в матрицы:

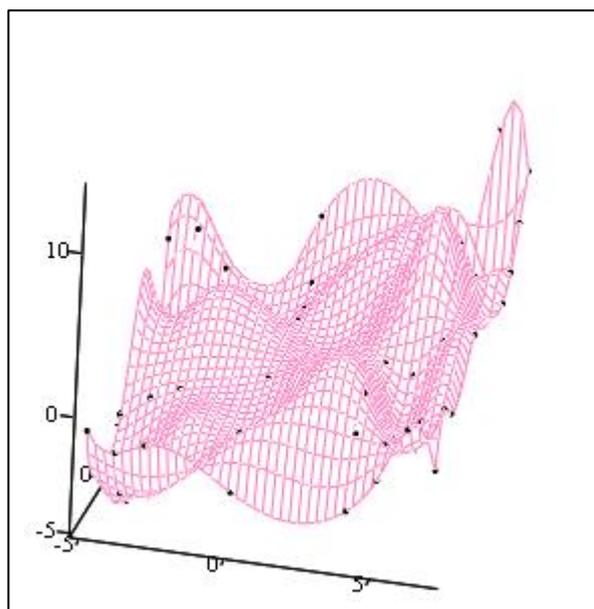
$$i := 0..7$$

$$XX^{(i)} := X$$

$$YY^{(i)} := Y$$

$$XX = \begin{pmatrix} -5 & -5 & -5 & -5 & -5 & -5 & -5 & -5 \\ -4 & -4 & -4 & -4 & -4 & -4 & -4 & -4 \\ -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \end{pmatrix}$$

$$YY = \begin{pmatrix} -5 & -5 & -5 & -5 & -5 & -5 & -5 & -5 \\ -4 & -4 & -4 & -4 & -4 & -4 & -4 & -4 \\ -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \end{pmatrix}$$



$$(XX, YY^T, Z), M$$

В **примере 3** для формирования значений элементов матрицы Z используется встроенная функция Mathcad **rnd(i)**, которая возвращает случайное число с равномерным распределением в диапазоне от 0 до i .

А для построения графика интерполированной поверхности использовалась функция:

$M := \text{CreateMesh}(f, x_n, x_k, y_n, y_k, s_1, s_2)$,

где f — функция, определяющая поверхность; x_n, x_k, y_n, y_k — начальные и конечные значения независимых переменных x и y ; s_1, s_2 — размерность сетки.

2. Функции для решения задач аппроксимации

Аппроксимация, или приближение — научный метод, состоящий в замене одних объектов другими, в том или ином смысле близкими к исходным, но более простыми.

Аппроксимация используется при решении разнообразных практических задач обработки массивов данных, в которых отразились результаты случайных воздействий и возмущений. Задачи, возникающие при этом, связаны, как правило, с попыткой найти математическое описание (математическую модель), адекватно представляющее совокупность данных. Такие задачи аппроксимации отличаются от только что рассмотренных задач интерполяции, которые не были связаны с получением математической модели, а состояли лишь в определении промежуточных значений функции между узлами.

Одномерная аппроксимация. Общую параметрическую аппроксимацию осуществляет функция:

linfit(X, Y, F) — вычисляет весовые коэффициенты набора локальных функций, введенных в виде выражений в F .

Пример 4. Реализация аппроксимации с использованием функции **linfit**:

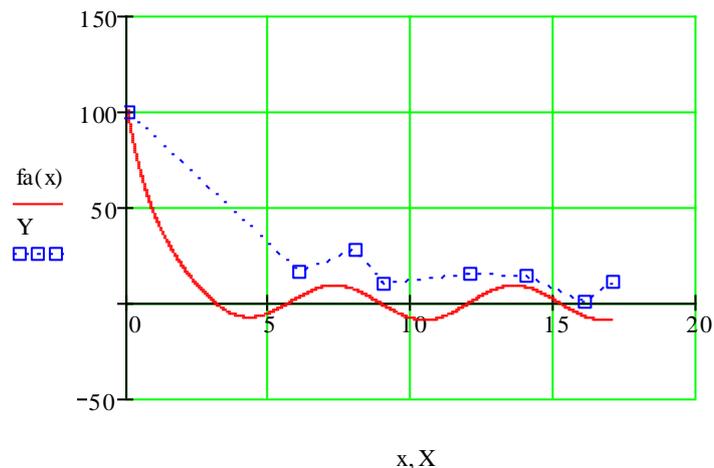
Информационные технологии в менеджменте

$$X := \begin{pmatrix} 0 \\ 6 \\ 8 \\ 9 \\ 12 \\ 14 \\ 16 \\ 17 \end{pmatrix} \quad Y := \begin{pmatrix} 100 \\ 16 \\ 28 \\ 9.9 \\ 15.2 \\ 14.4 \\ 0.16 \\ 10.7 \end{pmatrix} \quad F(z) := \begin{pmatrix} \exp(-z) \\ \sin(z) \\ \cos(z) \end{pmatrix}$$

$$L := \text{linfit}(X, Y, F)$$

$$L = \begin{pmatrix} 94.919 \\ 7.425 \\ 5.121 \end{pmatrix} \quad x := X_0, X_0 + 0.01.. X_{\text{last}}(X)$$

$$\text{fa}(x) := F(x) \cdot L$$



В Mathcad имеются функции, позволяющие решать задачи аппроксимации для специальных случаев, когда пользователь с достаточной степенью уверенности может назначить для исследуемого массива данных аппроксимирующее выражение в виде одной из простейших функций (экспоненциальной, синусоидальной, логарифмической и т. д.).

Встроенные функции этой группы несколько отличаются от ранее описанных. Они реализуют поисковые алгоритмы неизвестных коэффициентов и нуждаются, поэтому, в предварительном задании начальных приближений этих параметров.

В рассматриваемую группу входят функции, которые вычисляют коэффициенты следующих зависимостей, наилучшим образом аппроксимирующих данные, представленные в X, Y :

expfit(X,Y,g) — коэффициенты экспоненциальной функции $a \cdot \exp(bx) + c$;

lgsfit(X,Y,g) — коэффициенты логистической функции $\frac{a}{1 + b \cdot \exp(-cx)}$;

Информационные технологии в менеджменте

logfit(X,Y,g) — коэффициенты логарифмической функции $a \cdot \ln(x + b) + c$;

pwrfit(X,Y,g) — коэффициенты степенной функции вида $ax^b + c$;

sinfit(X,Y,g) — коэффициенты синусоидальной функции $a \cdot \sin(x + b) + c$.

Вектор g содержит начальные приближения искоемых коэффициентов. Перечисленные функции формируют результат определения параметров a , b и c в виде вектора-столбца $(a, b, c)^T$.

Пример 5. Применение одной из функций (**sinfit**) для определения параметров выражения $y(x)$ в условиях, когда значения функции были искажены аддитивной нормально распределенной помехой.

Для моделирования значений помехи используется функция **rnorm(n,mu,sigma)**, которая возвращает вектор из n случайных чисел, имеющих нормальное распределение со средним значением μ и стандартным отклонением σ .

`i := 0..10`

`X1 := i`

`пом := rnorm(11,0,0.4)`

`Y1 := 0.75 + sin(X1 + 0.5) + пом1`

$$g := \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

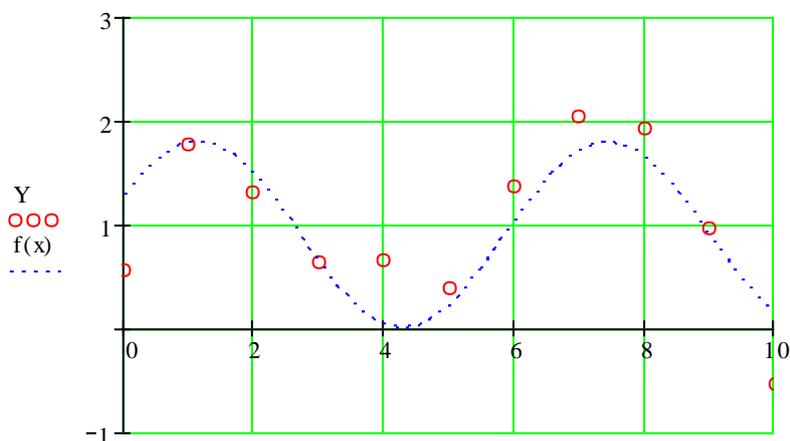
`C := sinfit(X, Y, g)`

$$C = \begin{pmatrix} 0.895 \\ 0.444 \\ 0.907 \end{pmatrix}$$

	0
0	-0.674
1	0.017
2	-0.048
3	0.223
4	0.877
5	0.323
6	0.394
7	0.345
8	0.366
9	0.269
10	-0.418

`x := 0,0.1..10`

`f(x) := C0 · sin(x + C1) + C2`



X, x

Существует также функция для обеспечения аппроксимации при произвольной степени аппроксимирующего полинома:

regress(X,Y,n).

Информационные технологии в менеджменте

Она возвращает вектор D , запрашиваемый функцией **interp(D,X,Y,x)**, содержащий коэффициенты многочлена l -й степени, который наилучшим образом приближает точки с координатами, хранящимися в векторах X и Y .

На практике не рекомендуется делать степень аппроксимирующего полинома выше 4—6, поскольку погрешности реализации сильно возрастают.

Функция **regress** создает единственный приближающий полином, коэффициенты которого вычисляются по всей совокупности заданных точек, т. е. глобально. Иногда полезна другая функция полиномиальной регрессии, дающая локальные приближения отрезками полиномов второй степени — **loess(X,Y,span)**. Эта функция возвращает используемый функцией **interp(D,X,Y,x)** вектор D , дающий наилучшее приближение данных (с координатами точек в векторах X и Y) отрезками полиномов второй степени.

Аргумент $span > 0$ задает степень сглаженности данных, определяя размер отрезков полиномов, из которых сшивается аппроксимирующая кривая. На практике $0,2 < span < 2$. При $span = 2$ результат аппроксимации тот же, что при аппроксимации одной параболой. При $span = 0,2$ аппроксимирующая кривая почти точно описывает любой набор данных. Рекомендуемое начальное значение — 0,75.

Пример 6. Одномерная аппроксимация с использованием функций **regress** и **loess**:

Информационные технологии в менеджменте

n := 4

span := 0.75

D1 := regress (X, Y, n)

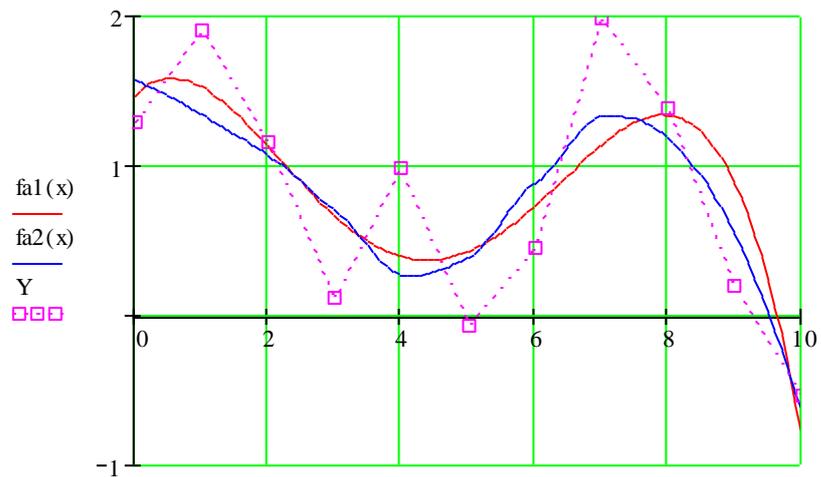
D2 := loess (X, Y, span)

$$D1 = \begin{pmatrix} 3 \\ 3 \\ 4 \\ 1.473 \\ 0.455 \\ -0.492 \\ 0.101 \\ -5.894 \times 10^{-3} \end{pmatrix}$$

	0
0	1
1	1.15
2	1.584
3	1.355
4	1.075
5	0.703
6	0.27
7	0.381
8	0.88
9	1.331
10	1.193
11	0.542
12	-0.66
13	-0.29
14	0.563
15	0.083

fa1(x) := interp(D1, X, Y, x)

fa2(x) := interp(D2, X, Y, x)



Двумерная аппроксимация. Mathcad позволяет выполнять также многомерную аппроксимацию, типичный случай которой — приближение трехмерных поверхностей. Их можно характеризовать вектором значений высот Z , соответствующих двумерному массиву XY координат точек на горизонтальной плоскости (XY — матрица $n \times 2$, содержащая координаты x и y).

Новых функций для этого не задано. Используются уже описанные функции в несколько иной форме (рис. 7):

Информационные технологии в менеджменте

— **regress(XY,Z,n)** — возвращает вектор D , запрашиваемый функцией **interp(D,XY,Z,v)** для вычисления многочлена n -й степени, который наилучшим образом приближает точки множеств XY и Z ; v — вектор с двумя компонентами x и y ;

— **loess(XY,Z,span)** — аналогична **loess(X,Y,span)**, но в многомерном случае.

Пример 7. Реализация двумерной аппроксимации с использованием функций **regress** и **loess**:

```
ii:=0..30      jj:=0..30
```

```
Xii:=ii      Yjj:=jj
```

```
XY := augment(X, Y)
```

```
Z := | for i ∈ 0..length(X) - 1
      | Zi ← sin(md(5))
      | Z
```

```
D1 := regress (XY, Z, 4)
```

```
D2 := loess (XY, Z, 1)
```

D1 =

	0
0	3
1	3
2	4
3	-2.871·10 ⁻⁶
4	-2.871·10 ⁻⁶
5	2.23·10 ⁻⁴
6	-5.389·10 ⁻³
7	2.23·10 ⁻⁴
8	-2.871·10 ⁻⁶
9	0.039
10	-5.389·10 ⁻³
11	2.23·10 ⁻⁴
12	-2.871·10 ⁻⁶
13	0.095
14	0.039
15	-5.389·10 ⁻³

D2 =

	0
0	1
1	196
2	0.218
3	0.191
4	0.168
5	0.147
6	0.124
7	0.102
8	0.084
9	0.068
10	0.054
11	0.04
12	0.028
13	0.021
14	0.016
15	9.509·10 ⁻³

Информационные технологии в менеджменте

$$x := \min(X), \min(X) + 0.01 \cdot \max(X)$$

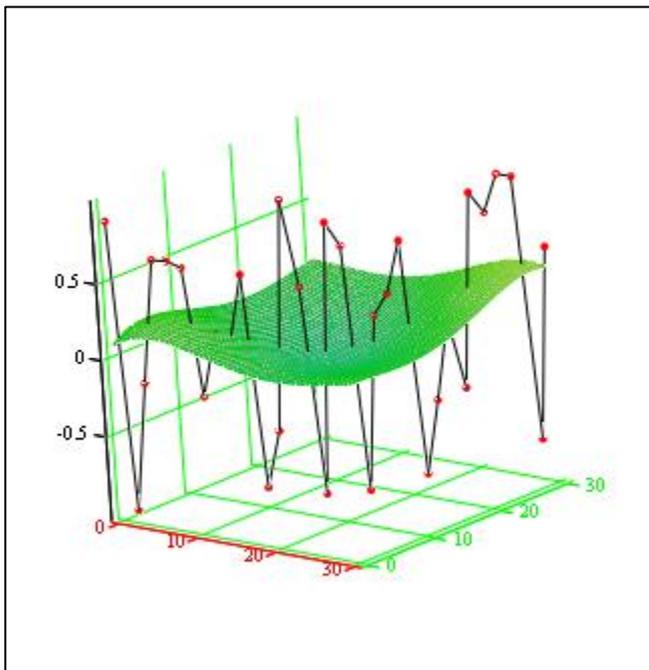
$$y := \min(Y), \min(Y) + 0.01 \cdot \max(Y)$$

$$C1(x, y) := \text{interp}\left[D1, XY, Z, \begin{pmatrix} x \\ y \end{pmatrix}\right]$$

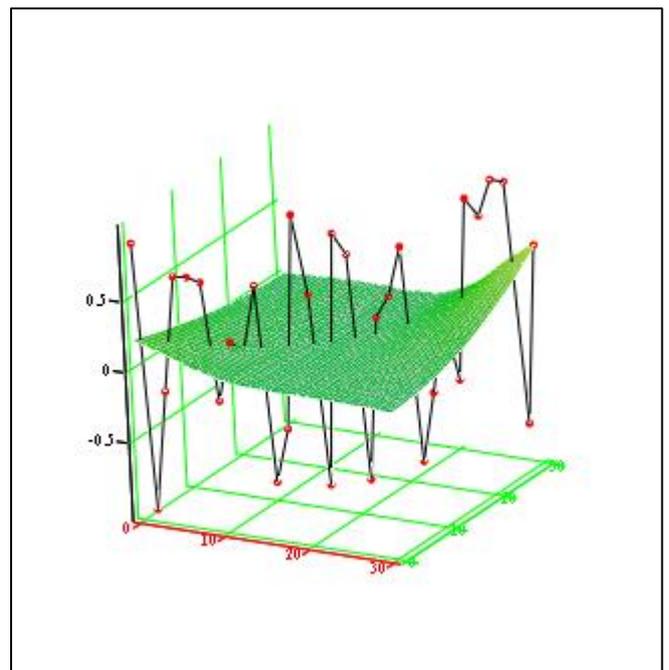
$$I1 := \text{CreateMesh}(C1, \min(X), \max(X), \min(Y), \max(Y), 50, 50)$$

$$C2(x, y) := \text{interp}\left[D2, XY, Z, \begin{pmatrix} x \\ y \end{pmatrix}\right]$$

$$I2 := \text{CreateMesh}(C2, \min(X), \max(X), \min(Y), \max(Y), 50, 50)$$



I1, (X, Y, Z)



I2, (X, Y, Z)

Лекция № 9

Mathcad: Функции для решения задач сглаживания и экстраполяции (предсказания)

1. Функции для решения задач сглаживания
2. Функции для решения задач экстраполяции (предсказания)

1. Функции для решения задач сглаживания

Под *сглаживанием* понимается процедура повышения достоверности данных, снижения уровня помех. Необходимость сглаживания часто возникает на этапе предварительной обработки данных, перед их использованием для получения количественных характеристик, построения математических моделей и т. д.. В качестве алгоритма такого предварительного сглаживания обычно используется процедура так называемого *скользящего окна*, под которым понимается обработка данных на выбранной части диапазона изменения аргумента при поэтапном перемещении этой части диапазона (скольжении «окна») по оси абсцисс. Для сглаживания в таком понимании в Mathcad предусмотрены три функции. При этом исходный массив задается для функций в виде векторов X и Y одинаковой размерности $(n \times 1)$. Перечисленные функции выводят результат в виде вектора сглаженных значений той же размерности:

medsmooth(Y,m) — предназначена для обработки исходного массива чисел, заданных вектором Y , методом скользящей медианы и вычисления сглаженных (по сравнению с исходными) оценок переменных y ; t — ширина окна обработки, значение t должно быть нечетным и меньшим, чем число элементов вектора Y .

Медианы давно использовались и изучались в статистике как альтернатива средним арифметическим значениям отсчетов в оценке выборочных средних значений. *Медианой числовой последовательности* x_1, x_2, \dots, x_n при нечетном n является средний по положению член ряда, получающегося при упорядочивания этой последовательности по возрастанию (или убыванию). Для четных n медиану определяют как среднее арифметическое двух средних отсчетов упорядоченной последовательности.

ksmooth(X,Y,b) — в предположении нормального распределения вычисляет сглаженные оценки методом средневзвешенной обработки, b — ширина окна обработки, которая должна в несколько раз превышать интервал между точками по оси x ; вектор X должен содержать элементы, ранжированные в порядке возрастания.

Среднее взвешенное, точнее среднее арифметическое взвешенное набора вещественных чисел x_1, x_2, \dots, x_n с вещественными весами w_1, w_2, \dots, w_n определяется как:

$$\bar{x} = \frac{\sum_{i=1}^n w_i \cdot x_i}{\sum_{i=1}^n w_i}.$$

Часто подразумевается, что сумма весов равна 1, тогда формула выглядит следующим образом:

$$\bar{x} = \sum_{i=1}^n w_i \cdot x_i.$$

В том случае, если все веса равны между собой, среднее арифметическое взвешенное будет равно среднему арифметическому.

supsmooth(X,Y) — вычисляет сглаженные оценки методом наименьших квадратов для k ближайших соседних точек, параметр k выбирается адаптивно.

Метод наименьших квадратов (МНК, OLS, Ordinary Least Squares) — математический метод, применяемый для решения различных задач, основанный на минимизации суммы квадратов некоторых функций от искомых переменных. Этот метод широко используется в регрессионном анализе (аппроксимации данных), поскольку позволяет построить модель на основе зашумленных экспериментальных данных. При этом помимо собственно построения модели обычно осуществляется оценка погрешности, с которой были вычислены ее параметры.

Суть метода заключается в том, чтобы найти коэффициенты C_0, C_1, \dots, C_m линейной комбинации:

$$L(t) = C_0 \varphi_0(t) + C_1 \varphi_1(t) + C_2 \varphi_2(t) + \dots + C_m \varphi_m(t),$$

таким образом, чтобы обеспечить близость кривой $L(t)$ и заданной функции.

Коэффициенты определяются из следующего матричного уравнения:

$$C = (A^T \cdot A)^{-1} \cdot A^T \cdot f,$$

где f — заданная зашумленная функция; A — прямоугольная матрица размерностью $(n+1) \times (m+1)$; n — количество точек заданной функции f ; m — порядок функции, используемой для аппроксимации сигнала.

Матрица A имеет следующий вид:

$$A = \begin{bmatrix} \varphi_0(t_0) & \varphi_1(t_0) & \dots & \varphi_m(t_0) \\ \varphi_0(t_1) & \varphi_1(t_1) & \dots & \varphi_m(t_1) \\ \dots & \dots & \dots & \dots \\ \varphi_0(t_n) & \varphi_1(t_n) & \dots & \varphi_m(t_n) \end{bmatrix}.$$

При использовании для аппроксимации степенных функций $L(t)$ будет иметь вид: $L(t) = C_0 t^0 + C_1 t^1 + C_2 t^2 + \dots + C_m t^m$, а матрица A :

$$A = \begin{bmatrix} t_0^0 & t_0^1 & \dots & t_0^m \\ t_1^0 & t_1^1 & \dots & t_1^m \\ \dots & \dots & \dots & \dots \\ t_n^0 & t_n^1 & \dots & t_n^m \end{bmatrix}.$$

При использовании тригонометрических функций (параметр $T = 2$):

$$L(t) = C_0 + C_1 \sin\left(\frac{\pi t}{T}\right) + C_2 \cos\left(\frac{\pi t}{T}\right) + C_3 \sin\left(\frac{2\pi t}{T}\right) + C_4 \cos\left(\frac{2\pi t}{T}\right) + \dots,$$

$$A = \begin{bmatrix} 1 & \sin\left(\frac{\pi t_0}{T}\right) & \cos\left(\frac{\pi t_0}{T}\right) & \sin\left(\frac{2\pi t_0}{T}\right) & \cos\left(\frac{2\pi t_0}{T}\right) & \dots & \sin\left(\frac{m\pi t_0}{T}\right) & \cos\left(\frac{m\pi t_0}{T}\right) \\ 1 & \sin\left(\frac{\pi t_1}{T}\right) & \cos\left(\frac{\pi t_1}{T}\right) & \sin\left(\frac{2\pi t_1}{T}\right) & \cos\left(\frac{2\pi t_1}{T}\right) & \dots & \sin\left(\frac{m\pi t_1}{T}\right) & \cos\left(\frac{m\pi t_1}{T}\right) \\ \dots & \dots \\ 1 & \sin\left(\frac{\pi t_n}{T}\right) & \cos\left(\frac{\pi t_n}{T}\right) & \sin\left(\frac{2\pi t_n}{T}\right) & \cos\left(\frac{2\pi t_n}{T}\right) & \dots & \sin\left(\frac{m\pi t_n}{T}\right) & \cos\left(\frac{m\pi t_n}{T}\right) \end{bmatrix}$$

Пример 1. Примеры использования функций сглаживания для случайного процесса.

Для функции **medsmooth** выбрана ширина окна обработки $m = 9$. Анализируя влияние ширины окна обработки на конечный результат, можно убедиться, что при снижении ширины окна сглаживающие способности использованной функции будут снижаться.

Для функции **ksmooth** была принята малая величина окна обработки, что привело к увеличению разброса результирующей кривой.

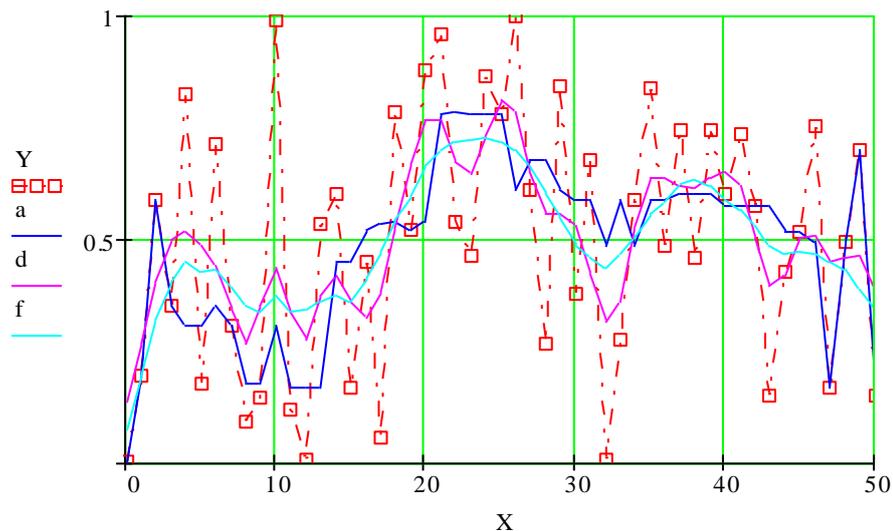
Применение функции **supsmooth** в тех же условиях дает лучший результат.

```
i := 0..50      m := 9      b := 3

X1 := i

Y1 := md(1)

a := medsmooth(Y, m)      d := ksmooth(X, Y, b)      f := supsmooth(X, Y)
```



Из приведенных результатов использования функций сглаживания виден эффект «подавления помех», важный при предварительной обработке данных.

2. Функции для решения задач экстраполяции (предсказания)

Прогнозирование поведения переходных процессов, сигналов, зависимостей в Mathcad осуществляется функцией:

predict(Y, m, k)

где Y — вектор данных; m — число последних точек существующих данных, на основе которых происходит расчет предсказываемых точек; k — число точек, в которых необходимо предсказать данные.

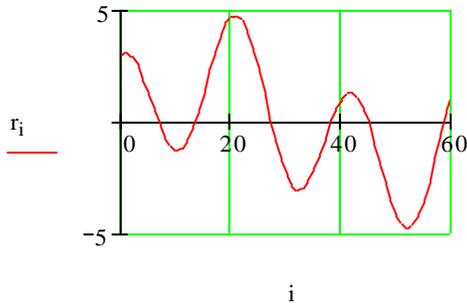
Она по ряду заданных равномерно расположенных точек позволяет рассчитать некоторое число k последующих точек, т. е. по существу выполняет экстраполяцию произвольной (но достаточно гладкой и предсказуемой) зависимости. Функция предсказания обеспечивает высокую точность при монотонных исходных функциях или функциях, представляемых полиномом невысокой степени.

Эта функция использует простейший вид алгоритма экстраполяции — линейный, поэтому исходные данные должны быть достаточно гладкими и с хорошо проявляющимися динамическими особенностями, т. е. не должны быть сильно искажены помехами.

Пример 2. Пример получения экстраполированных значений процесса, имеющего шумовую составляющую.

$i := 0..60$

$r_i := 2 \cdot \sin(i \cdot 0.1) + 3 \cdot \cos(i \cdot 0.3)$



$y := \text{morm}(61, 0, 0.1)$

$y1 := \text{morm}(61, 0, 0.3)$

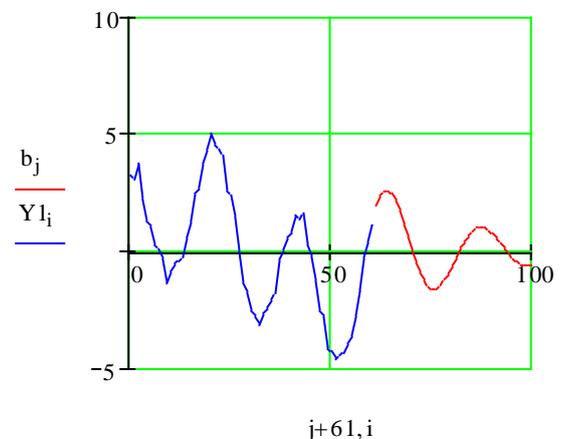
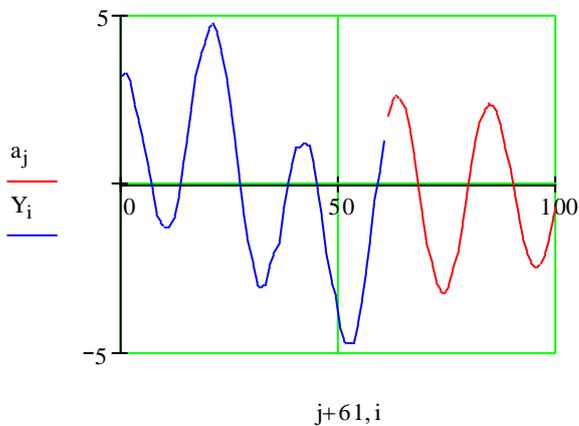
$Y := r + y$

$Y1 := r + y1$

$a := \text{predict}(Y, 7, 40)$

$b := \text{predict}(Y1, 7, 40)$

$j := 0..39$



При этом используется функция **rnorm(n,mu,sigma)**, которая возвращает вектор из n случайных чисел, имеющих нормальное распределение со средним значением μ и стандартным отклонением σ .

При незначительном среднеквадратическом отклонении шума от нулевого математического ожидания, равном 0,1 (левый график), исходный процесс имеет небольшие искажения и близок к синусоидальному. Линейная экстраполяция осуществляется с 61-й точки и, при опоре на 7 точек, дает достаточно хорошие результаты. Увеличение стандарта отклонения помехи до 0,3 (правый график) заметно искажает исходные данные и вызывает ухудшение качества результирующих значений оценок экстраполяции.

Лекция № 10

Ознакомление с пакетом Matlab. Выражения

1. Основные возможности пакета Matlab
2. Наборы инструментов пакета Matlab
3. Структура и рабочие окна пакета Matlab
4. Работа в командном режиме
5. Основные элементы языка программирования Matlab

1. Основные возможности пакета Matlab

MATLAB (сокращение от англ. «Matrix Laboratory») — пакет прикладных программ для решения задач технических вычислений и одноименный язык программирования, используемый в этом пакете. MATLAB используют более 1 000 000 инженерных и научных работников, он работает на большинстве современных операционных систем, включая Linux, Mac OS, Solaris (начиная с версии R2010b поддержка Solaris прекращена) и Microsoft Windows.

История. MATLAB как язык программирования был разработан Кливом Моулером (англ. Cleve Moler) в конце 1970-х годов когда он был деканом факультета компьютерных наук в Университете Нью-Мексико. Целью разработки служила задача дать студентам факультета возможность использования программных библиотек Linpack и EISPACK без необходимости изучения Фортрана. Вскоре новый язык распространился среди других университетов и был с большим интересом встречен учеными, работающими в области прикладной математики. До сих пор в Интернете можно найти версию 1982 года, написанную на Фортране, распространяемую с открытым исходным кодом. Инженер Джон Литтл (англ. John N. (Jack) Little) познакомился с этим языком во время визита Клива Моулера в Стэнфордский университет в 1983 году. Поняв, что новый язык обладает большим коммерческим потенциалом, он объединился с Кливом Моулером и Стивом Бангертом (англ. Steve Bangert). Совместными усилиями они переписали MATLAB на С и основали в 1984 компанию The MathWorks для дальнейшего развития. Эти переписанные на С библиотеки долгое время были известны под именем JASCRAS. Первоначально MATLAB предназначался для проектирования систем управления (основная специальность Джона Литтла), но быстро завоевал популярность во многих других научных и инженерных областях. Он также широко использовался и в образовании, в частности, для преподавания линейной алгебры и численных методов.

Информационные технологии в менеджменте

Описание языка MATLAB. Язык MATLAB является высокоуровневым интерпретируемым языком программирования, включающим основанные на матрицах структуры данных, широкий спектр функций, интегрированную среду разработки, объектно-ориентированные возможности и интерфейсы к программам, написанным на других языках программирования.

Программы, написанные на MATLAB, бывают двух типов — *функции* и *скрипты*.

Функции имеют входные и выходные аргументы, а также собственное рабочее пространство для хранения промежуточных результатов вычислений и переменных.

Скрипты же используют общее рабочее пространство. Как скрипты, так и функции не компилируются в машинный код и сохраняются в виде текстовых файлов.

Существует также возможность сохранять так называемые *pre-parsed* программы — функции и скрипты, обработанные в вид, удобный для машинного исполнения. В общем случае такие программы выполняются быстрее обычных, особенно если функция содержит команды построения графиков.

Основной особенностью языка MATLAB являются его широкие возможности по работе с матрицами, которые создатели языка выразили в лозунге «думай векторно» (англ. *Think vectorized*).

Применение MATLAB.

Математика и вычисления. MATLAB предоставляет пользователю большое количество (несколько сотен) функций для анализа данных, покрывающие практически все области математики, в частности:

- Матрицы и линейная алгебра — алгебра матриц, линейные уравнения, собственные значения и вектора, сингулярности, факторизация матриц и другие.
- Многочлены и интерполяция — корни многочленов, операции над многочленами и их дифференцирование, интерполяция и экстраполяция кривых и другие.
- Математическая статистика и анализ данных — статистические функции, статистическая регрессия, цифровая фильтрация, быстрое преобразование Фурье и другие.
- Обработка данных — набор специальных функций, включая построение графиков, оптимизацию, поиск нулей, численное интегрирование (в квадратурах) и другие.
- Дифференциальные уравнения — решение дифференциальных и дифференциально-алгебраических уравнений, дифференциальных уравнений с запаздыванием, уравнений с ограничениями, уравнений в частных производных и другие.
- Разреженные матрицы — специальный класс данных пакета MATLAB, использующийся в специализированных приложениях.
- Целочисленная арифметика — выполнение операций целочисленной арифметики в среде MATLAB.

Разработка алгоритмов. MATLAB предоставляет удобные средства для разработки алгоритмов, включая высокоуровневые с использованием концепций объектно-ориентированного программирования. В нем имеются все необходимые средства интегрированной среды разработки, включая отладчик и профайлер. Функции для работы с целыми типами данных облегчают создание алгоритмов для микроконтроллеров и других приложений, где это необходимо.

Визуализация данных. В составе пакета MATLAB имеется большое количество функций для построения графиков, в том числе трехмерных, визуального анализа данных и создания анимированных роликов.

Встроенная среда разработки позволяет создавать графические интерфейсы пользователя с различными элементами управления, такими как кнопки, поля ввода и другими.

Независимые приложения. Программы MATLAB, как консольные, так и с графическим интерфейсом пользователя, могут быть собраны с помощью компоненты *MATLAB Compiler* в независимые от MATLAB исполняемые приложения или динамические библиотеки, для запуска которых на других компьютерах, однако, требуется установка свободно распространяемой среды *MATLAB Compiler Runtime* (MCR).

Внешние интерфейсы. Пакет MATLAB включает различные интерфейсы для получения доступа к внешним подпрограммам, написанным на других языках программирования, данным, клиентам и серверам, общающимся через технологии Component Object Model или Dynamic Data Exchange, а также периферийным устройствам, которые взаимодействуют напрямую с MATLAB. Многие из этих возможностей известны под названием MATLAB API.

COM. Пакет MATLAB предоставляет доступ к функциям, позволяющим создавать, манипулировать и удалять COM-объекты (как клиенты, так и серверы). Поддерживается также технология [ActiveX](#). Все COM-объекты принадлежат к специальному COM-классу пакета MATLAB. Все программы, имеющие функции контроллера автоматизации ([англ. Automation controller](#)) могут иметь доступ к MATLAB как к серверу автоматизации ([англ. Automation server](#)).

.NET. Пакет MATLAB в Microsoft Windows предоставляет доступ к программной платформе .NET Framework. Имеется возможность загружать .NET сборки (Assemblies) и работать с объектами .NET классов из среды MATLAB. В версии MATLAB 7.11 (R2010b) поддерживается .NET Framework версий 2.0, 3.0, 3.5 и 4.0.

DDE. Пакет MATLAB содержит функции, которые позволяют ему получать доступ к другим приложениям среды [Windows](#), равно как и этим приложениям получать доступ к данным MATLAB, посредством технологии динамического обмена данными (DDE). Каждое приложение, которое может быть DDE-сервером, имеет свое уникальное идентификационное имя. Для MATLAB это имя — *Matlab*.

Веб-сервисы. В MATLAB существует возможность вызывать методы веб-сервисов. Специальная функция создает класс, основываясь на методах API веб-сервиса.

MATLAB взаимодействует с клиентом веб-сервиса с помощью принятия от него посылок, их обработки и посылок ответа. Поддерживаются следующие технологии: Simple Object Access Protocol (SOAP) и Web Services Description Language (WSDL).

COM-порт. Интерфейс для последовательного порта пакета MATLAB обеспечивает прямой доступ к периферийным устройствам, таким как модемы, принтеры и научное оборудование, подключающееся к компьютеру через последовательный порт (COM-порт). Интерфейс работает путем создания объекта

Информационные технологии в менеджменте

специального класса для последовательного порта. Имеющиеся методы этого класса позволяют считывать и записывать данные в последовательный порт, использовать события и обработчики событий, а также записывать информацию на диск компьютера в режиме реального времени. Это бывает необходимо при проведении экспериментов, симуляции систем реального времени и для других приложений.

MEX-файлы. Пакет MATLAB включает интерфейс взаимодействия с внешними приложениями, написанными на языках С и Фортран. Осуществляется это взаимодействие через MEX-файлы. Существует возможность вызова подпрограмм, написанных на С или Фортране из MATLAB, как будто это встроенные функции пакета. MEX-файлы представляют собой динамически подключаемые библиотеки, которые могут быть загружены и исполнены интерпретатором, встроенным в MATLAB. MEX-процедуры имеют также возможность вызывать встроенные команды MATLAB.

DLL. Интерфейс MATLAB, относящийся к общим DLL позволяет вызывать функции, находящиеся в обычных динамически подключаемых библиотеках, прямо из MATLAB. Эти функции должны иметь С-интерфейс.

Кроме того, в MATLAB имеется возможность получить доступ к его встроенным функциям через С-интерфейс, что позволяет использовать функции пакета во внешних приложениях, написанных на С. Эта технология в MATLAB называется *C Engine*.

Альтернативные пакеты. Существует большое количество программных пакетов для решения задач численного анализа. Многие из таких пакетов являются свободным программным обеспечением.

Совместимые с MATLAB на уровне языка программирования:

- [GNU Octave](#)
- [FreeMat](#)
- [Maxima](#)
- [Scilab](#)

Близкие по функциональности:

- [R](#), [S](#) и [SPlus](#)
- [APL](#) и его потомки: например [J](#)
- [Python](#), при использовании пакета программ [Python\(x,y\)](#), а также с такими библиотеками как [NumPy](#), [SciPy](#) и [matplotlib](#) реализует сходные возможности.

- [IDL](#) ([англ. Interactive Data Language](#), интерактивный язык описания данных), когда-то был коммерческим конкурентом MATLAB, сейчас остается серьезным конкурентом во многих прикладных областях, хотя его доля на рынке программных продуктов для численного анализа резко упала.

- [Fortress](#), язык программирования, созданный Sun Microsystems, является наследником Фортрана, но с ним не совместим.

- При необходимости разработки больших проектов для численного анализа возможно использование языков программирования общего назначения, поддерживающих статическую типизацию и модульную структуру. Примерами могут служить [Modula-3](#), [Haskell](#), [Ада](#), [Java](#). При этом рекомендуется использовать известные в научно-инженерной среде специализированные библиотеки.

2. Наборы инструментов пакета Matlab

В Matlab важная роль отводится специализированным группам программ, называемых *toolboxes*. *Toolboxes* — это всесторонняя коллекция функций (м-файлов), написанных на языке MATLAB для решения определенного класса задач. Компания Mathworks поставляет наборы инструментов, которые используются во многих областях, включая следующие:

- **Цифровая обработка сигналов, изображений и данных:** *DSP Toolbox, Image Processing Toolbox, Wavelet Toolbox, Communication Toolbox, Filter Design Toolbox* — наборы функций, позволяющих решать широкий спектр задач обработки сигналов, изображений, проектирования цифровых фильтров и систем связи.

- **Системы управления:** *Control Systems Toolbox, μ -Analysis and Synthesis Toolbox, Robust Control Toolbox, System Identification Toolbox, LMI Control Toolbox, Model Predictive Control Toolbox, Model-Based Calibration Toolbox* — наборы функций, облегчающих анализ и синтез динамических систем, проектирование, моделирование и идентификацию систем управления, включая современные алгоритмы управления, такие как робастное управление, [H \$\infty\$ -управление](#), ЛМН-синтез, [\$\mu\$ -синтез](#) и другие.

- **Финансовый анализ:** *GARCH Toolbox, Fixed-Income Toolbox, Financial Time Series Toolbox, Financial Derivatives Toolbox, Financial Toolbox, Datafeed Toolbox* — наборы функций, позволяющие быстро и эффективно собирать, обрабатывать и передавать различную финансовую информацию.

- **Анализ и синтез географических карт, включая трехмерные:** *Mapping Toolbox*.

- **Сбор и анализ экспериментальных данных:** *Data Acquisition Toolbox, Image Acquisition Toolbox, Instrument Control Toolbox, Link for Code Composer Studio* — наборы функций, позволяющих сохранять и обрабатывать данные, полученные в ходе экспериментов, в том числе в реальном времени. Поддерживается широкий спектр научного и инженерного измерительного оборудования.

- **Визуализация и представление данных:** *Virtual Reality Toolbox* — позволяет создавать интерактивные миры и визуализировать научную информацию с помощью технологий виртуальной реальности и языка VRML.

- **Средства разработки:** *MATLAB Builder for COM, MATLAB Builder for Excel, MATLAB Builder for NET, MATLAB Compiler, Filter Design HDL Coder* — наборы функций, позволяющих создавать независимые приложения из среды MATLAB.

- **Взаимодействие с внешними программными продуктами:** *MATLAB Report Generator, Excel Link, Database Toolbox, MATLAB Web Server, Link for ModelSim* — наборы функций, позволяющие сохранять данные таким образом, чтобы другие программы могли с ними работать.

- **Базы данных:** *Database Toolbox* — инструменты работы с базами данных.

- **Научные и математические пакеты:** *Bioinformatics Toolbox, Curve Fitting Toolbox, Fixed-Point Toolbox, Fuzzy Logic Toolbox, Genetic Algorithm and Direct Search Toolbox, OPC Toolbox, Optimization Toolbox, Partial Differential Equation Toolbox, Spline Toolbox, Statistic Toolbox, RF Toolbox* — наборы специализированных математических функций, позволяющие решать широкий спектр научных и инженерных задач, включая разработку

Информационные технологии в менеджменте

генетических алгоритмов, решения задач в частных производных, целочисленные проблемы, оптимизацию систем и другие.

• **Нейронные сети:** *Neural Network Toolbox* — инструменты для синтеза и анализа нейронных сетей.

• **Нечеткая логика:** *Fuzzy Logic Toolbox* — инструменты для построения и анализа нечетких множеств.

• **Символьные вычисления:** *Symbolic Math Toolbox* — инструменты для символьных вычислений с возможностью взаимодействия с символьным процессором программы [Maple](#).

Помимо вышеперечисленных, существуют тысячи других наборов инструментов для MATLAB, написанных другими компаниями и энтузиастами.

3. Структура и рабочие окна пакета Matlab

Система Matlab состоит из пяти основных частей.

Язык Matlab. Это язык матриц и массивов высокого уровня с управлением потоками, функциями, структурами данных, вводом-выводом и особенностями объектно-ориентированного программирования.

Среда Matlab. Это набор инструментов и приспособлений, с которыми работает пользователь или программист Matlab. Она включает в себя средства для управления переменными в рабочем пространстве Matlab, вводом и выводом данных, а также создания, контроля и отладки М-файлов и приложений Matlab.

Управляемая графика. Это графическая система Matlab, которая включает в себя команды высокого уровня для визуализации двух- и трехмерных данных, обработки изображений, анимации и иллюстрированной графики. Она также включает в себя команды низкого уровня, позволяющие полностью редактировать внешний вид графики, также как при создании Графического Пользовательского Интерфейса (GUI) для Matlab приложений.

Библиотека математических функций. Это обширная коллекция вычислительных алгоритмов от элементарных функций, таких как сумма, синус, косинус, комплексная арифметика, до более сложных, таких как обращение матриц, нахождение собственных значений, функции Бесселя, быстрое преобразование Фурье.

Программный интерфейс. Это библиотека, которая позволяет писать программы на Си и Фортране, которые взаимодействуют с Matlab. Она включает средства для вызова программ из Matlab (динамическая связь), вызывая Matlab как вычислительный инструмент и для чтения-записи mat-файлов.

О Simulink. Simulink, сопутствующая Matlab программа, — это интерактивная система для моделирования динамических систем. Она представляет собой среду, управляемую мышью, которая позволяет моделировать процесс путем перетаскивания блоков диаграмм на экране и их манипуляцией. Simulink работает с линейными, нелинейными, непрерывными, дискретными, многомерными системами.

При загрузке системы Matlab появляется несколько окон системы.

Окно Command Window. Окно **Command Window (Окно команд)** является для пользователя наиболее важным. Посредством этого окна вводятся математические выражения, получаются результаты вычислений, а также выдаются сообщения, посылаемые системой. Данное окно становится доступным пользователю сразу же после запуска программы. Математические выражения пишутся в командной строке после знака приглашения **>>**.

Необходимо набрать в строке ввода выражение

Информационные технологии в менеджменте

Для выполнения действия нажать клавишу Enter.

Результат выведется ниже

Если нужно исправить одну из цифр, то ничего не получится. Невозможность редактирования ранее введенной команды простой установкой курсора в нужную строку является одной из особенностей системы Matlab. Для того чтобы отредактировать ранее введенную команду, необходимо установить курсор в строку ввода и воспользоваться клавишами ↑ и ↓. Эти клавиши позволяют пролистать стек введенных ранее команд и оставить в строке ту команду, которая необходима. Команду можно выполнить сразу (нажав клавишу Enter) или после редактирования.

Понятие о файлах-сценариях. В командном режиме выполнять решения задач нецелесообразно, т. к. исправить возможные ошибки после нажатия клавиши Enter уже нельзя. Поэтому записывать программы, их редактировать и отлаживать необходимо в так называемых *m-файлах*. m-файл создается при выполнении команды **New** меню **File**. Для ускорения этой команды выведена специальная пиктограмма в виде белой странички с загнутым уголком на панели инструментов. Щелкнув по пиктограмме стрелкой мышки, получаем окно m-файла, на котором можно записывать, редактировать и отлаживать любые программы решения научных и инженерных задач. Данный m-файл по умолчанию имеет название **Untitled (Безымянный)**. Чтобы дать ему имя, необходимо в меню этого окна **File** выполнить команду **Save as** и в другом окне указать папку и имя этого файла. После указания имени и сохранения m-файла он готов для выполнения записанной программы. Для этого необходимо щелкнуть мышкой по пиктограмме **Выполнить**. Она выполнена в виде страницы со стрелкой, направленной вниз. Результат выполнения программы или сообщения об ошибках появится в окне команд. Описанный процесс называется *созданием m-файла сценария сессии*. Файл-сценарий, именуемый также *Script-файлом*, имеет простую структуру:

% Основной комментарий, если необходимо.

% Дополнительный комментарий, если необходимо.

Тело программы с любыми выражениями.

Важными являются следующие свойства файлов-сценариев:

1. Они не имеют входных и выходных аргументов.
2. Работают с данными из рабочей области.
3. В процессе выполнения не компилируются.
4. Представляют собой *последовательность операции*, аналогичную той, что используется в сессии.

Окно команд имеет еще два управляющих элемента, в правом верхнем углу. Это кнопка закрытия окна ×, и расположенная левее кнопка отделения окна от интерфейса системы ↵. После нажатия на эту кнопку окно становится автономным, имеющим собственное меню. Возврат окна в общий интерфейс происходит посредством выбора из его главного меню пунктов **View** → **Dock Command Window (Вид** → **Пристыковать окно команд)**. Такими же кнопками снабжены другие четыре окна.

Рассмотренное окно является основным и, в принципе, при работе с системой Matlab можно было бы обойтись без других.

Однако наличие остальных окон делают диалог с компьютером легким и комфортным.

Окно Workspace. В процессе работы используются переменные различных типов. Созданные переменные хранятся в специально отведенной области памяти

компьютере. Они не исчезают сами по себе, а только при выходе из программы или с помощью специальных команд.

При этом переменные (точнее их значения) можно использовать в любом вводимом математическом выражении. **Окно Workspace (Рабочая область)** предоставляет пользователю список всех переменных, хранящихся в рабочем пространстве. Выбрать можно любую переменную, просмотреть ее содержимое или выполнить какие-либо другие действия.

Упомянутые действия выполняются посредством контекстного меню (нужно щелкнуть правой кнопкой мыши по имени переменной в списке).

Окно Current Directory. Окно Current Directory (Текущий каталог) является аналогом известной программы Проводник, но имеет для Matlab свое особое предназначение.

Дело в том что, кроме работы с математическими выражениями из командного окна, пользователь также может работать с файлами. К тому же используемые математические функции, физически представляют собой файлы, названные по именам функции. В этих файлах записаны программы, реализующие функции. Таким образом, пользователь постоянно использует файлы. Например, указывая встроенную функцию, фактически пишется имя файла (без расширения), в котором хранится текст программы. В этом случае система должна будет искать требуемый файл в текущем каталоге или в пути доступа.

Окно Command History. Все команды, которые набираются в командной строке Command Window (Окно команд), автоматически образуют список, который и выводится в окне Command History (История команд). Чем полезен этот список? Если появилась необходимость повторить ранее выполненную команду, то она отыскивается в списке **Command History (История команд)**, и, дважды щелкнув по ней левой кнопкой мыши, можно команду выполнить.

Можно выполнить нужную последовательность команд из командной строки и получить соответствующую последовательность команд в **Command History (История команд)**. Несколькими простыми действиями полученная последовательность может быть преобразована в программу. Содержимое данного окна не теряется после выхода из системы и выключения компьютера. Удалить список команд можно только с помощью меню.

Окно Launch Pad. Окно Launch Pad (Панель запуска) содержит дерево файловой системы, где отображены только установленные на компьютере программные продукты, входящие в систему Matlab. С помощью этого окна можно запустить любой из них. Те же действия легко осуществить посредством кнопки **Start (Пуск)**, расположенной в левом нижнем углу окна Matlab.

4. Работа в командном режиме

В командной строке в режиме диалога можно набрать команду (оператор) или выражение и, нажав Enter, получить ответ (answer). Например, после набора команды (оператора присваивания)

a=3.2

в последующих строках появится

a = 3.200000000000000 (переменной a присвоено значение 3.2), после набора выражения

sin(a)/a

увидим его значение

ans = -0.01824191982112

Информационные технологии в менеджменте

Если вы хотите выполнить команду без вывода результата, в конце команды ставьте символ точки с запятой.

Если математическое выражение выходит за размер экрана монитора, то целесообразно перенести его часть на следующую строку. Для этого используется символ многоточие ... — три и более точки. В командном режиме число возможных символов в одной строке 4096, в m-файле не ограничено, но с такими длинными строками работать неудобно. Поэтому применение в файлах-сценариях символа переноса строки улучшает наглядность программ.

Любой фрагмент окна командной строки можно выделить и копировать в буфер, например, для переноса в Word. Возможен перенос в командную строку текстовых фрагментов из других систем.

Сеанс работы с Matlab принято именовать *сессией*. Сессия, в сущности, является текущим документом, отражающим работу пользователя с системой Matlab. В ней имеются строки ввода, вывода и сообщения об ошибках. Некоторые команды управления окном командного режима:

clc — очищает экран и размещает курсор в левом верхнем углу пустого экрана;

echo name_file on — включает режим вывода на экран текста Script-файла (файла-сценария);

echo name_file off — выключает режим вывода на экран текста Script-файла;

echo on all — включает режим вывода на экран текста всех m-файлов;

echo off all — выключает режим вывода на экран текста всех m-файлов;

clear — уничтожение определений всех переменных;

clear X Y — уничтожение определений переменных X и Y;

type name_file — выводит на экран текст примеров и m-файлов.

Сохранение рабочей области сессии — команда save. Переменные и определение новых функций в системе Matlab хранятся в особой области памяти, именуемой рабочей областью. Matlab позволяет сохранить значения всех переменных и определений в сессии, т. е. рабочей области, в виде бинарных файлов с расширением **.mat**. Для этого служит команда **save**, которая может использоваться в ряде форм:

save f_name — записывается рабочая область всех переменных в файл бинарного формата с именем **f_name** и расширением **.mat**;

save f_name X — записывает только значение переменной X;

save f_name X Y Z — записывает значение переменных X, Y и Z.

Следует отметить, что возможности сохранения всего текста сессии, формируемой в командном режиме, с помощью команды **save** нет. Для этого служит команда **diary**.

Ведение дневника — команда diary. Сессии не записываются на диск стандартной командой **save**. Однако если такая необходимость есть, можно воспользоваться специальной командой для ведения так называемого *дневника сессии*.

diary file_name — ведет запись на диск в виде текстового файла с указанным именем всех команд в строках ввода и получения результата;

diary off — приостанавливает запись в файл;

diary on — вновь начинает запись в файл.

Таким образом, чередуя команды **diary off** и **diary on**, можно сохранять нужные фрагменты сессии в их формальном виде.

Информационные технологии в менеджменте

Следующий пример в командном режиме поясняет технику применения команды **diary**:

```
>>diary myfile.txt
>>1+2
ans =
3
>>diary off
>> 2+3
ans =
5
>>diary on
>>sin(1)
ans =
0.8415
>>diary off
```

Команда **type** позволяет просматривать текст такого файла со всеми записанными действиями в режиме сессии:

```
>> type myfile.txt
1+2
ans =
3
diary off
sin(1)
ans =
0.8415
diary off
```

Загрузка рабочей области сессии – команда load. Для загрузки рабочей области ранее проведенной сессии (если она была сохранена) можно использовать команду **load**.

load f_name – загрузка ранее сохраненных в файле **fname.mat**.

Если команда **load** используется в ходе проведения сессии, то произойдет замена значений текущих переменных теми значениями, которые были сохранены в считываемом mat-файле.

Следующий пример в командном режиме поясняет технику применения команд **load** и **save**:

```
>> a = [1 2; 3 4]
a =
1 2
3 4
>> b=2+3
b =
5
>> save myfile
```

Для демонстрации работы команды **load** отчистим экран и удалим все переменные:

```
>>clc, clear
>>load myfile
>> a
a =
1 2
```

3 4**>> b****b =****5**

Работа с интерактивной справочной системой. Matlab имеет интерактивную систему помощи, которая реализуется в командном режиме с помощью ряда команд. Одна из них является команда

>>help

которая выводит весь список папок (каталогов), содержащих m-файлы с определениями операторов, функций и иных объектов.

Для получения справки по какому-либо конкретному объекту используется команда

>>help имя

где имя — имя объекта (функции или операции), для которого требуется вывод справочной информации.

Ввиду обилия m-функций в системе Matlab, большое значение имеет поиск m-функций по ключевым словам. Для этого служат команда:

lookfor Ключевое слово

или

lookfor 'Ключевое слово'

В первом случае ищутся все m-файлы, в заголовках которых встречается заданное ключевое слово, а заголовки обнаруженных файлов выводятся на экран. Следует отметить, что широкий поиск по одному ключевому слову может привести к выводу подчас многих десятков определений и длится довольно долго. Для уточнения и сокращения поиска следует использовать вторую форму команды **lookfor**. Ниже приведен пример использования команды **help**:

>> help clc**CLC Clear command window.****CLC clears the command window and homes the cursor.****See also HOME.**

5. Основные элементы языка программирования Matlab

Как и большинство других языков программирования, Matlab предоставляет возможность использования математических выражений, но в отличие от многих из них, эти выражения в Matlab включают матрицы. Основные составляющие выражения:

- переменные;
- числа;
- операторы;
- функции.

Переменные. В Matlab нет необходимости в определении типа переменных или размерности. Когда Matlab встречает новое имя переменной, он автоматически создает переменную и выделяет соответствующий объем памяти. Если переменная уже существует, Matlab изменяет ее состав и если это необходимо выделяет дополнительную память. Например,

num_students = 25

создает матрицу 1x1 с именем *num_students* и сохраняет значение 25 в ее единственном элементе.

Имена переменных состоят из букв, цифр или символов подчеркивания. Matlab использует только первые 31 символ имени переменной. Matlab

Информационные технологии в менеджменте

чувствителен к регистрам, он различает заглавные и строчные буквы. Поэтому A и a — не одна и та же переменная. Чтобы увидеть матрицу, связанную с переменной, просто введите название переменной.

Числа. Matlab использует принятую десятичную систему счисления, с необязательной десятичной точкой и знаками плюс-минус для чисел. Научная система счисления использует букву **e** для определения множителя степени десяти. Мнимые числа используют **i** или **j** как суффикс. Некоторые примеры правильных чисел приведены ниже:

3	-99	0.0001
9.639 7238	1.60210e- 20	6.02252 e23
1i	-3.14159j	3e5i

Все числа для хранения используют формат **long**, это числа с плавающей точкой обладающие ограниченной точностью — приблизительно 16 значащих цифр и ограниченным диапазоном — приблизительно от 10^{-308} до 10^{308} .

Операторы. Выражения используют обычные арифметические операции и правила старшинства (табл. 1).

Таблица 1

Арифметические операции пакета Matlab

Функции	Название	Оператор	Синтаксис
<i>plus</i>	Плюс	+	$M1+M2$
<i>uplus</i>	Унарный плюс	+	$+M$
<i>minus</i>	Минус	-	$M1-M2$
<i>uminus</i>	Унарный минус	-	$-M$
<i>mtimes</i>	Матричное умножение	*	$M1*M2$
<i>times</i>	Почленное умножение массивов	.*	$M1.*M2$
<i>mpower</i>	Возведение в степень матрицы	^	M^x
<i>power</i>	Возведение в степень массива почленное	.^	$M.^x$
<i>mldivide</i>	Обратное (справа налево) деление матриц	\	$M1\M2$
<i>mrdivide</i>	Деление матриц слева направо	/	$M1/M2$
<i>ldivide</i>	Деление почленное массивов справа налево	.\	$M1.\M2$
<i>rdivide</i>	Деление почленное массивов слева направо	./	$M1./M2$

Функции. Matlab предоставляет большое количество элементарных математических функций, таких как **abs**, **sqrt**, **exp**, **sin**. Вычисление квадратного корня или логарифма отрицательного числа не является ошибкой: в этом случае результатом является соответствующее комплексное число. Matlab также предоставляет и более сложные функции, включая Гамма функцию и функции Бесселя. Большинство из этих функций имеют комплексные аргументы. Чтобы вывести список всех элементарных математических функций, наберите:

help elfun

Чтобы посмотреть список всех функций Matlab для анализа данных:

help datafun

Если вам нужно узнать о **Statistics Toolbox**, введите:

help stats

Список элементарных функций представлен в табл. 2.

Таблица 2

Элементарные функции пакета Matlab

Функция	Синтаксис
$ x $ – модуль	<i>abs(x)</i>
e^x – экспонента	<i>exp(x)</i>
$\ln x$ – натуральный логарифм	<i>log(x)</i>
$\log_2 x$ – логарифм по основанию 2	<i>log2(x)</i>
$\lg x$ – десятичный логарифм	<i>log10(x)</i>
2^x – 2 в степени x	<i>pow(x)</i>
\sqrt{x} – квадратный корень	<i>sqrt(x)</i>
<i>arccos x</i> – арккосинус	<i>acos(x)</i>
<i>arcctg x</i> – арккотангенс	<i>acot(x)</i>
<i>arccosec x</i> – арккосеканс	<i>acsc(x)</i>
<i>arcces x</i> – арксеканс	<i>asec(x)</i>
<i>arcsin x</i> – арксинус	<i>asin(x)</i>
<i>arctg x</i> – арктангенс	<i>atan(x)</i>
<i>cos x</i> – косинус	<i>cos(x)</i>
<i>ctg x</i> – котангенс	<i>cot(x)</i>
<i>sec x</i> – секанс	<i>sec(x)</i>
<i>cosec x</i> – косеканс	<i>csc(x)</i>
<i>sin x</i> – синус	<i>sin(x)</i>
<i>tg x</i> – тангенс	<i>tan(x)</i>
<i>arcch x</i> – арккосинус гиперболический	<i>acosh(x)</i>
<i>arccth x</i> – арккотангенс гиперболический	<i>acoth(x)</i>
<i>arccosech x</i> – арккосеканс гиперболический	<i>acsch(x)</i>
<i>arcsech x</i> – арксеканс гиперболический	<i>asech(x)</i>

Информационные технологии в менеджменте

Функция	Синтаксис
$\operatorname{arcsch} x$ – арккосинус гиперболический	$\operatorname{asinh}(x)$
$\operatorname{arctgh} x$ – арктангенс гиперболический	$\operatorname{atanh}(x)$
$\operatorname{ch} x$ – косинус гиперболический	$\operatorname{cosh}(x)$
$\operatorname{ctgh} x$ – котангенс гиперболический	$\operatorname{coth}(x)$
$\operatorname{cosech} x$ – косеканс гиперболический	$\operatorname{csch}(x)$
$\operatorname{sech} x$ – секанс гиперболический	$\operatorname{sech}(x)$
$\operatorname{sh} x$ – синус гиперболический	$\operatorname{sinh}(x)$
$\operatorname{tgh} x$ – тангенс гиперболический	$\operatorname{tanh}(x)$

Логарифм числа b по основанию a : $\log_a b = \frac{\log_c b}{\log_c a}$.

Для вывода более сложных математических и матричных функций, наберите:

help specfun

help elmat

соответственно.

Некоторые функции, такие как **sqrt** и **sin**, — встроенные. Они являются частью Matlab, поэтому они очень эффективны, но их вычислительные детали трудно доступны. В то время как другие функции, такие как **gamma** и **sinh**, реализованы в m-файлах. Поэтому можно увидеть их код и, в случае необходимости, даже модифицировать его.

Несколько специальных функций предоставляют значения часто используемых констант:

pi	Число 3,14159265...
i или j	Мнимая единица $\sqrt{-1}$
eps	Относительная точность числа с плавающей точкой
realmin	Наименьшее число с плавающей точкой
realmax	Наибольшее число с плавающей точкой
Inf	Бесконечность
NaN	Не число (неопределенность вида 0/0 или ∞/∞)

Бесконечность появляется при делении на ноль или при выполнении математического выражения, приводящего к переполнению, т. е. к превышению **realmax**. Не число (**NaN**) генерируется при вычислении выражений типа 0/0 или **Inf/Inf**, которые не имеют определенного математического значения.

Имена функций не являются зарезервированными, поэтому возможно изменять их значения на новые, например:

eps = 1.e-6

и далее использовать это значение в последующих вычислениях. Начальное значение может быть восстановлено следующим образом:

clear eps

Операторы отношения служат для сравнения двух величин, векторов или матриц, все операторы отношения имеют две сравниваемые величины и записываются, как показано в табл. 3.

Операторы отношения пакета Matlab

Функция	Оператор (синтаксис)
Равно	<code>== (x == y)</code>
Не равно	<code>~= (x ~= y)</code>
Меньше	<code>< (x < y)</code>
Больше	<code>> (x > y)</code>
Меньше или равно	<code><= (x <= y)</code>
Больше или равно	<code>>= (x >= y)</code>

Данные операторы выполняют поэлементное сравнение векторов или матриц одинакового размера, и логическое выражение принимает значение 1 (True), если элементы идентичны, и значение 0 (False) в противном случае.

Логические операторы служат для реализации поэлементных логических операций над элементами одинаковых по размеру массивов согласно табл. 4.

Таблица 4

Логические операторы пакета Matlab

Функция	Оператор (синтаксис)
Логическое И	<code>&; and (and (a, b))</code>
Логическое ИЛИ	<code> ; or (or (a, b))</code>
Логическое НЕ	<code>~; not (not (a, b))</code>
Исключающее ИЛИ	<code>xor (xor (a, b))</code>
Верно, если все элементы вектора равны нулю	<code>any (any (a))</code>
Верно, если все элементы вектора не равны нулю	<code>all (all (a))</code>

Текстовые комментарии. Поскольку Matlab используется для достаточно сложных вычислений, большое значение имеет наглядность их описания. Она достигается, в частности, использованием текстовых комментариев. *Текстовые комментарии* вводятся с помощью оператора — символа `%`:

% Ниже представлено задание функции вычисления факториала

Интересные факты:

- Если в командной строке напечатать «why» (почему), MATLAB ответит на этот вопрос. Ответы выбираются из ограниченного множества ответов случайным образом всякий раз, когда вводится эта команда.
- В MATLAB встроена программа виртуальной машины `vrcar`.

Лекция № 11

Matlab: Работа с матрицами

1. Ввод матриц
2. Индексы элементов матриц
3. Матричные функции линейной алгебры
4. Массивы

1. Ввод матриц

Лучший способ начать работу с Matlab — это научиться обращаться с матрицами. В Matlab *матрица* — это прямоугольный массив чисел. Особое значение придается матрицам 1×1 , которые являются скалярами, и матрицам, имеющим один столбец или одну строку, — векторам.

Хороший пример матрицы, можно найти на гравюре времен Ренессанса художника и любителя математики Альбрехта Дюрера (рис. 1, а). Это изображение содержит много математических символов, и если хорошо присмотреться, то в верхнем правом углу можно заметить квадратную матрицу (рис. 1, б). Это матрица известна как магический квадрат и во времена Дюрера считалось, что она обладает магическими свойствами. Она и на самом деле обладает замечательными свойствами, стоящими изучения.



Рис. 1. Гравюра времен Ренессанса художника Альбрехта Дюрера (а), магический квадрат (б)

Матрицы в Matlab можно вводить несколькими способами:

- вводить полный список элементов;
- генерировать матрицы, используя встроенные функции;
- загружать матрицы из внешних файлов;

Информационные технологии в менеджменте

— создавать матрицы с помощью собственных функций в m-файлах.

1. Ввод матриц как списков элементов. Необходимо следовать условиям:

— отделять элементы строки пробелами или запятыми;

— использовать точку с запятой ; для обозначения окончания каждой строки;

— окружать весь список элементов квадратными скобками [].

Чтобы ввести матрицу Дюрера запишем:

A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]

Matlab отобразит матрицу:

```
A =
    16     3     2    13
     5    10    11     8
     9     6     7    12
     4    15    14     1
```

Особые свойства магического квадрата связаны с различными способами суммирования его элементов. Если найти сумму элементов вдоль какой-либо строки или столбца, или вдоль какой-либо из двух главных диагоналей, то получится одно и тоже число:

sum(A) — сумма элементов столбцов матрицы A.

Matlab выдаст ответ:

```
ans =
    34    34    34    34
```

Когда выходная переменная не определена, Matlab использует переменную **ans**, коротко от answer — ответ, для хранения результатов вычисления.

Matlab предпочитает работать со столбцами матрицы, таким образом, лучший способ получить сумму в строках — это транспонировать матрицу, подсчитать сумму в столбцах, а потом транспонировать результат. Операция транспонирования обозначается апострофом. Она зеркально отображает матрицу относительно главной диагонали и меняет строки на столбцы. Таким образом:

```
A'
вызывает:
ans =
    165     9     4
     310     6    15
     211     7    14
     138    12     1
```

A выражение:

sum(A)'

вызывает результат вектор-столбец, содержащий суммы в строках:

```
ans =
    34
    34
    34
    34
```

Сумму элементов на главной диагонали можно получить с помощью функции **diag**, которая выбирает эту диагональ:

```
diag(A)
ans =
    16
    10
```

7

1

A функция:

sum(diag(A))

вызывает:

ans = 34

Другая диагональ, называемая антидиагональю, не так важна математически, поэтому Matlab не имеет специальной функции для нее. Но функция, которая вначале предполагалась для использования в графике, **fliplr**, зеркально отображает матрицу слева направо:

sum(diag(fliplr(A)))

ans = 34

Таким образом, доказано, что матрица на гравюре Дюрера действительно магическая.

Функция magic. Matlab на самом деле обладает встроенной функцией, которая создает магический квадрат почти любого размера. Не удивительно, что эта функция называется **magic**:

B=magic(4)

B =

```

162    3    13
 511   10    8
  97    6   12
   4   14   15    1
    
```

Эта матрица почти та же матрица, что и на гравюре Дюрера, и она имеет все те же магические свойства. Единственное отличие заключается в том, что два средних столбца поменялись местами. Для того чтобы преобразовать *B* в матрицу Дюрера *A*, переставим их местами:

A=B(:,[1 3 2 4])

Это означает, что для каждой строки матрицы *B* элементы переписываются в порядке 1, 3, 2, 4:

A =

```

163    2    13
 510   11    8
   9   67   12
   4  1514    1
    
```

Почему Дюрер переупорядочил столбцы, по сравнению с тем, что использует Matlab? Без сомнения, он хотел включить дату гравюры, 1514, в нижнюю часть магического квадрата.

2. Генерирование матриц с использованием встроенных функций.

Matlab имеет четыре функции, которые создают основные матрицы:

zeros(m, n)	Матрица размерности $m \times n$ с нулевыми элементами
ones(m, n)	Все единицы
rand(m, n)	Равномерное распределение случайных элементов
randn(m, n)	Нормальное распределение случайных элементов

Пример 1.

```

Z = zeros(2,4)

Z =
     0     0     0     0
     0     0     0     0

F = 5*ones(3,3)

F =
     5     5     5
     5     5     5
     5     5     5

R = randn(4,4)

R =
 -0.4326  -1.1465   0.3273  -0.5883
 -1.6656   1.1909   0.1746   2.1832
  0.1253   1.1892  -0.1867  -0.1364
  0.2877  -0.0376   0.7258   0.1139

```

3. Загрузка матриц из внешних файлов. Команда **load** считывает двоичные файлы, содержащие матрицы, созданные в Matlab ранее, или текстовые файлы, содержащие численные данные. Текстовые файлы должны быть сформированы в виде прямоугольной таблицы чисел, отделенных пробелами, с равным количеством элементов в каждой строке. Например, создадим в Блокноте текстовый файл, содержащий 4 строки:

```

16.0    3.0    2.0    13.0
 5.0   10.0   11.0    8.0
 9.0    6.0    7.0   12.0
 4.0   15.0   14.0    1.0

```

Сохраним этот файл под именем **magik.dat**. Тогда команда:
load magik.dat

прочитает этот файл и создаст переменную **magik**, содержащую матрицу.

4. Создание матриц с помощью собственных функций в m-файлах.

Создадим файл, включающий следующие пять строк:

```

A = [16.0    3.0    2.0    13.0
      5.0    10.0   11.0    8.0
      9.0     6.0    7.0  12.0
      4.0    15.0   14.0    1.0];

```

Сохраним его под именем **magik.m**. Тогда выражение
magik

прочитает файл и создаст переменную **A**, содержащую исходную матрицу.

Объединение — это процесс соединения маленьких матриц для создания больших. Пара квадратных скобок — это оператор объединения. Например, начнем с матрицы **A** (магического квадрата 4×4) и сформируем:

```

B = [A A+32; A+48 A+16]

```

Результатом будет матрица 8×8 , получаемая соединением четырех подматриц:

Информационные технологии в менеджменте

B =

16	2	3	13	48	34	35	45
5	11	10	8	37	43	42	40
9	7	6	12	41	39	38	44
4	14	15	1	36	46	47	33
64	50	51	61	32	18	19	29
53	59	58	56	21	27	26	24
57	55	54	60	25	23	22	28
52	62	63	49	20	30	31	17

2. Индексы элементов матриц

Элемент в строке **i** и столбце **j** матрицы **A** обозначается **A(i,j)**. Например, **A(4,2)** — это число в четвертой строке и втором столбце. Для нашего магического квадрата **A(4,2) = 15**. Таким образом, можно вычислить сумму элементов в четвертом столбце матрицы **A**, набрав:

A(1,4) + A(2,4) + A(3,4) + A(4,4)

получим

ans = 34

Однако это не самый лучший способ суммирования отдельной строки. Также возможно обращаться к элементам матрицы через один индекс, **A(k)**. В этом случае массив рассматривается как длинный вектор, сформированный из столбцов исходной матрицы.

Так, для магического квадрата, **A(8)** — это другой способ сослаться на значение 15, хранящееся в **A(4,2)**.

Если использовать значение элемента вне матрицы, Matlab выдаст ошибку:

t=A(4,5)

??? Index exceeds matrix dimensions.

С другой стороны, если сохранить значение вне матрицы, то размер матрицы увеличивается:

X=A;

X(4,5) = 17

X =

163	2	13	0
510	11	8	0
96	7	12	0
415	14	1	17

Оператор двоеточия. Двоеточие **:** — это один из наиболее важных операторов Matlab. Он проявляется в различных формах. Выражение:

1:10

— это вектор-строка, содержащая целые числа от 1 до 10:

1 2 3 4 5 6 7 8 9 10

Для получения обратного интервала:

100:-7:50

что дает:

100 93 86 79 72 65 58 51

или:

0:pi/4:pi

что приводит к:

0 0.7854 1.5708 2.3562 3.1416

Информационные технологии в менеджменте

Индексное выражение, включая двоеточие, относится к части матрицы: **A(1:k, j)** это первые k элементов j -го столбца матрицы A . Так:

sum(A(1:4,4))

вычисляет сумму четвертой строки. Но есть и лучший способ. Двоеточие, само по себе, обращается ко всем элементам в строке и столбце матрицы, а слово **end** — к последней строке или столбцу. Так:

sum(A(:,end))

вычисляет сумму элементов в последнем столбце матрицы A :

ans = 34

Почему магическая сумма квадрата 4×4 равна 34? Если целые числа от 1 до 16 отсортированы в четыре группы с равными суммами, эта сумма должна быть:

sum(1:16)/4

которая равна:

ans = 34

3. Матричные функции линейной алгебры

Математические операции, определенные на матрицах являются объектом линейной алгебры (табл. 1).

Таблица 1

Матричные функции линейной алгебры

det(A) — возвращает определитель квадратной матрицы A .
rank(A) — возвращает ранг матрицы A .
trace(A) — возвращает след матрицы A .
inv(A) — возвращает матрицу, обратную квадратной матрицы A .
size(A) — возвращает вектор-строку, содержащую количество строк и столбцов в массиве A .
length(X) — возвращает длину вектора X .
max(A) — возвращает вектор-строку, содержащую значения максимальных элементов в столбцах матрицы A .
min(A) — возвращает вектор-строку, содержащую значения минимальных элементов в столбцах матрицы A .

4. Массивы

Когда мы выходим из мира линейной алгебры, матрицы становятся *двумерными численными массивами*. Арифметические операции на массивах производятся поэлементно. Это означает, что суммирование и вычитание являются одинаковыми операциями для матриц и массивов, а умножение для них различно. Matlab использует точку, как часть записи для операции умножения массивов.

Список операторов включает в себя:

Информационные технологии в менеджменте

+	суммирование
-	вычитание
.*	поэлементное умножение
./	поэлементное деление
.\	поэлементное левое деление
.^	поэлементное возведение в степень
.'	несопряженное матричное транспонирование

Если магический квадрат Дюрера умножить на себя по правилам умножения массивов:

A.*A

результатом будет массив, содержащий квадраты целых чисел от 1 до 16:

ans =

```

2569    4    169
25100 121    64
8136   49   144
16225 196    1
    
```

Операции над массивами полезны для создания таблиц. Пусть n — это вектор-столбец:

n = (0:9)';

Тогда:

pows = [n n.^2 2.^n]

создает таблицу квадратов и степеней двойки:

```

pows =
    0     0     1
    1     1     2
    2     4     4
    3     9     8
    4    16    16
    5    25    32
    6    36    64
    7    49   128
    8    64   256
    9    81   512
    
```

Элементарные математические функции работают с массивами поэлементно.

Так:

x = (1:0.1:2)'

s = [x sin(x)]

создает таблицу значений синуса.

Лекция № 12

Matlab: Элементы графики

1. Создание графика
2. Окна изображений. Добавление кривых на существующий график. Подграфики
3. Управление осями. Подписи к осям и заголовки
4. Трехмерные графики

1. Создание графика

Matlab имеет широкие возможности для графического изображения векторов и матриц, а также для создания комментариев и печати графики.

Функция **plot** имеет различные формы, связанные с входными параметрами, например **plot(y)** создает кусочно-линейный график зависимости элементов y от их индексов. Если вы задаете два вектора в качестве аргументов, **plot(x,y)** создаст график зависимости y от x .

Например, построение графика значений функции **sin** от 0 до 2π (рис. 1):

```
t = 0:pi/100:2*pi;  
y = sin(t);  
plot(t,y)
```

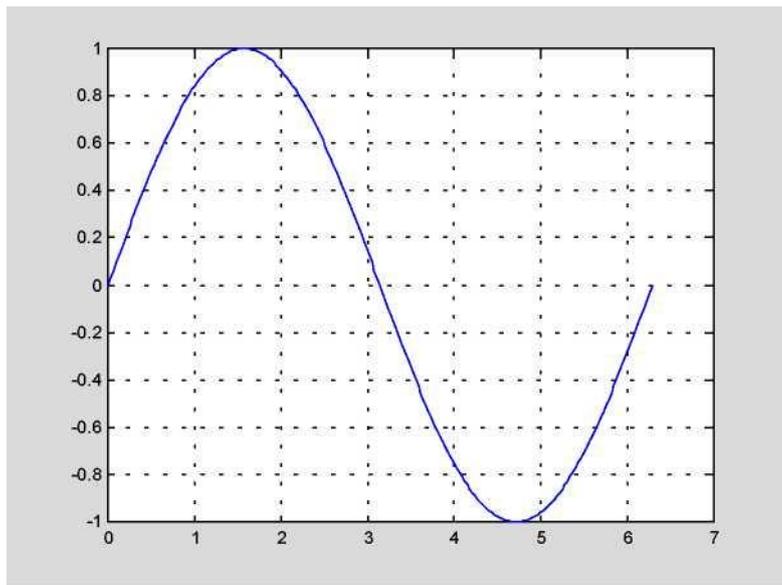


Рис. 1.

Вызов функции **plot** с многочисленными парами x - y создает многочисленные графики. Matlab автоматически присваивает каждому графику свой цвет (исключая случаи, когда это делает пользователь), что позволяет различать заданные наборы данных. Например, следующие три строки отображают график близких функций, и каждой кривой соответствует свой цвет (рис. 2):

```
y2 = sin(t-.25);  
y3 = sin(t-.5);  
plot(t, y, t, y2, t, y3)
```

Информационные технологии в менеджменте

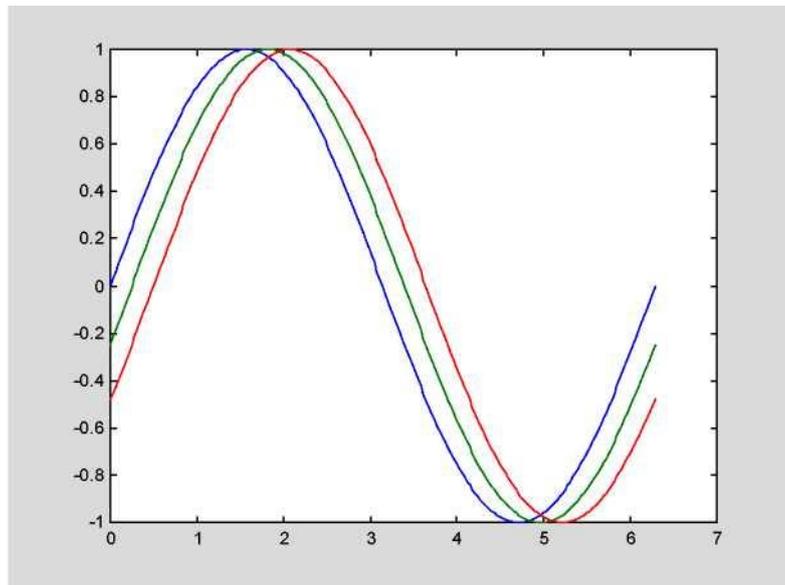


Рис. 2.

Функция **semilogx(x,y)** строит график $y(x)$ в логарифмическом масштабе по оси x , векторы x и y должны иметь одинаковые размеры.

Возможно изменение цвета, стиля линий и маркеров, таких как знаки плюс или кружки, следующим образом (табл. 1):

plot(x, y, 'цвет стиль маркер')

цвет стиль маркер это 1-, 2-, 3- символьная строка (заклученная в одинарные кавычки), составленная из типов цвета, стиля линий и маркеров:

— символы, относящие к цвету: 'c', 'm', 'y', 'r', 'g', 'b', 'w' и 'k'. Они обозначают голубой, малиновый, желтый, красный, зеленый, синий, белый и черный цвета соответственно;

— символы, относящиеся к типу линий: '-' для сплошной, '--' для разрывной, ':' для пунктирной, '-.' для штрихпунктирной линий и 'none' для её отсутствия;

— наиболее часто встречающиеся маркеры '+', 'o', '*' и 'x'.

Таблица 1

Стили графиков

Цвет линии	Тип точки	Тип линии
$s = y$ – желтый	$s = \bullet$ – точка	$s = -$ – сплошная
m – фиолетовый	\circ – окружность	$:$ – двойной пунктир
c – голубой	x – крест	$-. .$ – штрихпунктир
r – красный	$+$ – плюс	$--$ – штриховая
g – зеленый	$*$ – звездочка	
b – синий	s – квадрат	
w – белый	d – ромб	
k – черный	\vee – треугольник (вниз)	
	\wedge – треугольник (вверх)	
	$<$ – треугольник (влево)	
	$>$ – треугольник (вправо)	
	p – пятиугольник	
	h – шестиугольник	

Например, выражение:

plot(x,y,'y:+')

строит желтый пунктирный график и помещает маркеры '+' в каждую точку данных. Если вы определяете только тип маркера, но не определяете тип стиля линий, то Matlab выведет только маркеры.

Графики дискретных отсчетов функции

В сопротивлении материалов и строительной механике принято графики параметров напряженно-деформированного состояния закрашивать (штриховать). Для построения подобных графиков используется команда *stem(...)*.

stem(x, y) – строит график с закрашенными частями между нулевой линией и кривой $y = f(x)$.

stem(... , 'LINESPEC ') – аналогична команде *stem(x, y)*, но имеет спецификацию линий 'LINESPEC', подобную строковой константе *s* команды *plot(x, y, s)*.

Stem(... , 'filled ') – строит график функции $y = f(x)$ с закрашенными маркерами.

Мнимые и комплексные данные. Если аргумент функции **plot** комплексное число, то мнимая часть игнорируется, за исключением случая, когда комплексный аргумент один. Для этого специального случая происходит построение графика зависимости реальной части аргумента от мнимой. Поэтому **plot(Z)**, где *Z* – комплексный вектор или матрица, эквивалентно:

plot(real(Z),imag(Z))

Например:

t = 0:pi/10:2*pi;

plot(exp(i*t),'-o')

отобразит двадцатисторонний многоугольник с кружками на вершинах (рис. 5).

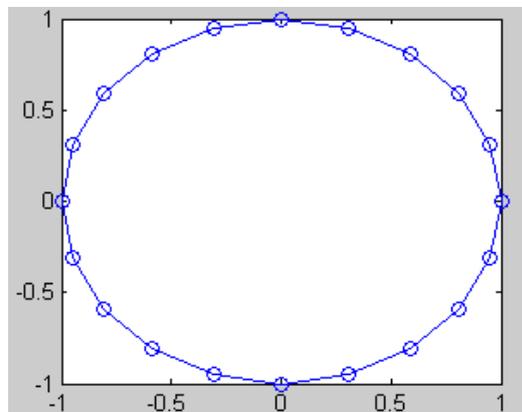


Рис. 5

2. Окна изображений. Добавление кривых на существующий график. Подграфики

Окна изображений. Функция **plot** автоматически открывает новое окно изображения (далее окно), если до этого его не было на экране. Если же оно существует, то **plot** использует его по умолчанию. Для открытия нового окна и выбора его по умолчанию, наберите:

figure

Для того чтобы сделать существующее окно текущим:

figure(n)

где n — это номер в заголовке окна. В этом случае результаты всех последующих команд будут выводиться в это окно.

Добавление кривых на существующий график. Команда **hold** позволяет добавлять кривые на существующий график. Когда вы набираете:

hold on

Matlab не стирает существующий график, а добавляет в него новые данные, изменяя оси, если это необходимо. Например, следующий элемент кода вначале создает график функции **sin(x)**, а затем накладывает на него график функции **sin(2*x)** (рис. 3):

```
x=0:pi/100:4*pi;
```

```
plot(x,sin(x),'r')
```

```
hold on
```

```
plot(x,sin(2*x),'b')
```

```
grid on
```

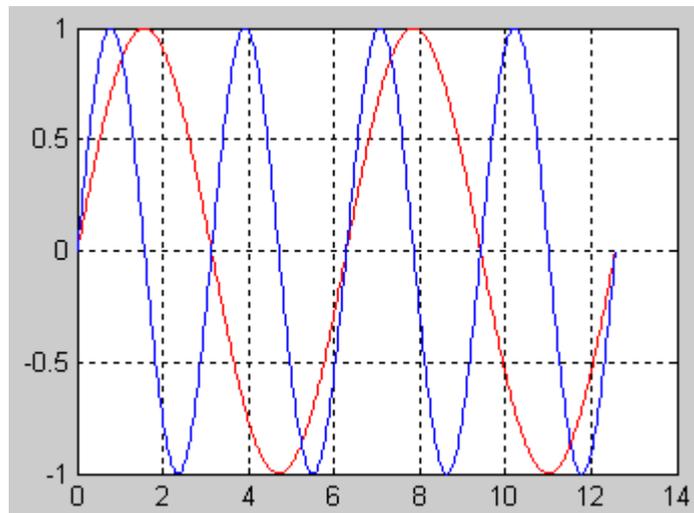


Рис. 3.

Подграфики. Функция **subplot** позволяет выводить множество графиков в одном окне:

subplot(m,n,p)

разбивает окно изображений на матрицу m на n подграфики и выбирает p -ый подграфик текущим. Графики нумеруются вдоль первого в верхней строке, потом во второй и т. д. Например, для того, чтобы представить графические данные в четырех разных подобластях окна необходимо выполнить следующее (рис. 4):

```
x=0:pi/100:4*pi;
```

```
subplot(2,2,1)
```

```
plot(x,sin(x),'r')
```

```
grid on
```

```
subplot(2,2,2)
plot(x,sin(2*x),'b')
grid on
```

```
subplot(2,2,3)
plot(x,sin(4*x),'m')
grid on
```

```
subplot(2,2,4)
plot(x,sin(6*x),'g')
grid on
```

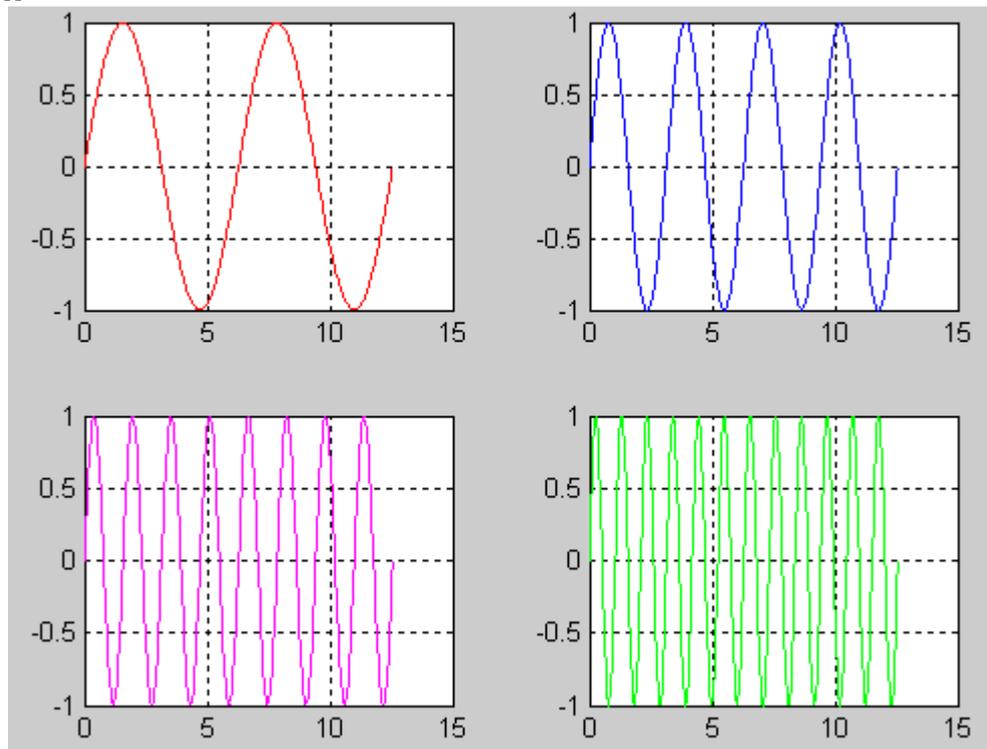


Рис. 4.

3. Управление осями. Подписи к осям и заголовки

Функция **axis** имеет несколько возможностей для настройки масштаба, ориентации и коэффициента сжатия.

Обычно Matlab находит максимальное и минимальное значение и выбирает соответствующий масштаб и маркирование осей. Функция **axis** заменяет значения по умолчанию предельными значениями, вводимыми пользователем:

```
axis( [xmin xmax ymin ymax] )
```

В функции **axis** можно также использовать ключевые слова для управления внешним видом осей. Например:

```
axis square
```

создает x и y оси равной длины, а:

```
axis equal
```

создает отдельные отметки приращений для x и y осей одинаковой длины.

Так функция:

```
plot(exp(i*t))
```

следующая либо за **axis square**, либо за **axis equal** превращает овал в правильный круг.

axis auto

возвращает значения по умолчанию и переходит в автоматический режим.

axis on

включает обозначения осей и метки промежуточных делений.

axis off выключает обозначения осей и метки промежуточных делений.

grid off выключает сетку координат, а

grid on включает ее заново.

Подписи к осям и заголовки. Функции **xlabel('имя оси')**, **ylabel('имя оси')**, **zlabel('имя оси')** добавляют подписи к соответствующим осям, функция **title('название графика')** добавляет заголовок в верхнюю часть окна, а функция **text** вставляет текст в любое место графика.

4. Трехмерные графики

Трехмерные поверхности описываются функцией двух переменных $Z = f(X, Y)$. Функции Matlab, предназначенные для построения трехмерных графиков, требуют определения X и Y как матриц, состоящие из повторяющихся строк и столбцов.

Для создания таких матриц служит функция **meshgrid**, которая преобразует область определения, заданную через один вектор или два вектора x и y , в матрицы X и Y . Строки матрицы X дублируют вектор x , а столбцы Y — вектор y .

Функция записывается следующим образом:

[X,Y]=meshgrid(x,y)

Пример:

```
>> [X Y] = meshgrid(4 : 7, 9 : 13)
    X
      4  5  6  7
      4  5  6  7
      4  5  6  7
      4  5  6  7
      4  5  6  7
    Y
      9  9  9  9
     10 10 10 10
     11 11 11 11
     12 12 12 12
     13 13 13 13
```

Приведем еще пример применения функции *meshgrid*:

```
>> [X Y] = meshgrid(-1 : 0.1 : 1, -1 : 0.1 : 1);
```

Такой вызов функции создает опорную плоскость для построения трехмерной поверхности при изменении x и y от -1 до 1 с шагом 0.1 .

Функции **mesh** и **surface** отображают поверхность в трех измерениях. При этом **mesh** создает каркасную поверхность, где цветные линии соединяют только заданные точки, а функция **surface** вместе с линиями отображает в цвете и саму поверхность.

Информационные технологии в менеджменте

Для вычисления двумерной функции $\sin(R)/R$, в области x - y поступают следующим образом (рис. 6):

```
[X, Y] = meshgrid(-8:5:8);  
R = sqrt(X.^2+Y.^2)+eps;  
Z = sin(R)./R;  
mesh(X,Y,Z)
```

Здесь X , Y , Z — это матрицы значений координат поверхности.

В этом примере R — это расстояние от начала координат, которому соответствует центр матрицы. Добавление eps позволяет избежать неопределенности $0/0$ в начале координат.

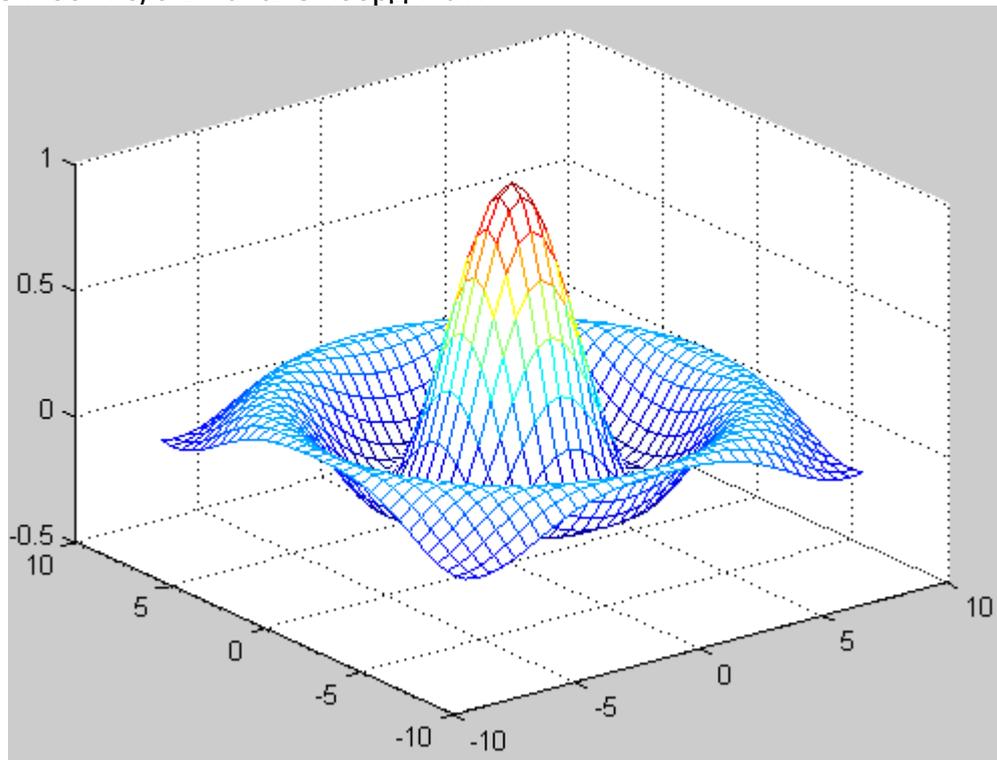


Рис. 6.

При использовании функции (рис. 7):

```
surface(X,Y,Z)
```

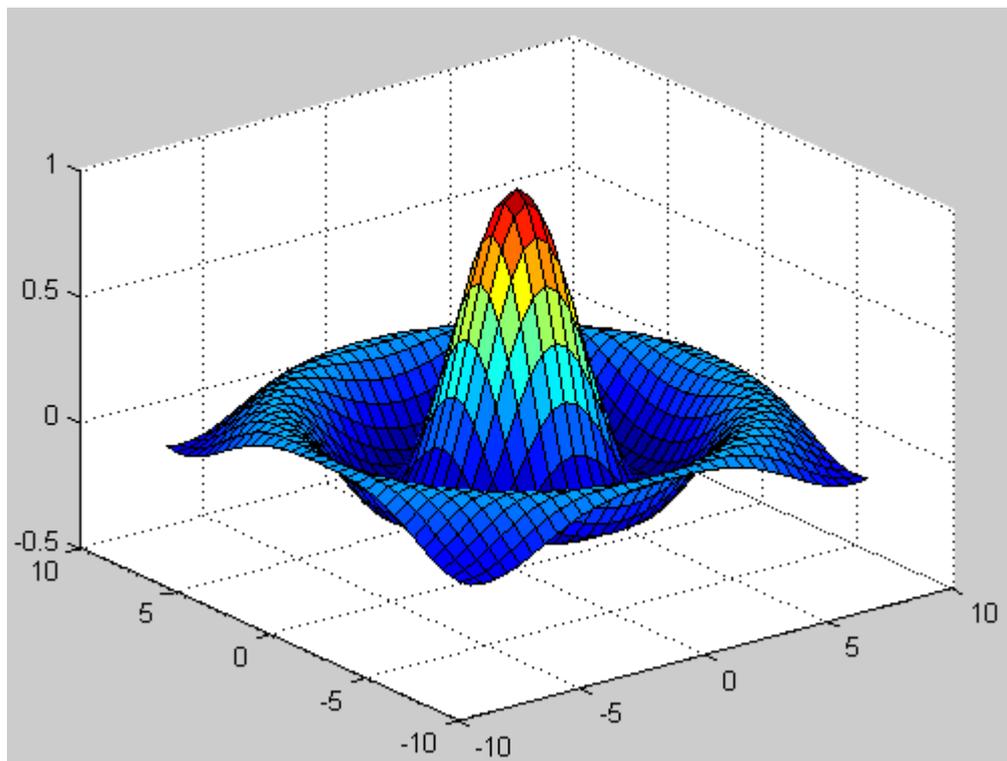


Рис. 7.

Для построения графика в трехмерном пространстве можно использовать функцию `plot3(x,y,z)` (рис. 8):

```
t=0:0.1:30;  
x=0.1*t.*cos(t);  
y=0.1*t.*sin(t);  
z=0.05*t;  
plot3(x,y,z);  
grid on
```

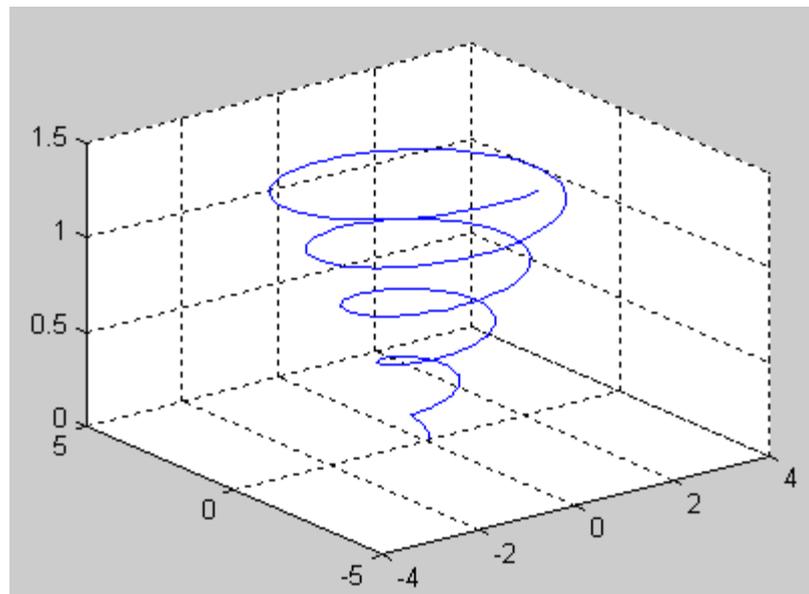


Рис. 8.

В этом случае x , y , z — это вектора значений координат функции в пространстве.

Лекция № 13

Matlab: Символьные вычисления

1. Создание символьных переменных и выражений

2. Символьные операции с выражениями

Система Matlab является самой крупной системой компьютерной математики, ориентированной на матричные и численные вычисления. Однако Matlab имеет также и средства аналитических вычислений. Пакет **Symbolic Math Toolbox** добавил системе Matlab качественной новые возможности, связанные с выполнением символьных вычислений и преобразований.

С помощью команды:

```
>> help symbolic
```

Можно получить перечень входящих в пакет команд и функций. Для получения справки по любой команде или функции можно использовать команду:

```
>> help sym / name.m,
```

где **name** — имя соответствующей команды или функции, а **name.m** — имя m-файла, задающего данную команду или функцию.

1. Создание символьных переменных и выражений

Команда **sym** создает символьную переменную и/или выражение. Например, выполнение команды:

```
x=sym('x')
```

приведет к созданию символьной переменной x .

Если необходимо использовать символьное обозначение для выражения:

$$h = \frac{1 + \sqrt{5}}{2},$$

тогда команда будет выглядеть:

```
h=sym('(1+sqrt(5))/2');
```

Теперь можно использовать h в различных математических выражениях.

Например:

```
f=h^2-h-1
```

вернет:

```
f = (1/2+1/2*5^(1/2))^2-3/2-1/2*5^(1/2)
```

Выражение:

```
f=sym('a*x^2+b*x+c')
```

придаст символьное выражение $ax^2 + bx + c$ переменной f . Но и только, **Symbolic Math Toolbox** не создаст переменных, используемых в данном выражении, а именно a, b, c, x . Для дальнейшего осуществления математических операций (таких как интегрирование, дифференцирование, упрощение и т. д.) над функцией f , необходимо вручную создать эти переменные:

```
a=sym('a');
```

```
b=sym('b');
```

```
c=sym('c');
```

```
x=sym('x');
```

или

```
syms a b c x;
```

Рекомендуется использовать **syms**, что позволит избежать создания избыточных кодов.

Функция создания списка символьных переменных *findsym*

В математических выражениях могут использоваться как обычные, так и символьные переменные. Функция *findsym* позволяет выделить символьные переменные в составе выражения S:

- *findsym*(S) — возвращает в алфавитном порядке список всех символьных переменных выражения S. При отсутствии таковых возвращается пустая строка.

Примеры:

```
» a =2; b = 4 ;
» findsym (a*x^2 + b*y + z)
ans =
x, y, z
» findsym(a + b + x + y + z, 2)
ans =
x, y, z
```

2. Символьные операции с выражениями

В Matlab имеется несколько функций для преобразования одной формы представления функции к другой, возможно более простой (табл. 1).

Таблица 1

Символьные операции

<i>Вид преобразования</i>	<i>Ключевое слово</i>	<i>Описание</i>
Приведение подобных слагаемых	collect	Приведение подобных слагаемых полинома
Расширение	expand	Раскрываются суммы и произведения, тригонометрические функции разлагаются по тригонометрическим тождествам
Разложение на множители	factor	Производится разложение выражений или чисел на простые множители.
Упрощение	simplify	Приводит выражение к более простому виду
Обеспечение подстановок	subs	

Эти функции в качестве входного параметра могут принимать символьную матрицу. В таком случае выполнение операции происходит поэлементно.

Функция **collect(f)** представляет *f* как полином от символьной переменной, например *x*, и собирает коэффициенты при степенях *x*. Второй аргумент определяет коэффициенты какой переменной необходимо собрать, если в выражении более одной переменной. Примеры:

f	collect(f)
$(x-1)*(x-2)*(x-3)$	$x^3-6*x^2+11*x-6$
$x*(x*(x-6)+11)-6$	$x^3-6*x^2+11*x-6$
$(1+x)*t + x*t$	$2*x*t+t$

Функция **expand(f)** расширяет выражение **f**. Рациональные выражения она раскладывает на простые дроби, полиномы — на полиномиальные разложения и т.д. Примеры:

f	expand(f)
$a*(x + y)$	$a*x+a*y$
$(x-1)*(x-2)*(x-3)$	$x^3-6*x^2+11*x-6$
$x*(x*(x-6)+11)-6$	$x^3-6*x^2+11*x-6$
$\exp(a+b)$	$\exp(a)*\exp(b)$
$\cos(x+y)$	$\cos(x)*\cos(y)-\sin(x)*\sin(y)$
$\cos(3*\arccos(x))$	$4*x^3-3*x$

Функция **factor(f)** разлагает выражение **f** на простые множители, а целые числа — на произведение простых чисел. Если **f** не может быть разложен, то результатом выполнения будет само выражение **f**. Примеры:

f	factor(f)
$x^3-6*x^2+11*x-6$	$(x-1)*(x-2)*(x-3)$
$x^3-6*x^2+11*x-5$	$x^3-6*x^2+11*x-5$
x^6+1	$(x^2+1)*(x^4-x^2+1)$

Функция **simplify(f)** наиболее мощное средство, она предназначена для алгебраических преобразований, сверток степеней, преобразования тригонометрических, экспоненциальных и логарифмических функций.

Примеры:

```

» syms a b x;
» V = [sin(x)^2 + cos(x)^2 log(a*b)]
» simplify(V)
ans =
    [ 1, log(a*b)]
» simplify((a^2 - 2*a*b + b^2) / (a - b))
ans =
    a - b
    
```

Функция **simple(f)** использует все правомерные математические способы упрощения для нахождения варианта с наименьшей длиной символьной строки результата. Функция добивается такого результата путем последовательного

Информационные технологии в менеджменте

применения **simplify**, **collect**, **factor**, и других функций упрощения к выражению и сохранением информации о длине строки результата. Возвращает функция наименьший результат (тот ответ, строка которого имеет наименьшую длину).

Функция **simple** имеет несколько форм записи:

simple(f)

возвращает наименьший результат. Например, выражение:

f = simple(cos(x)^2+sin(x)^2)

вернет **f = 1**.

Если вы хотите знать каким из методов был получен этот результат, то необходимо использовать вторую форму записи:

[F, how] = simple(f)

В данном виде функция вернет в качестве *F* наикратчайший результат, а в качестве *how* — название метода. Например:

[f, how] = simple(cos(x)^2+sin(x)^2)

вернет **f = 1**

how = combine

Одной из самых эффективных и часто используемых операций символьной математики является операция подстановки. Она реализуется функцией **subs**, имеющей ряд форм записи:

subs(S) — заменяет в символьном выражении **S** все переменные их символьными значениями, которые берутся из вычисляемой функции или рабочей области системы MATLAB.

subs(S, NEW) — заменяет все свободные символьные переменные в **S** из списка **NEW**.

subs(S, OLD, NEW) — заменяет **OLD** на **NEW** в символьном выражении **S**. При одинаковых размерах массивов **OLD** и **NEW** замена идет поэлементно. Если **S** и **OLD** — скаляры, а **NEW** — числовой массив или массив ячеек, то скаляры расширяются до массива результатов.

Примеры:

>>syms a b x y;

>>subs(x-y,y,1)

ans =

x-1

>>subs(sin(x)+cos(y),[x y],[a b])

ans =

sin(a)+cos(b)

В задачах аппроксимации и приближения функций $f(x)$ важное место занимает их разложение в ряд Тейлора в окрестности точки a :

$$f(x) = \sum_{n=0}^{\infty} (x - a)^n \frac{f^n(a)}{n!}.$$

Частным случаем этого ряда при $a = 0$ является ряд Маклорена:

$$f(x) = \sum_{n=0}^{\infty} x^n \frac{f^n(0)}{n!}.$$

Функция разложения в ряд Тейлора – *taylor*

- `taylor(f)` – возвращает шесть членов ряда Маклорена функции f ;
- `taylor(f, n, x, a)` – возвращает n членов ряда Тейлора в точке $x = a$;
- `taylor(f, n)` – возвращает $(n - 1)$ членов ряда Маклорена функции f ;
- `taylor(f, a)` – возвращает шесть членов ряда Тейлора функции f в точке a .

Примеры:

```
»x = sym('x')
»taylor(sin(x))
ans =
  x - 1/6*x^3 + 1/120*x^5
»taylor(int(sin(x)))
ans =
  - 1 + 1/2*x^2 - 1/24*x^4
```

Функция вычисления пределов – *limit*

Для вычисления пределов аналитически заданной функции $F(x)$ служит функция *limit*

- `limit(F, x, a)` – возвращает предел символьного выражения F в точке $x = a$.

Пример:

```
»limit(sin(x)/x, x, 0)
ans =
  1
```

Функция вычисления суммы рядов – *symsum*

Для аналитического вычисления суммы ряда служит команда `symsum`:

- `symsum(S)` — возвращает символьное значение суммы бесконечного ряда по переменной, найденной автоматически с помощью функции `findsym`;
- `symsum(S, v)` — возвращает сумму бесконечного ряда по переменной v ;
- `symsum(S, a, b)` и `symsum(S, v, a, b)` — возвращают конечную сумму ряда в пределах номеров слагаемых от a до b .

Примеры:

```
» x = sym('x');
» symsum(x^2)
ans =
  1/3*x^3-1/2*x^2+1/6*x
» symsum([x, x^2, x^3], 1, 5)
ans =
  [ 15, 55, 225]
```

Лекция № 14

Matlab: Дифференцирование и интегрирование

1. Дифференцирование
2. Интегрирование

1. Дифференцирование

Аналитическое дифференцирование. Для вычисления производных в символьном виде от выражения f служит функция **diff(f)**.

Создадим символьное выражение:

```
syms a x;
```

```
f = sin(a*x);
```

тогда **diff(f)** приведет к дифференцированию f по ее символьной переменной (в данном случае это x):

```
ans = cos(a*x)*a
```

Для взятия производной по переменной a , явно укажем: **diff(f,a)**. Данное выражение вернет df/da :

```
ans = cos(a*x)*x
```

Чтобы найти производные более высоких порядков, необходимо указать порядок производной. Например, производная второго порядка:

```
diff(f,2); или diff(f,x,2);
```

```
ans = -sin(a*x)*a^2
```

и

```
diff(f,a,2);
```

```
вернет ans = -sin(a*x)*x^2
```

Функция **diff** в качестве входного параметра может принимать символьную матрицу. В таком случае дифференцирование происходит поэлементно:

```
syms a x;
```

```
A = [cos(a*x),sin(a*x);-sin(a*x),cos(a*x)];
```

Т. е. матрица A имеет вид:

```
A =
```

```
[ cos(a*x), sin(a*x)]
```

```
[-sin(a*x), cos(a*x)]
```

Тогда результат команды **diff(A)** будет:

```
ans =
```

```
[-sin(a*x)*a, cos(a*x)*a]
```

```
[-cos(a*x)*a, -sin(a*x)*a]
```

Численное дифференцирование. Численное дифференцирование строится на использовании аппарата конечных разностей и соответствующего многообразия аппроксимаций. Используются функции:

diff(X) — для вектора X вычисление первых конечных разностей $[X(2)-X(1), X(3)-X(2), \dots, X(n)-X(n-1)]$, а для матрицы X вычисление первых конечных разностей между столбцами $[X(2:n,:) - X(1:n-1,:)]$;

diff(X, n) — вычисление конечных разностей n -го порядка;

diff(X, n, dim) — вычисление конечных разностей n -го порядка по указанному измерению.

2. Интегрирование

Аналитическое интегрирование. Функция **int(f,x)** аналитически вычисляет интеграл от функции f по символьной переменной x :

```
syms x;  
int(sin(x)./x.^2,x)  
ans =  
-sin(x)/x+cosint(x)
```

Функция **int(f,x,a,b)** численно вычисляет интеграл от функции f по символьной переменной x в границах a,b :

```
int(sin(x)./x.^2,x,0.1,pi)  
ans =  
cosint(pi)+10*sin(1/10)-cosint(1/10)  
>> cosint(pi)+10*sin(1/10)-cosint(1/10)  
ans =  
2.7999
```

Численное интегрирование. **Численное интегрирование в Matlab выполняется с помощью функций:**

trapz — **интегрирование методом трапеций;**

quad, quad8 — **вычисление интегралов методом квадратур.**

Интегрирование методом трапеций.

Здесь используется линейная аппроксимация, т. е. график функции $y = f(x)$ представляется в виде ломаной, соединяющей точки y_i . Формула трапеций

при постоянном шаге $h = \frac{b-a}{n}$, где n – число участков, имеет вид

$$\int_a^b f(x)dx = h \left(\frac{y_0 + y_n}{2} + \sum_{i=1}^{n-1} y_i \right). \quad (2.8)$$

В Matlab данную формулу реализует функция **trapz(x, y)**, которая вычисляет интеграл от функции y по переменной x , используя метод трапеций. Аргументы x и y могут быть одномерными массивами одинакового размера, либо массив y может быть двумерным, но тогда должно выполняться условие $\text{size}(y, 1) = \text{length}(x)$. В последнем случае вычисляется интеграл для каждого столбца.

Функция **trapz(y)** вычисляет интеграл, предполагая, что шаг интегрирования постоянен и равен единице; в случае, когда шаг h отличен от единицы, но постоянен, достаточно вычисленный интеграл умножить на h .

Пример 1. Вычислим интеграл: $I = \int_0^{\pi} \sin(x)dx$. Его точное значение равно

двум.

Выберем равномерную сетку:

```
x = 0:pi/100:pi;
```

```
y = sin(x);
```

тогда оба интеграла

```
I = trapz(x, y)
```

и

```
I = pi/100*trapz(y)
```

дают одинаковый результат:

I = 1.9998

Образует неравномерную сетку, используя генератор случайных чисел:

x = sort(rand(1,101)*pi);

y = sin(x);

I = trapz(x, y)

I = 1.9987

Результат еще менее точен, поскольку максимальный из шагов **max(diff(x))** равен 0,1810.

Вычисление интегралов методом квадратур. Квадратура — это численный метод вычисления площади под графиком функции, то есть вычисление определенного интеграла вида:

$$I = \int_a^b f(x)dx.$$

Функции **quad** и **quad8** используют разные квадратурные формулы. Функция **quad** использует квадратурные формулы Ньютона-Котеса 2-го порядка (правило Симпсона), а функция **quad8** — формулы 8-го порядка. При наличии в подынтегральной функции особенностей типа:

$$I = \int_0^1 \sqrt{x}dx,$$

предпочтительнее использовать процедуру **quad8**.

Функции **[I, cnt] = quad('имя функции', a, b)** и **[I, cnt] = quad8('имя функции', a, b)** вычисляют интеграл от заданной функции. Переменная **cnt** содержит информацию о том, сколько раз в процессе интегрирования вычислялась подынтегральная функция.

Функции **[I, cnt] = quad('имя функции', a, b, tol)** и **[I, cnt] = quad8('имя функции', a, b, tol)** вычисляют интеграл с заданной относительной погрешностью **tol**. По умолчанию **tol = 1e-3**.

Функции **[I, cnt] = quad('имя функции', a, b, tol, trace)** и **[I, cnt] = quad8('имя функции', a, b, tol, trace)**, когда аргумент **trace** не равен нулю, строят график, показывающий ход вычисления интеграла.

Пример 2. Вычислим интеграл: $I = \int_0^{\pi} \sin(x)dx$. Его точное значение равно

двум.

[I, cnt] = quad('sin', 0, pi)	[I, cnt] = quad('sin', 0, pi, 1e-4, 1)
I = 2.0000	I = 2.0000
cnt = 17	cnt = 33

Как следует из анализа полученных данных, при увеличении точности вычисления на порядок почти вдвое увеличивается трудоемкость расчетов. График с результатами промежуточных расчетов показан на [рис. 1](#).

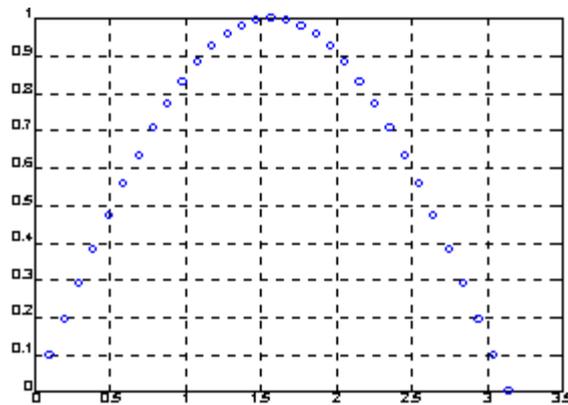


Рис. 1.

Функции **quad** и **quad8** используют рекурсивный вызов. Для того чтобы предотвратить бесконечную рекурсию при вычислении сингулярных интегралов, глубина рекурсии ограничена уровнем 10. При достижении этого ограничения выдается сообщение: «Recursion level limit reached in quad. Singularity likely» (В процедуре **quad** достигнута предельная глубина рекурсии. Функция, возможно, сингулярная).

Функции **quad** и **quad8** не позволяют интегрировать функции с особенностями типа:

$$I = \int_0^1 \frac{1}{\sqrt{x}} dx.$$

В этом случае рекомендуется выделить такие члены и проинтегрировать их аналитически, а к остатку применить процедуры **quad** и **quad8**.

Лекция № 15

Matlab: Типовые средства программирования

1. Назначение языка программирования Matlab
2. Управляющие структуры
3. Векторизация циклов

1. Назначение языка программирования Matlab

Практически невозможно предусмотреть в одной, даже самой большой и мощной математической системе возможность решения всех задач, которые могут интересовать пользователя. *Программирование* в системе MATLAB является эффективным средством ее расширения и адаптации к решению специфических проблем. Оно реализуется с помощью *языка программирования* системы.

Большинство объектов этого языка, в частности все команды, операторы и функции, одновременно являются объектами *входного языка* общения с системой в командном режиме работы.

Так в чем же отличие входного языка от языка программирования? В основном — в способе фиксации создаваемых ими кодов. Сессии в командном режиме работы не сохраняются в памяти компьютера (ведение дневника не в счет). Хранятся только определения созданных в ходе их выполнения переменных и функций. А вот программы на языке программирования MATLAB сохраняются в виде текстовых *m-файлов*. При этом могут сохраняться как целые программы в виде файлов-сценариев, так и отдельные полноценные *программные модули* —

функции. Кроме того, важно, что программа может менять структуру алгоритмов вычислений в зависимости от входных данных и данных, создаваемых в ходе вычислений.

С позиций программиста язык программирования системы является типичным *проблемно-ориентированным* языком программирования высокого уровня интерпретирующего типа. Точнее говоря, это даже язык *сверхвысокого* уровня, содержащий сложные операторы и функции, реализация которых на обычных языках (например, Бейсике, Паскале или Си) потребовала бы много усилий и времени. К таким функциям относятся матричные функции, функции быстрого преобразования Фурье (БПФ) и др., а к операторам — операторы построения разнообразных графиков, генерации матриц определенного вида и т.д.

2. Управляющие структуры

Помимо программ с *линейной структурой*, инструкции которых исполняются строго по порядку, существует множество программ, структура которых *нелинейна*. При этом ветви программ могут выполняться в зависимости от определенных условий, иногда с конечным числом повторений — циклов, иногда в виде циклов, завершаемых при выполнении заданного условия. Практически любая серьезная программа имеет нелинейную структуру. Для создания таких программ необходимы специальные *управляющие структуры*.

Диалоговый ввод. Пример диалоговой программы:

```
% Вычисление длины окружности с диалоговым вводом радиуса
r=0;
while r>=0,
  r=input('Введите радиус окружности r=');
  if r>=0 disp(' Длина окружности l='); disp(2*pi*r), end;
end;
```

Эта программа служит для многократного вычисления длины окружности по вводимому пользователем значению радиуса **r**. Диалог реализован с помощью команды **input**:

```
r=input('Введите радиус окружности r=');
```

При выполнении этой команды вначале выводится запрос в виде строки, затем происходит остановка работы программы и ожидается ввод значения радиуса **r** (в общем случае числа). Ввод, как обычно, подтверждается нажатием клавиши **Enter**, после чего введенное число присваивается переменной **r**. Следующая строка:

```
if r>=0 disp(' Длина окружности l='); disp(2*pi*r), end
```

с помощью команды **disp** при **r>=0** выводит надпись «**Длина окружности l=**» и

вычисленное значение длины окружности. Она представляет собой одну из наиболее простых управляющих структур типа **if...end**. В данном случае она нужна для остановки вычислений, если вводится отрицательное значение **r**.

Приведенные строки включены в управляющую структуру **while...end**. Это необходимо для циклического повторения вычислений с вводом значений **r**. Пока **r>=0**, цикл повторяется. Но стоит задать **r<0**, вычисление длины окружности перестает выполняться, а цикл завершается.

Если данная программа записана в виде m-файла **circ.m**, то работа с ней будет выглядеть следующим образом:

```
>> circ
Введите радиус окружности R=1
```

```

Длина окружности l=
6.2832
Введите радиус окружности R=2
Длина окружности l=
12.5664
Введите радиус окружности R=-1
>>

```

Итак, на примере даже простой программы мы видим пользу применения управляющих структур типа **if...end** и **while...end**, а также функций диалогового ввода **input('String')** и вывода **disp**.

Функция **input** может использоваться и для ввода произвольных строковых выражений. При этом она задается в следующем виде:

```
input('Комментарий','s')
```

При выполнении этой функции она останавливает вычисления и ожидает ввода строкового комментария. После ввода возвращается набранная строка. Это иллюстрирует следующий пример:

```

>> S=input('Введите выражение ','s')
Введите выражение (Вводим) 2*sin(1)
S =
2*sin(1)
>> eval(S)
ans =
1.6829

```

Обратите внимание на то, что функция **eval** позволяет вычислить выражение, заданное (полученное от функции **input**) в символьном виде. Вообще говоря, возможность ввода любого символьного выражения в сочетании с присущими языку программирования MATLAB управляющими структурами открывает путь к созданию диалоговых программ любой сложности.

Условный оператор if...elseif...else...end. Условный оператор **if** в общем виде записывается следующим образом:

```

if Условие
Инструкции_1
elseif Условие
Инструкции_2
else
Инструкции_3
end

```

Эта конструкция допускает несколько частных вариантов. В простейшем, типа **if...end**:

```
if Условие Инструкции end
```

пока **Условие** возвращает логическое значение 1 (то есть «истина»), выполняются **Инструкции**, составляющие тело структуры **if...end**. При этом оператор **end** указывает на конец перечня инструкций. Инструкции в списке разделяются оператором **,** (запятая) или **;** (точка с запятой). Если **Условие** не выполняется (дает логическое значение 0, то есть «ложь»), то **Инструкции** также не выполняются.

Еще одна конструкция

```
if Условие Инструкции_1 else Инструкции_2 end
```

выполняет **Инструкции_1**, если выполняется **Условие**, или **Инструкции_2** в противном случае.

Условия записываются в виде:

Выражение_1 Оператор_отношения Выражение_2

причем в качестве **Операторов_отношения** используются следующие операторы: `==`, `<`, `>`, `<=`, `>=` или `~=`. Все эти операторы представляют собой пары символов без пробелов между ними.

Циклы типа `for...end`. Циклы типа **`for...end`** обычно используются для организации вычислений с заданным числом повторяющихся циклов. Конструкция такого цикла имеет следующий вид:

for var=Выражение, Инструкция, ..., Инструкция end

Выражение чаще всего записывается в виде **`s:d:e`**, где **`s`** — начальное значение переменной цикла **`var`**, **`d`** — приращение этой переменной и **`e`** — конечное значение управляющей переменной, при достижении которого цикл завершается. Возможна и запись в виде **`s:e`** (в этом случае **`d=1`**). Список выполняемых в цикле инструкций завершается оператором **`end`**.

Точка с запятой после выражения в теле цикла предотвращает повторения вывода результатов на экран.

Хорошим стилем являются отступы при использовании циклов для лучшей читаемости, особенно, когда они вложенные.

Пример 1. Применение цикла для получения квадратов значений переменной цикла:

```
for i=1:5
    i^2,
end;
ans = 1
ans = 4
ans = 9
ans = 16
ans = 25
for x=0:.25:1
    x^2,
end;
ans = 0
ans = 0.0625
ans = 0.2500
ans = 0.5625
ans = 1
```

Возможны *вложенные циклы*.

Пример 2. Программа создания вектора *A*:

```
n=10;
for k = 1:n;
    if k<=3
        A(k)=k+1;
    elseif (k>3 & k<=10)
        A(k)=k+2;
    else
        A(k)=k+3;
    end;
end;
```

A =

2 3 4 6 7 8 9 10 11 12

Пример 3. Программа создания матрицы H :

```
for i = 1:m
    for j = 1:n
        H(i,j) = 1/(i+j);
    end;
end;
```

Пример 4. Программа создания матрицы A :

```
for i=1:3
    for j=1:3
        A(i,j)=i+j;
    end
end
```

В результате выполнения этого цикла формируется матрица A :

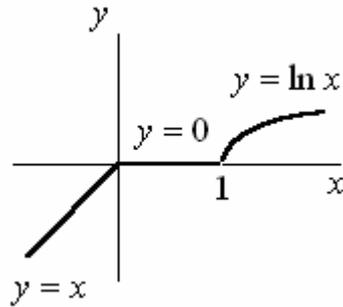
```
>> A
A =
2 3 4
3 4 5
4 5 6
```

Следует отметить, что формирование матриц с помощью оператора `:` (двоеточие) обычно занимает меньше места при записи и дает меньшее время исполнения, чем с помощью цикла. Однако применение цикла нередко оказывается более наглядным и понятным.

MATLAB допускает использование в качестве переменной цикла массива A размера $m \times n$. При этом цикл выполняется столько раз, сколько столбцов в массиве A , и на каждом шаге переменная `var` представляет собой вектор, соответствующий текущему столбцу массива A :

```
>> A=[1 2 3;4 5 6]
A =
1 2 3
4 5 6
>> for var=A; var, end
var =
1
4
2
5
3
6
```

Пример 5. Для функции, заданной графически, вычислите значения функции y для всех значений аргумента x на заданном интервале $[x_H, x_K]$ с заданным шагом d_x .



```
%Вычисление функции
deltax=0.05;
xn=-2;
xk=2;
i=1;
for xx=xn:deltax:xk;
    if xx<0;
        y(i)=xx;
    end;
    if xx<=1 && xx>=0;
        y(i)=0;
    end;
    if xx>1;
        y(i)=log(xx);
    end;
    x(i)=xx;
    i=i+1;
end;
```

```
plot(x,y);
grid on;
```

Конструкция переключателя switch...case...end. Для осуществления множественного выбора (или ветвления) используется конструкция с переключателем типа **switch**:

```
switch switch_Выражение
case case_Выражение
    Список_инструкций
case {case_Выражение1, Case_выражение2, case_Выражение3,...}
    Список_инструкций
...
otherwise,
    Список_инструкций
end
```

Если выражение после заголовка **switch** имеет значение одного из выражений **case_Выражение...**, то выполняется блок операторов **case**, в противном случае — список инструкций после оператора **otherwise**. При выполнении блока **case** исполняются те списки инструкций, для которых **case_Выражение** совпадает с **switch_Выражением**. Обратите внимание на то, что **case_Выражение** может быть числом, константой, переменной, вектором ячеек или даже строчной переменной. В последнем случае оператор **case**

Информационные технологии в менеджменте

истинен, если функция **strcmp(значение, выражение)** возвращает логическое значение «истинно».

Выполняется только первый соответствующий случай.

Замечание для программистов Си. В отличие от языка Си, оператор **switch** в Matlab не «проваливается». Если первый случай является истинным, другие случаи не выполняются. Таким образом, нет необходимости в использовании оператора **break**.

Пример 6.

```
g=3
switch g
  case {1}
    h=10;
  case {2}
    h=20;
  case {3}
    h=30;
end;
```

Циклы типа while...end. Цикл **while** повторяет группу операторов определенное число раз, пока выполняется логическое условие. Ключевое слово **end** очерчивает используемые операторы.

Пример 7. Программа создания меню для выбора пользователем действия:

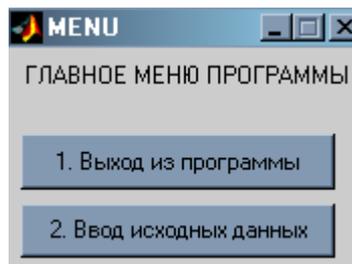
```
clear all;
id=0;
while not(id==1); %выход из программы
```

```
id=menu(['ГЛАВНОЕ МЕНЮ ПРОГРАММЫ'];{''},...
        '1. Выход из программы',...
        '2. Ввод исходных данных');
```

```
switch id
case {1}

case {2}
  clc;
  data; %вызов программы ввода исходных данных
end;
end;

clear id;
```



Пример 8. Программа создания меню для выбора пользователем действия:

```
g=0;
while not(g==1);
```

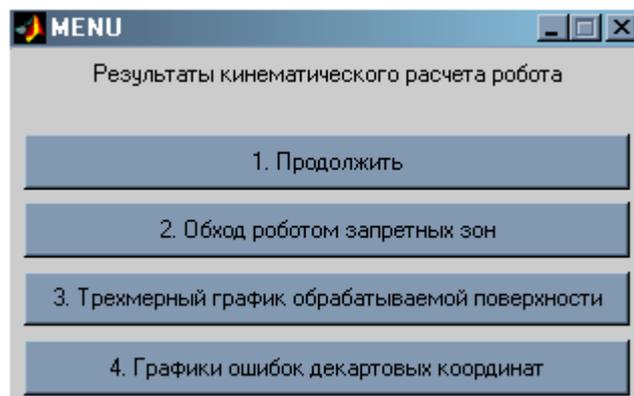
```
g=menu(['Результаты кинематического расчета робота'];{''}),...
    '1. Продолжить',...
    '2. Обход роботом запретных зон',...
    '3. Трехмерный график обрабатываемой поверхности',...
    '4. Графики ошибок декартовых координат');
```

```
switch g
```

```
    case {1}
        dinamica; %вызов файла
```

```
    case {2} %Инструкции
    case {3} %Инструкции
    case {4} %Инструкции
```

```
end;
end;
```



Операторы *break*, *continue* и *return*. В управляющих структурах, в частности в циклах **for** и **while**, часто используются операторы, влияющие на их выполнение. Так, оператор **break** может использоваться для досрочного прерывания выполнения цикла. Как только он встречается в программе, цикл прерывается. Во вложенных циклах **break** осуществляет выход только из самого внутреннего цикла.

Пример 9.

```
for i=1:10
    i,
    if i==5
        break,
    end,
end;
i = 1
i = 2
i = 3
i = 4
i = 5
```

Оператор **continue** передает управление в следующую итерацию цикла, пропуская операторы, которые записаны за ним, причем во вложенном цикле он передает управление на следующую итерацию основного цикла.

Оператор **return** обеспечивает нормальный возврат в вызывающую функцию или в режим работы с клавиатурой.

```
Пример 10.  
if isempty(A)  
    d = 1;  
    return  
else  
    ...  
end
```

В данном примере, если матрица *A* пустая, будет выведено значение 1, после чего управление будет передано в блок **else...end**.

3. Векторизация циклов

MATLAB — это язык, специально разработанный для обработки массивов и выполнения матричных операций. Всюду, где это возможно, пользователь должен учитывать это обстоятельство.

Под векторизацией понимается преобразование циклов **for** и **while** к эквивалентным векторным или матричным выражениям. При векторизации алгоритма ускоряется выполнение М-файла.

Пример 1. Один из способов вычислить 1001 значение функции синуса на интервале [0 10], используя оператор цикла:

```
i = 0;  
for t = 0:.01:10  
    i = i + 1;  
    y(i) = sin(t);  
end;
```

Эквивалентная векторизованная форма имеет вид:

```
t = 0:.01:10;  
y = sin(t);
```

Пример 2. Программа создания таблицы логарифмов:

```
x = 0;  
for k = 1:1001  
    y(k) = log10(x);  
    x = x + .01;  
end;
```

Векторизованная версия этого кода выглядит следующим образом:

```
x = 0: .10:10;  
y = log10(x);
```

Для более сложных программ возможности векторизации не так очевидны. Однако, когда важна скорость, вы должны всегда искать способы векторизации вашего алгоритма.

Лекция № 16

Matlab: Составление функций пользователя

1. Функции, задаваемые пользователем
2. Понятие о файлах-функциях
3. Глобальные переменные

1. Функции, задаваемые пользователем

В пакете Matlab существует три способа задания функций пользователя.

1. Использование команды **inline** приводит к созданию функции, зависящей от указанных аргументов.

Пример 1. Создание функции одной переменной:

```
>> f=inline('x^2','x')
```

```
f =
```

```
Inline function:
```

```
f(x) = x^2
```

```
>> f(5)
```

```
ans =
```

```
25
```

Пример 2. Создание функции нескольких переменных:

```
>> f=inline('x^2+y^5+sin(z)','x','y','z')
```

```
f =
```

```
Inline function:
```

```
f(x,y,z) = x^2+y^5+sin(z)
```

```
>> f(5,6,9)
```

```
ans =
```

```
7.8014e+003
```

```
>> f(0.5,0.6,0.9)
```

```
ans =
```

```
1.1111
```

Поскольку большинство функций программы Matlab могут оперировать как векторами, так и скалярами, чтобы быть уверенным, что созданная функция может оперировать с векторами, необходимо использовать поэлементные операции умножения, деления и возведения в степень (**.* ./ .^**). Таким образом, получается векторизованная версия функции.

Пример 3. Если ставится задача вычислить значения функции в некотором диапазоне значений аргумента, то необходимо использовать поэлементные операции:

```
>> f=inline('x.^2','x')
```

```
f =
```

```
Inline function:
```

```
f(x) = x.^2
```

```
>> f(5:9)
```

```
ans =
```

```
25 36 49 64 81
```

```
%-----
```

```
>> f=inline('x.^2+y.^5+sin(z)','x','y','z')
```

```
f =
```

```
Inline function:
```

```

f(x,y,z) = x.^2+y.^5+sin(z)
>> f(0:3,0:3,0:3)
ans =
0 2.8415 36.9093 252.1411

```

2. Использование оператора @ позволяет создавать так называемые *анонимные функции*. Этот метод является новым в программе Matlab 7, и в настоящее время этому методу отдается предпочтение.

Анонимная (безымянная) функция — в программировании особый вид функций, которые объявляются в месте использования и не получают уникального идентификатора для доступа к ним. Обычно при создании они либо вызываются напрямую, либо ссылка на функцию присваивается переменной, с помощью которой затем можно косвенно вызывать данную функцию.

Пример 4.

```

>> f=@(x)x.^2
f=
@(x)x.^2
>> f=@(x,y,z) x.^2+y.^5+sin(z);
>> f([1 2],[3 4])

```

дает значения функции в точках (1, 3) и (2, 4).

3. Задание функции в отдельных файлах (создание файла-функции). Это метод подробно рассмотрим в следующем разделе.

2. Понятие о файлах-функциях

Кроме m-файла сценария, в Matlab существует *m-файл функция*. Структура m-файла функции имеет вид:

function [Список выходных параметров] = funcname(Список входных параметров)

Текст программы, реализующей алгоритм функции

Выходной параметр 1 = выражение

Выходной параметр 2 = выражение

...

Выходной параметр N = выражение

Правила создания функций пользователя:

1. Файл, содержащий программу реализации функции, должен иметь расширение «**m**» и называться также как имя функции. Matlab автоматически присваивает данное имя при выполнении команды **Save as**.

2. Первая строка файла должна содержать описание функции в виде:

function [x1,x2,...,xN] = funcname(param1, param2, ...,paramM);,

где **function** — ключевое слово; **x1,x2,...,xN** — переменные, необходимые для сохранения результата функции (выходные параметры функции); **param1, param2, ...,paramM** — параметры, передаваемые в функцию.

3. Далее следует текст программы, реализующей алгоритм функции. Каждое действие в программе должно быть завершено символом точкой с запятой.

4. В основном файле происходит вызов функции путем упоминания имени, например, если необходимо в переменные **y1,y2,...,yN** записать результат работы функции:

[y1,y2,...,yN] = funcname(param1, param2, ...,paramM);

Причем, имена возвращаемых переменных и имена параметров функции могут быть разными в основной программе и в функции, главное соблюдать порядок их перечисления.

Информационные технологии в менеджменте

Функция может вызываться в цикле, например:

```
for i=1:10;  
y(i) = funcname(param1(i), param2(i), ...);  
end;
```

В этом случае `param1`, `param2`, ... должны быть векторами соответствующей размерности.

Все переменные, используемые в файле-функции, являются локальными, т. е. действуют только в пределах тела функции.

Файл-функция является самостоятельным программным модулем, который связан с другими модулями и головной программой через входные и выходные параметры. При обнаружении файла-функции он компилируется и затем исполняется.

Таким образом, пользователь Matlab может сам создавать функции, аналогичные тем, которые входят в этот программный пакет. Создав такую функцию, можно пользоваться ею в среде Matlab, набирая ее имя и параметры в скобках.

Пример 1.

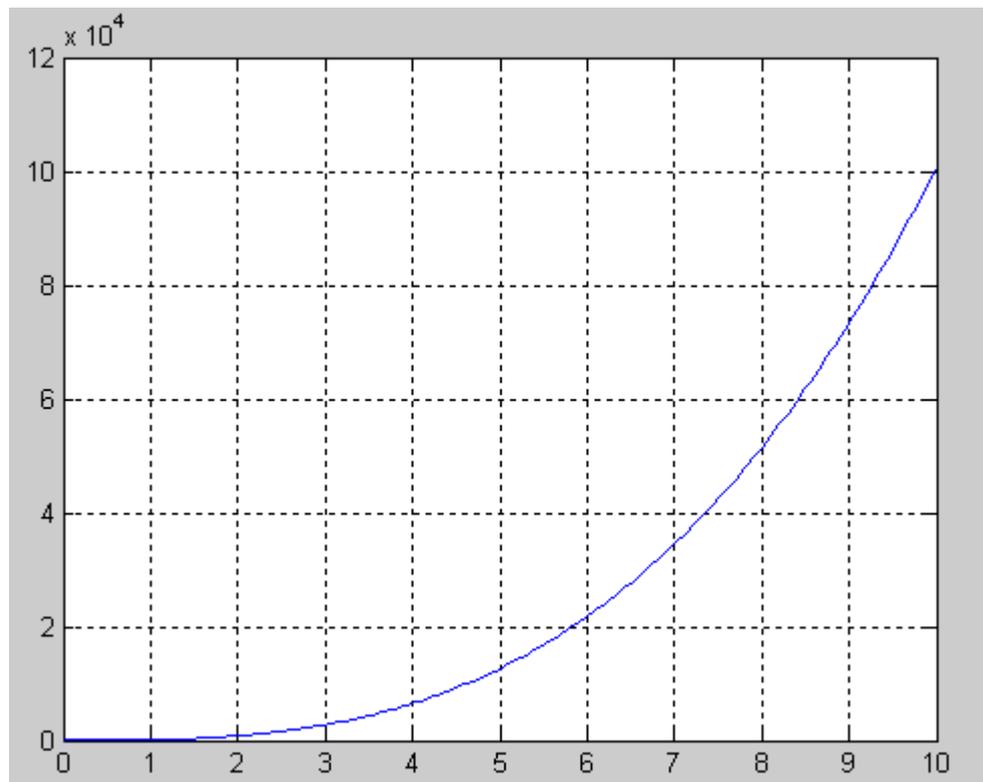
Напишите функцию пользователя, которая возвращает вектор, элементы которого являются значениями функции $f = y^2 \cdot t^3 + x \cdot t^2$.

Входными параметрами функции должны быть значения переменных x и y , а также вектор значений моментов времени t .

Результат отобразите в виде графика зависимости $f(t)$.

Содержание файла основной программы:

```
x=5;  
y=10;  
deltat=0.05;  
tk=10;  
t=0:deltat:tk;  
f=fun_07_1(x,y,t); %вызов функции  
figure  
plot(t,f)  
grid on  
Содержание файла-функции fun_07_1:  
function [f]=fun_07_1(x,y,t);  
f=y^2*t.^3+x*t.^2;
```



Пример 2. Напишите функцию пользователя, которая возвращает значения функции $f = y^2 \cdot t^3 + x \cdot t^2$ в дискретных точках t .

Входными параметрами функции должны быть значения переменных x и y , а также дискретные значения переменной t .

Результат отобразите в виде графика зависимости $f(t)$.

Содержание файла основной программы:

```
i=0;
for t=0:deltat:tk;
    i=i+1;
    f(i)=fun_07_2(x,y,t);
end;
```

```
t=0:deltat:tk;
```

```
figure
plot(t,f);
grid on;
```

Содержание файла-функции **fun_07_2**:

```
function [f]=fun_07_2(x,y,t);
f=y^2*t^3+x*t^2;
```

3. Глобальные переменные

Если необходимо чтобы более одной функции использовали некоторую переменную можно объявить ее как глобальную во всех функциях, например:

global XX.

Хотя это не обязательно, но использование больших букв для имени глобальной переменной поможет отличить их от других переменных.

Определение **global** должно быть до использования самой переменной. Например, создадим m-файл **falling.m**:

```
function h = falling(t)
```

```
global GRAVITY
```

```
h = 1/2*GRAVITY*t^2;
```

В основном файле или командном окне запишем:

```
global GRAVITY
```

```
GRAVITY = 32;
```

```
y = falling((0: .1: 5)');
```

Таким образом, строки определения **GRAVITY** делают ее доступной внутри функции. Можно впоследствии изменить **GRAVITY** и получить новое решение, не редактируя какие-либо файлы-функции.

Лекция № 17

Matlab: Решение уравнений и их систем

1. Решение уравнений и их систем
2. Решение дифференциальных уравнений и их систем

1. Решение уравнений и их систем

Аналитическое решение уравнений и их систем.

Функция решение алгебраических уравнений – *solve*

Для решения систем алгебраических уравнений и одиночных уравнений служит функция *solve*:

solve(*expr1*, *expr2*,... *exprN*, *var1*, *var2*,... *varN*) — возвращает значения переменных *var1*, при которых соблюдаются равенства, заданные выражениями *expr1*. Если в выражениях не используются знаки равенства, то полагается *expr1 = 0*;

Примеры решения уравнений:

```

» syms x y;
» solve(x^3 -1, x)
ans =
[ 1]
[ -1/2+1/2*i*3^(1/2)]
[ -1/2-1/2*i*3^(1/2)]
» syms a b c
» solve(a*x^2+b*x+c)
ans =
[ 1/2/a*(-b+(b^2-4*a*c)^(1/2))]
[ 1/2/a*(-b-(b^2-4*a*c)^(1/2))]
» S = solve('x+y=3', 'x*y^2=4', x, y)
S =
    x : [ 3x1 sym ]
    y : [ 3x1 sym ]
» S.x
ans =
[4]
[1]
[1]
» S.y
ans =
[-1]
[2]
[2]
» solve('sin(x)=0.5', x)
ans =
0.52359877559829887307710723054658

```

Численное решение уравнений и их систем. Задача нахождения корней уравнений встречается в различных областях научно-технических исследований. Проблема формулируется следующим образом. Пусть задана непрерывная функция $f(x)$ и требуется найти корень уравнения:

$$f(x) = 0.$$

Будем предполагать, что имеется интервал изменения x $[a;b]$, на котором необходимо исследовать функцию $f(x)$ и найти значение x_0 , при котором $f(x_0)$ равно или весьма мало отличается от нуля.

Данная задача в системе Matlab может быть решена следующим образом. Вначале необходимо построить график функции $f(x)$ на заданном интервале и убедиться в существовании корня или нескольких корней. Затем применить функции поиска корней.

Если существует один корень и график $f(x)$ пересекает ось Ox , то можно применить функцию:

Информационные технологии в менеджменте

- `fzero(fun, x)` возвращает уточненное значение x , при котором достигается нуль функции `fun`, представленной строкой, при начальном значении аргумента x . Возвращенное значение близко к точке, где функция меняет знак, или равно NaN, если такая точка не найдена;
- `fzero(fun, [x1 x2])` возвращает значение x , при котором $fun(x) = 0$ с заданием интервала поиска с помощью вектора $x = [x1 \ x2]$, такого что знак $fun(x(1))$ отличается от знака $fun(x(2))$. Если это не так, выдается сообщение об ошибке. Вызов функции `fzero` с интервалом гарантирует, что `fzero` возвратит значение, близкое к точке, где `fun` изменяет знак;
- `fzero(fun, x, tol)` возвращает результат с заданной погрешностью `tol`;
- `fzero(fun, x, tol, trace)` выдает на экран информацию о каждой итерации;

Для функции `fzero` нуль рассматривается как точка, где график функции `fun` пересекает ось x , а не касается ее. В зависимости от формы задания функции `fzero` реализуются следующие хорошо известные численные методы поиска нуля функции: деления отрезка пополам, секущей и обратной квадратичной интерполяции.

Если $f(x)$ имеет больше одного корня и может касаться и пересекать ось Ox , то следует применить более мощную функцию **`fsolve`** из пакета **Optimization Toolbox**, которая решает задачу методом наименьших квадратов:

`fsolve(fun, x0:h:x1),`

поиск корней выполняется на интервале от x_0 до x_1 с шагом h .

Пример 7. Найти корень нелинейного уравнения $10^x + 2x - 100 = 0$ на интервале $[1.0; 2.0]$.

Протокол программы

```
>> % Строим график заданной функции
>> x = 1.0 : 0.001 : 2.0; y = 10.0.^x + 2.0*x - 100.0;
>> plot(x, y) ; grid on
```

Появляется окно с графиком функции $10^x + 2x - 100$ (см. рис. 2.4), из которого следует, что корень функции на заданном интервале существует. Для точного определения корня применяем `fzero` и `fsolve`.

Информационные технологии в менеджменте

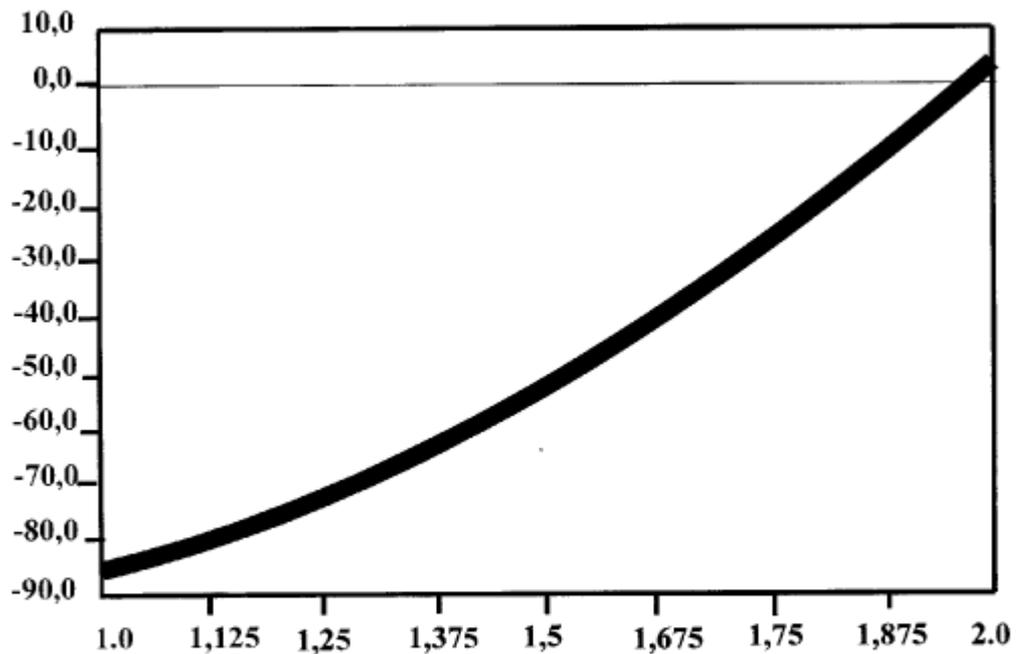


Рис. 2.4

```
>> X1 = fzero (' (10.0.^x + 2.0*x - 100.0) ', [1.0 2.0])
```

Результат решения

```
X1 =  
1.9824
```

```
>> X2 = fsolve (' (10.0.^x + 2.0*x - 100.0) ', 1.0 : 2.0)
```

Результат решения

```
X2 =  
1.9824 1.9824
```

Функция `fsolve` может использоваться и для решения систем нелинейных уравнений. В этом случае речь идет о нахождении значений двух и более переменных, удовлетворяющих решению заданной системы нелинейных уравнений.

Пример. Численно решить систему уравнений:

$$\begin{cases} 2x - y - 2z = -1 \\ 2y + z = 3 \\ -2x - 2y + 2z = -2 \end{cases}$$

В этом случае необходимо создать файл-функцию для расчета значений уравнений от x :

```
function F = fun_07(x)  
F = [2*x(1)-x(2)-2*x(3)+1;  
2*x(2)+x(3)-3;  
-2*x(1)-2*x(2)+2*x(3)+2];
```

При этом ввели вектор x размерности 3×1 , такой что $x = (x \ y \ z)$.

В основной программе делаем вызов функции **fun_07** в функции **fsolve**:

x0=[10;30;-1];

X = fsolve('fun_07',x0)

Решение задачи определения корней полинома осуществляется при помощи встроенной функции **roots(v)**, где v — вектор-столбец коэффициентов полинома, первым элементом которого является коэффициент при x^n , вторым — коэффициент при x^{n-1} и т. д.

2. Решение дифференциальных уравнений и их систем

Анализ поведения многих систем и устройств в динамике, а также решение многих задач в теории колебаний и в поведении упругих оболочек обычно базируется на решении систем *обыкновенных дифференциальных уравнений* (ОДУ), или, в оригинале, *ordinary differential equations* (ODEs). Их, как правило, представляют в виде системы из дифференциальных уравнений первого порядка в форме Коши:

$$\frac{dy}{dt} = y' = f(y, t) \quad (1)$$

с граничными условиями $y(t_0, t_{\text{end}}, p) = b$, где t_{end}, t_0 — начальные и конечные точки интервалов. Параметр t (независимая переменная) необязательно означает время, хотя чаще всего решение дифференциальных уравнений ищется во временной области.

Система дифференциальных уравнений в форме Коши записывается аналогично (1), но под y в этом случае подразумевается вектор-столбец зависимых переменных. Вектор p задает начальные условия.

Для решения дифференциальных уравнений второго и высшего порядка их нужно свести к системе дифференциальных уравнений первого порядка.

Возможны дифференциальные уравнения, не разрешенные относительно производной:

$$F(t, y, dy/dt) = 0. \quad (2)$$

Уравнения (2) аналитически к форме (1) обычно привести не удастся.

Однако численное решение особых трудностей не вызывает — достаточно для определения $f(y, t)$ решить (2) численно относительно производной при заданных y и t .

Наряду с ОДУ MATLAB может оперировать с *дифференциальными алгебраическими уравнениями* (ДАУ, или *differential algebraic equations* — DAEs). ОДУ и ДАУ являются основой математического моделирования динамических нелинейных (и линейных) систем. Автоматическое их составление и решение реализованы в специальном расширении **Simulink**.

Аналитическое решение дифференциальных уравнений и их систем.

Для решения дифференциальных уравнений в форме Коши MATLAB имеет следующую функцию:

- `dsolve('eqn1', 'eqn2', ...)` — возвращает аналитическое решение системы дифференциальных уравнений с начальными условиями. Они задаются равенствами `eqn1` (вначале задаются уравнения, затем начальные условия).

По умолчанию независимой переменной считается переменная 't', обычно обозначающая время. Можно использовать и другую переменную, добавив ее в конец списка параметров функции `dsolve`. Символ D обозначает производную по независимой переменной, то есть d/dt , при этом D2 означает d^2/dt^2 и т. д. Имя независимой переменной не должно начинаться с буквы D.

Начальные условия задаются в виде равенств `y(a)=b` или `Dy(a)=b`, где `y` — независимая переменная, `a` и `b` — константы. Если число начальных условий меньше, чем число дифференциальных уравнений, то в решении будут присутствовать произвольные постоянные C1, C2 и т. д.

Примеры применения функции `dsolve`:

```
» dsolve('D2x = -2*x')
ans =
    C1*cos(2^(1/2)*t) + C2*sin(2^(1/2)*t)
» dsolve('D2y = -2*x + y', 'y(0) = 1', 'x')
ans =
    (2*x*exp(x) + (-C2 + 1)*exp(x)^2 + C2 / exp(x))
```

Пример 1. Аналитически решите дифференциальное уравнение первого порядка $y'(t) = y \cos t + \sin 2t$:

```
dsolve('Dy=y*cos(t)+sin(2*t)')
```

```
ans =
-2*sin(t)-2+exp(sin(t))*C1
```

Пример 2. Аналитически решите дифференциальное уравнение второго порядка $mx'' + Hx' + kx = -\sqrt{t}$:

```
dsolve('m*D2x+H*Dx+k*x=-sqrt(t)')
```

```
ans =
-2*m*(2*(-(2*H-2*(H^2-4*m*k)^(1/2))/m)^(1/2)*t^(1/2)*(-
(2*H+2*(H^2-4*m*k)^(1/2))/m)^(1/2)*(H^2-4*m*k)^(1/2)+exp(-
1/2*(H+(H^2-4*m*k)^(1/2))*t/m)*(-(2*H-2*(H^2-
4*m*k)^(1/2))/m)^(1/2)*pi^(1/2)*erf(1/2*(-(2*H+2*(H^2-
4*m*k)^(1/2))/m)^(1/2)*t^(1/2))*H-exp(-1/2*(H+(H^2-
4*m*k)^(1/2))*t/m)*(-(2*H-2*(H^2-
4*m*k)^(1/2))/m)^(1/2)*pi^(1/2)*erf(1/2*(-(2*H+2*(H^2-
4*m*k)^(1/2))/m)^(1/2)*t^(1/2))*(H^2-4*m*k)^(1/2)-exp(-1/2*(H-
(H^2-4*m*k)^(1/2))*t/m)*(-(2*H+2*(H^2-
```

Информационные технологии в менеджменте

$$\frac{4mk^{1/2}}{m^{1/2}} \pi^{1/2} \operatorname{erf}\left(\frac{1}{2} \sqrt{-2H-2(H^2-4mk)^{1/2}}\right) t^{1/2} H - \exp\left(-\frac{1}{2} \sqrt{H-(H^2-4mk)^{1/2}}\right) \frac{4mk^{1/2}}{m^{1/2}} \pi^{1/2} \operatorname{erf}\left(\frac{1}{2} \sqrt{-2H+2(H^2-4mk)^{1/2}}\right) t^{1/2} (H^2-4mk)^{1/2} / (H+(H^2-4mk)^{1/2}) / (-2H+2(H^2-4mk)^{1/2}) / m^{1/2} / (H-(H^2-4mk)^{1/2}) / (-2H-2(H^2-4mk)^{1/2}) / m^{1/2} + C1 \exp\left(-\frac{1}{2} \sqrt{H+(H^2-4mk)^{1/2}}\right) t / m + C2 \exp\left(-\frac{1}{2} \sqrt{H-(H^2-4mk)^{1/2}}\right) t / m$$

Пример 3. Аналитически решите систему дифференциальных уравнений

$$\begin{cases} y_0' = ay_1 \\ y_1' = y_0 + y_1 \end{cases}$$

F=dsolve('Dy0=a*y1','Dy1=y0+y1')

F =

y0: [1x1 sym]

y1: [1x1 sym]

>> F.y0

ans =

$$C1 \exp\left(\frac{1}{2}t + \frac{1}{2}t(1+4a)^{1/2}\right) + C2 \exp\left(\frac{1}{2}t - \frac{1}{2}t(1+4a)^{1/2}\right)$$

>> F.y1

ans =

$$\frac{1}{2} \left(C1 \exp\left(\frac{1}{2}t + \frac{1}{2}t(1+4a)^{1/2}\right) + C1 (1+4a)^{1/2} \exp\left(\frac{1}{2}t + \frac{1}{2}t(1+4a)^{1/2}\right) + C2 \exp\left(\frac{1}{2}t - \frac{1}{2}t(1+4a)^{1/2}\right) - C2 (1+4a)^{1/2} \exp\left(\frac{1}{2}t - \frac{1}{2}t(1+4a)^{1/2}\right) \right) / a$$

Численное решение дифференциальных уравнений и их систем.

Решение дифференциальных уравнений осуществляют функции **ode23** и **ode45**. Они могут применяться как для решения простых дифференциальных уравнений, так и для моделирования сложных динамических систем.

Известно, что система дифференциальных уравнений может быть представлена в форме Коши:

$$\frac{dy}{dt} = f(y,t)$$

где y — вектор переменных состояния системы; t — аргумент (обычно время); f — нелинейная вектор-функция от переменных состояния y и аргумента t .

Обращение к процедурам численного интегрирования дифференциальных уравнений имеет вид:

[t,y]=ode23('имя функции',tspan, y0, options)

[t,y]=ode45('имя функции',tspan, y0, options)

где *имя функции* — строка символов, являющаяся именем m-файла, в котором вычисляется вектор-функция $f(y,t)$, т. е. в котором находятся правые части системы дифференциальных уравнений; y_0 — вектор начальных значений переменных состояния; t — массив значений аргумента, соответствующих шагам интегрирования; y — матрица проинтегрированных значений фазовых

Информационные технологии в менеджменте

переменных, в которой каждый столбец соответствует одной из переменных состояния, а строка содержит значения переменных состояния, соответствующих определенному шагу интегрирования; `tspan` — вектор-строка $[t_0, t_{last}]$, содержащая два значения: t_0 — начальное значение аргумента t ; t_{last} — конечное значение аргумента t ; `options` — строка параметров, определяющих значения допустимой относительной и абсолютной погрешности интегрирования.

Для получения решений дифференциальных уравнений в определенные моменты времени можно указать явным образом `tspan = [t_0, t_1, t_2, ..., t_n]`.

Параметр `options` можно не указывать. Тогда по умолчанию допустимая относительная погрешность интегрирования принимается равной $1.0e-3$, а абсолютная (по каждой из переменных состояния) — $1.0e-6$.

Функция **ode23** осуществляет интегрирование численным методом Рунге-Кутта 2-го порядка, а с помощью метода 3-го порядка контролирует относительные и абсолютные ошибки интегрирования на каждом шаге и изменяет величину шага интегрирования так, чтобы обеспечить заданные пределы ошибок интегрирования.

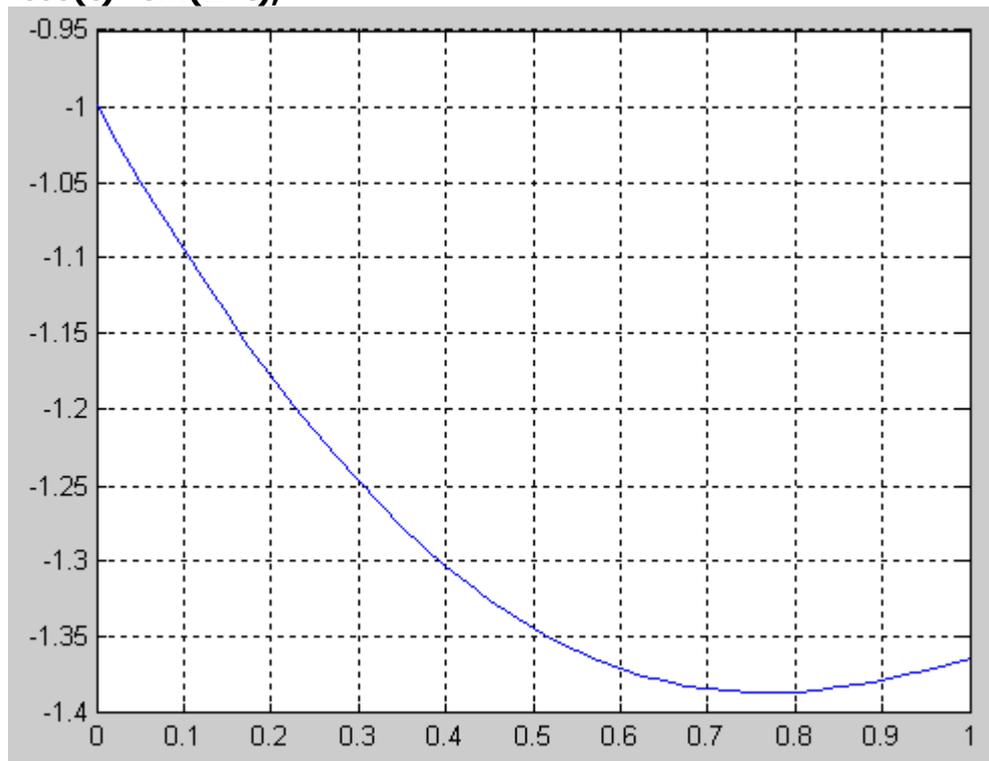
Для **ode45** основным методом является метод Рунге-Кутта 4-го порядка, а величина шага контролируется методом 5-го порядка.

Пример 4. Численно решите дифференциальное уравнение первого порядка $y'(t) = y \cos t + \sin 2t$:

```
tspan=[0 1];
y0=-1;
[T,Y] = ode45('fun_08_1',tspan,y0);
figure
plot(T,Y)
grid on
```

Содержание файла-функции **fun_08_1**:

```
function dy = fun_08_1(t,y)
dy=y*cos(t)+sin(2*t);
```



Пример 5. Численно решите дифференциальное уравнение второго порядка $mx'' + Hx' + kx = -\sqrt{t}$.

При этом необходимо преобразовать дифференциальное уравнение второго порядка в систему двух дифференциальных уравнений первого порядка (переход к переменным состояния). Для этого введем так называемые *переменные состояния* x_0, x_1, x_2 :

$$x_0 = x, \quad x_1 = x', \quad x_2 = x''.$$

Тогда можно записать:

$$\begin{cases} x_0' = x_1 \\ x_1' = x_2 \\ x_2' = x_3 = \frac{-Hx_1 - kx_0 - \sqrt{t}}{m} \end{cases}$$

Однако, поскольку в Matlab нумерация индексов массива начинается с 1 (а не с нуля), то систему уравнений необходимо переписать в следующем виде:

$$\begin{cases} x_1' = x_2 \\ x_2' = x_3 \\ x_3' = x_4 = \frac{-Hx_2 - kx_1 - \sqrt{t}}{m} \end{cases}$$

tspan=[0 15];

x0=[0 -10];

[T,X] = ode45('fun_08_2',tspan,x0);

Для наглядного представления результата строится график зависимостей столбцов возвращенной матрицы решений X , от вектора T :

figure

plot(T,X(:,1),'b',T,X(:,2),'g')

grid on

Содержание файла-функции **fun_08_2**:

function dx = fun_08_2(t,x)

m=0.5;

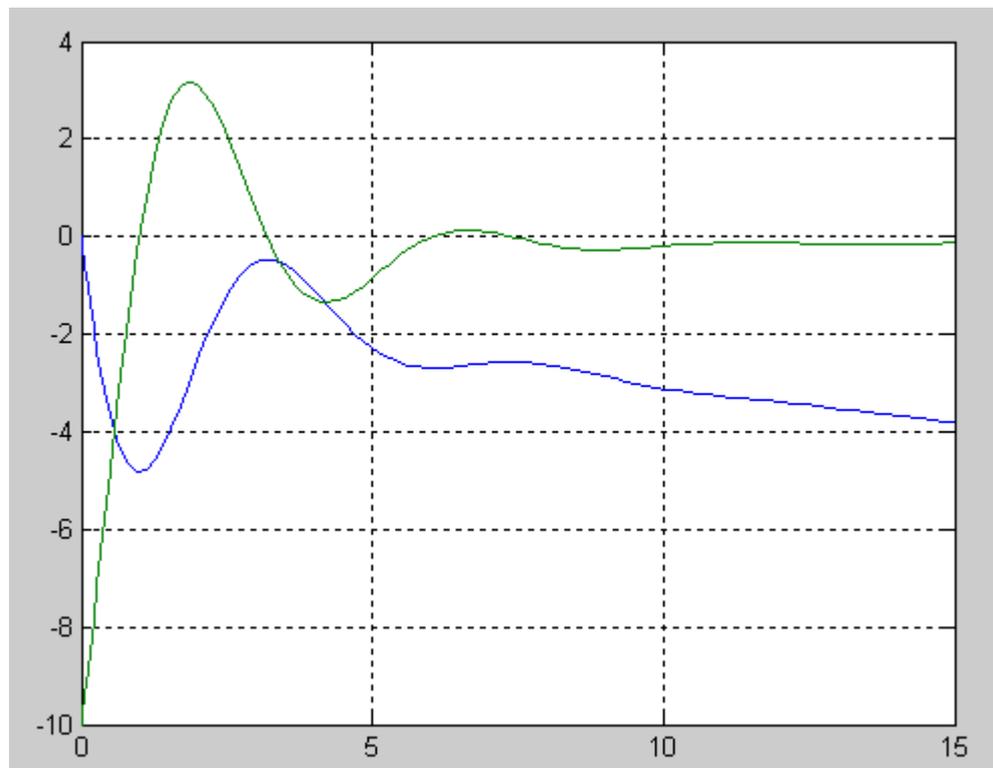
H=0.5;

k=1;

dx = zeros(2,1); % создает нулевой вектор-столбец, где будут храниться полученные на данном этапе значения переменных состояния

dx(1)=x(2);

dx(2)=(-H*x(2)-k*x(1)-sqrt(t))/m;



Пример 6. Решить систему дифференциальных уравнений:
$$\begin{cases} y_1' = y_2 y_3 \\ y_2' = -y_1 y_3 \\ y_3' = -0,51 y_1 y_2 \end{cases}$$

$$y_1(0) = 0$$

с начальными условиями $y_2(0) = 1$

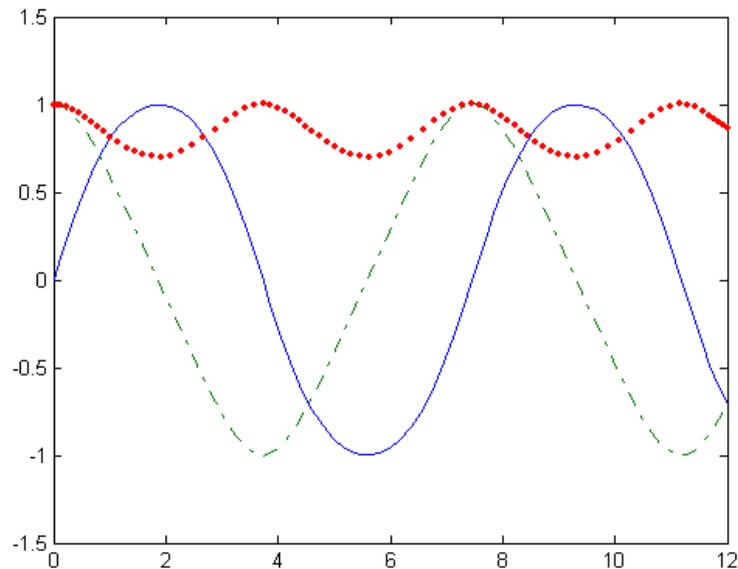
$$y_3(0) = 1$$

Для моделирования данной системы, создаем функцию **rigid**, следующего содержания:

```
function dy=rigid(t,y)
dy=zeros(3,1);
dy(1)=y(2)*y(3);
dy(2)=-y(1)*y(3);
dy(3)=-0.51*y(1)*y(2);
```

Решение находится на интервале [0; 12] с начальными условиями [0 1 1]:

```
[T,Y] = ode45('rigid',[0 12],[0 1 1]);
plot(T,Y(:,1),'-',T,Y(:,2),'-.',T,Y(:,3),'-')
```



Лекция № 18

Matlab: Функции для решения задач интерполяции и аппроксимации

1. Функции для решения задач интерполяции
2. Функции для решения задач аппроксимации

1. Функции для решения задач интерполяции

В Matlab для решения задач интерполяции имеются следующие функции:

spline — интерполяция функции одной переменной кубическим сплайном;

interp1 — одномерная табличная интерполяция;

interp2 — двумерная табличная интерполяция;

griddata — двумерная табличная интерполяция на неравномерной сетке.

Интерполяция функции одной переменной кубическим сплайном.

Функция **yi = spline(x, y, xi)** интерполирует значения функции y в точках xi внутри области определения функции, используя кубические сплайны.

Пример 1. Зададим синусоиду 11 точками и проведем интерполяцию кубическими сплайнами, используя мелкую сетку (рис. 1):

```
x = 0:10;  
y = sin(x);  
xi = 0:.25:10;  
yi = spline(x, y, xi);  
plot(x, y, 'o', xi, yi, 'g')  
grid on
```

Информационные технологии в менеджменте

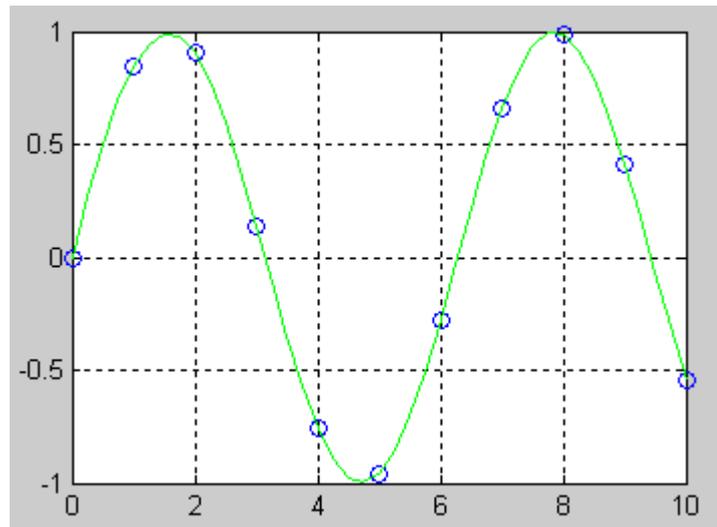


Рис. 1.

Большие возможности для работы со сплайнами предоставляет пользователю специализированный пакет **Spline Toolbox**.

Одномерная табличная интерполяция. Функция $y_i = \text{interp1}(x, y, x_i)$ строит интерполирующую кривую для одномерного массива y , заданного на сетке x ; выходной массив y_i может быть определен на более мелкой сетке x_i . Если Y — двумерный массив, то интерполирующая кривая строится для каждого столбца. По умолчанию реализована линейная интерполяция.

Функция $y_i = \text{interp1}(x, y, x_i, \text{'метод'})$ позволяет задать метод интерполяции:

linear	линейная
cubic	кубическая
spline	кубические сплайны

Принято, что аргумент x изменяется монотонно; кроме того, для кубической интерполяции предполагается, что сетка по x равномерна.

Пример 2. Зададим синусоиду 11 точками и проведем интерполяцию, используя мелкую сетку (рис. 2):

```
x = 0:10;
y = sin(x);
xi = 0:.25:10;
yi = interp1(x, y, xi);
plot(x, y, 'o', xi, yi, 'g')
hold on
yi = interp1(x, y, xi, 'spline');
plot(x, y, 'ob', xi, yi, 'm')
grid on
```

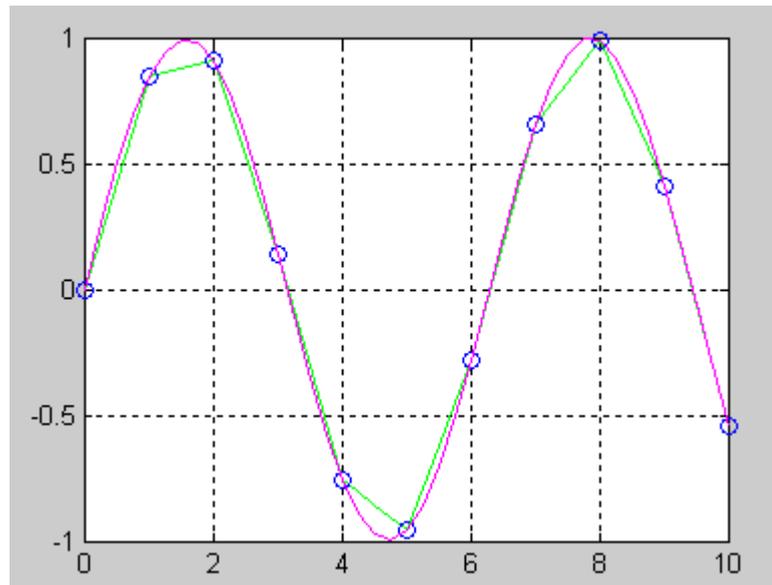


Рис. 2.

Двумерная табличная интерполяция. Функция **ZI = interp2(X, Y, Z, XI, YI)** интерполирует данные, определяющие некоторую поверхность на двумерной сетке (X, Y) ; выходной массив ZI может быть определен на более мелкой сетке (XI, YI) . По умолчанию реализована линейная интерполяция.

Функция **ZI = interp2(X, Y, Z, XI, YI, 'метод')** позволяет задать метод интерполяции:

ar'	'line	линейная
c'	'cubi	кубическая

Принято, что аргументы X и Y изменяются монотонно; кроме того, для кубической интерполяции предполагается, что сетка (X, Y) равномерна.

Пример 3. Проведем интерполяцию функции **peaks**, используя мелкую сетку (рис. 3):

```
[X, Y] = meshgrid(-3:0.25:3);
Z = peaks(X, Y);
[XI, YI] = meshgrid(-3:0.125:3);
ZI = interp2(X, Y, Z, XI, YI);
mesh(X, Y, Z)
hold on
mesh(XI, YI, ZI+15)
```

Функция **peaks** осуществляет построение поверхности с нормальным законом распределения значений по координате Z .

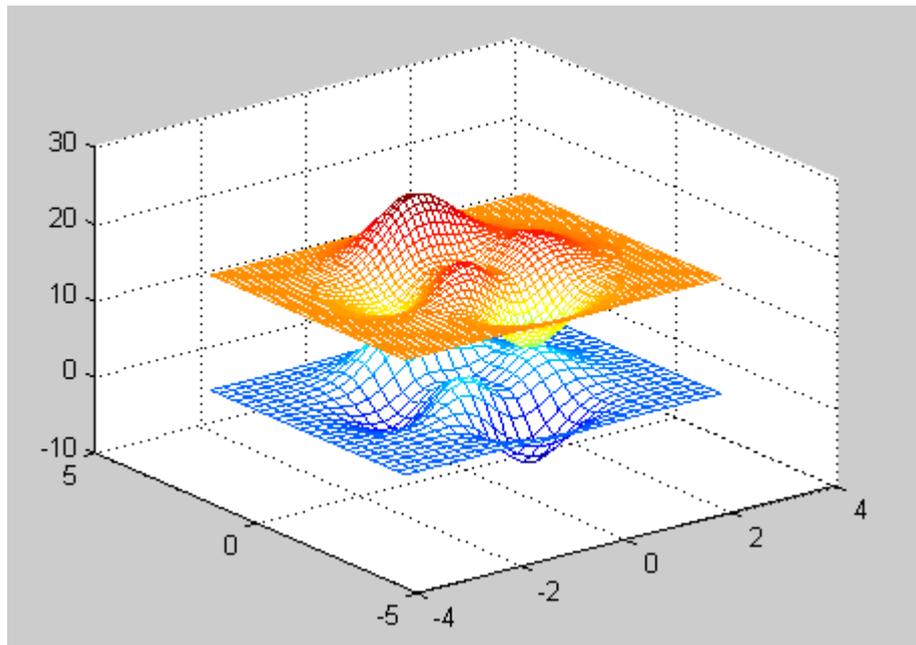


Рис. 3.

Двумерная интерполяция на неравномерной сетке. Функция **ZI = griddata(X, Y, Z, XI, YI)** возвращает массив **ZI**, который определен на новой сетке (**XI**, **YI**) в результате интерполяции исходной функции **Z**, заданной на неравномерной сетке (**X**, **Y**).

Функция **[XI, YI, ZI] = griddata(X, Y, Z, XI, YI)** кроме массива **ZI** возвращает массивы **XI**, **YI**, упорядоченные по аналогии с функцией **meshgrid**.

Пример 4. Определим функцию на сетке, заданной 100 точками, выбранными случайно на отрезке $[-2; 2]$:

```
X = rand(100, 1) * 4 - 2;
Y = rand(100, 1) * 4 - 2;
Z = X.*exp(-X.^2 - Y.^2);
```

При этом функция **X = rand(m, n)** формирует массив размера $m \times n$, элементами которого являются случайные величины, распределенные по равномерному закону в интервале (0, 1).

Векторы **X**, **Y**, **Z** определяют 100 случайных точек на поверхности функции **ZI**, которую зададим на следующей равномерной сетке:

```
ti = -2:0.25:2;
[XI, YI] = meshgrid(ti, ti);
ZI = griddata(X, Y, Z, XI, YI);
```

Построим поверхность функции, полученной в результате интерполяции на неравномерной случайной сетке (рис. 4):

```
mesh(XI, YI, ZI)
hold on
plot3(X, Y, Z, 'r o')
```

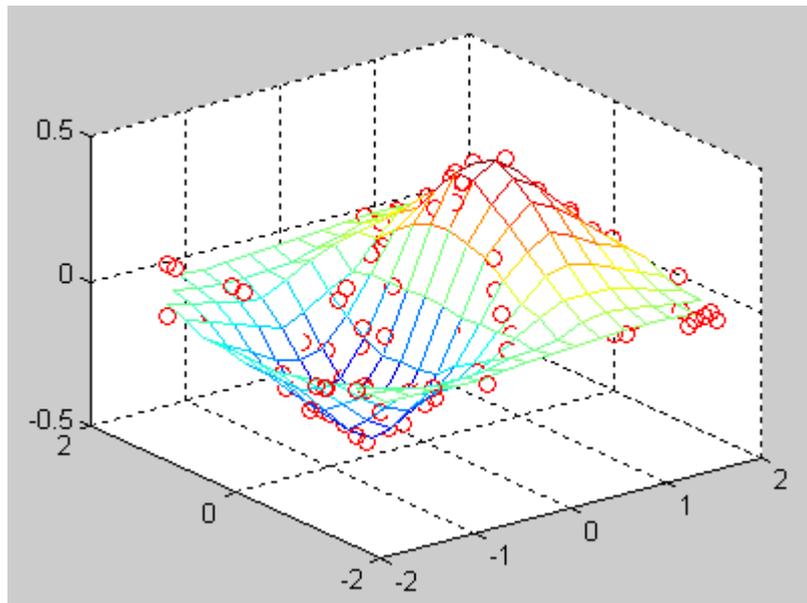


Рис. 4.

2. Функции для решения задач аппроксимации

В Matlab для решения задач аппроксимации имеются следующие функции:

polyfit — аппроксимация данных полиномом;

interpft — аппроксимация периодической функции на основе быстрого преобразования Фурье.

Аппроксимация данных полиномом. Функция **p = polyfit(x, y, n)** находит коэффициенты полинома степени n , который аппроксимирует функцию $y(x)$ методом наименьших квадратов. Результатом является строка p длины $n + 1$, содержащая коэффициенты аппроксимирующего полинома.

Пример 1. Рассмотрим аппроксимацию функции $y = \sin(x) + \cos(x)$:

```
x = (0:0.1:10)';
```

```
y = sin(x)+cos(x);
```

вычислим коэффициенты аппроксимирующего полинома степени 6:

```
p = polyfit(x, y, 6)
```

```
p = 0.0004 -0.0079 0.0323 0.2245 -1.6581 2.0673 0.8047
```

вычислим значения полинома в точках сетки:

```
f = polyval(p, x);
```

Построим графики функции и аппроксимирующего полинома (рис. 1):

```
plot(x, y, 'b o', x, f, 'r')
```

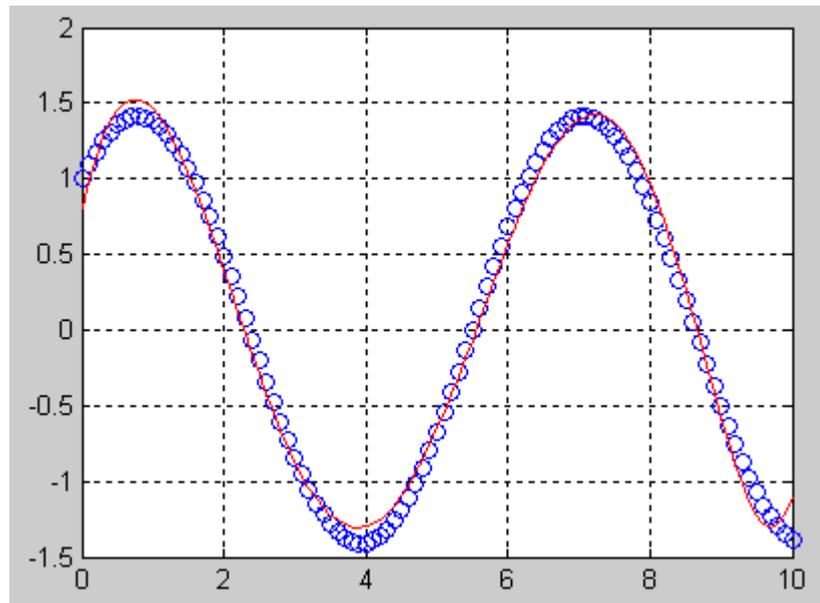


Рис. 1.

Рассмотрим аппроксимацию функции $y = \sin(2x) + \cos(3x)$ (рис. 2). В этом случае для качественной аппроксимации необходимо использовать полином 30 степени:

```
x = (0:0.1:10)';
y = sin(2*x)+cos(3*x);
p = polyfit(x, y, 30)
f = polyval(p, x);
plot(x, y, 'b o', x, f, 'r')
grid on
```

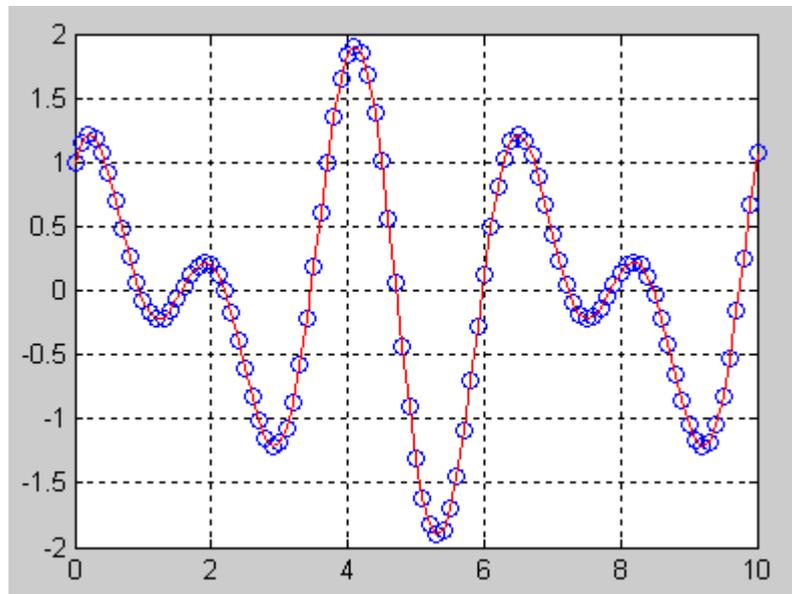


Рис. 2.

Аппроксимация периодической функции на основе быстрого преобразования Фурье. Функция **yp = interpft(y, n)** возвращает одномерный массив чисел, который является периодической функцией, определенной в n точках и аппроксимирующей одномерный массив y . Если $\text{length}(x) = m$, а интервал дискретности dx , то интервал дискретности для y определяется по формуле

$$dy = dx \cdot \frac{m}{n}, \text{ причем } n \text{ всегда превышает } m.$$

Пример 2. Рассмотрим аппроксимацию функции $y = \sin(x)$, которая задана 11 точками на интервале $[0; 10]$ (рис. 3):

```
x = 0:10;
y = sin(x);
xp = 0:0.25:10;
yp = interpft(y, 41);
plot(x, y, 'bo-', xp, yp, 'r')
grid on
```

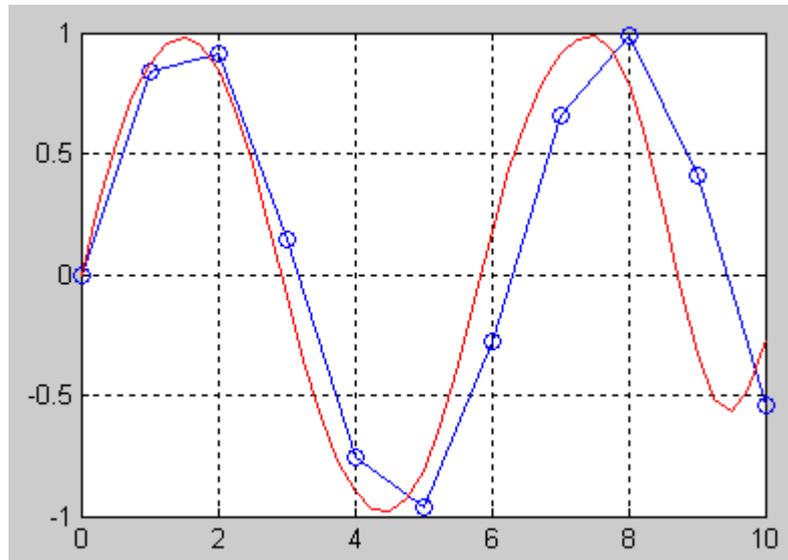


Рис. 3.

Лекция № 19

Matlab: Создание пользовательского интерфейса

Пример 1. Главное меню программы:

```
clear all;
id=0;
while not(id==2); % выход из программы
id=menu(['Интерполяция и аппроксимация функций'];{''},...
        '1. Ввод исходных данных',...
        '2. Выход из программы');
```

```
switch id
```

```
case {1}
```

```
clc; % очистка командного окна
```

```
id=1;
```

```
data_fun; % запуск файла ввода исходных данных
```

```
end;
```

```
end;
```

Пример 2. Организация ввода исходных данных с клавиатуры:

```
disp(' '); % пустая строка
```

```
disp('Начальное значение!');
```

Информационные технологии в менеджменте

```
xn=input([sprintf(' xn%g') ' =']); % в переменную xn заносим начальное
значение диапазона изменения аргумента, введенное с клавиатуры
disp(' ');
disp ('Функция:');
y=input([sprintf(' y%g') ' =']); % в переменную y заносим значение функции,
введенное с клавиатуры
```

Лекция № 20

Пакеты расширения Matlab

1. **Вывод списка пакетов расширения**
2. **Обзор пакетов расширения Matlab**

1. Вывод списка пакетов расширения

Пакеты расширения системы MATLAB являются основными средствами профессионального расширения системы и ее адаптации под решение определенных классов математических и научно-технических задач. Несомненно, что хотя бы части из этих пакетов должен быть посвящен отдельный учебный курс или справочник, быть может, и не один. За рубежом по большинству таких расширений опубликованы отдельные книги, а объем документации по ним составляет сотни мегабайт.

Полный состав системы MATLAB содержит ряд компонентов, название, номер версии и дату создания которых можно вывести на просмотр командой **ver**:

```
>> ver
```

```
-----
MATLAB Version 6.5.0.180913a (R13)
```

```
MATLAB License Number: 0
```

```
Operating System: Microsoft Windows XP Version 5.1 (Build 2600: Service Pack 3)
```

```
Java VM Version: Java 1.3.1_01 with Sun Microsystems Inc. Java HotSpot(TM)
```

```
Client VM
```

```
-----
MATLAB                Version 6.5      (R13)
Simulink              Version 5.0      (R13)
Aerospace Blockset   Version 1.0.1    (R13)
CDMA Reference Blockset   Version 1.1      (R13)
Communications Blockset   Version 2.5      (R13)
Communications Toolbox   Version 2.1      (R13)
Control System Toolbox   Version 5.2      (R13)
Curve Fitting Toolbox   Version 1.1      (R13)
DSP Blockset          Version 5.0      (R13)
Data Acquisition Toolbox   Version 2.2      (R13)
Database Toolbox       Version 2.2.1    (R13)
Datafeed Toolbox       Version 1.3.1    (R13)
Dials & Gauges Blockset   Version 1.1.2    (R13)
Embedded Target for Motorola MPC555   Version 1.0.1    (R13)
Embedded Target for Texas Instrumen...   Version 1.0      (R13)
Excel Link             Version 2.0      (R13)
```

Информационные технологии в менеджменте

Filter Design Toolbox	Version 2.2	(R13)
Financial Derivatives Toolbox	Version 2.0	(R13)
Financial Time Series Toolbox	Version 2.0	(R13)
Financial Toolbox	Version 2.2.1	(R13)
Fixed-Point Blockset	Version 4.0	(R13)
Fuzzy Logic Toolbox	Version 2.1.2	(R13)
GARCH Toolbox	Version 1.0.2	(R13)
Image Processing Toolbox	Version 3.2	(R13)
Instrument Control Toolbox	Version 1.2	(R13)
LMI Control Toolbox	Version 1.0.8	(R13)
MATLAB COM Builder	Version 1.0	(R13)
MATLAB Compiler	Version 3.0	(R13)
MATLAB Excel Builder	Version 1.1	(R13)
MATLAB Link for Code Composer Studi...	Version 1.0	(R13)
MATLAB Report Generator	Version 1.3	(R13)
MATLAB Runtime Server Development Kit	Version 6.1.1	(R13)
MATLAB Web Server	Version 1.2.2	(R13)
Mapping Toolbox	Version 1.3	(R13)
Model Predictive Control Toolbox	Version 1.0.7	(R13)
Model-Based Calibration Toolbox	Version 1.1	(R13)
Mu-Analysis and Synthesis Toolbox	Version 3.0.7	(R13)
Neural Network Toolbox	Version 4.0.2	(R13)
Nonlinear Control Design Blockset	Version 1.1.6	(R13)
Optimization Toolbox	Version 2.2	(R13)
Partial Differential Equation Toolbox	Version 1.0.4	(R13)
Real-Time Windows Target	Version 2.2	(R13)
Real-Time Workshop	Version 5.0	(R13)
Real-Time Workshop Embedded Coder	Version 3.0	(R13)
Requirements Management Interface	Version 1.0.4	(R13)
Robust Control Toolbox	Version 2.0.9	(R13)
SB2SL (converts SystemBuild to Simu...	Version 2.5	(R13)
Signal Processing Toolbox	Version 6.0	(R13)
SimMechanics	Version 1.1	(R13)
SimPowerSystems	Version 2.3	(R13)
Simulink Performance Tools	Version 1.2	(R13)
Simulink Report Generator	Version 1.3	(R13)
Spline Toolbox	Version 3.1.1	(R13)
Stateflow	Version 5.0	(R13)
Stateflow Coder	Version 5.0	(R13)
Statistics Toolbox	Version 4.0	(R13)
Symbolic Math Toolbox	Version 2.1.3	(R13)
System Identification Toolbox	Version 5.0.2	(R13)
Virtual Reality Toolbox	Version 3.0	(R13)
Wavelet Toolbox	Version 2.2	(R13)
xPC Target	Version 2.0	(R13)
xPC Target Embedded Option	Version 2.0	(R13)

>>

Описание этих пакетов расширения в PDF-формате занимает много более десятка тысяч страниц. Ниже дано краткое описание основных пакетов расширения.

2. Обзор пакетов расширения Matlab

Пакет моделирования динамических систем (Simulink)

Пакет расширения **Simulink** служит для имитационного моделирования моделей, состоящих из графических блоков с заданными свойствами (параметрами). Компоненты моделей, в свою очередь, являются графическими блоками и моделями, которые содержатся в ряде библиотек и с помощью мыши могут переноситься в основное окно и соединяться друг с другом необходимыми связями. В состав моделей могут включаться источники сигналов различного вида, виртуальные регистрирующие приборы, графические средства анимации. Двойной щелчок мышью на блоке модели выводит окно со списком его параметров, которые пользователь может менять. Запуск имитации обеспечивает математическое моделирование построенной модели с наглядным визуальным представлением результатов.

Simulink составляет и решает уравнения состояния модели и позволяет подключать в нужные ее точки разнообразные виртуальные измерительные приборы.

Подсистема имитационного моделирования (Real Time Windows Target и Workshop)

Подключающаяся к **Simulink** подсистема имитационного моделирования в реальном масштабе времени (при наличии дополнительных аппаратных средств в виде плат расширения компьютера), представленная пакетами расширения **Real Time Windows Target** и **Workshop**, — мощное средство управления реальными объектами и системами. Кроме того, эти расширения позволяют создавать исполняемые коды моделей.

Достоинством такого моделирования является его математическая и физическая наглядность. В компонентах моделей **Simulink** можно задавать не только фиксированные параметры, но и математические соотношения, описывающие поведение моделей.

Пакет **Real-Time Workshop** обладает следующими возможностями:

- ◆ Обеспечивает автоматическую генерацию кодов для непрерывных, дискретных, событийно-управляемых и гибридных систем.
- ◆ Создает оптимизированный, переносимый, краткий и четкий код. Код содержит необходимые комментарии, и его легко модернизировать.
- ◆ Позволяет задавать метки сигналов и названия блоков, которые переносятся из названий моделей в генерируемый код.API, который использует пользовательские файлы для автоматического создания и загрузки объектных файлов.
- ◆ Содержит компилятор Target Language Compiler (TLC), позволяющий настраивать сгенерированный код. С помощью TLC вы можете выполнять подключенные S-функции или создавать новые алгоритмы для встроженных блоков.
- ◆ Обеспечивает интерактивную загрузку параметров из блок-диаграмм во внешние устройства для оперативной настройки системы.
- ◆ Имеет наглядный Graphical User Interface (GUI).
- ◆ Обеспечивает поддержку проектной среды Spectron Microsystems (SPOXworks).

Генераторы отчетов (Report Generator для MATLAB и Simulink)

Генераторы отчетов — дают информацию о работе системы MATLAB и пакета расширения Simulink. Это средство очень полезно при отладке сложных вычислительных алгоритмов или при моделировании сложных систем. Генераторы

отчетов запускаются командой **Report**. Отчеты могут быть представлены в виде программ и редактироваться.

Генераторы отчетов могут запускать входящие в отчеты команды и фрагменты программ и позволяют проконтролировать поведение сложных вычислений.

Средства для построения нейронных сетей (Neural Networks Toolbox)

Пакет прикладных программ, содержащих средства для построения нейронных сетей, базирующихся на поведении математического аналога нейрона. Пакет обеспечивает эффективную поддержку проектирования, обучения и моделирования множества известных сетевых парадигм, от базовых моделей персептрона до самых современных ассоциативных и самоорганизующихся сетей. Пакет может быть использован для исследования и применения нейронных сетей к таким задачам, как обработка сигналов, нелинейное управление и финансовое моделирование. Обеспечена возможность генерации переносимого С-кода с помощью **Real Time Workshop**.

В пакет включены более 15 известных типов сетей и обучающих правил, позволяющих пользователю выбирать наиболее подходящую для конкретного приложения или исследовательской задачи парадигму. Для каждого типа архитектуры и обучающих правил имеются функции инициализации, обучения, адаптации, создания и моделирования, демонстрации и пример приложения сети.

Для управляемых сетей можно выбрать прямую или рекуррентную архитектуру, используя множество обучающих правил и методов проектирования, таких как персептрон, обратное распространение, обратное распространение Левенберга, сети с радиальным базисом и рекуррентные сети. Можно легко изменять любые архитектуры, обучающие правила или переходные функции, добавлять новые, — и все это без написания программ.

Пакет теории нечетких множеств (Fuzzy Logic Toolbox)

Пакет прикладных программ **Fuzzy Logic** относится к теории нечетких (размытых) множеств. Обеспечивается поддержка современных методов нечеткой кластеризации и адаптивных нечетких нейронных сетей. Графические средства пакета позволяют интерактивно отслеживать особенности поведения системы.

Основные возможности пакета:

- определение переменных, нечетких правил и функций принадлежности;
- интерактивный просмотр нечеткого логического вывода;
- современные методы: адаптивный нечеткий вывод с использованием нейронных сетей, нечеткая кластеризация;
- интерактивное динамическое моделирование в **Simulink**;
- генерация переносимого Си кода с помощью **Real-Time Workshop**.

Пакет символьной математики (Symbolic Math Toolbox)

Пакет прикладных программ, дающих системе MATLAB принципиально новые возможности — возможности решения задач в символьном (аналитическом) виде, включая реализацию точной арифметики произвольной разрядности. Пакет базируется на применении ядра символьной математики одной из самых мощных систем компьютерной алгебры — Maple V R4. Обеспечивает выполнение символьного дифференцирования и интегрирования, вычисление сумм и произведений, разложение в ряды Тейлора и Маклорена, операции со степенными многочленами (полиномами), вычисление корней полиномов, решение в аналитическом виде нелинейных уравнений, всевозможные символьные

преобразования, подстановки и многое другое. Имеет команды прямого доступа к ядру системы Maple V.

Пакет позволяет готовить процедуры с синтаксисом языка программирования системы Maple V R4 и устанавливать их в системе MATLAB. К сожалению, по возможностям символьной математики пакет сильно уступает специализированным системам компьютерной алгебры, таким как новейшие версии Maple и Mathematica.

Пакеты математических вычислений. NAG Foundation Toolbox

В MATLAB входит множество пакетов расширения, усиливающих математические возможности системы, повышающих скорость, эффективность и точность вычислений.

Одна из самых мощных библиотек математических функций, созданная специальной группой The Numerical Algorithms Group, Ltd. Пакет содержит сотни новых функций. Названия функций и синтаксис их вызова заимствованы из известной библиотеки **NAG Foundation Library**. Вследствие этого опытные пользователи NAG ФОРТРАН могут без затруднений работать с пакетом NAG в MATLAB. Библиотека **NAG Foundation** предоставляет свои функции в виде объектных кодов и соответствующих m-файлов для их вызова. Пользователь может легко модифицировать эти MEX-файлы на уровне исходного кода.

Пакет обеспечивает следующие возможности:

- корни многочленов и модифицированный метод Лагерра;
- вычисление суммы ряда: дискретное и эрмитово-дискретное преобразование Фурье;
- обыкновенные дифференциальные уравнения: методы Адамса и Рунге-Кутты;
- уравнения в частных производных;
- интерполяция;
- вычисление собственных значений и векторов, сингулярных чисел, поддержка комплексных и действительных матриц;
- аппроксимация кривых и поверхностей: полиномы, кубические сплайны, полиномы Чебышева;
- минимизация и максимизация функций: линейное и квадратичное программирование, экстремумы функций нескольких переменных;
- разложение матриц;
- решение систем линейных уравнений;
- линейные уравнения (LAPACK);
- статистические расчеты, включая описательную статистику и распределения вероятностей;
- корреляционный и регрессионный анализ: линейные, многомерные и обобщенные линейные модели;
- многомерные методы: главных компонент, ортогональные вращения;
- генерация случайных чисел: нормальное распределение, распределения Пуассона, Вейбулла и Коши;
- непараметрические статистики: Фридмана, Крускала-Уоллиса, Манна-Уитни;
- временные ряды: одномерные и многомерные;
- аппроксимации специальных функций: интегральная экспонента, гамма-функция, функции Бесселя и Ганкеля.

Информационные технологии в менеджменте

Пакет для работы со сплайнами (Spline Toolbox)

Пакет прикладных программ для работы со сплайнами поддерживает одномерную, двумерную и многомерную сплайн-интерполяцию и аппроксимацию. Обеспечивает представление и отображение сложных данных и поддержку графики.

Пакет позволяет выполнять интерполяцию, аппроксимацию и преобразование сплайнов из В-формы в кусочно-полиномиальную, интерполяцию кубическими сплайнами и сглаживание, выполнение операций над сплайнами: вычисление производной, интеграла и отображение.

Пакет **Spline** оснащен программами работы с В-сплайнами, описанными в работе «A Practical Guide to Splines» Карлом Дебуром, создателем сплайнов и автором пакета **Spline**. Функции пакета в сочетании с языком MATLAB и подробным руководством пользователя облегчают понимание сплайнов и их эффективное применение к решению разнообразных задач.

В пакет включены программы для работы с двумя наиболее широко распространенными формами представления сплайнов: В-формой и кусочно-полиномиальной формой. В-форма удобна на этапе построения сплайнов, в то время как кусочно-полиномиальная форма более эффективна во время постоянной работы со сплайном. Пакет включает функции для создания, отображения, интерполяции, аппроксимации и обработки сплайнов в В-форме и в виде отрезков полиномов.

Пакет по статистике (Statistics Toolbox)

Пакет прикладных программ по статистике **Statistics Toolbox** расширяет возможности системы MATLAB в области реализации статистических вычислений и статистической обработки данных. Содержит весьма представительный набор средств генерации случайных чисел, векторов, матриц и массивов с различными законами распределения, а также множество статистических функций. Следует отметить, что наиболее распространенные статистические функции входят в состав ядра системы MATLAB (в том числе функции генерации случайных данных с равномерным и нормальным распределением). Основные возможности пакета:

- описательная статистика;
- распределения вероятностей;
- оценка параметров и аппроксимация;
- проверка гипотез;
- множественная регрессия;
- интерактивная пошаговая регрессия;
- моделирование Монте-Карло;
- аппроксимация на интервалах;
- статистическое управление процессами;
- планирование эксперимента;
- моделирование поверхности отклика;
- аппроксимация нелинейной модели;
- анализ главных компонент;
- статистические графики;
- графический интерфейс пользователя.

Пакет включает 20 различных распределений вероятностей, включая t (Стьюдента), F и χ^2 (Хи-квадрат). Подбор параметров, графическое отображение распределений и способ вычисления лучших аппроксимаций предоставляются для всех типов распределений. Предусмотрено множество интерактивных инструментов для динамической визуализации и анализа данных. Имеются

Информационные технологии в менеджменте

специализированные интерфейсы для моделирования поверхности отклика, визуализации распределений, генерации случайных чисел и линий уровня.

Пакет решения оптимизационных задач и нелинейных уравнений (Optimization Toolbox)

Пакет **Optimization Toolbox** предназначен для решения оптимизационных задач и систем нелинейных уравнений. Поддерживает основные методы оптимизации функций ряда переменных:

- безусловная оптимизация нелинейных функций;
- метод наименьших квадратов и нелинейная интерполяция;
- решение нелинейных уравнений;
- линейное программирование;
- квадратичное программирование;
- условная минимизация нелинейных функций;
- метод минимакса;
- многокритериальная оптимизация.

Разнообразные примеры демонстрируют эффективное применение функций пакета. С их помощью можно также сравнить, как одна и та же задача решается разными методами.

Пакет решения систем дифференциальных уравнений в частных производных (Partial Differential Equations Toolbox)

Пакет прикладных программ **Partial Differential Equations Toolbox** содержит множество функций для решения систем дифференциальных уравнений в частных производных. В пакете используется метод конечных элементов. Команды и графический интерфейс пакета могут быть использованы для математического моделирования уравнений в частных производных применительно к широкому классу инженерных и научных приложений, включая задачи сопротивления материалов, расчеты электромагнитных устройств, задачи тепломассопереноса и диффузии. Основные возможности пакета:

- полноценный графический интерфейс для обработки уравнений с частными производными второго порядка;
- автоматический и адаптивный выбор сетки;
- задание граничных условий: Дирихле, Неймана и смешанных;
- гибкая постановка задачи с использованием синтаксиса MATLAB;
- полностью автоматическое сеточное разбиение и выбор величины конечных элементов;
- нелинейные и адаптивные расчетные схемы;
- возможность визуализации полей различных параметров и функций решения, демонстрация принятого разбиения и анимационные эффекты.

Пакет интуитивно следует шести шагам решения дифференциальных уравнений в частных производных (PDE) с помощью метода конечных элементов. Эти шаги и соответствующие режимы пакета таковы: определение геометрии (режим рисования), задание граничных условий (режим граничных условий), выбор коэффициентов, определяющих задачу (режим PDE), дискретизация конечных элементов (режим сетки), задание начальных условий и решение уравнений (режим решения), последующая обработка решения (режим графика).

Пакеты анализа и синтеза систем управления (*Control System Toolbox, Nonlinear Control Design Toolbox, Robust Control Toolbox, Model Predictive Control Toolbox, (Мю)-Analysis and Synthesis, Stateflow, Quantitative Feedback Theory Toolbox, Fixed-Point Blockset*)

Control System. Пакет **Control System** предназначен для моделирования, анализа и проектирования систем автоматического управления — как непрерывных, так и дискретных. Функции пакета реализуют традиционные методы передаточных функций и современные методы пространства состояний. Частотные и временные отклики, диаграммы расположения нулей и полюсов могут быть быстро вычислены и отображены на экране. В пакете реализованы:

- полный набор средств для анализа МИМО-систем (множество входов — множество выходов) систем;
- временные характеристики: передаточная и переходная функции, реакция на произвольное воздействие;
- частотные характеристики: диаграммы Боде, Николса, Найквиста и др.;
- разработка обратных связей;
- проектирование LQR/LQE-регуляторов;
- характеристики моделей: управляемость, наблюдаемость, понижение порядка моделей;
- поддержка систем с запаздыванием.

Дополнительные функции построения моделей позволяют конструировать более сложные модели. Временной отклик может быть рассчитан для импульсного входа, единичного скачка или произвольного входного сигнала. Имеются также функции для анализа сингулярных чисел.

Интерактивная среда для сравнения временного и частотного отклика систем предоставляет пользователю графические управляющие элементы для одновременного отображения откликов и переключения между ними. Можно вычислять различные характеристики откликов, такие как время разгона и время регулирования.

Пакет **Control System** содержит средства для выбора параметров обратной связи. Среди традиционных методов: анализ особых точек, определение коэффициента усиления и затухания. Среди современных методов: линейно-квадратичное регулирование и др. Пакет **Control System** включает большое количество алгоритмов для проектирования и анализа систем управления. Кроме того, он обладает настраиваемым окружением и позволяет создавать свои собственные m-файлы.

Nonlinear Control Design Toolbox. Nonlinear Control Design (NCD) Blockset реализует метод динамической оптимизации для проектирования систем управления. Этот инструмент, разработанный для использования с Simulink, автоматически настраивает системные параметры, основываясь на определенных пользователем ограничениях на временные характеристики. Пакет использует перенос объектов мышью для изменения временных ограничений прямо на графиках, что позволяет легко настраивать переменные и указывать неопределенные параметры, обеспечивает интерактивную оптимизацию, реализует моделирование методом Монте-Карло, поддерживает проектирование SISO- (один вход — один выход) и МИМО-систем управления, позволяет моделировать подавление помех, слежение и другие типы откликов, поддерживает проблемы повторяющегося параметра и задачи управления

Информационные технологии в менеджменте

системами с запаздыванием, позволяет осуществлять выбор между удовлетворенными и недостижимыми ограничениями.

Robust Control Toolbox. Пакет **Robust Control** включает средства для проектирования и анализа многопараметрических устойчивых систем управления. Это системы с ошибками моделирования, динамика которых известна не полностью или параметры которых могут изменяться в ходе моделирования. Мощные алгоритмы пакета позволяют выполнять сложные вычисления с учетом изменения множества параметров. Возможности пакета:

- синтез LQG-регуляторов на основе минимизации равномерной и интегральной нормы;
- многопараметрический частотный отклик;
- построение модели пространства состояний;
- преобразование моделей на основе сингулярных чисел;
- понижение порядка модели;
- спектральная факторизация.

Пакет **Robust Control** базируется на функциях пакета **Control System**, одновременно предоставляя усовершенствованный набор алгоритмов для проектирования систем управления. Пакет обеспечивает переход между современной теорией управления и практическими приложениями. Он имеет множество функций, реализующих современные методы проектирования и анализа многопараметрических робастных регуляторов.

Проявления неопределенностей, нарушающих устойчивость систем, многообразны — шумы и возмущения в сигналах, неточность модели передаточной функции, немоделируемая нелинейная динамика. Пакет **Robust Control** позволяет оценить многопараметрическую границу устойчивости при различных неопределенностях. Среди используемых методов: алгоритм Перрона, анализ особенностей передаточных функций и др.

Пакет **Robust Control** обеспечивает различные методы проектирования обратных связей, среди которых: **LQR**, **LQG**, **LQG/LTR** и др. Необходимость понижения порядка модели возникает в нескольких случаях: понижение порядка объекта, понижение порядка регулятора, моделирование больших систем. Качественная процедура понижения порядка модели должна быть численно устойчива. Процедуры, включенные в пакет **Robust Control**, успешно справляются с этой задачей.

Model Predictive Control Toolbox. Пакет **Model Predictive Control** содержит полный набор средств для реализации стратегии предиктивного (упреждающего) управления. Эта стратегия была разработана для решения практических задач управления сложными многоканальными процессами при наличии ограничений на переменные состояния и управление. Методы предиктивного управления используются в химической промышленности и для управления другими непрерывными процессами. Пакет обеспечивает:

- моделирование, идентификацию и диагностику систем;
- поддержку MISO (много входов — один выход), MIMO, переходных характеристик, моделей пространства состояний;
- системный анализ;
- конвертирование моделей в различные формы представления (пространство состояний, передаточные функции);
- предоставление учебников и демонстрационных примеров.

Предиктивный подход к задачам управления использует явную линейную динамическую модель объекта для прогнозирования влияния будущих изменений

Информационные технологии в менеджменте

управляющих переменных на поведение объекта. Проблема оптимизации формулируется в виде задачи квадратичного программирования с ограничениями, решаемой на каждом такте моделирования заново. Пакет позволяет создавать и тестировать регуляторы как для простых, так и для сложных объектов.

Пакет содержит более полусотни специализированных функций для проектирования, анализа и моделирования динамических систем с использованием предикативного управления. Он поддерживает следующие типы систем: импульсные, непрерывные и дискретные по времени, пространство состояний. Обрабатываются различные виды возмущений. Кроме того, в модель могут быть явно включены ограничения на входные и выходные переменные.

Средства моделирования позволяют осуществлять слежение и стабилизацию. Средства анализа включают вычисление полюсов замкнутого контура, частотного отклика, другие характеристики системы управления. Для идентификации модели в пакете имеются функции взаимодействия с пакетом **System Identification**. Пакет также включает две функции **Simulink**, позволяющие тестировать нелинейные модели.

(Мю)-Analysis and Synthesis. Пакет **(Мю)-Analysis and Synthesis** содержит функции для проектирования устойчивых систем управления. Пакет использует оптимизацию в равномерной норме и сингулярный параметр. В этот пакет включен графический интерфейс для упрощения операций с блоками при проектировании оптимальных регуляторов. Свойства пакета:

- проектирование регуляторов, оптимальных в равномерной и интегральной норме;
- оценка действительного и комплексного сингулярного параметра **мю**;
- D-K-итерации для приближенного **мю**-синтеза;
- графический интерфейс для анализа отклика замкнутого контура;
- средства понижения порядка модели;
- непосредственное связывание отдельных блоков больших систем.

Модель пространства состояний может быть создана и проанализирована на основе системных матриц. Пакет поддерживает работу с непрерывными и дискретными моделями. Пакет обладает полноценным графическим интерфейсом, включающим в себя: возможность устанавливать диапазон вводимых данных, специальное окно для редактирования свойств D-K итераций и графическое представление частотных характеристик. Имеет функции для матричного сложения, умножения, различных преобразований и других операций над матрицами. Обеспечивает возможность понижения порядка моделей.

Stateflow. **Stateflow** — пакет моделирования событийно-управляемых систем, основанный на теории конечных автоматов. Этот пакет предназначен для использования вместе с пакетом моделирования динамических систем **Simulink**. В любую Simulink-модель можно вставить Stateflow-диаграмму (или SF-диаграмму), которая будет отражать поведение компонентов объекта (или системы) моделирования. SF-диаграмма является анимационной. По ее выделяющимся цветом блокам и связям можно проследить все стадии работы моделируемой системы или устройства и поставить ее работу в зависимость от тех или иных событий.

Для создания SF-диаграмм пакет имеет удобный и простой редактор, а также средства пользовательского интерфейса.

Stateflow дает возможность моделировать поведение сложных событийно-управляемых систем, базируясь на теории конечных автоматов, что позволяет пользователям **Simulink** применять событийно-управляемое поведение к их моде-

Информационные технологии в менеджменте

лям. Используя **Simulink** и **Stateflow** можно проектировать, моделировать и имитировать, в том числе с помощью потоковых диаграмм, работу системы управления в единой интегрированной среде.

Потоковые диаграммы позволяют легко создавать схемы традиционных программных структур, таких, как циклы **for** и операторы **if-then-else**, что превосходит возможности большинства графических сред программирования. **Stateflow** объединяет достоинства диаграмм состояний и структурных схем с потоковыми диаграммами. Эти свойства делают **Stateflow** идеальной средой для разработки систем управления в авто-, аэрокосмических и телекоммуникационных приложениях.

Quantitative Feedback Theory Toolbox. Пакет содержит функции для создания робастных (устойчивых) систем с обратной связью **QFT** (количественная теория обратных связей) — инженерный метод, использующий частотное представление моделей для удовлетворения различных требований к качеству при наличии неопределенных характеристик объекта. В основе метода лежит наблюдение, что обратная связь необходима в тех случаях, когда некоторые характеристики объекта неопределены и/или на его вход подаются неизвестные возмущения. Возможности пакета:

- оценка частотных границ неопределенности, присущей обратной связи;
- графический интерфейс пользователя, позволяющий оптимизировать процесс нахождения требуемых параметров обратной связи;
- функции для определения влияния различных блоков, вводимых в модель (мультиплексоров, сумматоров, петель обратной связи) при наличии неопределенностей;
- поддержка моделирования аналоговых и цифровых контуров обратной связи, каскадов и многоконтурных схем;
- разрешение неопределенности в параметрах объекта с использованием параметрических и непараметрических моделей или комбинации этих типов моделей.

Теория обратных связей является естественным продолжением классического частотного подхода к проектированию. Ее основная цель — проектирование простых регуляторов небольшого порядка с минимальной шириной полосы пропускания, удовлетворяющих качественным характеристикам при наличии неопределенностей.

Пакет позволяет вычислять различные параметры обратных связей, фильтров, проводить тестирование регуляторов, как в непрерывном, так и в дискретном пространстве. Имеет удобный графический интерфейс, позволяющий создавать простые регуляторы, удовлетворяющие требованиям пользователя.

QFT позволяет проектировать регуляторы, удовлетворяющие различным требованиям, несмотря на изменения параметров модели. Измеряемые данные могут быть непосредственно использованы для проектирования регуляторов, без необходимости идентификации сложного отклика системы.

Fixed-Point Blockset. Этот специальный пакет ориентирован на моделирование цифровых систем управления и цифровых фильтров в составе пакета **Simulink**. Специальный набор компонентов позволяет быстро переключаться между вычислениями с фиксированной и плавающей запятой (точкой). Можно указывать 8-, 16- или 32-битовую длину слова. Пакет обладает рядом полезных свойств:

- применение беззнаковой или двоичной арифметики;

Информационные технологии в менеджменте

- выбор пользователем положения двоичной точки;
- автоматическая установка положения двоичной точки;
- просмотр максимального и минимального диапазонов сигнала модели;
- переключение между вычислениями с фиксированной и плавающей точкой;
- коррекция переполнения и наличие ключевых компонентов для операций с фиксированной точкой; логические операторы, одно- и двумерные справочные таблицы.

Пакет для постановки и решения задач линейного программирования (LMI Control Toolbox)

Пакет **LMI (Linear Matrix Inequality) Control** обеспечивает интегрированную среду для постановки и решения задач линейного программирования. Предназначенный первоначально для проектирования систем управления пакет позволяет решать любые задачи линейного программирования практически в любой сфере деятельности, где такие задачи возникают. Основные возможности пакета:

- решение задач линейного программирования: задачи совместности ограничений, минимизация линейных целей при наличии линейных ограничений, минимизация собственных значений;
- исследование задач линейного программирования;
- графический редактор задач линейного программирования;
- задание ограничений в символьном виде;
- многокритериальное проектирование регуляторов;
- проверка устойчивости: квадратичная устойчивость линейных систем, устойчивость по Ляпунову, проверка критерия Попова для нелинейных систем.

Пакет **LMI Control** содержит современные симплексные алгоритмы для решения задач линейного программирования. Использует структурное представление линейных ограничений, что повышает эффективность и минимизирует требования к памяти. Пакет имеет специализированные средства для анализа и проектирования систем управления на основе линейного программирования.

С помощью решателей задач линейного программирования можно легко выполнять проверку устойчивости динамических систем и систем с нелинейными компонентами. Ранее этот вид анализа считался слишком сложным для реализации. Пакет позволяет даже такое комбинирование критериев, которое ранее считалось слишком сложным и разрешимым лишь с помощью эвристических подходов.

Пакет является мощным средством для решения выпуклых задач оптимизации, возникающих в таких областях, как управление, идентификация, фильтрация, структурное проектирование, теория графов, интерполяция и линейная алгебра. Пакет **LMI Control** включает два вида графического интерфейса пользователя: редактор задачи линейного программирования (**LMI Editor**) и интерфейс **Magshape**. LMI Editor позволяет задавать ограничения в символьном виде, а Magshape обеспечивает пользователя удобными средствами работы с пакетом.

Пакеты идентификации систем (System Identification Toolbox, Frequency Domain System Identification Toolbox)

System Identification Toolbox. Пакет **System Identification** содержит средства для создания математических моделей динамических систем на основе

Информационные технологии в менеджменте

наблюдаемых входных и выходных данных. Он имеет гибкий графический интерфейс, помогающий организовать данные и создавать модели. Методы идентификации, входящие в пакет, применимы для решения широкого класса задач, от проектирования систем управления и обработки сигналов до анализа временных рядов и вибрации. Основные свойства пакета:

- простой и гибкий интерфейс;
- предварительная обработка данных, включая предварительную фильтрацию, удаление трендов и смещений;
- выбор диапазона данных для анализа;
- методы авторегрессии;
- анализ отклика во временной и частотной области;
- отображение нулей и полюсов передаточной функции системы;
- анализ невязок при тестировании модели;
- построение сложных диаграмм, таких как диаграмма Найквиста и др.

Графический интерфейс упрощает предварительную обработку данных, а также диалоговый процесс идентификации модели. Возможна также работа с пакетом в командном режиме и с применением расширения **Simulink**. Операции загрузки и сохранения данных, выбора диапазона, удаления смещений и трендов выполняются с минимальными усилиями и находятся в главном меню.

Представление данных и идентифицированных моделей организовано графически таким образом, что в процессе интерактивной идентификации пользователь легко может вернуться к предыдущему шагу работы. Для новичков существует возможность просматривать следующие возможные шаги. Специалисту графические средства позволяют отыскать любую из ранее полученных моделей и оценить ее качество в сравнении с другими моделями.

Начав с измерения выхода и входа, можно создать параметрическую модель системы, описывающую ее поведение в динамике. Пакет поддерживает все традиционные структуры моделей, включая авторегрессию, структуру Бокса-Дженкинса и др. Он поддерживает линейные модели пространства состояний, которые могут быть определены как в дискретном, так и в непрерывном пространстве. Эти модели могут включать произвольное число входов и выходов. В пакет включены функции, которые можно использовать как тестовые данные для идентифицированных моделей. Идентификация линейных моделей широко используется при проектировании систем управления, когда требуется создать модель объекта. В задачах обработки сигналов модели могут быть использованы для адаптивной обработки сигналов. Методы идентификации успешно применяются и для финансовых приложений.

Frequency Domain System Identification Toolbox. Пакет **Frequency Domain System Identification** предоставляет специализированные средства для идентификации линейных динамических систем по их временному или частотному отклику. Частотные методы направлены на идентификацию непрерывных систем, что является мощным дополнением к более традиционной дискретной методике. Методы пакета могут быть применены к таким задачам, как моделирование электрических, механических и акустических систем. Свойства пакета:

- периодические возмущения, пик-фактор, оптимальный спектр, псевдослучайные и дискретные двоичные последовательности;
- расчет доверительных интервалов амплитуды и фазы, нулей и полюсов;
- идентификация непрерывных и дискретных систем с неизвестным запаздыванием;

Информационные технологии в менеджменте

- диагностика модели, включая моделирование и вычисление невязок;
- преобразование моделей в формат **System Identification Toolbox** и обратно.

Используя частотный подход, можно добиться наилучшей модели в частотной области; избежать ошибок дискретизации; легко выделять постоянную составляющую сигнала; существенно улучшить отношение сигнал/шум. Для получения возмущающих сигналов пакет предоставляет функции генерации двоичных последовательностей, минимизации величины пика и улучшения спектральных характеристик. Пакетом обеспечивается идентификация непрерывных и дискретных линейных статических систем, автоматическая генерация входных сигналов, а также графическое изображение нулей и полюсов передаточной функции результирующей системы. Функции для тестирования модели включают вычисление невязок, передаточных функций, нулей и полюсов, прогонку модели с использованием тестовых данных.

Пакет для построения и моделирования телекоммуникационных устройств (Communications Toolbox)

Пакет **Communications Toolbox** является как бы самостоятельным средством для моделирования коммуникационных систем и устройств. Он позволяет вести разработку, анализ и тестирование моделей цифровых и аналоговых систем и устройств связи и передачи информации. В то же время он является частью пакета **Simulink** и включает в себя более 100 функций Matlab и около 150 компонентов **Simulink** для разработки и моделирования таких систем, как устройства радиосвязи, модемы и устройства хранения информации. Пакет является прекрасным средством для использования в научных разработках в области коммуникационных устройств, а также средством обучения студентов по специальностям, связанным с информационными технологиями.

Пакет **Communications** обеспечивает возможности моделирования следующих систем и устройств:

- ◆ Кодирования и оцифровки
- ◆ Контроля ошибок при кодировании
- ◆ Модуляции и демодуляции
- ◆ Фильтрации при передаче и приеме
- ◆ Систем синхронизации
- ◆ Аналоговых и цифровых систем фазовой автоподстройки частоты
- ◆ Коллективного доступа
- ◆ Полей Галуа
- ◆ Генераторов сложных сигналов
- ◆ Устройств анализа и построения графиков

Пакет **Communications** имеет обширную библиотеку блоков для создания линейных и нелинейных, дискретных, непрерывных, гибридных и иных моделей. Пакет наделен следующими свойствами:

- ◆ Иерархическая структура моделей с неограниченной вложенностью
- ◆ Скалярные и векторные связи
- ◆ Средство для создания пользовательских блоков и библиотек
- ◆ Интерактивное моделирование с «живым» отображением на экране
- ◆ Семь методов интегрирования с фиксированным и переменным шагом
- ◆ Возможность линеаризации задач
- ◆ Возможность моделирования методом Монте-Карло
- ◆ Анализ устойчивости и определение точек равновесия
- ◆ Различные способы вывода на экран и библиотека входных сигналов

Пакет для проектирования устройств, использующих процессоры цифровой обработки сигналов (Digital Signal Processing (DSP) Blockset)

Пакет прикладных программ **Digital Signal Processing (DSP) Blockset** предназначен для проектирования устройств, использующих процессоры цифровой обработки сигналов. Это, прежде всего, высокоэффективные цифровые фильтры с заданной или адаптируемой к параметрам сигналов частотной характеристикой (АЧХ). Результаты моделирования и проектирования цифровых устройств с помощью этого пакета могут использоваться для построения высокоэффективных цифровых фильтров на современных микропроцессорах цифровой обработки сигналов.

Блоки **DSP** представляют собой интуитивные графические инструменты быстрого проектирования, моделирования и макетирования цифровых систем обработки сигналов. Библиотека содержит более 100 блоков для создания и анализа моделей DSP.

В сочетании с Matlab и **Simulink** библиотека блоков **DSP** упрощает создание DSP-систем, интегрируя разработку алгоритмов и моделирование в реальном масштабе времени.

Библиотека **DSP Blockset** включает следующие функции обработки сигналов:

- ◆ Быстрое прямое и обратное преобразование Фурье, фильтры и средства буферизации.
- ◆ Введение новых блоков — переключателей и счетчиков.
- ◆ Поддержка выборочной обработки и возможность расширения окна.
- ◆ Комплекс математических операций по обработке сигналов с определением величины, фазы, гармоника, сопряжением и разделением сигналов.
- ◆ Блоки векторной математики и статистики, обеспечивающие быстрые вычисления и более совершенную обработку сложных сигналов.
- ◆ Адаптивное и множественное фильтрование с дополнительным использованием FIR, IIR и других блоков для создания фильтров.
- ◆ Включение блоков для моделирования обработки сигналов во временной и частотной областях.
- ◆ Блоки спектрального анализа, реализующие метод Велша (Welch)
- ◆ Новая «истинная библиотека» упрощает динамическое связывание, обновление и разделение блоков.

Пакеты для обработки сигналов и изображений (Signal Processing Toolbox, Higher-Order Spectral Analysis Toolbox, Image Processing Toolbox)

Signal Processing Toolbox. Пакет **Signal Processing** обеспечивает чрезвычайно обширные возможности создания программ обработки сигналов для современных научных и технических приложений. В пакете используется разнообразная техника фильтрации и новейшие алгоритмы спектрального анализа. Пакет содержит модули для разработки линейных систем и анализа временных рядов. Пакет будет полезен, в частности, в таких областях, как обработка аудио- и видеоинформации, телекоммуникации, геофизика, задачи управления в реальном режиме времени, экономика, финансы и медицина. Основные свойства пакета:

- моделирование сигналов и линейных систем;
- проектирование, анализ и реализация цифровых и аналоговых фильтров;
- быстрое преобразование Фурье, дискретное косинусное и другие преобразования;

Информационные технологии в менеджменте

- оценка спектров и статистическая обработка сигналов;
- параметрическая обработка временных рядов;
- генерация сигналов различной формы.

Пакет **Signal Processing** — идеальная оболочка для анализа и обработки сигналов. В нем используются проверенные практикой алгоритмы, выбранные по критериям максимальной эффективности и надежности. Пакет содержит широкий спектр алгоритмов для представления сигналов и линейных моделей. Этот набор позволяет пользователю достаточно гибко подходить к созданию сценария обработки сигналов. Пакет включает алгоритмы для преобразования модели из одного представления в другое.

Пакет **Signal Processing** включает полный набор методов для создания цифровых фильтров с разнообразными характеристиками. Он позволяет быстро разрабатывать фильтры верхних и нижних частот, полосовые пропускающие и задерживающие фильтры, многополосные фильтры, в том числе фильтры Чебышева, Юла-Уолкера, эллиптические и др.

Графический интерфейс позволяет проектировать фильтры, задавая требования к ним в режиме переноса объектов мышью. В пакет включены следующие новые методы проектирования фильтров:

- обобщенный метод Чебышева для создания фильтров с нелинейной фазовой характеристикой, комплексными коэффициентами или произвольным откликом. Алгоритм разработан Макленаном и Карамом в 1995 г.;
- метод наименьших квадратов с ограничениями позволяет пользователю явно контролировать максимальную ошибку (сглаживание);
- метод расчета минимального порядка фильтра с окном Кайзера;
- обобщенный метод Баттерворта для проектирования низкочастотных фильтров с максимально однородными полосами пропускания и затухания.

Основанный на оптимальном алгоритме быстрого преобразования Фурье пакет **Signal Processing** обладает непревзойденными характеристиками для частотного анализа и спектральных оценок. Пакет включает функции для вычисления дискретного преобразования Фурье, дискретного косинусного преобразования, преобразования Гильберта и других преобразований, часто применяемых для анализа, кодирования и фильтрации. В пакете реализованы такие методы спектрального анализа как метод Вельха, метод максимальной энтропии и др.

Графический интерфейс позволяет просматривать и визуально оценивать характеристики сигналов, проектировать и применять фильтры, производить спектральный анализ, исследуя влияние различных методов и их параметров на получаемый результат. Графический интерфейс особенно полезен для визуализации временных рядов, спектров, временных и частотных характеристик, расположения нулей и полюсов передаточных функций систем.

Пакет **Signal Processing** является основой для решения многих других задач. Например, комбинируя его с пакетом **Image Processing**, можно обрабатывать и анализировать двумерные сигналы и изображения. В паре с пакетом **System Identification** пакет **Signal Processing** позволяет выполнять параметрическое моделирование систем во временной области. В сочетании с пакетами **Neural Network** и **Fuzzy Logic** может быть создано множество средств для обработки данных или выделения классификационных характеристик. Средство генерации сигналов позволяет создавать импульсные сигналы различной формы.

Higher-Order Spectral Analysis Toolbox. Пакет **Higher-Order Spectral Analysis** содержит специальные алгоритмы для анализа сигналов с использованием моментов высшего порядка. Пакет предоставляет широкие возможности для анализа негауссовых сигналов, так как содержит алгоритмы, пожалуй, самых передовых методов для анализа и обработки сигналов. Основные возможности пакета:

- оценка спектров высокого порядка;
- традиционный или параметрический подход;
- восстановление амплитуды и фазы;
- адаптивное линейное прогнозирование;
- восстановление гармоник;
- оценка запаздывания;
- блочная обработка сигналов.

Пакет **Higher-Order Spectral Analysis** позволяет анализировать сигналы, поврежденные негауссовым шумом, и процессы, происходящие в нелинейных системах. Спектры высокого порядка, определяемые в терминах моментов высокого порядка сигнала, содержат дополнительную информацию, которую невозможно получить, пользуясь только анализом автокорреляции или спектра мощности сигнала.

Спектры высокого порядка позволяют:

- подавить аддитивный цветной гауссов шум;
- идентифицировать неминимально-фазовые сигналы;
- выделить информацию, обусловленную негауссовым характером шума;
- обнаружить и проанализировать нелинейные свойства сигналов.

Возможные приложения спектрального анализа высокого порядка включают акустику, биомедицину, эконометрию, сейсмологию, океанографию, физику плазмы, радары и локаторы. Основные функции пакета поддерживают спектры высокого порядка, взаимную спектральную оценку, линейные модели прогноза и оценку запаздывания.

Image Processing Toolbox. Пакет **Image Processing** предоставляет ученым, инженерам и даже художникам широкий спектр средств для цифровой обработки и анализа изображений. Будучи тесно связанным со средой разработки приложений MATLAB, пакет **Image Processing Toolbox** освобождает вас от выполнения длительных операций кодирования и отладки алгоритмов, позволяя сосредоточить усилия на решении основной научной или практической задачи. Основные свойства пакета:

- восстановление и выделение деталей изображений;
- работа с выделенным участком изображения;
- анализ изображения;
- линейная фильтрация;
- преобразование изображений;
- геометрические преобразования;
- увеличение контрастности важных деталей;
- бинарные преобразования;
- обработка изображений и статистика;
- цветовые преобразования;
- изменение палитры;
- преобразование типов изображений.

Информационные технологии в менеджменте

Пакет **Image Processing** дает широкие возможности для создания и анализа графических изображений в среде MATLAB. Этот пакет обеспечивает чрезвычайно гибкий интерфейс, позволяющий манипулировать изображениями, интерактивно разрабатывать графические картины, визуализировать наборы данных и аннотировать результаты для технических описаний, докладов и публикаций. Гибкость, соединение алгоритмов пакета с такой особенностью MATLAB, как матрично-векторное описание делают пакет очень удачно приспособленным для решения практически любых задач по разработке и представлению графики. В MATLAB входят специально разработанные процедуры, позволяющие повысить эффективность графической оболочки. Можно отметить, в частности, такие особенности:

- интерактивная отладка при разработке графики;
- профилировщик для оптимизации времени выполнения алгоритма;
- средства построения интерактивного графического интерфейса пользователя (**GUI Builder**) для ускорения разработки GUI-шаблонов, позволяющие настраивать его под задачи пользователя.

Этот пакет позволяет пользователю тратить значительно меньше времени и сил на создание стандартных графических изображений и, таким образом, сконцентрировать усилия на важных деталях и особенностях изображений.

MATLAB и пакет **Image Processing** максимально приспособлены для развития, внедрения новых идей и методов пользователя. Для этого имеется набор сопрягаемых пакетов, направленных на решение всевозможных специфических задач и задач в нетрадиционной постановке.

Пакет **Image Processing** в настоящее время интенсивно используется в более чем 4000 компаниях и университетах по всему миру. При этом имеется очень широкий круг задач, которые пользователи решают с помощью данного пакета, например космические исследования, военные разработки, астрономия, медицина, биология, робототехника, материаловедение, генетика и т. д.

Пакет исследования многомерных нестационарных явлений с помощью вейвлетов (Wavelet Toolbox)

Пакет **Wavelet** предоставляет пользователю полный набор программ для исследования многомерных нестационарных явлений с помощью вейвлетов (коротких волновых пакетов), методы пакета **Wavelet** расширяют возможности пользователя в тех областях, где обычно применяется техника Фурье-разложения. Пакет может быть полезен для таких приложений, как обработка речи и аудиосигналов, телекоммуникации, геофизика, финансы и медицина. Основные свойства пакета:

- усовершенствованный графический пользовательский интерфейс и набор команд для анализа, синтеза, фильтрации сигналов и изображений;
- преобразование многомерных непрерывных сигналов;
- дискретное преобразование сигналов;
- декомпозиция и анализ сигналов и изображений;
- широкий выбор базисных функций, включая коррекцию граничных эффектов;
- пакетная обработка сигналов и изображений;
- анализ пакетов сигналов, основанный на энтропии;
- фильтрация с возможностью установления жестких и нежестких порогов;
- оптимальное сжатие сигналов.

Пользуясь пакетом, можно анализировать такие особенности, которые упускают другие методы анализа сигналов, т. е. тренды, выбросы, разрывы в производных высоких порядков. Пакет позволяет сжимать и фильтровать сигналы без явных потерь даже в тех случаях, когда нужно сохранить и высоко – и низкочастотные компоненты сигнала. Имеются алгоритмы сжатия и фильтрации и для пакетной обработки сигналов. Программы сжатия выделяют минимальное число коэффициентов, представляющих исходную информацию наиболее точно, что очень важно для последующих стадий работы системы сжатия. В пакет включены следующие базисные наборы вейвлетов: биортогональный, Хаара, «Мексиканская шляпа», Майера и др. Вы также можете добавить в пакет свои собственные базисы.

Пакет по финансово-экономическим расчетам (Financial Toolbox)

Пакет **Financial Toolbox** содержит множество функций по расчету сложных процентов, операций по банковским вкладам, вычисления прибыли и многое другое. К сожалению, из-за многочисленных (хотя, в общем-то, не слишком принципиальных) различий в финансово-экономических формулах его применение в наших условиях не всегда разумно — есть множество отечественных программ для таких расчетов, — например «Бухгалтерия 1С». Но если вы хотите подключиться к базам данных агентств финансовых новостей — **Bloom-berg, IDC** через пакет **Datafeed Toolbox** MATLAB, то, конечно, обязательно пользуйтесь и финансовыми пакетами расширения MATLAB.

Пакет **Financial** является основой для решения в MATLAB множества финансовых задач, от простых вычислений до полномасштабных распределенных приложений. Пакет **Financial** может быть использован для расчета процентных ставок и прибыли, анализа производных доходов и депозитов, оптимизации портфеля инвестиций. Основные возможности пакета:

- обработка данных;
- дисперсионный анализ эффективности портфеля инвестиций;
- анализ временных рядов;
- расчет доходности ценных бумаг и оценка курсов;
- статистический анализ и анализ чувствительности рынка;
- калькуляция ежегодного дохода и расчет денежных потоков;
- методы начисления износа и амортизационных отчислений.

Учитывая важность даты той или иной финансовой операции, в пакет **Financial** включены несколько функций для манипулирования датами и временем в различных форматах. Пакет **Financial** позволяет рассчитывать цены и доходы при инвестициях в облигации. Пользователь имеет возможность задавать нестандартные, в том числе нерегулярные и несовпадающие друг с другом, графики дебитных и кредитных операций и окончательного расчета при погашении векселей. Экономические функции чувствительности могут быть вычислены с учетом разновременных сроков погашения.

Алгоритмы пакета **Financial** для расчета показателей движения денежных средств и других данных, отражаемых в финансовых счетах, позволяют вычислять, в частности, процентные ставки по займам и кредитам, коэффициенты рентабельности, кредитные поступления и итоговые начисления, оценивать и прогнозировать стоимость инвестиционного портфеля, вычислять показатели износа и т. п. Функции пакета могут быть использованы с учетом положительных и отрицательных денежных потоков (**cash-flow**) (превышения денежных поступлений над платежами или денежных выплат над поступлениями соответственно).

Информационные технологии в менеджменте

Пакет **Financial** содержит алгоритмы, которые позволяют анализировать портфель инвестиций, динамику и экономические коэффициенты чувствительности. В частности, при определении эффективности инвестиций функции пакета позволяют сформировать портфель, удовлетворяющий классической задаче Г. Марковица. Пользователь может комбинировать алгоритмы пакета для вычисления коэффициентов Шарпе и ставок дохода. Анализ динамики и экономических коэффициентов чувствительности позволяет пользователю определить позиции для стрэддл-сделок, хеджирования и сделок с фиксированными ставками. Пакет **Financial** обеспечивает также обширные возможности для представления и презентации данных и результатов в виде традиционных для экономической и финансовой сфер деятельности графиков и диаграмм. Денежные средства могут по желанию пользователя отображаться в десятичном, банковском и процентном форматах.

Пакет для анализа географических данных, отображения карт (Mapping Toolbox)

Пакет **Mapping** предоставляет графический и командный интерфейс для анализа географических данных, отображения карт и доступа к внешним источникам данных по географии. Кроме того, пакет пригоден для работы с множеством широко известных атласов. Все эти средства в комбинации с MATLAB предоставляют пользователям все условия для продуктивной работы с научными географическими данными. Основные возможности пакета:

- визуализация, обработка и анализ графических и научных данных;
- более 60 проекций карт (прямые и инверсные);
- проектирование и отображение векторных, матричных и составных карт;
- графический интерфейс для построения и обработки карт и данных;
- глобальные и региональные атласы данных и сопряжение с правительственными данными высокого разрешения;
- функции географической статистики и навигации;
- трехмерное представление карт со встроенными средствами подсветки и затенения;
- конвертеры для популярных форматов географических данных: DCW, TIGER, ETOPO5.

Пакет **Mapping** включает более 60 наиболее широко известных проекций, включая цилиндрическую, псевдоцилиндрическую, коническую, поликоническую и псевдоконическую, азимутальную и псевдоазимутальную. Возможны прямые и обратные проекции, а также нестандартные виды проекции, задаваемые пользователем.

В пакете **Mapping** *картой* называется любая переменная или множество переменных, отражающих или назначающих численное значение географической точке или области. Пакет позволяет работать с векторными, матричными и смешанными картами данных. Мощный графический интерфейс обеспечивает интерактивную работу с картами, например, возможность подвести указатель к объекту и, щелкнув на нем, получить информацию. Графический интерфейс **MARPTOOL** — полная среда разработки приложений для работы с картами.

Наиболее широко известные атласы мира, Соединенных Штатов, астрономические атласы входят в состав пакета. Географическая структура данных упрощает извлечение и обработку данных из атласов и карт. Географическая структура данных и функции взаимодействия с внешними географическими данными форматов **Digital Chart of the World (DCW)**, **TIGER**,

Информационные технологии в менеджменте

TBASE и **ЕТОР05** собраны воедино, чтобы обеспечить мощный и гибкий инструмент для доступа к уже существующим и будущим географическим базам данных. Тщательный анализ географических данных часто требует математических методов, работающих в сферической системе координат. Пакет **Mapping** снабжен подмножеством географических, статистических и навигационных функций для анализа географических данных. Функции навигации дают широкие возможности для выполнения задач перемещения, таких как позиционирование и планирование маршрутов.

Пакет моделирования энергетических систем (Power System Blockset)

Пакет **Power System Blockset** предназначен для моделирования мощных энергетических (в основном электротехнических) систем, таких как линии передачи, силовые ключи, регуляторы напряжения и тока, устройства управления электродвигателями различного типа и нагревательными системами.

Этот пакет обеспечивает моделирование широкого спектра энергетических систем и устройств — начиная с анализа простейших электрических цепей и кончая моделированием сложных преобразовательных устройств и даже целых электрических систем. Результаты моделирования отображаются разнообразными виртуальными измерительными приборами, такими как графопостроители, осциллографы и др.

Пакеты для получения данных от устройств (Data Acquisition Toolbox и Instrument Control Toolbox)

Data Acquisition Toolbox — пакет расширения, относящийся к области сбора данных через блоки, подключаемые к внутренней шине компьютера, функциональных генераторов, анализаторов спектра — словом, приборов, широко используемых в исследовательских целях для получения данных. Они поддерживаются соответствующей вычислительной базой. Блок **Instrument Control Toolbox** позволяет подключать приборы и устройства с последовательным интерфейсом и с интерфейсами Канал общего пользования и VXI.

Пакет для обеспечения обмена информацией с системами управления базами данных (Database toolbox)

Пакет **Database toolbox** обеспечивает обмен информацией с целым рядом систем управления базами данных через драйверы ODBC или JDBC:

- Oracle 7.3.3;
- Access 95 или 97 Microsoft;
- Microsoft SQL Server 6.5 или 7.0;
- Sybase Adaptive Server 11;
- Sybase (бывший Watcom) SQL Server Anywhere 5.0;
- IBM DB2 Universal 5.0;
- Informix 7.2.2;
- Computer Associates Ingres (все версии).

Все данные предварительно преобразуются в массив ячеек или массив структур. Визуальный конструктор (**Visual Query Builder**) позволяет составлять сколь угодно сложные запросы на диалектах языка SQL этих баз данных даже без знания SQL. В одном сеансе может быть открыто много неоднородных баз данных.

Пакет трехмерной анимации (Virtual Reality Toolbox)

Пакет **Virtual Reality Toolbox** позволяет осуществлять трехмерную анимацию и мультипликацию, в том числе моделей **Simulink**. Язык программирования — **VRML** — язык моделирования виртуальной реальности

(**Virtual Reality Modeling Language**). Просмотр анимации возможен с любого компьютера, оснащенного браузером с поддержкой VRML.

MATLAB Compiler

Компилятор для программ, создаваемых на языке программирования системы MATLAB. Транслирует коды этих программ в программы на языке Си++. Применение компилятора обеспечивает возможность создания исполняемых кодов (полностью законченных программ), время выполнения которых для программ с большим числом циклических операций уменьшается в 10—15 раз. Может интегрироваться в среду **Microsoft Visual Studio** и использоваться вместе с Visual C++. Помимо него можно использовать и другие компиляторы Си++.

Лекция № 21

Matlab: Simulink

- 1. Назначение пакета Simulink**
- 2. Библиотека компонентов пакета Simulink**

1. Назначение пакета Simulink

Пакет **Simulink** предназначен для моделирования линейных и нелинейных динамических систем и устройств, представленных своей функциональной схемой, в итоге строится *модель*. При этом возможны различные варианты моделирования: во временной области, в частотной области, с событийным управлением, на основе спектральных преобразований Фурье, с использованием метода Монте-Карло и т. д.

Simulink автоматизирует наиболее трудоемкий этап моделирования: он составляет и решает сложные системы алгебраических и дифференциальных уравнений, описывающих заданную функциональную схему (модель), обеспечивая удобный и наглядный визуальный контроль за поведением созданного пользователем *виртуального устройства*. Достаточно уточнить (если нужно) вид анализа и запустить **Simulink** в режиме *симуляции* (откуда и название пакета — **Simulink**) созданной модели системы или устройства. В русском языке слово «симуляция» носит несколько нарицательный оттенок, поэтому чаще используют термин «моделирование».

Для построения функциональной блок-схемы моделируемых устройств **Simulink** имеет обширную *библиотеку блочных компонентов*. Используя *палитры компонентов* (наборы) блок-схем, пользователь с помощью мыши переносит нужные компоненты с палитр на рабочий лист пакета **Simulink** и соединяет линиями входы и выходы блоков. Таким образом, создается блок-схема моделирования системы или устройства.

Simulink практически мгновенно меняет математическое описание модели по мере ввода новых блоков даже в том случае, когда этот процесс сопровождается сменой порядка системы уравнений и ведет к существенному качественному изменению поведения системы. Это следует отнести к одной из главных целей пакета **Simulink**.

Библиотека компонентов включает в себя источники сигналов с практически любыми временными зависимостями, масштабирующие, линейные и нелинейные преобразователи с разнообразными формами передаточных характеристик, квантующее устройство, интегрирующие и дифференцирующие блоки и так далее.

Информационные технологии в менеджменте

В библиотеке имеется целый набор виртуальных регистрирующих устройств — от простых измерителей типа вольтметра или амперметра до универсальных осциллографов, позволяющих просматривать временные зависимости выходных параметров моделируемых систем — например, токов и напряжений, перемещений, давлений и тому подобного. Имеется даже графопостроитель для построения фигур в полярной системе координат, например, фигур Лиссажу и фазовых портретов колебаний. **Simulink** имеет средства анимации и звукового сопровождения. А в дополнительных библиотеках можно отыскать и такие «дорогие приборы», как анализаторы спектра сложных сигналов, многоканальные самописцы и средства анимации графиков.

Средства анимации **Simulink** позволяют строить виртуальные физические лаборатории с наглядным представлением результатов моделирования. Возможности **Simulink** охватывают задачи математического моделирования сложных динамических систем в физике, электро- и радиотехнике, в биологии и химии, словом, — во всех областях науки и техники. Этим объясняется популярность данного пакета как в университетах и институтах, так и в научных лабораториях.

Как программное средство **Simulink** — типичный представитель *визуально-ориентированного языка программирования*. На всех этапах работы, особенно при подготовке моделей схем, пользователь практически не имеет дела с обычным программированием. Программа автоматически генерируется в процессе ввода выбранных блоков компонентов, их соединений и задания параметров компонентов.

Важное достоинство **Simulink** — это интеграция с системой Matlab и рядом других пакетов расширения, что обеспечивает, по существу, неограниченные возможности в применении **Simulink** для решения практически любых задач имитационного моделирования.

Понятие об S-функциях. Своими возможностями пакет **Simulink** во многом обязан специальному аппарату создания и применения так называемых системных *S-функций* (System Functions). Эти функции позволяют в ходе решения осуществлять сложные функциональные преобразования по различным математическим алгоритмам, например, алгоритмам решения систем дифференциальных уравнений.

Для разработки S-функций **Simulink** имеет специальный редактор. Создав S-функцию, пользователь фактически создает блок библиотеки, который может использоваться по всем правилам применения блоков. Блок можно переносить в окно редактирования с помощью мыши, менять заданную S-функцию, использовать необходимые связи между блоками и т. д.

2. Библиотека компонентов пакета Simulink

Запуск Simulink. Для запуска программы необходимо предварительно запустить пакет Matlab. После открытия основного окна программы Matlab нужно запустить программу **Simulink**. Это можно сделать одним из трех способов:

— нажать кнопку  (**Simulink**) на панели инструментов командного окна Matlab;

— в командной строке главного окна Matlab напечатать **Simulink** и нажать клавишу Enter на клавиатуре;

— выполнить команду **Open...** в меню **File** и открыть файл модели (mdl-файл). Этот вариант удобно использовать для запуска уже готовой и отлаженной

Информационные технологии в менеджменте

модели, когда требуется лишь провести расчеты и не нужно добавлять новые блоки в модель.

Использование первого и второго способов приводит к открытию окна обозревателя разделов библиотеки **Simulink** (рис. 1).

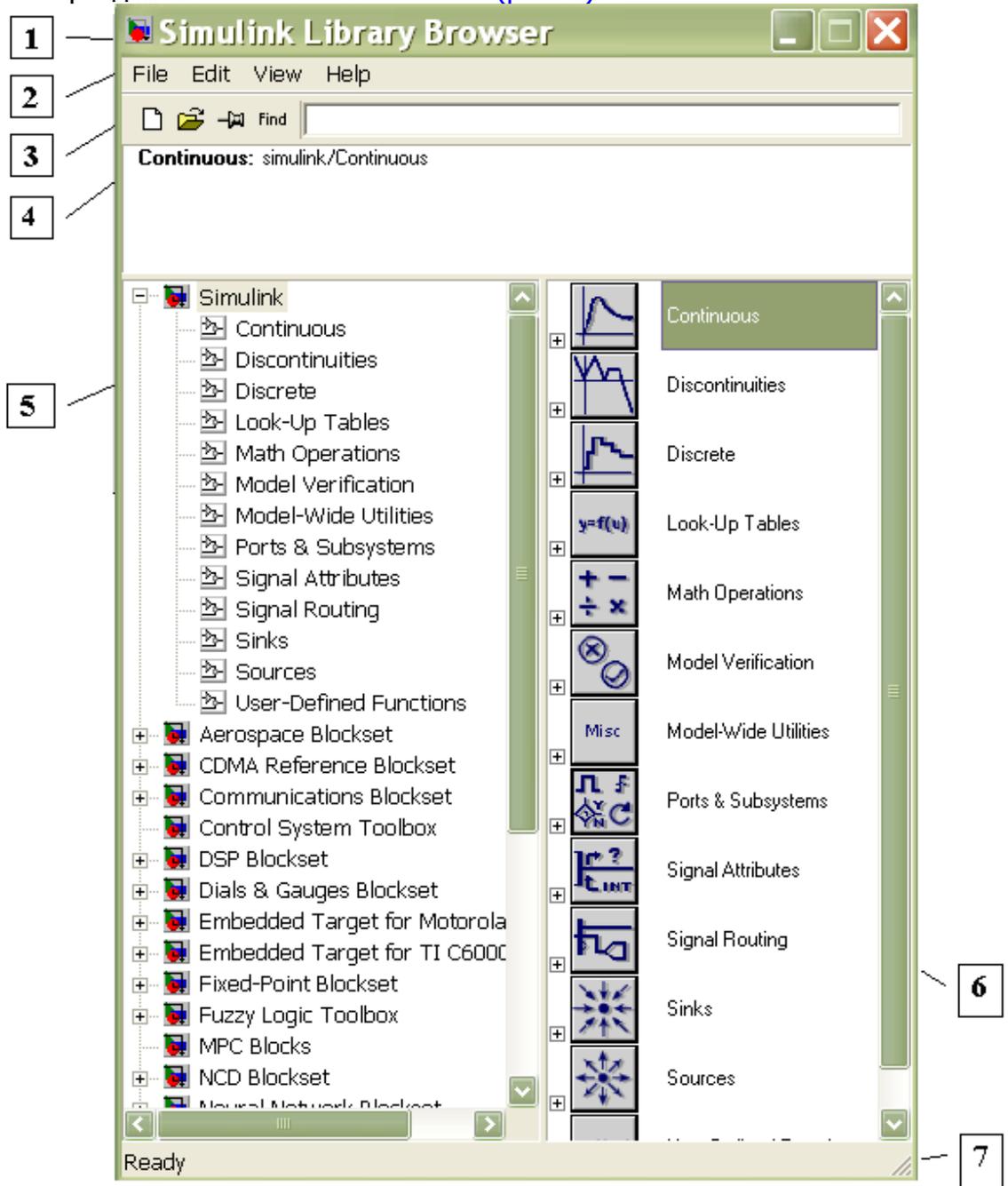


Рис. 1. Окно обозревателя разделов библиотеки **Simulink**

Обозреватель разделов библиотеки Simulink. Окно обозревателя библиотеки блоков содержит следующие элементы:

1. Заголовок, с названием окна — **Simulink Library Browser**.
2. Меню, с командами **File, Edit, View, Help**.
3. Панель инструментов, с ярлыками наиболее часто используемых команд.
4. Окно комментария для вывода поясняющего сообщения о выбранном блоке.
5. Список разделов библиотеки, реализованный в виде дерева.

Информационные технологии в менеджменте

6. Окно содержимого раздела библиотеки (список вложенных разделов библиотеки или блоков)

7. Строка состояния, содержащая подсказку по выполняемому действию.

При выборе соответствующего раздела библиотеки в правой части окна отображается его содержимое (рис. 2).

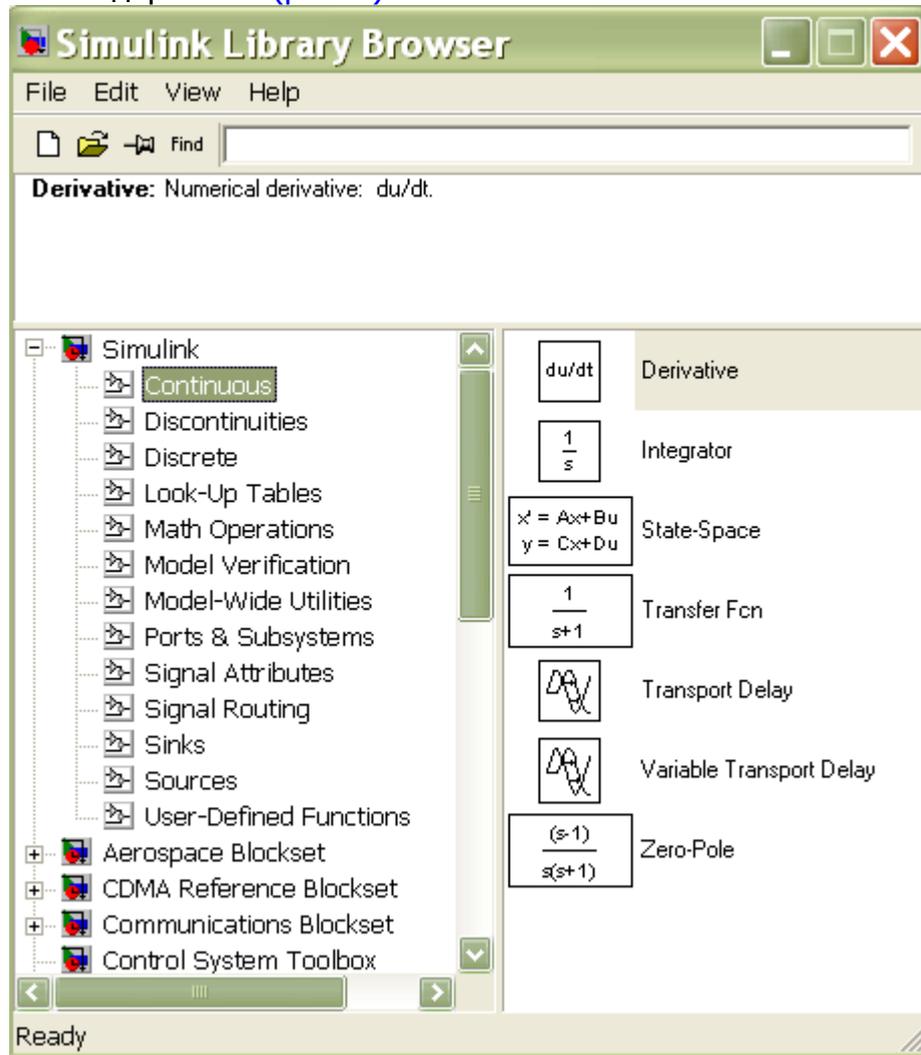


Рис. 2. Окно обозревателя с набором блоков раздела библиотеки

Основные разделы библиотеки:

Continuous — линейные непрерывные компоненты;

Discontinuous — нелинейные компоненты;

Discrete — дискретных компоненты;

Math Operation — математические компоненты;

Ports & Subsystems — компоненты для создания подсистем;

Signal Routing — компоненты для разделения сигналов;

Sinks — регистрирующие компоненты;

Sources — источники сигналов и воздействий;

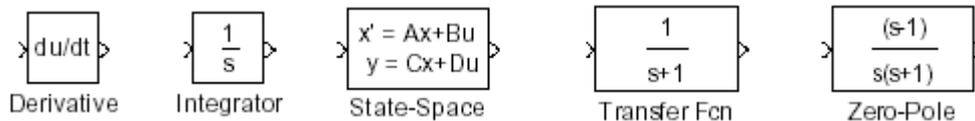
User-Defined Function — компоненты для создания функций пользователя.

Если два раза щелкнуть левой кнопкой мыши на каком-нибудь блоке в окне библиотек, то открывается окно параметров блока, где отображаются установки параметров по умолчанию. Как правило, они нормализованы — например, задана единичная частота, единичная амплитуда, нулевая фаза и так далее. Возможность изменения параметров в этом случае отсутствует. Она появляется после переноса графических элементов в окно подготовки и редактирования функциональных

схем (для его открытия необходимо выбрать иконку  на панели инструментов или **File** → **New** → **Model**).

Подробно рассмотрим наиболее используемые разделы библиотеки.

Линейные непрерывные компоненты (Continuous). Линейные компоненты играют важную роль в создании математических моделей многих устройств. Имеются следующие типы линейных компонентов: **Derivative** — аналоговое дифференцирующее устройство; **Integrator** — аналоговый интегратор; **State-Space** — модель в пространстве состояний; **Transfer Fcn** — передаточные функции, и ряд других устройств.



Название блока	Тип блока	Параметры
Derivative	Производная	нет
Integrator	Интегратор	External reset: <ul style="list-style-type: none"> • none • rising • falling • either • level Initial condition Limit output: <ul style="list-style-type: none"> • upper saturation limit • lower saturation limit
State-space	Описание в пространстве состояний (ОПС)	Initial condition Limit output: • upper saturation limit • lower saturation limit Initial conditions Numerator Denominator Zeros Poles Gain
Transfer Fcn.	Передаточная функция	Initial condition Limit output: • upper saturation limit • lower saturation limit Initial conditions Numerator Denominator Zeros Poles Gain
Zero-pole	Ноль-полюсное описание системы	Initial condition Limit output: • upper saturation limit • lower saturation limit Initial conditions Numerator Denominator Zeros Poles Gain

Нелинейные компоненты (Discontinuous). Simulink предназначен главным образом для моделирования нелинейных динамических систем. Среди нелинейных компонентов следует отметить компоненты с типичными нелинейностями, например, вида **abs(u)**, с характеристиками, описанными типовыми математическими функциями, компонентами вида идеальных и неидеальных ограничителей и так далее. Представлены и такие сложные компоненты, как квантователи, блоки нелинейности, моделирующие нелинейные петли гистерезиса, временные задержки и ключи-переключатели.



Saturation – насыщение, в параметрах задаются верхний и нижний пределы (**Upper limit** и **Lower limit**).



Dead zone – нечувствительность, «мертвая зона». В параметрах задаются пределы нечувствительности (**Start of dead zone** и **End of dead zone**).



Rate Limiter – ограничитель скорости изменения сигнала, в параметрах задаются пределы на скорость увеличения (**Rising slew rate**) и на скорость уменьшения (**Falling slew rate**).

Информационные технологии в менеджменте



Relay – реле, в параметрах задаются точки переключения (**Switch on point** и **Switch off point**), в также величины сигналов в режимах «включено» (**Output when on**) и «выключено» (**Output when off**).

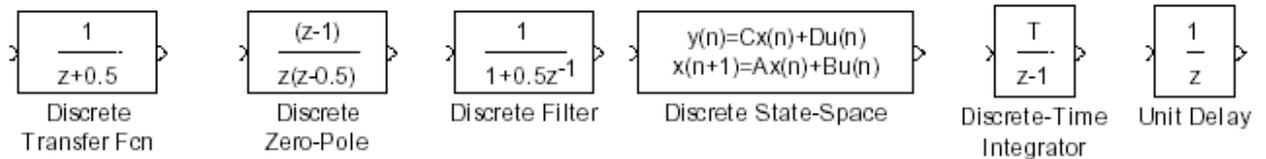


Backlash – люфт, «мертвый ход». В параметрах задаются величина мертвого хода (**Deadband width**) и начальное значение выхода (**Initial output**).



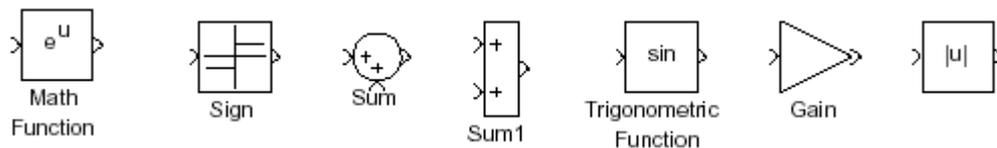
Coulomb and Viscous Friction – кулоновское и вязкое трение.

Дискретные компоненты (Discrete). Дискретные компоненты включают в себя устройства задержки, дискретно-временной интегратор, дискретный фильтр и т. д. Дискретные блоки аналогичны непрерывным блокам:



Среди них есть два блока, не имеющих аналога среди непрерывных блоков. Это блок задержки **Unit delay** и блок **Discrete Filter** — дискретный фильтр, который определяет рациональную функцию от оператора задержки $z / 1$.

Математические компоненты (Math Operation). Окно **Math Operation** позволяет задавать множество математических компонентов с описываемыми пользователями свойствами, имеет решающее значение для выполнения прозрачного для пользователя математического моделирования как простых, так и сложных устройств и систем. Некоторые из них:



Информационные технологии в менеджменте

Название блока	Тип блока	Параметры	
Math function	Математическая функция	Function: exp, log, 10^u , log10, sqrt, Output signal type • Auto • Real • Complex	Вид функции Тип выходного сигнала • Не определен • Вещественный • Комплексный нет
Sign	Знак	Icon shape: • Rectangular • Round List of signs	Изображение блока: • Прямоугольное • Круглое Знаки при входах
Trigonometric function	Тригонометрическая функция	Function: sin, cos, tan, asin, acos, sinh, cosh, Output signal type • Auto • Real • Complex	Вид функции Тип выходного сигнала • Не определен • Вещественный • Комплексный
Gain	Усилитель	Gain Multiplication: • Element-wise ($K \cdot u$) • Matrix ($K \cdot u$) • Matrix ($u \cdot K$)	Коэффициент усиления Способ умножения: • Поэлементное • Матричное ($K \cdot u$) • Матричное ($u \cdot K$)
Abs	Модуль		нет

Компоненты для создания подсистем (Ports & Subsystems). В случае если моделируемая система имеет сложную схему, удобным может быть объединение нескольких элементов в один с помощью блока **Subsystem**. При этом для задания требуемого количества входов и выходов нового элемента используются блоки **In1** и **Out1**.

Компоненты для разделения сигналов (Signal Routing). Иногда для большей наглядности необходимо отобразить несколько графиков в одном окне, для этого может быть использован блок **Mux** из библиотеки **Signal Routing**, который объединяет любое количество входных воздействий в один выходной сигнал, представляющий собой вектор. С помощью блока **Demux** выполняется обратная операция — разъединение одного сигнала (вектора) на несколько.

Manual Switch — ручной переключатель, позволяет двойным щелчком переключать выход на один из двух входных сигналов.

Mux — мультиплексор, объединяет несколько сигналов в один «жгут» (векторный сигнал), в параметрах задается число входов (**Number of Inputs**).

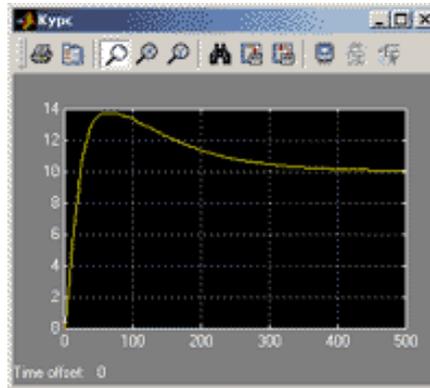
Demux — демultipлексор, позволяет «разбить» векторный сигнал на несколько скалярных, в параметрах задается число выходов (**Number of Outputs**).

Регистрирующие элементы (Sinks). Устройства наблюдения за процессами на выходах звеньев схемы сосредоточены в библиотеке приемников сигналов **Sinks**. Приемники сигналов оформлены в виде блоков, имеющих только входы (один или несколько).

На схеме можно разместить несколько приемников, воспринимающих сигналы на выходах различных звеньев, в том числе и непосредственно от источников сигналов.

Информационные технологии в менеджменте

Наиболее простым устройством для отображения процесса является **Scope**, имитирующий осциллограф. Он имеет один вход, и соответственно одно окно для отображения результата. Если необходимо вывести несколько графиков, то на панели инструментов графика выбирается **Parametrs**→**General** и в поле **Number of axes** выставляется количество осей, которому будет соответствовать количество входов блока **Scope**.



В окне блока **Scope** изображается график изменения входного сигнала. Если вход соединен с выходом мультиплексора, сразу строится несколько графиков (по размерности входного «жгута»).

По умолчанию на оси ординат используется диапазон от -5 до 5. Если этот вариант не подходит, выбрать масштаб автоматически (так, чтобы весь график был виден) можно с помощью кнопки . Соседняя кнопка  сохраняет эти настройки для следующих запусков.

Кнопка  открывает окно настроек, причем наиболее важные данные содержатся на вкладке **Data history**. Если не сбросить флажок **Limit data points**, в памяти будет сохраняться только заданное число точек графика, то есть, при большом времени моделирования начало графика будет потеряно.

Отметив на этой же странице флажок **Save data to workspace** можно сразу передать результаты моделирования в рабочую область MATLAB для того, чтобы их можно было дальше обрабатывать, выводить на графики и сохранять в файле. Поле **Variable name** задает имя переменной в рабочей области, в которой сохраняются данные. В простейшем случае выбирается формат **Array** (в списке **Format**). Это означает, что данные будут сохраняться в массиве из нескольких столбцов (первый столбец – время, второй – первый сигнал, третий – второй сигнал и т.д., по порядку входов мультиплексора).

Блок **XY Scope** применяется при необходимости построения фазовых портретов и статических характеристик.



Display – цифровой дисплей, показывает изменение входного сигнала в цифровом виде.



Scope – осциллограф, показывает изменение сигнала в виде графика.



To File — передача данных в файл **untitled.mat**.



To Workspace — сохранение данных в переменной **simout** и передача в **Workspace**.

Источники сигналов и воздействий (Sources). Окно **Sources** содержит графические элементы — источники воздействий. В электро- и радиотехнике их принято называть источниками сигналов, но в механике и в других областях науки и техники такое название не очень подходит — природа воздействий может быть самой разнообразной, например, в виде перепада давления или температуры, механического перемещения, звуковой волны и так далее.

Большинство элементов содержит рисунок, представляющий временную зависимость воздействий, например, перепад для блока **Step**, синусоиду для блока **Sine Wave** и т. д.

Набор блоков содержит практически все часто используемые при моделировании источники воздействий с самой различной функциональной и временной зависимостью. Возможно задание произвольного воздействия из файла — блок **From File**.



Constant – сигнал постоянной величины.



Step – ступенчатый сигнал, меняется время скачка (**Step Time**), начальное (**Initial Value**) и конечное значение (**Final Value**).



Ramp – линейно возрастающий сигнал с заданным наклоном (**Slope**). Можно задать также время начала изменения сигнала (**Start Time**) и начальное значение (**Initial Value**).



Pulse Generator – генератор прямоугольных импульсов, задаются амплитуда (**Amplitude**), период (**Period**), ширина (**Pulse Width**, в процентах от периода), фаза (**Phase Delay**).



Repeating Sequence – последовательность импульсов, их форма задается в виде пар чисел (время; величина сигнала)



Sine Wave – синусоидальный сигнал, задается амплитуда (**Amplitude**), частота (**Frequency**), фаза (**Phase**) и среднее значение (**Bias**).



Signal Builder – построитель сигналов, позволяющий задавать форму сигнала, перетаскивая мышью опорные точки.



Random Number – случайные числа с нормальным (гауссовым) распределением. Можно задать среднее значение (**Mean Value**), дисперсию (**Variance**), период изменения сигнала (**Sample Time**).

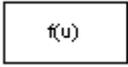
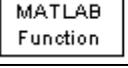
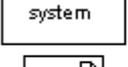
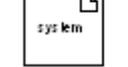


Uniform Random Number – случайные числа с равномерным распределением в заданном интервале от **Minimum** до **Maximum**.



Band Limited White Noise – случайный сигнал, ограниченный по полосе белый шум (имеющий равномерный спектр до некоторой частоты). Блок используется как источник белого шума для моделей непрерывных систем. Задается интенсивность (**Noise Power**) и интервал дискретизации (**Sample Time**), в течение которого удерживается постоянное значение сигнала. Чем меньше интервал, тем точнее моделирование, однако больше вычислительные затраты.

Компоненты для создания функций пользователя (User-Defined Function).

 Fcn	Fcn — задание функции преобразования входного сигнала.
 MATLAB Fcn	Matlab Fcn — вызов файла-функции Matlab (в параметрах блока указывается имя файла).
 S-Function  S-Function Builder	S-function S-Function Builder — создание S-функции.

Построение схемы и проведение имитации. После того как все необходимые для построения модели устройства блоки перенесены на рабочий лист пакета **Simulink**, их соединяют линиям связи, по которым распространяются сигналы. Для этого надо щелкнуть левой кнопкой мыши по источнику сигнала и затем, при нажатой клавише **Ctrl**, по блоку-приемнику. Можно также протянуть мышкой линию связи между нужными выходом и входом.

Для выделения одного блока или соединительной линии надо щелкнуть левой кнопкой мыши по нужному элементу. Для того чтобы выделить несколько блоков, надо «обвести» их при нажатой левой кнопки мыши. Клавиша **Delete** удаляют выделенную часть. Чтобы скопировать блок (или выделенную часть), надо перетащить его при нажатой правой кнопке мыши.

Для изменения оформления выделенного блока предназначено меню **Format** на верхней панели инструментов. Также для этой цели можно использовать контекстное меню **Format** при нажатии на блоке правой кнопкой мыши. Для выделенного блока можно изменить цвет текста и линий (**Foreground color**), цвет фона (**Background color**), вывести тень (**Show drop shadow**), переместить название на другую сторону (**Flip name**).

Для имитационного моделирования следует отредактировать «параметры» имитации и запустить процесс.

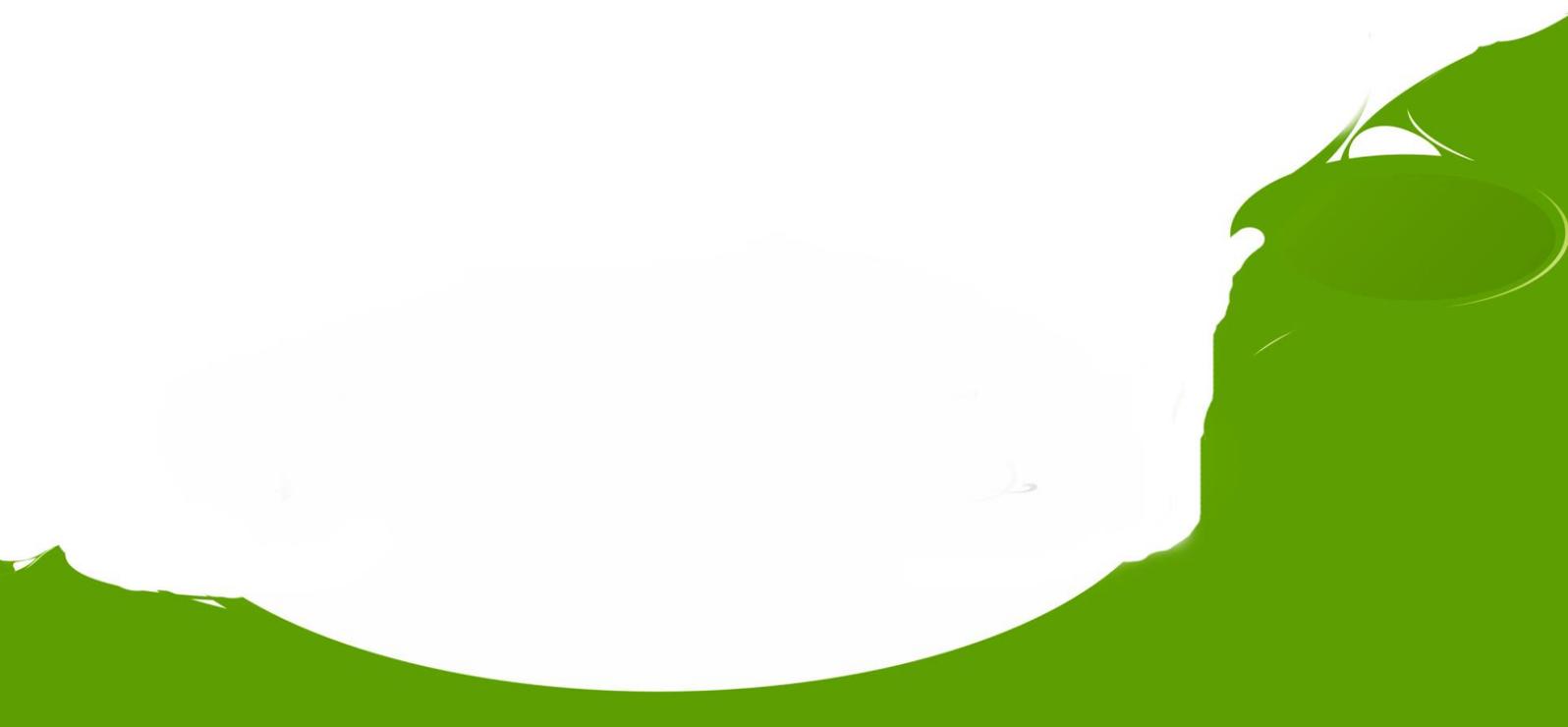
Команда из меню **Simulation** → **Parameters** раскрывает диалоговое поле редактирования параметров имитации, что подразумевает выбор алгоритма численного интегрирования уравнений и задание числовых параметров:

- **Start Time** — время начала процесса (обычно равно 0);
- **Stop Time** — время окончания процесса имитации;
- **Min Step Size** — минимальная величина шага интегрирования;
- **Max Step Size** — максимальная величина шага интегрирования;
- **Tolerance** — допуск (допустимая погрешность).

Величину минимального шага следует назначать с учетом скорости протекания процессов в системе. Максимальный шаг интегрирования можно принять на один или два порядка большим.

Использование переменных для задания параметров блоков. Параметры блоков **Simulink** можно задавать не только как числовые параметры, но и как имена переменных. Значения переменных необходимо задать перед началом расчета в командном окне Matlab или в m-файле. Такой способ задания параметров удобен, если требуется проводить расчеты для множества вариантов параметров блоков модели.

Рекомендации для организации самостоятельной работы по дисциплине



Лабораторная работа

Ознакомление с пакетом Mathcad

Цель работы: ознакомление с пакетом Mathcad, получение навыков работы с основными арифметическими операциями и стандартными функциями, создания функций пользователя.

Теоретические сведения

Структура Mathcad-документа и состав панели инструментов Math (Математика)

При запуске системы Mathcad появляется основное окно, которое содержит три панели (стандартную, форматирования текста, математическую) и рабочую область с автоматически созданным листом вычислений. Включать и отключать панели можно из верхнего меню в разделе **View** → **Toolbars** (**Вид** → **Панели инструментов**).

MathCAD-документ представляет собой совокупность областей типа: Формула, График, Текст. Каждая область имеет форму прямоугольника и может размещаться в любом месте рабочего листа. Для позиционирования областей используется курсор (красного цвета), имеющий крестообразную форму. В текстовой области курсор имеет форму вертикальной черты красного цвета, а в области типа формула и график — форму уголка синего цвета, называемый *следом курсора формул*. Перемещение курсора выполняется с помощью клавиш управления курсором или перемещением указателя мыши с последующим однократным щелчком ее левой кнопки. Область, которая является *активной* или *текущей* (создается, редактируется), выделяется прямоугольной рамкой. Рамка исчезает, как только курсор выводится из области.

Вставка текста и рисунков. Чтобы *вставить текст*, необходимо щелкнуть мышью в той точке листа, где необходим текст, и выполнить команду верхнего меню **Insert** → **Text Region** (**Вставка** → **Текстовый регион**). Появится рамка, в которую можно ввести текст. При этом иногда возникает необходимость сменить шрифт для правильного отображения русских букв — это можно сделать с помощью списка шрифтов на панели форматирования.

Вставку рисунка проще всего осуществить через буфер обмена. Предварительно необходимо поместить изображение в буфер (например, с помощью графического редактора Paint), затем вернуться в Mathcad и выполнить пункт верхнего меню **Edit** → **Paste** (**Правка** → **Вставить**).

Mathcad может одновременно работать с несколькими листами. Создание нового листа осуществляется из верхнего меню — **File** → **New** (**Файл** → **Новый**). Загрузить уже имеющийся лист — **File** → **Open** (**Файл** → **Открыть**). Сохранить созданный лист — **File** → **Save** (**Файл** → **Сохранить**). Переключение между загруженными листами — комбинация клавиш Ctrl+F6 или на панели инструментов **Окно**.

На панели **Math** (**Математика**) находятся кнопки, при нажатии на которые открываются дополнительные панели с шаблонами ввода различных выражений:

Информационные технологии в менеджменте

Кнопка	Панель инструментов
	Калькулятор (Calculator): арифметические операторы.
	График (Graph): двух- и трехмерные типы графиков и графические инструменты.
	Матрица (Matrix): матричные и векторные операторы.
	Выражения (Evaluation): знак равенства при вычислениях и в определении.
	Математический анализ (Calculus): производные, интегралы, пределы, а также итерируемые суммы и произведения.
	Булева алгебра (Boolean): операторы сравнения и логические операторы для булевых выражений.
	Программирование (Programming): программные конструкции.
	Греческие символы (Greek): буквы греческого алфавита.
	Символика (Symbolic): Символьные ключевые слова и модификаторы.

Выражения и стандартные функции Mathcad

Выражения в Mathcad представляют собой совокупность имен переменных, чисел, функций, соединенных знаками арифметических и/или логических операторов. Скобки в записи выражения позволяют изменить принятый по умолчанию порядок выполнения действий, а также служат для указания аргументов функций.

Любая формула может быть записана в любом месте листа. Необходимо щелкнуть мышью в предполагаемой точке ввода формулы, там должен появиться указатель ввода — крест. Во время ввода или редактирования формулы Mathcad обводит ее рамкой (рис. 1).

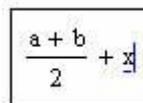


Рис. 1. Формула на листе Mathcad в рамке ввода

Смена уровня ввода формул. Наиболее частая ошибка заключается в том, что в процессе ввода формулы пользователь неосторожно выходит из указанной рамки ввода (при этом рамка исчезает и в рабочей области появляется указатель ввода — крест) и все равно продолжает ввод. При этом на листе вместо одной обычно оказываются две формулы, каждая из которых — лишь часть требуемой. Проблема в том, что визуально эти две формулы могут находиться близко друг к другу и выглядеть как единое целое, но это не так. Единственный способ проверить целостность формулы — щелкнуть по ней мышью. Если все правильно, то в рамке ввода окажется вся формула.

Например, необходимо ввести выражение вида $\frac{a+b}{2}$. После ввода последовательности «**a+b**» формула приобретет вид $\frac{a+b}{}$. Курсор подчеркивает символ «**b**», поэтому если сразу ввести символ деления и затем цифру, то будет введено выражение $a + \frac{b}{2}$. Поэтому после ввода «**a+b**» следует перевести курсор на более низкий уровень, нажав клавишу «пробел». Тогда выражение принимает вид $\frac{a+b}{}$, и если далее ввести символ деления и цифру, то будет получена требуемая формула.

На рис. 2 приведены примеры аналитического и числового вычислений. Левая часть, включая знаки вычисления, вводится пользователем, правая часть — результат, вычисленный системой Mathcad.

The image shows a dashed rectangular box containing three mathematical expressions. The first is an integral: $\int_a^b \exp(x) dx \rightarrow \exp(b) - \exp(a)$. The second is a real number division: $\frac{5 - \sqrt{3}}{2} = 1.634$. The third is a complex number division: $\frac{1 + 2j}{1 - 2j} = -0.6 + 0.8i$.

Рис. 2. Фрагмент листа Mathcad с вычисленными выражениями

При наборе выражения можно последовательно отменять произведенные действия, нажимая Ctrl+Z. Однако следует учесть, что в младших версиях Mathcad (до Mathcad 11) отмена возможна только *до выхода из рамки ввода*. Закончить ввод формулы можно, либо нажав Enter или Tab, либо щелкнув мышью где-нибудь вне рамки ввода формулы.

Если при вводе выражения произошла какая-либо ошибка, то ошибочный фрагмент будет автоматически выделен красным цветом.

Для *удаления формулы* достаточно выделить ее (например, щелкнув мышью) и нажать Ctrl+D или Delete. Для *перемещения формулы* необходимо выделить ее и, ухватившись мышью за появившуюся рамку ввода, переместить. Можно одновременно перемещать блок формул, который необходимо предварительно выделить методом протягивания.

Основные операторы, которые используются для записи формул, перечислены в [таблицах 1.4 и 1.5](#).

Таблица 1.4. Арифметические операторы

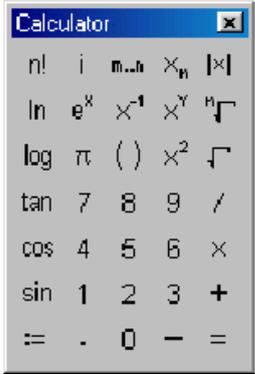
Операция	Запись	Клавиша (комбинация клавиш)	Панель инструментов Math
Сложение, вычитание	$x + y, x - y$	+, -	
Умножение	$x \cdot y$	*	
Деление	$\frac{x}{y}$	/	
Возведение в степень	x^y	^	
Факториал	$x!$!	
Квадратный корень	\sqrt{x}	\	
Корень n-й степени	$\sqrt[n]{x}$	[Ctrl+\]	
Модуль числа	$ x $		
Скобки	(x)	' (апостроф)	

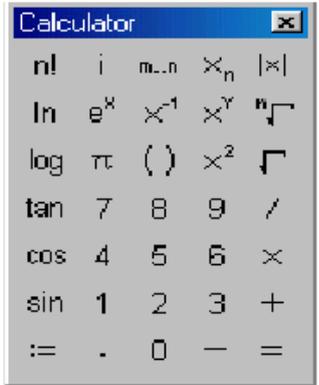
Таблица 1.5. Логические операторы и операторы сравнения

Операция	Запись	Клавиша (комбинация клавиш)	Панель инструментов Math
Равно	$x = y$	Ctrl+ +	
Не равно	$x \neq y$	Ctrl+3	
Логическое отрицание	$\neg x$	Ctrl+Shift+1	
Логическое и	$x \wedge y$	Ctrl+Shift+7	
Логическое или	$x \vee y$	Ctrl+Shift+6	
Меньше	$<$	<	
Больше	$>$	>	
Меньше или равно	$x \leq y$	Ctrl+9	
Больше или равно	$x \geq y$	Ctrl+0	

Оператор, для которого требуется ввести один операнд, называют *унарным* (например, корень квадратный, модуль, факториал). Если же оператор служит для обработки двух операндов, то его называют *бинарным* (например, сложение, вычитание, умножение, деление).

В [таблицах 1.6 и 1.7](#) перечислены некоторые функции Mathcad.

Таблица 1.6. Основные встроенные математические функции

Название функции	Запись в MathCAD	Панель инструментов Math
Тригонометрические: sin, cos, tg, ctg, sec, cosec	sin(x), cos(x), tan(x), cot(x), sec(x), csc(x)	
Обратные тригонометрические: arcsin, arcos, arctg, arcctg, arcsec, arcsec	asin(x), acos(x), atan(x), acot(x), asec(x), acsc(x)	
Экспонента, e^x	$e^x, \exp(x)$	
Натуральный логарифм, ln	ln(x)	
Десятичный логарифм, lg	log(x)	

log(X,b) — логарифм от X по основанию b.

Информационные технологии в менеджменте

sinh(X), cosh(X), tanh(X), coth(X), sech(X), csch(X) — гиперболические синус, косинус, тангенс, котангенс, секанс, косеканс от X , аргументы указываются в радианах.

asin(z), acos(z), atan(z), acot(z), asec(z), acsc(z) — соответственно арксинус, арккосинус, арктангенс, арккотангенс, арксеканс, арккосеканс от z .

Функции для работы с комплексными числами:

Re(Z), Im(Z) — соответственно вещественная и мнимая части комплексного числа Z ;

arg(z) — аргумент комплексного числа z (в радианах).

Элементы статистического анализа данных:

gmean(G1,G2,G3...) — среднее геометрическое аргументов;

mean(G1,G2,G3...) — среднее арифметическое аргументов;

var(G1,G2,G3...) — дисперсия;

stdev(G1,G2,G3...) — среднеквадратичное отклонение.

Таблица 1.7. Прочие функции MathCAD

Назначение функции	Запись в MathCAD	Пример
Наибольшее целое, меньшее или равное x	floor(x)	floor(5.6)=5, floor(-3.56)=-4
Наименьшее целое, большее или равное x	ceil(x)	ceil(5.6)=6, ceil(-3.56)=-3
Угол в радианах между осью OX и радиус-вектором (x, y), значение угла лежит в промежутке $[0,2\pi]$	angle(x,y)	angle(1,1)=0.785
Остаток от целочисленного деления m на n	mod(m,n)	mod(45,7)=3

Ввод функции может быть выполнен с помощью алфавитно-цифровой клавиатуры, или (наиболее распространенных) — по нажатию соответствующей кнопки панели инструментов **Calculator**, или с помощью мастера функций (выводится полный список функций, объединенных по категориям). Последний режим ввода активизируется по команде вставки функции в документ **Insert** → **Function** (Вставка → Функция), когда выбор требуемой функции осуществляется в диалоговом окне (рис. 1.5).

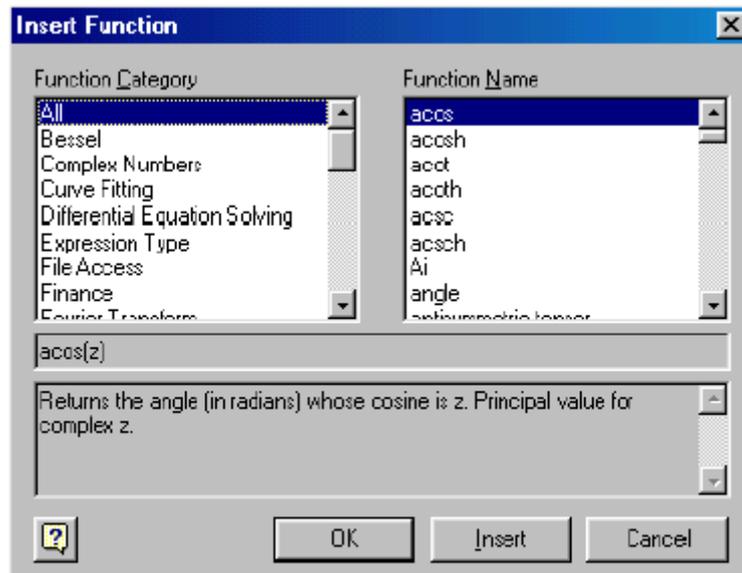


Рис. 1.5. Диалоговое окно вставки функции

Определение переменных (присвоение значений переменным)

Как и в языках программирования, в Mathcad различают:

- локальные переменные,
- глобальные переменные.

В обоих случаях определение переменной означает присваивание ей значения. В качестве оператора присваивания используется обозначение $:=$ для локального и \equiv для глобального присваивания.

Для ввода операторов используют символы на панели инструментов **Evaluation**. Слева от оператора должна находиться определяемая переменная. Справа — выражение, имеющее определенное значение, которое присваивается переменной. Например, с помощью следующего оператора $x := 5.6$ локальной переменной x присваивается значение 5,6. Глобальная переменная определяется аналогично. Например, $x \equiv 3.56$.

Если ввести оператор без задания имени справа, то на экран в зависимости от вида оператора присваивания будет выведена одна из приведенных ниже конструкций:

$$\blacksquare := \blacksquare \qquad \blacksquare \equiv \blacksquare$$

Теперь останется только заполнить знакоместа слева и справа оператора, указанные черными квадратами.

Различие между локальными и глобальными переменными заключается в процессе, посредством которого Mathcad производит вычисления в документе. Mathcad читает документ **слева направо и сверху вниз** несколько раз. Во время «первого чтения» вычисляются значения глобальных переменных, в последующем чтении, когда происходит вычисление выражения, все переменные анализируются и интерпретируются. Таким образом, глобальные переменные могут использоваться в любом месте документа, в том числе слева от их определения или над ним. Для локальных переменных должно соблюдаться правило логической последовательности — определение переменной слева на одной высоте или выше вычисляемого выражения, в котором используется данная переменная.

Информационные технологии в менеджменте

Вычисление результата. После ввода выражения и знака «=» справа будет показан его результат. Таким же образом можно выводить на экран значения переменных, вектора, матрицы, таблицы значений функций.

Значения переменных можно изменить в любой момент, отредактировав соответствующую формулу. При этом обычно автоматически пересчитываются все формулы, которые прямо или опосредованно зависят от данной переменной. Если же этого не происходит, то необходимо включить автовычисление с помощью верхнего меню Mathcad (пункт **Math** → **Automatic calculation** или **Математика** → **Автовычисление**).

Функции, определяемые пользователем

Важнейшим объектом в математике является *функция*. Функция характеризуется своим именем и аргументами (список параметров) и однозначно ставит в соответствие значениям аргументов значение функции. Имя функции — это идентификатор, удовлетворяющий требованиям MathCAD (смотри правила создания имени в разделе «Переменные»). Список параметров представляет список переменных, разделенных запятыми, и заключенный в круглые скобки. Формат функции:

Имя_функции(переменная_1, переменная_2, ...)

Определение функции (локальное или глобальное) производится так же, как и определение переменной:

Имя_функции(переменная_1, переменная_2, ...):=определяющее_выражение

При глобальном определении функции следует использовать символ \equiv .

Поскольку аргументы функции представляют собой *формальные переменные*, значения которым присваиваются непосредственно при вычислении функции, то их не нужно определять заранее.

Примеры.

1. Функция расстояния между точками (x_1, y_1) и (x_2, y_2) :

$$r(x_1, y_1, x_2, y_2) := \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

2. Функция общего сопротивления участка электрической цепи с параллельно соединенными сопротивлениями:

$$R_p(R_1, R_2) := \frac{R_1 \cdot R_2}{R_1 + R_2}$$

3. Функция округления числа x до заданного количества N знаков после запятой:

$$\text{Round}(x, N) := \text{floor}(x \cdot 10^N + 0.5) \cdot 10^{-N}$$

4. Функция для вычисления тангенса (с именем, которое вам известно еще из школьного курса математики):

$$\text{tg}(x) := \tan(x)$$

Если выражение, определяющее функцию, содержит переменные, которые не входят в список ее аргументов, то они должны быть предварительно заданы. Например,

Информационные технологии в менеджменте

$$a := 4.5$$

$$f(x) := a \cdot x^2 + \cos(x)$$

К определенной пользователем функции можно выполнить обращение, указав ее *фактические аргументы*, которыми могут быть соответствующие им по типу константы, заданные до обращения к функции переменные, произвольные выражения. Например,

$$r(0, 2, 4, -5) = 8.062$$

$$RP(2, 5) = 1.429$$

$$\text{Round}(r(0, 0, -3, 7), 2) = 7.62$$

$$a := 2 \quad b := 4 \quad R := r(0, 0, a, b) \quad R = 4.472$$

Аргументом функции может быть и имя функции (встроенной или определенной пользователем), например:

$$g(x, f) := 2 \cdot x + f(x)$$

$$s := g(3, \sin) \quad s = 6.141$$

Запись логических выражений. Функция if

Часто при проведении вычислений требуется установить выполнение некоторого условия или условий для заданных величин. К условиям, например, относятся простые или двойные неравенства, их совокупности или системы.

В Mathcad выделяют **простые** и **составные условия** (логические выражения). Простое условие имеет следующую структуру:

$$\text{Выражение}_1 \text{ Оператор}_\text{сравнения} \text{ Выражение}_2$$

Например,

$$x \geq 5 \quad (x + y) > x^2 \quad \sin(x) \neq a \quad (a + b) = 12$$

Операторы сравнения и варианты их ввода приведены в [таблице 1.5](#).

Составные условия содержат простые условия, соединенные логическими операторами ([см. таблица 1.5](#)). Например, двойное неравенство $-4 < x \leq 56$ можно записать следующим образом:

$$(x > -4) \wedge (x \leq 56)$$

а совокупность неравенств $-4 < x \leq 7, 12 < x \leq 34$:

$$(x > -4) \wedge (x \leq 7) \vee (x \leq 34) \wedge (x > 12)$$

При записи условий следует помнить, что существует определенный порядок выполнения операций — слева направо в соответствии с приоритетом операций:

- Сначала вычисляются значения числовых выражений,
- Затем устанавливаются результаты простых условий (выполняются операции сравнения),
- Далее в случае составного условия выполняются логические операции в последовательности: отрицание, умножение, сложение. Изменить такой, принятый по умолчанию порядок выполнения логических операций, можно, используя в

записи логического выражения скобки. При этом выражение, содержащееся в самых внутренних скобках, вычисляется первым.

Например,

$$x := 2 \quad y := 2$$

$$(x > 1) \vee (y < 1) \wedge (x^2 + y^2 \leq 4) = 1$$

$$[(x > 1) \vee (y < 1)] \wedge (x^2 + y^2 \leq 4) = 0$$

Здесь в первом выражении сначала выполняется операция умножения \wedge , а затем — сложения \vee , во втором — наоборот, сначала — сложения \vee , а затем — умножения \wedge .

Поскольку результатом оценивания условия является значение 1, соответствующее значению ИСТИНА, или 0, соответствующее значению ЛОЖЬ, и, более того, любое отличное от нуля числовое значение соответствует в Mathcad логическому значению ИСТИНА, то вместо логических операторов умножения \wedge и сложения \vee могут быть использованы арифметические операторы умножения и сложения соответственно. Например,

$$x := 4 \quad y := -1$$

$$x > 7 = 0$$

$$x < 10 = 1$$

$$(x > 7) \wedge (y < 0) = 0$$

$$(x > 7) \cdot (y < 0) = 0$$

$$(x > 7) \vee (y < 0) = 1$$

$$(x > 7) + (y < 0) = 1$$

А для ранее записанных с помощью логических операторов составных условий можно предложить и такой вариант записи:

Пример 2.1. Составьте логическое выражение, которое принимает значение ИСТИНА только в случае, когда точка с заданными координатами (x, y) принадлежит заштрихованной области, показанной на рис. 2.1.

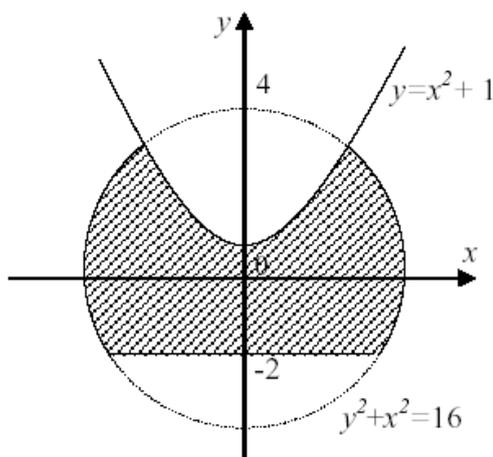


Рис. 2.1

Решение: Заштрихованная область – это решение системы неравенств:

$$\begin{cases} x^2 + y^2 \leq 16, \\ y \geq -2, \\ y \leq x^2 + 1, \end{cases}$$

которую, используя операции логического умножения, можно записать в виде следующего логического выражения:

$$(x^2 + y^2 \leq 16) \wedge (y \geq -2) \wedge (y \leq x^2 + 1)$$

Информационные технологии в менеджменте

Выясним, какое значение принимает логическое выражение для заданных x и y , подготовив MathCAD-документ следующего содержания:

$$\begin{aligned} x &:= 0 & y &:= 0 \\ (x^2 + y^2 \leq 16) \wedge (y \geq -2) \wedge (y \leq x^2 + 1) &= 1 \\ x &:= -2 & y &:= -3.5 \\ (x^2 + y^2 \leq 16) \wedge (y \geq -2) \wedge (y \leq x^2 + 1) &= 0 \end{aligned}$$

Логические выражения используются для задания аргумента-условия встроеной функции **if**. Функция имеет следующий формат:

$$\mathbf{if}(\text{условие}, \text{выражение}_1, \text{выражение}_2)$$

Функция **if** принимает значение выражения_1, если условие является истинным, иначе значения выражения_2.

Пример 2.2. Требуется для заданного значения переменной x найти значение функции

$$f(x) = \begin{cases} \cos(x), & x < 0, \\ x+1, & 0 \leq x < 10, \\ \operatorname{tg}(x), & x \geq 10 \end{cases}$$

Решение: MathCAD-документ может быть составлен следующим образом:

$$\begin{aligned} a &:= 7 \\ f(x) &:= \mathbf{if}(x < 0, \cos(x), \mathbf{if}(x < 10, x + 1, \tan(x))) \\ f(a) &= 8 \end{aligned}$$

Пример 2.3. Задавая значения двух сопротивлений R_1 и R_2 и тип их соединения \emptyset – последовательное, 1 – параллельное), найдите общее сопротивление участка электрической цепи.

Решение: Возможный вариант оформления MathCAD-документа:

$$\begin{aligned} R_1 &:= 2 & R_2 &:= 4 & \text{type} &:= 1 \\ R &:= \mathbf{if}\left(\text{type} = 0, R_1 + R_2, \frac{R_1 \cdot R_2}{R_1 + R_2}\right) \\ R &= 1.333 \end{aligned}$$

Переменные диапазона (ранжированные переменные)

Переменная диапазона (или **ранжированная переменная**) — это переменная, которая принимает конечное множество значений из заданного диапазона при каждом обращении к ней.

С помощью переменной диапазона можно многократно вычислять одно и то же выражение для различных значений переменной. Результаты вычислений могут быть выведены в виде таблицы или графика.

Переменная диапазона задается с помощью оператора присваивания следующим образом:

$$\text{Имя_переменной} := \text{Выражение}_1 \dots \text{Выражение}_K$$

или

$$\text{Имя_переменной} := \text{Выражение}_1, \text{Выражение}_2 \dots \text{Выражение}_K$$

Информационные технологии в менеджменте

Здесь символ .. (две точки) указывает на задание диапазона и вводится нажатием клавиши [;] (точка с запятой) или нажатием кнопки **m..n** панели инструментов **Math, Calculator** (см. таблица 1.4), а остальные элементы имеют следующий смысл:

- Выражение_1 – первое значение переменной,
- Выражение_2 – второе значение переменной,
- Выражение_K – последнее значение переменной.

Разность **Выражение_2-Выражение_1** задает шаг изменения переменной. Если **Выражение_2** не указано (первая форма задания переменной), то шаг по умолчанию считается равным 1. Таким образом, множество значений переменной диапазона образуют арифметическую прогрессию, все члены которой принадлежат промежутку:

$$[\text{Значение Выражения}_1; \text{Значение Выражения}_K]$$

Все параметры в выражениях должны быть определены до задания переменной диапазона. Например,

$$x0 := -12.4 \quad hx := 0.1 \quad xk := 14.3$$

$$x := x0, x0 + hx .. xk$$

Если значение **Выражения_K** окажется меньше значения **Выражения_1**, а шаг изменения значения переменной диапазона отрицательным, например, $x := 6, 4 .. 18$, то будет выдано следующее сообщение об ошибке:

Can't understand the way this range variable is defined

Пример 2.4. Вычислите значения функции $y = 4x^3 - 2x + 5$ для значений x , изменяющихся от -1.15 до 1 с шагом $0,3$.

Решение: MathCAD-документ может быть оформлен следующим образом:

$$f(x) := 4x^3 - 2x + 5$$

$$x0 := -1.15 \quad xk := 1 \quad h := 0.3$$

$$x := x0, x0 + h .. xk \quad x = \quad f(x) =$$

-1.15	1.217
-0.85	4.244
-0.55	5.434
-0.25	5.438
0.05	4.9
0.35	4.471
0.65	4.799
0.95	6.53

Обратите внимание на то, что последнее значение, которое принимает переменная, может не совпадать с последним, указанным при задании диапазона. Это имеет место тогда, когда последний член арифметической прогрессии не совпадает с границей диапазона.

Если выражение содержит более двух переменных диапазона, то при выводе его значений в виде таблицы сначала все свои значения примет первая в записи выражения. Проиллюстрируем это следующим примером.

Информационные технологии в менеджменте

Пример 2.5. Вычислите значения функции $f(x,y)=x^2 + y$ для значений x , изменяющихся от 1 до 1,5 с шагом 0,2, и y , изменяющихся на промежутке от -1 до 0 с шагом 0,3.

Решение:

$x := 1, 1.2, 1.5$	$y := -1, -0.7, 0$	$f(x,y) := x^2 + y$																											
$x =$	$y =$	$f(x,y) =$	$f(x,-1) =$	$f(x,-0.7) =$																									
<table border="1" style="display: inline-table; border-collapse: collapse; text-align: center;"><tr><td>1</td></tr><tr><td>1.2</td></tr><tr><td>1.4</td></tr></table>	1	1.2	1.4	<table border="1" style="display: inline-table; border-collapse: collapse; text-align: center;"><tr><td>-1</td></tr><tr><td>-0.7</td></tr><tr><td>-0.4</td></tr><tr><td>-0.1</td></tr></table>	-1	-0.7	-0.4	-0.1	<table border="1" style="display: inline-table; border-collapse: collapse; text-align: center;"><tr><td>0</td></tr><tr><td>0.44</td></tr><tr><td>0.96</td></tr><tr><td>0.3</td></tr><tr><td>0.74</td></tr><tr><td>1.26</td></tr><tr><td>0.6</td></tr><tr><td>1.04</td></tr><tr><td>1.56</td></tr><tr><td>0.9</td></tr><tr><td>1.34</td></tr><tr><td>1.86</td></tr></table>	0	0.44	0.96	0.3	0.74	1.26	0.6	1.04	1.56	0.9	1.34	1.86	<table border="1" style="display: inline-table; border-collapse: collapse; text-align: center;"><tr><td>0</td></tr><tr><td>0.44</td></tr><tr><td>0.96</td></tr></table>	0	0.44	0.96	<table border="1" style="display: inline-table; border-collapse: collapse; text-align: center;"><tr><td>0.3</td></tr><tr><td>0.74</td></tr><tr><td>1.26</td></tr></table>	0.3	0.74	1.26
1																													
1.2																													
1.4																													
-1																													
-0.7																													
-0.4																													
-0.1																													
0																													
0.44																													
0.96																													
0.3																													
0.74																													
1.26																													
0.6																													
1.04																													
1.56																													
0.9																													
1.34																													
1.86																													
0																													
0.44																													
0.96																													
0.3																													
0.74																													
1.26																													
			$f(x,-0.4) =$	$f(x,-0.1) =$																									
			<table border="1" style="display: inline-table; border-collapse: collapse; text-align: center;"><tr><td>0.6</td></tr><tr><td>1.04</td></tr><tr><td>1.56</td></tr></table>	0.6	1.04	1.56	<table border="1" style="display: inline-table; border-collapse: collapse; text-align: center;"><tr><td>0.9</td></tr><tr><td>1.34</td></tr><tr><td>1.86</td></tr></table>	0.9	1.34	1.86																			
0.6																													
1.04																													
1.56																													
0.9																													
1.34																													
1.86																													

Сопоставьте таблицы вывода значений функции $f(x,y)$ с фиксированными значениями второго аргумента (в последовательности -1, -0.7, -0.4, -0.1) с таблицей вывода значений функции $f(x,y)$.

Задания

Задание 1. Определите переменные с именами x, y, a, b, p, q , выбрав их значения произвольно. Определите переменную d , значением которой является выражение:

1., 20.
$$\left(\frac{x^2 - y^2}{x - y} - \frac{x^2 + y^2}{x + y} \right) \cdot \frac{x^2 - y^2}{2xy}$$

2., 21.
$$\left(\frac{x^3 - 8}{x^2 + 4 + 2x} + 4 \right) \cdot \frac{1}{x + 2}$$

3., 22.
$$\frac{4x(x + \sqrt{x^2 - 1})^2}{(x + \sqrt{x^2 - 1})^4 - 1}$$

4., 23.
$$\left(\frac{2}{2x + y} - \frac{1}{2x - y} - \frac{3y}{y^2 - 4x^2} \right) \cdot \left(\frac{y^2}{8x^2} - 0,5 \right)$$

5., 24.
$$\left(\frac{a}{a + \sqrt{ab}} - \frac{b}{\sqrt{ab} + b} \right) \left(\frac{\sqrt{a}}{\sqrt{b}} - \frac{\sqrt{b}}{\sqrt{a}} \right) \sqrt{ab}$$

6., 25.
$$\frac{(a - b)^3 (\sqrt{a} + \sqrt{b})^{-3} + 2a\sqrt{a} + b\sqrt{b}}{a\sqrt{a} + b\sqrt{b}} + \frac{3(\sqrt{ab} - b)}{a - b}$$

$$7., 26. \left(p - q + \frac{4q^2 - p^2}{p + q} \right) \left(\frac{p}{p^2 - q^2} + \frac{2}{q - p} + \frac{1}{p + q} \right)$$

$$8., 27. \frac{0,25(a^{-3/2} - a^{3/2})}{a^{-1} + a + 1} \left(\frac{\sqrt{a} - 1}{\sqrt{a + 1}} - \frac{\sqrt{a} + 1}{\sqrt{a} - 1} \right)$$

$$9., 28. (\sqrt{1 - x^2} + 1) \left(\frac{1}{\sqrt{1 + x}} + \sqrt{1 - x} \right)$$

$$10., 29. \left(\frac{a\sqrt{a} + b\sqrt{b}}{\sqrt{a} + \sqrt{b}} - \sqrt{ab} \right) : (a - b)$$

$$11., 30. \left(\frac{a\sqrt{a} + b\sqrt{b}}{\sqrt{a} + \sqrt{b}} - \sqrt{ab} \right) \left(\frac{\sqrt{a} + \sqrt{b}}{a - b} \right)^2$$

$$12. \left(\frac{a}{a - b} - \frac{b}{a + b} \right) \left(\frac{a + b}{b} - \frac{a - b}{a} \right)$$

$$13. \left(\frac{\sqrt{a} + 2}{(\sqrt{a} + 1)^2} - \frac{\sqrt{a} - 2}{a - 1} \right) \frac{\sqrt{a} + 1}{\sqrt{a}}$$

$$14. \left(\frac{a}{a - 1} + \frac{1}{ab - b} \right) \left(\frac{ab + 1}{b} \right)^{-1} - \frac{a}{a^2 - 1}$$

$$15. \frac{2b\sqrt{x^2 - 1}}{x - \sqrt{x^2 - 1}} = A, \text{ где } x = \frac{1}{2} \left(\sqrt{\frac{a}{b}} + \sqrt{\frac{b}{a}} \right)$$

$$16. \left(\frac{\sqrt[4]{ab^3} - \sqrt[4]{a^3b} + \sqrt{ab + 1}}{\sqrt{a} - \sqrt{b}} + \frac{\sqrt{ab + 1}}{\sqrt[4]{ab}} \right)^{-2} \cdot \sqrt{1 + \frac{a}{b} + 2\sqrt{\frac{a}{b}}}$$

$$17. \left(\left(\frac{\sqrt{a} + \sqrt{b}}{\sqrt[4]{a} - \sqrt[4]{b}} \right)^{-1} - \frac{2\sqrt[4]{ab}}{b^{\frac{3}{4}} - a^{\frac{1}{4}}b^{\frac{1}{2}} + a^{\frac{1}{2}}b^{\frac{3}{4}} - a^{\frac{3}{4}}} \right)^{-1}$$

$$18. \left(\frac{\sqrt{a^3} - \sqrt{b^3}}{\sqrt{a} - \sqrt{b}} + \sqrt{ab} \right) \left(\frac{\sqrt{a} - \sqrt{b}}{a - b} \right)^2$$

$$19. \left(2x + 1 - \frac{1}{1 - 2x} \right) \left(2x - \frac{4x^2}{2x - 1} \right)$$

Выведите результат.

Преобразуйте переменную d в функцию $d(x, y, a, b, p, q)$. Проверьте, возможно, некоторые аргументы являются излишними. Исправьте определение функции.

Выведите результат при разных значениях аргументов.

Информационные технологии в менеджменте

Задание 2. Определите функцию $h(x, y)$ выражением:

№	Выражение	№	Выражение
1.	$h = (xy^2 + x) + (x^2y - y)$	16.	$h = x^2y + 2xy + \ln x$
2.	$h = y \sin^2 x + x \ln y$	17.	$h = xy + y - 2y^2 \ln x$
3.	$h = (x + 2y) - \lg(xy)$	18.	$h = -y/\sqrt{x^2 - y^2}$
4.	$h = x^2y - (2xy - y^2)x$	19.	$h = x(y^2 - 3) + (x^2y - 2\sqrt{y})$
5.	$h = (1 + x^2)y - 2xy + (1 + x^2)^2$	20.	$h = y \cos x + (y + 1) \sin x$
6.	$h = (xy^5 - 1) \ln x + 2y$	21.	$h = y \operatorname{ctg} x + y + 2$
7.	$h = xy - 3y - x^4y^2$	22.	$h = \cos^2 x \cdot \operatorname{ctg} y + \sin^2 y \cdot \operatorname{tg} x$
8.	$h = 4x^3e^y + y^4e^x + 4y^3e^x$	23.	$h = xy + y(\ln y - \ln x)$
9.	$h = \cos y \cdot \sin x + \sin y \cdot \cos x$	24.	$h = (x^2 - 2y^2)x + 2xy$
10.	$h = (x^2 - y^2)y + 2xy$	25.	$h = xy^{2-y} + y - x - 1$
11.	$h = y \cos x - \frac{y}{\ln y}$	26.	$h = y - y \operatorname{ctg} x + \frac{1}{\sin x}$
12.	$h = 3e^x \operatorname{tg} y + (1 - e^x) \cos^{-2} y$	27.	$h = x^2y - 2xy + y^2$
13.	$h = xy + y - x e^{\frac{y}{x}}$	28.	$h = 2x \cos^2 y + 2y - x^2$
14.	$h = y^2 + x^2y + xy$	29.	$h = \left(1 + e^{\frac{x}{y}}\right) x + e^{\frac{x}{y}} (1 - x/y) y$
15.	$h = x^2y + 2xy + 3$	30.	$h = xy + y - y^2$

Выведите результат при разных значениях аргументов.

Задание 3. Определите функцию $g(x_1, x_2, \dots)$, зависящую от аргументов функций $d(x, y, a, b, p, q)$ и $h(x, y)$, и содержащую следующие математические операции над функциями $d(x, y, a, b, p, q)$ и $h(x, y)$: возведение в степень, извлечение корня, тригонометрические функции, функцию логарифма от функций. Выведите результат при разных значениях аргументов.

Задание 4. Используя, логические операторы, операторы сравнения и функцию **if**, определите функцию $y(x)$:

$$1) \quad y(x) = \begin{cases} ab-x^3, & \text{если } a < x < b, \\ 4(x^2+a), & \text{если } x \geq b, \\ x^2(a+b), & \text{если } x \leq a; \end{cases}$$

$$a = -0.0000124; \quad b = 2/5;$$

$$2) \quad y(x) = \begin{cases} 7(b/x - a/b)^2, & \text{если } x \leq a; \\ 2(a-b)^3(ax+bc), & \text{если } a < x < b, \\ 3abx, & \text{если } x \geq b \end{cases}$$

$$a = 2; \quad b = 3.5; \quad c = 4.1;$$

Информационные технологии в менеджменте

- 3)
$$y(x) = \begin{cases} \cos(ax^2+b)^2, & \text{если } -19 \leq x < -16, \\ 5(bx+|a+d|), & \text{если } x \geq -16, \\ 2xd+a^2, & \text{если } x < -19; \end{cases}$$

 $a=-9/2; b=5.61; d=24;$
- 4)
$$y(x) = \begin{cases} a^2b/9-|x|, & \text{если } x \leq -2, \\ 4(a^3x^2-c^2), & \text{если } -2 < x < 4, \\ \operatorname{tg}[(a-b)^2+x^2], & \text{если } x \geq 4; \end{cases}$$

 $a=-0.0025; b=-2.7; c=3;$
- 5)
$$y(x) = \begin{cases} \sqrt{a^2|x|+|bc|}, & \text{если } -1 \leq x < a, \\ 10(|ax|+bx), & \text{если } x > a, \\ ax^2, & \text{если } x < -1; \end{cases}$$

 $a=21.345; b=-2/5; c=1;$
- 6)
$$y(x) = \begin{cases} x-a, & \text{если } x < -1, \\ \sqrt{x-a}, & \text{если } -1 \leq x \leq 1, \\ 9x^2-a, & \text{если } x > 1; \end{cases}$$

 $a=-1.5;$
- 7)
$$y(x) = \begin{cases} \sqrt{(b+x^2)}, & \text{если } -2 < x \leq 3, \\ ab/x-x^2, & \text{если } x > 3, \\ 4(x+a/b), & \text{если } x \leq -2; \end{cases}$$

 $a=-0.024; b=0.752479;$
- 8)
$$y(x) = \begin{cases} 5|b-a|/(|b|-|c|)x^2, & \text{если } 3 \leq x \leq 5, \\ 2.5(|x|-c)(ax+b), & \text{если } x < 3, \\ 10^2dbc/x, & \text{если } x > 5; \end{cases}$$

 $a=7.4; b=2.5; c=3.1; d=0.5;$
- 9)
$$y(x) = \begin{cases} 5c^2d^2b, & \text{если } x=3.5, \\ 4(cd^2+x^2), & \text{если } 1.5 \leq x < 3.5, \\ (ax+b)^2, & \text{если } x > 3.5; \end{cases}$$

 $a=4.7; b=2.1; c=1.7; d=0.5;$
- 10)
$$y(x) = \begin{cases} 4(x/a^2-a), & \text{если } x < -1, \\ \sqrt{x+a}, & \text{если } -1 \leq x \leq 1, \\ x^3, & \text{если } x > 1; \end{cases}$$

 $a=12.1;$
- 11)
$$y(x) = \begin{cases} x^2+\cos(a+bx), & \text{если } x < 0, \\ 4(x-1), & \text{если } x > 1, \\ (x+2)+0.0024, & \text{если } 0 \leq x \leq 1; \end{cases}$$

 $a=-0.24; b=2;$
- 12)
$$y(x) = \begin{cases} (a^2-0.008b^2)/x, & \text{если } a < x \leq b, \\ bx-a, & \text{если } x \leq a, \\ (a+b)\cos x, & \text{если } x > b; \end{cases}$$

 $a=-0.2; b=2;$
- 13)
$$y(x) = \begin{cases} 0.003a/x, & \text{если } x \leq c, \\ 2x/a, & \text{если } x \geq a+b, \\ c(ax+b)^2, & \text{если } c < x < a+b; \end{cases}$$

 $a=0.8; b=-0.26; c=-0.24;$
- 14)
$$y(x) = \begin{cases} |a|b-x^2, & \text{если } x \leq 0, \\ a^2x+b^2, & \text{если } 0 < x \leq 0.1, \\ b(x^2-a^2), & \text{если } x > 0.1; \end{cases}$$

 $a=2/7; b=-3.01;$

Информационные технологии в менеджменте

$$15) \quad y(x) = \begin{cases} a^3+dx^2, & \text{если } 0.1 \leq x \leq a, \\ (b-x)^2, & \text{если } x > a, \\ (b^2+x^2)a-x, & \text{если } x < 0.1; \end{cases}$$

$a=2.51; b=-3.06; d=5/3;$

$$16) \quad y(x) = \begin{cases} x^2+a, & \text{если } x < a, \\ 8.4(x+a), & \text{если } a \leq x \leq d, \\ dx/15-a^x, & \text{если } x > d; \end{cases}$$

$a=0.2; d=1.9;$

$$17) \quad y(x) = \begin{cases} e^{ax}, & \text{если } x < -1, \\ \cos x + 0.0085, & \text{если } -1 \leq x < 0, \\ 4x^2-a, & \text{если } x \geq 0; \end{cases}$$

$a=29.4;$

$$18) \quad y(x) = \begin{cases} (ax^2+2)/(x^2+1), & \text{если } 1 < |x| < 3, \\ a, & \text{если } |x| \geq 3, \\ ax/(x+2), & \text{если } |x| \leq 1; \end{cases}$$

$a=9/4;$

$$19) \quad y(x) = \begin{cases} (|a|-|b|)/(ab+x)/c, & \text{если } 3 < x \leq 5, \\ (a-b)^2x^3, & \text{если } 5 < x < 9.8, \\ (ax^2+b)/4x, & \text{если } x \geq 9.8; \end{cases}$$

$a=-19.2457; b=24.7; c=1.5;$

$$20) \quad y(x) = \begin{cases} 4[ax-6(a+b)^3], & \text{если } a+b \leq x < a, \\ a^2+(a-b)+x^2, & \text{если } x \geq a, \\ x^2(a+b), & \text{если } x < a+b; \end{cases}$$

$a=2.005; b=-0.0009;$

$$21) \quad y(x) = \begin{cases} a^2(a-b)x, & \text{если } |a| < x \leq |b|, \\ 5(ax^2+b), & \text{если } x > |b|, \\ (ax-|a|)^2, & \text{если } x \leq |a|; \end{cases}$$

$b=-1.5; a=0.6;$

$$22) \quad y(x) = \begin{cases} (ab-cx)/abcx^2, & \text{если } x < 0, \\ \sqrt{ax^2 + |bx|}, & \text{если } 0 \leq x \leq 0.83, \\ x^2+ab, & \text{если } x > 0.83; \end{cases}$$

$a=-5/3; b=2.3; c=-3.335;$

$$23) \quad y(x) = \begin{cases} \cos(x^3-ab-a^{1/2}), & \text{если } a < x \leq b, \\ (x-a)^3+b^2, & \text{если } x \leq a, \\ (c+d)^2+cx, & \text{если } x > b; \end{cases}$$

$a=0.24; b=2; c=8/3; d=0.5;$

$$24) \quad y(x) = \begin{cases} 9.25-|ac+dx|, & \text{если } a \leq x \leq d, \\ 4(d^2/a+c^2x/b^2), & \text{если } x > d, \\ ad^2cx, & \text{если } x < a; \end{cases}$$

$a=-2.3; b=5.61; c=4; d=2.5;$

$$25) \quad y(x) = \begin{cases} \cos(ax+z), & \text{если } a \leq x < z, \\ x+z+4.28, & \text{если } x < a, \\ 9(x+za), & \text{если } x \geq z; \end{cases}$$

$a=-0.025; z=0.25;$

Проверьте правильность определения функции $y(x)$: задайте значения переменной x из разных условий и выведите результаты.

Задание 5. Определите переменную x как ранжированную переменную в произвольно выбранном диапазоне, не указав второй элемент. Каким будет шаг аргумента? Измените определение переменной, введя второй элемент. Как изменился шаг аргумента? Ответ проиллюстрируйте выводом таблиц значений ранжированной переменной и таблиц значений функций $d(x, y, a, b, p, q)$, $h(x, y)$ и $g(d, h)$.

Лабораторная работа Mathcad: Создание графиков

Цель работы: получение навыков создания и форматирования двумерных и трехмерных графиков в пакете Mathcad.

Теоретические сведения

1. Состав панели инструментов Math → Graph (Математика → График)

В системе Mathcad предусмотрена возможность вставки в документ графиков различного типа. Полный их список можно узнать из подменю команды **Insert → Graph (Вставка → График)** (рис. 3.1).

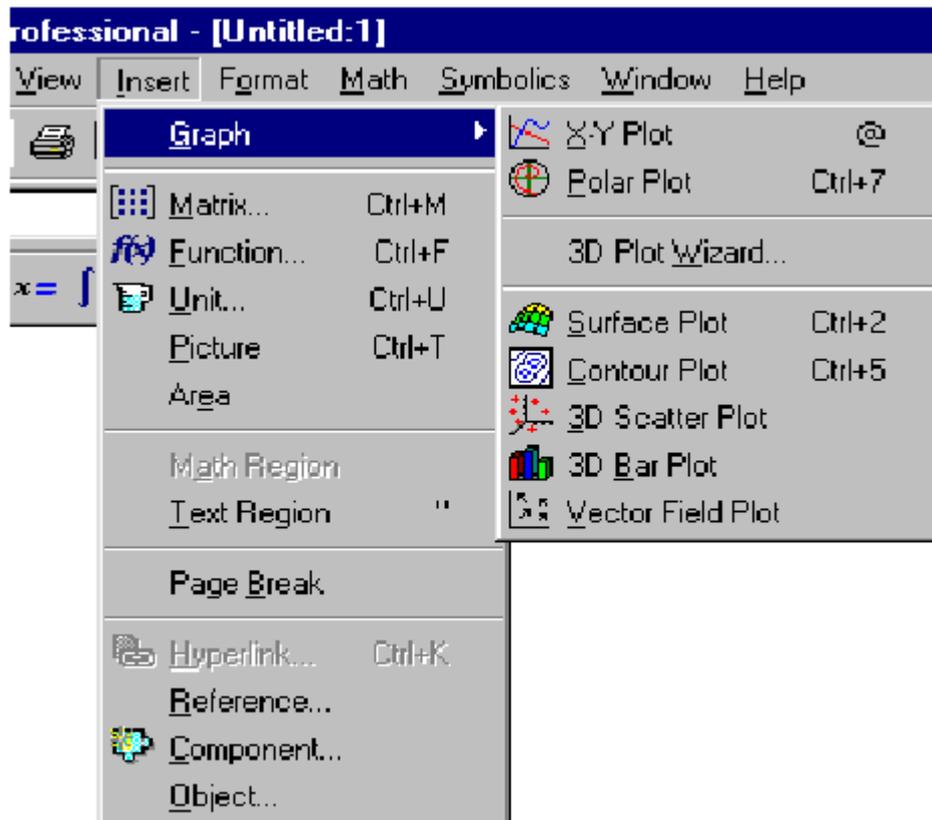


Рис. 3.1. Раскрытое подменю **Graph**

Все типы графиков можно разбить на две большие группы:

Двумерные графики	Трёхмерные графики
<ul style="list-style-type: none"> ▪ X–Y Plot (Декартов график) ▪ Polar Plot (Полярный график) 	<ul style="list-style-type: none"> ▪ Surface Plot (График поверхности) ▪ Contour Plot (Контурный график, линии уровня поверхности) ▪ 3D Scatter Plot (Точечный график) ▪ 3D Bar Plot (Трёхмерная гистограмма) ▪ Vector Field Plot (Векторное поле)

Деление графиков на типы несколько условно, так как, управляя установками многочисленных параметров, можно создавать комбинации различных типов графиков, а также новые типы. Например, двумерная гистограмма распределения (столбиковая диаграмма) или ступенчатый график являются разновидностями простого X–Y-графика (рис. 3.2, 3.3).

Информационные технологии в менеджменте

Большинство параметров графического процессора, необходимых для построения графиков, по умолчанию задаются автоматически. Поэтому для начального построения графика достаточно выбрать его тип. Графики могут иметь различные размеры и перемещаться в окне редактирования документа точно так же, как и любая другая область Mathcad-документа.

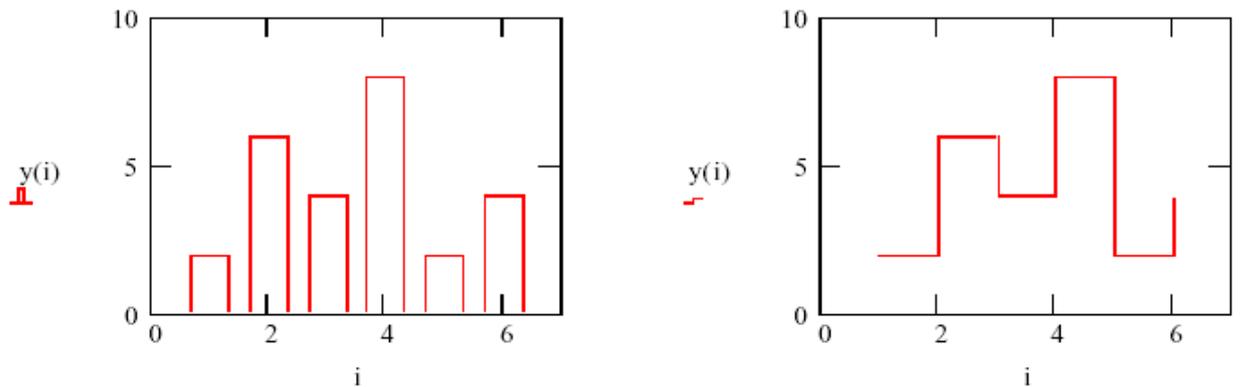


Рис. 3.2. Примеры столбиковой диаграммы и ступенчатого графика

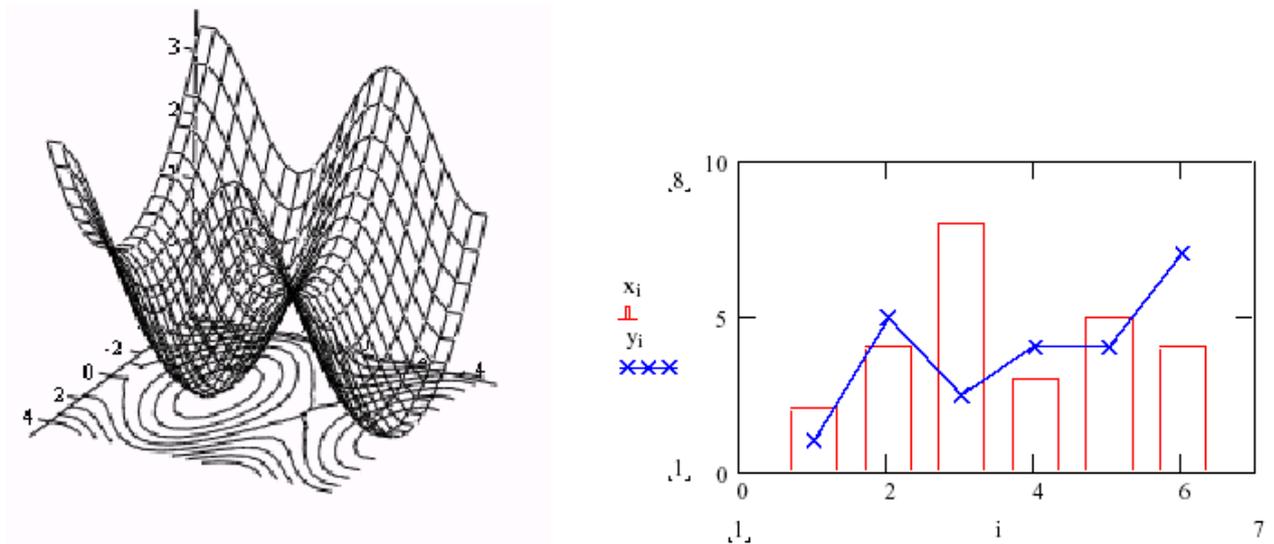


Рис. 3.3. Примеры комбинированных графиков

Вывести шаблон можно любым из следующих способов:

1. Выбором из меню команды **Insert** → **Graph** (**Вставка** → **График**) (рис. 3.1) соответствующего пункта.
2. Нажатием соответствующей кнопки на панели инструментов **Math** → **Graph** (**Математика** → **График**) (рис. 3.4).



Рис. 3.4. Панель инструментов Graph

3. Нажатием сочетания двух клавиш.

Рассмотрим более подробно команды меню **Math** → **Graph**:

 **X-Y Plot (Декартов график)** клавиша @. Служит для построения графика функции $y=f(x)$ в виде связанных друг с другом пар координат (x_i, y_i) при заданном промежутке изменения для i .

 **Polar Plot (Полярный график)** клавиши Ctrl+7. Служит для построения графика функции $r(q)$, заданной в полярных координатах, где полярный радиус r зависит от полярного угла q .

 **Surface Plot (График поверхности)** клавиши Ctrl+2. Служит для представления функции $z=f(x, y)$ в виде поверхности в трехмерном пространстве. При этом должны быть заданы векторы значений x_i и y_i , а также определена матрица вида $A_{i,j} = f(x_i, y_j)$. Имя матрицы A указывается при заполнении рамки-шаблона.

 **Contour Plot (Карта линий уровня)**. Строит диаграмму линий уровня функции вида $z=f(x, y)$, т. е. отображает точки, в которых данная функция принимает фиксированное значение $z=const$.

 **3D Bar Plot (3D Столбиковая гистограмма)**. Служит для представления матрицы значений $A_{i,j}$ или отображения значений функции $z=f(x, y)$ в виде трехмерной столбчатой диаграммы.

 **3D Scatter Plot (3D Точечный график)**. Служит для точечного представления матрицы значений $A_{i,j}$ или отображения значений функции $z=f(x, y)$ в заданных точках. Эта команда может также использоваться для построения пространственных кривых. В этом случае при заполнении рамки-шаблона можно задать три координаты отдельными векторами одинаковой размерности в виде (X, Y, Z) .

 **Vector Field Plot (Векторное поле)**. Служит для представления двумерных векторных полей $V=(V_x, V_y)$. При этом компоненты векторного поля V_x и V_y должны быть представлены в виде матриц. При помощи этой команды можно построить поле градиента функции $f(x, y)$.

2. Построение X-Y-графиков

Незаполненный шаблон графика (рис. 3.5) представляет собой пустой прямоугольник с двумя знакозаполнителями (знакоместами) в виде темных маленьких квадратов, расположенных на осях X и Y. Ось X (горизонтальная) — ось абсцисс точек графика $y=f(x)$, а ось Y (вертикальная) — ось ординат точек графика. На рис. 3.5 курсор ввода размещен на знакозаполнителе оси X. Переход с одного знакоместа на другое выполняется перемещением указателя мыши с последующим щелчком левой кнопкой или нажатием клавиши **ТАВ**.

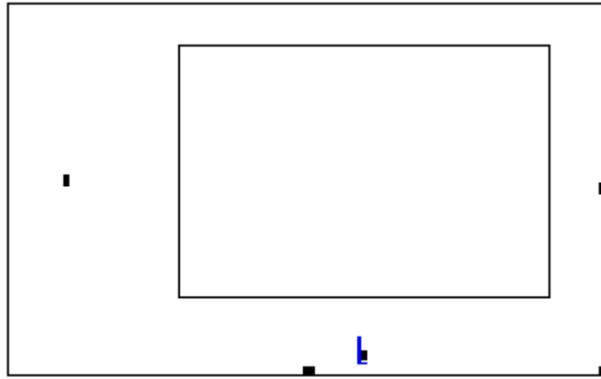


Рис. 3.5. Шаблон двумерного графика

Mathcad строит график линии по точкам, последовательно соединяя соседние точки отрезком. Их координаты определяются выражениями, указанными в соответствующих заполнителях осей. Таким образом, график линии — это ломаная. Mathcad предусматривает два способа построения графика функции одной переменной $y = f(x)$:

1. *Без задания переменной диапазона (быстрое построение графика)*. При этом пределы изменения независимой переменной x автоматически задаются от -10 до 10 , но с учетом ее допустимых значений (так, например, для функции $\ln(x-3)$ будут установлены пределы от 3 до 10).

2. *Заданием переменной диапазона (ранжированной переменной)*. Пользователь сам указывает границы изменения переменной x , учитывая допустимое множество ее значений, причем область задания переменной в документе должна располагаться строго левее или выше графической области.

Если график построен первым из перечисленных способов (*без задания переменной диапазона*), то автоматически установленное масштабирование может оказаться не совсем удачным, например, числовые значения меток осей не будут представлены неокругленными десятичными числами (как это хотелось бы), или не совсем ясно, каково поведение функции на отдельных промежутках изменения переменной x . Однако для первичного графического представления функции предпочтительнее автоматическое масштабирование. А затем, анализируя график, выбирается более подходящий масштаб.

Пример. Построить график функции $\cos(x)$:

Информационные технологии в менеджменте

$$x := -2\pi, -2\pi + 0.001.. 2\pi$$

$$2\pi = 6.283 \quad -2\pi = -6.283$$

x =

-6.283
-5.283
-4.283
-3.283
-2.283
-1.283
-0.283
0.717
1.717
2.717
3.717
4.717
5.717

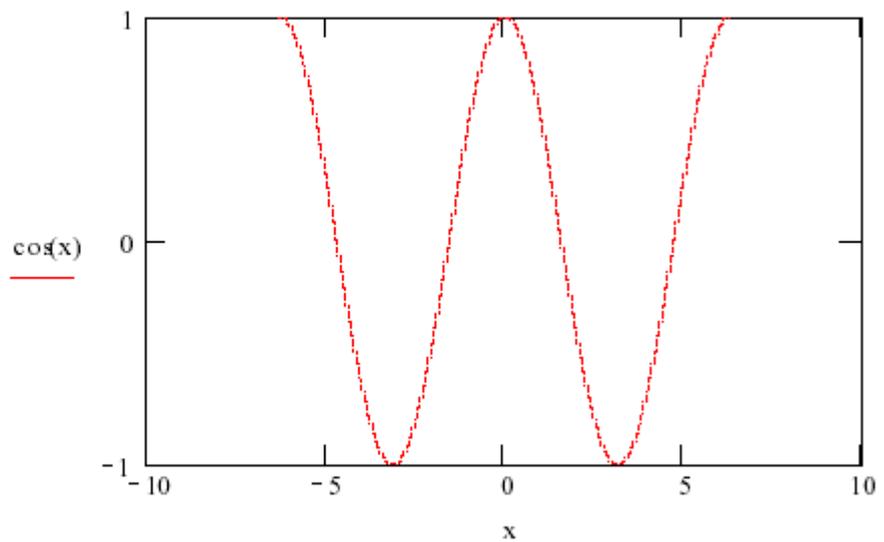


Рис. 3.8.

Как видно из рис. 3.8, график был построен только в пределах заданного диапазона изменения переменной x (от -6.283 до 6.283), а автоматически установленные границы на оси X им не соответствуют. Если графическую область сделать активной (переместите курсор на область графика), то вы увидите (рис. 3.9) числовые значения концов диапазона задания переменной x и границы диапазона изменения выражения на оси Y .

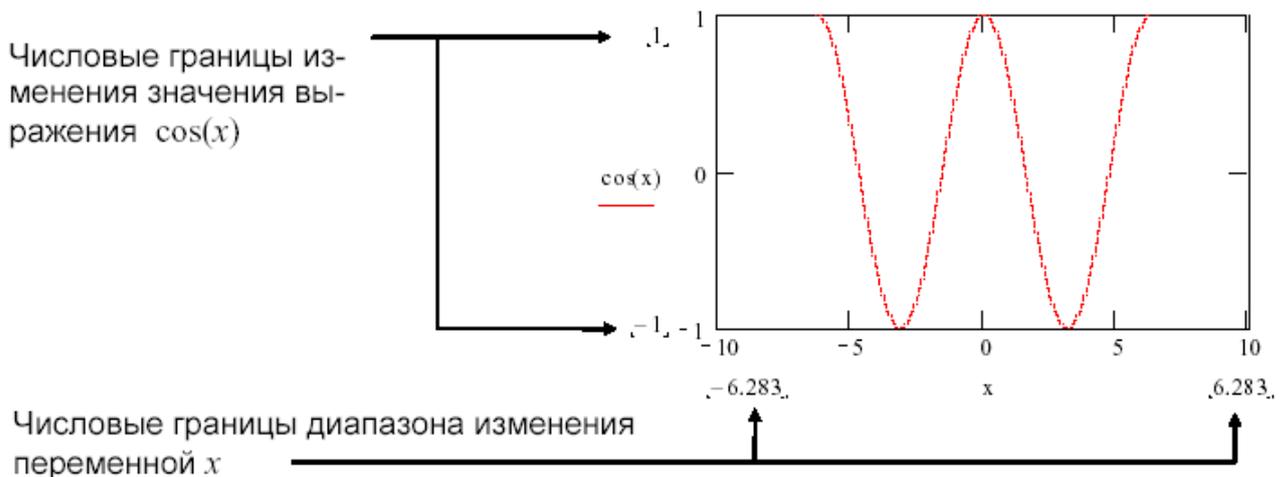


Рис. 3.9

Вы можете построить график на другом промежутке задания переменной x . Для этого достаточно отредактировать область задания переменной диапазона x .

Информационные технологии в менеджменте

Сделав активной область задания переменной диапазона x , отредактируйте ее следующим образом:

$$x := -7.5, -7.49, 12.1$$

После такого изменения график автоматически обновляется (рис. 3.10).

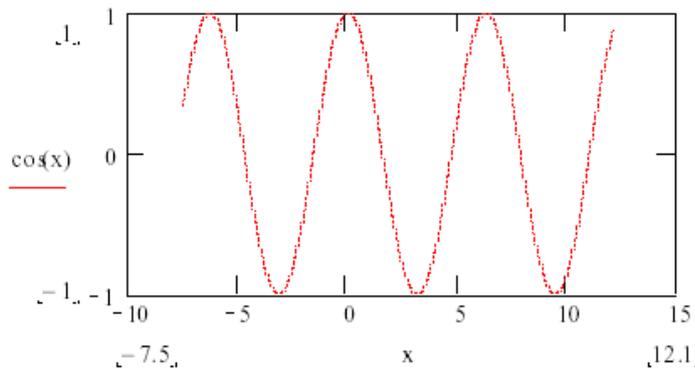


Рис. 3.10

Построение нескольких графиков в одной системе координат. В одной системе координат можно строить графики нескольких функций. Для этого необходимо ввести определяющие их выражения, заполняя соответствующее знакоместо на оси Y. Элементы списка выражений разделяются запятыми. Ввод разделителя-запятой приводит к появлению очередного знакоместа для его заполнения (рис. 3.15). После заполнения необходимых знакомест и выхода из области можно увидеть результат построения. Графики различных функций будут выведены линиями, отличающимися цветом и/или стилем, что зависит от настройки графических параметров. Если какое-либо знакоместо в списке выражений оказалось незаполненным, то будет выведено сообщение об ошибке.

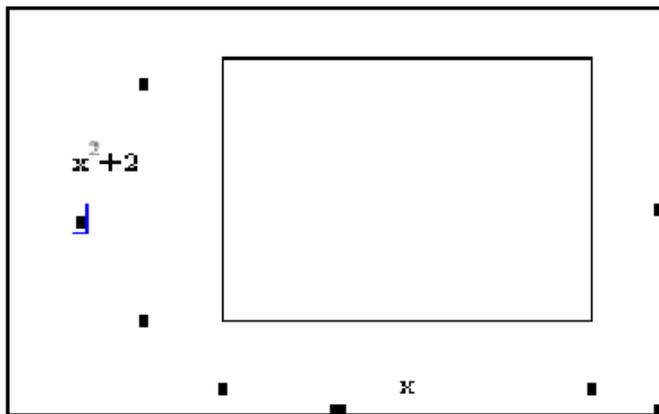


Рис. 3.15. Ввод выражений на оси Y

Пример.

В одной графической области построим графики следующих функций:

$$y = \frac{x \cdot \sin(x)}{5}, \quad y = \frac{5 \cdot x^2 - 15}{x^2 + 5}, \quad y = 0.$$

Если все функции зависят от одной переменной x и рассматриваются на одном и том же промежутке ее задания, то при заполнении знакоместа на оси X достаточно ввести имя переменной один раз.

Возможный вариант оформления MathCAD-документа показан на рис. 3.16.

$x := -20, -19.99, 20$

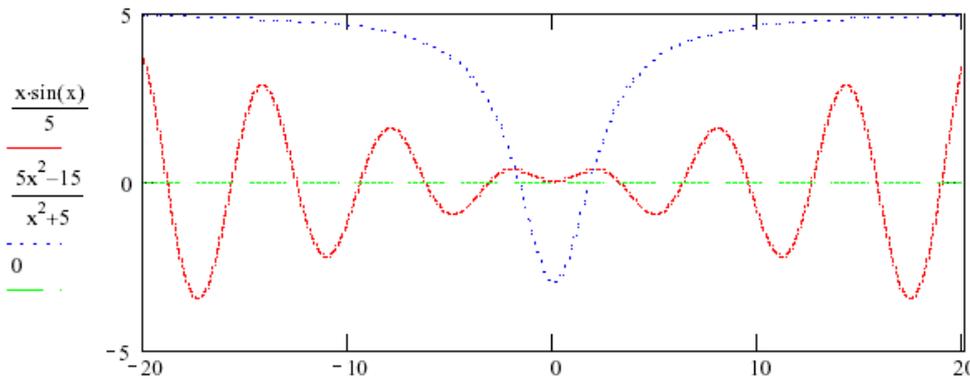


Рис. 3.16

Если выражение, определяющее функцию, является достаточно громоздким для ввода его в области построения графика или его необходимо неоднократно использовать в процессе решения задачи, то целесообразнее задать его с помощью функции пользователя. Если выражений несколько, то им должны соответствовать функции с разными именами.

Например, предыдущее задание можно выполнить так, как показано на рис. 3.17.

$x := -20, -19.99, 20$

$$f(x) := \frac{x \cdot \sin(x)}{5} \quad g(x) := \frac{5x^2 - 15}{x^2 + 5}$$

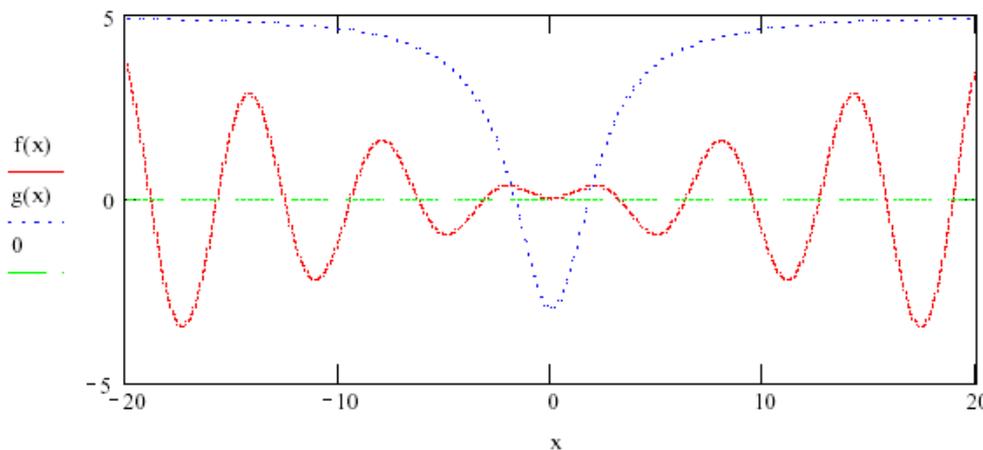


Рис. 3.17

Пример.

Исследуем изменение периода функции $y = \sin(\omega \cdot x)$ в зависимости от частоты ω . Зададим значения ω равными 1, 2 и 3 (рис. 3.18).

$$x := 0, 0.01..2\pi$$

$$f(x) := \sin(x)$$

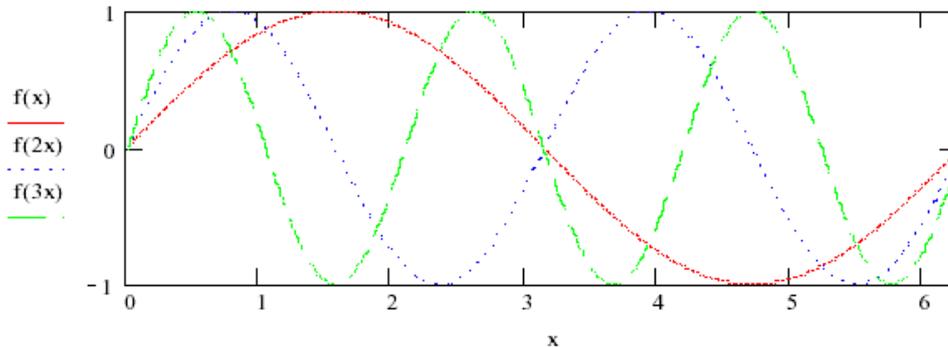


Рис. 3.18

Пример. Исследуем функцию на непрерывность, построив ее график:

$$y = \begin{cases} x^2, & 0 \leq x < 1, \\ 2, & 1 \leq x < 4, \\ 6\cos(7x), & 4 \leq x \leq 6. \end{cases}$$

Так как данная функция на разных промежутках задается различными выражениями (является кусочной), то ее график можно получить, построив отдельно графики трех выражений, рассматриваемых на соответствующих интервалах. Для чего удобно задать три различные переменные диапазона, например x , t , v (рис. 3.20), и ввести их списком при заполнении знакоместа на оси X. Элементы списка разделяются запятыми. Ввод запятой приводит к появлению нового знакоместа для его заполнения очередным элементом списка. На рис. 3.19 показан фрагмент графической области при вводе переменных на оси X.

Каждому выражению из списка на оси Y соответствует элемент списка на оси X. Если какому-либо выражению на оси Y нет соответствующего выражения на оси X, то по умолчанию ему будет поставлен в соответствие последний элемент из списка на оси X.

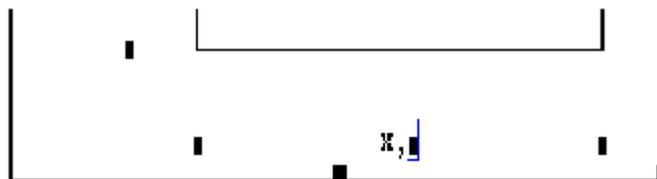


Рис. 3.19

Информационные технологии в менеджменте

$x := 0, 0.1..1$

$t := 1, 1.1..4$

$v := 4, 4.1..6$

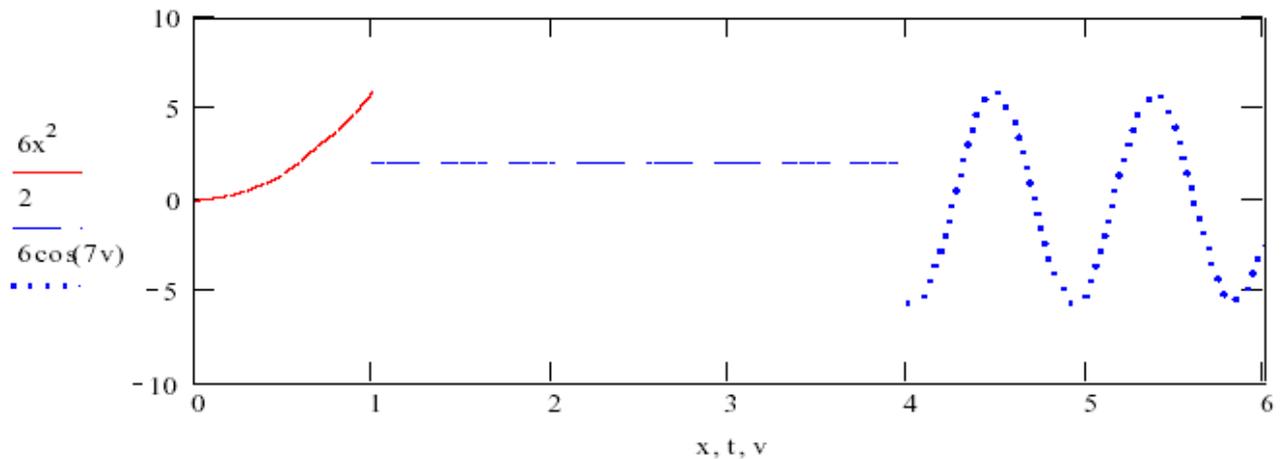


Рис. 3.20

Анализируя построенный график (рис. 3.20), можно сделать вывод о наличии двух точек разрыва функции $x_1=1$ и $x_2=4$ на заданном интервале $[0;6]$.

Рассматриваемую функцию на интервале $[0; 6]$ можно задать с помощью одного выражения таким образом:

$$y(x) := \text{if}(x < 1, 6x^2, \text{if}(x \geq 4, 6 \cos(7x), 2))$$

или

$$y(x) := 6x^2 \cdot (x \geq 0) \cdot (x < 1) + 2 \cdot (x \geq 1) \cdot (x < 4) + 6 \cos(7x) \cdot (x \geq 4) \cdot (x < 6)$$

Поскольку Mathcad строит график, последовательно соединяя соседние точки, то, выполнив построения так, как показано на рис. 3.21, увидим график в виде непрерывной (без разрывов) линии.

$$y(x) := 6x^2 \cdot (x \geq 0) \cdot (x < 1) + 2 \cdot (x \geq 1) \cdot (x < 4) + 6 \cos(7x) \cdot (x \geq 4) \cdot (x < 6)$$

$x := 0, 0.001..6$

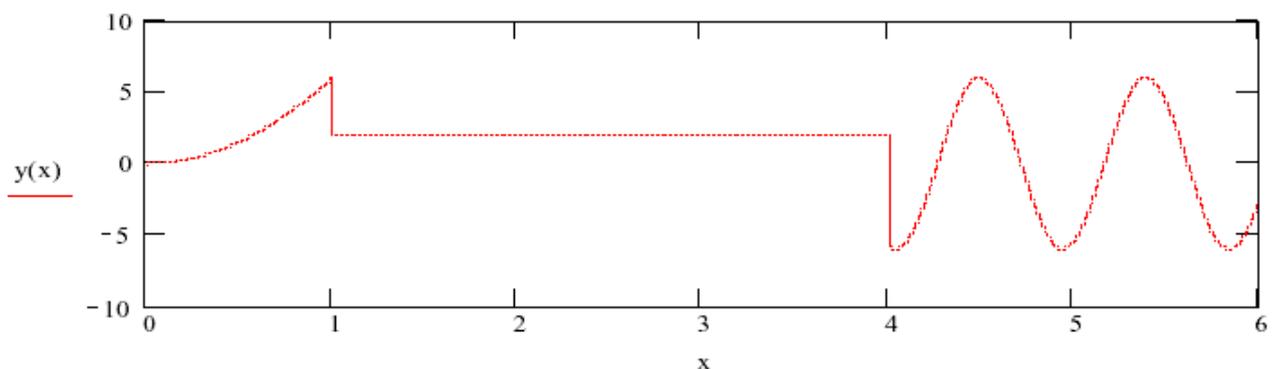


Рис. 3.21

Устранить недостаток такого построения можно, задав три переменные диапазона и строя график одной и той же функции $y(x)$, но на разных промежутках задания ее аргумента (рис. 3.22).

Информационные технологии в менеджменте

$$y(x) := 6x^2 \cdot (x < 1) + 2 \cdot (x \geq 1) \cdot (x < 4) + 6 \cos(7x) \cdot (x \geq 4)$$

$$x := 0, 0.001..0.99 \quad t := 1, 1.01..3.99 \quad v := 4, 4.01..6$$

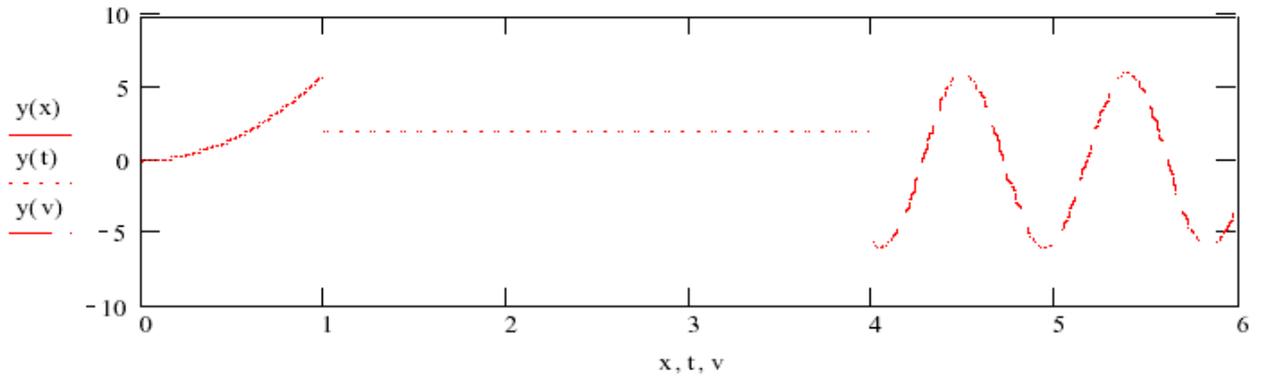


Рис. 3.22

Построение графиков параметрически заданных кривых. Графики параметрически заданных кривых: $x = x(t)$, $y = y(t)$ строятся на шаблоне двумерного графика. В этом случае при заполнении знакомест на осях X и Y вводятся выражения, задающие соответствующие координаты x и y точек кривой. И так же, как и при построении графиков функций, аргумент (параметр) t может быть заранее не задан (быстрое построение графика) или задан как переменная диапазона.

В одной системе координат могут быть построены графики различных кривых. Примеры построения кривых показаны на рис. 3.23—3.26.

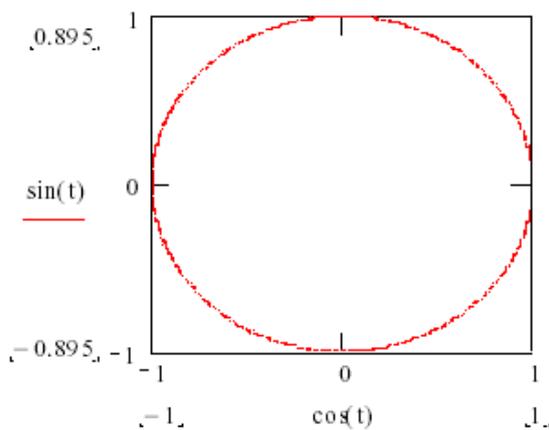


Рис. 3.23. График параметрически заданной окружности радиуса R=1: $x=\cos(t)$, $y=\sin(t)$

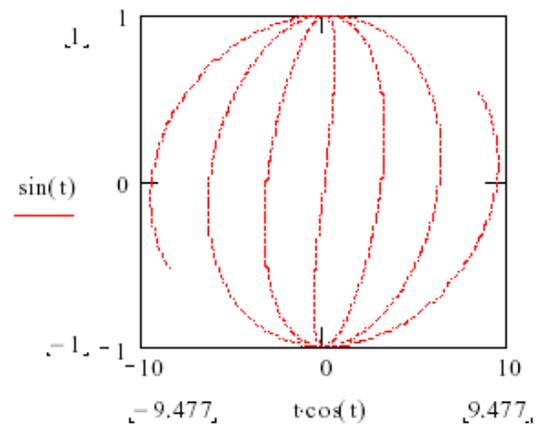


Рис. 3.24. График кривой $x=t\cos(t)$, $y=\sin(t)$

Информационные технологии в менеджменте

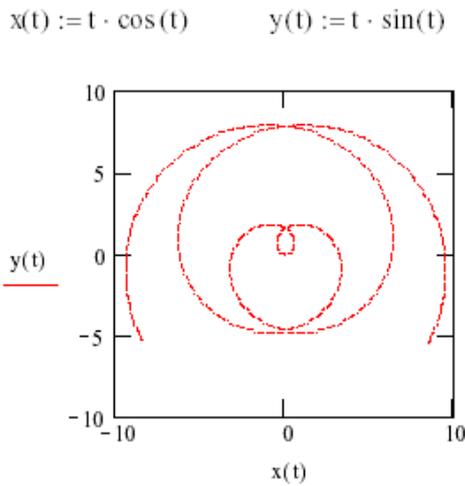


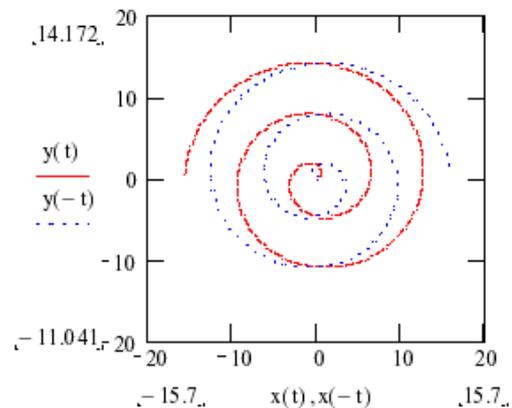
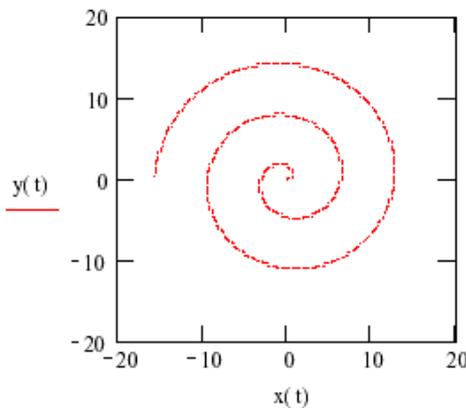
Рис. 3.25. Графики параметрически заданной кривой.

Первый построен без задания переменной t ; второй и третий – с указанием границ ее изменения.

На втором графике построена одна спираль кривой (при положительных значениях параметра t).

На третьем – две спирали, построенные отдельно при положительных и отрицательных значениях параметра

$t := 0, 0.01..5\pi$



Заметим, что графики любых функций можно рассматривать как графики параметрически заданных кривых, для которых роль параметра t играет независимая переменная x .

$x := 0, 0.001..6\pi$

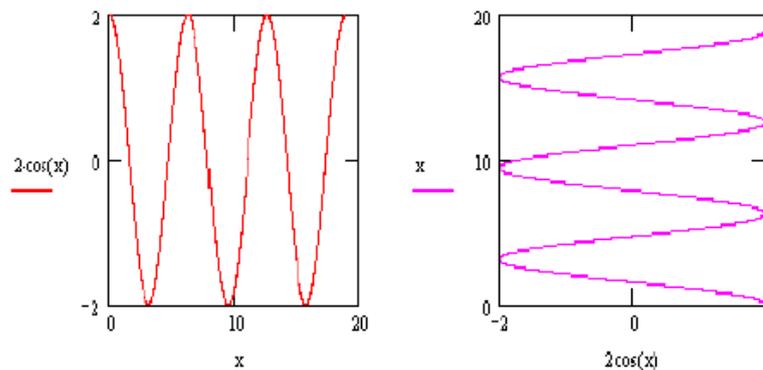


Рис. 3.26. Изменение графика функции $y=2\cos x$ (первый график) в результате перестановки выражений, заданных на осях X и Y (поворот графика на угол 90°)

3. Построение графиков в полярной системе координат

В полярной системе координат любая точка определяется двумя координатами R и φ , где R — полярный радиус; φ — полярный угол точки, $R=R(\varphi)$.

Вывести шаблон графика в полярной системе координат можно одним из трех способов:

1. Выберите из меню команды **Insert** → **Graph** (рис. 3.1) пункт **Polar Plot** (**Полярный график**).
2. Или нажмите комбинацию клавиш **Ctrl+7**.

3. Или нажмите кнопку  на панели **Graph** (рис. 3.4).

Незаполненный шаблон графика представляет собой окружность с двумя значащими для ввода данных в виде темных маленьких прямоугольников (рис. 3.38).

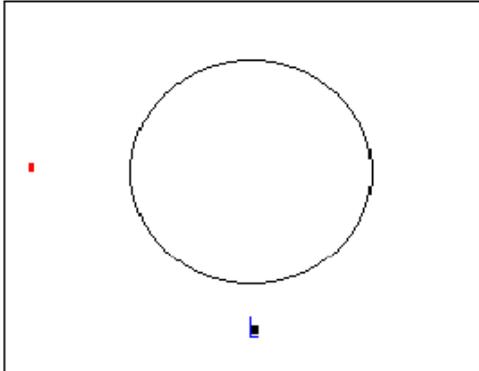


Рис. 3.38. Шаблон полярного графика с сообщением о необходимости заполнения знакомест ввода

Есть две возможности построения таких графиков:

- задавая пределы изменения переменной φ (причем переменная может иметь и другое имя), [рис. 3.39](#);
- без определения диапазона изменения переменной (быстрое построение), [рис. 3.40](#).

Информационные технологии в менеджменте

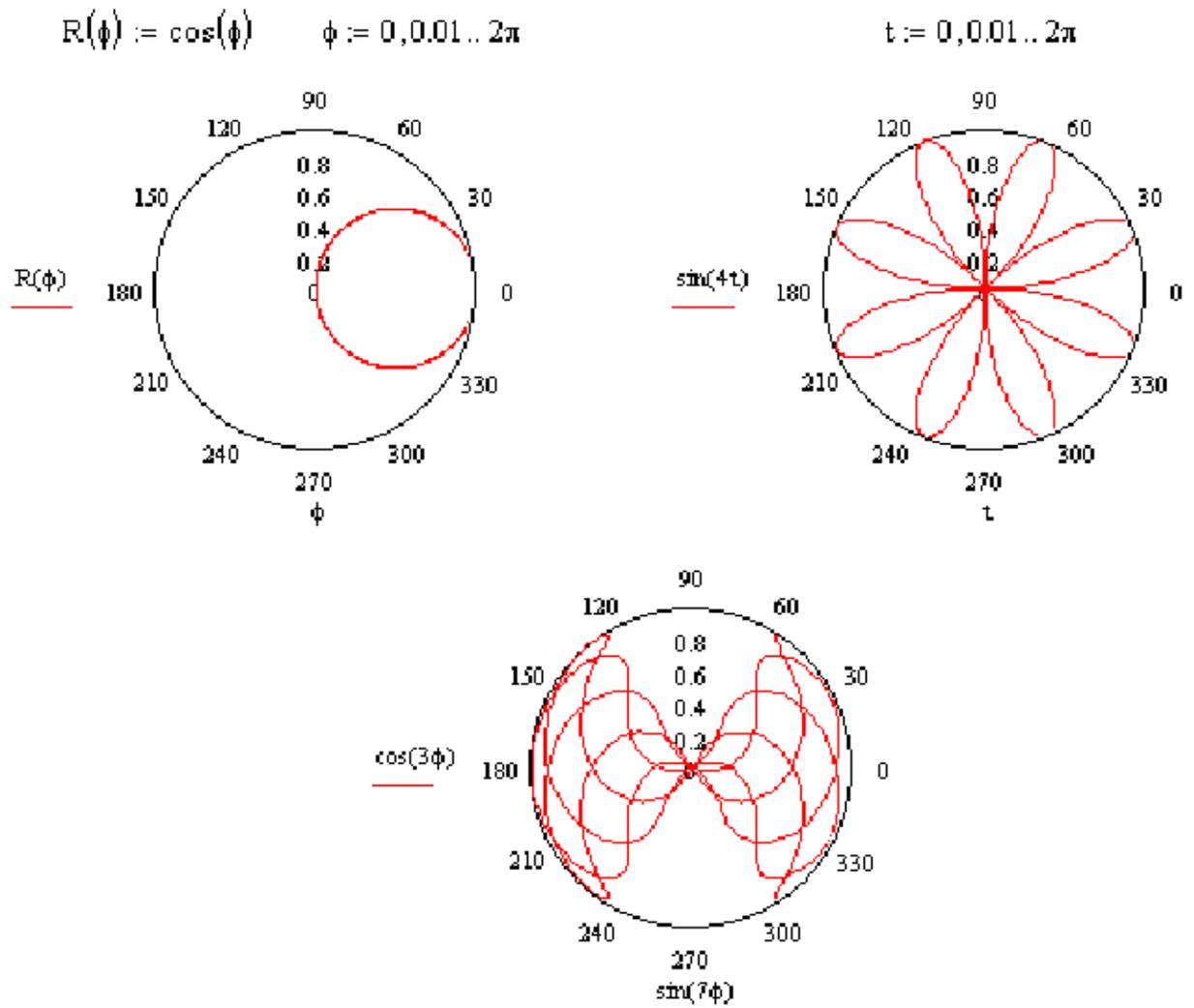


Рис. 3.39. Построение графиков с определением переменной

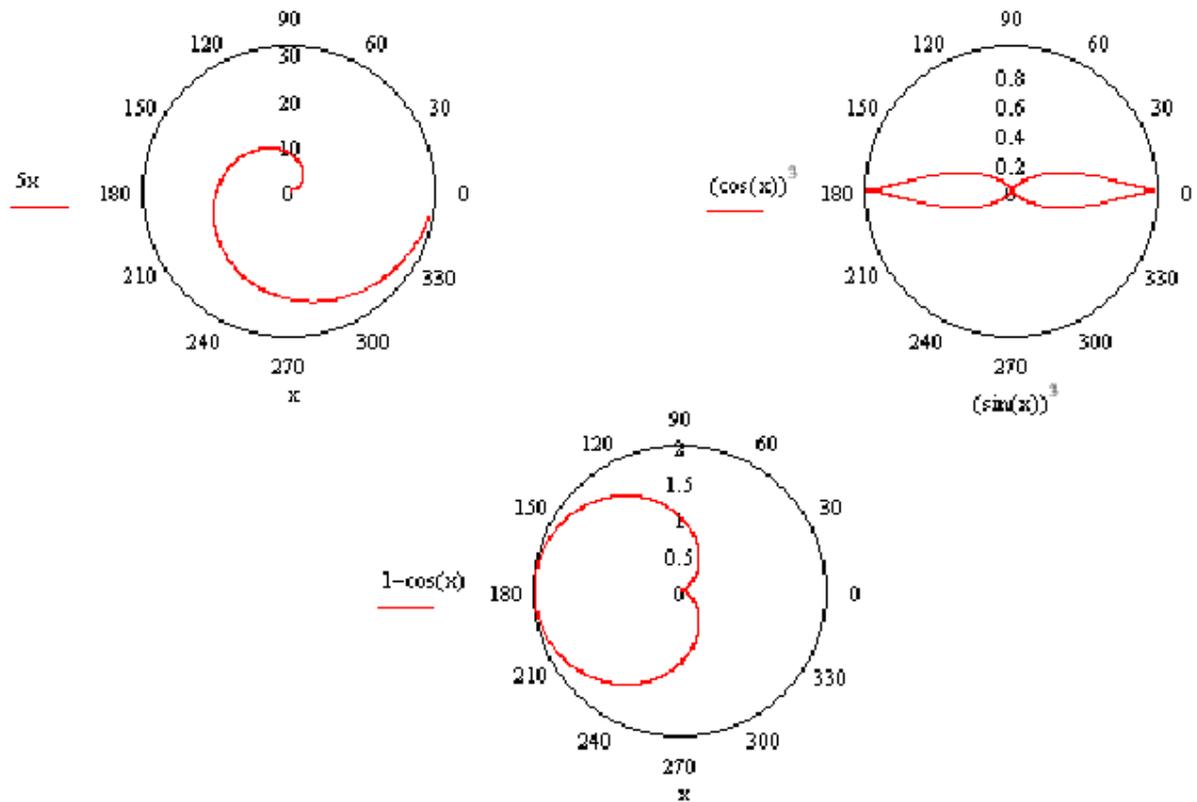


Рис. 3.40. Построение графиков без указания границ переменной (быстрое построение)

Аналогично двумерным графикам в полярной системе можно строить графики нескольких функций или кривых (рис. 3.41).

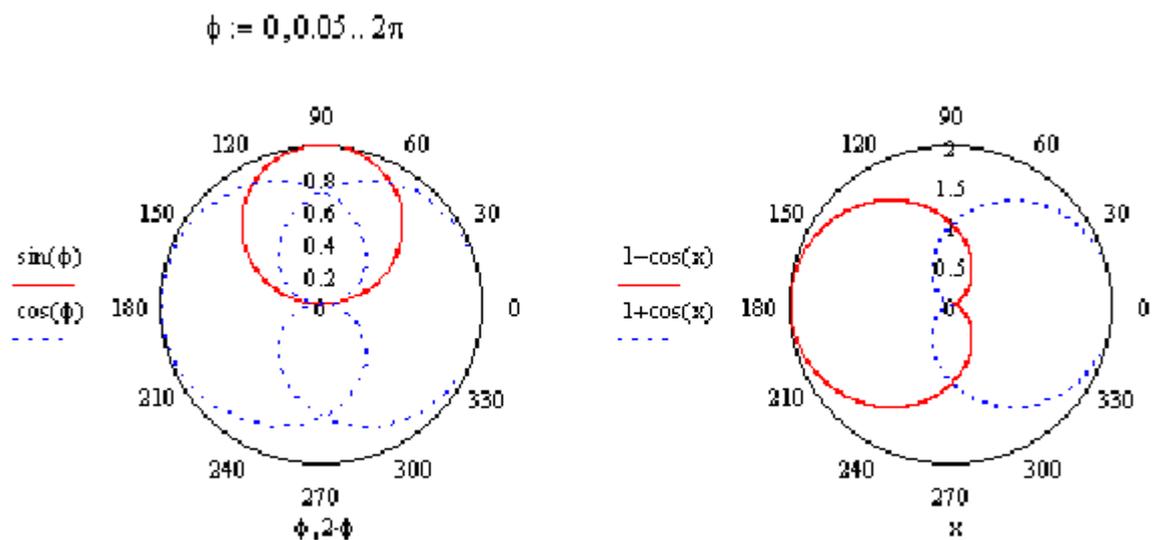


Рис. 3.41. Построение нескольких графиков на одном шаблоне

4. Форматирование двумерных графиков

Для вывода окна форматирования двумерного графика достаточно поместить указатель мыши в область графика и дважды щелкнуть левой кнопкой мыши. В окне документа появится окно форматирования (рис. 4).

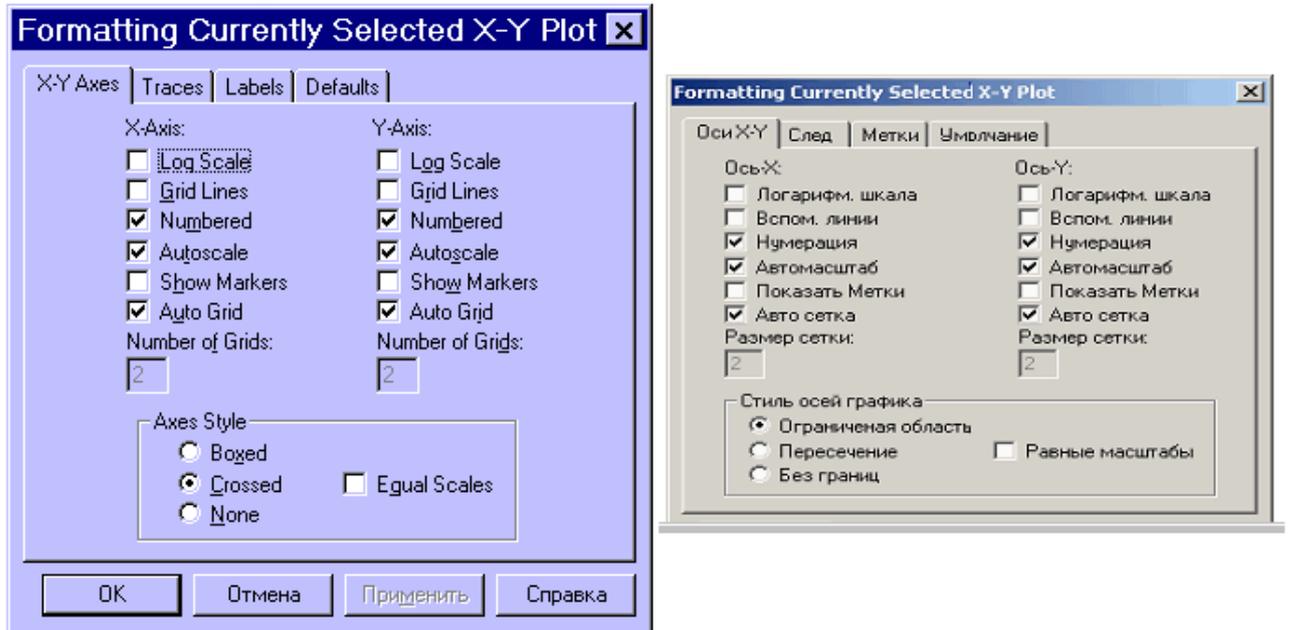


Рис. 4.

Оно имеет ряд вкладок:

- **X-Y Axes (Оси X-Y);**
- **Traces (Трассировки);**
- **Labels (Метки);**
- **Defaults (Умолчание).**

Вкладка становится активной, если установить на ее имя указатель мыши и щелкнуть левой кнопкой.

Первая из вкладок **X-Y Axes (Оси X-Y)** позволяет форматировать оси координат:

- **Log Scale (Логарифмическая шкала)** — задает логарифмические оси, в этом случае границы графика должны задаваться положительными числами;
- **Grid Lines (Вспомогательные линии)** — задает отображение сетки;
- **Numbered (Нумерация)** — задает отображение подписи к маркировкам на осях;

— **Autoscale (Автомасштаб)** — задает автоматическое нахождение подходящих границ для осей. Но если вы сами зададите в соответствующих ячейках минимальные и максимальные значения X_{min} , X_{max} , Y_{min} , Y_{max} , именно эти значения будут использоваться для определения границ графика;

— **Show Markers (Показать метки)** — если установить эту опцию, то в графической области появятся четыре дополнительные ячейки для создания красных линий маркировки, соответствующих двум специальным значениям x и двум специальным значениям y ;

— **Auto Grid (Авто сетка)** — при установке этой опции число линий сетки определяет Mathcad.

— **Axes Style (Стиль осей графика)** — группа кнопок этой области позволяет выбрать следующие варианты представления осей: **Boxed (Ограниченная область)**, **Crossed (Пересечение)** — оси пересекаются в точке

Информационные технологии в менеджменте

с координатами (0; 0), **None (Без границ)**. Флажок **Equal Scales (Равные масштабы)** позволяет задать одинаковый масштаб для обеих осей.

Форматирование оси графика можно произвести, выполнив на ней двойной щелчок.

Для изменения типа линий графиков необходимо активизировать вкладку **Traces (Трассировки)** (рис. 5):

— **Legend Label (Легенда)** — каждой кривой можно поставить в соответствие некоторый текст, называемый легендой. Легенда отображается в нижней части графической области, а рядом с каждой легендой отображается тип линии соответствующей кривой;

— **Symbol (Символ)** — позволяет выбрать символ для каждой точки кривой (плюс, крестик, кружок и др.);

— **Line (Линия)** — можно выбрать один из следующих типов линий: solid (сплошная), dash (штриховая), dot (точечная) или dadot (штрихпунктирная). Это поле списка доступно в случае, если в поле **Type (Тип)** выбран элемент lines;

— **Color (Цвет)** — задается цвет представления кривой на экране;

— **Type (Тип)** — позволяет выбрать один из видов графика: в виде линий, в виде точек и т. п.;

— **Weight (Вес)** — позволяет задавать толщину линий графика.

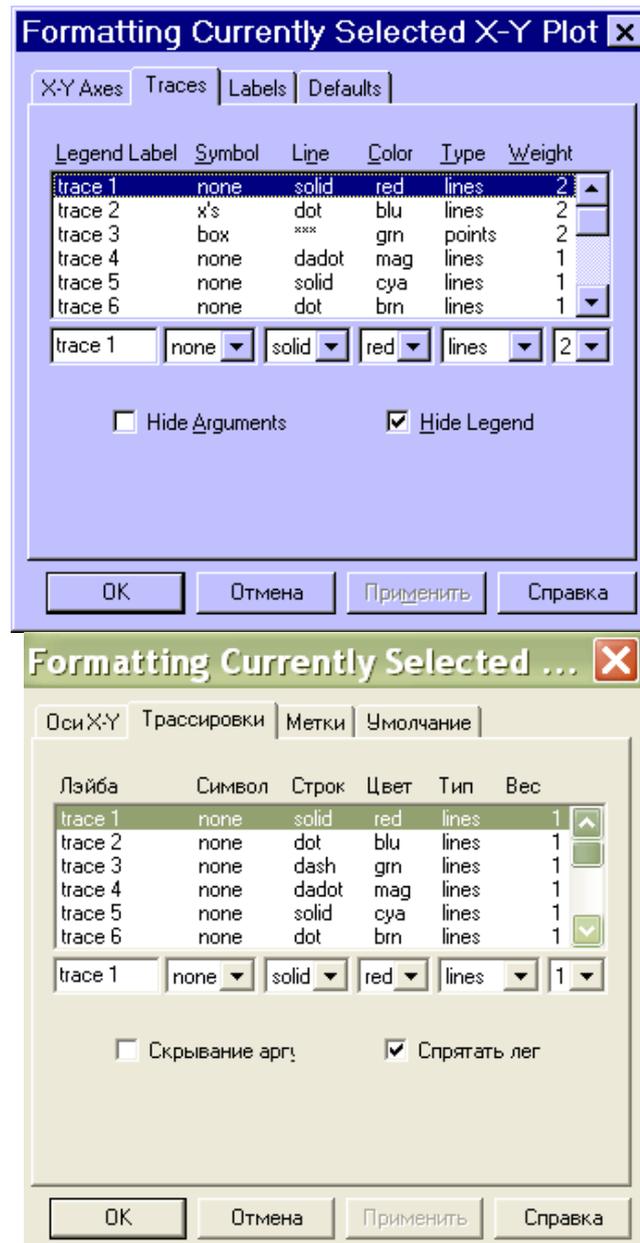


Рис. 5.

В нижней части вкладки **Traces** расположены опции:

— **Hide Arguments (Скрыть аргументы)** — эта опция по умолчанию отключена. В этом случае под именем функции рядом с осью ординат указывается текущий тип линий. Если установить данную опцию, указание типа линий исчезнет;

— **Hide Legend (Скрыть легенду)** — по умолчанию легенда не отображается. Если вы хотите отобразить под графиком текст легенды, его необходимо перед этим ввести в поле **Legend Lable (Легенда)** и подтвердить ввод, выполнив щелчок на кнопке **Применить**.

Вкладка **Labels (Метки)** позволяет ввести заголовок графика и подписи для осей (рис. 6).

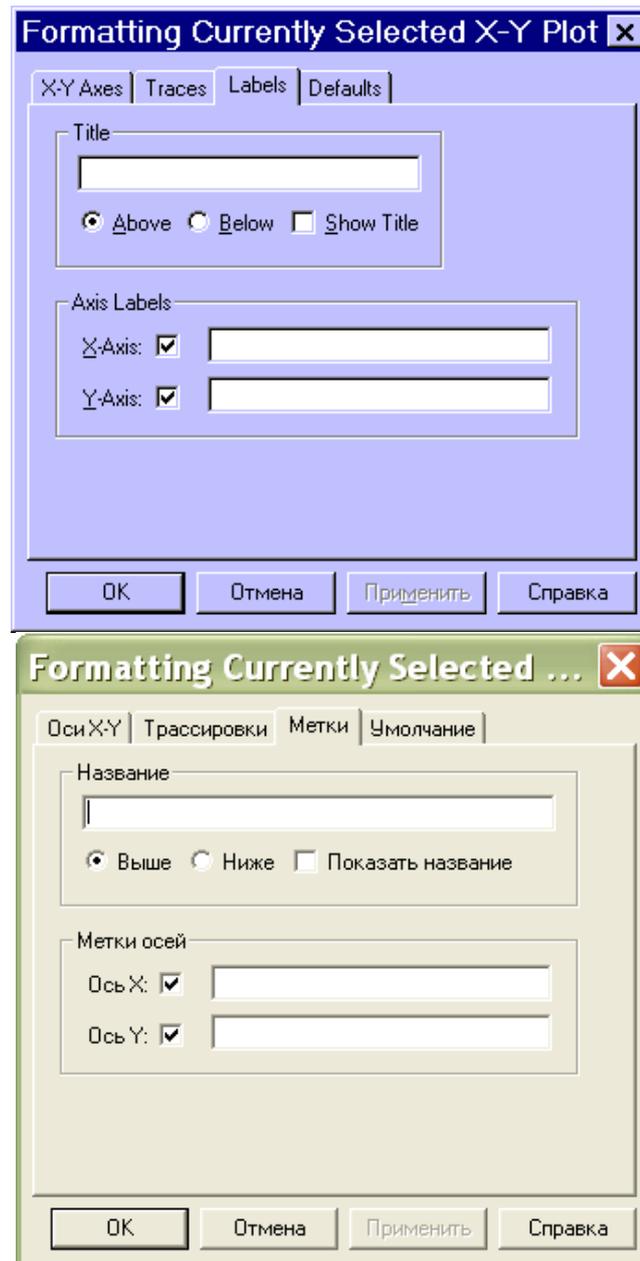


Рис. 6.

В меню **Format** → **Graph** (**Формат** → **График**) содержится команда **Zoom** (**Изменение масштаба**). При помощи этой команды можно увеличить фрагмент графика, предварительно выделив его протаскиванием мышки с нажатой левой клавишей. После отпускания клавиши координаты углов выделенной области будут отображены в полях окна **X-Y Zoom** (рис. 7). При помощи кнопки **Zoom** (**Масштаб +**) фрагмент можно увеличить, при помощи кнопки **Unzoom** (**Масштаб -**) отменить выделение фрагмента, а при помощи кнопки **Full View** (**Обзор**) — восстановить первоначальный вид графика. Если вы увеличили фрагмент графика, то при щелчке на кнопке **OK** в документе будет отображаться только этот фрагмент.

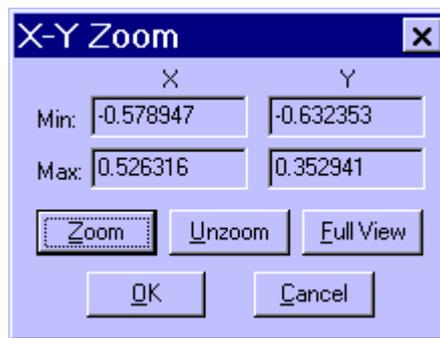
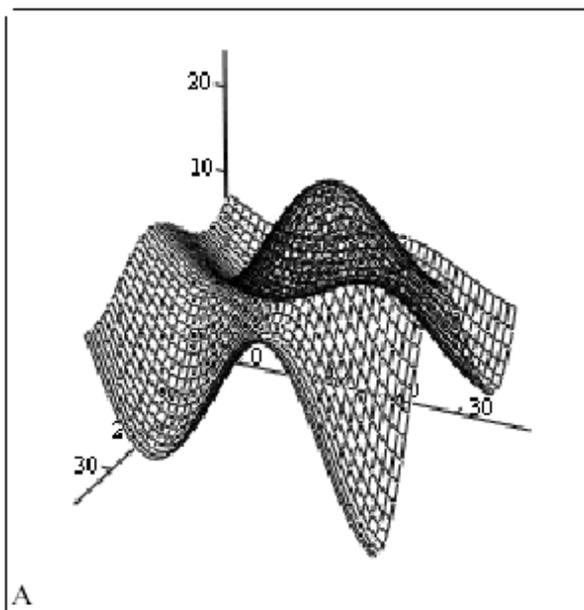


Рис. 7.

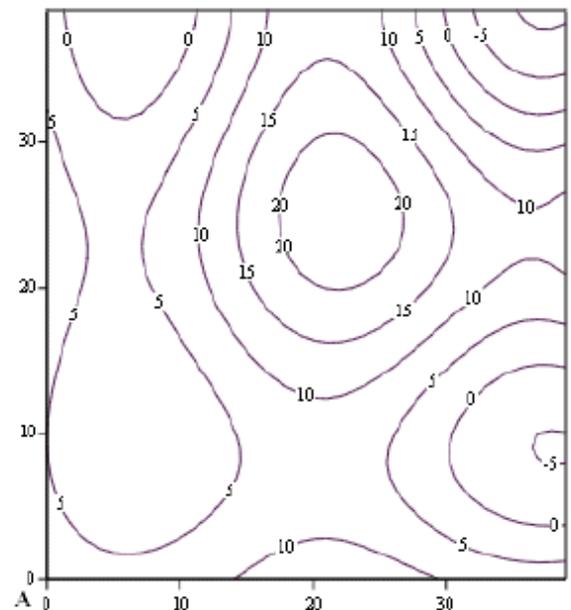
5. Построение трехмерных графиков

В Mathcad возможно построение следующих видов трехмерных графиков:

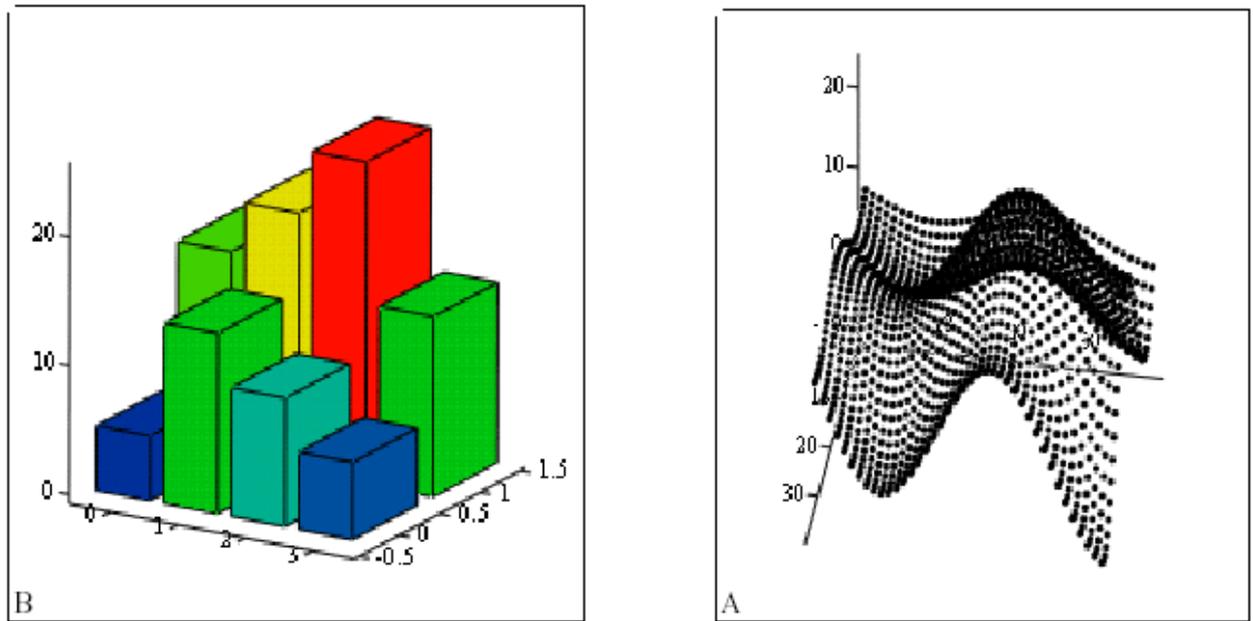
- График трехмерной поверхности (Surface Plot), [рис. 7.1а](#);
- График линий уровня (Contour Plot), [рис. 7.1б](#);
- Трехмерная столбчатая диаграмма/гистограмма (3D Bar Plot), [рис. 7.1в](#);
- Трехмерное множество точек (3D Scatter Plot), [рис. 7.1г](#);
- Векторное поле (Vector Field Plot).



а



б



В

Г

Рис. 7.1. Виды трехмерных графиков

Чтобы создать трехмерный график, требуется нажать кнопку с изображением любого из типов трехмерных графиков на панели инструментов **Graph** или выполнить команду вставки графика **Insert**→**Graph** (**Вставка**→**График**) или воспользоваться комбинацией клавиш **Ctrl+2** (для Surface Plot) или **Ctrl+5** (для Contour Plot).

В результате на рабочем листе будет выведена пустая область графика с тремя осями (рис.7.3) и единственным знакоместом в нижнем левом углу. В это знакоместо следует ввести имя функции двух переменных (не указывая список аргументов) для быстрого построения трехмерного графика, либо имя матричной переменной, которая задаст распределение данных на плоскости. Понятно, что функция или матрица к этому моменту должна быть определена.

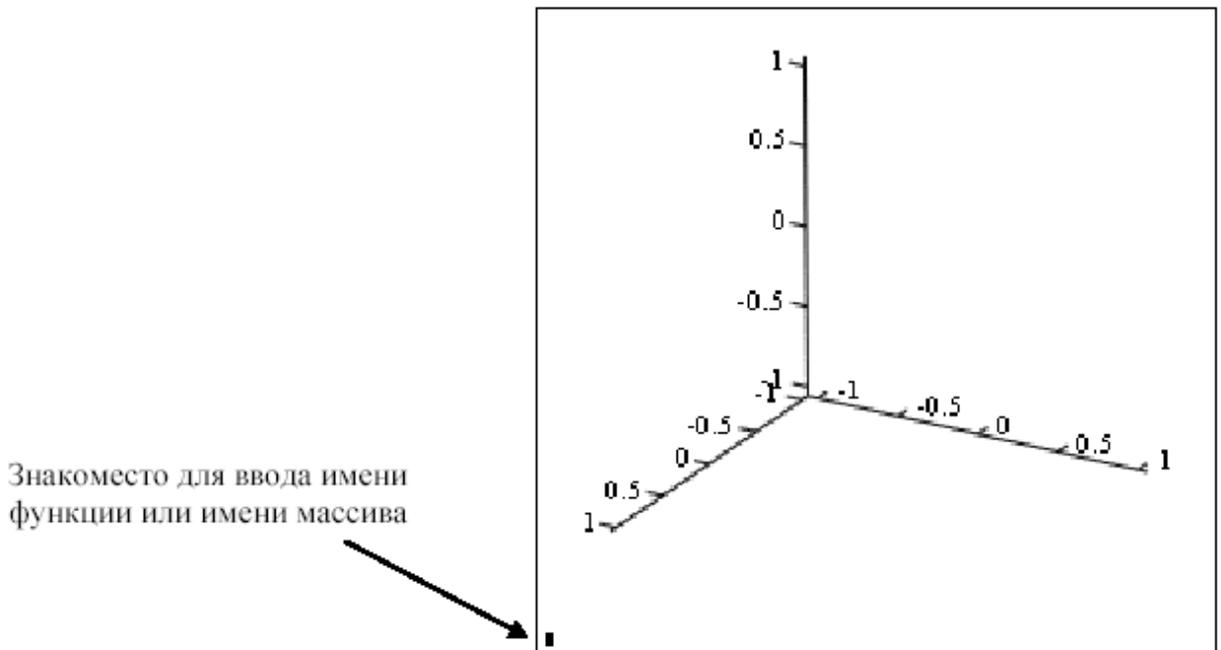


Рис. 7.3. Шаблон для построения трехмерного графика

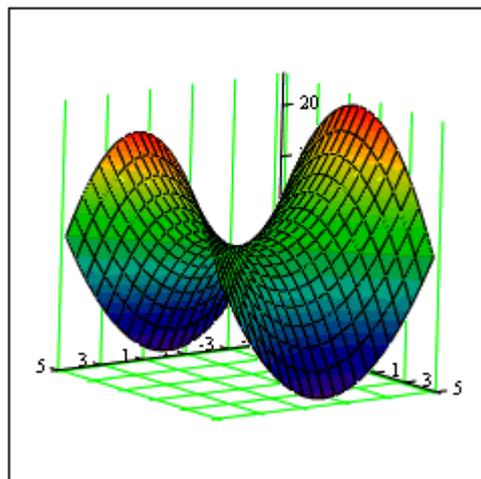
Замечание. При построении трехмерного графика можно воспользоваться услугами мастера трехмерной графики, сделав выбор пункта **3D Plot Wizard (Мастер 3D графиков)** из меню команды **Insert → Graph (Вставка → График)**.

Рассмотрим несколько примеров построения трехмерных графиков.

Для построения **графика функции $z=f(x, y)$ в виде поверхности в декартовой системе координат** можно воспользоваться двумя способами:

1. *Быстрое построение графика.* Необходимо определить функцию $f(x, y)$ и на панели **Graph** выбрать **Surface Plot (График поверхности)**. В появившейся графической области под осями на месте шаблона для ввода надо указать имя (без аргументов) функции. Независимые переменные x и y принимают значения из промежутка $[-5; 5]$ (рис. 8).

$$g(x, y) := x^2 - y^2$$



8

Рис. 8

Информационные технологии в менеджменте

При необходимости этот промежуток может быть уменьшен или увеличен. Для этого необходимо дважды щелкнуть левой кнопкой мыши по выделенному графику и в появившемся окне **3D Plot Format (Формат 3D графика)** на вкладке **QuickPlot Data** можно установить другие параметры изменения независимых переменных x и y (рис. 9).

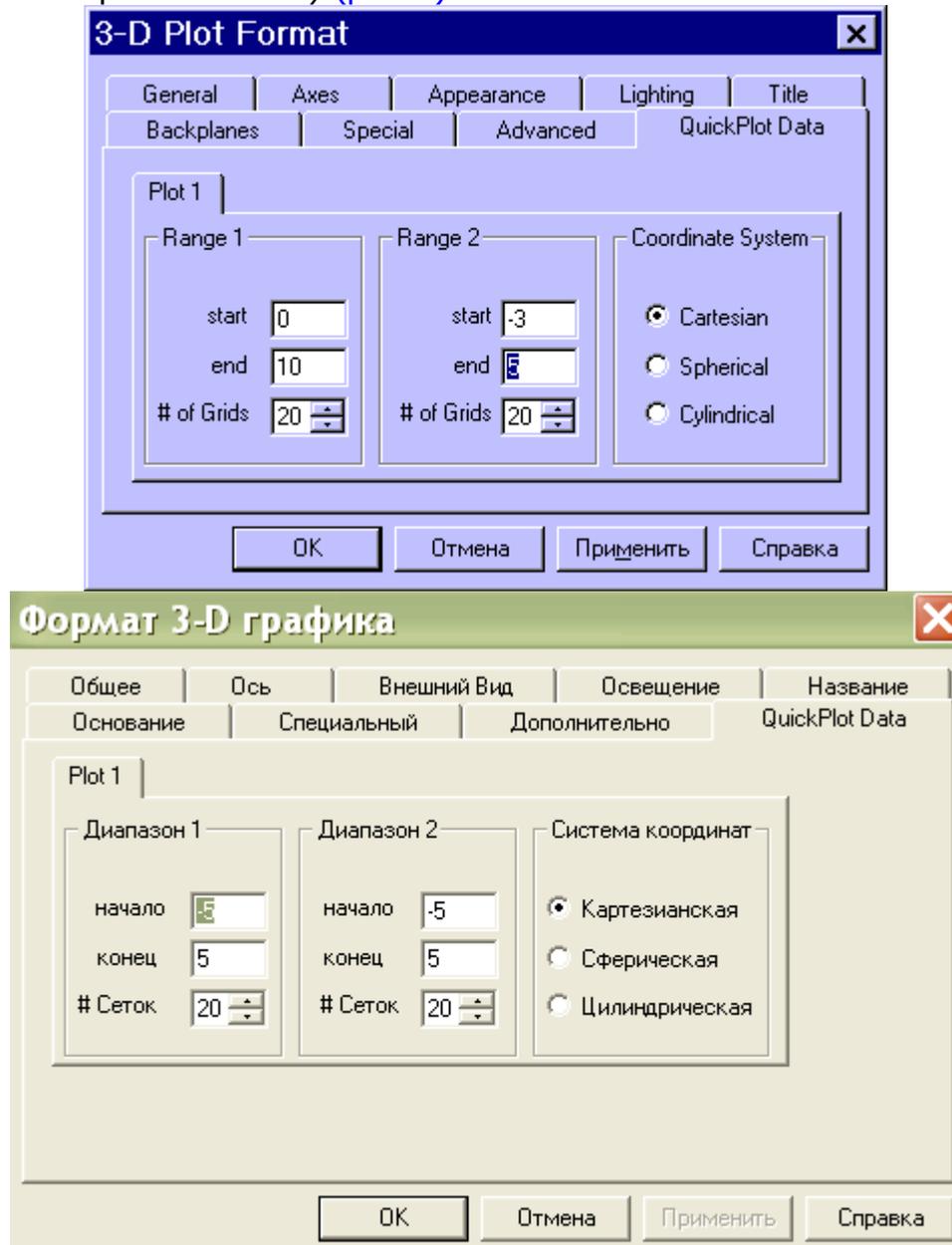


Рис. 9.

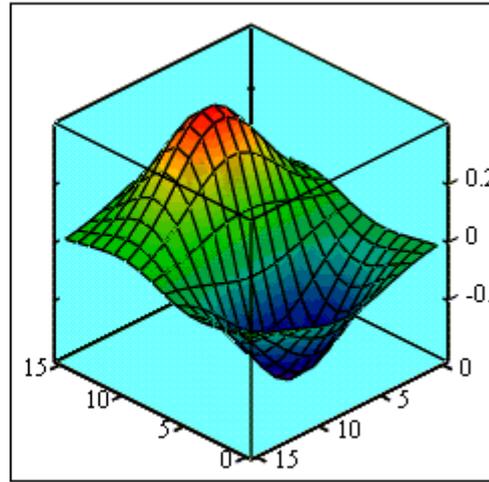
2. *С помощью заданной матрицы.* Для построения графика поверхности в определенной области изменения независимых переменных или с конкретным шагом их изменения необходимо сначала задать узловые точки x_i и y_j , в которых будут определяться значения функции (индексы i и j будут указывать координаты узла на сетке). После (а можно и до) этого надо определить функцию $f(x, y)$, график которой хотите построить. После этого необходимо сформировать матрицу значений функции в виде: $A_{i,j} = f(x_i, y_j)$ (рис. 10).

Теперь после выполнения команды **Graph** → **Surface Plot** в появившейся графической области достаточно ввести имя матрицы (без индексов).

```

N := 15
i := 0..N      j := 0..N
x_min := -1.5  x_max := 1.5
y_min := -1.5  y_max := 1.5

x_i := x_min + (i/N) * (x_max - x_min)
y_j := y_min + (j/N) * (y_max - y_min)
f(x,y) := sin(x) * exp(-x^2 - y^2)
A_{i,j} := f(x_i, y_j)
    
```



A

Рис. 10.

Также для построения графика поверхности в определенной области изменения независимых переменных или с конкретным шагом их изменения можно использовать функцию:

M:=CreateMesh(f,xn,xk,yn,yk,s1,s2),

где f — функция, определяющая поверхность; x_n, x_k, y_n, y_k — начальные и конечные значения независимых переменных x и y ; s_1, s_2 — размерность сетки.

После выполнения команды **Graph** → **Surface Plot** в появившейся графической области вводится имя переменной (в данном случае M).

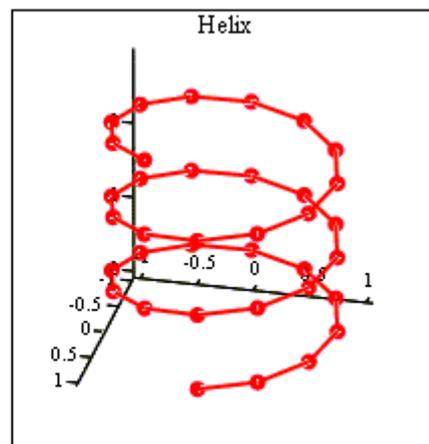
Построение графика кривой в пространстве. Трехмерные точечные графики можно использовать для построения изображения пространственных кривых. Пространственные кривые задаются, как правило, в виде $(x(t), y(t), z(t))$, где t представляет собой непрерывный действительный параметр (рис. 11).

Поскольку при построении трехмерной точечной диаграммы Mathcad позволяет отображать на графике только отдельные точки и соединяющие их линии, необходимо сначала определить три вектора координат — x_i, y_i, z_i . Пространственная кривая создается командой **Graph** → **3D Scatter Plot**.

```

N := 36      Число значений параметров
i := 0..N - 1  Индексная переменная i

x_i := cos(i/N * 6 * pi)  Задание векторов
                          координат x, y и z,
                          которые являются
y_i := sin(i/N * 6 * pi)  функциями от
                          индекса i. Данные
                          векторы координат
z_i := i/N * 3            описывают винтовую
                          линию.
    
```



(x,y,z)

Рис. 11.

6. Форматирование трехмерных графиков

Если вас не устраивает внешний вид созданного трехмерного графика, вы можете изменить его, выполнив команду **Format**→**Graph**→**3D Plot** или выполнив двойной щелчок мышкой на графической области. В результате на экране появится диалоговое окно **3D Plot Format**, позволяющее изменять параметры отображения графика.

Диалоговое окно **3D Plot Format** содержит несколько вкладок (рис. 12).

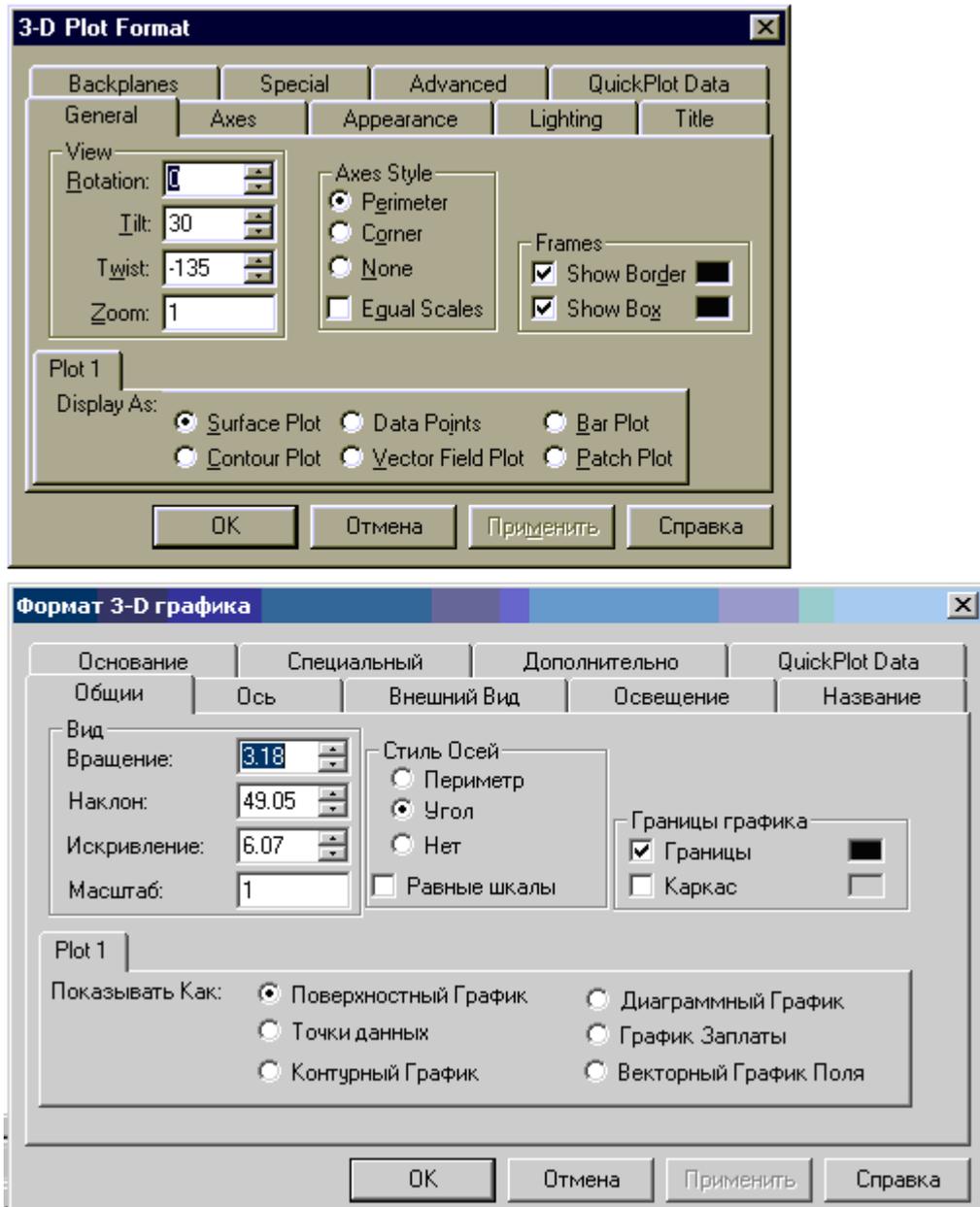


Рис. 12.

На вкладке **General (Общее)**:

— в области **View (Вид)** можно задать направление взгляда наблюдателя на трехмерный график. Значение в поле **Rotation** определяет угол поворота вокруг оси Z в плоскости $X-Y$. Значение в поле **Tilt** задает угол наклона линии взгляда к плоскости $X-Y$. Поле **Zoom** позволяет увеличить (уменьшить) графическое изображение в число раз, равное цифре, указанной в поле;

— в области **Axes Style (Стиль осей)** задать вид осей, выбрав селекторную кнопку **Perimetr (Периметр)** или **Corner (Угол)**. В первом случае

Информационные технологии в менеджменте

оси всегда находятся на переднем плане. При выборе кнопки **Corner** точка пересечения осей Ox и Oy задается элементом $A_{0,0}$ матрицы A ;

— в области **Frames (Границы графика)** опция **Show box (Каркас)** предназначена для отображения вокруг графика куба с прозрачными гранями, а опция **Show border (Границы)** позволяет заключить график в прямоугольную рамку;

— в области **Plot 1 (Plot 2...) Display as: (График 1 Показывать как:)** имеются селекторные кнопки для представления графика в других видах (контурный, точечный, векторное поле и др.);

Элементы вкладки **Axes (Ось)** позволяют изменять внешний вид осей координат (рис. 13).

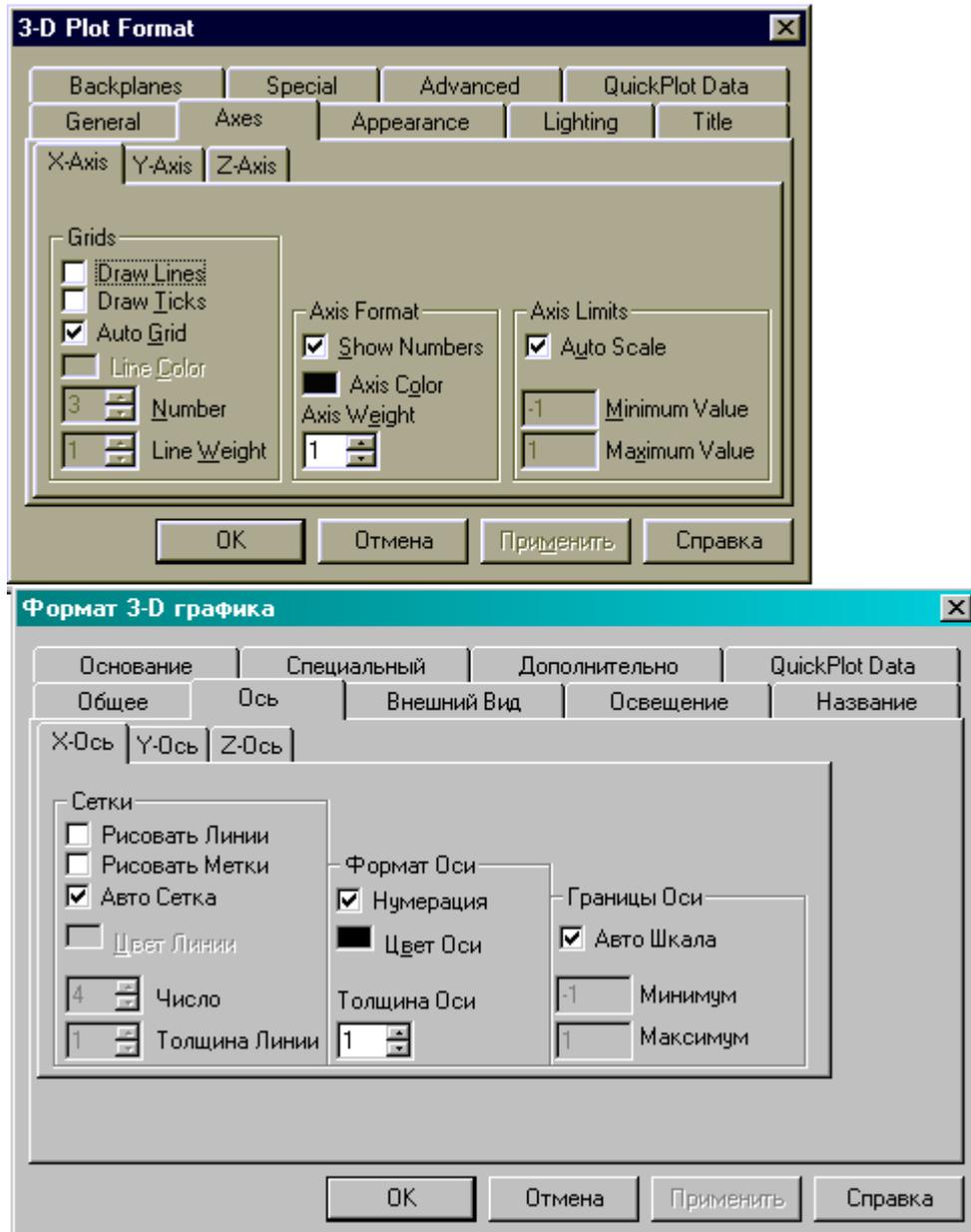


Рис. 13.

Посредством опций области **Grids (Сетки)** можно отобразить на графике линии, описываемые уравнениями $x, y, z = \text{const}$.

Если установлены опции **Show Numbers (Нумерация)**, отображаются метки на осях и подписи к ним.

Информационные технологии в менеджменте

При этом рядом с осями Ox и Oy указываются не значения узловых точек x_i, y_j , а значения индексов i и j , в то время как ось Oz размечается в соответствии с промежутком, которому принадлежат элементы матрицы значений $A_{i,j}$.

Если установлена опция **Auto Grid (Авто сетка)**, программа самостоятельно задает расстояние между соседними отметками на осях. Вы можете сами указать число линий сетки, если отключите указанную опцию.

Если установлена опция **Auto Scale (Авто шкала)**, то Mathcad сам определяет границы построения графика и масштабы по осям. Можно отключить данную опцию и для каждой оси самостоятельно задать пределы изменения переменных в полях **Minimum Value (Минимум)** и **Maximum Value (Максимум)**.

Вкладка **Appearance (Внешний вид)** позволяет изменять для каждого графика вид и цвет заливки поверхности (область **Fill Options**); вид, цвет и толщину дополнительных линий на графике (область **Line Options**); наносить на график точки данных (опция **Draw Points** области **Point Options**), менять их вид, размер и цвет.

Вкладка **Lighting (Освещение)** при включении опции **Enable Lighting (Включить освещение)** позволяет выбрать цветовую схему для освещения, установить несколько источников света, выбрав для них цвет освещения и определив его направление.

Вкладка **Backplanes (Основание)** позволяет изменить внешний вид плоскостей, ограничивающих область построения: цвет, нанесение сетки, определение ее цвета и толщины, прорисовка границ плоскостей.

На вкладке **Special (Специальный)** можно изменять параметры построения, специфичные для различных типов графиков.

Вкладка **Advanced (Дополнительно)** позволяет установить параметры печати и изменить цветовую схему для окрашивания поверхности графика, а также указать направление смены окраски (вдоль оси Ox , Oy или Oz). Включение опции **Enable Fog (Наличие тумана)** делает график нечетким, слегка размытым (полупрозрачным). При включении опции **Perspective (Перспектива)** появляется возможность указать в соответствующем поле расстояние до наблюдателя.

Создание анимации

Во многих случаях самый зрелищный способ представления результатов математических расчетов — это анимация. Mathcad позволяет создавать анимационные ролики и сохранять их в видеофайлах.

Основной принцип анимации в Mathcad — покадровая анимация. Ролик анимации — это просто последовательность кадров, представляющих собой некоторый участок документа, который выделяется пользователем. Расчеты производятся обособленно для каждого кадра, причем формулы и графики, которые в нем содержатся, должны быть функцией от номера кадра. Номер кадра задается системной переменной **FRAME**. По умолчанию, если не включен режим подготовки анимации, **FRAME=0**.

Последовательность действий для создания ролика анимации:

— ввести в документ необходимые выражения и графики, в которых участвует переменная номера кадра **FRAME**;

— подготовить часть документа, которая будет анимацией, таким образом, чтобы она находилась в поле зрения на экране. Выбрать панель инструментов **Tools / Animation (Вид / Анимация)**;

Информационные технологии в менеджменте

— в диалоговом окне **Animate (Анимация)** задать номер первого кадра в поле **From (От)**, номер последнего кадра в поле **To (До)** и скорость анимации в поле **At (Скорость)** в кадрах в секунду (рис. 1);

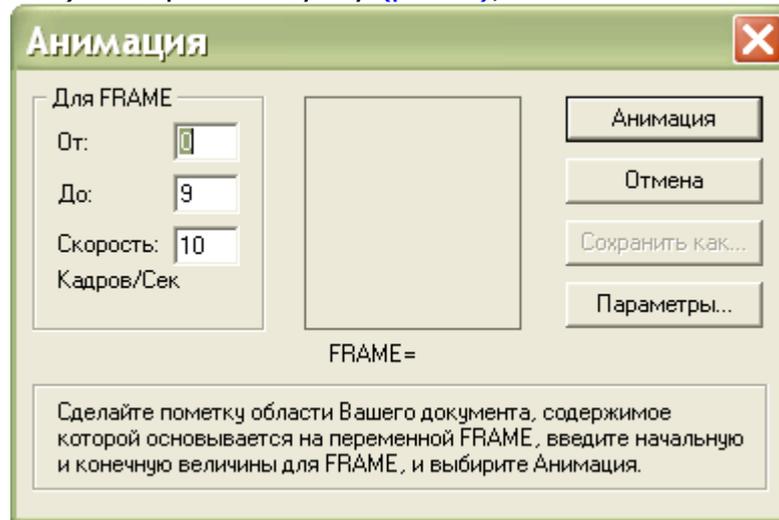


Рис. 1.

— выделить протаскиванием указателя мыши при нажатой левой кнопке мыши область в документе, которая станет роликом анимации;

— в диалоговом окне **Animate (Анимация)** нажать кнопку **Animate (Анимация)**. После этого в окошке диалогового окна **Animate (Анимация)** будут появляться результаты расчетов выделенной области, сопровождающиеся выводом текущего значения переменной **FRAME**. По окончании этого процесса на экране появится окно проигрывателя анимации (рис. 2);

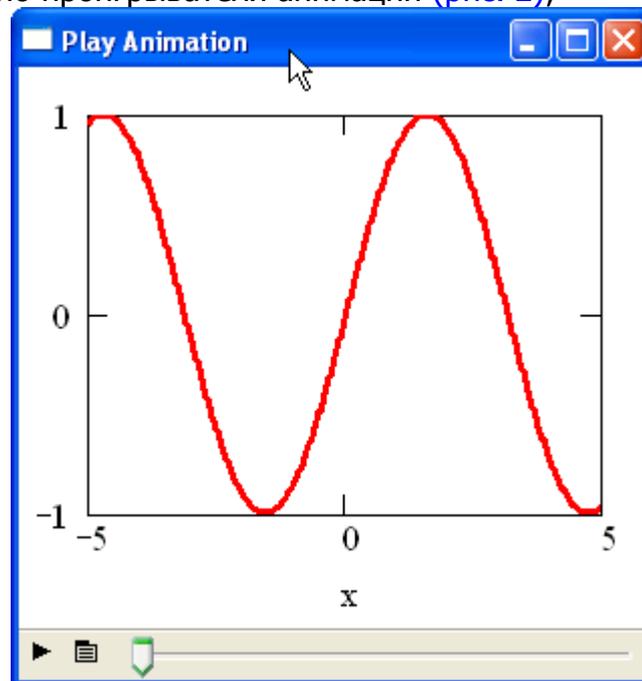


Рис. 2.

— запустить просмотр анимации в проигрывателе нажатием кнопки воспроизведения в левом нижнем углу окна проигрывателя;

— в случае если вид анимации Вас устраивает, сохраните ее в виде видеофайла, нажав кнопку **Save As (Сохранить как)** в диалоговом окне **Animate (Анимация)**. В появившемся диалоговом окне **Save Animation (Сохранить анимацию)** обычным для Windows способом укажите имя файла и его расположение на диске.

Информационные технологии в менеджменте

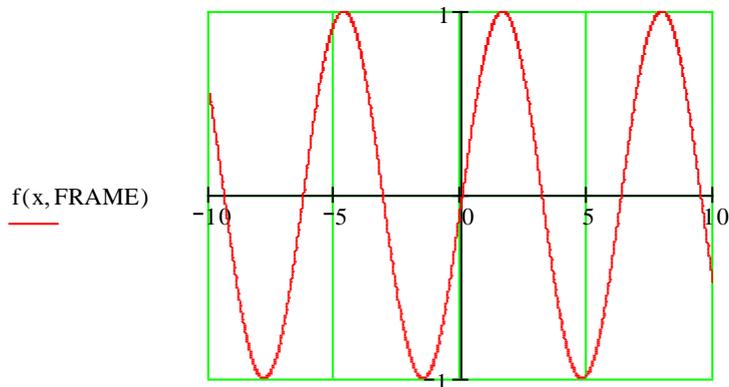
Сохраненный видеофайл можно использовать за пределами Mathcad. Таким образом, запуская видеофайлы, можно устроить красочную презентацию результатов работы.

При создании файлов анимации допускается выбирать программу видеосжатия (кодек) и качество компрессии. Делается это с помощью кнопки **Options (Параметры)** в диалоговом окне **Animate (Анимация)**.

Пример 1. Создание ролика анимации, демонстрирующего перемещение гармонической бегущей волны. При этом каждый момент времени будет задаваться переменной **FRAME**.

Подготовка сводится к определению функции $f(x,t) := \sin(x-t)$ и созданию ее графика $f(x, \text{FRAME})$.

$$f(x, t) := \sin(x - t)$$



Задания

Задание 1. Определите функцию $f(t, a)$ вида:

№	Выражение	№	Выражение
1.	$f(t, a) = ax / \ln(\omega t)$	16.	$f(t, a) = \sqrt{ xt^2 - 3 } / xa$
2.	$f(t, a) = xax^3 - 2\sin(\omega t)$	17.	$f(t, a) = xt^2 \sqrt{ a^2 - 1 }$
3.	$f(t, a) = (ax^2 - 1) \cdot t / (x\omega^2 + 1)$	18.	$f(t, a) = a / (\sin(\omega t) + \cos(\omega t))$
4.	$f(t, a) = 3at^4 - 4at^3 + 12at^2 + 1$	19.	$f(t, a) = a\omega / 2 + \arctg(\omega t)$
5.	$f(t, a) = xa - 2\ln(\omega t)$	20.	$f(t, a) = ae^{-t^2/2}$
6.	$f(t, a) = e^{-at} \cos(\omega t)$	21.	$f(t, a) = e^{-t^2} / a\omega$
7.	$f(t, a) = (1 - xt + xt^2) / (a\omega - x^2)$	22.	$f(t, a) = - x ^3 e^{-at^2/2}$
8.	$f(t, a) = -\sqrt{2a\omega t - xt^2}$	23.	$f(t, a) = xa^2 \ln(\omega t)$
9.	$f(t, a) = (x - 2) \cdot t^5 (2a\omega + 1) \cdot t^4$	24.	$f(t, a) = x\omega \ln^2(at)$
10.	$f(t, a) = x^{-\omega t} + \sin(ax)$	25.	$f(t, a) = x \arctg(at)$
11.	$f(t, a) = e^{-1/\omega t} + \cos(at)$	26.	$f(t, a) = \sin(x\omega t) + \cos(a\omega t)$
12.	$f(t, a) = xa\sqrt{1 - xt^2}$	27.	$f(t, a) = -a \ln(t) / x^2$
13.	$f(t, a) = ((e^x + e^{-x}) / 2) \cdot t^3 + a\omega$	28.	$f(t, a) = x^2 \ln^2(at)$
14.	$f(t, a) = -at / (x\omega^3 + 2)$	29.	$f(t, a) = a^2 \sin(t) / x$
15.	$f(t, a) = x\omega^2 / \sqrt[3]{at^3 - 4}$	30.	$f(t, a) = -ax^{1/\omega t}$

предварительно определив переменные ω , x , a и ранжированную переменную t . Покажите таблицы значений ранжированной переменной и функции.

Постройте графики функции $f(t, a)$ для двух разных значений аргумента a . Как изменить масштаб графика? Как изменить цвет и тип линий на графике?

Задание 2. Постройте график параметрически заданных кривых:

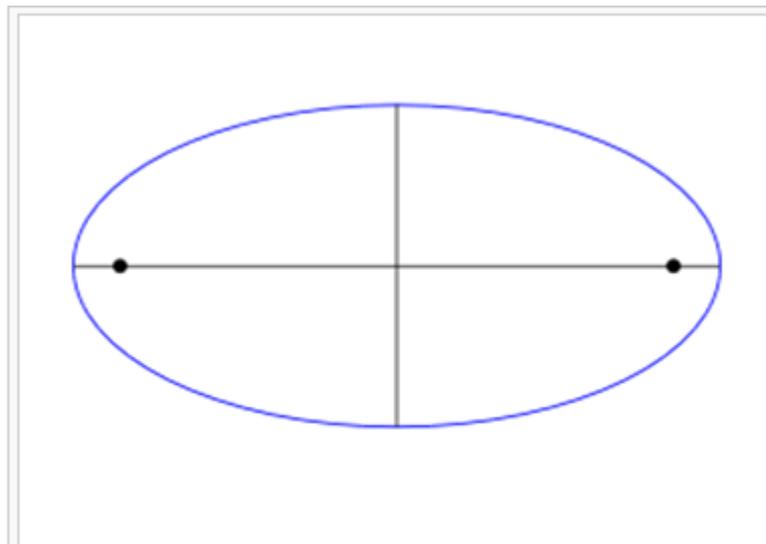
1., 8., 15., 22., 29. Эллипс:

$$x = a \cdot \cos(\alpha)$$

$$y = b \cdot \sin(\alpha) \quad \text{при } 0 \leq \alpha \leq 2\pi.$$

Эллипс (др.-греч. ἔλλειψις — опущение, недостаток, в смысле недостатка эксцентриситета, геометрическое место точек M Евклидовой плоскости, для которых сумма расстояний до двух данных точек F_1 и F_2 (называемых фокусами) постоянна и больше расстояния между фокусами.

Информационные технологии в менеджменте



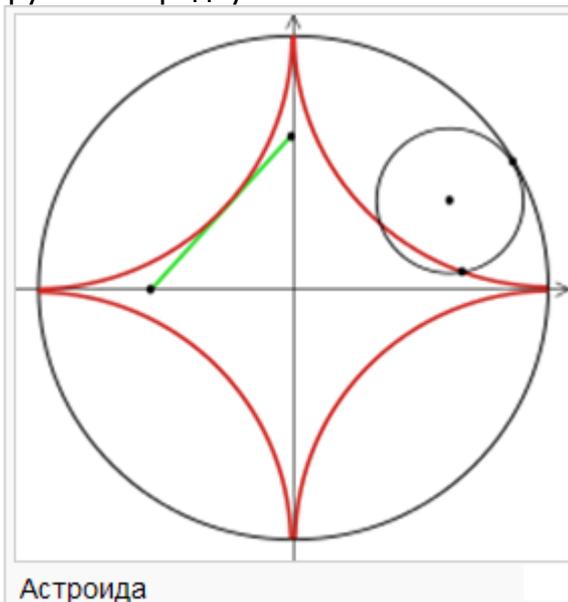
Эллипс, его фокусы и главные оси

2., 9., 16., 23., 30. Астроида:

$$x = R \cos^3 t$$

$$y = R \sin^3 t$$

Астроида (от греч. аотров — звезда и εἶδος — вид, то есть звездообразная) — плоская кривая, описываемая точкой окружности радиуса r , катящейся по внутренней стороне окружности радиуса $R = 4r$.



Астроида

3., 10., 17., 24. Циссоида Диокла:

$$x = \frac{2a}{1 + u^2}, \quad \text{где}$$

$$y = \frac{2a}{u(1 + u^2)}, \quad u = \operatorname{tg} \varphi.$$

Циссоида Диокла — плоская алгебраическая кривая третьего порядка.

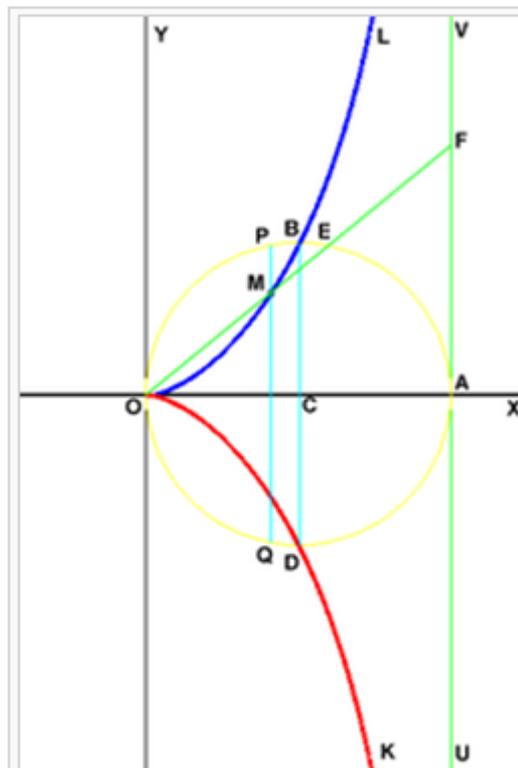


Рис. 1. Построение циссоиды.
Синяя и красная линии — ветви циссоиды.

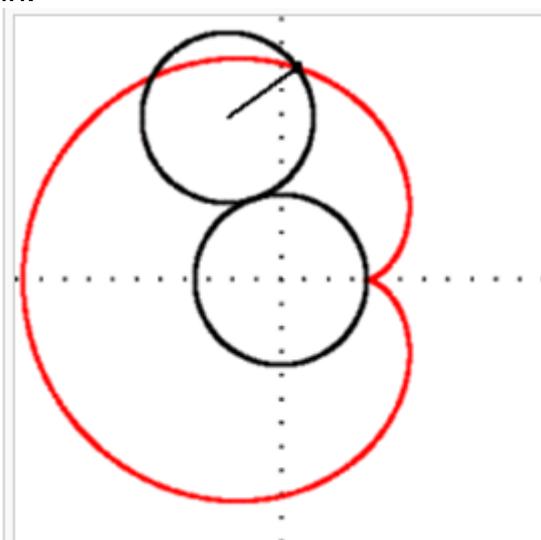
4., 11., 18., 25. Кардиоида:

$$x = 2a \cdot \cos(\alpha) - a \cdot \cos(2\alpha)$$

$$y = 2a \cdot \sin(\alpha) - a \cdot \sin(2\alpha)$$

Кардиоида (греч. καρδιά — сердце, греч. εἶδος — вид) — плоская линия, которая описывается фиксированной точкой окружности, катящейся по неподвижной окружности с таким же радиусом. Получила свое название из-за схожести своих очертаний со стилизованным изображением сердца.

Кардиоида является частным случаем улитки Паскаля, эпициклоиды и синусоидальной спирали.



Построение кардиоиды

5., 12., 19., 26. Строфоида:

$$x = \frac{a \cdot (t^2 - 1)}{t^2 + 1}$$

$$y = \frac{a \cdot t \cdot (t^2 - 1)}{t^2 + 1}, \text{ где } t = \text{tg}(\alpha).$$

Строфоида (от греч. *отроφή* — поворот) — алгебраическая кривая 3-го порядка.

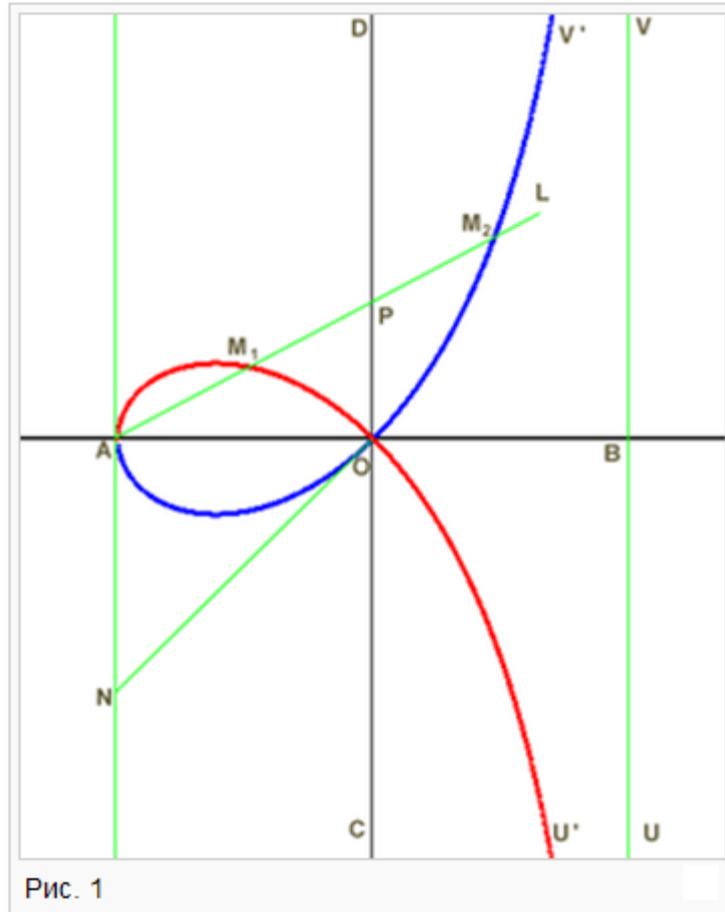


Рис. 1

6., 13., 20., 27 Лемниската Бернулли:

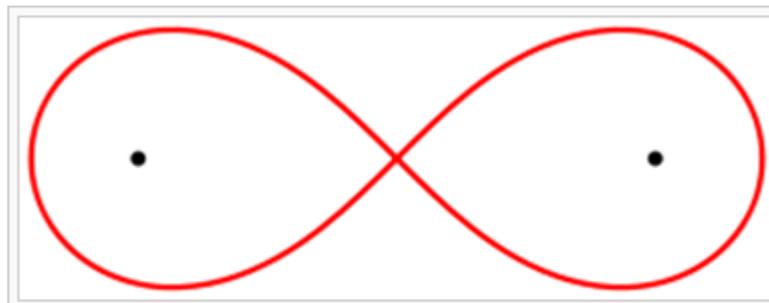
$$x = \frac{a \cdot \sqrt{2} \cdot (t + t^3)}{1 + t^4}$$

$$y = \frac{a \cdot \sqrt{2} \cdot (t - t^3)}{1 + t^4}, \text{ где } t = \pm \sqrt{\text{tg}\left(\frac{\pi}{4} - \alpha\right)}.$$

Лемниската Бернулли — плоская алгебраическая кривая. Определяется как геометрическое место точек, произведение расстояний от которых до двух заданных точек (фокусов) постоянно и равно квадрату половины расстояния между фокусами.

Лемниската по форме напоминает восьмерку или символ бесконечности. Точка, в которой лемниската пересекает саму себя, называется *узловой* или *двойной* точкой.

Информационные технологии в менеджменте

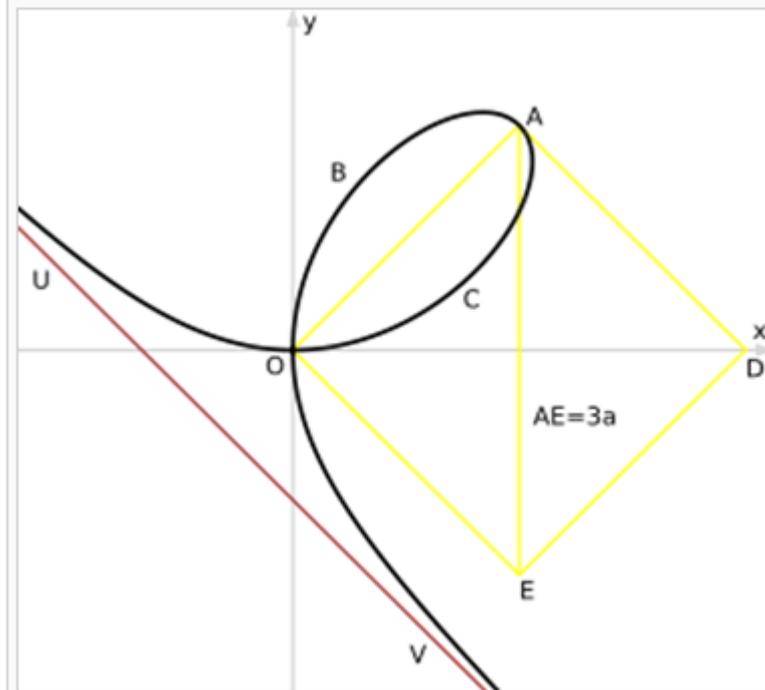


Лемниската и её фокусы

7., 14., 21., 28. Декартов лист:

$$\begin{cases} x = \frac{3at}{1+t^3} \\ y = \frac{3at^2}{1+t^3} \end{cases}, \text{ где } t = \operatorname{tg} \varphi.$$

Декартов лист — плоская алгебраическая кривая третьего порядка, удовлетворяющая уравнению в прямоугольной системе $x^3 + y^3 = 3axy$. Параметр $3a$ определяется как диагональ квадрата, сторона которого равна наибольшей хорде петли.



Декартов лист

Задание 3. Постройте график в полярной системе координат:

1., 20. Стрфоида:

$$R = \frac{-a \cdot \cos(2\alpha)}{\cos(\alpha)}$$

2., 21. Лемниската Бернулли:

$$\rho^2 = 2c^2 \cos 2\varphi$$

3., 22. Декартов лист:

$$R = \frac{3a \cdot \cos(\alpha) \cdot \sin(\alpha)}{\cos^3(\alpha) + \sin^3(\alpha)}$$

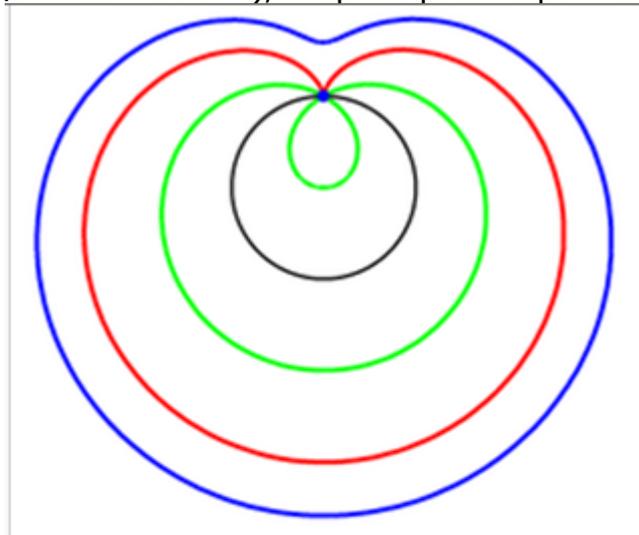
4., 23. Циссоида Диокла:

$$\rho = \frac{2a \sin^2 \varphi}{\cos \varphi}$$

5., 24. Улитка Паскаля:

$$\rho = \ell - a \sin \phi$$

Улитка Паскаля — плоская алгебраическая кривая 4-го порядка, частный случай Декартова овала, она также является эпитрохойдой. Названа по имени Этьена Паскаля (отца Блеза Паскаля), впервые рассмотревшего ее.



Три улитки паскаля, конхоиды чёрной окружности: зелёная $a > \ell$, красная (кардиоида) $a = \ell$ и синяя $a < \ell$

6., 25. Спираль Архимеда (Архимедова спираль):

$R = k \cdot \alpha$, где k — смещение точки M по лучу, при повороте на угол равный одному радиану.

Архимедова спираль — спираль, плоская кривая, траектория точки M , которая равномерно движется вдоль луча OV с началом в O , в то время как сам луч OV равномерно вращается вокруг O .

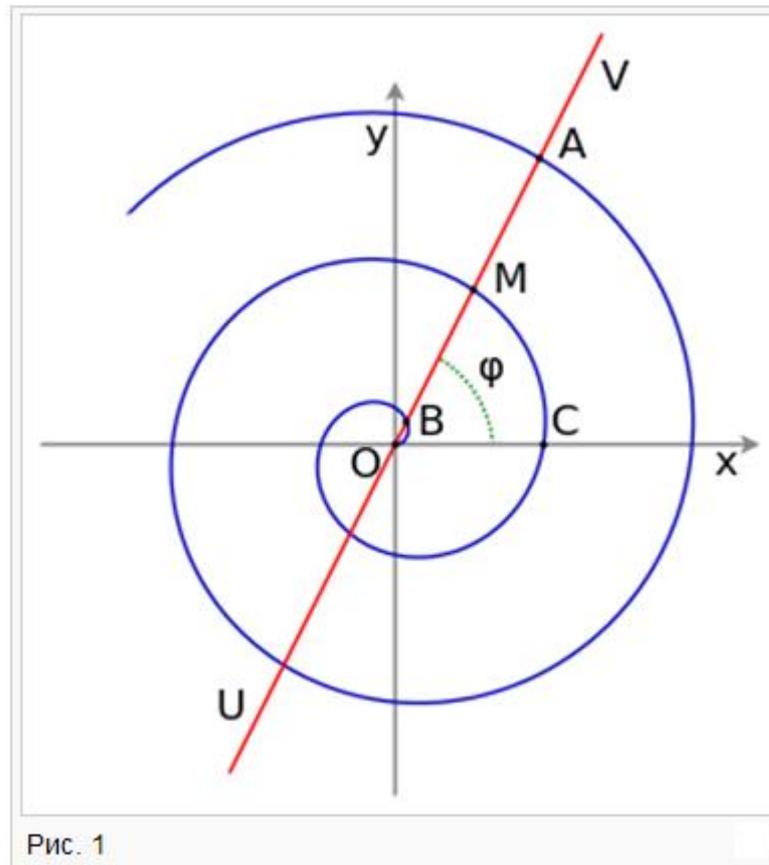


Рис. 1

7., 26. Логарифмическая спираль:

$R = a \cdot e^{k \cdot \alpha}$, где α — угол отклонения точки от нуля; R — радиус-вектор точки; a — коэффициент, отвечающий за расстояние между витками; b — коэффициент, отвечающий за густоту витков.

Логарифмическая спираль или **изогональная спираль** — особый вид спирали, часто встречающийся в природе. Логарифмическая спираль была впервые описана Декартом и позже интенсивно исследована Бернулли, который называл ее *Spira mirabilis* — «удивительная спираль».

8., 27. $\rho = a \cos \phi$

9., 28. $\rho = a \sqrt{\sin \phi}$

10., 29. $\rho = a(4 + \cos \phi)$

11., 30. $\rho = a \sqrt{\sin 2\phi}$

12. $\rho = a \cos 2\phi$

13. $\rho = a(1 + \cos \phi)$

14. $\rho = a \cos 3\phi$

15. $\rho = a(1 - \sin \phi)$

16. $\rho = a \sin^2 \phi$

17. $\rho = a \sin 2\phi$

18. $\rho = a(1 + \sin \phi)$

19. $\rho = a(2 - \cos \phi)$

Задание 4. Постройте график пересечения поверхностей:

Информационные технологии в менеджменте

$$f1(x, y) = \frac{(x + y)^2}{10}, \quad f2(x, y) = 5 \cdot \cos\left(\frac{x - y}{3}\right).$$

Выполните оформление графика (заливка поверхности, освещение, туман и т. д.).

Задание 5. Постройте график поверхности $z(x, y)$ в определенной области изменения независимых переменных x, y :

1. $z = \sqrt{\ln(x + y)}.$

2. $z = \ln \frac{x^2}{x + y}.$

3. $z = \ln \frac{\cos x}{y}.$

4. $z = \ln \frac{x - 3}{y - 5}.$

5. $z = \ln(y - \sin x).$

6. $z = \log_y(x^2 + y^2 - 9).$

7. $z = \frac{1}{x} \arcsin \frac{x + y}{y}.$

8. $z = \sqrt{\ln xy}.$

9. $z = \frac{\ln(x^2 y)}{\sqrt{y - x}}.$

10. $z = \ln(-x - y).$

11. $z = \ln x + \ln \sin y.$

12. $z = \frac{\ln x}{\sqrt{4 - x^2 - y^2}}.$

13. $z = \frac{1}{x} + \sqrt{(x^2 + y^2 - 1)(4 - x^2 - y^2)}.$

14. $z = \sqrt{1 - x^2} + \sqrt{y^2 - 1}.$

15. $z = \frac{1}{x - y} - \frac{1}{\sqrt{x^2 + y^2 - 9}}.$

16. $z = \frac{1}{|x|} \sqrt{y^2 - x}.$

$$17. z = (x + \sqrt{y}) \cdot \ln(y^2 - x^2).$$

$$18. z = \sqrt{1 - (x^2 + y^2)^2} \cdot \ln y.$$

$$19. z = \sqrt{1 - x^3} + \ln(y^2 - 1).$$

$$20. z = \frac{\sqrt{\cos x - y}}{\sqrt{y}}.$$

$$21. z = \arcsin(x + y) + \sqrt{9 - x^2 - y^2}.$$

$$22. z = \arccos \frac{x}{y^2} + \arcsin(1 - y^2).$$

$$23. z = \sqrt{x^2 + y^2 - 2x} - \sqrt{4 - x^2 - y^2}.$$

$$24. z = \sqrt{\sin(x^2 + y^2)} + \arcsin(x - y).$$

$$25. z = \arcsin \frac{y}{x^2} + \arccos(1 - x).$$

$$26. z = \sqrt{\sin x \cdot \cos y}.$$

$$27. z = \frac{\sqrt{x}}{\sin y}.$$

$$28. z = \frac{\sqrt{x - \sqrt{y}}}{\sqrt{3 - x^2 - y^2}}.$$

$$29. z = \frac{1}{\sqrt{(y^2 - 1)(x + 1)}}.$$

$$30. z = \sqrt{\ln x \cdot \ln y}.$$

Задание 6. Создайте ролик анимации, демонстрирующий перемещение гармонической бегущей волны следующего вида:

$$f(t, x) := \sum_{i=1}^3 \left(\sin\left(\frac{Nv}{i} \cdot x + t\right) + \cos\left(\frac{Nv}{i} \cdot x + t\right) \right),$$

где Nv — номер варианта.

При этом каждый момент времени будет задаваться переменной **FRAME**.

Лабораторная работа

Mathcad: Задание и обработка массивов

Цель работы: получение навыков работы с векторами и матрицами в пакете Mathcad.

Теоретические сведения

Задание и обработка массивов

Общее название для векторов и матриц в Mathcad — *массивы*.

Задание массивов. Существуют три способа задания массивов:

1. Путем заполнения пустых полей. Для этого необходимо нажать **Ctrl+M** (или выбрать пункт меню **Insert** → **Matrix (Insert** → **Matrix)** или панель инструментов **Math** → **Matrix (Математика** → **Матрица)**) и в появившемся диалоговом окне выбрать размерности матрицы (если одна из них равна 1 — вектора) (рис. 5.2). После нажатия кнопки «ОК» в рабочем листе Mathcad появится изображение матрицы заданной размерности с пустыми полями для числовых значений, которые далее необходимо заполнить вручную.

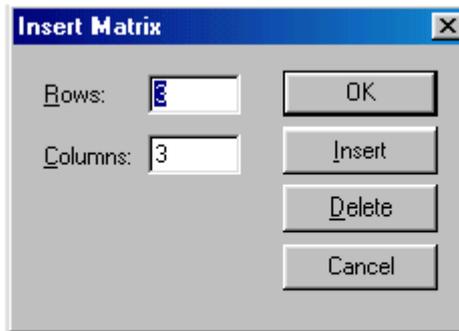


Рис. 5.2. Диалоговое окно **Insert Matrix** (Вставка массива)

Изменение размерностей матриц производится с помощью того же диалогового окна (кнопки **Insert** и **Delete**), в котором указывается количество добавляемых или удаляемых строк и столбцов. Предварительно в матрице курсор устанавливают в позицию добавления или удаления.

В Mathcad вектор-строка задается как двумерный массив, состоящий из одной строки.

После закрытия окна по нажатию кнопки ОК в документ будет вставлен шаблон массива заданной размерности с незаполненными знаками. Ниже в таблице приведены шаблоны, соответствующие указанной в клетках первой строки размерности:

3 x 1	3 x 3	1 x 3
$\begin{pmatrix} \blacksquare \\ \blacksquare \\ \blacksquare \end{pmatrix}$	$\begin{pmatrix} \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \end{pmatrix}$	$(\blacksquare \ \blacksquare \ \blacksquare)$

Шаблон следует заполнить заданными значениями для элементов массива, перемещая курсор по знакам. Заполнение может выполняться в любом порядке, редактирование введенных значений подчиняется тем же правилам, что и редактирование формул. Знакоместа могут быть заполнены не обязательно числами, но и произвольными выражениями, допустимыми в Mathcad, например:

$$\begin{pmatrix} 3.6 & a + b & -12 \\ \sqrt{a} & 0 & \frac{\sin(x)}{4} \\ 1.5 & a^2 + 4 & 17 \end{pmatrix}$$

Массиву можно присвоить имя с помощью оператора присваивания:

$$\text{Имя} := \text{Массив}$$

Например,

$$X := \begin{pmatrix} 12 \\ -4 \\ 5 \end{pmatrix} \quad A := \begin{pmatrix} 1 & -2 & 0 \\ 23 & 2.5 & 1 \\ 12 & -1 & 1 \end{pmatrix} \quad y := (12 \ -2 \ 1)$$

Обращение к элементам массива осуществляется по имени с указанием индекса (для вектора-столбца) или индексов (для матрицы). Элемент матрицы

Информационные технологии в менеджменте

идентифицируется парой индексов, которые разделяются запятыми. Первый индекс определяет номер строки матрицы, второй – номер ее столбца, на пересечении которых находится элемент. Например $A_{i,j}$, где i — номер строки, j — номер столбца. Ввод индексов осуществляется при помощи клавиши «[» или используя знак X_n на панели инструментов **Matrix (Матрица)**.

Понятно, что элементы массива могут быть заданы с помощью индексированной переменной.

Наименьшее значение номеров строки и столбца определяет встроенная переменная **ORIGIN**. Если она имеет значение 0, то нумерация строк и столбцов в массиве начинается с 0. Установить желаемый способ нумерации можно, изменив значение переменной **ORIGIN** с помощью оператора присваивания.

По умолчанию нумерация столбцов и строк начинается с 0.

Часто размерность массива очень велика и он становится очень громоздким для отображения, поэтому Mathcad отображает матрицы и вектора с размерностями больше девяти в виде таблиц с полосами прокрутки.

2. Используя ранжированные переменные (когда имеется явная зависимость между элементами массива и их индексами).

Пример.

$i := 0..10$ $j := 0..10$

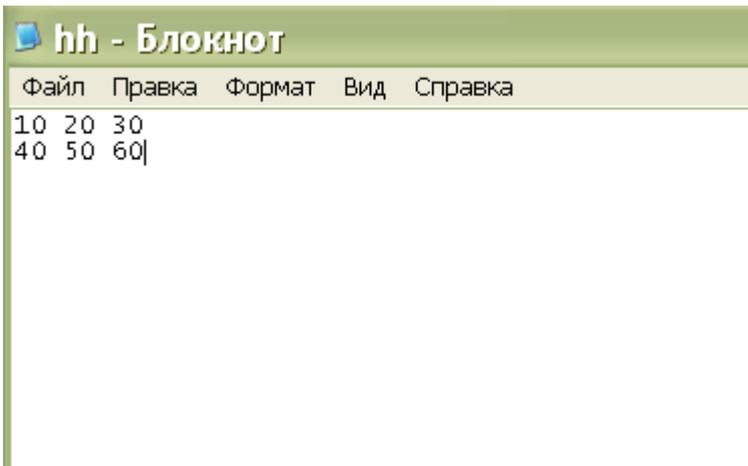
$$A_{i,j} := i^2 - j \cdot \sin(i)$$

A =

	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	1	0.159	-0.683	-1.524	-2.366	-3.207	-4.049	-4.89	-5.732	-6.573	-7.415
2	4	3.091	2.181	1.272	0.363	-0.546	-1.456	-2.365	-3.274	-4.184	-5.093
3	9	8.859	8.718	8.577	8.436	8.294	8.153	8.012	7.871	7.73	7.589
4	16	16.757	17.514	18.27	19.027	19.784	20.541	21.298	22.054	22.811	23.568
5	25	25.959	26.918	27.877	28.836	29.795	30.754	31.712	32.671	33.63	34.589
6	36	36.279	36.559	36.838	37.118	37.397	37.676	37.956	38.235	38.515	38.794
7	49	48.343	47.686	47.029	46.372	45.715	45.058	44.401	43.744	43.087	42.43
8	64	63.011	62.021	61.032	60.043	59.053	58.064	57.074	56.085	55.096	54.106
9	81	80.588	80.176	79.764	79.352	78.939	78.527	78.115	77.703	77.291	76.879
10	100	100.544	101.088	101.632	102.176	102.72	103.264	103.808	104.352	104.896	105.44

3. Считывая элементы массива из файлов данных. Для этого необходимо на панели инструментов **Вставка** → **Компонент** выбрать программу, из которой требуется получить данные (например, Excel, Matlab). Также здесь можно выбрать операцию **Чтение или запись файла**.

Пример. В файле hh.txt, созданном в Блокноте содержатся данные вида (пробелом отделяются элементы, а переход на новую строка определяет новую строку матрицы):



Чтобы иметь возможность использовать эти данные в Mathcad необходимо выбрать: **Вставка** → **Компонент** → **Чтение или запись файла**, затем указать какую операцию выполнять (чтение или запись) и путь к файлу с данными.

Mathcad осуществит вставку фрагмента на лист:

$$s := \text{E:\..\hh.txt}$$

$$s = \begin{pmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \end{pmatrix}$$

Или можно воспользоваться функциями:

$x := \text{READPRN}(\text{"hh.txt"})$

$$x = \begin{pmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \end{pmatrix}$$

— для чтения данных из файла;

$\text{WRITEPRN}(\text{"hh1.txt"}) := x \cdot 1$ — для записи данных в файл.

Обработка массивов. Над массивами в Mathcad могут выполняться бинарные арифметические операции: сложение, вычитание, умножение массивов при условии согласования размерностей операндов, а также умножение массива на скалярное выражение. Перечень унарных операций дан в [таблице 5.1](#).

Таблица 5.1. Унарные операции над массивами

Операция	Запись	Комбинация клавиш	Панель инструментов
Транспонирование матрицы	A^T	Ctrl+1	Matrix
Обращение матрицы	A^{-1}	^	Matrix
Определитель матрицы, длина вектора	$ A $		Matrix
Столбец матрицы	$A^{<n>}$	Ctrl+6	Matrix
Элемент вектора	a_n	[Matrix
Элемент матрицы	$A_{i,j}$	[Matrix

Информационные технологии в менеджменте

Пример 5.16. Даны две матрицы A и B:

$$A = \begin{pmatrix} 2 & -3 & 2 \\ 1 & 2 & 4 \\ -1 & 5 & 2 \end{pmatrix} \quad B = \begin{pmatrix} -1 & 4 & 0 \\ -2 & 3 & 1 \\ 2 & 0 & 4 \end{pmatrix}$$

Выполните над матрицами следующие действия: $3(A+B)(2B^{-1}-A)+A^2$

Решение:

$$A := \begin{pmatrix} 2 & -3 & 2 \\ 1 & 2 & 4 \\ -1 & 5 & 2 \end{pmatrix} \quad B := \begin{pmatrix} -1 & 4 & 0 \\ -2 & 3 & 1 \\ 2 & 0 & 4 \end{pmatrix}$$

$$3 \cdot (A + B) \cdot (2 \cdot B^{-1} - A) + A^2 = \begin{pmatrix} -1.857 & -29.857 & -30.786 \\ 7.714 & -85.286 & -60.429 \\ 3.571 & -85.429 & -71.643 \end{pmatrix}$$

Таблица 1

Функции для оперирования с массивами в пакете Mathcad

<i>Имя функции</i>	<i>Возвращаемое значение</i>
rows(A)	количество строк в массиве A
cols(A)	количество столбцов в массиве A
length(v)	длина вектора v
last(v)	индекс последнего элемента вектора v
max(A)	максимальный элемент массива A
min(A)	минимальный элемент массива A
tr(A)	след матрицы A (сумма диагональных элементов)
rank(A)	ранг матрицы A
norme(A)	евклидова норма матрицы A, то есть корень квадратный из суммы квадратов всех элементов
mean(A)	среднее значение элементов массива A
sort(v)	упорядочивание (сортировка) элементов вектора v по возрастанию
csort(A,n)	сортировка матрицы A по элементам n-го столбца (перестановка строк при упорядочении элементов указанного столбца в возрастающем порядке)
rsort(A,n)	сортировка матрицы A по элементам n-й строки (перестановка столбцов при упорядочении элементов указанной строки в возрастающем порядке)
reverse(A)	перестановка строк массива A в обратном порядке, так что результатом обращения к функциям reverse(sort(v)) будет вектор, элементы которого отсортированы в убывающем порядке
polyroots(v)	поиск корней полинома с коэффициентами, заданными вектором-столбцом v. Коэффициенты перечисляются в направлении увеличения степени неизвестной, начиная с нулевой
augment(A,B)	создание массива, полученного добавлением к массиву A

	справа массива B (склеивание массивов с одинаковым количеством строк). Список аргументов может содержать и более двух массивов
identity(n)	единичная матрица размерности $n \times n$
diag(v)	диагональная матрица с элементами вектора v на диагонали
matrix(m,n,f)	матрица размером $m \times n$, значения элементов матрицы определяются f — функцией $f(i,j)$ от двух переменных (номера строки и номера столбца). Эта функция должна быть предварительно определена пользователем. Пример: <div style="border: 1px dashed black; padding: 5px; margin: 5px 0;"> $f(i,j) := i + j$ $i := 0..2 \quad j := 0..3$ $M := \text{matrix}(3,4,f)$ $R_{i,j} := i + j$ $M = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \end{pmatrix}$ $R = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \end{pmatrix}$ </div>
eigenvals(A)	вектор, элементы которого являются собственными числами матрицы A
eigenvecs(A)	матрица, состоящая из нормализованных собственных векторов матрицы A

Задания

Задание 1. Введите матрицу A , заданную в индивидуальном варианте. Определите число столбцов и строк матрицы. Определите максимальный и минимальный элемент. Найдите детерминант (определитель матрицы), след, ранг, эвклидову норму матрицы, среднее значение элементов массива. Выполните операции транспонирования и нахождения обратной матрицы.

Задание 2. Скопируйте матрицу A , присвоив ей новое имя. Затем поместите курсор в центр матрицы и удалите строку. Затем для тех же условий удалите столбец. Затем добавьте строку/столбец. Каковы правила, по которым Mathcad изменяет размерности матриц?

Задание 3. Задайте вектор b , который представляет собой j -й столбец матрицы A . Определите длину вектора (число элементов), индекс последнего элемента в нем.

Затем прибавьте к нему вектор такой же размерности с произвольно выбранными элементами и покажите результат операции. Прodelайте то же для вычитания и умножения. Примените к вектору b оператор транспонирования.

Задание 4. Выполните сортировку элементов вектора b по возрастанию и по убыванию.

Выполните сортировку элементов матрицы A по элементам i -ой строки и элементам j -го столбца.

Задание 5. Создайте массив G , состоящий из матрицы A и вектора b , используя функцию **augment**.

Задание 6. Задайте матрицу размерности 5×5 , используя ранжированные переменные, при этом зависимость между элементами матрицы и их индексами выбрать самостоятельно.

Задание 7. Введите матрицу A путем считывания ее элементов из файлов данных (для этого необходимо предварительно создать файл данных с помощью Блокнота).

Варианты.

$$1., 17. \quad A = \begin{pmatrix} 1 & 2 & 12 & 9 & 14 \\ 2 & 3 & 7 & 15 & 29 \\ 11 & 6 & 5 & 23 & 16 \\ 27 & 4 & 17 & 18 & 27 \\ 13 & 9 & 25 & 6 & 19 \end{pmatrix} \quad 2., 18. \quad A = \begin{pmatrix} 31 & 18 & 9 & 12 & 24 \\ 29 & 14 & 8 & 21 & 9 \\ 15 & 7 & 6 & 32 & 13 \\ 17 & 19 & 33 & 27 & 25 \\ 11 & 22 & 16 & 35 & 23 \end{pmatrix}$$

$$3., 19. \quad A = \begin{pmatrix} 31 & 12 & 12 & 9 & 14 \\ 32 & 31 & 7 & 15 & 29 \\ 41 & 16 & 5 & 23 & 16 \\ 57 & 14 & 17 & 18 & 27 \\ 23 & 19 & 25 & 6 & 19 \end{pmatrix} \quad 4., 20. \quad A = \begin{pmatrix} 44 & 33 & 19 & 72 & 53 \\ 67 & 14 & 43 & 21 & 9 \\ 15 & 71 & 22 & 32 & 13 \\ 17 & 66 & 45 & 27 & 25 \\ 11 & 15 & 46 & 35 & 23 \end{pmatrix}$$

$$5., 21. \quad A = \begin{pmatrix} 52 & 2 & 12 & 9 & 14 \\ 21 & 3 & 7 & 15 & 29 \\ 11 & 6 & 5 & 23 & 16 \\ 27 & 4 & 17 & 18 & 27 \\ 13 & 9 & 25 & 6 & 19 \end{pmatrix} \quad 6., 22. \quad A = \begin{pmatrix} 31 & 18 & 49 & 12 & 24 \\ 29 & 14 & 58 & 21 & 19 \\ 15 & 17 & 26 & 32 & 13 \\ 17 & 19 & 73 & 27 & 25 \\ 11 & 22 & 56 & 35 & 23 \end{pmatrix}$$

$$7., 23. \quad A = \begin{pmatrix} 20 & 50 & 12 & 9 & 14 \\ 50 & 31 & 7 & 15 & 9 \\ 11 & 6 & 51 & 23 & 16 \\ 27 & 24 & 17 & 18 & 27 \\ 13 & 9 & 25 & 6 & 19 \end{pmatrix} \quad 8., 24. \quad A = \begin{pmatrix} 75 & 15 & 90 & 12 & 24 \\ 29 & 14 & 81 & 21 & 9 \\ 15 & 7 & 63 & 32 & 13 \\ 17 & 19 & 33 & 27 & 25 \\ 11 & 22 & 16 & 35 & 23 \end{pmatrix}$$

$$9., 25. \quad A = \begin{pmatrix} 11 & 2 & 12 & 9 & 14 \\ 2 & 13 & 7 & 15 & 22 \\ 11 & 6 & 51 & 23 & 16 \\ 27 & 4 & 17 & 18 & 27 \\ 23 & 13 & 25 & 6 & 29 \end{pmatrix} \quad 10., 26. \quad A = \begin{pmatrix} 39 & 18 & 9 & 12 & 24 \\ 29 & 14 & 38 & 21 & 44 \\ 15 & 7 & 56 & 32 & 13 \\ 17 & 19 & 33 & 27 & 25 \\ 11 & 22 & 16 & 35 & 23 \end{pmatrix}$$

$$11., 27. \quad A = \begin{pmatrix} 26 & 2 & 12 & 9 & 14 \\ 2 & 3 & 7 & 15 & 29 \\ 11 & 6 & 5 & 23 & 16 \\ 27 & 4 & 17 & 18 & 27 \\ 13 & 82 & 25 & 6 & 11 \end{pmatrix} \quad 12., 28. \quad A = \begin{pmatrix} 67 & 18 & 69 & 12 & 24 \\ 29 & 14 & 8 & 21 & 9 \\ 15 & 7 & 6 & 32 & 13 \\ 17 & 19 & 33 & 27 & 25 \\ 11 & 22 & 16 & 35 & 23 \end{pmatrix}$$

$$13., 29. \quad A = \begin{pmatrix} 63 & 18 & 49 & 12 & 24 \\ 72 & 14 & 48 & 21 & 9 \\ 15 & 67 & 46 & 32 & 13 \\ 17 & 19 & 53 & 27 & 25 \\ 11 & 22 & 56 & 35 & 23 \end{pmatrix} \quad 14., 30. \quad A = \begin{pmatrix} 19 & 12 & 12 & 9 & 14 \\ 12 & 32 & 17 & 15 & 25 \\ 1 & 16 & 5 & 23 & 16 \\ 27 & 14 & 17 & 18 & 25 \\ 35 & 19 & 25 & 61 & 19 \end{pmatrix}$$

$$15. \quad A = \begin{pmatrix} 41 & 18 & 59 & 11 & 14 \\ 39 & 14 & 8 & 21 & 23 \\ 15 & 55 & 6 & 32 & 13 \\ 17 & 19 & 33 & 27 & 25 \\ 11 & 22 & 16 & 35 & 53 \end{pmatrix} \quad 16. \quad A = \begin{pmatrix} 87 & 54 & 12 & 67 & 75 \\ 54 & 48 & 81 & 15 & 21 \\ 11 & 6 & 46 & 23 & 16 \\ 27 & 51 & 17 & 18 & 27 \\ 13 & 80 & 25 & 50 & 22 \end{pmatrix}$$

Лабораторная работа

Mathcad: Символьные вычисления

Цель работы: изучение возможностей символьного процессора пакета Mathcad.

Теоретические сведения

Символьные вычисления

Системы компьютерной математики снабжаются специальным процессором для выполнения аналитических (символьных) вычислений. Его основой является ядро, хранящее всю совокупность формул и формульных преобразований. Чем больше этих формул в ядре, тем надежней работа символьного процессора и тем вероятней, что поставленная задача будет решена.

Информационные технологии в менеджменте

Ядро символьного процессора Mathcad — несколько упрощенный вариант известной системы символьной математики Maple фирмы Waterloo Maple Software, у которой фирма MathSoft (разработчик системы Mathcad) приобрела лицензию на его применение в системе Mathcad. Благодаря этому Mathcad стала системой символьной математики и перешла в разряд универсальных систем компьютерной математики. Правда, доступ к ядру осуществляется лишь частично, так как все-таки основное назначение Mathcad — это производить численные расчеты.

Символьные вычисления осуществляются с помощью команд меню **Symbolics** на панели **Math** (рис. 1). В табл. 8.4 приведены функции основных символьных преобразований.

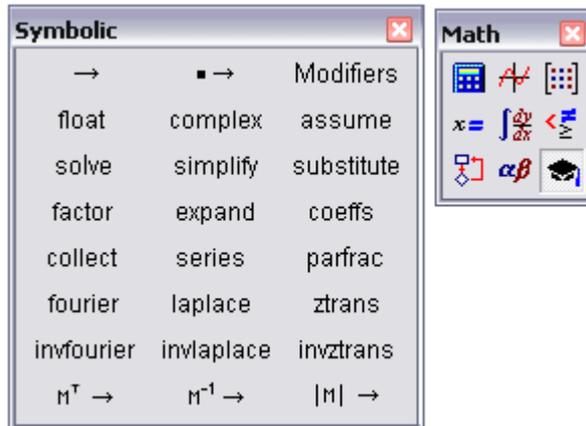


Рис. 1. Математическая панель для символьных вычислений

Информационные технологии в менеджменте

Таблица 8.4. Описание директив панели **Symbolic**

Назначение	Шаблон
Символьное вычисление выражения	■ →
Символьное вычисление выражения с указанием ключевого слова операции	■ ■ →
Преобразование в формат чисел с плавающей точкой	■ float, ■ →
Представление в комплексной форме	■ complex →
Решение уравнений, неравенств и их систем	■ solve, ■ →
Упрощение выражения	■ simplify →
Замена в исходном выражении переменной или подвыражения заданным выражением	■ substitute, ■ = ■ →
Разложение выражения на простые множители	■ factor, ■ →
Разложение выражения по степеням	■ expand, ■ →
Возвращение вектора с коэффициентами полинома	■ coeffs, ■ →
Разложение выражений с приведением подобных	■ collect, ■ →
Разложение в ряд по заданным переменным	■ series, ■, ■ →
Разложение на элементарные дроби	■ convert, parfrac, ■ →
Транспонирование матрицы	$M^T \rightarrow$
Построение обратной матрицы	$M^{-1} \rightarrow$
Вычисление определителя матрицы	■ →

<i>Вид преобразования</i>	<i>Пункты меню Symbolics</i>
Преобразование Лапласа	Laplace
Обратное преобразование Лапласа	Invlaplace
Преобразование Фурье	Fourier
Обратное преобразование Фурье	Invfourier

Для ряда операций следует указать переменную, относительно которой выполняется та или иная символьная операция.

В результате преобразований могут появляться специальные функции — как встроенные в систему (функции Бесселя, гамма-функция, интеграл вероятности и др.), так и ряд функций, дополнительно определенных при загрузке символьного процессора (интегральные синус и косинус, интегралы Френеля, эллиптические интегралы и др.). Последние нельзя использовать при создании математических выражений.

Пример 8.24 (Использование оператора символьного вывода).

$$2 + \sqrt{56} - \frac{4}{5} \rightarrow \frac{6}{5} + 2 \cdot \sqrt{14}$$

$$\sqrt{144} + 4\sqrt{24} + 5\sqrt{96} \rightarrow 12 + 28 \cdot \sqrt{6}$$

$$\sin(2\pi) + \cos\left(12 \frac{\pi}{5}\right) \rightarrow \frac{1}{4} \cdot \sqrt{5} - \frac{1}{4}$$

$$\begin{pmatrix} 1 & 3 \\ a & 2 \end{pmatrix} \cdot \begin{pmatrix} b & a \\ 1 & b \end{pmatrix} \rightarrow \begin{pmatrix} b + 3 & a + 3 \cdot b \\ a \cdot b + 2 & a^2 + 2 \cdot b \end{pmatrix}$$

$$\frac{d^2}{dx^2}(1 + a \cdot \cos(x)) \rightarrow -a \cdot \cos(x)$$

$$\int \cos(x) dx \rightarrow \sin(x)$$

$$\lim_{x \rightarrow 0} (1 + a \cdot x)^{\frac{1}{x}} \rightarrow \exp(a)$$

Символьная операция **Simplify (Упрощение выражения)** — одна из самых важных. Эта операция позволяет упрощать математические выражения, содержащие алгебраические и тригонометрические функции, а также выражения со степенными многочленами (полиномами). Упрощение означает замену более сложных фрагментов выражений на более простые. Приоритет тут отдается простоте функций. К примеру, функция **tan(x)** считается более сложной, чем функции **sin(x)** и **cos(x)**. Поэтому **tan(x)** упрощается так, что получает представление через соотношение этих функций.

Эта команда открывает широкие возможности для упрощения сложных и плохо упорядоченных алгебраических выражений. На [рис. 2](#) даны примеры применения операции **Simplify**.

$$\begin{array}{ll} \sin(x)^2 + \cos(x)^2 & 1 \\ \frac{x^2 - 3 \cdot x - 4}{x - 4} + 2 \cdot x - 5 & 3 \cdot x - 4 \\ \frac{d}{dx} \sin(3 \cdot x) & 3 \cdot \cos(3 \cdot x) \\ \int_a^b e^{-t} dt & -\exp(-b) + \exp(-a) \end{array}$$

Рис. 2.

Информационные технологии в менеджменте

Система Mathcad содержит встроенную функцию для вычисления значений определенных интегралов приближенным численным методом. Ею целесообразно пользоваться, когда нужно просто получить значение определенного интеграла в виде числа.

Однако команда **Simplify** применительно к вычислениям определенных интегралов делает гораздо больше — она ищет аналитическое выражение для интеграла. Более того, она способна делать это и при вычислении кратных интегралов, пределы которых — функции. Наглядный пример этому продемонстрирован на рис. 3.

$$\int_0^1 \int_0^{\sqrt{1+x^2}} \frac{1}{(x^2 + y^2 + 1)} dy dx = \frac{-1}{4} \cdot \ln(\sqrt{2} - 1) \cdot \pi$$

$$\int_0^{\frac{\pi}{2}} \int_0^{\frac{\pi}{4}} \int_0^{4 \cdot \cos(\psi)} \rho^2 \sin(\psi) d\rho d\psi d\theta = 2 \cdot \pi$$

Рис. 3.

Операцию **Simplify** можно использовать и для вычисления сумм и произведений символьных последовательностей. Результат операции, как и следовало ожидать, получается в символьной форме (если она существует) (рис. 4).

$$\sum_i j^3 = \frac{1}{4} \cdot j^4 - \frac{1}{2} \cdot j^3 + \frac{1}{4} \cdot j^2$$

$$\prod_k \frac{k^2}{(k-1)} = \Gamma(k) \cdot (k-1)$$

Рис. 4.

Пример 8.33 (Упрощение выражения).

$$23 + \frac{\frac{1}{2} + \frac{3}{4} - \frac{12}{49}}{45} + \frac{1}{2} - \frac{12}{36} \text{ simplify} \rightarrow \frac{204527}{8820}$$

$$\left(\frac{a-3}{a^2-3a+9} - \frac{6a-18}{a^3+27} \right) \cdot \left[\frac{5(a-3)^2}{2a^3+54} \right]^{-1} \text{ simplify} \rightarrow \frac{2}{5}$$

$$\left(1 + \frac{3x+x^2}{3+x} \right) \cdot \left(\frac{1}{x+1} - \frac{x}{1+2x+x^2} \right) \text{ simplify} \rightarrow \frac{1}{(x+1)}$$

Информационные технологии в менеджменте

$$\frac{d}{dx} (x \cdot \sqrt{1+x^2}) \text{ simplify} \rightarrow \frac{(1+2 \cdot x^2)}{(1+x^2)^{\frac{3}{2}}}$$

$$\frac{d}{dx} \frac{d}{dx} \left(\frac{x}{\sqrt{a^2-x^2}} \right) \text{ simplify} \rightarrow 3 \cdot x \cdot \frac{a^2}{(a^2-x^2)^{\frac{5}{2}}}$$

$$\int \frac{1}{1-\cos(x)} dx \text{ simplify} \rightarrow \frac{-1}{\tan\left(\frac{1}{2} \cdot x\right)}$$

$$\int_0^2 \int_a^b (a \cdot y - x^2) dx dy \text{ simplify} \rightarrow 2 \cdot a \cdot b - \frac{2}{3} \cdot b^3 - 2 \cdot a^2 + \frac{2}{3} \cdot a^3$$

$$\sum_{n=1}^4 2 \frac{n}{n^2+1} \text{ simplify} \rightarrow \frac{244}{85}$$

$$\prod_{n=1}^5 \frac{n}{n^3+2} \text{ simplify} \rightarrow \frac{2}{121539}$$

$$\sum_{i=0}^n a^i \text{ simplify} \rightarrow \frac{[a^{(n+1)} - 1]}{(a-1)}$$

$$\sum_{n=0}^{\infty} a^n \text{ simplify} \rightarrow \frac{-1}{(a-1)}$$

$$\sum_{n=0}^{\infty} \frac{x^n}{2^n \cdot n!} \text{ simplify} \rightarrow \exp\left(\frac{1}{2} \cdot x\right)$$

Действие операции **Expand (Раскрытие скобок)** в известном смысле противоположно действию операции **Simplify**. Подвергаемое преобразованию выражение расширяется с использованием известных (и введенных в символьное ядро) соотношений, например алгебраических разложений многочленов, произведений углов и т. д. Разумеется, расширение происходит только в том случае, когда его результат однозначно возможен. Иначе нельзя считать, что действие этой операции противоположно действию операции **Simplify**. К примеру, операция **Simplify** преобразует сумму квадратов синуса и косинуса в 1, тогда как обратное преобразование многозначно и потому в общем виде невыполнимо.

При преобразовании выражений операция **Expand** старается более простые функции представить через более сложные, свести алгебраические выражения, представленные в сжатом виде, к выражениям в развернутом виде и т. д. Примеры действия операции **Expand** даны на [рис. 5](#).

Информационные технологии в менеджменте

$$\begin{array}{ll} \sin(5 \cdot x) & 16 \cdot \sin(x) \cdot \cos(x)^4 - 12 \cdot \sin(x) \cdot \cos(x)^2 + \sin(x) \\ (x - y)^2 \cdot (x + y) & x^3 - x^2 \cdot y - x \cdot y^2 + y^3 \\ (a + b)^n & (a + b)^n \\ \prod_n \frac{1}{n^2} & \frac{1}{\Gamma(n)^2} \end{array}$$

Рис. 5.

Последний пример показывает, что результатом операции может быть специальная математическая функция, которая считается более сложным выражением, чем порождающее ее выражение. С виду, однако, выражения со специальными математическими функциями обычно выглядят гораздо проще, чем исходные выражения.

Пример 8.28 (Раскрытие скобок). Второе знакоместо шаблона можно не заполнять, удалив его.

$$\begin{array}{l} (a + b)^3 \text{ expand} \rightarrow b^3 + 3 \cdot b^2 \cdot a + 3 \cdot b \cdot a^2 + a^3 \\ \sin(4x) \text{ expand} \rightarrow 8 \cdot \sin(x) \cdot \cos(x)^3 - 4 \cdot \sin(x) \cdot \cos(x) \\ \sin(4x) \text{ expand, x} \rightarrow 8 \cdot \sin(x) \cdot \cos(x)^3 - 4 \cdot \sin(x) \cdot \cos(x) \end{array}$$

Отличия в результатах выполнения преобразования с заполнением второго знакоместа шаблона или без заполнения шаблона можно увидеть на следующих примерах:

$$\begin{array}{l} \left(\frac{1}{x+1} - x^2\right) \cdot (x+1) \text{ expand, x} \rightarrow \frac{1}{(x+1)} \cdot x + \frac{1}{(x+1)} - x^3 - x^2 \\ \left(\frac{1}{x+1} - x^2\right) \cdot (x+1) \text{ expand} \rightarrow -x^3 - x^2 + 1 \\ \left(\frac{1}{x+1} - x^2\right) \cdot \left(1 + x + \frac{1}{x^2}\right) \text{ expand, x} \rightarrow \frac{1}{(x+1)} \cdot x + \frac{1}{(x+1)} + \frac{1}{(x+1) \cdot x^2} - x^3 - x^2 - 1 \\ \left(\frac{1}{x+1} - x^2\right) \cdot \left(1 + x + \frac{1}{x^2}\right) \text{ expand} \rightarrow \frac{(1 - x^6 - 2 \cdot x^5 - x^4)}{(x^3 + x^2)} \end{array}$$

Пример 8.25 (Разложение на элементарные дроби).

$$\frac{x^3 + 1}{x^3 - 5x^2 + 6x} \text{ convert, parfrac, } x \rightarrow 1 + \frac{1}{6 \cdot x} - \frac{9}{2 \cdot (x - 2)} + \frac{28}{3 \cdot (x - 3)}$$

$$\left(\frac{x}{x^2 - 3x + 2}\right)^2 \text{ convert, parfrac, } x \rightarrow \frac{1}{(x - 1)^2} + \frac{4}{(x - 1)} + \frac{4}{(x - 2)^2} - \frac{4}{(x - 2)}$$

$$\frac{a + x}{(x - a) \cdot (x + b)} \text{ convert, parfrac, } a \rightarrow \frac{-1}{(x + b)} + 2 \cdot \frac{x}{(x - a) \cdot (x + b)}$$

$$\frac{a + x}{(x - a) \cdot (x + b)} \text{ convert, parfrac, } x \rightarrow 2 \cdot \frac{a}{(a + b) \cdot (x - a)} - \frac{(-b + a)}{(a + b) \cdot (x + b)}$$

Операция **Coeffs (Определение вектора коэффициентов полинома)** применяется, если заданное выражение — полином (степенной многочлен) или может быть представлено таковым относительно выделенной переменной. Результатом операции является вектор с коэффициентами полинома (рис. 8).

$$a \cdot x^3 + b \cdot x^2 + c \cdot x + d \begin{pmatrix} d \\ c \\ b \\ a \end{pmatrix} \text{ относительно переменной } x$$

$$a \cdot x^3 + b \cdot x^2 + c \cdot x + d \begin{pmatrix} a \cdot x^3 + b \cdot x^2 + d \\ x \end{pmatrix} \text{ относительно переменной } c$$

Рис. 8.

Пример 8.26 (Определение вектора коэффициентов полинома).

$$x^3 - 3 \cdot x^2 + x - 5 \text{ coeffs, } x \rightarrow \begin{pmatrix} -5 \\ 1 \\ -3 \\ 1 \end{pmatrix}$$

$$\sin(x)^3 - 2 \cdot \sin(x) + 5 \text{ coeffs, } \sin(x) \rightarrow \begin{pmatrix} 5 \\ -2 \\ 0 \\ 1 \end{pmatrix}$$

$$x^3 - 2 \cdot a \cdot x^2 + (a + b) \cdot x - 7 \text{ coeffs, } x \rightarrow \begin{pmatrix} -7 \\ b + a \\ -2 \cdot a \\ 1 \end{pmatrix}$$

Операция **Factor (Разложение на множители)** используется для факторизации — разложения выражений или чисел на простые множители. Она способствует выявлению математической сущности выражений; к примеру, наглядно выявляет представление полинома через его действительные корни, а в

Информационные технологии в менеджменте

том случае, когда разложение части полинома содержит комплексно-сопряженные корни, порождающее их выражение представляется квадратичным трехчленом. Примеры действия этой операции даны на рис. 6.

$$\begin{array}{ll} x^3 - 6x^2 + 11x - 6 & (x-1) \cdot (x-2) \cdot (x-3) \\ x^3 + 11x - 6x^2 - 6 & x^3 + 11x - 6 \cdot (x-1) \cdot (x+1) \\ x^3 - 6x^2 + 21x - 52 & (x-4) \cdot (x^2 - 2x + 13) \\ 123456 & (2)^6 \cdot (3) \cdot (643) \end{array}$$

Рис. 6.

В большинстве случаев (но не всегда) операция факторизации ведет к упрощению выражений. Термин «*факторизация*» не является общепризнанным в отечественной математической литературе, но мы его оставляем в связи с созвучностью с англоязычным именем этой операции.

Пример 8.27 (Разложение на множители). При заполнении шаблона следует удалить второе знакоместо.

$$7739550 \text{ factor} \rightarrow 2 \cdot 3^5 \cdot 5^2 \cdot 7^2 \cdot 13$$

$$x^2 - 5x + 6 \text{ factor} \rightarrow (x-2) \cdot (x-3)$$

$$(b^2 \cdot a^4 - 1) \cdot (4 \cos(x)^2 - 9) \text{ factor} \rightarrow (b \cdot a^2 - 1) \cdot (b \cdot a^2 + 1) \cdot (2 \cdot \cos(x) - 3) \cdot (2 \cdot \cos(x) + 3)$$

$$\frac{x+1}{x^2-4} \text{ factor} \rightarrow \frac{(x+1)}{(x-2) \cdot (x+2)}$$

$$\frac{1}{x^2-4} + \frac{x+1}{x^2-6x+8} \text{ factor} \rightarrow \frac{(x^2+4 \cdot x-2)}{(x-2) \cdot (x-4) \cdot (x+2)}$$

Операция **Collect (Приведение подобных слагаемых)** обеспечивает замену указанного выражения выражением, скомплектованным по базису указанной переменной, если такое представление возможно (рис. 7). В противном случае появляется окно с сообщением о невозможности комплектования по указанному базису.

$$\begin{array}{lll} (a+b+c)^2 & a^2 + (2 \cdot b + 2 \cdot c) \cdot a + (b+c)^2 & \text{по } a \\ (a+b+c)^2 & c^2 + (2 \cdot a + 2 \cdot b) \cdot c + (a+b)^2 & \text{по } c \\ (y-x) \cdot (z+y) & (y-x) \cdot z + (y-x) \cdot y & \text{по } z \\ (y-x) \cdot (z+y) & y^2 + (-x+z) \cdot y - xz & \text{по } y \end{array}$$

Рис. 7.

Эта команда особенно удобна, когда заданное выражение есть функция ряда переменных и нужно представить его в виде функции заданной переменной имеющей вид степенного многочлена. При этом другие переменные входят в сомножители указанной переменной, представленной в порядке уменьшения ее степени.

Пример 8.29 (Приведение подобных слагаемых).

$$5x^2 - [(x + 1) \cdot x - 4] \cdot x \text{ collect, } x \rightarrow -x^3 + 4 \cdot x^2 + 4 \cdot x$$

$$x^2 \cdot (y + 1) - (3x^2 \cdot y) - x - 6 \text{ collect, } x \rightarrow (-2 \cdot y + 1) \cdot x^2 - x - 6$$

$$x^2 \cdot (y + 1) - (3x^2 \cdot y) - x - 6 \text{ collect, } y \rightarrow -2 \cdot x^2 \cdot y + x^2 - x - 6$$

$$\sin(x) \cdot x + (\cos(x) \cdot \sin(x) + 3x^2 + 4) \cdot x \text{ collect, } x \rightarrow 3 \cdot x^3 + (\sin(x) + \sin(x) \cdot \cos(x) + 4) \cdot x$$

$$\sin(x) \cdot x + (\cos(x) \cdot \sin(x) + 3x^2 + 4) \cdot x \text{ collect, } \sin(x) \rightarrow (x + \cos(x) \cdot x) \cdot \sin(x) + (3 \cdot x^2 + 4) \cdot x$$

Пример 8.30 (Подстановка в выражение).

$$x^2 - 5x + 6 \text{ substitute, } x = u^2 \rightarrow u^4 - 5 \cdot u^2 + 6$$

$$a + \frac{S}{a - v} \text{ substitute, } a = v \cdot t \rightarrow v \cdot t + \frac{S}{(v \cdot t - v)}$$

$$\sin(x)^2 - (a \cdot \sin(x) + b) \text{ substitute, } \sin(x) = u \rightarrow u^2 - a \cdot u - b$$

$$\frac{1}{x + 1} - x^2 \text{ substitute, } x = 4 \rightarrow \frac{-79}{5}$$

Замечание. В записи команды символ = соответствует логическому оператору сравнения.

В выражении можно выполнить сразу же несколько подстановок вводом нужного количества двойных маркеров ■ = ■, разделенных запятыми:

$$a + b + \frac{a}{b + a} \text{ substitute, } a = 2, b = 4 \rightarrow \frac{19}{3}$$

или одновременным использованием нескольких команд:

$$a + b + \frac{a}{b + a} \left| \begin{array}{l} \text{substitute, } a = 2 \\ \text{substitute, } b = 4 \end{array} \right. \rightarrow \frac{19}{3}$$

✓ Ввод нескольких команд выполняется следующим образом:

- 1) ввести первую команду и охватить ее справа уголком курсора;
- 2) выбрать на панели очередную команду;
- 3) продолжить ввод в случае использования более двух команд одновременно.

Информационные технологии в менеджменте

Пример 8.31 (Задание формата представления).

$$2 + \sqrt{56} - \frac{4}{5} \text{ float},5 \rightarrow 8.6834$$

$$\sqrt{144} + 4\sqrt{24} + 5\sqrt{96} \text{ float},7 \rightarrow 80.58572$$

$$\sin(2\pi) + \cos\left(12 \frac{\pi}{5}\right) \text{ float},4 \rightarrow .3090$$

$$\sqrt{-124} \text{ complex} \rightarrow 2 \cdot i \cdot \sqrt{31}$$

$$23 + \sqrt{-6} \text{ complex} \rightarrow 23 + i \cdot \sqrt{6}$$

Пример 8.32 (Разложение в степенной ряд). Для команды **series** двойные маркеры вводятся: имя (или значение) переменной, относительно которой выполняется разложение в степенной ряд. Последний маркер содержит число, задающее порядок ошибки. Но последний маркер может быть исключен, тогда в этом случае команда **series** выводит ряд с точностью до членов шестого порядка. Если в первом маркере после названия команды указано только имя переменной, то это воспринимается командой как $x=0$:

$$e^x \text{ series},x,5 \rightarrow 1 + x + \frac{1}{2} \cdot x^2 + \frac{1}{6} \cdot x^3 + \frac{1}{24} \cdot x^4$$

$$e^x \text{ series},x=0 \rightarrow 1 + x + \frac{1}{2} \cdot x^2 + \frac{1}{6} \cdot x^3 + \frac{1}{24} \cdot x^4 + \frac{1}{120} \cdot x^5$$

$$\sin(x) \text{ series},x,6 \rightarrow x - \frac{1}{6} \cdot x^3 + \frac{1}{120} \cdot x^5$$

$$e^x \text{ series},x=1,4 \rightarrow \exp(1) + \exp(1) \cdot (x-1) + \frac{1}{2} \cdot \exp(1) \cdot (x-1)^2 + \frac{1}{6} \cdot \exp(1) \cdot (x-1)^3$$

$$(e^x - x \cdot a) \text{ series},x,4 \rightarrow 1 + (1-a) \cdot x + \frac{1}{2} \cdot x^2 + \frac{1}{6} \cdot x^3$$

В первый маркер после названия команды можно вводить через запятые несколько переменных (возможно, с указанием их значений) при разложении в ряд функции многих переменных в окрестности заданной точки:

$$e^{xy} \text{ series},x,y \rightarrow 1 + x \cdot y + \frac{1}{2} \cdot x^2 \cdot y^2$$

$$e^{xy} \text{ series},x=1,y=0,4 \rightarrow 1 + y + (x-1) \cdot y + \frac{1}{2} \cdot y^2 + (x-1) \cdot y^2 + \frac{1}{6} \cdot y^3$$

$$(x^2 \cdot y + y) \text{ series},x=1,y=1,3 \rightarrow -2 + 2 \cdot y + 2 \cdot x + 2 \cdot (x-1) \cdot (y-1) + (x-1)^2$$

Пример 8.34 (Решение уравнений, неравенств и их систем).

$$(x^3 - 5 \cdot x^2 + 10 \cdot x - 6) = 0 \text{ solve, } x \rightarrow \begin{pmatrix} 1 \\ 2 + i \cdot \sqrt{2} \\ 2 - i \cdot \sqrt{2} \end{pmatrix}$$

Если надо найти решение уравнения вида $f(x)=0$, то в записи команды можно не указывать правую часть уравнения:

$$x^3 - 5 \cdot x^2 + 10 \cdot x - 6 \text{ solve, } x \rightarrow \begin{pmatrix} 1 \\ 2 + i \cdot \sqrt{2} \\ 2 - i \cdot \sqrt{2} \end{pmatrix}$$

Указав уравнение и условие $\text{Im}(x)=0$ как элементы матрицы-столбца, можно найти только вещественные корни уравнения:

$$\begin{pmatrix} x^3 - 5 \cdot x^2 + 10 \cdot x - 6 \\ \text{Im}(x) = 0 \end{pmatrix} \text{ solve, } x \rightarrow 1 \quad \left[\begin{pmatrix} x^3 - 5 \cdot x^2 + 10 \cdot x - 6 \\ \text{Im}(x) = 0 \end{pmatrix} = 0 \right] \text{ solve, } x \rightarrow 1$$

Решение системы уравнений можно оформить следующим образом:

$$\begin{pmatrix} x + y = 3 \\ x^3 + y^3 = 9 \end{pmatrix} \text{ solve, } x, y \rightarrow \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \quad \begin{pmatrix} x + y = 3 \\ x^3 + y^3 = 9 \end{pmatrix} \text{ solve, } \begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

Если среди решений системы есть комплексные, например:

$$\begin{pmatrix} \frac{x+y}{x-y} + \frac{x-y}{x+y} = 5.2 \\ x \cdot y = 6 \end{pmatrix} \text{ solve, } x, y \rightarrow \begin{pmatrix} 3 \cdot i & -2 \cdot i \\ -3 \cdot i & 2 \cdot i \\ 3 & 2 \\ -3 & -2 \end{pmatrix}$$

то выделить только вещественные пары можно так:

$$\begin{pmatrix} \frac{x+y}{x-y} + \frac{x-y}{x+y} = 5.2 \\ x \cdot y = 6 \\ \text{Im}(x) = 0 \\ \text{Im}(y) = 0 \end{pmatrix} \text{ solve, } \begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} 3 & 2 \\ -3 & -2 \end{pmatrix}$$

Можно решить уравнение при заданном условии, например:

$$\left[\begin{pmatrix} \frac{x^2 - 1}{(x - 1)^2} + \frac{x^3 - 8}{(x - 2)^2} = x \\ x > 1 \end{pmatrix} \right] \text{ solve, } x \rightarrow \frac{6}{5}$$

Пример решения неравенства:

$$\frac{2x^2 - 5x - 7}{x - 5} \geq 3 \text{ solve, } x \rightarrow \begin{pmatrix} (x \leq -2) \\ 5 < x \end{pmatrix}$$

Пример 8.35 (Символьные преобразования над функциями пользователя).

$$f(x) := \frac{x^2 + 2x - 3}{x \cdot (x^2 - 4x + 4)}$$

$$f(x) = 0 \text{ solve, } x \rightarrow \begin{pmatrix} -3 \\ 1 \end{pmatrix}$$

$$f(x) \text{ convert, parfrac, } x \rightarrow \frac{-3}{4 \cdot x} + \frac{5}{2 \cdot (x-2)^2} + \frac{7}{4 \cdot (x-2)}$$

$$f\left(\frac{1}{3}\right) \rightarrow \frac{-12}{5} \quad f\left(\frac{1}{3}\right) \text{ float, } 5 \rightarrow -2.4000$$

$$f(z) \text{ factor} \rightarrow (z+3) \cdot \frac{(z-1)}{z \cdot (z-2)^2}$$

$$\frac{d}{dx} f(x) \rightarrow \frac{(2 \cdot x + 2)}{x \cdot (x^2 - 4 \cdot x + 4)} - \frac{(x^2 + 2 \cdot x - 3)}{x^2 \cdot (x^2 - 4 \cdot x + 4)} - \frac{(x^2 + 2 \cdot x - 3)}{x \cdot (x^2 - 4 \cdot x + 4)^2} \cdot (2 \cdot x - 4)$$

$$\frac{d}{dx} f(x) \text{ simplify} \rightarrow \frac{-(x^3 + 6 \cdot x^2 - 9 \cdot x + 6)}{(x-2) \cdot (x^2 - 4 \cdot x + 4) \cdot x^2}$$

$$\frac{d}{dx} f(x) \left| \begin{array}{l} \text{solve, } x \\ \text{float, } 3 \end{array} \right. \rightarrow \begin{pmatrix} -7.34 \\ .67 - .606 \cdot i \\ .67 + .606 \cdot i \end{pmatrix}$$

Задания

Задание 1. Упростите алгебраическое выражение:

1., 20. $\left(\frac{a\sqrt{a} + b\sqrt{b}}{\sqrt{a} + \sqrt{b}} - \sqrt{ab} \right) : (a-b)$

2., 21. $\left(\frac{\sqrt{a^3} - \sqrt{b^3}}{\sqrt{a} - \sqrt{b}} + \sqrt{ab} \right) \left(\frac{\sqrt{a} - \sqrt{b}}{a-b} \right)^2$

3., 22. $\left(\left(\frac{\sqrt{a} + \sqrt{b}}{\sqrt[4]{a} - \sqrt[4]{b}} \right)^{-1} - \frac{2\sqrt[4]{ab}}{b^{\frac{3}{4}} - a^{\frac{1}{4}}b^{\frac{1}{2}} + a^{\frac{1}{2}}b^{\frac{1}{4}} - a^{\frac{3}{4}}} \right)^{-1}$

$$4., 23. \left(\frac{\sqrt[4]{ab^3} - \sqrt[4]{a^3b} + \sqrt{ab+1}}{\sqrt{a} - \sqrt{b}} + \frac{\sqrt{ab+1}}{\sqrt[4]{ab}} \right)^{-2} \cdot \sqrt{1 + \frac{a}{b} + 2\sqrt{\frac{a}{b}}}$$

$$5., 24. \frac{2b\sqrt{x^2-1}}{x-\sqrt{x^2-1}} = A, \text{ где } x = \frac{1}{2} \left(\sqrt{\frac{a}{b}} + \sqrt{\frac{b}{a}} \right)$$

$$6., 25. \left(\frac{a}{a-1} + \frac{1}{ab-b} \right) \left(\frac{ab+1}{b} \right)^{-1} - \frac{a}{a^2-1}$$

$$7., 26. \left(\frac{\sqrt{a}+2}{(\sqrt{a}+1)^2} - \frac{\sqrt{a}-2}{a-1} \right) \frac{\sqrt{a}+1}{\sqrt{a}}$$

$$8., 27. \left(\frac{a}{a-b} - \frac{b}{a+b} \right) \left(\frac{a+b}{b} - \frac{a-b}{a} \right)$$

$$9., 28. \left(\frac{a\sqrt{a}+b\sqrt{b}}{\sqrt{a}+\sqrt{b}} - \sqrt{ab} \right) \left(\frac{\sqrt{a}+\sqrt{b}}{a-b} \right)^2$$

$$10., 29. \left(2x+1 - \frac{1}{1-2x} \right) \left(2x - \frac{4x^2}{2x-1} \right)$$

$$11., 30. (\sqrt{1-x^2}+1) \left(\frac{1}{\sqrt{1+x}} + \sqrt{1-x} \right)$$

$$12. \frac{0,25(a^{-3/2} - a^{3/2})}{a^{-1} + a + 1} \left(\frac{\sqrt{a}-1}{\sqrt{a}+1} - \frac{\sqrt{a}+1}{\sqrt{a}-1} \right)$$

$$13. \left(p-q + \frac{4q^2-p^2}{p+q} \right) \left(\frac{p}{p^2-q^2} + \frac{2}{q-p} + \frac{1}{p+q} \right)$$

$$14. \frac{(a-b)^3(\sqrt{a}+\sqrt{b})^{-3} + 2a\sqrt{a}+b\sqrt{b}}{a\sqrt{a}+b\sqrt{b}} + \frac{3(\sqrt{ab}-b)}{a-b}$$

$$15. \left(\frac{a}{a+\sqrt{ab}} - \frac{b}{\sqrt{ab}+b} \right) \left(\frac{\sqrt{a}}{\sqrt{b}} - \frac{\sqrt{b}}{\sqrt{a}} \right) \sqrt{ab}$$

$$16. \left(\frac{2}{2x+y} - \frac{1}{2x-y} - \frac{3y}{y^2-4x^2} \right) \cdot \left(\frac{y^2}{8x^2} - 0,5 \right)$$

$$17. \frac{4x(x+\sqrt{x^2-1})^2}{(x+\sqrt{x^2-1})^4 - 1}$$

18. $\left(\frac{x^3 - 8}{x^2 + 4 + 2x} + 4 \right) \cdot \frac{1}{x + 2}$

19. $\left(\frac{x^2 - y^2}{x - y} - \frac{x^2 + y^2}{x + y} \right) \cdot \frac{x^2 - y^2}{2xy}$

Задание 2. Разложите выражение по степеням (раскройте скобки) (функция **expand**):

1., 7., 13., 19., 25. $\sin(x) - 2\sin(2x) + 4\sin(3x)$

2., 8., 14., 20., 26. $\cos(2x) - \sin(2x) + \sin(3x)$

3., 9., 15., 21., 27. $\text{tg}(2x) + \text{ctg}(2x) + \cos(2x)$

4., 10., 16., 22., 28. $\text{ctg}(2x) - \sin(x) \cdot \cos(3x)$

5., 11., 17., 23., 29. $\cos(2x) \cdot \sin(x) \cdot \sin(3x)$

6., 12., 18., 24., 30. $\text{tg}(2x) \cdot \sin(2x) - \sin(3x)$

Задание 3. Разложите на простые множители выражения (функция **factor**):

1., 9., 17., 25. $a^5 - 32$

2., 10., 18., 26. $a^4 - 16$

3., 11., 19., 27. $3x^2 + 10xy - 8y^2 - 8x + 10y - 3$

4., 12., 20., 28. $\frac{x^2 + 5x + 4}{x^4 + 5x^2 + 4}$

5., 13., 21., 29. $18ab^2 + 27a^2b$

6., 17., 22., 30. $1 - 64b^2$

7., 15., 23. $2x^2 - 20xy + 50y^2 - 2$

8., 16., 24. $x^3y - xy - x^3 + x$

Задание 4. Сделайте замену $x = y + z$ в функции $g(x)$, используя функцию замены **substitute**.

Затем приведите подобные слагаемые в полученном выражении относительно каждой из переменных y, z (функция **collect**).

№ варианта	$g(x)$	№ варианта	$g(x)$
1., 16.	$x^4 - 2x^3 + x^2 - 12x + 20$	9., 24.	$x^4 + x^3 - 17x^2 - 45x - 100$
2., 17.	$x^4 + 6x^3 + x^2 - 4x - 60$	10., 25.	$x^4 - 5x^3 + x^2 - 15x + 50$
3., 18.	$x^4 - 14x^2 - 40x - 75$	11., 26.	$x^4 - 4x^3 - 2x^2 - 20x + 25$
4., 19.	$x^4 - x^3 + x^2 - 11x + 10$	12., 27.	$x^4 + 5x^3 + 7x^2 + 7x - 20$
5., 20.	$x^4 - x^3 - 29x^2 - 71x - 140$	13., 28.	$x^4 - 7x^3 + 7x^2 - 5x + 100$
6., 21.	$x^4 + 7x^3 + 9x^2 + 13x - 30$	14., 29.	$x^4 + 10x^3 + 36x^2 + 70x + 75$
7., 22.	$x^4 + 3x^3 - 23x^2 - 55x - 150$	15., 30.	$x^4 + 9x^3 + 31x^2 + 59x + 60$
8., 23.	$x^4 - 6x^3 + 4x^2 + 10x + 75$		

Информационные технологии в менеджменте

Задание 5. Для выражения из задания 4 определите коэффициенты полинома относительно каждой из переменных y, z .

Задание 6. Разложите дробь на элементарные дроби:

№	Выражение	№	Выражение
1.	$\frac{1 + 2x}{0,5 + 3x + 4,5x^2 + 3x^3}$	16.	$\frac{11x^2 + 9x + 1}{x^2 + 3x + 2}$
2.	$\frac{1 + 4x}{1,2 + 5x + 9x^2 + 3x^3}$	17.	$\frac{0,5 + 2x + 0,5x^2 + x^3}{2,78x + 3,83x^2 + 2,592x^3}$
3.	$\frac{0,5 + 3x}{0,5 + 2x + 9x^2 + 3x^3}$	18.	$\frac{1,58 + 0,92x - 0,64x^2 + x^3}{11,71x + 10,49x^2 + 5,16x^3 + x^4}$
4.	$\frac{1 + x + 0,5x^2}{1 + 2x + 7x^2 + 3x^3}$	19.	$\frac{x + 3x^2 + 0,5x^3 + x^4}{1 + 4x + 6x^2 + 4x^3 + x^4}$
5.	$\frac{1 + x + 0,5x^2}{2,787x + 3,83x^2 + 2,592x^3}$	20.	$\frac{0,08 + 0,36x + 0,30x^2 + x^3}{2,6x + 3,4x^2 + 2,6x^3 + x^4}$
6.	$\frac{0,5 + 2x + 0,5x^2}{1 + 2,78x + 3,83x^2 + 2,52x^3}$	21.	$\frac{0,5 + 3x + 4,25x^2 + 3x^3 + x^4}{0,01 + 2x + 9x^2 + 4x^3 + 2x^4}$
7.	$\frac{0,2 + 1,5x}{1,87x + 3,83x^2 + 2,592x^3}$	22.	$\frac{1,57 + 0,92x - 0,64x^2 + x^3}{1 + 2,78x + 3,36x^2 + 2,52x^3}$
8.	$\frac{1 + 1,5x}{4x + 6x^2 + 4x^3}$	23.	$\frac{0,08 + 0,36x + 0,01x^2 + x^3}{9,44x^2 + 10,35x^3 + 4,13x^4}$
9.	$\frac{1 + 2x}{2,6x + 3,4x^2 + 2,6x^3}$	24.	$\frac{x + 3x^2 + 0,5x^3}{1 + 2,78x + 3,83x^2 + 2,92x^3}$
10.	$\frac{1 + 0,5x}{2x + 9x^2 + 4x^3}$	25.	$\frac{1 + 3x + 0,5x^2 + x^3}{4,80 + 11x + 10,49x^2 + 5x^3}$
11.	$\frac{1 + 2x + 5x^2 + 4x^3}{0,5 + 3x + 4,25x^2 + 3x^3}$	26.	$\frac{0,5 + 2x + 0,5x^2 + x^3}{x + 5x^2 + 6x^3 + 3x^4 + x^5}$
12.	$\frac{0,083 + 0,367x + 0,301x^2}{x + 5x^2 + 6x^3 + 3x^4}$	27.	$\frac{x + 3x^2 + 0,5x^3}{1,57x + 3,55x^2 + 3,16x^3}$
13.	$\frac{4,62 + 8,31x + 5x^2}{0,18 + 1,57x + 3,55x^2 + 3,11x^3}$	28.	$\frac{1,58 + 0,92x - 0,64x^2 + x^3}{2,6x + 3,4x^2 + 2,6x^3 + x^4}$
14.	$\frac{1 + 3x + 0,5x^2 + x^3}{9,44x^2 + 10,3x^3 + 4,17x^4}$	29.	$\frac{1 + 3x + 0,5x^2 + x^3}{0,5 + 3x + 4,25x^2 + 3x^3 + x^4}$
15.	$\frac{x + 3x^2 + 0,5x^3}{2,78x + 3,83x^2 + 2,52x^3}$	30.	$\frac{3,5 + 4,2x}{5,67x + 4,89x^2 + 7,52x^3}$

Задание 7. Разложите в ряд выражение:

№ варианта	$f(x)$	№ варианта	$f(x)$	№ варианта	$f(x)$
1, 16	$1/(\operatorname{tg}2x + 1)$	6, 21	$x^2 \operatorname{arctg}(x/3)$	11, 26	$(2x + 3) \sin x$
2, 17	$\cos x/(2x + 5)$	7, 22	$e^{2x} \sin 3x$	12, 27	$\cos 3x/(1 - \cos 3x)^2$
3, 18	$1/(x\sqrt{x^3 + 4})$	8, 23	$\operatorname{ctg}2x/(\sin 2x)^2$	13, 28	$1/(1 + x + x^2)$
4, 19	$\sin x/(1 + \sin x)$	9, 24	$(x + 1) \sin x$	14, 29	$(1 + x)/(2 + x)$
5, 20	$x^2 \lg(x + 2)$	10, 25	$5x + x \lg x$	15, 30	$\sqrt{1 + e^{-x}}$

Порядок разложения 6.

Задание 8. Решите уравнение с помощью функции **solve**:

$$1) \ 9) \ 17) \ 25) \quad 2 - 3x^2 + \frac{1}{x-2} = 5x + \frac{x+2}{x^2-4};$$

$$3) \ 11) \ 18) \ 26) \quad x^5 + 5x^4 - 6x^3 - 44x^2 + 8x + 96 = 0;$$

$$5) \ 13) \ 19) \ 27) \quad |x^2 - 3x + 2| + |x^2 - 5x + 6| = 2;$$

$$7) \ 15) \ 20) \ 28) \quad \frac{1+x-\sqrt{2x+x^2}}{1+x+\sqrt{2x+x^2}} = \frac{1}{8} \cdot \frac{\sqrt{2+x}+\sqrt{x}}{\sqrt{2+x}-\sqrt{x}};$$

$$2) \ 10) \ 21) \ 29) \quad \frac{x^3-3x+12}{x} + \frac{x^2-3x-16}{x+1} = 1;$$

$$4) \ 12) \ 22) \ 30) \quad |x-1| + |x-3| = 2x-4;$$

$$6) \ 14) \ 23) \quad \left| x^3 - \sqrt{x+1} - 3 \right| = x^3 + \sqrt{x+1} - 7;$$

$$8) \ 16) \ 24) \quad \sqrt[3]{\frac{x+1}{x-1}} - \sqrt[3]{\frac{x-1}{x+1}} = \frac{64}{73} \cdot \frac{x}{x^2-1}.$$

Задание 9. Решите неравенство с помощью функции **solve**:

1) 9) 17) 25) $(x^2 - x - 1)(x^2 - x - 7) < -5;$

3) 11) 18) 26) $|x-1| + |x-2| \leq |2x-3|;$

5) 13) 19) 27) $|x+3| \leq 6 - 3\sqrt{1-x};$

7) 15) 20) 28) $\sqrt{2 - \frac{2}{x+1}} < \sqrt{2 + \frac{2}{x}} + 1;$

2) 10) 21) 29) $\frac{1}{2x+1} - \frac{2}{2+3x} + \frac{3}{3+4x} - \frac{4}{4+5x} < 0;$

4) 12) 22) 30) $\frac{|x-4| - |x-1|}{|x-3| - |x-2|} < \frac{|x-3| + |x-2|}{|x-4|};$

6) 14) 23) $\sqrt{4+3x-x^2} > \frac{\sqrt{6}}{5} - \frac{1}{2};$

8) 16) 24) $|x^3 - 7x + 6| + |x^3 - 3x + 2| \leq 4(1-x).$

Задание 10. Найдите преобразование Лапласа от функции: $x^2 + 4$.

Найдите обратное преобразование Лапласа от функции: $\frac{2}{s^3} + \frac{4}{s}$.

Лабораторная работа

Mathcad: Операторы суммы Σ и произведения Π .

Дифференцирование и интегрирование

Цель работы: получение навыков работы с операторами суммы Σ и произведения Π , изучение способов дифференцирования и интегрирования в пакете Mathcad.

Теоретические сведения

Операторы суммы Σ и произведения Π

Вычисление сумм и произведений в пакете Mathcad осуществляется с использованием панели **Calculus (Калькуляция)** на панели инструментов **Math (Математика)**.

Операция	Запись	Клавиша или комбинация клавиш
Сумма (индекс суммирования должен быть определен заранее)	$\sum_i x_i$	Shift+4 (символ \$)
Сумма (значения индекса суммирования задаются как элементы оператора)	$\sum_{i=1}^N x_i$	Ctrl+Shift+4
Произведение (индекс произведения должен быть определен заранее)	$\prod_i x_i$	Shift+3 (символ #)
Произведение (значения индекса произведения задаются как элементы оператора)	$\prod_{i=1}^N x_i$	Ctrl+Shift+3

Ввод оператора приводит к выводу на экран соответствующего шаблона с пустыми знаками, которые следует заполнить требуемыми выражениями:

$$\sum, \quad \sum_{i=}, \quad \prod, \quad \prod_{i=}$$

Индекс в операторах является простой переменной и может принимать только неотрицательные целые значения. Выражения, стоящие под знаками суммы или произведения, сами в свою очередь могут содержать названные операторы. Например,

$$\sum_{i=1}^{10} \left(\cos(2 \cdot j) + \prod_{j=1}^i j + \sum_{j=i}^N j^2 \right)$$

Использование операторов суммы и произведения, для которых указывается только имя переменной индекса, требует предварительного ее задания как переменной диапазона с произвольным целым шагом.

Пример 5.5. Для заданного N найдите сумму:

$$\frac{1}{2 \cdot 3} + \frac{2}{3 \cdot 4} + \dots + \frac{N}{(N+1) \cdot (N+2)}$$

Решение:

$$N := 100$$

$$S := \sum_{n=1}^N \frac{n}{(n+1) \cdot (n+2)} \quad S = 3.217$$

Информационные технологии в менеджменте

Пример 5.6. Для числовой последовательности, состоящей из N членов, n-е значение которой задается выражением $\sin(2n+1)$, найти отношение суммы членов с четными номерами к сумме членов с нечетными номерами. Нумерация членов начинается с единицы.

Решение:

$$N := 20$$

$$i := 2, 4..N \quad j := 1, 3..N$$

$$k := \frac{\sum_{i} \sin(2 \cdot i + 1)}{\sum_{j} \sin(2 \cdot j + 1)} \quad k = -1.011$$

Второй вид операторов суммы и произведения содержит задание диапазона изменения переменной индекса, шаг изменения значений которой по умолчанию равен 1. С использованием оператора суммы второго вида решение может быть оформлено следующим образом:

$$N := 20$$

$$\frac{\sum_{i=1}^N \sin(2 \cdot i + 1) \cdot (\text{mod}(i, 2) = 0)}{\sum_{i=1}^N \sin(2 \cdot i + 1) \cdot (\text{mod}(i, 2) = 1)} = -1.011$$

Пример 5.7. Найти значение выражения $b_1 + b_2 s_1 + b_3 s_1 s_2 + \dots + b_N s_1 s_2 \dots s_{N-1}$, где $b_i = i^2 \cdot \sin(i)$ и $s_i = \cos(2i)$.

Решение:

$$N := 10$$

$$i := 1..N$$

$$b_i := i^2 \cdot \sin(i) \quad s_i := \cos(2 \cdot i)$$

$$b_1 + \sum_{i=2}^N \left(b_i \cdot \prod_{j=1}^{i-1} s_j \right) = -1.791$$

Дифференцирование и интегрирование

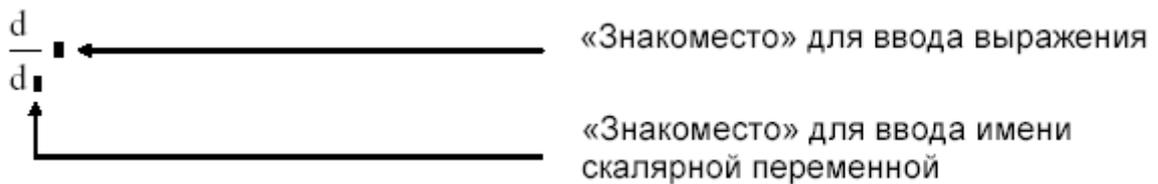
Интегрирование и дифференцирование функций в пакете Mathcad осуществляется с использованием панели **Calculus (Калькуляция)** на панели инструментов **Math (Математика)**.

Дифференцирование. Для вычисления производной первого порядка служит оператор дифференцирования, ввод которого можно выполнить нажатием



кнопки панели **Calculus** или комбинацией двух клавиш **Shift** и **?**. Шаблон оператора имеет вид:

Информационные технологии в менеджменте



Например, найдем производную функции $y=x^2$ в точке $x=2$:

$$y(x) := x^2 \quad x := 2 \quad \frac{d}{dx} y(x) = 4$$

Чтобы вычислить производную порядка выше первого, можно последовательно применить несколько раз оператор дифференцирования. Например,

$$y(x) := x^3 + 2x \quad x := 2 \quad \frac{d}{dx} \frac{d}{dx} y(x) = 12 \quad \frac{d}{dx} \frac{d}{dx} \frac{d}{dx} y(x) = 6$$

Однако лучше воспользоваться оператором дифференцирования n -го порядка ($0 \leq n \leq 5$), вводимого нажатием кнопки  панели инструментов **Calculus** или комбинацией клавиш **Ctrl+Shift+?**. Шаблон оператора имеет вид:



Заметим, что при заполнении любого из двух знакомест для ввода порядка производной автоматически заполняется второе знакоместо. Теперь предыдущие вычисления можно оформить так:

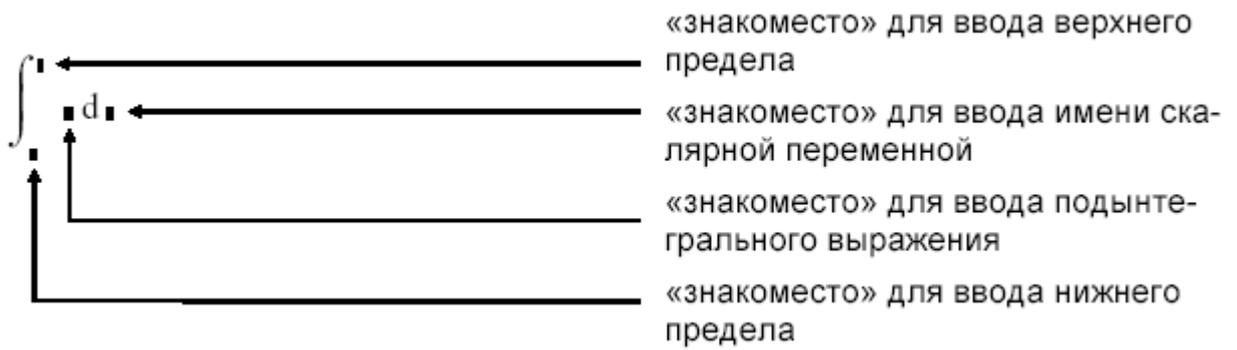
$$y(x) := x^3 + 2x \quad x := 2 \quad \frac{d^2}{dx^2} y(x) = 12 \quad \frac{d^3}{dx^3} y(x) = 6$$

Однако если требуется вычислить производную порядка выше 5, то придется последовательно применить несколько раз оператор дифференцирования n -го порядка.

Например, найдем производную 7-го порядка функции $e^x \cos x$ в точке $x=0.1\pi$:

$$x := 0.1\pi \quad \frac{d^5}{dx^5} \frac{d^2}{dx^2} (e^x \cos(x)) = 13.801$$

Интегрирование. В Mathcad встроена также возможность вычисления определенных интегралов. Ввод оператора определенного интеграла выполняется нажатием кнопки  панели инструментов **Calculus** или комбинацией клавиш **Shift+7** (символ **&**). Шаблон оператора имеет вид:



Задания

Задание 1. Найдите предел выражения:

- | | | |
|---|---|--|
| 1) $\lim_{x \rightarrow 0} \frac{(1+x)^5 - (1+5x)}{x^2 + x^5};$ | 2) $\lim_{x \rightarrow 1} \frac{x^4 - 3x + 2}{x^5 - 4x + 3};$ | 3) $\lim_{x \rightarrow 3} \frac{\sqrt{x+13} - 2\sqrt{x+1}}{x^2 - 9};$ |
| 16) $\lim_{x \rightarrow 0} \frac{(1+x)^5 - (1+5x)}{x^2 + x^5};$ | 17) $\lim_{x \rightarrow 1} \frac{x^4 - 3x + 2}{x^5 - 4x + 3};$ | 18) $\lim_{x \rightarrow 3} \frac{\sqrt{x+13} - 2\sqrt{x+1}}{x^2 - 9};$ |
| 4) $\lim_{x \rightarrow 0} \frac{\sqrt{1+x} - \sqrt{1-x}}{\sqrt[3]{1+x} - \sqrt[3]{1-x}};$ | 5) $\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k(k+1)};$ | 6) $\lim_{n \rightarrow \infty} \frac{\sum_{k=1}^n (2k-1)^2}{\sum_{k=1}^n (2k)^2};$ |
| 19) $\lim_{x \rightarrow 0} \frac{\sqrt{1+x} - \sqrt{1-x}}{\sqrt[3]{1+x} - \sqrt[3]{1-x}};$ | 20) $\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k(k+1)};$ | 21) $\lim_{n \rightarrow \infty} \frac{\sum_{k=1}^n (2k-1)^2}{\sum_{k=1}^n (2k)^2};$ |
| 7) $\lim_{x \rightarrow a} \frac{\cos(x) - \cos(a)}{x - a};$ | 8) $\lim_{x \rightarrow \infty} \frac{\ln(1 + \sqrt{x} + \sqrt[3]{x})}{\ln(1 + \sqrt[3]{x} + \sqrt[4]{x})};$ | 9) $\lim_{x \rightarrow 0} \frac{\int_0^x \cos(t^2) dt}{x}.$ |
| 22) $\lim_{x \rightarrow a} \frac{\cos(x) - \cos(a)}{x - a};$ | 23) $\lim_{x \rightarrow \infty} \frac{\ln(1 + \sqrt{x} + \sqrt[3]{x})}{\ln(1 + \sqrt[3]{x} + \sqrt[4]{x})};$ | 24) $\lim_{x \rightarrow 0} \frac{\int_0^x \cos(t^2) dt}{x}.$ |
| 10) $\lim_{x \rightarrow 1} \frac{x^2 + 2 \cdot x + 5}{x^2 + 1}$ | 11) $\lim_{x \rightarrow \frac{\pi}{2}} (2 \cdot \sin(x) - \cos(x) + \operatorname{ctg}(x))$ | |
| 25) $\lim_{x \rightarrow 1} \frac{x^2 + 2 \cdot x + 5}{x^2 + 1}$ | 26) $\lim_{x \rightarrow \frac{\pi}{2}}$ | |
| 12) $\lim_{h \rightarrow 0} \frac{(x+h)^3 - x^3}{h}$ | 13) $\lim_{x \rightarrow \infty} \frac{\sqrt{x^2 - 3}}{\sqrt[3]{x^3} + 1}$ | |
| 27) $\lim_{h \rightarrow 0} \frac{(x+h)^3 - x^3}{h}$ | 28) $\lim_{x \rightarrow \infty} \frac{\sqrt{x^2 - 3}}{\sqrt[3]{x^3} + 1}$ | |
| 14) $\lim_{x \rightarrow +\infty} x \cdot (\sqrt{x^2 + 1} - x)$ | 15) $\lim_{x \rightarrow -\infty} x \cdot (\sqrt{x^2 + 1} - x)$ | |
| 29) $\lim_{x \rightarrow +\infty} x \cdot (\sqrt{x^2 + 1} - x)$ | 30) $\lim_{x \rightarrow -\infty} x \cdot (\sqrt{x^2 + 1} - x)$ | |

Вычисление пределов в пакете Mathcad осуществляется с использованием панели **Calculus (Калькуляция)** на панели инструментов **Math (Математика)**.

Задание 2. Вычислите выражение:

Номер варианта	Функция	Рабочий набор	
		n	x
1	$y = x + \frac{x}{7+x} \prod_{i=4}^n \ln \left(1 + \frac{x}{7+x} \sqrt{i} \right)$	75	3,5
2	$y = -5x + 6 \sum_{i=1}^n ((-1)^{i+1} (\frac{x}{x+5})^i)$	30	8
3	$y = \sin(0,01 \prod_{i=2}^n (1 + e^{-\frac{i}{x+4}}))$	80	5,3
4	$y = 2^{x + \sum_{i=1}^n \frac{1}{i} (-1)^{i+1}}$	100	1,5
5	$y = \sqrt{x} + x \sum_{i=1}^n (-1)^i \frac{x^i \ln^i x}{i!}$	11	0,8
6	$y = x \sum_{k=1}^n \frac{1}{k} + \prod_{k=1}^n \frac{k+x}{k+1}$	25	1,8
7	$z = 5x + (\sum_{i=1}^n (-1)^{i-1} \frac{x^{2i-1}}{(2i-1)!(2i-1)})^2$	10	0,8
8	$z = x \sum_{k=1}^n (-1)^{k+1} \frac{1}{2k-1} + \prod_{k=1}^n \frac{k^2+x}{k^2+1}$	20	4,5
9	$y = -4x + \prod_{i=1}^n (1 + \frac{1}{i})^{\frac{1}{i}}$	25	-2,1
10	$z = \ln(\prod_{i=1}^n (1 + i^{-1,5} + i^{-3})) + x$	55	3,2
11	$y = 1 + 2 \sum_{k=1}^n (\frac{x}{x+5})^k \cos kx$	20	3,1415
12	$y = 4x + \frac{2}{3} \sum_{k=1}^n \frac{1}{(2k-1)9^{k-1}}$	15	0,4
13	$y = 2,4 \sin x + \prod_{i=6}^n (\frac{x}{x+1} + \frac{i^2}{1+i^2})$	30	9

Информационные технологии в менеджменте

14	$y = 0,5x + \sum_{k=1}^n \frac{x^{2k-2}}{(2k-1)!}$	15	1,2
15	$y = \sin \pi + \prod_{i=5}^n (1 + \frac{x}{1+i^2})^{\ln i}$	35	1,1
16	$y = x + 2 \sum_{k=1}^n 0,5^k \cos kx$	25	1,2
17	$y = 4,2x + \prod_{i=1}^n (\frac{x+0,8}{x} + \cos xi)^{\frac{1}{i}}$	20	0,8
18	$z = \frac{1}{x} + x \sum_{k=1}^n \frac{2x}{x^2 - k^2 \pi^2}$	20	0,5
19	$y = 5x - \left \ln \prod_{i=7}^n \sqrt{\frac{1}{i} + x} \right $	55	0,5
20	$z = \frac{1}{x} + \sum_{k=1}^n (-1)^k \frac{kx}{x^2 + k^2}$	15	1,5
21	$y = 3x + \sum_{k=1}^n \frac{1}{k^2} + (\sum_{k=1}^n \frac{1}{k} (-1)^{k+1})^2$	25	0,35
22	$y = 3x + \sqrt{\prod_{i=8}^n (1 + \frac{\ln i}{i})}$	40	0,5
23	$z = x + 0,1 \sum_{i=8}^n (\frac{3i}{1+x} - \frac{1+x}{3} \sqrt{i})$	45	0,4
24	$y = 4 \prod_{i=2}^n (\frac{x}{x+4} - x \sqrt{1 + \frac{1}{i}})$	40	10
25	$y = \frac{x}{2} + \sum_{i=1}^n \frac{\sin ix}{i}$	25	1,5
26	$y = 4x + \prod_{i=3}^n (1 + \frac{\ln(2 + \cos 0,05i)}{i})$	45	2,4
27	$z = x \prod_{k=1}^n (1 - \frac{x^2}{1+k^2})$	25	0,75
28	$y = 3,4x + x \prod_{i=3}^n (1 + \frac{x}{10+x} \sqrt{i})$	40	0,85

Информационные технологии в менеджменте

29	$z = -\cos\left(0,1 \prod_{i=1}^n \left(1 + \frac{10+x}{x}\right)^{\frac{1}{i}}\right)$	40	5
30	$y = \frac{\sum_{i=1}^n (-1)^i \frac{1}{(i-1)!} \frac{x^{2i+1}}{2i+1}}{5,5+x^2}$	12	4,75
31	$y = 6,3x - 4 \sum_{k=3}^n (2x^3k + \cos(k)\sqrt{x+1} - \frac{2,3}{k})$	20	4,75

Задание 3. Для функции $f(x, y)$, равной следующему выражению:

№	Выражение	№	Выражение
1.	$y \sin x - 3 + \sqrt{6y}(4 + x^2)$	16.	$(1 + e^y)(x^2 + 4) - 8y$
2.	$\sqrt{6x}(4 + x^2) + y + xe^{\frac{3}{y}} - x^2$	17.	$y^7x - 3x^2 - yx^4$
3.	$y + xe^{\frac{3}{x}} - y^2 - y \sin x - 3$	18.	$(x + \ln y) + \left(1 + \frac{x}{5} + \sin y\right)$
4.	$2x^3 - 3y^2 - x^4 + 2tg x - \sin y$	19.	$2y - 5 + 3e^{2x} + xy^2$
5.	$2tg y - \sin x - \sqrt{6y}(4 + x^2)$	20.	$y^6 + x^3 + 45yx + x^2$
6.	$5\cos^2 y + 7 - x^2(y - 1)$	21.	$(1 - x)y^2 + 6x$
7.	$x^2(y - 1) - x^3 - x(y - 2)$	22.	$6y - 2\sqrt{xy} \cdot \ln x$
8.	$\frac{2x(1 - e^{yx})}{(1 + y^2)^2}$	23.	$xy + \frac{x + 3}{x + 2xy^3}$
9.	$2 - 9y^2 + (4x^2 - 6x^3)y$	24.	$6yx^4 - 2x + xy^2$
10.	$x(1 + y^2) + 1 + yx^2$	25.	$5tg(yx) + \sin((2x) + y^5)$
11.	$\cos y - (x^2 + 1) \cdot \sin(yx)$	26.	$xy^4 + x(2\ln(xy) + 1)$
12.	$\frac{5}{x(y-1)} + \frac{3y}{x+2}$	27.	$x - 4xy^2 + x^3e^{-y}$
13.	$y - 45 + \sqrt{x^2 + xy^5}$	28.	$8xy \left(\sqrt[3]{y} - \frac{x^2}{8} \sin(2xy) \right)$
14.	$x \cdot (\ln y^4 - \ln x + 4) - \ln \frac{3}{y}$	29.	$2xy^{-3} + (xy^4 - 3x^2)x^{-4}$
15.	$y^6 + 2x + xy^3$	30.	$xy^3 + \sqrt{2x} - y - 1$

найдите первую и вторую частные производные по x и по y аналитически и в точке $(1; 0,1)$.

Частные производные в Mathcad определяются так же, как и обычные.

Задание 4. Аналитически найдите производные сложных функций:

1. $z = x \sin y + y \cos x$, где $x = \frac{u}{v}$, $y = u^3 v^2$; $\frac{\partial z}{\partial u}$, $\frac{\partial z}{\partial v} = ?$

2. $z = e^{4xy}$, где $x = \cos(1-t)$, $y = \sin t^2$; $\frac{dz}{dt} = ?$

3. $z = x^2 - y^2 + 2xy$, где $x = \sin t$, $y = \arccos(e^t)$; $\frac{dz}{dt} = ?$

4. $z = (x + y^3) \cdot e^{x^2+y^2}$, где $x = \cos(t^2)$, $y = \sqrt{t}$; $\frac{dz}{dt} = ?$

5. $z = \arcsin \frac{x^2}{y}$, где $x = \sin t$, $y = \cos^2 \frac{t}{2}$; $\frac{dz}{dt} = ?$

6. $z = \operatorname{tg}(x + 2x^2 - y)$, где $x = \frac{1}{t}$, $y = \sqrt{t}$; $\frac{dz}{dt} = ?$

7. $z = \frac{1}{\sqrt{x^2 + y^2}}$, где $x = v - u^2 v$, $y = u + v^2 u$; $\frac{\partial z}{\partial u}$, $\frac{\partial z}{\partial v} = ?$

$$8. z = \operatorname{arctg} \frac{x}{\sqrt{y}}, \text{ где } y = \sqrt{x^2 + 1}, \quad \frac{dz}{dx} = ?$$

$$9. z = \frac{4y}{\sqrt{y^2 - x}}, \text{ где } x = t \cdot \cos t, y = t \cdot \sin t; \quad \frac{dz}{dt} = ?$$

$$10. z = \arcsin(x - y), \text{ где } x = \ln(\sqrt{t} + 1), y = 4t^3; \quad \frac{dz}{dt} = ?$$

$$11. z = \operatorname{arctg} \frac{x + y}{1 + xy}, \text{ где } x = t^2 + 1, y = \sin t; \quad \frac{dz}{dt} = ?$$

$$12. z = \ln(\sqrt{x} \cdot \ln y), \text{ где } x = \sin t, y = \arccos(t^5); \quad \frac{dz}{dt} = ?$$

$$13. z = \ln(e^{2x} + e^{6y}), \text{ где } y = x\sqrt{x}; \quad \frac{dz}{dx} = ?$$

$$14. z = \frac{1}{2} \ln \frac{x}{y}, \text{ где } x = \operatorname{tg}^2 t, y = \operatorname{ctg}^3 \frac{t}{2}; \quad \frac{dz}{dt} = ?$$

$$15. z = \ln(x^2 y), \text{ где } x = u^v, y = v^u; \quad \frac{\partial z}{\partial u}, \frac{\partial z}{\partial v} = ?$$

$$16. z = x^3 \sin y + y^3 \cos x, \text{ где } x = 3t^2 - \sqrt{t}, y = \frac{t^2}{t-1}; \quad \frac{dz}{dt} = ?$$

$$17. z = \frac{x+2y}{xy}, \text{ где } x = tg(t^2 + 1), y = ctg(t^4 - 1); \frac{dz}{dt} = ?$$

$$18. z = xy^2 + \frac{x}{y}, \text{ где } x = \ln(t^2 + t), y = 10^t; \frac{dz}{dt} = ?$$

$$19. z = \ln(x^2 + y^2) - x\sqrt{x}, \text{ где } x = \sin t, y = \frac{t}{t+1}; \frac{dz}{dt} = ?$$

$$20. z = \frac{x-2y}{x+2y}, \text{ где } x = \frac{u-v}{u}, y = \frac{v}{u+v}; \frac{\partial z}{\partial u}, \frac{\partial z}{\partial v} = ?$$

$$21. z = \frac{x+y}{1-xy}, \text{ где } x = \frac{t}{\sin t}, y = \frac{t}{\cos t}; \frac{dz}{dt} = ?$$

$$22. z = \sqrt{xy + \sin x}, \text{ где } x = tg(3t^2 + t), y = ctg(2t + 1); \frac{dz}{dt} = ?$$

$$23. z = \sqrt{\frac{x+y}{x-y}}, \text{ где } x = tg(e^t + 1), y = ctg \frac{t}{2}; \frac{dz}{dt} = ?$$

$$24. z = x^{\ln y}, \text{ где } x = \sin(uv), y = \cos(v^2 - u); \frac{\partial z}{\partial u}, \frac{\partial z}{\partial v} = ?$$

$$25. z = \frac{\sin^2 x}{\sin^2 y}, \text{ где } x = \ln t, y = te^t; \frac{dz}{dt} = ?$$

$$26. z = \ln x - \frac{y^2}{x}, \text{ где } x = \sin t^2, y = \cos t^2; \frac{dz}{dt} = ?$$

$$27. z = \arccos(xy^2), \text{ где } x = t^2 \ln t, y = te^t; \frac{dz}{dt} = ?$$

$$28. z = \cos(y^{\sin x}), \text{ где } x = t^3, y = 2^t; \frac{dz}{dt} = ?$$

$$29. z = \sin^2 x + \cos^2 y, \text{ где } x = \pi^{\ln t}, y = \ln^\pi t; \frac{dz}{dt} = ?$$

$$30. z = \frac{x^2 + 2y^3}{x+2y}, \text{ где } x = \frac{u+v^2}{u}, y = \frac{v^2}{u+v}; \frac{\partial z}{\partial u}, \frac{\partial z}{\partial v} = ?$$

Задание 5. Найдите неопределенные интегралы от функции $f(x)$:

№	f(x)	№	f(x)	№	f(x)
1	$\frac{1}{\sin^2 x(1 - \cos x)}$	11	$\frac{1 + \cos^2 x}{1 + \cos 2x}$	21	$\frac{\cos x + 1}{(1 + \sin x + \cos x)}$
2	$\frac{\cos x - \sin x}{(1 + \sin x)^2}$	12	$\frac{1}{(\sin x + \cos x)^2}$	22	$\frac{\sin x}{(1 + \sin x + \cos x)}$
3	$\frac{1}{\sin x(1 - \sin x)}$	13	$\frac{1}{(2 \sin x + 3 \cos x)}$	23	$\frac{\cos x}{(1 + \sin x + \cos x)}$
4	$\frac{\cos x}{(5 + 4 \cos x)}$	14	$\frac{1}{(5 - 3 \cos x)}$	24	$\frac{\cos x}{(1 + \sin x + \cos x)^2}$
5	$\frac{\cos x}{(1 + \sin x - \cos x)}$	15	$\frac{2 - \sin x}{(2 + \cos x)}$	25	$\frac{\sin x}{(1 + \sin x)^2}$
6	$\frac{\cos x}{(2 + \sin x)}$	16	$\frac{1}{(5 + 4 \sin x)}$	26	$\frac{\cos x}{(1 - \sin x + \cos x)}$
7	$\frac{\cos x}{(1 - \cos x)^2}$	17	$\frac{1}{\sin^2 x + 1}$	27	$\frac{\sin x}{(1 - \sin x + \cos x)^2}$
8	$\frac{1}{\cos x(1 - \cos x)}$	18	$\frac{1}{1 - \sin^4 x}$	28	$\frac{\sin^2 x}{(1 + \sin x + \cos x)^2}$
9	$\frac{1}{(1 + \sin x - \cos x)^2}$	19	$\frac{1}{1 + \operatorname{tg} x}$	29	$\frac{1}{\sin x(1 + \sin x)}$
10	$\frac{\sin x + 1}{(1 + \sin x - \cos x)}$	20	$\frac{\sin^2 x}{1 - \operatorname{tg} x}$	30	$\frac{\sin x}{(2 + \sin x)}$

Задание 6. Найдите определенные интегралы:

№	задание	№	задание	№	задание
1	$\int_{1,6}^{2,2} \frac{dx}{\sqrt{x^2 + 2,5}}$;	11	$\int_{0,5}^{1,2} \frac{\operatorname{tg}(x^2)}{x+1} dx.$	21	$\int_{1,4}^2 \frac{dx}{\sqrt{2x^2 + 0,7}}$;
2	$\int_{0,6}^{1,6} \frac{dx}{\sqrt{x^2 + 0,8}}$;	12	$\int_{1,3}^{2,1} \frac{\sin(x^2 - 1)}{2\sqrt{x}} dx.$	22	$\int_{3,2}^4 \frac{dx}{\sqrt{0,5x^2 + 1}}$;
3	$\int_{1,2}^2 \frac{dx}{\sqrt{x^2 + 1,2}}$;	13	$\int_{0,2}^{1,0} (x+1) \cos(x^2) dx.$	23	$\int_{1,2}^{2,8} \frac{\lg(1+x^2)}{2x-1} dx.$
4	$\int_{1,2}^2 \frac{\lg(x^2 + 3)}{2x} dx.$	14	$\int_{0,8}^{1,2} \frac{\sin(x^2 - 0,4)}{x+2} dx.$	24	$\int_{0,8}^{4,7} \frac{dx}{\sqrt{2x^2 + 0,3}}$;
5	$\int_{2,5}^{3,3} \frac{\lg(x^2 + 0,8)}{x-1} dx.$	15	$\int_{0,15}^{0,63} \sqrt{x+1} \lg(x+3) dx.$	25	$\int_{1,2}^{2,0} \frac{dx}{\sqrt{0,5x^2 + 1,5}}$;
6	$\int_{2,1}^{3,6} \frac{dx}{\sqrt{x^2 - 3}}$;	16	$\int_{1,3}^{2,5} \frac{dx}{\sqrt{0,2x^2 + 1}}$;	26	$\int_{0,6}^{1,4} \frac{dx}{\sqrt{12x^2 + 0,5}}$;
7	$\int_{0,4}^{1,2} \frac{\cos(x^2)}{x+1} dx.$	17	$\int_{0,6}^{1,4} x^2 \cos x dx.$	27	$\int_{1,3}^{2,1} \frac{dx}{\sqrt{3x^2 - 0,4}}$;
8	$\int_{0,6}^{0,72} (\sqrt{x} + 1) \operatorname{tg} 2x dx.$	18	$\int_{0,8}^{1,2} \frac{\cos x}{x^2 + 1} dx.$	28	$\int_{1,4}^{2,6} \frac{dx}{\sqrt{1,5x^2 + 0,7}}$;
9	$\int_{1,6}^{3,2} \frac{x}{2} \lg\left(\frac{x^2}{2}\right) dx.$	19	$\int_{1,2}^{2,8} \left(\frac{x}{2} + 1\right) \sin \frac{x}{2} dx.$	29	$\int_{0,8}^{1,6} \frac{\lg(x^2 + 1)}{x+1} dx.$
10	$\int_{0,32}^{0,66} \frac{dx}{\sqrt{x^2 + 2,3}}$;	20	$\int_{2,3}^{0,5} \frac{dx}{\sqrt{x^2 - 4}}$;	30	$\int_{0,15}^{0,5} \frac{dx}{\sqrt{2x^2 + 1,6}}$;

Лабораторная работа

Mathcad: Решение уравнений и их систем

Цель работы: получение навыков решения уравнений и их систем в пакете Mathcad.

Теоретические сведения

Численное решение уравнения. Многие уравнения, например трансцендентные, не имеют аналитических решений. Однако корни таких уравнений могут быть найдены численными методами с некоторой заранее заданной погрешностью. В Mathcad для этой цели запрограммирован *метод хорд*. Для поиска корней уравнений вида $y(x)=0$ служит встроенная функция **root**. Она

Информационные технологии в менеджменте

возвращает найденное с заданной точностью значение неизвестной x , обращающее функцию $y(x)$ в 0 (*корень функции*). Функция **root** имеет следующий формат:

root(y(x),x)

Таким образом, **root** — это функция двух аргументов. Перед обращением к ней, как и в случае любой другой функции, ее аргументы должны быть уже определены. Первый аргумент **y(x)** — это сама функция, корень которой мы ищем, а **x** — скалярная переменная, относительно которой ищется корень (или решается уравнение). Переменной **x** перед обращением к функции должно быть присвоено некоторое значение — приближенное значение корня, которое можно определить графически (причем для поиска комплексного корня должно быть задано приближенное комплексное значение корня).

Чтобы получить искомый корень, необходимо, чтобы между приближенным и точным значениями корня не лежали точки перегиба функции.

Пример 4.1. Решить уравнение $x + \sin(x) = 2$.

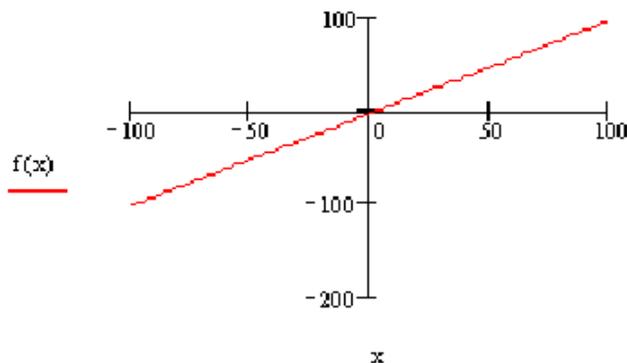
Решение:

1. Определим функцию, корни которой мы ищем:

$$f(x) := x + \sin(x) - 2$$

2. Построим предварительный (на достаточно большом промежутке) график функции, чтобы определить наличие и расположение корней:

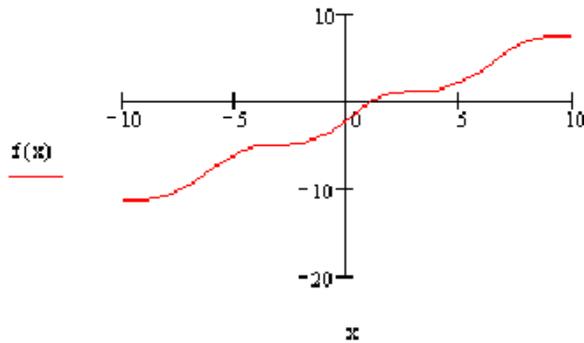
$$x := -100..100$$



Из графика видно, что корень (или корни) функции лежат в окрестности точки $x=0$. Этот же факт можно было установить, проанализировав поведение функции на бесконечности.

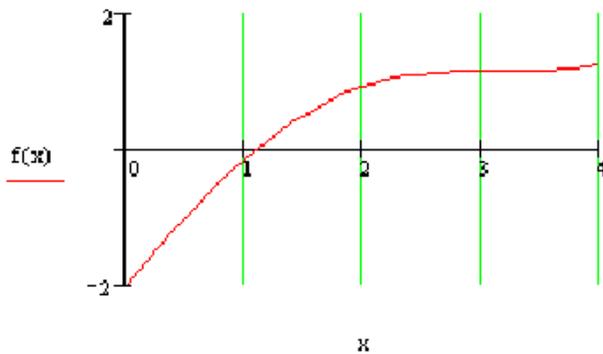
3. Уменьшим промежуток задания переменной x :

$x := -10..10$



Теперь видно, что корень только один, и мы можем уточнить его расположение следующим образом:

$x := 0,0.1..4$



Из графика видно, что корень лежит внутри промежутка $[1; 2]$. Примем приближенное значение $x=1.5$ и определим корень с большей точностью:

$x := 1.5 \quad \text{root}(f(x), x) = 1.106$

Если требуется определить корень с еще большей точностью, то надо переопределить значение встроенной переменной **TOL** задающей точность вычислений, а затем снова воспользоваться функцией **root**. При этом потребуются также изменить значение формата результата в диалоговом окне команды **Format** → **Result** (**Формат** → **Результат**).

$\text{TOL} := 0.0001 \quad \text{root}(f(x), x) = 1.10606$

Вернитесь к точности, установленной по умолчанию: $\text{TOL} := 0.001$

Начиная с версии Mathcad 8, допустим еще один способ обращения к функции **root**, когда не требуется задавать начальное приближение для корня. Функция в этом случае имеет следующий формат:

$\text{root}(y(x), x, a, b)$

Здесь a и b — начало и конец интервала, на котором находится корень. Причем значения $f(a)$ и $f(b)$ должны иметь разный знак. Решение уравнения из примера можно оформить следующим образом:

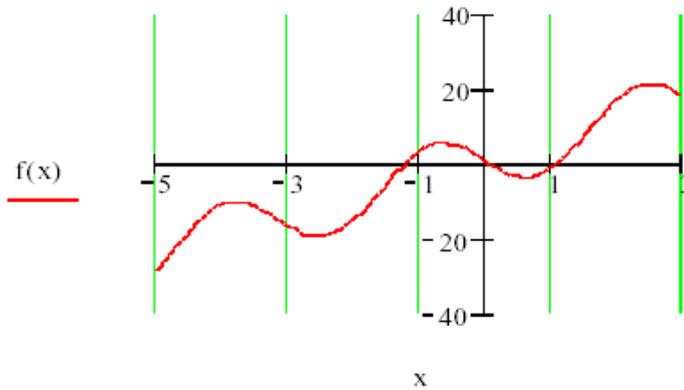
$\text{root}(f(x), x, -10, 10) = 1.106$

Если функция имеет несколько корней, то можно определить новую функцию, например $r(x)$, обращаясь к которой, можно найти корень, задавая его начальное приближение в качестве аргумента.

Пример 4.2. Найдите все корни уравнения $5x - 8\sin(2x) + 1 = 0$.

Решение: Определив функцию $f(x)$, построим ее график:

$$f(x) := 5x - 8\sin(2x) + 1 \quad x := -5, -4.9..3$$



Из графика видно, что функция имеет три корня, поэтому удобнее определить новую функцию $r(x)$:

$$r(x) := \text{root}(f(x), x)$$

Теперь найдем значения функции $r(x)$ для трех начальных приближений x_1 , x_2 и x_3 , определенных по графику:

$$\begin{aligned} x_1 &:= -1 & x_2 &:= 0 & x_3 &:= 1 \\ r(x_1) &= -1.224 & r(x_2) &= 0.092 & r(x_3) &= 1.098 \end{aligned}$$

Нахождение корней полинома. Для нахождения корней выражения, имеющего вид:

$$v_n x^n + \dots + v_2 x^2 + v_1 x + v_0,$$

лучше использовать функцию **polyroots**, нежели **root**. В отличие от функции **root**, функция **polyroots** не требует начального приближения. Кроме того, функция **polyroots** возвращает сразу все корни, как вещественные, так и комплексные.

Функция **polyroots** имеет следующий формат:
polyroots(v),

где $v = \begin{pmatrix} v_0 \\ v_1 \\ \dots \\ v_n \end{pmatrix}$ — вектор-столбец коэффициентов полинома, перечисленных в

направлении увеличения степени неизвестной, начиная с нулевой.

Функция **polyroots** всегда возвращает значения корней полинома, найденные численно. Чтобы найти корни аналитически (символьно), можно использовать функцию **Solve** на панели **Math** → **Symbolics** (**Математика** → **Символика**).

Численное решение системы уравнений. Для решения систем уравнений и неравенств (можно решить и одно уравнение/неравенство) используется конструкция, которая называется «**решающий блок**». Как и функция **root**, решающий блок использует для поиска корней итерационные

Информационные технологии в менеджменте

методы, поэтому необходимо задавать начальные приближения для значений переменных, относительно которых решается система.

Решающий блок состоит из нескольких компонент, следующих на листе в строго определенном порядке:

1. Начальное приближение (присваивание начальных значений искомым переменным).

2. Директива **Given**, которую необходимо набрать с клавиатуры.

3. Уравнения, которые необходимо решить. Уравнения вводятся в обычной математической форме, но вместо простого знака равенства используется оператор логического равенства (вводится путем нажатия $\text{Ctrl}+=$ или с панели инструментов **Math** → **Boolean (Математика** → **Булевое)**).

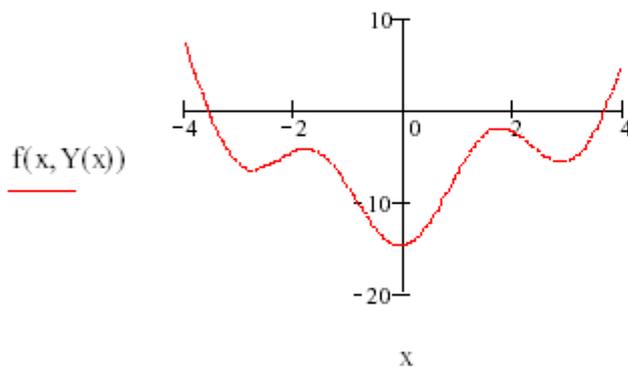
4. Обращение к функции **Find**. Аргументами функции являются имена переменных, относительно которых решается система. Функция **Find** возвращает вектор-столбец значений, где первый элемент соответствует первой переменной в списке аргументов, второй элемент — второй переменной и т. д.

Пример 4.4. Найти решение системы уравнений:

$$\begin{cases} x^2 + y^2 = 15, \\ y = 3 \sin x + 0.2 \end{cases}$$

Решение:

$$Y(x) := 3 \sin(x) + 0.2 \quad f(x, y) := x^2 + y^2 - 15 \quad x := -4, -3.9.. 4$$

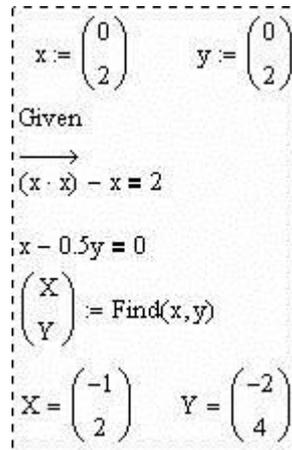


$$\begin{array}{l} x := -4 \quad y := Y(x) \\ \text{Given} \quad x^2 + y^2 - 15 = 0 \quad y = 3 \sin(x) + 0.2 \quad \text{Find}(x, y) = \begin{pmatrix} -3.581 \\ 1.476 \end{pmatrix} \end{array}$$

$$\begin{array}{l} x := 4 \quad y := Y(x) \\ \text{Given} \quad x^2 + y^2 - 15 = 0 \quad y = 3 \sin(x) + 0.2 \quad \text{Find}(x, y) = \begin{pmatrix} 3.656 \\ -1.277 \end{pmatrix} \end{array}$$

В Mathcad существует возможность *одновременно найти несколько решений*. Для этого система уравнений и начальные приближения должны быть переписаны в векторной форме (рис. 2). Каждая переменная будет вектором, содержащим столько компонент, сколько решений находится. Допустим, что в уравнении присутствует выражение $\mathbf{x}^* \mathbf{x}$. Если $\mathbf{x} = (x_1; x_2)$ — вектор, то $\mathbf{x} \cdot \mathbf{x} = x_1 \cdot x_1 + x_2 \cdot x_2$. Нам же необходим результат поэлементного перемножения

$(x_1 \cdot x_1; x_2 \cdot x_2)$. Для этого существует специальная операция, записываемая как $\overline{(x \cdot x)}$ (знак $\overrightarrow{f(M)}$ на панели инструментов **Math** → **Matrix** (**Математика** → **Матрица**)).



$$x := \begin{pmatrix} 0 \\ 2 \end{pmatrix} \quad y := \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

Given

$$\overrightarrow{(x \cdot x) - x} = 2$$

$$x - 0.5y = 0$$

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \text{Find}(x, y)$$

$$X = \begin{pmatrix} -1 \\ 2 \end{pmatrix} \quad Y = \begin{pmatrix} -2 \\ 4 \end{pmatrix}$$

Рис. 2. Пример одновременного поиска нескольких решений

Изменения коснулись и части получения результата. В данном случае функция **Find** вернет вектор из двух элементов, которые мы обозначили как *X* и *Y*. Каждый из этих элементов есть вектор значений *x* или *y* для решений. Соответственно первое решение — (-1; -2); второе решение — (2; 4).

Аналитическое решение уравнений и их систем. Аналитическое решение уравнений и их систем может быть получено с помощью функции **Solve** на панели **Math** → **Symbolics** (**Математика** → **Символика**).

Пример 1. Аналитическое решение уравнения:

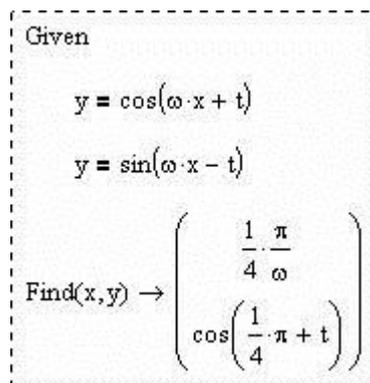
$$a \cdot x = x^2 + b \text{ solve, } x \rightarrow \begin{pmatrix} \frac{1}{2} \cdot a + \frac{1}{2} \cdot (a^2 - 4b)^{\frac{1}{2}} \\ \frac{1}{2} \cdot a - \frac{1}{2} \cdot (a^2 - 4b)^{\frac{1}{2}} \end{pmatrix}$$

Пример 2. Аналитическое решение системы уравнений:

$$\begin{pmatrix} a \cdot y = x + b \\ c \cdot x = y - d \end{pmatrix} \text{ solve, } x, y \rightarrow \begin{pmatrix} \frac{-(d \cdot a - b)}{(c \cdot a - 1)} & \frac{(c \cdot b - d)}{(c \cdot a - 1)} \end{pmatrix}$$

Для получения аналитического решения можно также применить **решающий блок Given/Find**. При этом уже нет необходимости указывать начальные приближения, а после функции **Find** используется оператор аналитического вычисления «→» вместо оператора числового вычисления «=».

Пример 3.



Given

$$y = \cos(\omega \cdot x + t)$$

$$y = \sin(\omega \cdot x - t)$$

$$\text{Find}(x, y) \rightarrow \begin{pmatrix} \frac{1}{4} \cdot \frac{\pi}{\omega} \\ \cos\left(\frac{1}{4} \cdot \pi + t\right) \end{pmatrix}$$

Задания

Задание 1. Решите уравнение $f(x) = 0$ аналитически с помощью:

- решающего блока **Given/Find**;
- функции **solve**.

При этом в уравнении замените все числовые коэффициенты буквенными. Сравните результаты.

Задание 2. Решите уравнение $f(x) = 0$ численно с помощью:

- функции **root**;
- решающего блока **Given/Find**.

Сравните результаты.

№	уравнение	№	уравнение
1	$2^x + 5x - 3 = 0; 0,5^x + 1 = (x - 2)^2;$ $x - \sin x = 0,25; \ln x + (x + 1)^3 = 0;$	16	$\operatorname{arctg} x - \frac{1}{3x^3} = 0; (x - 1)^2 \cdot 2^x = 1;$ $\operatorname{tg}^3 x = x - 1, x \cdot 2^x = 1;$
2	$x + \cos x = 1; [\log(-x)] \cdot (x + 2) = -1;$ $\operatorname{tg}(0,58 + 0,1) = x^2; \sqrt{x + 1} = \frac{1}{x};$	17	$2\operatorname{arctg} x - x + 3 = 0; x - \cos x = 0;$ $2 \lg x - \frac{x}{2} + 1 = 0. x^2 - 20 \sin x = 0;$
3	$5^x + 3x = 0; x^2 - 2 + 0,5^x = 0;$ $\sqrt{x} - \cos(0,387x) = 0; 3x + \cos x + 1 = 0;$	18	$3^x - 2x + 5 = 0; 2x^2 - 0,5^x - 2 = 0;$ $x \lg(x + 1) = 1. \operatorname{ctg} x - \frac{x}{3} = 0;$
4	$2e^x \cdot 5x = 2; x \log_3(x + 1) = 1;$ $\cos(x + 0,5) = x^3. 2 - x = \ln x;$	19	$(x - 2)^2 \cdot 2^x = 1; x^2 - 20 \sin x = 0;$ $\operatorname{tg}(0,47x + 0,2) = x^2; (x - 1)^2 = \frac{1}{2}e^x;$
5	$3^{x-1} - 2 - x = 0; 3x + \cos x + 1 = 0;$ $5 \sin x = x. \operatorname{tg} x - \frac{7}{2x + 6} = 0;$	20	$e^x + x + 1 = 0; 0,5^x - 3 = (x + 2)^2;$ $x^2 + 4 \sin x = 0; x^2 = \ln(x + 1);$
6	$2\operatorname{arctg} x - \frac{1}{2x^3} = 0; x^2 \cdot 2^x = 1;$ $\operatorname{tg}(0,5x + 0,2) = x^2; (2 - x)e^x = 0,5;$	21	$2^x - 3x - 2 = 0; (0,5)^x + 1 = (x - 2)^2;$ $\operatorname{ctg} x - \frac{x}{2} = 0; 2,2x - 2^x = 0;$

Информационные технологии в менеджменте

7	$e^{-2x} - 2x + 1 = 0; 0,5^x - 1 = (x + 2)^2;$ $x^2 \cos 2x = -1. 3x - \cos x - 1 = 0;$	22	$\arctg x + 2x - 1 = 0; (x + 2) \log_2(x) = 1;$ $\sin(x + 1) = 0,5x. 2x - \lg x - 7 = 0;$
8	$5^x - 6x - 3 = 0; 2x^2 - 0,5^x - 3 = 0;$ $x \lg(x + 1) = 1. 5x - 8 \ln x = 8;$	23	$x^2 - 4 + 0,5^x = 0; (x - 2)^2 \lg(x + 11) = 1.$ $\operatorname{tg}(0,44x + 0,3) = x^2; 3x - e^x = 0;$
9	$\arctg(x - 1) + 2x = 0; (x - 2)^2 \cdot 2^x = 1;$ $\operatorname{tg}(0,5x + 0,1) = x^2; x(x + 1)^2 = 1;$	24	$2e^x - 2x - 3 = 0; x \log_3(x + 1) = 1;$ $\cos(x + 0,5) = x^3. 3x - \cos x - 1 = 0;$
10	$2 \arctg x - x + 3 = 0; x^3 = \sin x;$ $2 \lg x - \frac{x}{2} + 1 = 0. x^2 + 4 \sin x = 0;$	25	$3^x + 2 + x = 0; x = \sqrt{\lg(x + 2)};$ $5 \sin x = x - 0,5. \operatorname{ctg} x - \frac{x}{10} = 0;$
11	$3^x + 2x - 2 = 0; [(x - 2)^2 - 1]2^x = 1;$ $\operatorname{ctg} 1,05x - x^2 = 0; x^2 = \ln(x + 1);$	26	$\arctg(x - 1) + 2x - 3 = 0; (x - 1)^2 2^x = 1;$ $x^2 + 4 \sin x = 0; 2x + \lg x = -0,5;$
12	$2 \arctg x - 3x + 2 = 0; [\log_2(x + 2)](x - 1) = 1;$ $\operatorname{tg}(0,4x + 0,3) = x^2; 2x + \cos x + 0,5;$	27	$e^{-2x} - 2x + 1 = 0; 0,5^x - 3 = -(x + 1)^2;$ $x^2 \cos 2x = -1. \operatorname{tg}(0,36x + 0,4) = x^2;$
13	$3^x + 2x - 5 = 0; x^2 - 3 + 0,5^x = 0;$ $(x - 2)^2 \lg(x + 11) = 1. x \lg x - 1,2 = 0;$	28	$3^x - 2x - 5 = 0; 2x^2 - 0,5^x - 3 = 0;$ $x \lg(x + 1) = 1. \sin(0,5 + x) = 2x - 0,5;$
14	$2e^x + 3x + 1 = 0; x \log_3(x + 1) = 2;$ $\cos(x + 0,3) = x^2. 1,8x^2 - \sin 10x = 0;$	29	$\arctg(x - 1) + 2x = 0; (x - 2)^2 2^x = 1;$ $x^2 - 10 \sin x = 0. \lg(1 + 2x) = 2 - x;$
15	$3^{x-1} - 4 - x = 0; (x - 3)^2 \log_{0,5}(x - 2) = -1;$ $5 \sin x = x - 1. \operatorname{ctg} x - \frac{x}{4} = 0;$	30	$3^x + 5x - 2 = 0; 0,5^x + 1 = (x - 2)^2;$ $21 \lg x - \frac{x}{2} + 1 = 0; x + \lg(1 + x) = 1,5;$

Задание 3. Решите систему уравнений аналитически с помощью:

— решающего блока **Given/Find**;

— функции **solve**.

При этом в уравнениях замените все числовые коэффициенты буквенными.

Сравните результаты.

Задание 4. Решите систему уравнений численно с помощью решающего блока **Given/Find**:

Информационные технологии в менеджменте

№	Система уравнений	№	Система уравнений
1	$\begin{cases} \sin(x+1) - y = 1,2; \\ 2x + \cos y = 2. \end{cases} \begin{cases} \operatorname{tg}(xy+0,4) = x^2; \\ 0,6x^2 + 2y^2 = 1, x > 0, y > 0. \end{cases}$	16	$\begin{cases} \cos(x-1) + y = 0,5; \\ x - \cos y = 3. \end{cases} \begin{cases} \sin(x+y) - 1,6x = 0; \\ x^2 + y^2 = 1, x > 0, y > 0. \end{cases}$
2	$\begin{cases} \cos(x-1) + y = 0,5; \\ x - \cos y = 3. \end{cases} \begin{cases} \sin(x+y) - 1,6x = 0; \\ x^2 + y^2 = 1, x > 0, y > 0. \end{cases}$	17	$\begin{cases} \sin x + 2y = 2; \\ \cos(y-1) + x = 0,7; \end{cases} \begin{cases} \operatorname{tg}(xy+0,1) = x^2; \\ x^2 + 2y^2 = 1. \end{cases}$
3	$\begin{cases} \sin x + 2y = 2; \\ \cos(y-1) + x = 0,7; \end{cases} \begin{cases} \operatorname{tg}(xy+0,1) = x^2; \\ x^2 + 2y^2 = 1. \end{cases}$	18	$\begin{cases} \cos x + y = 1,5; \\ 2x - \sin(y-0,5) = 1. \end{cases} \begin{cases} \sin(x+y) - 1,2x = 0,2; \\ x^2 + y^2 = 1. \end{cases}$
4	$\begin{cases} \sin(x+0,5) - y = 1; \\ \cos(y-2) + x = 0. \end{cases} \begin{cases} \operatorname{tg}(xy+0,3) = x^2; \\ 0,9x^2 + 2y^2 = 1. \end{cases}$	19	$\begin{cases} \sin(y+1) - x = 1,2; \\ 2y + \cos x = 2. \end{cases} \begin{cases} \operatorname{tg}(xy+0,2) = x^2; \\ 0,6x^2 + 2y^2 = 1. \end{cases}$
5	$\begin{cases} \sin(x+2) - y = 1,5; \\ x + \cos(y-2) = 0,5. \end{cases} \begin{cases} \sin(x+y) - 1,2x = 0,1; \\ x^2 + y^2 = 1. \end{cases}$	20-	$\begin{cases} 2y - \cos(x+1) = 0; \\ x + \sin y + 0,4. \end{cases} \begin{cases} \sin(x+y) - 1,5x = 0,1; \\ x^2 + y^2 = 1. \end{cases}$
6	$\begin{cases} \cos(x+0,5) - y = 2; \\ \sin y - 2x = 1. \end{cases} \begin{cases} \operatorname{tg}xy = x^2; \\ 0,7x^2 + 2y^2 = 1. \end{cases}$	21	$\begin{cases} \sin(x-1) = 1,3 - y; \\ x - \sin(y+1) = 0. \end{cases} \begin{cases} \operatorname{tg}xy = x^2; \\ 0,8x^2 + 2y^2 = 1. \end{cases}$
7	$\begin{cases} \cos(x+0,5) - y = 1; \\ \sin y - 2x = 1,6. \end{cases} \begin{cases} \sin(x+y) - 1,3x = 0; \\ x^2 + y^2 = 1. \end{cases}$	22	$\begin{cases} \cos(y-1) + x = 0,5; \\ y - \cos x = 3. \end{cases} \begin{cases} \sin(x+y) + 1,5x - 0,1; \\ x^2 + y^2 = 1. \end{cases}$
8	$\begin{cases} \sin y + 2x = 2; \\ \cos(x-1) + y = 0,7. \end{cases} \begin{cases} \operatorname{tg}(xy+0,4) = x^2; \\ 0,8x^2 + 2y^2 = 1. \end{cases}$	23	$\begin{cases} \cos y + x + 1,5; \\ 2y - \sin(x-0,5) = 1. \end{cases} \begin{cases} \sin(x+y) = 1,2x - 0,1; \\ x^2 + y^2 = 1. \end{cases}$
9	$\begin{cases} \sin(y+0,5) - x = 1; \\ \cos(x-2) + y = 0. \end{cases} \begin{cases} \operatorname{tg}(xy+0,1) = x^2; \\ 0,9x^2 + 2y^2 = 1. \end{cases}$	24	$\begin{cases} \sin(y-1) + x = 1,3; \\ y - \sin(x+1) = 0,8. \end{cases} \begin{cases} \operatorname{tg}(xy+0,1) = x^2; \\ 0,5x^2 + 2y^2 = 1. \end{cases}$
10	$\begin{cases} \cos(y+0,5) + x = 0,8; \\ \sin x - 2y = 1,6. \end{cases} \begin{cases} \sin(x+y) - 1,4x = 0; \\ x^2 + y^2 = 1. \end{cases}$	25	$\begin{cases} 2x - \cos(y+1) = 0; \\ y + \sin x + 0,4. \end{cases} \begin{cases} \sin(x+y) = 1,1x - 0,1; \\ x^2 + y^2 = 1. \end{cases}$
11	$\begin{cases} \cos(y+0,5) - x = 2; \\ \sin x - 2y = 1. \end{cases} \begin{cases} \operatorname{tg}(x-y) - xy = 0; \\ x^2 + 2y^2 = 1. \end{cases}$	26	$\begin{cases} \sin(y+2) - x = 1,5; \\ y + \cos(x-2) = 0,5. \end{cases} \begin{cases} \sin(x-y) - xy = -1; \\ x^2 - y^2 = \frac{3}{4}. \end{cases}$
12	$\begin{cases} \sin(x+1) - y = 1; \\ 2x + \cos y = 2. \end{cases} \begin{cases} \operatorname{tg}(xy+0,2) = x^2; \\ x^2 + 2y^2 = 1. \end{cases}$	27	$\begin{cases} \cos(x-1) + y = 0,8; \\ x - \cos y + 2. \end{cases} \begin{cases} \sin(x+y) - 1,5x = 0; \\ x^2 + y^2 = 1. \end{cases}$
13	$\begin{cases} \sin x + 2y + 1,6; \\ \cos(y-1) + x = 1. \end{cases} \begin{cases} \operatorname{tg}xy = x^2; \\ 0,5x^2 + 2y^2 = 1. \end{cases}$	28	$\begin{cases} \cos x + y = 1,2; \\ 2x - \sin(y-0,5) = 2. \end{cases} \begin{cases} \sin(x+y) = 1,2x - 0,2; \\ x^2 + y^2 = 1. \end{cases}$
14	$\begin{cases} \sin(x+0,5) - y = 1,2; \\ \cos(y-2) + x = 0. \end{cases} \begin{cases} \operatorname{tg}(xy+0,1) = x^2; \\ 0,7x^2 + 2y^2 = 1. \end{cases}$	29	$\begin{cases} \cos(x+0,5) + y = 1; \\ \sin y - 2x = 2. \end{cases} \begin{cases} \sin(x+y) - 1,5x + 0,2; \\ x^2 + y^2 = 1. \end{cases}$
15	$\begin{cases} \sin(x-1) + y = 1,5; \\ x - \sin(y+1) = 1. \end{cases} \begin{cases} \operatorname{tg}xy = x^2; \\ 0,6x^2 + 2y^2 = 1. \end{cases}$	30	$\begin{cases} \sin(y+1) - x + 1; \\ 2y + \cos x = 2. \end{cases} \begin{cases} \sin(x+y) - 1,2x = 0; \\ x^2 + y^2 = 1. \end{cases}$

Задание 5. Найдите корни полинома:

№	уравнение	№	уравнение
1	$x^4 - x - 1 = 0; \quad x^3 + x - 3 = 0.$	16	$x^4 - x - 1 = 0; \quad x^3 - 3x^2 + 9x - 10 = 0.$
2	$2x^4 - 9x^2 - 60x + 1 = 0; \quad x^3 - 2x + 2 = 0.$	17	$x^4 + 4x^3 - 8x^2 - 17 = 0; \quad x^3 + 3x - 1 = 0.$
3	$3x^4 + 4x^3 - 12x^2 - 5 = 0; \quad x^3 + 2x^2 + 2 = 0.$	18	$3x^4 + 8x^3 + 6x^2 - 10 = 0; \quad x^3 + 0,4x^2 + 0,6x - 1,6 = 0.$
4	$2x^4 - x^2 - 10 = 0; \quad x^3 - 0,2x^2 + 0,4x - 1,4 = 0.$	19	$x^4 - 18x^2 + 6 = 0; \quad x^3 - 0,1x^2 + 0,4x + 2 = 0.$
5	$3x^4 + 8x^3 + 6x^2 - 10 = 0; \quad x^3 + 3x^2 + 12x + 3 = 0.$	20	$2x^4 - x^2 - 10 = 0; \quad x^3 - 0,2x^2 + 0,5x - 1 = 0.$
6	$x^4 - 18x^2 + 6 = 0; \quad x^3 - 0,1x^2 + 0,4x + 1,2 = 0.$	21	$x^4 - x^3 - 2x^2 + 3x - 3 = 0; \quad x^3 - 3x^2 + 6x - 5 = 0.$
7	$x^4 + 4x^3 - 8x^2 - 17 = 0; \quad x^3 - 0,2x^2 + 0,5x - 1,4 = 0.$	22	$3x^4 + 4x^3 - 12x^2 + 1 = 0; \quad x^3 + 0,2x^2 + 0,5x + 0,8 = 0.$
8	$x^4 - x^3 - 2x^2 + 3x - 3 = 0; \quad x^3 + 4x - 6 = 0.$	23	$3x^4 - 8x^3 - 18x^2 + 2 = 0; \quad x^3 + 0,1x^2 + 0,4x - 1,2 = 0.$
9	$3x^4 + 4x^3 - 12x^2 + 1 = 0; \quad x^3 + 3x^2 + 6x - 1 = 0.$	24	$3x^4 + 4x^3 - 12x^2 - 5 = 0; \quad x^3 - 0,1x^2 + 0,4x - 1,5 = 0.$
10	$3x^4 - 8x^3 - 18x^2 + 2 = 0; \quad x^3 - 3x^2 + 6x - 2 = 0.$	25	$2x^3 - 9x^2 - 60x + 1 = 0; \quad x^3 + 2x + 4 = 0.$
11	$2x^4 - 8x^3 + 8x^2 - 1 = 0; \quad x^3 - 3x^2 + 12x - 12 = 0.$	26	$x^4 - x - 1 = 0; \quad x^3 - 0,2x^2 + 0,3x - 1,2 = 0.$
12	$2x^4 + 8x^3 + 8x^2 - 1 = 0; \quad x^3 - 3x^2 + 12x - 9 = 0.$	27	$2x^4 - x^2 - 10 = 0; \quad x^3 + 0,2x^2 + 0,5x - 2 = 0.$
13	$x^4 - 4x^3 - 8x^2 + 1 = 0; \quad x^3 + 3x + 1 = 0.$	28	$3x^4 + 8x^3 + 10 = 0; \quad x^3 + 0,2x^2 + 0,5x - 1,2 = 0.$
14	$3x^4 + 4x^3 - 12x^2 - 5 = 0; \quad x^3 - 3x^2 + 9x + 2 = 0.$	29	$x^3 + 0,1x^2 + 0,4x - 1,2 = 0. \quad x^4 - 18x^2 + 6 = 0;$
15	$2x^3 - 9x^2 - 60x + 1 = 0; \quad x^3 - 3x^2 + 6x + 3 = 0.$	30	$3x^4 + 4x^3 - 12x^2 + 1 = 0; \quad x^3 - 0,1x^2 + 0,3x - 0,6 = 0.$

Лабораторная работа

Mathcad: Решение дифференциальных уравнений и их систем

Цель работы: получение навыков решения дифференциальных уравнений и их систем в пакете Mathcad.

Теоретические сведения

Вычислительный процессор Mathcad может работать с дифференциальными уравнениями, представленными только в форме Коши, т. е. в виде:

$$y'(t) = f(y(t), t).$$

Задача Коши — одна из основных задач теории дифференциальных уравнений (обыкновенных и с частными производными); состоит в нахождении решения (интеграла) дифференциального уравнения, удовлетворяющего начальным условиям.

Задача Коши обычно возникает при анализе процессов, определяемых дифференциальным законом эволюции и начальным состоянием (математическим выражением которых и являются уравнение и начальное условие). Этим мотивируется терминология и выбор обозначений: начальные данные задаются при $t = 0$, а решение отыскивается при $t > 0$.

От краевых задач задача Коши отличается тем, что область, в которой должно быть определено искомое решение, здесь заранее не указывается. Тем не менее, задачу Коши можно рассматривать как одну из краевых задач.

Информационные технологии в менеджменте

Основные вопросы, которые связаны с задачей Коши, таковы:

5. Существует ли (хотя бы локально) решение задачи Коши?
6. Если решение существует, то какова область его существования?
7. Является ли решение единственным?
8. Если решение единственно, то будет ли оно корректным, то есть непрерывным (в каком-либо смысле) относительно начальных данных?

У пользователя имеется выбор — либо использовать вычислительный блок **Given/Odesolve**, либо встроенные функции.

Для решения дифференциальных уравнений при помощи блока Given/Odesolve:

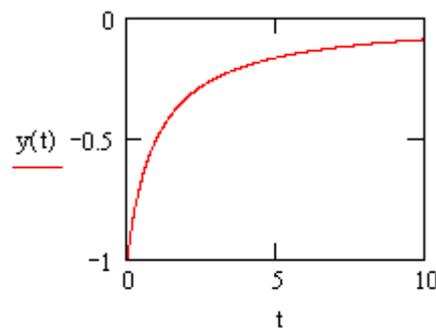
- ввести ключевое слово **Given**;
- ввести дифференциальное уравнение в форме Коши, записанное с помощью логических операторов;
- ввести начальное условие в форме $y(t_0) = b$;
- решить уравнение с помощью функции **Odesolve(t, t₁, step)**, где t — переменная дифференцирования в дифференциальном уравнении; t_1 — правая граница интервала, на котором производится поиск решения; step — необязательный параметр, определяющий количество шагов метода Рунге-Кутты, который применяется при решении. Чем больше step, тем больше точность решения, тем больше времени будет затрачено на его поиск. По умолчанию в методе применяется фиксированный шаг (Fixed). Для замены его на адаптивный (Adaptive) необходимо вызвать контекстное меню над функцией **Odesolve**.

При использовании блока **Given/Odesolve** для решения **дифференциального уравнения высшего порядка** необходимо кроме начального значения самой функции, задать начальные значения для всех ее производных, входящих в уравнение, причем производные должны быть записаны при помощи обычного обозначения производной (один, два и т. д. штриха справа от имени функции, Ctrl+F7).

Пример 1. Решение дифференциального уравнения первого порядка

$y' = y^2$ при помощи блока **Given/Odesolve**:

```
Given
d
— y(t) = y(t)2
dt
y(0) = -1
y := Odesolve(t,10)
```



Пример 2. Решение дифференциального уравнения 3-го порядка при помощи блока **Given/Odesolve**:

$$y'''(t) + 4y''(t) + 3y'(t) + y(t) = 0;$$

$$y(0) = 50, y'(0) = 10, y''(0) = 0.$$

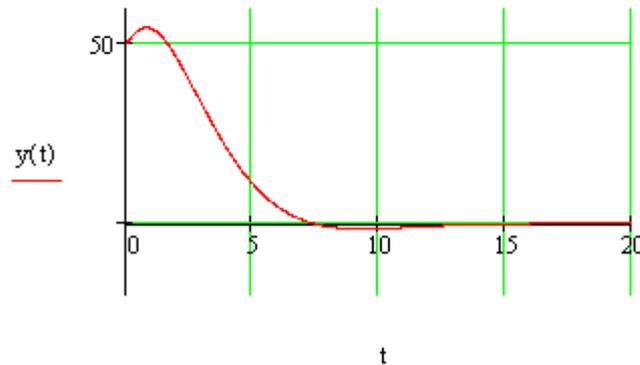
Информационные технологии в менеджменте

Given

$$\frac{d^3}{dt^3}y(t) + 4 \frac{d^2}{dt^2}y(t) + 3 \frac{d^1}{dt}y(t) + y(t) = 0$$

$$y(0) = 50 \quad y'(0) = 10 \quad y''(0) = 0$$

`y := odesolve(t,20)`



Для решения дифференциального уравнения или системы дифференциальных уравнений при помощи встроенных функций необходимо:

- переменной y_0 присвоить начальное значение либо вектор начальных значений в случае решения системы;
- функции $D(t, y)$ присвоить функцию из правой части уравнения в форме Коши (либо вектор-функцию при решении системы);
- решить уравнение (или систему) при помощи одной из функций:
rkfixed (y_0, t_0, t_1, M, D) — метод Рунге-Кутты с фиксированным шагом,
rkadapt (y_0, t_0, t_1, M, D) — метод Рунге-Кутты с переменным шагом,
bulstoer (y_0, t_0, t_1, M, D) — метод Булирша-Штера,

где y_0 — вектор начальных значений в точке t_0 ; t_0 — начальная точка расчета; t_1 — конечная точка расчета; M — количество точек интегрирования (число шагов); D — векторная функция размера $n \times 1$ (где n — порядок решаемого дифференциального уравнения или количество уравнений 1-го порядка в системе уравнений) двух аргументов — скалярного t и векторного y (для одного дифференциального уравнения — просто функция).

Результатом будет матрица, 0-й столбец которой представляет собой значения аргумента t , 1-ый — соответствующие значения искомой функции; 2-ой — значения первой производной искомой функции и т.д. до $n-1$ производной. Количество строк матрицы равно числу шагов M .

Пример 3. Решение дифференциального уравнения первого порядка $y' = y^2$ при помощи встроенной функции **rkfixed**:

Информационные технологии в менеджменте

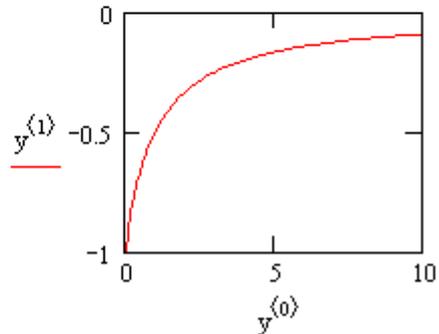
$$y_0 := -1$$

$$D(t,y) := y^2$$

$$M := 100$$

$$y := \text{rkfixed}(y_0, 0, 10, M, D)$$

	0	1
0	0	-1
1	0.1	-0.909
2	0.2	-0.833
3	0.3	-0.769
4	0.4	-0.714
5	0.5	-0.667
6	0.6	-0.625



Пример 4. Решение дифференциального уравнения 3-го порядка при помощи встроенной функции **rkfixed**:

$$y'''(t) + 4y''(t) + 3y'(t) + y(t) = 0;$$

$$y(0) = 50, y'(0) = 10, y''(0) = 0.$$

При этом необходимо преобразовать дифференциальное уравнение третьего порядка в систему трех дифференциальных уравнений первого порядка (переход к переменным состояния). Для этого введем так называемые *переменные состояния* y_0, y_1, y_2, y_3 :

$$y_0 = y, y_1 = y', y_2 = y'', y_3 = y''''.$$

Тогда можно записать:

$$\begin{cases} y_0' = y_1 = y_1 \\ y_1' = y_2 = y_2 \\ y_2' = y_3 = y_3 = -4y_2 - 3y_1 - y_0 \end{cases}$$

Информационные технологии в менеджменте

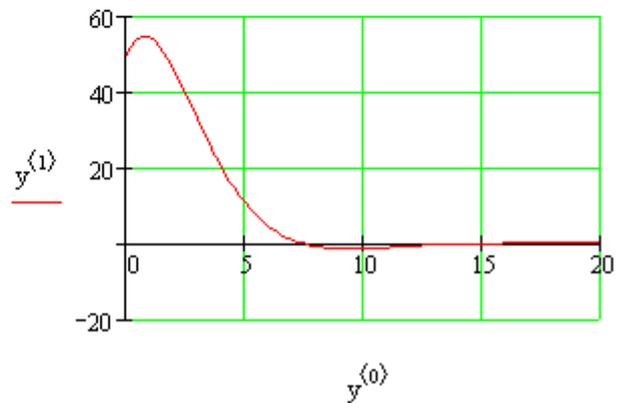
$$y_0 := \begin{pmatrix} 50 \\ 10 \\ 0 \end{pmatrix}$$

$$D(t, y) := \begin{pmatrix} y_1 \\ y_2 \\ -y_0 - 3 \cdot y_1 - 4 \cdot y_2 \end{pmatrix}$$

$$M := 100$$

$$y := \text{rkfixed}(y_0, 0, 20, M, D)$$

	0	1	2	3
0	0	50	10	0
1	0.2	51.914	8.747	-10.923
2	0.4	53.41	6.053	-15.177
3	0.6	54.308	2.892	-16.009
4	0.8	54.57	-0.244	-15.149
5	1	54.229	-3.119	-13.514
6	1.2	53.347	-5.632	-11.585
7	1.4	52.002	-7.751	-9.614
8	1.6	50.273	-9.484	-7.729



Пример 5. Решение системы двух дифференциальных уравнений при помощи встроенной функции **rkfixed**:

$$\begin{cases} \frac{dy_0}{dt} = y_1; \\ \frac{dy_1}{dt} = -y_0 - 0,1y_1; \end{cases}$$

$$y_0(0) = 0,1, \quad y_1(0) = 0.$$

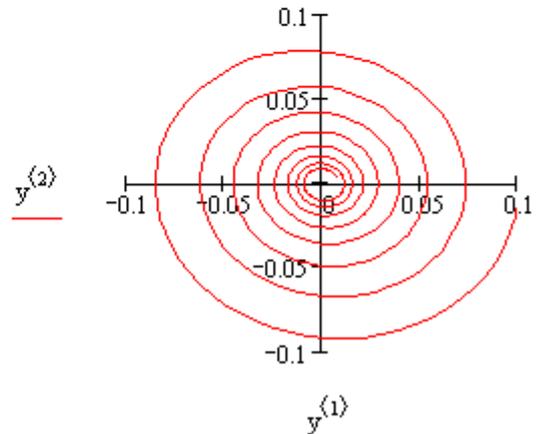
$$D(t, y) := \begin{pmatrix} y_1 \\ -y_0 - 0.1 \cdot y_1 \end{pmatrix}$$

$$y_0 := \begin{pmatrix} 0.1 \\ 0 \end{pmatrix}$$

$$M := 500$$

$$y := \text{rkfixed}(y_0, 0, 50, M, D)$$

	0	1	2
0	0	0.1	0
1	0.1	0.1	-9.934·10 ⁻³
2	0.2	0.098	-0.02
3	0.3	0.096	-0.029
4	0.4	0.092	-0.038
5	0.5	0.088	-0.047
6	0.6	0.083	-0.055
7	0.7	0.077	-0.062



Задания

Задание 1. Решите дифференциальное уравнение первого порядка $y'(t) = f(t, y(t))$. Решение найдите двумя способами: при помощи блока **Given/Odesolve** и встроенной функции **rkfixed**.

Постройте графики решений.

№	$f(t, y(t))$	t_0	t_1	y_0
1.	$y/t + t^2$	1	2	0
2.	$yctgt + 2t \sin t$	$\frac{\pi}{2}$	$\frac{\pi}{2} + 1$	0
3.	$-y \cos t + \frac{\sin(2t)}{2}$	0	1	0
4.	$-ytgt + \cos^2 t$	$\frac{\pi}{4}$	$\frac{\pi}{4} + 1$	0,5
5.	$\frac{y}{t+2} + t^2 + 2t$	-1	0	1,5
6.	$\frac{y}{t+1} + e^t(t+1)$	0	1	1
7.	$y/t + t \sin t$	$\frac{\pi}{2}$	$\frac{\pi}{2} + 1$	1
8.	$-y/t + \sin t$	π	$\pi + 1$	$\frac{1}{\pi}$

Информационные технологии в менеджменте

9.	$-\frac{y}{2t} + t^2$	1	2	1
10.	$-\frac{2t}{1+t^2}y + \frac{2t^2}{1+t^2}$	0	1	$\frac{2}{3}$
11.	$\frac{2t-5}{t^2}y + 5$	2	3	4
12.	$-y/t + \frac{t+1}{t}e^t$	1	2	e
13.	$y/t - 2\ln t/t$	1	2	1
14.	$y/t - 12/t^3$	1	2	4
15.	$-2y/t + t^3$	1	2	$-\frac{5}{6}$
16.	$-y/t + 3t$	1	2	1
17.	$\frac{2ty}{1+t^2} + 1 + t^2$	1	2	3
18.	$\frac{2t-1}{t^2}y + 1$	1	2	1
19.	$-\frac{3y}{t} + \frac{2}{t^3}$	1	2	1
20.	$-2ty - 2t^3$	1	2	e^{-1}
21.	$y/t - 2/t^2$	1	1	1
22.	$-ty - t^3$	0	1	3
23.	$\frac{2}{t+1}y + e^t(t+1)^2$	0	1	1
24.	$-2ty + te^{-t^2} \sin t$	0	1	1
25.	$\frac{2y}{t+1} + (t+1)^3$	0	1	0,5
26.	$y \cos t - \sin 2t$	0	1	3
27.	$4ty - 4t^3$	0	1	-0,5
28.	$y/t - \ln t/t$	1	2	1
29.	$3t^2y + t^2(1+t^3)/3$	0	1	0
30.	$y \cos t + \sin 2t$	0	1	-1

Задание 2. Решите дифференциальное уравнение третьего порядка на интервале $t = [A, B]$:

$$a_0y''' + a_1y'' + a_2y' + a_3y = f(t)$$

$$y(A) = b_1, y'(A) = b_2, y''(A) = b_3$$

Информационные технологии в менеджменте

Решение найдите двумя способами: при помощи блока **Given/Odesolve** и встроенной функции **rkfixed**.

Постройте графики решений.

№	A	B	b ₁	b ₂	b ₃	a ₀	a ₁	a ₂	a ₃	f(t)
1, 11, 21	0	1,5	1	2,5	6	1	-2	0,25	45,75	$e^{-2t} + 3t + 1$
2, 12, 22	0	1,5	1	2,0	4	1	-1,8	0,36	44,28	$e^{-2t} - 1.5t + 1$
3, 13, 23	0	1,5	1	2,5	6	1	-1,4	0,64	41,52	$\cos(2t) + 3t + 1$
4, 14, 24	0	2,0	1	1,5	2	1	-1,4	1,88	45,24	$\sin(2t) + 2t - 1$
5, 15, 25	0	1,5	1	3,0	10	1	-2,4	0,09	48,87	$\sin(t) - 7t + 2$
6, 16, 26	0	1,0	1	3,5	9	1	-1	8,8	29,00	$\cos(t) + 5t + 3$
7, 17, 27	0	1,5	1	2,8	5	1	-1,5	- 1,25	53,375	$e^{-t} + \cos(2t)$
8, 18, 28	0	1,5	1,5	4,0	10	1	-4,6	3,94	34,28	$e^{-1.5t} + 2\sin(3t)$
9, 19, 29	0	1,5	0	2,5	8	1	-4,1	0,64	42,85	$e^{-2t} + 3\sin(2.5t)$
10, 20, 30	0	1,5	0	3,1	9	1	-3,9	9,43	26,295	$\sin(2t) + 2\cos(3t)$

Задание 3. При помощи встроенной функции **rkfixed** решите систему дифференциальных уравнений. Постройте графики решений.

№	Система дифференциальных уравнений	№	Система дифференциальных уравнений
1.	$\begin{cases} x' = -x + 4y + e^t \\ y' = x + 2y \end{cases}$	2.	$\begin{cases} x' = 2\sqrt{1+x} - 2e^y, \\ y' = \cos x - 2y - 1. \end{cases}$
3.	$\begin{cases} x' = -4x - 5y + 3e^t \\ y' = x + 2y \end{cases}$	4.	$\begin{cases} x' = y - 4, \\ y' = -2y - x^2 + 3x. \end{cases}$
5.	$\begin{cases} x' = x - y + \frac{1}{\cos t}, \\ y' = 2x - y. \end{cases}$	6.	$\begin{cases} x' = 2x + 2y - 2e^t, \\ y' = x + 3y + e^t \end{cases}$
7.	$\begin{cases} x' = x - 3y + x \sin y, \\ y' = x + 4y + 1 - \cos y. \end{cases}$	8.	$\begin{cases} x' = -x + y + 4t^2 e^{t^2}, \\ y' = x + y. \end{cases}$

Информационные технологии в менеджменте

9.	$\begin{cases} x' = \frac{x}{x+y}, \\ y' = \frac{y}{x+y}, \end{cases}$	10.	$\begin{cases} x' = y, \\ y' = -y - x^3 + 4x^2 - 3x. \end{cases}$
11.	$\begin{cases} x' = x + 3y, \\ y' = -2x - y \end{cases}$	12.	$\begin{cases} x' = 3x - 4y + x^2, \\ y' = -2x + 2y - \sin y. \end{cases}$
13.	$\begin{cases} x' = -2x + 8\sin y, \\ y' = -x - 3y + 4x^3. \end{cases}$	14.	$\begin{cases} x' = y, \\ y' = 0,4y - x^3 + 9x. \end{cases}$
15.	$\begin{cases} x' = y + 1, \\ y' = -y + x^2 - x. \end{cases}$	16.	$\begin{cases} x' = 3x - 4y + t \\ y' = 2x - 3y \end{cases}$
17.	$\begin{cases} x' = -x - 6y, \\ y' = x + 4y + \cos t \end{cases}$	18.	$\begin{cases} x' = -2x - 2y, \\ y' = 3x - 10y \end{cases}$
19.	$\begin{cases} x' = x^2 + xy, \\ y' = xy + y^2, \end{cases}$	20.	$\begin{cases} x' = 2x - 5y + 4\sin t, \\ y' = x - 2y. \end{cases}$
21.	$\begin{cases} x' = 2x - 2y + e^t \sin(e^t), \\ y' = x - y + e^t \sin(e^t). \end{cases}$	22.	$\begin{cases} x' = y - 1 + e^x, \\ y' = -x + \sin 2y. \end{cases}$
23.	$\begin{cases} x' = -2x - 8\sin y, \\ y' = x - 3y + 4x^3 + x^5. \end{cases}$	24.	$\begin{cases} x' = y, \\ y' = -4y + x^3 - 4x^2 - 5x. \end{cases}$
25.	$\begin{cases} x' = y, \\ y' = -2y - x(x^2 - 1). \end{cases}$	26.	$\begin{cases} x' = 3x + y - \sin 2t \\ y' = x + 3y \end{cases}$
27.	$\begin{cases} x' = -x - 8y + \cos 2t, \\ y' = x + 5y \end{cases}$	28.	$\begin{cases} x' = 3x - 2y, \\ y' = 2x - y + 15e^t \sqrt{t}. \end{cases}$

29.	$\begin{cases} x' = 4x - 3y + 2,5e^{3t}, \\ y' = 2x - y + e^{-3t}. \end{cases}$	30.	$\begin{cases} x' = 3\sqrt{1+x} - 3e^y, \\ y' = -2\sin x + 0,5y. \end{cases}$
-----	---	-----	---

Лабораторная работа

Mathcad: Элементы программирования

Цель работы: получение навыков программирования в пакете Mathcad.

Теоретические сведения

На одном листе Mathcad могут определяться один или несколько программных блоков. Обычно их используют при разработке функций, которые осуществляют какую-либо сложную обработку данных.

В программном блоке можно читать значения переменных, определенных в Mathcad до этого блока. Однако изменить значения этих переменных внутри программного блока невозможно. Все переменные, которым присваиваются значения внутри программного блока, будут локальными переменными, которые недоступны вне блока.

Программный блок представляет собой группу операторов присваивания и управляющих операторов. Необходимо обратить особое внимание, что все ключевые слова (например, **if**) в этих операторах обязательно вводятся с помощью панели **Programming (Программирование)** (рис. 1) включаемой кнопкой  на панели **Math**. Их ввод с клавиатуры — ошибка.

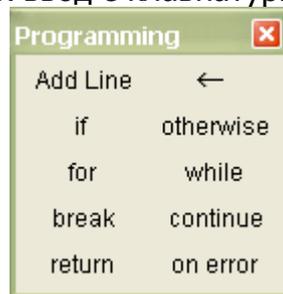


Рис. 1.

В табл. 1 приведены управляющие операторы панели **Programming**.

Управляющие операторы панели Programming

Оператор Mathcad	Комментарий
Add line	Команда вычерчивает вертикальную линию, ограничивающую область программы.
$A \leftarrow B$	Присваивание переменной A значения B .
или <pre> оператор1 if условие if условие оператор1 оператор2 .. </pre>	Условный оператор (оператор ветвления) if . <i>Условием</i> может быть любое логическое выражение (логические операторы находятся на панели Boolean). <i>Условие</i> должно стоять после if , а оператор, который выполняется, если выполнено заданное условие, — перед if .
<pre> оператор1 if условие оператор2 otherwise Пример: f(x) := -x if x < 0 x otherwise f(-3) = 3 f(4) = 4 </pre>	Полный оператор ветвления. Otherwise — оператор, который должен быть исполнен, если условие оператора if не выполняется.
<pre> for инд ∈ нач..кон оператор Пример: Sum(n) := s ← 0 for i ∈ 1..n s ← s + i Sum(4) = 10 </pre>	Оператор цикла с параметром for . За ключевым словом for следует переменная-счетчик, которая принимает значения от начального до конечного с шагом, равным единице. Внутренние операторы цикла сдвинуты немного вправо.
<pre> while условие оператор Пример: N(x, f, f_x) := while f(x) > 10⁻⁶ x ← x - f(x)/f_x(x) N(2, sin, cos) = 3.142 </pre>	Оператор цикла с предусловием while . Внутренние операторы цикла будут исполняться до тех пор, пока будет истинным <i>условие</i> , следующее за ключевым словом while .
<pre> continue Пример: continue if x ≥ 10 </pre>	Оператор continue служит для преждевременного завершения текущей итерации цикла; сам цикл при этом продолжается.
<pre> break Пример: break if i ≥ 10 </pre>	Оператор break служит для преждевременного завершения цикла, чтобы, например, избежать закливания или слишком продолжительных вычислений.
<pre> return Пример: return y </pre>	Оператор преждевременного завершения программы; указанное в ячейке значение будет возвращено.
<pre> on error Пример: expr1 on error expr2 </pre>	Если при вычислении выражения $expr2$ возникла ошибка, вычисляется выражение $expr1$.

Пример 1. Функция, возвращающая -1 , 0 или 1 в зависимости от знака аргумента:

Информационные технологии в менеджменте

$$\text{Sign}(x) := \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$

$$\text{Sign}(2) = 1$$

Пример 2.

$$f(x) := \begin{cases} \text{"negative"} & \text{if } x < 0 \\ \text{"positive"} & \text{if } x > 0 \\ \text{"zero"} & \text{otherwise} \end{cases}$$

$$f(-1) = \text{"negative"}$$

$$f(1) = \text{"positive"}$$

$$f(0) = \text{"zero"}$$

Пример 3. Оператор цикла **for** с ранжированной переменной:

$$x := \begin{cases} z \leftarrow 0 \\ \text{for } i \in 0..5 \\ z \leftarrow z + i \end{cases}$$

$$x = 15$$

Пример 4. Оператор цикла **for** с вектором:

$$x := \begin{cases} z \leftarrow 0 \\ \text{for } i \in (1 \ 2 \ 3) \\ z \leftarrow z + i \end{cases}$$

$$x = 6$$

Пример 5. Оператор цикла **while**:

$$x := \begin{cases} z \leftarrow 0 \\ \text{while } z < 10 \\ z \leftarrow z + 1 \end{cases}$$

$$x = 10$$

Пример 6. Оператор **break** внутри цикла **for**:

$$x := \begin{cases} z \leftarrow 0 \\ \text{for } i \in 0..5 \\ \quad z \leftarrow z + i \\ \quad \text{break if } i = 2 \end{cases}$$

$$x = 3$$

Пример 7. Оператор **break** внутри цикла **while**:

$$x := \begin{cases} z \leftarrow 0 \\ \text{while } z < 10 \\ \quad \begin{cases} z \leftarrow z + 1 \\ \text{break if } z > 5 \end{cases} \end{cases}$$

$x = 6$

Пример 8. Оператор **continue** в конце цикла **while**:

$$x := \begin{cases} z \leftarrow 0 \\ \text{while } z < 10 \\ \quad \begin{cases} z \leftarrow z + 1 \\ \text{continue} \end{cases} \end{cases}$$

$x = 10$

Пример 9. Оператор **continue** в конце цикла **for**:

$$x := \begin{cases} z \leftarrow 0 \\ \text{for } i \in (1 \ 2 \ 3) \\ \quad \begin{cases} z \leftarrow z + i \\ \text{continue} \end{cases} \end{cases}$$

$x = 6$

Пример 10. Возврат значения обозначен явно в последней строке программы:

$$f(x) := \begin{cases} y \leftarrow x^2 \\ z \leftarrow y + 1 \\ z \end{cases}$$

$f(2) = 5$

Вместе с тем, можно прервать выполнение программы в любой ее точке (например, с помощью условного оператора) и выдать некоторое значение, применив оператор **return**. В этом случае при выполнении указанного условия (пример 11) значение, введенное в местозаполнитель после **return**, возвращается в качестве результата, а никакой другой код больше не выполняется.

Пример 11. Применение оператора **return**:

$$f(x) := \begin{cases} z \leftarrow x^2 \\ \text{return "zero" if } x = 0 \\ z \end{cases}$$

$f(-1) = 1$ $f(0) = \text{"zero"}$

$f(2) = 4$ $f(6) = 36$

Программирование в MathCAD позволяет осуществлять дополнительную **обработку ошибок**. Если пользователь предполагает, что выполнение кода в каком-либо месте программного модуля способно вызвать ошибку (например, деление на ноль), то эту ошибку можно перехватить с помощью оператора **on error**.

Пример 12. Перехват ошибки деления на ноль:

Информационные технологии в менеджменте

$$f(n) := \begin{cases} z \leftarrow n \\ \text{"user error: can't divide by zero"} \text{ on error } \frac{1}{z} \end{cases}$$

$$f(-2) = -0.5$$

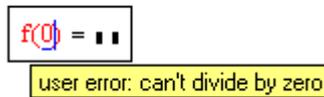
$$f(0) = \text{"user error: can't divide by zero"}$$

Оператор перехвата ошибок удобно применять в комбинации со встроенной функцией **error(S)**. Она приводит к генерации ошибки в обычной для MathCAD форме с сообщением **S**. Пример усовершенствования листинга [примера 12](#) для такого стиля обработки ошибки деления на ноль:

$$f(n) := \begin{cases} z \leftarrow n \\ \text{error("user error: can't divide by zero")} \text{ on error } \frac{1}{z} \end{cases}$$

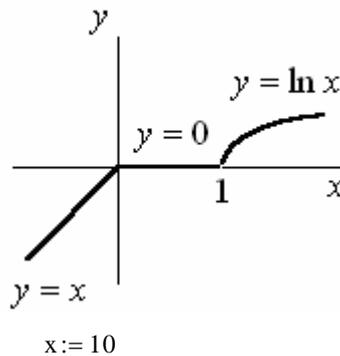
$$f(-2) = -0.5$$

$$f(0) = \blacksquare$$



Обратите внимание, что сделанные изменения свелись к помещению текста сообщения об ошибке в аргумент функции **error**.

Пример 13. Напишите программу для вычисления значения функции, заданной графически, с использованием операторов ветвления **if**.



$$y := \begin{cases} y \leftarrow x & \text{if } x < 0 \\ y \leftarrow 0 & \text{if } 0 \leq x \leq 1 \\ y \leftarrow \ln(x) & \text{if } x > 1 \\ y \end{cases}$$

$$y = 2.303$$

Пример 14. Для функции, заданной графически, вычислите значения функции y для всех значений аргумента x на заданном интервале $[x_H, x_K]$ с заданным шагом d_x .

```

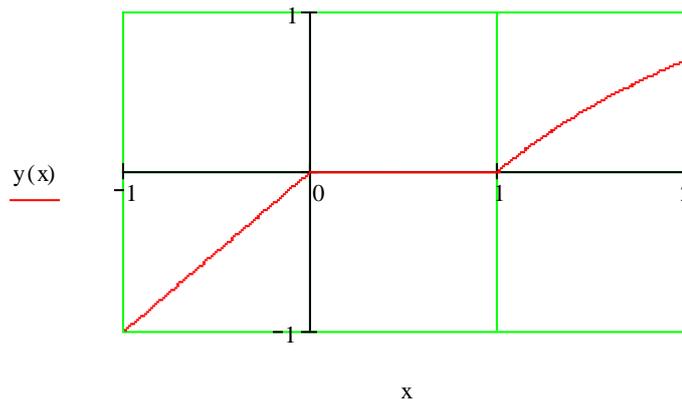
(x
 y) := | dx ← 0.05
        | xn ← -2
        | xk ← 2
        | i ← 0
        | for xx ∈ xn, xn + dx..xk
        |   | yi ← xx if xx < 0
        |   | yi ← 0 if 0 ≤ xx ≤ 1
        |   | yi ← ln(xx) if xx > 1
        |   | xi ← xx
        |   | i ← i + 1
        | (x
        | (y)
    
```

Или более просто:

```
x := -1, -0.99..2
```

```

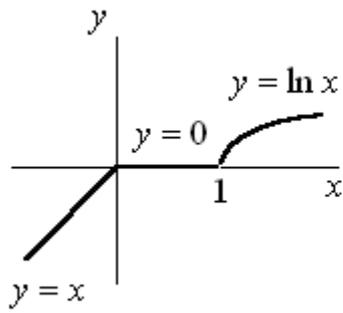
y(x) := | y ← x if x < 0
        | y ← 0 if 0 ≤ x ≤ 1
        | y ← ln(x) if x > 1
        | y
    
```



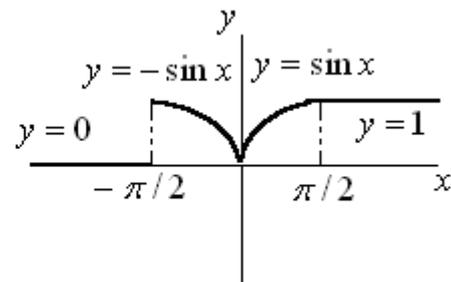
Задания

Задание 1. Напишите программу для вычисления значений функции y для всех значений аргумента x на заданном интервале $[x_H, x_K]$ с заданным шагом d_x с использованием операторов ветвления **if** и оператора цикла **for**.

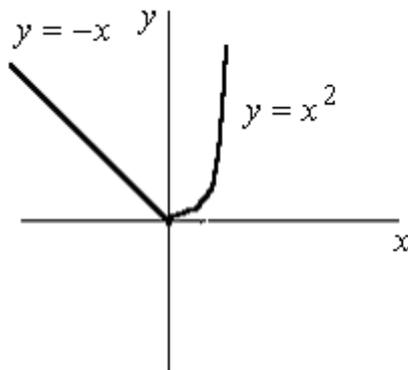
Результат выведите на экран в виде графика.



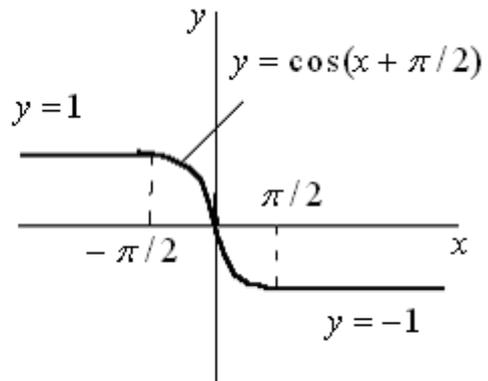
1)



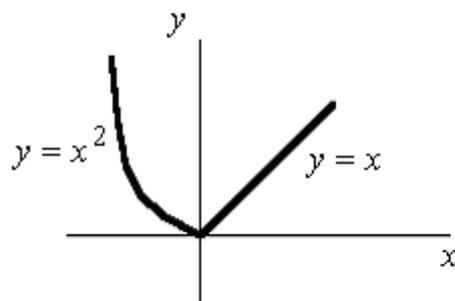
2)



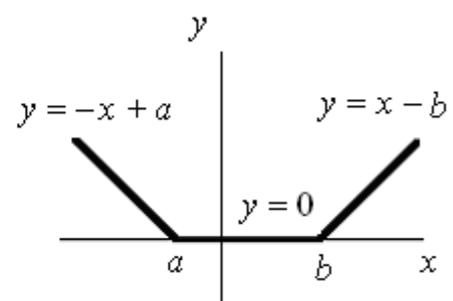
3)



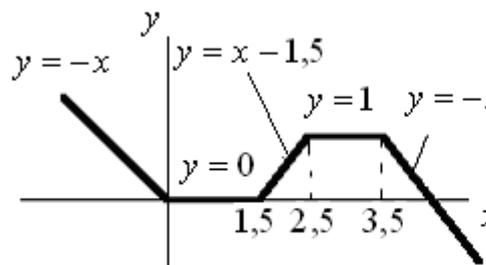
4)



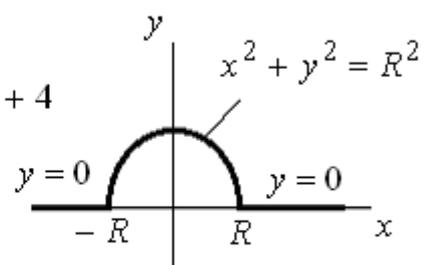
5)



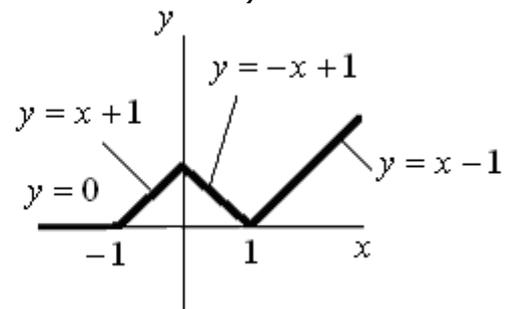
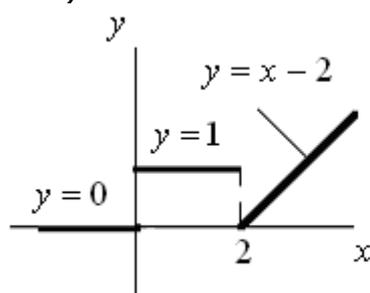
6)



7)

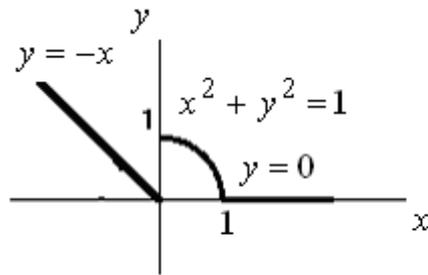


8)

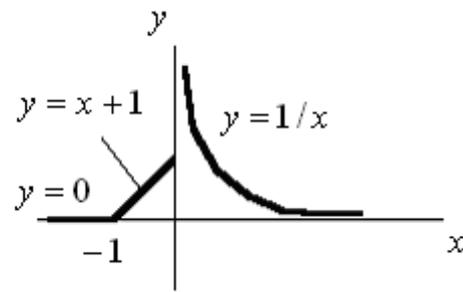


Информационные технологии в менеджменте

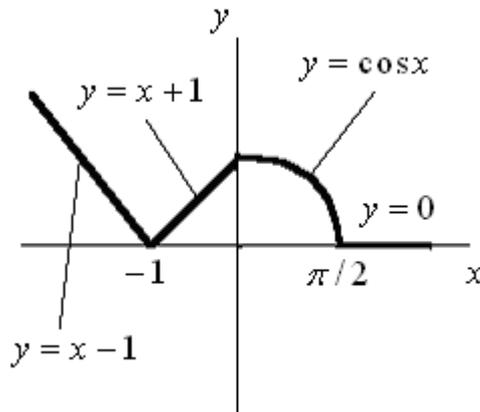
9)



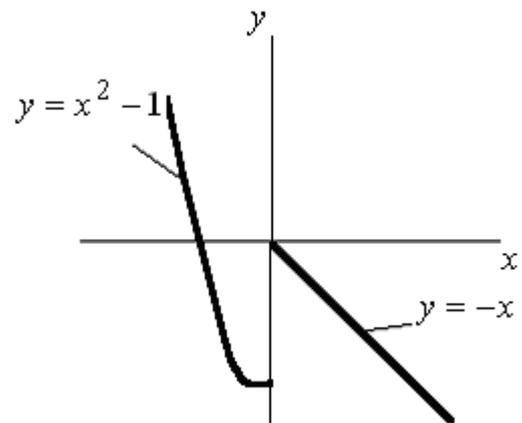
10)



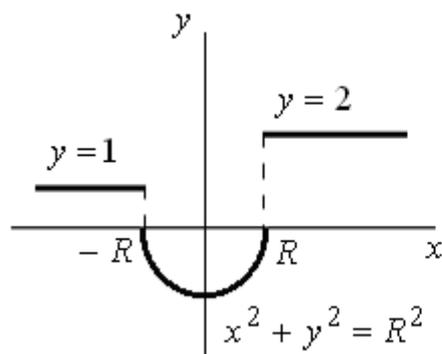
11)



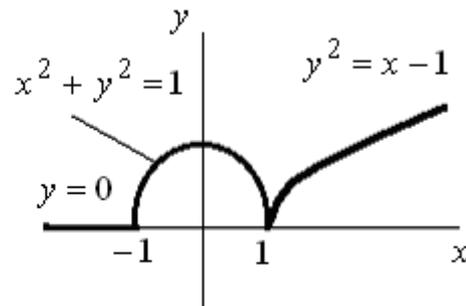
12)



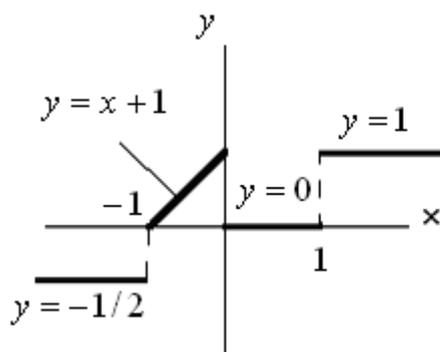
13)



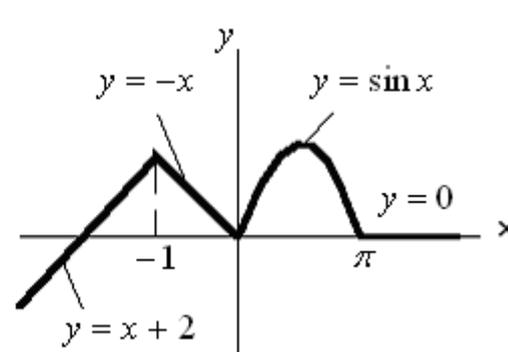
14)



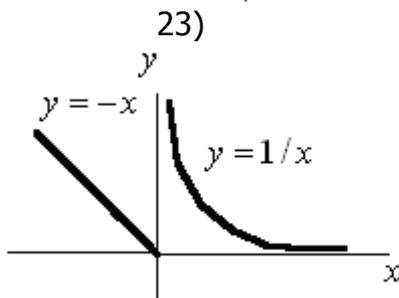
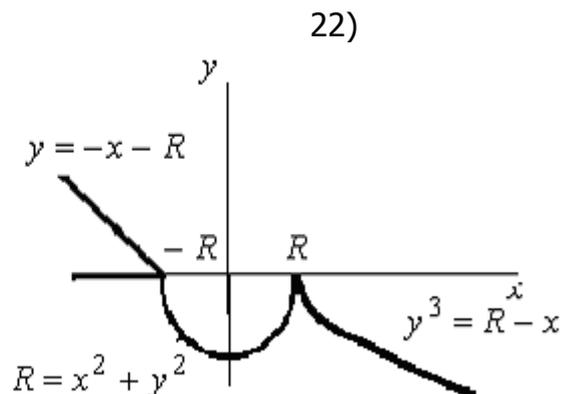
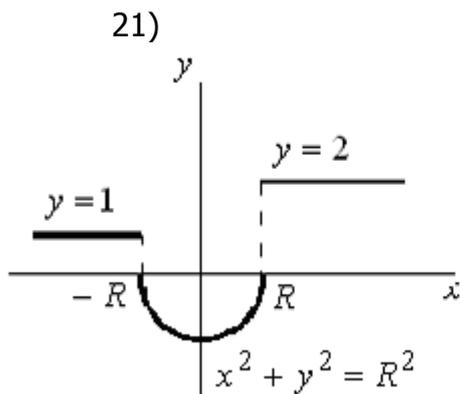
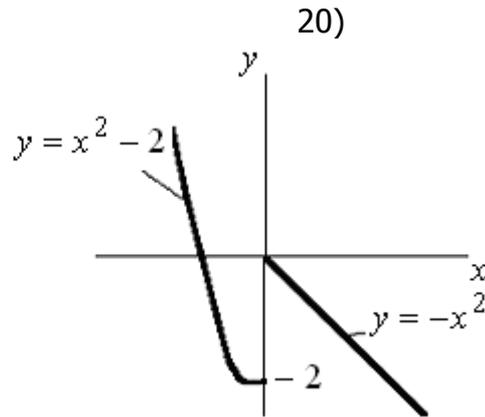
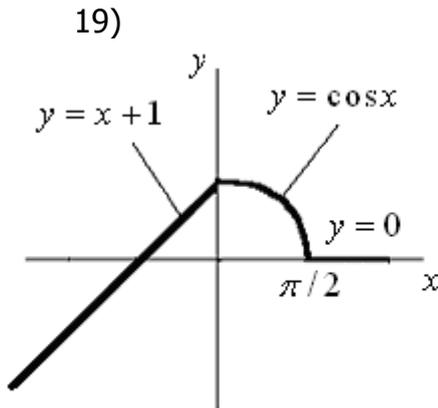
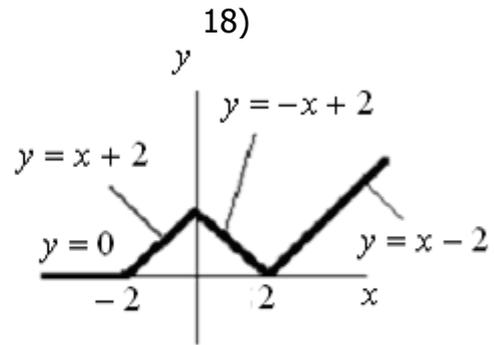
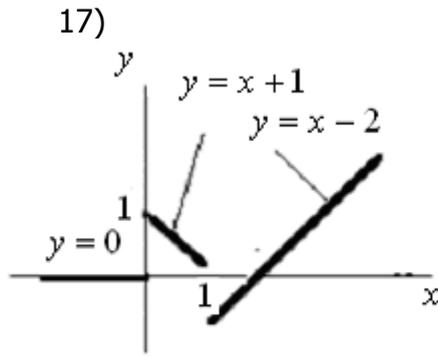
15)



16)



Информационные технологии в менеджменте



25)

Задание 2. Напишите программу вычисления функции $F(x)$. Диапазон изменения аргумента задайте самостоятельно.

Результат выведите на экран в виде графика.

$$1., 11., 21. F = \frac{3y + x^2z}{\pi} \begin{cases} y = \frac{-8x \cdot \sin x}{e^{\sqrt{|x|}}}, & z = \frac{8}{-x}, & x \leq 0; \\ y = \frac{0,8x}{|\sin x|} + \cos(x - \frac{\pi}{3}), & z = \frac{2x}{\sqrt{x^3 - 1}}, & x > 0. \end{cases}$$

$$\begin{array}{l}
 2., 12., 22. \quad F = \sqrt{|xyz|} - \frac{z}{2+x} \quad \left\{ \begin{array}{l} y = \frac{\pi \sin 2x + 2}{\ln(x+2)}, \quad z = \ln x - 8, \quad x > 1,5; \\ y = \sqrt{\frac{2x^2 - 1,5}{|\cos 2x - \frac{\pi}{4}|}}, \quad z = \frac{e^{-x}}{x + 2 \ln|x|}, \quad x \leq 1,5. \end{array} \right. \\
 \\
 3., 13., 23. \quad F = (xy + z)^2 \quad \left\{ \begin{array}{l} y = \frac{(x^2 + \pi)^3}{2x - \operatorname{ctg} \frac{x}{2}}, \quad z = \frac{|x - \sin 2x|}{\pi}, \quad x < 0; \\ y = \sqrt{\frac{\ln 2x + 0,5}{15}}, \quad z = \frac{\pi x}{2x + \cos 3x} + 1, \quad x > 0. \end{array} \right. \\
 \\
 4., 14., 24. \quad F = \ln(|y + z|) \quad \left\{ \begin{array}{l} y = \frac{\ln(|2x|)}{e^{3x^2}}, \quad z = \frac{\cos 2x}{\cos^2 x} \quad x \leq 2,5; \\ y = \frac{2,1x \cdot \lg x}{\sqrt{2x - 3} + 10}, \quad z = \frac{\sin 2x}{x + \frac{\pi}{3}} \quad x > 2,5. \end{array} \right. \\
 \\
 5., 15., 25. \quad F = x + 3 \frac{y}{z} \quad \left\{ \begin{array}{l} y = \frac{\operatorname{arctg}(x)}{2 + x^2}, \quad z = \sin 2x \quad x < 1; \\ y = \frac{|2 - x|}{3,1 \operatorname{tg}(x) - \pi}, \quad z = \frac{e^{x+1}}{\sin x - 2 \cos 2x} \quad x \geq 1. \end{array} \right. \\
 \\
 6., 16., 26. \quad F = e^{xy} - z \quad \left\{ \begin{array}{l} y = \frac{|\operatorname{tg} x| - 2}{\sqrt{|x|} + x^2}, \quad z = \frac{1}{\sin(x) - \frac{\pi}{3}}, \quad x < 0; \\ y = 3 \operatorname{ctg} x, \quad z = \frac{\sin x}{2,15 + \cos 3x}, \quad x \geq 0. \end{array} \right. \\
 \\
 7., 17., 27. \quad F = 2x^3 + \frac{y}{z} \quad \left\{ \begin{array}{l} y = \frac{x^2 - 3x}{\ln 3x - 2}, \quad z = \frac{\operatorname{ctg} x - 1,1}{\cos x^2 + \sin^3 x}, \quad x \geq 3,5; \\ y = \frac{\sin^2 x}{1 + \sqrt{\ln|2x|}}, \quad z = \sqrt[5]{|2x^3|}, \quad x < 3,5. \end{array} \right. \\
 \\
 8., 18., 28. \quad F = \frac{\sin x + \cos 2y}{z + \frac{\pi}{4}} \quad \left\{ \begin{array}{l} y = \frac{|x+2| - 2}{3\sqrt{|x^3|}}, \quad z = \frac{x + \sin x}{\cos x - 1,2} \quad x < 0; \\ y = \frac{e^{\sqrt{x}} + 1}{\sqrt[3]{x^2} + 2}, \quad z = \frac{\cos x + 1}{x^3 - \sqrt{3x^2} 3} \quad x > 0. \end{array} \right.
 \end{array}$$

$$9., 19., 29. F = \sqrt{xyz^3} - 1 \quad \begin{cases} y = \frac{\operatorname{arctg}x + 1}{\operatorname{ctg}(x - \frac{\pi}{2})}, & z = \frac{\sin x + x^2}{\cos(x+1) - 1}, & x > 0,5; \\ y = x^3 + 2x^2 + 5, & z = \frac{1}{|\operatorname{tg}x + 1|}, & x \leq 0,5. \end{cases}$$

$$10., 20., 30. F = \frac{x + z^2}{(x + y + z)^2} \quad \begin{cases} y = \frac{2x^2 - 5}{\sqrt{x^3} - \frac{2}{x+4}}, & z = \sqrt[3]{3x-2} & x > 0; \\ y = \frac{3x^4 - |5-x|}{\lg|x| + 3}, & z = \frac{5x+2}{\operatorname{tg}|x-2| - 0,3} & x \leq 0. \end{cases}$$

Лабораторная работа

Mathcad: Функции для решения задач интерполяции и аппроксимации

Цель работы: получение навыков использования функций Mathcad для решения задач интерполяции и аппроксимации.

Теоретические сведения

1. Функции для решения задач интерполяции

Одномерная интерполяция

Интерполяция является одним из распространенных методов обработки массивов данных, заданных своими узловыми значениями в табличной форме, и предполагает нахождение значений исследуемых функций между узловыми точками.

В Mathcad применяются два вида интерполяции:

— линейная — соседние точки соединяются отрезками прямых (линейным сплайном);

— нелинейная сплайн-интерполяция — соседние точки соединяются отрезками кубического сплайна (от [англ.](#) spline — планка, рейка).

Суть любого метода интерполяции заключается в разбиении отрезка, на котором исследуется функция на n отрезков. На каждом из отрезков вычисляются значения функции и строится кривая, соединяющая две точки на отрезке.

В случае линейной интерполяции такой кривой является прямая, а в случае кубической — парабола.

Линейная интерполяция позволяет построить «грубую» интерполяцию, а кубическая более точную, т. к. кубическая интерполяция не только содержит в себе формулу линейной интерполяции $P(x) = d + cx$, но к ней добавляется уточнение $P(x) = d + cx + bx^2 + ax^3$.

Линейная интерполяция. Пусть вектор X содержит координаты абсцисс заданных точек (абсциссы должны идти по возрастанию), а вектор Y — координаты ординат точек. При этом векторы X и Y должны быть одинаковой размерности.

Информационные технологии в менеджменте

Чтобы предсказать значение функции, например, в точках 2,5 и 1,2 достаточно набрать:

Linterp(X,Y,2.5)=

Аналогично:

Linterp(X,Y,1.2)=

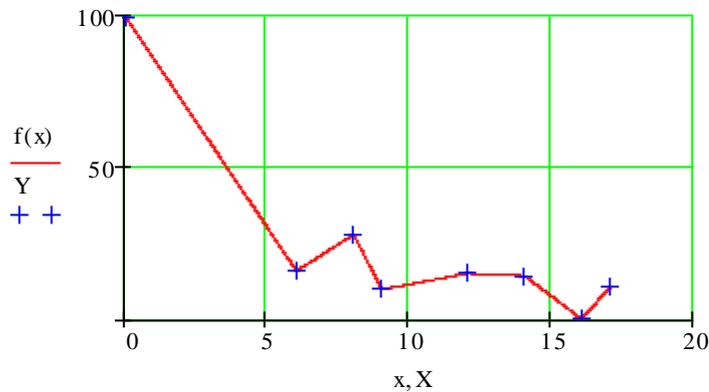
Можно также построить график интерполяционной кривой (в случае линейной интерполяции это ломаная).

Пример 1. Реализация линейной интерполяции:

$$X := \begin{pmatrix} 0 \\ 6 \\ 8 \\ 9 \\ 12 \\ 14 \\ 16 \\ 17 \end{pmatrix} \quad Y := \begin{pmatrix} 100 \\ 16 \\ 28 \\ 9.9 \\ 15.2 \\ 14.4 \\ 0.16 \\ 10.7 \end{pmatrix}$$

$$x := X_0, X_0 + 0.01.. X_{\text{last}(X)} \quad X_{\text{last}(X)} = 17$$

$$f(x) := \text{linterp}(X, Y, x)$$



Кубическая сплайн-интерполяция. Кубическая интерполяция проводит кривую через заданные точки таким образом, чтобы первые и вторые производные кривой были непрерывны в каждой точке. Это обеспечивает гладкость интерполяционной кривой.

Для решения задач одномерной кубической интерполяции предусмотрены функции **lspline**, **pspline**, **cspline**, **interp**. При этом функции **lspline**, **pspline**, **cspline** имеют вспомогательное значение и осуществляют предварительную подготовку к решению задачи в соответствии с выбранным методом интерполяции данных, а **interp** использует эти предварительно полученные результаты для определения значений функции во всем диапазоне изменения аргумента.

Функции **lspline**, **pspline**, **cspline** возвращают вектор D , размерностью $(n + 3) \times 1$, который в качестве первых трех элементов содержит служебные параметры, определяющие особенности функций **lspline**, **pspline**, **cspline**. Остальными n элементами вектора D служат вторые производные в узловых точках.

Информационные технологии в менеджменте

Рассмотрим функции более подробно:

lspline(X,Y) — вычисляет вектор D с линейным приближением в конечных точках;

pspline(X,Y) — вычисляет вектор D с параболическим приближением в конечных точках;

cspline(X,Y) — вычисляет вектор D с полиномиально-кубическим приближением в конечных точках;

interp(D,X,Y,x) — вычисляет значения функции $f(x)$ с использованием кубических сплайнов для произвольного значения аргумента x по определенному одной из перечисленных выше функций вектору D .

Векторы X и Y размерности $n \times 1$ задают совокупность исходных узловых точек. Вектор X должен быть ранжирован в порядке возрастания значений его элементов.

Особенностью функций **lspline**, **pspline**, **cspline** служит то, что состыковка сплайнов осуществляется в узловых точках; между узлами функция $f(x)$ вычисляется по кубическому полиному.

Пример 2. Реализация кубической интерполяции:

Информационные технологии в менеджменте

$Dl := lspline(X, Y)$

$Dp := pspline(X, Y)$

$Dc := cspline(X, Y)$

$Dl =$

	0
0	0
1	3
2	0
3	0
4	11.427
5	-31.413
6	21.026
7	-5.865
8	-8.715
9	20.565
10	0

$Dp =$

	0
0	0
1	3
2	1
3	8.212
4	8.212
5	-30.328
6	20.943
7	-6.006
8	-7.886
9	17.39
10	17.39

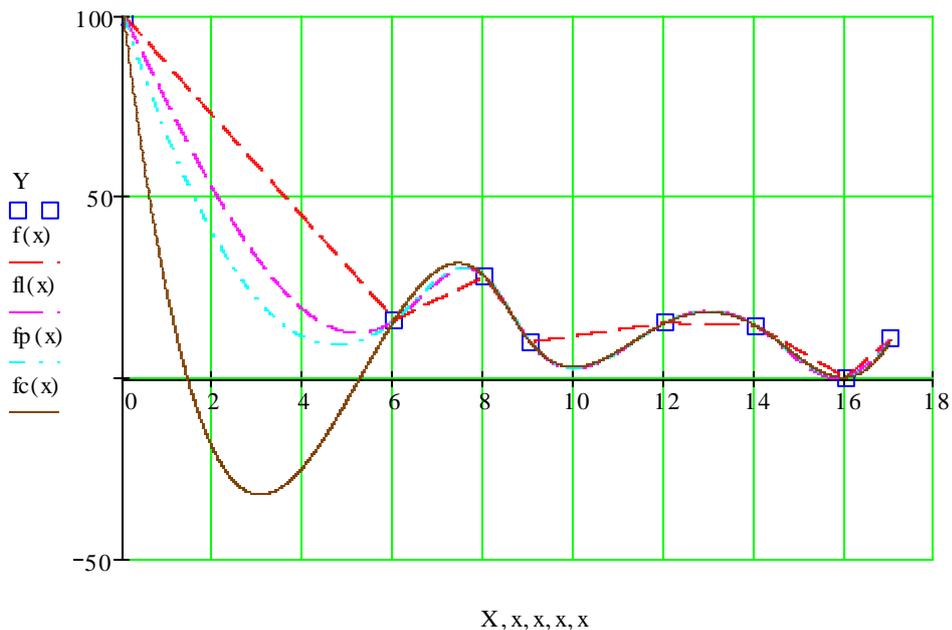
$Dc =$

	0
0	0
1	3
2	2
3	47.174
4	-7.05
5	-25.124
6	20.245
7	-5.878
8	-7.476
9	15.623
10	27.173

$fl(x) := interp(Dl, X, Y, x)$

$fp(x) := interp(Dp, X, Y, x)$

$fc(x) := interp(Dc, X, Y, x)$



Двумерная кубическая сплайн-интерполяция

Функции **lspline**, **pspline**, **cspline**, **interp** могут быть использованы также и для осуществления двумерной кубической сплайн-интерполяции поверхностей $z = F(x, y)$.

Для двумерного случая вызов функции осуществляется следующим образом:

cspline(XY,Z) — возвращает вектор D вторых производных функции в узловых точках, заданных значениями элементов двухстолбцовой матрицы XY и значениями элементов матрицы Z размерности $n \times n$.

Информационные технологии в менеджменте

Элементы векторов X и Y должны быть ранжированы по возрастанию их значений. Двухстолбцовая матрица XY может быть создана соединением векторов X и Y с использованием функции **augment**:

$$X := \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \end{pmatrix} \quad Y := \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \end{pmatrix}$$

$$XY := \text{augment}(X, Y)$$

$$XY = \begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 2 & 2 \\ 3 & 3 \end{pmatrix}$$

interp(D,XY,Z,v) — вычисляет $z = F(x, y)$ на всем диапазоне изменения значений x и y . Вектор v служит для рассматриваемого двумерного случая векторным аргументом, т. е. вектором с двумя компонентами x и y : $v = \begin{pmatrix} x \\ y \end{pmatrix}$.

Пример 3. Реализация двумерной интерполяции:

$$X := \begin{pmatrix} -5 \\ -4 \\ -3 \\ 0 \\ 4 \\ 5 \\ 6 \\ 7 \end{pmatrix} \quad Y := \begin{pmatrix} -5 \\ -4 \\ -3 \\ 0 \\ 4 \\ 5 \\ 6 \\ 7 \end{pmatrix}$$

$$XY := \text{augment}(X, Y) \quad XY = \begin{pmatrix} -5 & -5 \\ -4 & -4 \\ -3 & -3 \\ 0 & 0 \\ 4 & 4 \\ 5 & 5 \\ 6 & 6 \\ 7 & 7 \end{pmatrix}$$

$$Z := \begin{cases} \text{for } i \in 0..\text{length}(X) - 1 \\ \quad \text{for } j \in 0..\text{length}(Y) - 1 \\ \quad \quad Z_{i,j} \leftarrow \text{md}(X_i) + \text{md}(Y_j) \\ Z \end{cases} \quad Z = \begin{pmatrix} -0.973 & -4.326 & -4.637 & -3.552 & 0.132 & -4.347 & 3.145 & -1.845 \\ -2.088 & -5.213 & -6.372 & -2.157 & -0.33 & -0.93 & 3.976 & 3.237 \\ -1.406 & -5.114 & -3.686 & -1.374 & 1.143 & -0.959 & 1.827 & -1.072 \\ -3.499 & -0.566 & -1.28 & 0 & 0.765 & 0.778 & 1.678 & 5.053 \\ -3.681 & 0.363 & 2.149 & 1.887 & 5.756 & 3.937 & 6.363 & 4.118 \\ -1.559 & -0.081 & -0.393 & 1.43 & 6.574 & 1.272 & 5.295 & 3.562 \\ 1.796 & -0.165 & 0.886 & 3.772 & 4.934 & 3.358 & 1.152 & 11.851 \\ -0.462 & 2.59 & 1.62 & 4.373 & 3.511 & 4.724 & 7.059 & 9.528 \end{pmatrix}$$

$$D := \text{cspline}(XY, Z)$$

$$D = \begin{array}{|c|c|} \hline & 0 \\ \hline 0 & 0 \\ \hline 1 & 3 \\ \hline 2 & 2 \\ \hline 3 & 4.979 \\ \hline 4 & 1.798 \\ \hline 5 & -1.383 \\ \hline 6 & 0.328 \\ \hline 7 & 0.866 \\ \hline 8 & 3.035 \\ \hline 9 & -5.613 \\ \hline \end{array}$$

Информационные технологии в менеджменте

$$x := X_0, X_0 + 0.01.. X_{\text{last}(X)} \quad y := Y_0, Y_0 + 0.01.. Y_{\text{last}(Y)}$$

$$C(x, y) := \text{interp} \left[D, XY, Z, \begin{pmatrix} x \\ y \end{pmatrix} \right]$$

$$M := \text{CreateMesh} (C, \text{min}(X), \text{max}(X), \text{min}(Y), \text{max}(Y), 40, 40)$$

Для построения исходных опорных точек в пространстве необходимо преобразовать векторы X и Y в матрицы:

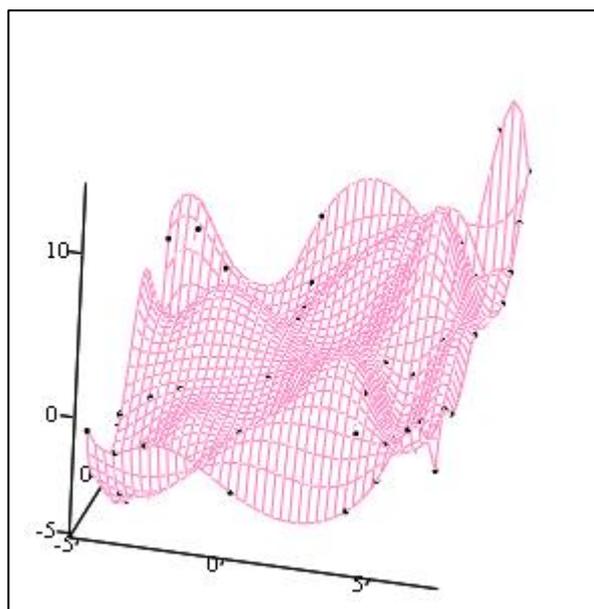
$$i := 0..7$$

$$XX^{(i)} := X$$

$$YY^{(i)} := Y$$

$$XX = \begin{pmatrix} -5 & -5 & -5 & -5 & -5 & -5 & -5 & -5 \\ -4 & -4 & -4 & -4 & -4 & -4 & -4 & -4 \\ -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \end{pmatrix}$$

$$YY = \begin{pmatrix} -5 & -5 & -5 & -5 & -5 & -5 & -5 & -5 \\ -4 & -4 & -4 & -4 & -4 & -4 & -4 & -4 \\ -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \end{pmatrix}$$



$$(XX, YY^T, Z), M$$

В **примере 3** для формирования значений элементов матрицы Z используется встроенная функция Mathcad **rnd(i)**, которая возвращает случайное число с равномерным распределением в диапазоне от 0 до i .

А для построения графика интерполированной поверхности использовалась функция:

M:=CreateMesh(f,xn,xk,yn,yk,s1,s2),

где f — функция, определяющая поверхность; xn, xk, yn, yk — начальные и конечные значения независимых переменных x и y ; $s1, s2$ — размерность сетки.

2. Функции для решения задач аппроксимации

Аппроксимация, или *приближение* — научный метод, состоящий в замене одних объектов другими, в том или ином смысле близкими к исходным, но более простыми.

Аппроксимация используется при решении разнообразных практических задач обработки массивов данных, в которых отразились результаты случайных воздействий и возмущений. Задачи, возникающие при этом, связаны, как правило, с попыткой найти математическое описание (математическую модель), адекватно представляющее совокупность данных. Такие задачи аппроксимации отличаются от только что рассмотренных задач интерполяции, которые не были связаны с получением математической модели, а состояли лишь в определении промежуточных значений функции между узлами.

Одномерная аппроксимация. Общую параметрическую аппроксимацию осуществляет функция:

linfit(X,Y,F) — вычисляет весовые коэффициенты набора локальных функций, введенных в виде выражений в F .

Пример 4. Реализация аппроксимации с использованием функции **linfit**:

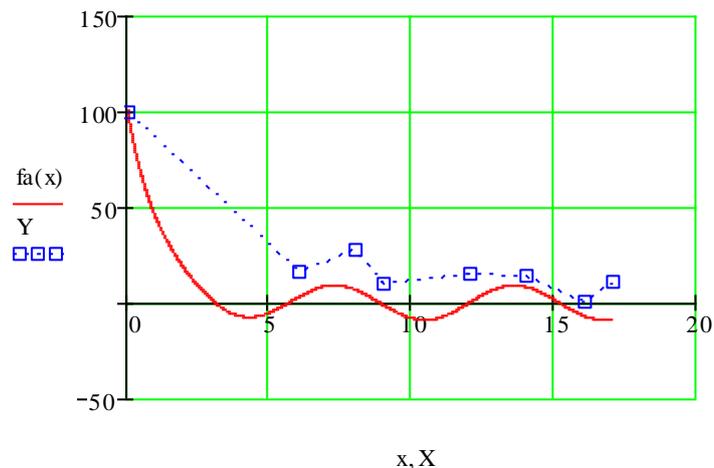
Информационные технологии в менеджменте

$$X := \begin{pmatrix} 0 \\ 6 \\ 8 \\ 9 \\ 12 \\ 14 \\ 16 \\ 17 \end{pmatrix} \quad Y := \begin{pmatrix} 100 \\ 16 \\ 28 \\ 9.9 \\ 15.2 \\ 14.4 \\ 0.16 \\ 10.7 \end{pmatrix} \quad F(z) := \begin{pmatrix} \exp(-z) \\ \sin(z) \\ \cos(z) \end{pmatrix}$$

$$L := \text{linfit}(X, Y, F)$$

$$L = \begin{pmatrix} 94.919 \\ 7.425 \\ 5.121 \end{pmatrix} \quad x := X_0, X_0 + 0.01.. X_{\text{last}}(X)$$

$$\text{fa}(x) := F(x) \cdot L$$



В Mathcad имеются функции, позволяющие решать задачи аппроксимации для специальных случаев, когда пользователь с достаточной степенью уверенности может назначить для исследуемого массива данных аппроксимирующее выражение в виде одной из простейших функций (экспоненциальной, синусоидальной, логарифмической и т. д.).

Встроенные функции этой группы несколько отличаются от ранее описанных. Они реализуют поисковые алгоритмы неизвестных коэффициентов и нуждаются, поэтому, в предварительном задании начальных приближений этих параметров.

В рассматриваемую группу входят функции, которые вычисляют коэффициенты следующих зависимостей, наилучшим образом аппроксимирующих данные, представленные в X, Y :

expfit(X,Y,g) — коэффициенты экспоненциальной функции $a \cdot \exp(bx) + c$;

Информационные технологии в менеджменте

Существует также функция для обеспечения аппроксимации при произвольной степени аппроксимирующего полинома:

regress(X,Y,n).

Она возвращает вектор D , запрашиваемый функцией **interp(D,X,Y,x)**, содержащий коэффициенты многочлена n -й степени, который наилучшим образом приближает точки с координатами, хранящимися в векторах X и Y .

На практике не рекомендуется делать степень аппроксимирующего полинома выше 4—6, поскольку погрешности реализации сильно возрастают.

Функция **regress** создает единственный приближающий полином, коэффициенты которого вычисляются по всей совокупности заданных точек, т. е. глобально. Иногда полезна другая функция полиномиальной регрессии, дающая локальные приближения отрезками полиномов второй степени — **loess(X,Y,span)**. Эта функция возвращает используемый функцией **interp(D,X,Y,x)** вектор D , дающий наилучшее приближение данных (с координатами точек в векторах X и Y) отрезками полиномов второй степени.

Аргумент $span > 0$ задает степень сглаженности данных, определяя размер отрезков полиномов, из которых сшивается аппроксимирующая кривая. На практике $0,2 < span < 2$. При $span = 2$ результат аппроксимации тот же, что при аппроксимации одной параболой. При $span = 0,2$ аппроксимирующая кривая почти точно описывает любой набор данных. Рекомендуемое начальное значение — 0,75.

Пример 6. Одномерная аппроксимация с использованием функций **regress** и **loess**:

Информационные технологии в менеджменте

n := 4

span := 0.75

D1 := regress (X, Y, n)

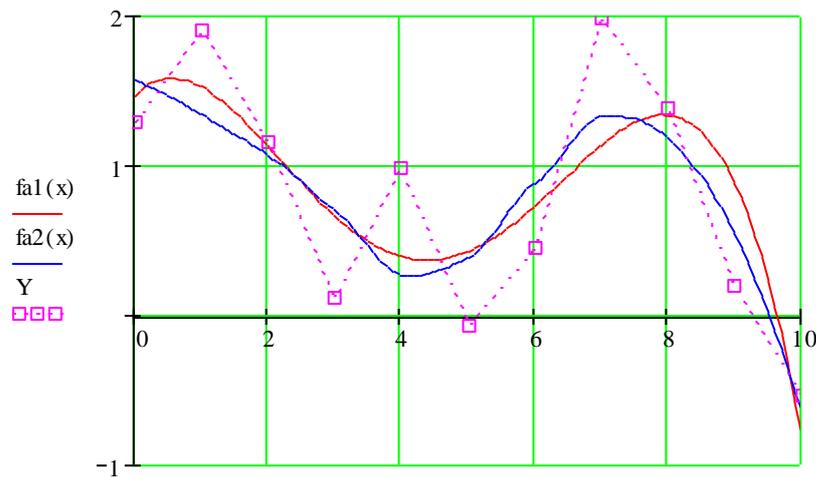
D2 := loess (X, Y, span)

$$D1 = \begin{pmatrix} 3 \\ 3 \\ 4 \\ 1.473 \\ 0.455 \\ -0.492 \\ 0.101 \\ -5.894 \times 10^{-3} \end{pmatrix}$$

	0
0	1
1	1.15
2	1.584
3	1.355
4	1.075
5	0.703
6	0.27
7	0.381
8	0.88
9	1.331
10	1.193
11	0.542
12	-0.66
13	-0.29
14	0.563
15	0.083

fa1(x) := interp(D1, X, Y, x)

fa2(x) := interp(D2, X, Y, x)



Двумерная аппроксимация. Mathcad позволяет выполнять также многомерную аппроксимацию, типичный случай которой — приближение трехмерных поверхностей. Их можно характеризовать вектором значений высот Z , соответствующих двумерному массиву XY координат точек на горизонтальной плоскости (XY — матрица $n \times 2$, содержащая координаты x и y).

Новых функций для этого не задано. Используются уже описанные функции в несколько иной форме (рис. 7):

Информационные технологии в менеджменте

— **regress(XY,Z,n)** — возвращает вектор D , запрашиваемый функцией **interp(D,XY,Z,v)** для вычисления многочлена n -й степени, который наилучшим образом приближает точки множеств XY и Z ; v — вектор с двумя компонентами x и y ;

— **loess(XY,Z,span)** — аналогична **loess(X,Y,span)**, но в многомерном случае.

Пример 7. Реализация двумерной аппроксимации с использованием функций **regress** и **loess**:

```
ii:=0..30      jj:=0..30
```

```
Xii:=ii      Yjj:=jj
```

```
XY := augment(X, Y)
```

```
Z := | for i ∈ 0..length(X) - 1
      | Zi ← sin(rnd(5))
      | Z
```

```
D1 := regress(XY,Z,4)
```

```
D2 := loess(XY,Z,1)
```

D1 =

	0
0	3
1	3
2	4
3	-2.871·10 ⁻⁶
4	-2.871·10 ⁻⁶
5	2.23·10 ⁻⁴
6	-5.389·10 ⁻³
7	2.23·10 ⁻⁴
8	-2.871·10 ⁻⁶
9	0.039
10	-5.389·10 ⁻³
11	2.23·10 ⁻⁴
12	-2.871·10 ⁻⁶
13	0.095
14	0.039
15	-5.389·10 ⁻³

D2 =

	0
0	1
1	196
2	0.218
3	0.191
4	0.168
5	0.147
6	0.124
7	0.102
8	0.084
9	0.068
10	0.054
11	0.04
12	0.028
13	0.021
14	0.016
15	9.509·10 ⁻³

Информационные технологии в менеджменте

$$x := \min(X), \min(X) + 0.01 \cdot \max(X)$$

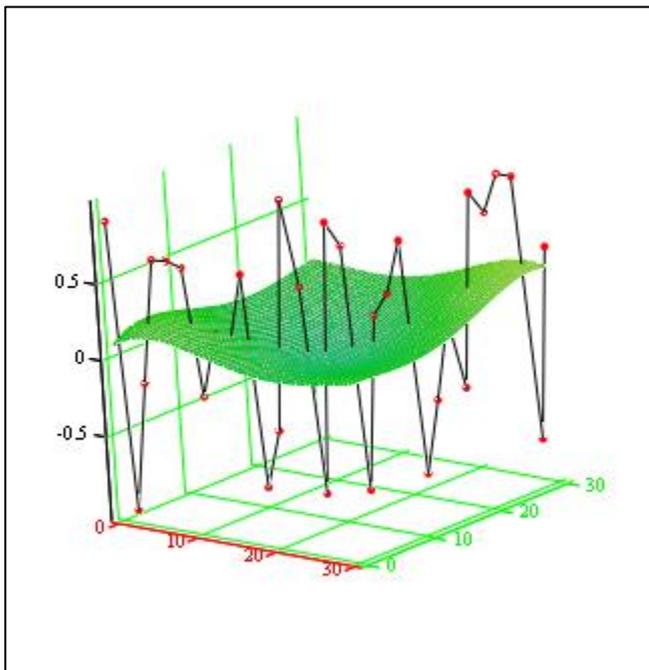
$$y := \min(Y), \min(Y) + 0.01 \cdot \max(Y)$$

$$C1(x, y) := \text{interp}\left[D1, XY, Z, \begin{pmatrix} x \\ y \end{pmatrix}\right]$$

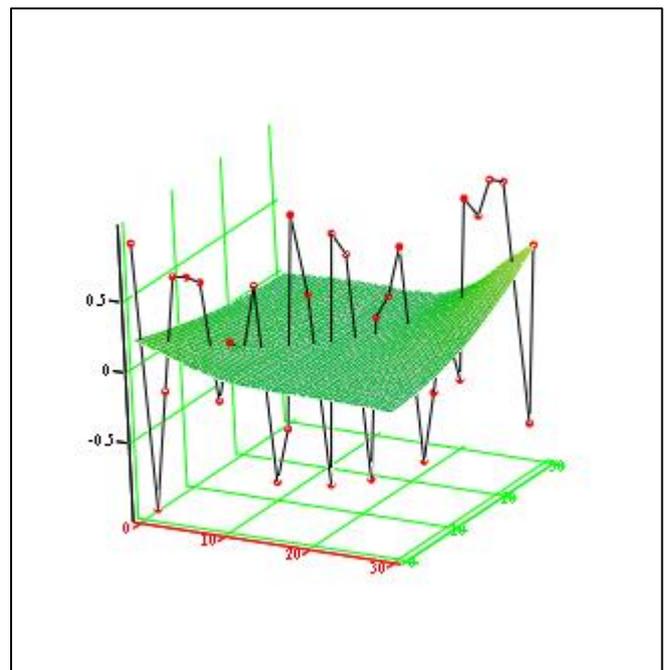
$$I1 := \text{CreateMesh}(C1, \min(X), \max(X), \min(Y), \max(Y), 50, 50)$$

$$C2(x, y) := \text{interp}\left[D2, XY, Z, \begin{pmatrix} x \\ y \end{pmatrix}\right]$$

$$I2 := \text{CreateMesh}(C2, \min(X), \max(X), \min(Y), \max(Y), 50, 50)$$



I1, (X, Y, Z)



I2, (X, Y, Z)

Задания

Задание 1. С помощью функции **linterp(X,Y,x)** проведите одномерную линейную интерполяцию функции, заданной узловыми точками в векторах X, Y размерности 8×1 .

Значения элементов вектора X задайте произвольно (с учетом требования к входным данным встроенных функций интерполяции).

Значения элементов вектора Y задайте следующим образом:

$$i = 0..length(X) - 1$$

$$Y_i = Nv + rnd(X_i)'$$

где Nv — номер варианта; $rnd(i)$ — встроенная функция Mathcad, возвращает случайное число с равномерным распределением в диапазоне от 0 до X_i .

Задание 2. С помощью функций **D:=lspline(X,Y)**, **D:=pspline(X,Y)**, **D:=cspline(X,Y)** проведите кубическую интерполяцию функции, сформированной в пункте 1.

Сделайте выводы о качестве интерполяции разными функциями.

Задание 3. С помощью функции **D:=cspline(XY,Z)** проведите двумерную кубическую интерполяцию поверхности, заданной узловыми точками в векторах X, Y размерности 8×1 и матрице высот Z размерности 8×8 .

Значения элементов векторов X, Y задайте произвольно (с учетом требования к входным данным встроенных функций интерполяции).

Значения элементов матрицы Z задайте следующим образом:

$$Z := \begin{array}{l} \text{for } i \in 0..length(X) - 1 \\ \quad \text{for } j \in 0..length(Y) - 1 \\ \quad \quad Z_{i,j} \leftarrow Nv + rnd(X_i) + rnd(Y_j) \end{array}$$

Задание 4. С помощью функции **linfit(X,Y,F)** проведите одномерную аппроксимацию функции, сформированной в пункте 1.

При этом необходимо методом подбора определить набор аппроксимационных функций, содержащихся в векторе F , который наилучшим образом приближает исходные узловые точки.

Задание 5. С помощью функций **regress(X,Y,n)** и **loess(X,Y,span)** проведите одномерную аппроксимацию функции, сформированной в пункте 1.

Оцените влияние порядка аппроксимирующего полинома n и аргумента $span$, задающего степень сглаженности данных, на качество аппроксимации.

Задание 6. С помощью функций **regress(XY,Z,n)** и **loess(XY,Z,span)** проведите двумерную аппроксимацию функции, заданной узловыми точками в векторах X, Y, Z .

Значения элементов векторов X, Y задайте следующим образом:

$$i = 0..40$$

$$X_i = i$$

$$Y_i = i$$

Значения элементов вектора Z :

$$i = 0..length(X) - 1$$

$$Z_i = rnd(Nv)$$

Оцените влияние порядка аппроксимирующего полинома n и аргумента $span$, задающего степень сглаженности данных, на качество аппроксимации.

Лабораторная работа

Mathcad: Функции для решения задач сглаживания и экстраполяции (предсказания)

Цель работы: получение навыков использования функций Mathcad для решения задач сглаживания и экстраполяции (предсказания).

Теоретические сведения

Функции для решения задач сглаживания

Под *сглаживанием* понимается процедура повышения достоверности данных, снижения уровня помех. Необходимость сглаживания часто возникает на этапе предварительной обработки данных, перед их использованием для получения количественных характеристик, построения математических моделей и т. д.. В качестве алгоритма такого предварительного сглаживания обычно используется процедура так называемого *скользящего окна*, под которым понимается обработка данных на выбранной части диапазона изменения аргумента при поэтапном перемещении этой части диапазона (скольжении «окна») по оси абсцисс. Для сглаживания в таком понимании в Mathcad предусмотрены три функции. При этом исходный массив задается для функций в виде векторов X и Y одинаковой размерности $(n \times 1)$. Перечисленные функции выводят результат в виде вектора сглаженных значений той же размерности:

medsmooth(Y,m) — предназначена для обработки исходного массива чисел, заданных вектором Y , методом скользящей медианы и вычисления сглаженных (по сравнению с исходными) оценок переменных y ; t — ширина окна обработки, значение t должно быть нечетным и меньшим, чем число элементов вектора Y .

Медианы давно использовались и изучались в статистике как альтернатива средним арифметическим значениям отсчетов в оценке выборочных средних значений. *Медианой числовой последовательности* x_1, x_2, \dots, x_n при нечетном n является средний по положению член ряда, получающегося при упорядочивания этой последовательности по возрастанию (или убыванию). Для четных n медиану определяют как среднее арифметическое двух средних отсчетов упорядоченной последовательности.

ksmooth(X,Y,b) — в предположении нормального распределения вычисляет сглаженные оценки методом средневзвешенной обработки, b — ширина окна обработки, которая должна в несколько раз превышать интервал между точками по оси x ; вектор X должен содержать элементы, ранжированные в порядке возрастания.

Среднее взвешенное, точнее *среднее арифметическое взвешенное* набора вещественных чисел x_1, x_2, \dots, x_n с вещественными весами w_1, w_2, \dots, w_n определяется как:

$$\bar{x} = \frac{\sum_{i=1}^n w_i \cdot x_i}{\sum_{i=1}^n w_i}.$$

Часто подразумевается, что сумма весов равна 1, тогда формула выглядит следующим образом:

$$\bar{x} = \sum_{i=1}^n w_i \cdot x_i.$$

В том случае, если все веса равны между собой, среднее арифметическое взвешенное будет равно среднему арифметическому.

supsmooth(X,Y) — вычисляет сглаженные оценки методом наименьших квадратов для k ближайших соседних точек, параметр k выбирается адаптивно.

Метод наименьших квадратов (МНК, OLS, Ordinary Least Squares) — математический метод, применяемый для решения различных задач, основанный на минимизации суммы квадратов некоторых функций от искомых переменных. Этот метод широко используется в регрессионном анализе (аппроксимации данных), поскольку позволяет построить модель на основе зашумленных экспериментальных данных. При этом помимо собственно построения модели обычно осуществляется оценка погрешности, с которой были вычислены ее параметры.

Суть метода заключается в том, чтобы найти коэффициенты C_0, C_1, \dots, C_m линейной комбинации:

$$L(t) = C_0 \varphi_0(t) + C_1 \varphi_1(t) + C_2 \varphi_2(t) + \dots + C_m \varphi_m(t),$$

таким образом, чтобы обеспечить близость кривой $L(t)$ и заданной функции.

Коэффициенты определяются из следующего матричного уравнения:

$$C = (A^T \cdot A)^{-1} \cdot A^T \cdot f,$$

где f — заданная зашумленная функция; A — прямоугольная матрица размерностью $(n+1) \times (m+1)$; n — количество точек заданной функции f ; m — порядок функции, используемой для аппроксимации сигнала.

Матрица A имеет следующий вид:

$$A = \begin{bmatrix} \varphi_0(t_0) & \varphi_1(t_0) & \dots & \varphi_m(t_0) \\ \varphi_0(t_1) & \varphi_1(t_1) & \dots & \varphi_m(t_1) \\ \dots & \dots & \dots & \dots \\ \varphi_0(t_n) & \varphi_1(t_n) & \dots & \varphi_m(t_n) \end{bmatrix}.$$

При использовании для аппроксимации степенных функций $L(t)$ будет иметь вид: $L(t) = C_0 t^0 + C_1 t^1 + C_2 t^2 + \dots + C_m t^m$, а матрица A :

$$A = \begin{bmatrix} t_0^0 & t_0^1 & \dots & t_0^m \\ t_1^0 & t_1^1 & \dots & t_1^m \\ \dots & \dots & \dots & \dots \\ t_n^0 & t_n^1 & \dots & t_n^m \end{bmatrix}.$$

При использовании тригонометрических функций (параметр $T = 2$):

Информационные технологии в менеджменте

$$L(t) = C_0 + C_1 \sin\left(\frac{\pi t}{T}\right) + C_2 \cos\left(\frac{\pi t}{T}\right) + C_3 \sin\left(\frac{2\pi t}{T}\right) + C_4 \cos\left(\frac{2\pi t}{T}\right) + \dots,$$

$$A = \begin{bmatrix} 1 & \sin\left(\frac{\pi t_0}{T}\right) & \cos\left(\frac{\pi t_0}{T}\right) & \sin\left(\frac{2\pi t_0}{T}\right) & \cos\left(\frac{2\pi t_0}{T}\right) & \dots & \sin\left(\frac{m\pi t_0}{T}\right) & \cos\left(\frac{m\pi t_0}{T}\right) \\ 1 & \sin\left(\frac{\pi t_1}{T}\right) & \cos\left(\frac{\pi t_1}{T}\right) & \sin\left(\frac{2\pi t_1}{T}\right) & \cos\left(\frac{2\pi t_1}{T}\right) & \dots & \sin\left(\frac{m\pi t_1}{T}\right) & \cos\left(\frac{m\pi t_1}{T}\right) \\ \dots & \dots \\ 1 & \sin\left(\frac{\pi t_n}{T}\right) & \cos\left(\frac{\pi t_n}{T}\right) & \sin\left(\frac{2\pi t_n}{T}\right) & \cos\left(\frac{2\pi t_n}{T}\right) & \dots & \sin\left(\frac{m\pi t_n}{T}\right) & \cos\left(\frac{m\pi t_n}{T}\right) \end{bmatrix}$$

Пример 1. Примеры использования функций сглаживания для случайного процесса.

Для функции **medsmooth** выбрана ширина окна обработки $m = 9$. Анализируя влияние ширины окна обработки на конечный результат, можно убедиться, что при снижении ширины окна сглаживающие способности использованной функции будут снижаться.

Для функции **ksmooth** была принята малая величина окна обработки, что привело к увеличению разброса результирующей кривой.

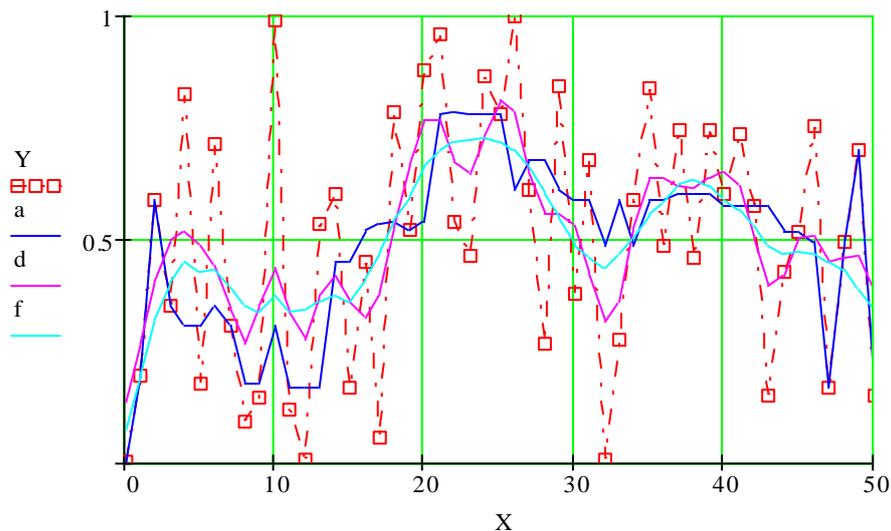
Применение функции **supsmooth** в тех же условиях дает лучший результат.

```
i:=0..50      m:=9      b:=3

Xi:=i

Yi:=md(1)

a:=medsmooth(Y,m)      d:=ksmooth(X,Y,b)      f:=supsmooth(X,Y)
```



Из приведенных результатов использования функций сглаживания виден эффект «подавления помех», важный при предварительной обработке данных.

Функции для решения задач экстраполяции (предсказания)

Прогнозирование поведения переходных процессов, сигналов, зависимостей в Mathcad осуществляется функцией:

predict(Y,m,k)

Информационные технологии в менеджменте

где Y — вектор данных; m — число последних точек существующих данных, на основе которых происходит расчет предсказываемых точек; k — число точек, в которых необходимо предсказать данные.

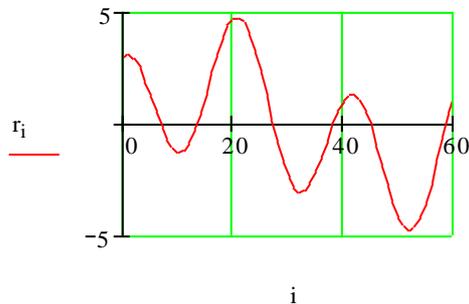
Она по ряду заданных равномерно расположенных точек позволяет рассчитать некоторое число k последующих точек, т. е. по существу выполняет экстраполяцию произвольной (но достаточно гладкой и предсказуемой) зависимости. Функция предсказания обеспечивает высокую точность при монотонных исходных функциях или функциях, представляемых полиномом невысокой степени.

Эта функция использует простейший вид алгоритма экстраполяции — линейный, поэтому исходные данные должны быть достаточно гладкими и с хорошо проявляющимися динамическими особенностями, т. е. не должны быть сильно искажены помехами.

Пример 2. Пример получения экстраполированных значений процесса, имеющего шумовую составляющую.

$i := 0..60$

$r_i := 2 \cdot \sin(i \cdot 0.1) + 3 \cdot \cos(i \cdot 0.3)$



$y := \text{morm}(61, 0, 0.1)$

$y1 := \text{morm}(61, 0, 0.3)$

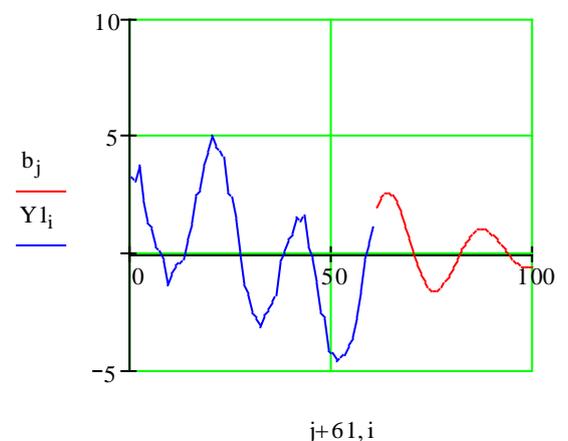
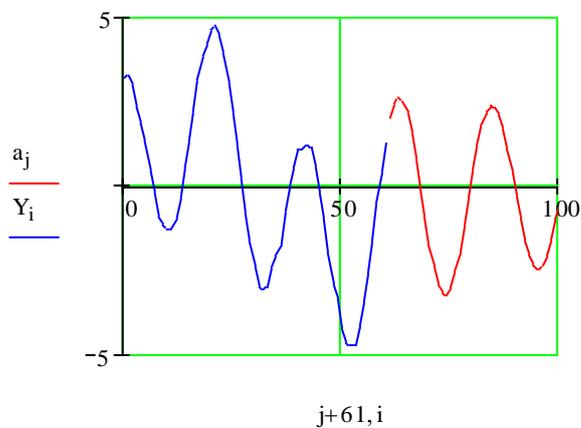
$Y := r + y$

$Y1 := r + y1$

$a := \text{predict}(Y, 7, 40)$

$b := \text{predict}(Y1, 7, 40)$

$j := 0..39$



При этом используется функция **rnorm(n,mu,sigma)**, которая возвращает вектор из n случайных чисел, имеющих нормальное распределение со средним значением μ и стандартным отклонением σ .

При незначительном среднеквадратическом отклонении шума от нулевого математического ожидания, равном 0,1 (левый график), исходный процесс имеет небольшие искажения и близок к синусоидальному. Линейная экстраполяция осуществляется с 61-й точки и, при опоре на 7 точек, дает достаточно хорошие результаты. Увеличение стандарта отклонения помехи до 0,3 (правый график) заметно искажает исходные данные и вызывает ухудшение качества результирующих значений оценок экстраполяции.

Задания

Задание 1. Функциями **medsmooth(Y,m)**, **ksmooth(X,Y,b)**, **supsmooth(X,Y)** осуществите сглаживание функции $fz(x) = f(x) + \text{pot}$ зашумленной наложением аддитивных помех pot .

Помехи формируются с помощью функции **rnorm(n,mu,sigma)**, которая возвращает вектор из n случайных чисел, имеющих нормальное распределение со средним значением μ и стандартным отклонением σ .

Результаты отобразите графически.

Проведите исследование влияния ширины m , b окна отработки на качество сглаживания.

Задание 2. Осуществите экстраполяцию (функция **predict(Y,m,k)**) значений функции $fz(x) = f(x) + \text{pot}$.

Результаты отобразите графически.

Проведите исследование влияния числа m используемых последних точек существующих данных и числа k точек, в которых необходимо предсказать данные на качество экстраполяции.

Варианты.

№	Выражения	№	Выражения
1.	$f(x) = x^2 + 2e^{\frac{x}{10}}$	16.	$f(x) = 3\cos^2 x - \sqrt{x}$
2.	$f(x) = e^x \sin x$	17.	$f(x) = 3\cos^2 x - \sqrt{x}$
3.	$f(x) = 2x + e^{4-x}$	18.	$f(x) = \sin^3 x + \cos^2 x$
4.	$f(x) = e^x - 2\sin x$	19.	$f(x) = x^2 \cos x$
5.	$f(x) = x^2 - 2^x$	20.	$f(x) = 4^x - 8x$
6.	$f(x) = 2^x - \ln x$	21.	$f(x) = x^5 - 5^x$
7.	$f(x) = x^3 - e^x$	22.	$f(x) = \ln x - 5\sin x$
8.	$f(x) = \sin x - 2\cos x$	23.	$f(x) = x^3 - e^x$
9.	$f(x) = x^2 - 2\sin x$	24.	$f(x) = \sin x + e^{-x}$
10.	$f(x) = e^x \cos x$	25.	$f(x) = \sin^2 x - \sqrt{x}$
11.	$f(x) = -3x + e^{x-1}$	26.	$f(x) = 2^x \cos x$
12.	$f(x) = x^3 - 3^x$	27.	$f(x) = e^{x-4} - 4x$
13.	$f(x) = e^x - \ln x$	28.	$f(x) = \ln x - 4^x$
14.	$f(x) = 3\cos x - \sin x$	29.	$f(x) = 2\sin x - 3\cos x$
15.	$f(x) = x^4 - e^x$	30.	$f(x) = x^6 - e^x$

Лабораторная работа

Ознакомление с пакетом Matlab. Операции с матрицами

Цель работы: ознакомление с пакетом Matlab, получение навыков работы с матрицами.

Теоретические сведения

Структура и рабочие окна пакета Matlab

Система Matlab состоит из пяти основных частей.

Язык Matlab. Это язык матриц и массивов высокого уровня с управлением потоками, функциями, структурами данных, вводом-выводом и особенностями объектно-ориентированного программирования.

Среда Matlab. Это набор инструментов и приспособлений, с которыми работает пользователь или программист Matlab. Она включает в себя средства для управления переменными в рабочем пространстве Matlab, вводом и выводом данных, а также создания, контроля и отладки М-файлов и приложений Matlab.

Управляемая графика. Это графическая система Matlab, которая включает в себя команды высокого уровня для визуализации двух- и трехмерных данных, обработки изображений, анимации и иллюстрированной графики. Она также включает в себя команды низкого уровня, позволяющие полностью редактировать внешний вид графики, также как при создании Графического Пользовательского Интерфейса (GUI) для Matlab приложений.

Библиотека математических функций. Это обширная коллекция вычислительных алгоритмов от элементарных функций, таких как сумма, синус, косинус, комплексная арифметика, до более сложных, таких как обращение матриц, нахождение собственных значений, функции Бесселя, быстрое преобразование Фурье.

Программный интерфейс. Это библиотека, которая позволяет писать программы на Си и Фортране, которые взаимодействуют с Matlab. Она включает средства для вызова программ из Matlab (динамическая связь), вызывая Matlab как вычислительный инструмент и для чтения-записи mat-файлов.

О Simulink. Simulink, сопутствующая Matlab программа, — это интерактивная система для моделирования динамических систем. Она представляет собой среду, управляемую мышью, которая позволяет моделировать процесс путем перетаскивания блоков диаграмм на экране и их манипуляцией. Simulink работает с линейными, нелинейными, непрерывными, дискретными, многомерными системами.

При загрузке системы Matlab появляется несколько окон системы.

Окно Command Window. Окно **Command Window (Окно команд)** является для пользователя наиболее важным. Посредством этого окна вводятся математические выражения, получаются результаты вычислений, а также выдаются сообщения, посылаемые системой. Данное окно становится доступным пользователю сразу же после запуска программы. Математические выражения пишутся в командной строке после знака приглашения **>>**.

Необходимо набрать в строке ввода выражение

Для выполнения действия нажать клавишу Enter.

Результат выведется ниже

Если нужно исправить одну из цифр, то ничего не получится. Невозможность редактирования ранее введенной команды простой установкой

Информационные технологии в менеджменте

курсора в нужную строку является одной из особенностей системы Matlab. Для того чтобы отредактировать ранее введенную команду, необходимо установить курсор в строку ввода и воспользоваться клавишами ↑ и ↓. Эти клавиши позволяют пролистать стек введенных ранее команд и оставить в строке ту команду, которая необходима. Команду можно выполнить сразу (нажав клавишу Enter) или после редактирования.

Понятие о файлах-сценариях. В командном режиме выполнять решения задач нецелесообразно, т. к. исправить возможные ошибки после нажатия клавиши Enter уже нельзя. Поэтому записывать программы, их редактировать и отлаживать необходимо в так называемых *m-файлах*. m-файл создается при выполнении команды **New** меню **File**. Для ускорения этой команды выведена специальная пиктограмма в виде белой странички с загнутым уголком на панели инструментов. Щелкнув по пиктограмме стрелкой мышки, получаем окно m-файла, на котором можно записывать, редактировать и отлаживать любые программы решения научных и инженерных задач. Данный m-файл по умолчанию имеет название **Untitled (Безымянный)**. Чтобы дать ему имя, необходимо в меню этого окна **File** выполнить команду **Save as** и в другом окне указать папку и имя этого файла. После указания имени и сохранения m-файла он готов для выполнения записанной программы. Для этого необходимо щелкнуть мышкой по пиктограмме **Выполнить**. Она выполнена в виде страницы со стрелкой, направленной вниз. Результат выполнения программы или сообщения об ошибках появится в окне команд. Описанный процесс называется *созданием m-файла сценария сессии*. Файл-сценарий, именуемый также *Script-файлом*, имеет простую структуру:

% Основной комментарий, если необходимо.

% Дополнительный комментарий, если необходимо.

Тело программы с любыми выражениями.

Важными являются следующие свойства файлов-сценариев:

1. Они не имеют входных и выходных аргументов.
2. Работают с данными из рабочей области.
3. В процессе выполнения не компилируются.

4. Представляют собой *последовательность операции*, аналогичную той, что используется в сессии.

Окно команд имеет еще два управляющих элемента, в правом верхнем углу. Это кнопка закрытия окна ×, и расположенная левее кнопка отделения окна от интерфейса системы ↵. После нажатия на эту кнопку окно становится автономным, имеющим собственное меню. Возврат окна в общий интерфейс происходит посредством выбора из его главного меню пунктов **View** → **Dock Command Window (Вид** → **Пристыковать окно команд)**. Такими же кнопками снабжены другие четыре окна.

Рассмотренное окно является основным и, в принципе, при работе с системой Matlab можно было бы обойтись без других.

Однако наличие остальных окон делают диалог с компьютером легким и комфортным.

Окно Workspace. В процессе работы используются переменные различных типов. Созданные переменные хранятся в специально отведенной области памяти компьютере. Они не исчезают сами по себе, а только при выходе из программы или с помощью специальных команд.

При этом переменные (точнее их значения) можно использовать в любом вводимом математическом выражении. **Окно Workspace (Рабочая область)**

Информационные технологии в менеджменте

предоставляет пользователю список всех переменных, хранящихся в рабочем пространстве. Выбрать можно любую переменную, просмотреть ее содержимое или выполнить какие-либо другие действия.

Упомянутые действия выполняются посредством контекстного меню (нужно щелкнуть правой кнопкой мыши по имени переменной в списке).

Окно Current Directory. Окно Current Directory (Текущий каталог) является аналогом известной программы Проводник, но имеет для Matlab свое особое предназначение.

Дело в том что, кроме работы с математическими выражениями из командного окна, пользователь также может работать с файлами. К тому же используемые математические функции, физически представляют собой файлы, названные по именам функции. В этих файлах записаны программы, реализующие функции. Таким образом, пользователь постоянно использует файлы. Например, указывая встроенную функцию, фактически пишется имя файла (без расширения), в котором хранится текст программы. В этом случае система будет искать требуемый файл в текущем каталоге или в пути доступа.

Окно Command History. Все команды, которые набираются в командной строке **Command Window (Окно команд)**, автоматически образуют список, который и выводится в окне **Command History (История команд)**. Чем полезен этот список? Если появилась необходимость повторить ранее выполненную команду, то она отыскивается в списке **Command History (История команд)**, и, дважды щелкнув по ней левой кнопкой мыши, можно команду выполнить.

Можно выполнить нужную последовательность команд из командной строки и получить соответствующую последовательность команд в **Command History (История команд)**. Несколькими простыми действиями полученная последовательность может быть преобразована в программу. Содержимое данного окна не теряется после выхода из системы и выключения компьютера. Удалить список команд можно только с помощью меню.

Окно Launch Pad. Окно Launch Pad (Панель запуска) содержит дерево файловой системы, где отображены только установленные на компьютере программные продукты, входящие в систему Matlab. С помощью этого окна можно запустить любой из них. Те же действия легко осуществить посредством кнопки **Start (Пуск)**, расположенной в левом нижнем углу окна Matlab.

Работа в командном режиме

В командной строке в режиме диалога можно набрать команду (оператор) или выражение и, нажав Enter, получить ответ (answer). Например, после набора команды (оператора присваивания)

a=3.2

в последующих строках появится

a = 3.2000000000000000 (переменной a присвоено значение 3.2), после набора выражения

sin(a)/a

увидим его значение

ans = -0.01824191982112

Если вы хотите выполнить команду без вывода результата, в конце команды ставьте символ точки с запятой.

Если математическое выражение выходит за размер экрана монитора, то целесообразно перенести его часть на следующую строку. Для этого используется символ многоточие ... — три и более точки. В командном режиме число возможных символов в одной строке 4096, в m-файле не ограничено, но с такими

Информационные технологии в менеджменте

длинными строками работать неудобно. Поэтому применение в файлах-сценариях символа переноса строки улучшает наглядность программ.

Любой фрагмент окна командной строки можно выделить и копировать в буфер, например, для переноса в Word. Возможен перенос в командную строку текстовых фрагментов из других систем.

Сеанс работы с Matlab принято именовать *сессией*. Сессия, в сущности, является текущим документом, отражающим работу пользователя с системой Matlab. В ней имеются строки ввода, вывода и сообщения об ошибках. Некоторые команды управления окном командного режима:

clc — очищает экран и размещает курсор в левом верхнем углу пустого экрана;

echo name_file on — включает режим вывода на экран текста Script-файла (файла-сценария);

echo name_file off — выключает режим вывода на экран текста Script-файла;

echo on all — включает режим вывода на экран текста всех m-файлов;

echo off all — выключает режим вывода на экран текста всех m-файлов;

clear — уничтожение определений всех переменных;

clear X Y — уничтожение определений переменных X и Y;

type name_file — выводит на экран текст примеров и m-файлов.

Сохранение рабочей области сессии — команда save. Переменные и определение новых функций в системе Matlab хранятся в особой области памяти, именуемой рабочей областью. Matlab позволяет сохранить значения всех переменных и определений в сессии, т. е. рабочей области, в виде бинарных файлов с расширением **.mat**. Для этого служит команда **save**, которая может использоваться в ряде форм:

save f_name — записывается рабочая область всех переменных в файл бинарного формата с именем **f_name** и расширением **.mat**;

save f_name X — записывает только значение переменной X;

save f_name X Y Z — записывает значение переменных X, Y и Z.

Следует отметить, что возможности сохранения всего текста сессии, формируемой в командном режиме, с помощью команды **save** нет. Для этого служит команда **diary**.

Ведение дневника — команда diary. Сессии не записываются на диск стандартной командой **save**. Однако если такая необходимость есть, можно воспользоваться специальной командой для ведения так называемого *дневника сессии*.

diary file_name — ведет запись на диск в виде текстового файла с указанным именем всех команд в строках ввода и получения результата;

diary off — приостанавливает запись в файл;

diary on — вновь начинает запись в файл.

Таким образом, чередуя команды **diary off** и **diary on**, можно сохранять нужные фрагменты сессии в их формальном виде.

Следующий пример в командном режиме поясняет технику применения команды **diary**:

```
>>diary myfile.txt
```

```
>>1+2
```

```
ans =
```

```
3
```

```
>>diary off
```

```
>> 2+3
ans =
5
>>diary on
>>sin(1)
ans =
0.8415
>>diary off
```

Команда **type** позволяет просматривать текст такого файла со всеми записанными действиями в режиме сессии:

```
>> type myfile.txt
1+2
ans =
3
diary off
sin(1)
ans =
0.8415
diary off
```

Загрузка рабочей области сессии – команда load. Для загрузки рабочей области ранее проведенной сессии (если она была сохранена) можно использовать команду **load**.

load f_name – загрузка ранее сохраненных в файле **fname.mat**.

Если команда **load** используется в ходе проведения сессии, то произойдет замена значений текущих переменных теми значениями, которые были сохранены в считываемом mat-файле.

Следующий пример в командном режиме поясняет технику применения команд **load** и **save**:

```
>> a = [1 2; 3 4]
a =
1 2
3 4
>> b=2+3
b =
5
>> save myfile
```

Для демонстрации работы команды **load** отчистим экран и удалим все переменные:

```
>>clc, clear
>>load myfile
>> a
a =
1 2
3 4
>> b
b =
5
```

Работа с интерактивной справочной системой. Matlab имеет интерактивную систему помощи, которая реализуется в командном режиме с помощью ряда команд. Одна из них является команда

>>help

которая выводит весь список папок (каталогов), содержащих m-файлы с определениями операторов, функций и иных объектов.

Для получения справки по какому-либо конкретному объекту используется команда

>>help имя

где имя — имя объекта (функции или операции), для которого требуется вывод справочной информации.

Ввиду обилия m-функций в системе Matlab, большое значение имеет поиск m-функций по ключевым словам. Для этого служат команда:

lookfor Ключевое слово

или

lookfor 'Ключевое слово'

В первом случае ищутся все m-файлы, в заголовках которых встречается заданное ключевое слово, а заголовки обнаруженных файлов выводятся на экран. Следует отметить, что широкий поиск по одному ключевому слову может привести к выводу подчас многих десятков определений и длится довольно долго. Для уточнения и сокращения поиска следует использовать вторую форму команды **lookfor**. Ниже приведен пример использования команды **help**:

>> help clc

CLC Clear command window.

CLC clears the command window and homes the cursor.

See also HOME.

Основные элементы языка программирования Matlab

Как и большинство других языков программирования, Matlab предоставляет возможность использования математических выражений, но в отличие от многих из них, эти выражения в Matlab включают матрицы. Основные составляющие выражения:

- переменные;
- числа;
- операторы;
- функции.

Переменные. В Matlab нет необходимости в определении типа переменных или размерности. Когда Matlab встречает новое имя переменной, он автоматически создает переменную и выделяет соответствующий объем памяти. Если переменная уже существует, Matlab изменяет ее состав и если это необходимо выделяет дополнительную память. Например,

num_students = 25

создает матрицу 1x1 с именем *num_students* и сохраняет значение 25 в ее единственном элементе.

Имена переменных состоят из букв, цифр или символов подчеркивания. Matlab использует только первые 31 символ имени переменной. Matlab чувствителен к регистрам, он различает заглавные и строчные буквы. Поэтому *A* и *a* — не одна и та же переменная. Чтобы увидеть матрицу, связанную с переменной, просто введите название переменной.

Числа. Matlab использует принятую десятичную систему счисления, с необязательной десятичной точкой и знаками плюс-минус для чисел. Научная система счисления использует букву **e** для определения множителя степени десяти. Мнимые числа используют **i** или **j** как суффикс. Некоторые примеры правильных чисел приведены ниже:

Информационные технологии в менеджменте

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

Все числа для хранения используют формат **long**, это числа с плавающей точкой обладающие ограниченной точностью — приблизительно 16 значащих цифр и ограниченным диапазоном — приблизительно от 10^{-308} до 10^{308} .

Операторы. Выражения используют обычные арифметические операции и правила старшинства (табл. 1).

Таблица 1

Арифметические операции пакета Matlab

Функции	Название	Оператор	Синтаксис
<i>plus</i>	Плюс	+	$M1+M2$
<i>uplus</i>	Унарный плюс	+	$+M$
<i>minus</i>	Минус	-	$M1-M2$
<i>uminus</i>	Унарный минус	-	$-M$
<i>mtimes</i>	Матричное умножение	*	$M1*M2$
<i>times</i>	Почленное умножение массивов	.*	$M1.*M2$
<i>mpower</i>	Возведение в степень матрицы	^	M^x
<i>power</i>	Возведение в степень массива почленное	.^	$M.^x$
<i>mldivide</i>	Обратное (справа налево) деление матриц	\	$M1\M2$
<i>mrdivide</i>	Деление матриц слева направо	/	$M1/M2$
<i>ldivide</i>	Деление почленное массивов справа налево	.\	$M1.\M2$
<i>rdivide</i>	Деление почленное массивов слева направо	./	$M1./M2$

Функции. Matlab предоставляет большое количество элементарных математических функций, таких как **abs**, **sqrt**, **exp**, **sin**. Вычисление квадратного корня или логарифма отрицательного числа не является ошибкой: в этом случае результатом является соответствующее комплексное число. Matlab также предоставляет и более сложные функции, включая Гамма функцию и функции Бесселя. Большинство из этих функций имеют комплексные аргументы. Чтобы вывести список всех элементарных математических функций, наберите:

help elfun

Чтобы посмотреть список всех функций Matlab для анализа данных:

help datafun

Если вам нужно узнать о **Statistics Toolbox**, введите:

help stats

Список элементарных функций представлен в табл. 2.

Элементарные функции пакета Matlab

Функция	Синтаксис
$ x $ – модуль	<i>abs(x)</i>
e^x – экспонента	<i>exp(x)</i>
$\ln x$ – натуральный логарифм	<i>log(x)</i>
$\log_2 x$ – логарифм по основанию 2	<i>log2(x)</i>
$\lg x$ – десятичный логарифм	<i>log10(x)</i>
2^x – 2 в степени x	<i>pow(x)</i>
\sqrt{x} – квадратный корень	<i>sqrt(x)</i>
<i>arccos</i> x – арккосинус	<i>acos(x)</i>
<i>arcctg</i> x – арккотангенс	<i>acot(x)</i>
<i>arccosec</i> x – арккосеканс	<i>acsc(x)</i>
<i>arcces</i> x – арксеканс	<i>asec(x)</i>
<i>arcsin</i> x – арксинус	<i>asin(x)</i>
<i>arctg</i> x – арктангенс	<i>atan(x)</i>
<i>cos</i> x – косинус	<i>cos(x)</i>
<i>ctg</i> x – котангенс	<i>cot(x)</i>
<i>sec</i> x – секанс	<i>sec(x)</i>
<i>cosec</i> x – косеканс	<i>csc(x)</i>
<i>sin</i> x – синус	<i>sin(x)</i>
<i>tg</i> x – тангенс	<i>tan(x)</i>
<i>arcch</i> x – арккосинус гиперболический	<i>acosh(x)</i>
<i>arccth</i> x – арккотангенс гиперболический	<i>acoth(x)</i>
<i>arccosech</i> x – арккосеканс гиперболический	<i>acsch(x)</i>
<i>arcsech</i> x – арксеканс гиперболический	<i>asech(x)</i>
Функция	Синтаксис
<i>arcsh</i> x – арккосинус гиперболический	<i>asinh(x)</i>
<i>artgh</i> x – арктангенс гиперболический	<i>atanh(x)</i>
<i>ch</i> x – косинус гиперболический	<i>cosh(x)</i>
<i>ctgh</i> x – котангенс гиперболический	<i>coth(x)</i>
<i>cosech</i> x – косеканс гиперболический	<i>csch(x)</i>
<i>sech</i> x – секанс гиперболический	<i>sech(x)</i>
<i>sh</i> x – синус гиперболический	<i>sinh(x)</i>
<i>tgh</i> x – тангенс гиперболический	<i>tanh(x)</i>

$$\text{Логарифм числа } b \text{ по основанию } a: \log_a b = \frac{\log_c b}{\log_c a}.$$

Для вывода более сложных математических и матричных функций, наберите:

help specfun

help elmat

соответственно.

Некоторые функции, такие как **sqrt** и **sin**, — встроенные. Они являются частью Matlab, поэтому они очень эффективны, но их вычислительные детали трудно доступны. В то время как другие функции, такие как **gamma** и **sinh**, реализованы в m-файлах. Поэтому можно увидеть их код и, в случае необходимости, даже модифицировать его.

Несколько специальных функций предоставляют значения часто используемых констант:

pi	Число 3,14159265...
i или j	Мнимая единица $\sqrt{-1}$
eps	Относительная точность числа с плавающей точкой
realmin	Наименьшее число с плавающей точкой
realmax	Наибольшее число с плавающей точкой
Inf	Бесконечность
NaN	Не число (неопределенность вида 0/0 или ∞/∞)

Бесконечность появляется при делении на ноль или при выполнении математического выражения, приводящего к переполнению, т. е. к превышению **realmax**. Не число (**NaN**) генерируется при вычислении выражений типа 0/0 или **Inf/Inf**, которые не имеют определенного математического значения.

Имена функций не являются зарезервированными, поэтому возможно изменять их значения на новые, например:

eps = 1.e-6

и далее использовать это значение в последующих вычислениях. Начальное значение может быть восстановлено следующим образом:

clear eps

Операторы отношения служат для сравнения двух величин, векторов или матриц, все операторы отношения имеют две сравниваемые величины и записываются, как показано в табл. 3.

Таблица 3

Операторы отношения пакета Matlab

Функция	Оператор (синтаксис)
Равно	== (x == y)
Не равно	~= (x ~= y)
Меньше	< (x < y)
Больше	> (x > y)
Меньше или равно	<= (x <= y)
Больше или равно	>= (x >= y)

Данные операторы выполняют поэлементное сравнение векторов или матриц одинакового размера, и логическое выражение принимает значение 1 (True), если элементы идентичны, и значение 0 (False) в противном случае.

Логические операторы служат для реализации поэлементных логических операций над элементами одинаковых по размеру массивов согласно [табл. 4](#).

Таблица 4

Логические операторы пакета Matlab

Функция	Оператор (синтаксис)
Логическое И	<code>&</code> ; <i>and</i> (<i>and</i> (<i>a</i> , <i>b</i>))
Логическое ИЛИ	<code> </code> ; <i>or</i> (<i>or</i> (<i>a</i> , <i>b</i>))
Логическое НЕ	<code>~</code> ; <i>not</i> (<i>not</i> (<i>a</i> , <i>b</i>))
Исключающее ИЛИ	<i>xor</i> (<i>xor</i> (<i>a</i> , <i>b</i>))
Верно, если все элементы вектора равны нулю	<i>any</i> (<i>any</i> (<i>a</i>))
Верно, если все элементы вектора не равны нулю	<i>all</i> (<i>all</i> (<i>a</i>))

Текстовые комментарии. Поскольку Matlab используется для достаточно сложных вычислений, большое значение имеет наглядность их описания. Она достигается, в частности, использованием текстовых комментариев. *Текстовые комментарии* вводятся с помощью оператора — символа `%`:

% Ниже представлено задание функции вычисления факториала

Интересные факты:

- Если в командной строке напечатать «why» (почему), MATLAB ответит на этот вопрос. Ответы выбираются из ограниченного множества ответов случайным образом всякий раз, когда вводится эта команда.
- В MATLAB встроена программа виртуальной машины `vcrar`.

Ввод матриц

Лучший способ начать работу с Matlab — это научиться обращаться с матрицами. В Matlab *матрица* — это прямоугольный массив чисел. Особое значение придается матрицам 1x1, которые являются скалярами, и матрицам, имеющим один столбец или одну строку, — векторам.

Хороший пример матрицы, можно найти на гравюре времен Ренессанса художника и любителя математики Альбрехта Дюрера ([рис. 1, а](#)). Это изображение содержит много математических символов, и если хорошо присмотреться, то в верхнем правом углу можно заметить квадратную матрицу ([рис. 1, б](#)). Это матрица известна как магический квадрат и во времена Дюрера считалось, что она обладает магическими свойствами. Она и на самом деле обладает замечательными свойствами, стоящими изучения.

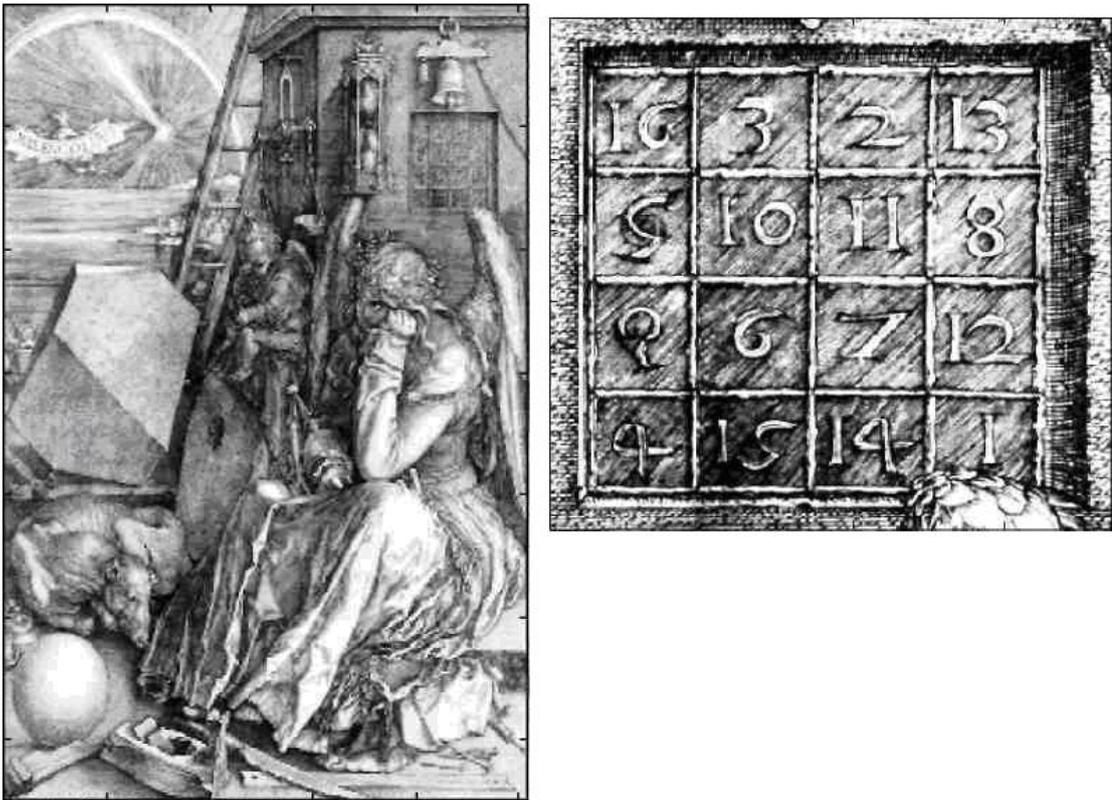


Рис. 1. Гравюра времен Ренессанса художника Альбрехта Дюрера (а), магический квадрат (б)

Матрицы в Matlab можно вводить несколькими способами:

- вводить полный список элементов;
- генерировать матрицы, используя встроенные функции;
- загружать матрицы из внешних файлов;
- создавать матрицы с помощью собственных функций в m-файлах.

1. Ввод матриц как списков элементов. Необходимо следовать условиям:

- отделять элементы строки пробелами или запятыми;
- использовать точку с запятой ; для обозначения окончания каждой строки;
- окружать весь список элементов квадратными скобками [].

Чтобы ввести матрицу Дюрера запишем:

A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]

Matlab отобразит матрицу:

```
A =
    16     3     2    13
     5    10    11     8
     9     6     7    12
     4    15    14     1
```

Особые свойства магического квадрата связаны с различными способами суммирования его элементов. Если найти сумму элементов вдоль какой-либо строки или столбца, или вдоль какой-либо из двух главных диагоналей, то получится одно и то же число:

sum(A) — сумма элементов столбцов матрицы A.

Matlab выдаст ответ:

```
ans =
    34    34    34    34
```

Информационные технологии в менеджменте

Когда выходная переменная не определена, Matlab использует переменную **ans**, коротко от answer — ответ, для хранения результатов вычисления.

Matlab предпочитает работать со столбцами матрицы, таким образом, лучший способ получить сумму в строках — это транспонировать матрицу, подсчитать сумму в столбцах, а потом транспонировать результат. Операция транспонирования обозначается апострофом. Она зеркально отображает матрицу относительно главной диагонали и меняет строки на столбцы. Таким образом:

A'

вызывает:

ans =

```

16    5    9    4
 3   10    6   15
 2   11    7   14
13    8   12    1

```

А выражение:

sum(A)'

вызывает результат вектор-столбец, содержащий суммы в строках:

ans =

```

34
34
34
34

```

Сумму элементов на главной диагонали можно получить с помощью функции **diag**, которая выбирает эту диагональ:

diag(A)

ans =

```

16
10
 7
 1

```

А функция:

sum(diag(A))

вызывает:

ans = 34

Другая диагональ, называемая антидиагональю, не так важна математически, поэтому Matlab не имеет специальной функции для нее. Но функция, которая вначале предполагалась для использования в графике, **fliplr**, зеркально отображает матрицу слева направо:

sum(diag(fliplr(A)))

ans = 34

Таким образом, доказано, что матрица на гравюре Дюрера действительно магическая.

Функция magic. Matlab на самом деле обладает встроенной функцией, которая создает магический квадрат почти любого размера. Не удивительно, что эта функция называется **magic**:

B=magic(4)

B =

```

16    2    3   13
 5   11   10    8
 9    7    6   12

```

4 14 15 1

Эта матрица почти та же матрица, что и на гравюре Дюрера, и она имеет все те же магические свойства. Единственное отличие заключается в том, что два средних столбца поменялись местами. Для того чтобы преобразовать B в матрицу Дюрера A , переставим их местами:

$A=B(:, [1\ 3\ 2\ 4])$

Это означает, что для каждой строки матрицы B элементы переписываются в порядке 1, 3, 2, 4:

$A =$

```

16    3    2   13
5   10 11    8
9    6 7   12
4   15 14    1
    
```

Почему Дюрер переупорядочил столбцы, по сравнению с тем, что использует Matlab? Без сомнения, он хотел включить дату гравюры, 1514, в нижнюю часть магического квадрата.

2. Генерирование матриц с использованием встроенных функций.

Matlab имеет четыре функции, которые создают основные матрицы:

zeros(m, n)	Матрица размерности $m \times n$ с нулевыми элементами
ones(m, n)	Все единицы
rand(m, n)	Равномерное распределение случайных элементов
randn(m, n)	Нормальное распределение случайных элементов

Пример 1.

```
Z = zeros(2,4)
```

```
Z =
    0    0    0    0
    0    0    0    0
```

```
F = 5*ones(3,3)
```

```
F =
    5    5    5
    5    5    5
    5    5    5
```

```
R = randn(4,4)
```

```
R =
 -0.4326  -1.1465    0.3273  -0.5883
 -1.6656    1.1909    0.1746    2.1832
  0.1253    1.1892   -0.1867   -0.1364
  0.2877   -0.0376    0.7258    0.1139
```

3. Загрузка матриц из внешних файлов. Команда **load** считывает двоичные файлы, содержащие матрицы, созданные в Matlab ранее, или текстовые файлы, содержащие численные данные. Текстовые файлы должны быть сформированы в виде прямоугольной таблицы чисел, отделенных пробелами, с равным количеством элементов в каждой строке. Например, создадим в Блокноте текстовый файл, содержащий 4 строки:

Информационные технологии в менеджменте

16.0	3.0	2.0	13.0
5.0	10.0	11.0	8.0
9.0	6.0	7.0	12.0
4.0	15.0	14.0	1.0

Сохраним этот файл под именем **magik.dat**. Тогда команда:

load magik.dat

прочитает этот файл и создаст переменную **magik**, содержащую матрицу.

4. Создание матриц с помощью собственных функций в m-файлах.

Создадим файл, включающий следующие пять строк:

```
A = [16.0 3.0 2.0 13.0
      5.0 10.0 11.0 8.0
      9.0 6.0 7.0 12.0
      4.0 15.0 14.0 1.0];
```

Сохраним его под именем **magik.m**. Тогда выражение

magik

прочитает файл и создаст переменную **A**, содержащую исходную матрицу.

Объединение — это процесс соединения маленьких матриц для создания больших. Пара квадратных скобок — это оператор объединения. Например, начнем с матрицы **A** (магического квадрата 4×4) и сформируем:

B = [A A+32; A+48 A+16]

Результатом будет матрица 8×8 , получаемая соединением четырех подматриц:

```
B =
    16     2     3    13    48    34    35    45
     5    11    10     8    37    43    42    40
     9     7     6    12    41    39    38    44
     4    14    15     1    36    46    47    33
    64    50    51    61    32    18    19    29
    53    59    58    56    21    27    26    24
    57    55    54    60    25    23    22    28
    52    62    63    49    20    30    31    17
```

Индексы элементов матриц

Элемент в строке **i** и столбце **j** матрицы **A** обозначается **A(i,j)**. Например, **A(4,2)** — это число в четвертой строке и втором столбце. Для нашего магического квадрата **A(4,2) = 15**. Таким образом, можно вычислить сумму элементов в четвертом столбце матрицы **A**, набрав:

A(1,4) + A(2,4) + A(3,4) + A(4,4)

получим

ans = 34

Однако это не самый лучший способ суммирования отдельной строки. Также возможно обращаться к элементам матрицы через один индекс, **A(k)**. В этом случае массив рассматривается как длинный вектор, сформированный из столбцов исходной матрицы.

Так, для магического квадрата, **A(8)** — это другой способ сослаться на значение 15, хранящееся в **A(4,2)**.

Если использовать значение элемента вне матрицы, Matlab выдаст ошибку:

t=A(4,5)

??? Index exceeds matrix dimensions.

Информационные технологии в менеджменте

С другой стороны, если сохранить значение вне матрицы, то размер матрицы увеличивается:

X=A;

X(4,5) = 17

X =

```

16   3   2  13   0
 5  10  11   8   0
 9   6   7  12   0
 4  15  14   1  17
    
```

Оператор двоеточия. Двоеточие : — это один из наиболее важных операторов Matlab. Он проявляется в различных формах. Выражение:

1:10

— это вектор-строка, содержащая целые числа от 1 до 10:

1 2 3 4 5 6 7 8 9 10

Для получения обратного интервала:

100:-7:50

что дает:

100 93 86 79 72 65 58 51

или:

0:pi/4:pi

что приводит к:

0 0.7854 1.5708 2.3562 3.1416

Индексное выражение, включая двоеточие, относится к части матрицы:

A(1:k, j) это первые *k* элементов *j*-го столбца матрицы *A*. Так:

sum(A(1:4,4))

вычисляет сумму четвертой строки. Но есть и лучший способ. Двоеточие, само по себе, обращается ко всем элементам в строке и столбце матрицы, а слово **end** — к последней строке или столбцу. Так:

sum(A(:,end))

вычисляет сумму элементов в последнем столбце матрицы *A*:

ans = 34

Почему магическая сумма квадрата 4×4 равна 34? Если целые числа от 1 до 16 отсортированы в четыре группы с равными суммами, эта сумма должна быть:

sum(1:16)/4

которая равна:

ans = 34

Матричные функции линейной алгебры

Математические операции, определенные на матрицах являются объектом линейной алгебры (табл. 1).

Таблица 1

Матричные функции линейной алгебры

det(A) — возвращает определитель квадратной матрицы <i>A</i> .
rank(A) — возвращает ранг матрицы <i>A</i> .
trace(A) — возвращает след матрицы <i>A</i> .
inv(A) — возвращает матрицу, обратную квадратной матрицы <i>A</i> .
size(A) — возвращает вектор-строку, содержащую количество строк и столбцов в массиве <i>A</i> .
length(X) — возвращает длину вектора <i>X</i> .

max(A) — возвращает вектор-строку, содержащую значения максимальных элементов в столбцах матрицы *A*.

min(A) — возвращает вектор-строку, содержащую значения минимальных элементов в столбцах матрицы *A*.

Массивы

Когда мы выходим из мира линейной алгебры, матрицы становятся *двумерными численными массивами*. Арифметические операции на массивах производятся поэлементно. Это означает, что суммирование и вычитание являются одинаковыми операциями для матриц и массивов, а умножение для них различно. Matlab использует точку, как часть записи для операции умножения массивов.

Список операторов включает в себя:

+	суммирование
-	вычитание
.*	поэлементное умножение
./	поэлементное деление
.\	поэлементное левое деление
.^	поэлементное возведение в степень
!	несопряженное матричное транспонирование

Если магический квадрат Дюрера умножить на себя по правилам умножения массивов:

A.*A

результатом будет массив, содержащий квадраты целых чисел от 1 до 16:

```
ans =
    256     9     4    169
     25    100    121     64
     81     36     49    144
     16    225    196     1
```

Операции над массивами полезны для создания таблиц. Пусть *n* — это вектор-столбец:

n = (0:9)';

Тогда:

pows = [n n.^2 2.^n]

создает таблицу квадратов и степеней двойки:

```
pows =
     0     0     1
     1     1     2
     2     4     4
     3     9     8
     4    16    16
     5    25    32
     6    36    64
     7    49   128
     8    64   256
     9    81   512
```

Элементарные математические функции работают с массивами поэлементно. Так:

x = (1:0.1:2)'

$s = [x \sin(x)]$

создает таблицу значений синуса.

Задания

Все задания рекомендуется выполнять в m-файлах.

Задание 1. Определите переменную d , значением которой является выражение:

1., 20.
$$\left(\frac{x^2 - y^2}{x - y} - \frac{x^2 + y^2}{x + y} \right) \cdot \frac{x^2 - y^2}{2xy}$$

2., 21.
$$\left(\frac{x^3 - 8}{x^2 + 4 + 2x} + 4 \right) \cdot \frac{1}{x + 2}$$

3., 22.
$$\frac{4x(x + \sqrt{x^2 - 1})^2}{(x + \sqrt{x^2 - 1})^4 - 1}$$

4., 23.
$$\left(\frac{2}{2x + y} - \frac{1}{2x - y} - \frac{3y}{y^2 - 4x^2} \right) \cdot \left(\frac{y^2}{8x^2} - 0,5 \right)$$

5., 24.
$$\left(\frac{a}{a + \sqrt{ab}} - \frac{b}{\sqrt{ab} + b} \right) \left(\frac{\sqrt{a}}{\sqrt{b}} - \frac{\sqrt{b}}{\sqrt{a}} \right) \sqrt{ab}$$

6., 25.
$$\frac{(a - b)^3 (\sqrt{a} + \sqrt{b})^{-3} + 2a\sqrt{a} + b\sqrt{b}}{a\sqrt{a} + b\sqrt{b}} + \frac{3(\sqrt{ab} - b)}{a - b}$$

7., 26.
$$\left(p - q + \frac{4q^2 - p^2}{p + q} \right) \left(\frac{p}{p^2 - q^2} + \frac{2}{q - p} + \frac{1}{p + q} \right)$$

8., 27.
$$\frac{0,25(a^{-3/2} - a^{3/2})}{a^{-1} + a + 1} \left(\frac{\sqrt{a} - 1}{\sqrt{a} + 1} - \frac{\sqrt{a} + 1}{\sqrt{a} - 1} \right)$$

9., 28.
$$(\sqrt{1 - x^2} + 1) \left(\frac{1}{\sqrt{1 + x}} + \sqrt{1 - x} \right)$$

10., 29.
$$\left(\frac{a\sqrt{a} + b\sqrt{b}}{\sqrt{a} + \sqrt{b}} - \sqrt{ab} \right) : (a - b)$$

11., 30.
$$\left(\frac{a\sqrt{a} + b\sqrt{b}}{\sqrt{a} + \sqrt{b}} - \sqrt{ab} \right) \left(\frac{\sqrt{a} + \sqrt{b}}{a - b} \right)^2$$

12.
$$\left(\frac{a}{a - b} - \frac{b}{a + b} \right) \left(\frac{a + b}{b} - \frac{a - b}{a} \right)$$

Информационные технологии в менеджменте

13. $\left(\frac{\sqrt{a}+2}{(\sqrt{a}+1)^2} - \frac{\sqrt{a}-2}{a-1} \right) \frac{\sqrt{a}+1}{\sqrt{a}}$
14. $\left(\frac{a}{a-1} + \frac{1}{ab-b} \right) \left(\frac{ab+1}{b} \right)^{-1} - \frac{a}{a^2-1}$
15. $\frac{2b\sqrt{x^2-1}}{x-\sqrt{x^2-1}} = A$, где $x = \frac{1}{2} \left(\sqrt{\frac{a}{b}} + \sqrt{\frac{b}{a}} \right)$
16. $\left(\frac{\sqrt[4]{ab^3} - \sqrt[4]{a^3b} + \sqrt{ab+1}}{\sqrt{a}-\sqrt{b}} + \frac{\sqrt{ab+1}}{\sqrt[4]{ab}} \right)^{-2} \cdot \sqrt{1 + \frac{a}{b} + 2\sqrt{\frac{a}{b}}}$
17. $\left(\left(\frac{\sqrt{a} + \sqrt{b}}{\sqrt[4]{a} - \sqrt[4]{b}} \right)^{-1} - \frac{2^4 \sqrt{ab}}{b^{\frac{3}{4}} - a^{\frac{1}{4}} b^{\frac{1}{4}} + a^{\frac{1}{4}} b^{\frac{3}{4}} - a^{\frac{3}{4}}} \right)^{-1}$
18. $\left(\frac{\sqrt{a^3} - \sqrt{b^3}}{\sqrt{a} - \sqrt{b}} + \sqrt{ab} \right) \left(\frac{\sqrt{a} - \sqrt{b}}{a-b} \right)^2$
19. $\left(2x+1 - \frac{1}{1-2x} \right) \left(2x - \frac{4x^2}{2x-1} \right)$

Предварительно необходимо определить переменные, входящие в выражение, выбрав их значения произвольно.

Задание 2. Определите переменную h , значением которой является выражение:

№	Выражение	№	Выражение
1.	$h = (xy^2 + x) + (x^2y - y)$	16.	$h = x^2y + 2xy + \ln x$
2.	$h = y \sin^2 x + x \ln y$	17.	$h = xy + y - 2y^2 \ln x$
3.	$h = (x + 2y) - \lg(xy)$	18.	$h = -y / \sqrt{x^2 - y^2}$
4.	$h = x^2y - (2xy - y^2)x$	19.	$h = x(y^2 - 3) + (x^2y - 2\sqrt{y})$
5.	$h = (1 + x^2)y - 2xy + (1 + x^2)^2$	20.	$h = y \cos x + (y + 1) \sin x$
6.	$h = (xy^5 - 1) \ln x + 2y$	21.	$h = y \operatorname{ctg} x + y + 2$
7.	$h = xy - 3y - x^4y^2$	22.	$h = \cos^2 x \cdot \operatorname{ctg} y + \sin^2 y \cdot \operatorname{tg} x$
8.	$h = 4x^3e^y + y^4e^x + 4y^3e^x$	23.	$h = xy + y(\ln y - \ln x)$
9.	$h = \cos y \cdot \sin x + \sin y \cdot \cos x$	24.	$h = (x^2 - 2y^2)x + 2xy$
10.	$h = (x^2 - y^2)y + 2xy$	25.	$h = xy^{2-y} + y - x - 1$
11.	$h = y \cos x - \frac{y}{\ln y}$	26.	$h = y - y \operatorname{ctg} x + \frac{1}{\sin x}$
12.	$h = 3e^x \operatorname{tg} y + (1 - e^x) \cos^{-2} y$	27.	$h = x^2y - 2xy + y^2$

Информационные технологии в менеджменте

13.	$h = xy + y - xe^{\frac{y}{x}}$	28.	$h = 2x \cos^2 y + 2y - x^2$
14.	$h = y^2 + x^2 y + xy$	29.	$h = \left(1 + e^{\frac{x}{y}}\right) x + e^{\frac{x}{y}} (1 - x/y)y$
15.	$h = x^2 y + 2xy + 3$	30.	$h = xy + y - y^2$

Предварительно необходимо определить переменные, входящие в выражение, выбрав их значения произвольно.

Задание 3. Введите матрицу A , заданную в индивидуальном варианте. Определите число столбцов и строк матрицы. Определите максимальный и минимальный элемент. Найдите детерминант (определитель матрицы), след, ранг. Выполните операции транспонирования и нахождения обратной матрицы.

Варианты.

$$1., 17. \quad A = \begin{pmatrix} 1 & 2 & 12 & 9 & 14 \\ 2 & 3 & 7 & 15 & 29 \\ 11 & 6 & 5 & 23 & 16 \\ 27 & 4 & 17 & 18 & 27 \\ 13 & 9 & 25 & 6 & 19 \end{pmatrix} \quad 2., 18. \quad A = \begin{pmatrix} 31 & 18 & 9 & 12 & 24 \\ 29 & 14 & 8 & 21 & 9 \\ 15 & 7 & 6 & 32 & 13 \\ 17 & 19 & 33 & 27 & 25 \\ 11 & 22 & 16 & 35 & 23 \end{pmatrix}$$

$$3., 19. \quad A = \begin{pmatrix} 31 & 12 & 12 & 9 & 14 \\ 32 & 31 & 7 & 15 & 29 \\ 41 & 16 & 5 & 23 & 16 \\ 57 & 14 & 17 & 18 & 27 \\ 23 & 19 & 25 & 6 & 19 \end{pmatrix} \quad 4., 20. \quad A = \begin{pmatrix} 44 & 33 & 19 & 72 & 53 \\ 67 & 14 & 43 & 21 & 9 \\ 15 & 71 & 22 & 32 & 13 \\ 17 & 66 & 45 & 27 & 25 \\ 11 & 15 & 46 & 35 & 23 \end{pmatrix}$$

$$5., 21. \quad A = \begin{pmatrix} 52 & 2 & 12 & 9 & 14 \\ 21 & 3 & 7 & 15 & 29 \\ 11 & 6 & 5 & 23 & 16 \\ 27 & 4 & 17 & 18 & 27 \\ 13 & 9 & 25 & 6 & 19 \end{pmatrix} \quad 6., 22. \quad A = \begin{pmatrix} 31 & 18 & 49 & 12 & 24 \\ 29 & 14 & 58 & 21 & 19 \\ 15 & 17 & 26 & 32 & 13 \\ 17 & 19 & 73 & 27 & 25 \\ 11 & 22 & 56 & 35 & 23 \end{pmatrix}$$

Информационные технологии в менеджменте

$$\begin{array}{l}
 7., 23. \quad A = \begin{pmatrix} 20 & 50 & 12 & 9 & 14 \\ 50 & 31 & 7 & 15 & 9 \\ 11 & 6 & 51 & 23 & 16 \\ 27 & 24 & 17 & 18 & 27 \\ 13 & 9 & 25 & 6 & 19 \end{pmatrix} \\
 8., 24. \quad A = \begin{pmatrix} 75 & 15 & 90 & 12 & 24 \\ 29 & 14 & 81 & 21 & 9 \\ 15 & 7 & 63 & 32 & 13 \\ 17 & 19 & 33 & 27 & 25 \\ 11 & 22 & 16 & 35 & 23 \end{pmatrix} \\
 9., 25. \quad A = \begin{pmatrix} 11 & 2 & 12 & 9 & 14 \\ 2 & 13 & 7 & 15 & 22 \\ 11 & 6 & 51 & 23 & 16 \\ 27 & 4 & 17 & 18 & 27 \\ 23 & 13 & 25 & 6 & 29 \end{pmatrix} \\
 10., 26. \quad A = \begin{pmatrix} 39 & 18 & 9 & 12 & 24 \\ 29 & 14 & 38 & 21 & 44 \\ 15 & 7 & 56 & 32 & 13 \\ 17 & 19 & 33 & 27 & 25 \\ 11 & 22 & 16 & 35 & 23 \end{pmatrix} \\
 11., 27. \quad A = \begin{pmatrix} 26 & 2 & 12 & 9 & 14 \\ 2 & 3 & 7 & 15 & 29 \\ 11 & 6 & 5 & 23 & 16 \\ 27 & 4 & 17 & 18 & 27 \\ 13 & 82 & 25 & 6 & 11 \end{pmatrix} \\
 12., 28. \quad A = \begin{pmatrix} 67 & 18 & 69 & 12 & 24 \\ 29 & 14 & 8 & 21 & 9 \\ 15 & 7 & 6 & 32 & 13 \\ 17 & 19 & 33 & 27 & 25 \\ 11 & 22 & 16 & 35 & 23 \end{pmatrix} \\
 13., 29. \quad A = \begin{pmatrix} 63 & 18 & 49 & 12 & 24 \\ 72 & 14 & 48 & 21 & 9 \\ 15 & 67 & 46 & 32 & 13 \\ 17 & 19 & 53 & 27 & 25 \\ 11 & 22 & 56 & 35 & 23 \end{pmatrix} \\
 14., 30. \quad A = \begin{pmatrix} 19 & 12 & 12 & 9 & 14 \\ 12 & 32 & 17 & 15 & 25 \\ 1 & 16 & 5 & 23 & 16 \\ 27 & 14 & 17 & 18 & 25 \\ 35 & 19 & 25 & 61 & 19 \end{pmatrix}
 \end{array}$$

$$15. \quad A = \begin{pmatrix} 41 & 18 & 59 & 11 & 14 \\ 39 & 14 & 8 & 21 & 23 \\ 15 & 55 & 6 & 32 & 13 \\ 17 & 19 & 33 & 27 & 25 \\ 11 & 22 & 16 & 35 & 53 \end{pmatrix} \quad 16. \quad A = \begin{pmatrix} 87 & 54 & 12 & 67 & 75 \\ 54 & 48 & 81 & 15 & 21 \\ 11 & 6 & 46 & 23 & 16 \\ 27 & 51 & 17 & 18 & 27 \\ 13 & 80 & 25 & 50 & 22 \end{pmatrix}$$

Задание 4. Задайте вектор b , который представляет собой j -й столбец матрицы A . Определите длину вектора (число элементов).

Затем прибавьте к нему вектор такой же размерности с произвольно выбранными элементами и покажите результат операции. Прделайте то же для вычитания и умножения. Примените к вектору b оператор транспонирования.

Задания 5. Задайте массив C такой же размерности как массив A с произвольными значениями элементов. Над массивами A и C проведите поэлементные операции умножения, деления и возведения в степень. Чем эти операции отличаются от операций умножения, деления и возведения в степень обычных матриц?

Лабораторная работа Matlab: Элементы графики

Цель работы: получение навыков использования графических возможностей пакета Matlab.

Теоретические сведения

Создание графика

Matlab имеет широкие возможности для графического изображения векторов и матриц, а также для создания комментариев и печати графики.

Функция **plot** имеет различные формы, связанные с входными параметрами, например **plot(y)** создает кусочно-линейный график зависимости элементов y от их индексов. Если вы задаете два вектора в качестве аргументов, **plot(x,y)** создаст график зависимости y от x .

Например, построение графика значений функции **sin** от 0 до 2π (рис. 1):

```
t = 0:pi/100:2*pi;
y = sin(t);
plot(t,y)
```

Информационные технологии в менеджменте

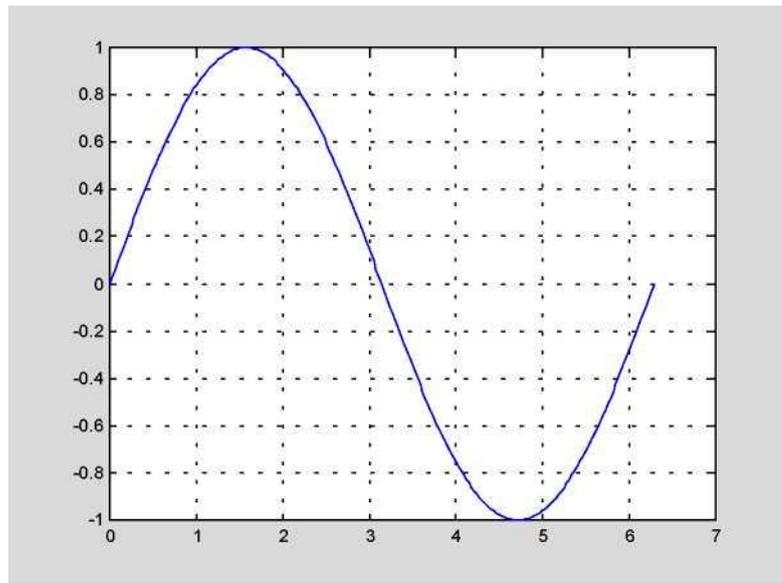


Рис. 1.

Вызов функции **plot** с многочисленными парами x - y создает многочисленные графики. Matlab автоматически присваивает каждому графику свой цвет (исключая случаи, когда это делает пользователь), что позволяет различать заданные наборы данных. Например, следующие три строки отображают график близких функций, и каждой кривой соответствует свой цвет (рис. 2):

```
y2 = sin(t-.25);
```

```
y3 = sin(t-.5);
```

```
plot(t, y, t, y2, t, y3)
```

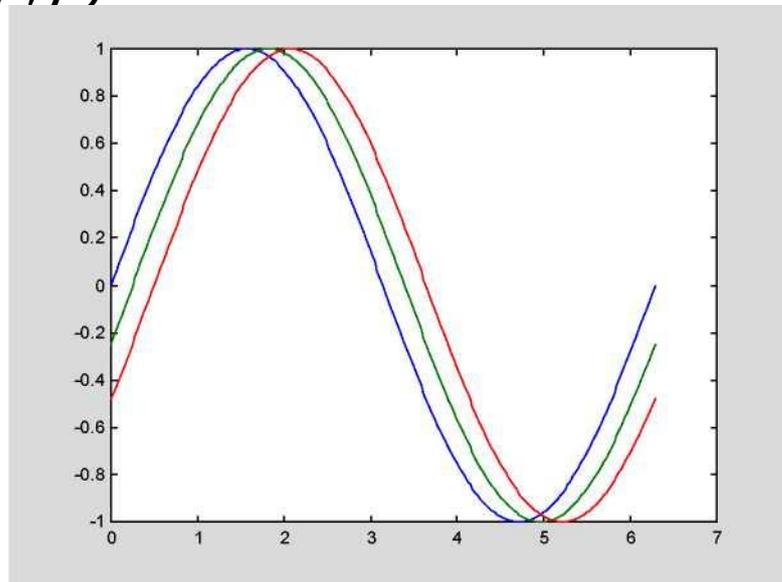


Рис. 2.

Функция **semilogx(x,y)** строит график $y(x)$ в логарифмическом масштабе по оси x , векторы x и y должны иметь одинаковые размеры.

Возможно изменение цвета, стиля линий и маркеров, таких как знаки плюс или кружки, следующим образом (табл. 1):

```
plot(x, y, 'цвет стиль маркер')
```

цвет стиль маркер это 1-, 2-, 3- символьная строка (заклученная в одинарные кавычки), составленная из типов цвета, стиля линий и маркеров:

— символы, относящие к цвету: 'c', 'm', 'y', 'r', 'g', 'b', 'w' и 'k'. Они обозна-

Информационные технологии в менеджменте

чают голубой, малиновый, желтый, красный, зеленый, синий, белый и черный цвета соответственно;

— символы, относящиеся к типу линий: ' - ' для сплошной, ' -- ' для разрывной, ':' для пунктирной, ' - . ' для штрихпунктирной линий и ' none ' для её отсутствия;

— наиболее часто встречающиеся маркеры ' + ', ' o ', ' * ' и ' x '.

Таблица 1

Стили графиков

Цвет линии	Тип точки	Тип линии
$s = y$ – желтый	$s = \bullet$ – точка	$s = -$ – сплошная
m – фиолетовый	\bigcirc – окружность	$:$ – двойной пунктир
c – голубой	x – крест	$- .$ – штрихпунктир
r – красный	$+$ – плюс	$--$ – штриховая
g – зеленый	$*$ – звездочка	
b – синий	s – квадрат	
w – белый	d – ромб	
k – черный	∇ – треугольник (вниз)	
	\wedge – треугольник (вверх)	
	$<$ – треугольник (влево)	
	$>$ – треугольник (вправо)	
	p – пятиугольник	
	h – шестиугольник	

Например, выражение:

plot(x,y,'y:+')

строит желтый пунктирный график и помещает маркеры '+' в каждую точку данных. Если вы определяете только тип маркера, но не определяете тип стиля линий, то Matlab выведет только маркеры.

Графики дискретных отсчетов функции

В сопротивлении материалов и строительной механике принято графики параметров напряженно-деформированного состояния закрашивать (штриховать). Для построения подобных графиков используется команда *stem*(...).

stem(x, y) – строит график с закрашенными частями между нулевой линией и кривой $y = f(x)$.

stem(... , 'LINESPEC') – аналогична команде *stem*(x, y), но имеет спецификацию линий 'LINESPEC', подобную строковой константе s команды *plot*(x, y, s).

Stem(... , 'filled') – строит график функции $y = f(x)$ с закрашенными маркерами.

Мнимые и комплексные данные. Если аргумент функции **plot** комплексное число, то мнимая часть игнорируется, за исключением случая, когда комплексный аргумент один. Для этого специального случая происходит построение графика зависимости реальной части аргумента от мнимой. Поэтому **plot(Z)**, где Z — комплексный вектор или матрица, эквивалентно:

plot(real(Z),imag(Z))

Например:

t = 0:pi/10:2*pi;

plot(exp(i*t),'-o')

отобразит двадцатисторонний многоугольник с кружками на вершинах (рис. 5).

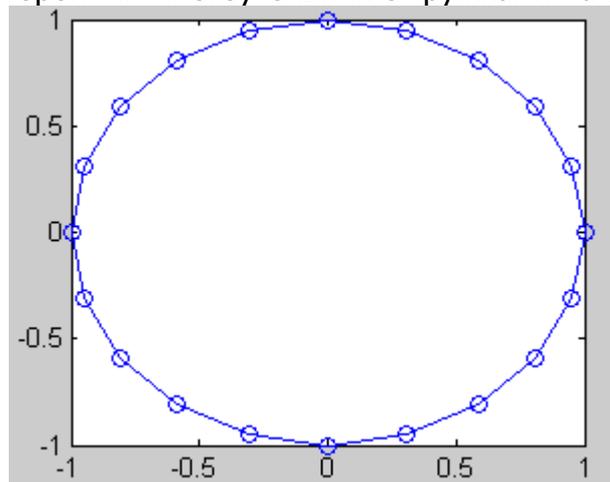


Рис. 5

Окна изображений. Добавление кривых на существующий график. Подграфики

Окна изображений. Функция **plot** автоматически открывает новое окно изображения (далее окно), если до этого его не было на экране. Если же оно существует, то **plot** использует его по умолчанию. Для открытия нового окна и выбора его по умолчанию, наберите:

figure

Для того чтобы сделать существующее окно текущим:

figure(n)

где n — это номер в заголовке окна. В этом случае результаты всех последующих команд будут выводиться в это окно.

Добавление кривых на существующий график. Команда **hold** позволяет добавлять кривые на существующий график. Когда вы набираете:

hold on

Matlab не стирает существующий график, а добавляет в него новые данные, изменяя оси, если это необходимо. Например, следующий элемент кода вначале создает график функции **sin(x)**, а затем накладывает на него график функции **sin(2*x)** (рис. 3):

x=0:pi/100:4*pi;

plot(x,sin(x),'r')

hold on

plot(x,sin(2*x),'b')

grid on

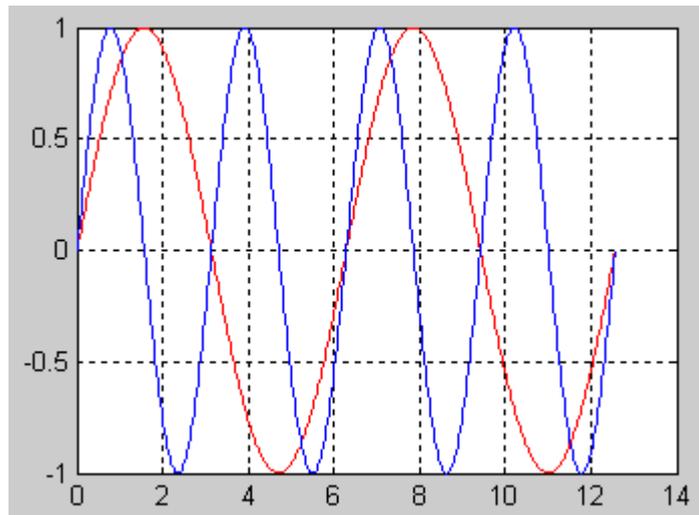


Рис. 3.

Подграфики. Функция **subplot** позволяет выводить множество графиков в одном окне:

subplot(m,n,p)

разбивает окно изображений на матрицу m на n подграфиков и выбирает p -ый подграфик текущим. Графики нумеруются вдоль первого в верхней строке, потом во второй и т. д. Например, для того, чтобы представить графические данные в четырех разных подобластях окна необходимо выполнить следующее (рис. 4):

```
x=0:pi/100:4*pi;
```

```
subplot(2,2,1)
```

```
plot(x,sin(x),'r')
```

```
grid on
```

```
subplot(2,2,2)
```

```
plot(x,sin(2*x),'b')
```

```
grid on
```

```
subplot(2,2,3)
```

```
plot(x,sin(4*x),'m')
```

```
grid on
```

```
subplot(2,2,4)
```

```
plot(x,sin(6*x),'g')
```

```
grid on
```

Информационные технологии в менеджменте

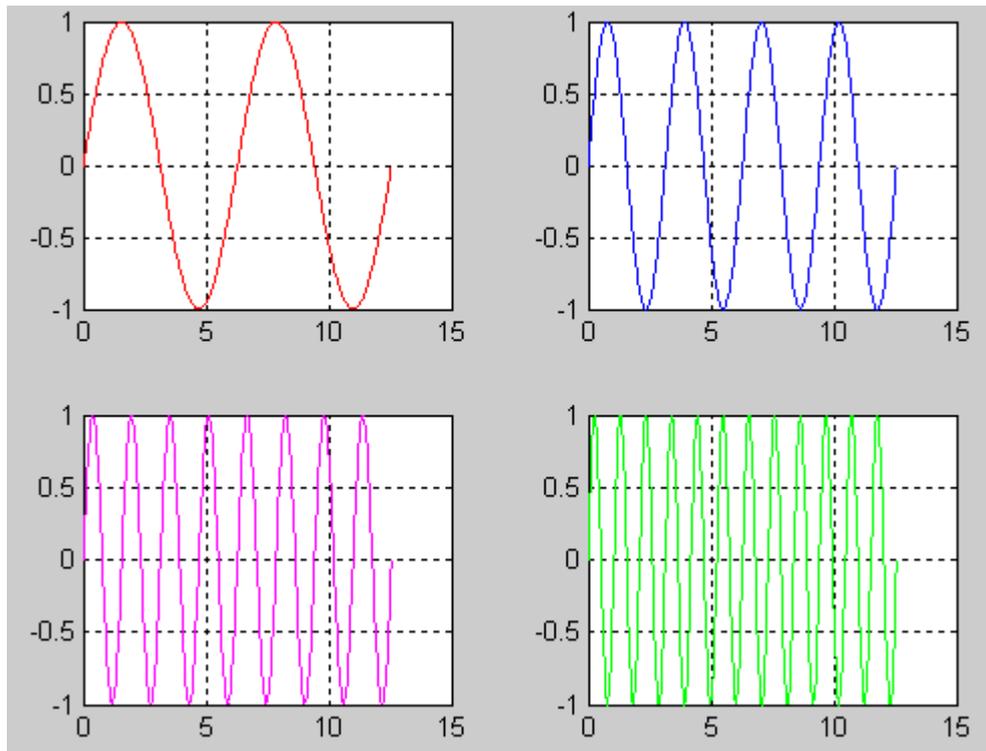


Рис. 4.

Управление осями. Подписи к осям и заголовки

Функция **axis** имеет несколько возможностей для настройки масштаба, ориентации и коэффициента сжатия.

Обычно Matlab находит максимальное и минимальное значение и выбирает соответствующий масштаб и маркирование осей. Функция **axis** заменяет значения по умолчанию предельными значениями, вводимыми пользователем:

axis([xmin xmax ymin ymax])

В функции **axis** можно также использовать ключевые слова для управления внешним видом осей. Например:

axis square

создает *x* и *y* оси равной длины, а:

axis equal

создает отдельные отметки приращений для *x* и *y* осей одинаковой длины. Так функция:

plot(exp(i*t))

следующая либо за **axis square**, либо за **axis equal** превращает овал в правильный круг.

axis auto

возвращает значения по умолчанию и переходит в автоматический режим.

axis on

включает обозначения осей и метки промежуточных делений.

axis off выключает обозначения осей и метки промежуточных делений.

grid off выключает сетку координат, а

grid on включает ее заново.

Подписи к осям и заголовки. Функции **xlabel('имя оси')**, **ylabel('имя оси')**, **zlabel('имя оси')** добавляют подписи к соответствующим осям, функция **title('название графика')** добавляет заголовок в верхнюю часть окна, а функция **text** вставляет текст в любое место графика.

Трехмерные графики

Трехмерные поверхности описываются функцией двух переменных $Z = f(X, Y)$. Функции Matlab, предназначенные для построения трехмерных графиков, требуют определения X и Y как матриц, состоящие из повторяющихся строк и столбцов.

Для создания таких матриц служит функция **meshgrid**, которая преобразует область определения, заданную через один вектор или два вектора x и y , в матрицы X и Y . Строки матрицы X дублируют вектор x , а столбцы Y — вектор y .

Функция записывается следующим образом:

[X,Y]=meshgrid(x,y)

Пример:

```
>> [X Y] = meshgrid(4 : 7, 9 : 13)
      X      =
          4   5   6   7
          4   5   6   7
          4   5   6   7
          4   5   6   7
          4   5   6   7
      Y      =
          9   9   9   9
         10  10  10  10
         11  11  11  11
         12  12  12  12
         13  13  13  13
```

Приведем еще пример применения функции *meshgrid*:

```
>> [X Y] = meshgrid(-1 : 0.1 : 1, -1 : 0.1 : 1);
```

Такой вызов функции создает опорную плоскость для построения трехмерной поверхности при изменении x и y от -1 до 1 с шагом 0.1 .

Функции **mesh** и **surface** отображают поверхность в трех измерениях. При этом **mesh** создает каркасную поверхность, где цветные линии соединяют только заданные точки, а функция **surface** вместе с линиями отображает в цвете и саму поверхность.

Для вычисления двумерной функции $\sin(R)/R$, в области x - y поступают следующим образом (рис. 6):

```
[X, Y] = meshgrid(-8:.5:8);
R = sqrt(X.^2+Y.^2)+eps;
Z = sin(R)./R;
mesh(X,Y,Z)
```

Здесь X , Y , Z — это матрицы значений координат поверхности.

В этом примере R — это расстояние от начала координат, которому соответствует центр матрицы. Добавление *eps* позволяет избежать неопределенности $0/0$ в начале координат.

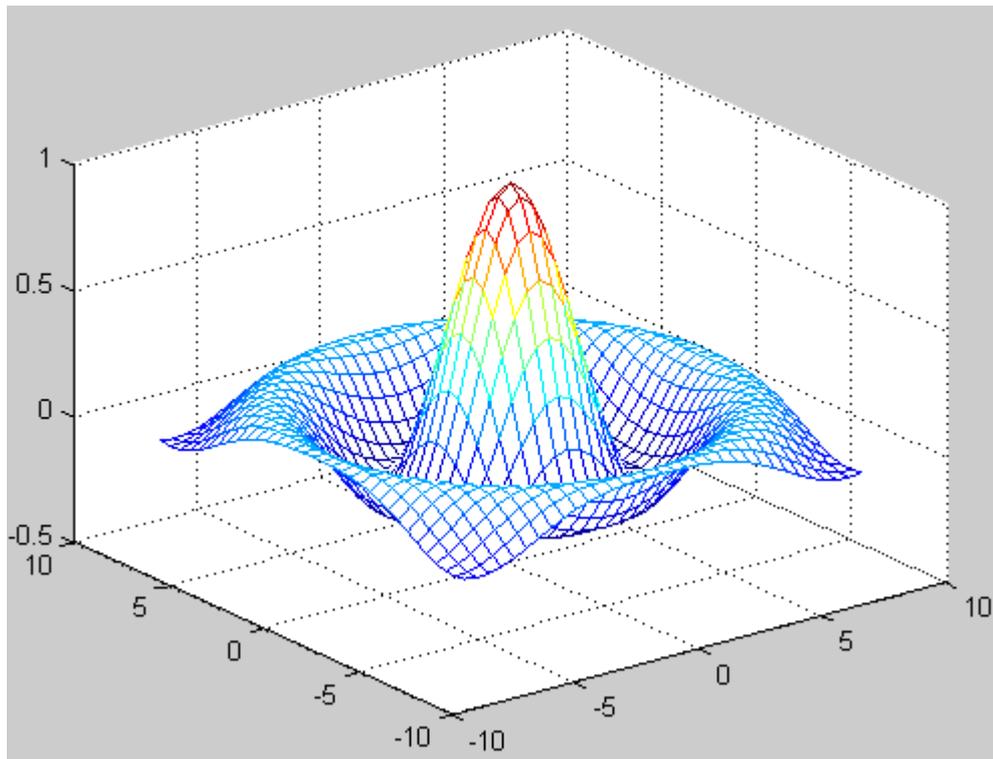


Рис. 6.

При использовании функции (рис. 7):
surface(X,Y,Z)

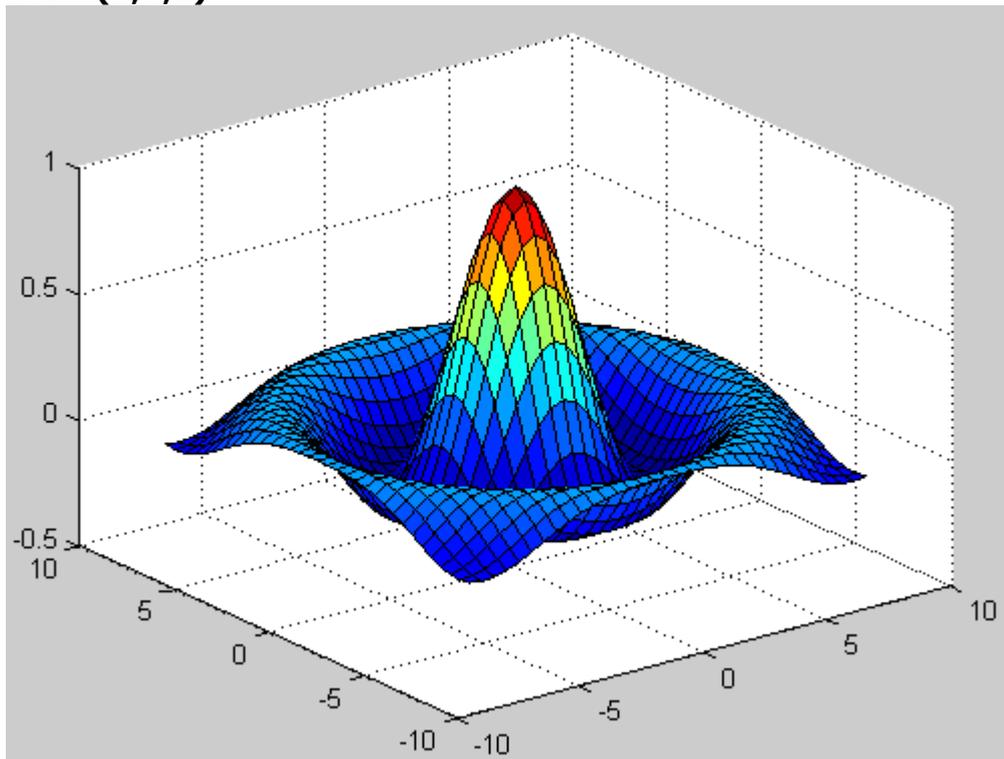


Рис. 7.

Для построения графика в трехмерном пространстве можно использовать функцию **plot3(x,y,z)** (рис. 8):

```
t=0:0.1:30;  
x=0.1*t.*cos(t);  
y=0.1*t.*sin(t);  
z=0.05*t;
```

`plot3(x,y,z);`
`grid on`

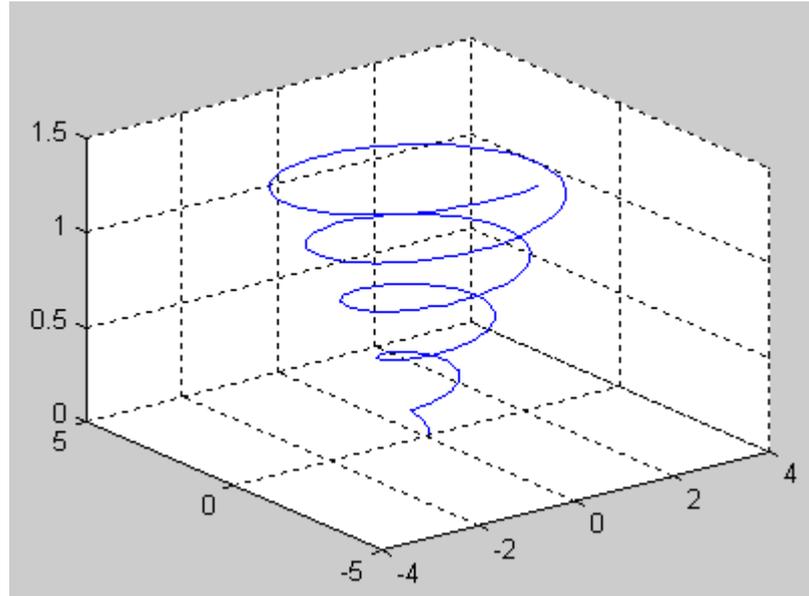


Рис. 8.

В этом случае x , y , z — это вектора значений координат функции в пространстве.

Задания

Все задания рекомендуется выполнять в m-файлах.

Задание 1. Определите дискретный аргумент t в произвольно выбранном диапазоне. Определите функцию $f(t, a)$ вида:

№	Выражение	№	Выражение
1.	$f(t, a) = ax / \ln(\omega t)$	16.	$f(t, a) = \sqrt{ xt^2 - 3 } / xa$
2.	$f(t, a) = xax^3 - 2\sin(\omega t)$	17.	$f(t, a) = xt^2 \sqrt{ a^2 - 1 }$
3.	$f(t, a) = (ax^2 - 1) \cdot t / (x\omega^2 + 1)$	18.	$f(t, a) = a / (\sin(\omega t) + \cos(\omega t))$
4.	$f(t, a) = 3at^4 - 4at^3 + 12at^2 + 1$	19.	$f(t, a) = a\omega / 2 + \arctg(\omega t)$
5.	$f(t, a) = xa - 2\ln(\omega t)$	20.	$f(t, a) = ae^{-t^2/2}$
6.	$f(t, a) = e^{-at} \cos(\omega t)$	21.	$f(t, a) = e^{-t^2} / a\omega$
7.	$f(t, a) = (1 - xt + xt^2) / (a\omega - x^2)$	22.	$f(t, a) = - x ^3 e^{-at^2/2}$
8.	$f(t, a) = -\sqrt{2a\omega t - xt^2}$	23.	$f(t, a) = xa^2 \ln(\omega t)$
9.	$f(t, a) = (x - 2) \cdot t^5 (2a\omega + 1) \cdot t^4$	24.	$f(t, a) = x\omega \ln^2(at)$
10.	$f(t, a) = x^{-\omega t} + \sin(ax)$	25.	$f(t, a) = x \arctg(at)$
11.	$f(t, a) = e^{-1/\omega t} + \cos(at)$	26.	$f(t, a) = \sin(x\omega t) + \cos(a\omega t)$

Информационные технологии в менеджменте

12.	$f(t, a) = xa\sqrt{1 - xt^2}$	27.	$f(t, a) = -a \ln(t) / x^2$
13.	$f(t, a) = ((e^x + e^{-x}) / 2) \cdot t^3 + a\omega$	28.	$f(t, a) = x^2 \ln^2(at)$
14.	$f(t, a) = -at / (x\omega^3 + 2)$	29.	$f(t, a) = a^2 \sin(t) / x$
15.	$f(t, a) = x\omega^2 / \sqrt[3]{at^3 - 4}$	30.	$f(t, a) = -ax^{1/\omega t}$

предварительно задав переменные ω, x, a . Постройте графики функции $f(t, a)$ для двух разных значений аргумента a .

Наиболее просто такая задача решается следующим образом, например для функции $f(t, a) = a \sin(t)$:

t=0:0.01:10; %вектор-строка дискретных значений времени

a=[1 2]; %вектор-строка значений параметра a

%задание функции f как матрицы размерности $2 \times n$, в которой строки соответствуют значения функции f в точках дискретизации при разных параметрах a

f(1,:)=a(1)*sin(t);

f(2,:)=a(2)*sin(t);

plot(t, f(1,:), t, f(2,:))

Как изменить цвет и стиль линий на графике? Как вывести графики в разных окнах? Что произойдет, если использовать функцию **subplot**? Как озаглавить график и оси?

Задание 2. С использованием функций **mesh** и **surface** постройте трехмерный график пересечения поверхностей:

$$f(x, y) = \frac{(x + y)^2}{10},$$

$$f(x, y) = 5 \cdot \cos\left(\frac{x - y}{3}\right).$$

Задание 3. Постройте график поверхности $z(x, y)$ в определенной области изменения независимых переменных x, y :

1. $z = \sqrt{\ln(x+y)}$.
2. $z = \ln \frac{x^2}{x+y}$.
3. $z = \ln \frac{\cos x}{y}$.
4. $z = \ln \frac{x-3}{y-5}$.
5. $z = \ln(y - \sin x)$.
6. $z = \log_y(x^2 + y^2 - 9)$.
7. $z = \frac{1}{x} \arcsin \frac{x+y}{y}$.
8. $z = \sqrt{\ln xy}$.
9. $z = \frac{\ln(x^2 y)}{\sqrt{y-x}}$.
10. $z = \ln(-x-y)$.
11. $z = \ln x + \ln \sin y$.
12. $z = \frac{\ln x}{\sqrt{4-x^2-y^2}}$.
13. $z = \frac{1}{x} + \sqrt{(x^2 + y^2 - 1)(4 - x^2 - y^2)}$.
14. $z = \sqrt{1-x^2} + \sqrt{y^2-1}$.
15. $z = \frac{1}{x-y} - \frac{1}{\sqrt{x^2 + y^2 - 9}}$.
16. $z = \frac{1}{|x|} \sqrt{y^2 - x}$.
17. $z = (x + \sqrt{y}) \cdot \ln(y^2 - x^2)$.
18. $z = \sqrt{1 - (x^2 + y^2)^2} \cdot \ln y$.
19. $z = \sqrt{1-x^3} + \ln(y^2 - 1)$.
20. $z = \frac{\sqrt{\cos x - y}}{\sqrt{y}}$.
21. $z = \arcsin(x+y) + \sqrt{9-x^2-y^2}$.
22. $z = \arccos \frac{x}{y^2} + \arcsin(1-y^2)$.
23. $z = \sqrt{x^2 + y^2 - 2x} - \sqrt{4-x^2-y^2}$.
24. $z = \sqrt{\sin(x^2 + y^2)} + \arcsin(x-y)$.
25. $z = \arcsin \frac{y}{x^2} + \arccos(1-x)$.
26. $z = \sqrt{\sin x \cdot \cos y}$.
27. $z = \frac{\sqrt{x}}{\sin y}$.
28. $z = \frac{\sqrt{x-\sqrt{y}}}{\sqrt{3-x^2-y^2}}$.
29. $z = \frac{1}{\sqrt{(y^2-1)(x+1)}}$.
30. $z = \sqrt{\ln x \cdot \ln y}$.

Задание 4. С использованием функции **plot3** постройте трехмерный график векторов X, Y, Z , значения элементов которых задайте самостоятельно.

Лабораторная работа

Matlab: Символьные вычисления

Цель работы: изучение возможностей пакета Matlab в области символьных вычислений.

Теоретические сведения

Система Matlab является самой крупной системой компьютерной математики, ориентированной на матричные и численные вычисления. Однако Matlab имеет также и средства аналитических вычислений. Пакет **Symbolic Math Toolbox** добавил системе Matlab качественной новые возможности, связанные с выполнением символьных вычислений и преобразований.

С помощью команды:

```
>> help symbolic
```

Можно получить перечень входящих в пакет команд и функций. Для получения справки по любой команде или функции можно использовать команду:

```
>> help sym / name.m,
```

где **name** — имя соответствующей команды или функции, а **name.m** — имя m-файла, задающего данную команду или функцию.

Создание символьных переменных и выражений

Команда **sym** создает символьную переменную и/или выражение. Например, выполнение команды:

```
x=sym('x')
```

приведет к созданию символьной переменной x .

Если необходимо использовать символьное обозначение для выражения:

$$h = \frac{1 + \sqrt{5}}{2},$$

тогда команда будет выглядеть:

```
h=sym('(1+sqrt(5))/2');
```

Теперь можно использовать h в различных математических выражениях.

Например:

```
f=h^2-h-1
```

вернет:

```
f = (1/2+1/2*5^(1/2))^2-3/2-1/2*5^(1/2)
```

Выражение:

```
f=sym('a*x^2+b*x+c')
```

придаст символьное выражение $ax^2 + bx + c$ переменной f . Но и только,

Symbolic Math Toolbox не создаст переменных, используемых в данном выражении, а именно a, b, c, x . Для дальнейшего осуществления математических операций (таких как интегрирование, дифференцирование, упрощение и т. д.) над функцией f , необходимо вручную создать эти переменные:

```
a=sym('a');
```

```
b=sym('b');
```

```
c=sym('c');
```

```
x=sym('x');
```

или

```
syms a b c x;
```

Информационные технологии в менеджменте

Рекомендуется использовать **syms**, что позволит избежать создания избыточных кодов.

Функция создания списка символьных переменных *findsym*

В математических выражениях могут использоваться как обычные, так и символьные переменные. Функция *findsym* позволяет выделить символьные переменные в составе выражения S:

- **findsym(S)** — возвращает в алфавитном порядке список всех символьных переменных выражения S. При отсутствии таковых возвращается пустая строка.

Примеры:

```
» a =2; b = 4 ;
» findsym (a*x^2 + b*y + z)
ans =
x, y, z
» findsym(a + b + x + y + z, 2)
ans =
x, y, z
```

Символьные операции с выражениями

В Matlab имеется несколько функций для преобразования одной формы представления функции к другой, возможно более простой (табл. 1).

Таблица 1

Символьные операции

<i>Вид преобразования</i>	<i>Ключевое слово</i>	<i>Описание</i>
Приведение подобных слагаемых	collect	Приведение подобных слагаемых полинома
Расширение	expand	Раскрываются суммы и произведения, тригонометрические функции разлагаются по тригонометрическим тождествам
Разложение на множители	factor	Производится разложение выражений или чисел на простые множители.
Упрощение	simplify	Приводит выражение к более простому виду
Обеспечение подстановок	subs	

Эти функции в качестве входного параметра могут принимать символьную матрицу. В таком случае выполнение операции происходит поэлементно.

Функция **collect(f)** представляет *f* как полином от символьной переменной, например *x*, и собирает коэффициенты при степенях *x*. Второй аргумент определяет коэффициенты какой переменной необходимо собрать, если в выражении более одной переменной. Примеры:

f	collect(f)
$(x-1)*(x-2)*(x-3)$	$x^3-6*x^2+11*x-6$
$x*(x*(x-6)+11)-6$	$x^3-6*x^2+11*x-6$
$(1+x)*t + x*t$	$2*x*t+t$

Функция **expand(f)** расширяет выражение **f**. Рациональные выражения она раскладывает на простые дроби, полиномы — на полиномиальные разложения и т.д. Примеры:

f	expand(f)
$a*(x + y)$	$a*x+a*y$
$(x-1)*(x-2)*(x-3)$	$x^3-6*x^2+11*x-6$
$x*(x*(x-6)+11)-6$	$x^3-6*x^2+11*x-6$
$\exp(a+b)$	$\exp(a)*\exp(b)$
$\cos(x+y)$	$\cos(x)*\cos(y)-\sin(x)*\sin(y)$
$\cos(3*\arccos(x))$	$4*x^3-3*x$

Функция **factor(f)** разлагает выражение **f** на простые множители, а целые числа — на произведение простых чисел. Если **f** не может быть разложен, то результатом выполнения будет само выражение **f**. Примеры:

f	factor(f)
$x^3-6*x^2+11*x-6$	$(x-1)*(x-2)*(x-3)$
$x^3-6*x^2+11*x-5$	$x^3-6*x^2+11*x-5$
x^6+1	$(x^2+1)*(x^4-x^2+1)$

Функция **simplify(f)** наиболее мощное средство, она предназначена для алгебраических преобразований, сверток степеней, преобразования тригонометрических, экспоненциальных и логарифмических функций.

Примеры:

```
» syms a b x;
» V = [sin(x)^2 + cos(x)^2 log(a*b)]
» simplify(V)
ans =
 [ 1, log(a*b)]
» simplify((a^2 - 2*a*b + b^2) / (a - b))
ans =
 a - b
```

Функция **simple(f)** использует все правомерные математические способы упрощения для нахождения варианта с наименьшей длиной символьной строки результата. Функция добивается такого результата путем последовательного применения **simplify**, **collect**, **factor**, и других функций упрощения к выражению и сохранением информации о длине строки результата. Возвращает функция наименьший результат (тот ответ, строка которого имеет наименьшую длину).

Функция **simple** имеет несколько форм записи:

simple(f)

возвращает наименьший результат. Например, выражение:

f = simple(cos(x)^2+sin(x)^2)

вернет **f = 1**.

Если вы хотите знать каким из методов был получен этот результат, то необходимо использовать вторую форму записи:

[F, how] = simple(f)

В данном виде функция вернет в качестве F наикратчайший результат, а в качестве how — название метода. Например:

[f, how] = simple(cos(x)^2+sin(x)^2)

вернет **f = 1**

how = combine

Одной из самых эффективных и часто используемых операций символьной математики является операция подстановки. Она реализуется функцией **subs**, имеющей ряд форм записи:

subs(S) — заменяет в символьном выражении **S** все переменные их символьными значениями, которые берутся из вычисляемой функции или рабочей области системы MATLAB.

subs(S, NEW) — заменяет все свободные символьные переменные в **S** из списка **NEW**.

subs(S, OLD, NEW) — заменяет **OLD** на **NEW** в символьном выражении **S**. При одинаковых размерах массивов **OLD** и **NEW** замена идет поэлементно. Если **S** и **OLD** — скаляры, а **NEW** — числовой массив или массив ячеек, то скаляры расширяются до массива результатов.

Примеры:

```
>>syms a b x y;
```

```
>>subs(x-y,y,1)
```

```
ans =
```

```
x-1
```

```
>>subs(sin(x)+cos(y),[x y],[a b])
```

```
ans =
```

```
sin(a)+cos(b)
```

В задачах аппроксимации и приближения функций $f(x)$ важное место занимает их разложение в ряд Тейлора в окрестности точки a :

$$f(x) = \sum_{n=0}^{\infty} (x - a)^n \frac{f^n(a)}{n!}.$$

Частным случаем этого ряда при $a = 0$ является ряд Маклорена:

$$f(x) = \sum_{n=0}^{\infty} x^n \frac{f^n(0)}{n!}.$$

Функция разложения в ряд Тейлора – *taylor*

- `taylor(f)` – возвращает шесть членов ряда Маклорена функции f ;
- `taylor(f, n, x, a)` – возвращает n членов ряда Тейлора в точке $x = a$;
- `taylor(f, n)` – возвращает $(n - 1)$ членов ряда Маклорена функции f ;
- `taylor(f, a)` – возвращает шесть членов ряда Тейлора функции f в точке a .

Примеры:

```

»x = sym('x')
»taylor(sin(x))
ans =
    x - 1/6*x^3 + 1/120*x^5
»taylor(int(sin(x)))
ans =
    - 1 + 1/2*x^2 - 1/24*x^4

```

Функция вычисления пределов – *limit*

Для вычисления пределов аналитически заданной функции $F(x)$ служит функция *limit*

- `limit(F, x, a)` – возвращает предел символьного выражения F в точке $x = a$.

Пример:

```

»limit(sin(x)/x, x, 0)
ans =
    1

```

Функция вычисления суммы рядов – *symsum*

Для аналитического вычисления суммы ряда служит команда `symsum`:

- `symsum(S)` — возвращает символьное значение суммы бесконечного ряда по переменной, найденной автоматически с помощью функции `findsym`;
- `symsum(S, v)` — возвращает сумму бесконечного ряда по переменной v ;
- `symsum(S, a, b)` и `symsum(S, v, a, b)` — возвращают конечную сумму ряда в пределах номеров слагаемых от a до b .

Примеры:

```

» x = sym('x');
» symsum(x^2)
ans =
    1/3*x^3-1/2*x^2+1/6*x
» symsum([x, x^2, x^3], 1, 5)
ans =
    [ 15, 55, 225]

```

Задания

Все задания рекомендуется выполнять в m-файлах.

Задание 1. Упростите алгебраическое выражение с помощью функций **simple** и **simplify**:

$$1., 20. \left(\frac{a\sqrt{a} + b\sqrt{b}}{\sqrt{a} + \sqrt{b}} - \sqrt{ab} \right) : (a - b)$$

$$2., 21. \left(\frac{\sqrt{a^3} - \sqrt{b^3}}{\sqrt{a} - \sqrt{b}} + \sqrt{ab} \right) \left(\frac{\sqrt{a} - \sqrt{b}}{a - b} \right)^2$$

$$3., 22. \left(\left(\frac{\sqrt{a} + \sqrt{b}}{\sqrt[4]{a} - \sqrt[4]{b}} \right)^{-1} - \frac{2^4 \sqrt{ab}}{b^{\frac{3}{4}} - a^{\frac{1}{4}} b^{\frac{1}{4}} + a^{\frac{1}{2}} b^{\frac{1}{4}} - a^{\frac{3}{4}}} \right)^{-1}$$

$$4., 23. \left(\frac{\sqrt[4]{ab^3} - \sqrt[4]{a^3b}}{\sqrt{a} - \sqrt{b}} + \frac{\sqrt{ab} + 1}{\sqrt[4]{ab}} \right)^{-2} \cdot \sqrt{1 + \frac{a}{b}} + 2\sqrt{\frac{a}{b}}$$

$$5., 24. \frac{2b\sqrt{x^2 - 1}}{x - \sqrt{x^2 - 1}} = A, \text{ где } x = \frac{1}{2} \left(\sqrt{\frac{a}{b}} + \sqrt{\frac{b}{a}} \right)$$

$$6., 25. \left(\frac{a}{a-1} + \frac{1}{ab-b} \right) \left(\frac{ab+1}{b} \right)^{-1} - \frac{a}{a^2-1}$$

$$7., 26. \left(\frac{\sqrt{a}+2}{(\sqrt{a}+1)^2} - \frac{\sqrt{a}-2}{a-1} \right) \frac{\sqrt{a}+1}{\sqrt{a}}$$

$$8., 27. \left(\frac{a}{a-b} - \frac{b}{a+b} \right) \left(\frac{a+b}{b} - \frac{a-b}{a} \right)$$

$$9., 28. \left(\frac{a\sqrt{a} + b\sqrt{b}}{\sqrt{a} + \sqrt{b}} - \sqrt{ab} \right) \left(\frac{\sqrt{a} + \sqrt{b}}{a - b} \right)^2$$

$$10., 29. \left(2x + 1 - \frac{1}{1-2x} \right) \left(2x - \frac{4x^2}{2x-1} \right)$$

$$11., 30. (\sqrt{1-x^2} + 1) \left(\frac{1}{\sqrt{1+x}} + \sqrt{1-x} \right)$$

$$12. \frac{0,25(a^{-3/2} - a^{3/2})}{a^{-1} + a + 1} \left(\frac{\sqrt{a}-1}{\sqrt{a}+1} - \frac{\sqrt{a}+1}{\sqrt{a}-1} \right)$$

$$13. \left(p - q + \frac{4q^2 - p^2}{p + q} \right) \left(\frac{p}{p^2 - q^2} + \frac{2}{q - p} + \frac{1}{p + q} \right)$$

$$14. \frac{(a-b)^3(\sqrt{a}+\sqrt{b})^{-3}+2a\sqrt{a}+b\sqrt{b}}{a\sqrt{a}+b\sqrt{b}} + \frac{3(\sqrt{ab}-b)}{a-b}$$

$$15. \left(\frac{a}{a+\sqrt{ab}} - \frac{b}{\sqrt{ab}+b}\right) \left(\frac{\sqrt{a}}{\sqrt{b}} - \frac{\sqrt{b}}{\sqrt{a}}\right) \sqrt{ab}$$

$$16. \left(\frac{2}{2x+y} - \frac{1}{2x-y} - \frac{3y}{y^2-4x^2}\right) \cdot \left(\frac{y^2}{8x^2} - 0,5\right)$$

$$17. \frac{4x(x+\sqrt{x^2-1})^2}{(x+\sqrt{x^2-1})^4-1}$$

$$18. \left(\frac{x^3-8}{x^2+4+2x} + 4\right) \cdot \frac{1}{x+2}$$

$$19. \left(\frac{x^2-y^2}{x-y} - \frac{x^2+y^2}{x+y}\right) \cdot \frac{x^2-y^2}{2xy}$$

Сравните результаты.

Задание 2. Разложите выражение по степеням (раскройте скобки) (функция **expand**):

- 1., 7., 13., 19., 25. $\sin(x) - 2\sin(2x) + 4\sin(3x)$
- 2., 8., 14., 20., 26. $\cos(2x) - \sin(2x) + \sin(3x)$
- 3., 9., 15., 21., 27. $\operatorname{tg}(2x) + \operatorname{ctg}(2x) + \cos(2x)$
- 4., 10., 16., 22., 28. $\operatorname{ctg}(2x) - \sin(x) \cdot \cos(3x)$
- 5., 11., 17., 23., 29. $\cos(2x) \cdot \sin(x) \cdot \sin(3x)$
- 6., 12., 18., 24., 30. $\operatorname{tg}(2x) \cdot \sin(2x) - \sin(3x)$

Задание 3. Разложите на простые множители выражения (функция **factor**):

- 1., 9., 17., 25. $a^5 - 32$
- 2., 10., 18., 26. $a^4 - 16$
- 3., 11., 19., 27. $3x^2 + 10xy - 8y^2 - 8x + 10y - 3$
- 4., 12., 20., 28. $\frac{x^2 + 5x + 4}{x^4 + 5x^2 + 4}$
- 5., 13., 21., 29. $18ab^2 + 27a^2b$
- 6., 17., 22., 30. $1 - 64b^2$
- 7., 15., 23. $2x^2 - 20xy + 50y^2 - 2$
- 8., 16., 24. $x^3y - xy - x^3 + x$

Задание 4. Сделайте замену $x = y + z$ в функции $g(x)$, используя функцию замены **subs**.

Информационные технологии в менеджменте

Затем приведите подобные слагаемые в полученном выражении относительно каждой из переменных y, z (функция **collect**).

№ варианта	$g(x)$	№ варианта	$g(x)$
1., 16.	$x^4 - 2x^3 + x^2 - 12x + 20$	9., 24.	$x^4 + x^3 - 17x^2 - 45x - 100$
2., 17.	$x^4 + 6x^3 + x^2 - 4x - 60$	10., 25.	$x^4 - 5x^3 + x^2 - 15x + 50$
3., 18.	$x^4 - 14x^2 - 40x - 75$	11., 26.	$x^4 - 4x^3 - 2x^2 - 20x + 25$
4., 19.	$x^4 - x^3 + x^2 - 11x + 10$	12., 27.	$x^4 + 5x^3 + 7x^2 + 7x - 20$
5., 20.	$x^4 - x^3 - 29x^2 - 71x - 140$	13., 28.	$x^4 - 7x^3 + 7x^2 - 5x + 100$
6., 21.	$x^4 + 7x^3 + 9x^2 + 13x - 30$	14., 29.	$x^4 + 10x^3 + 36x^2 + 70x + 75$
7., 22.	$x^4 + 3x^3 - 23x^2 - 55x - 150$	15., 30.	$x^4 + 9x^3 + 31x^2 + 59x + 60$
8., 23.	$x^4 - 6x^3 + 4x^2 + 10x + 75$		

Задание 5. Разложите в ряд выражение:

№ варианта	$f(x)$	№ варианта	$f(x)$	№ варианта	$f(x)$
1, 16	$1/(\operatorname{tg}2x + 1)$	6, 21	$x^2 \operatorname{arctg}(x/3)$	11, 26	$(2x + 3) \sin x$
2, 17	$\cos x / (2x + 5)$	7, 22	$e^{2x} \sin 3x$	12, 27	$\cos 3x / (1 - \cos 3x)^2$
3, 18	$1/(x\sqrt{x^3 + 4})$	8, 23	$\operatorname{ctg}2x / (\sin 2x)^2$	13, 28	$1/(1 + x + x^2)$
4, 19	$\sin x / (1 + \sin x)$	9, 24	$(x + 1) \sin x$	14, 29	$(1 + x)/(2 + x)$
5, 20	$x^2 \lg(x + 2)$	10, 25	$5x + x \lg x$	15, 30	$\sqrt{1 + e^{-x}}$

Порядок разложения 6.

Задание 6. Найдите предел выражения:

$$1) \lim_{x \rightarrow 0} \frac{(1+x)^5 - (1+5x)}{x^2 + x^5}; \quad 2) \lim_{x \rightarrow 1} \frac{x^4 - 3x + 2}{x^5 - 4x + 3}; \quad 3) \lim_{x \rightarrow 3} \frac{\sqrt{x+13} - 2\sqrt{x+1}}{x^2 - 9};$$

$$4) \lim_{x \rightarrow 0} \frac{\sqrt{1+x} - \sqrt{1-x}}{\sqrt[3]{1+x} - \sqrt[3]{1-x}}; \quad 5) \lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k(k+1)}; \quad 6) \lim_{n \rightarrow \infty} \frac{\sum_{k=1}^n (2k-1)^2}{\sum_{k=1}^n (2k)^2};$$

$$7) \lim_{x \rightarrow a} \frac{\cos(x) - \cos(a)}{x - a}; \quad 8) \lim_{x \rightarrow \infty} \frac{\ln(1 + \sqrt{x} + \sqrt[3]{x})}{\ln(1 + \sqrt[3]{x} + \sqrt[4]{x})}; \quad 9) \lim_{x \rightarrow 0} \frac{\int_0^x \cos(t^2) dt}{x}.$$

$$10) \lim_{x \rightarrow 1} \frac{x^2 + 2 \cdot x + 5}{x^2 + 1} \quad 11) \lim_{x \rightarrow \frac{\pi}{2}} (2 \cdot \sin(x) - \cos(x) + \operatorname{ctg}(x))$$

$$12) \lim_{h \rightarrow 0} \frac{(x+h)^3 - x^3}{h} \quad 13) \lim_{x \rightarrow \infty} \frac{\sqrt{x^2 - 3}}{\sqrt[3]{x^3 + 1}}$$

$$14) \lim_{x \rightarrow +\infty} x \cdot (\sqrt{x^2 + 1} - x) \quad 15) \lim_{x \rightarrow -\infty} x \cdot (\sqrt{x^2 + 1} - x)$$

Задание 7. Вычислите сумму ряда аналитически и численно при $x=M_V$, где M_V — номер варианта:

$$1., 26. \sum_{n=0}^{\infty} \frac{x}{n!}, \quad 2., 27. \sum_{n=1}^{\infty} \left(3 + \frac{1}{n}\right)^{-n} 4^{-\frac{n^2}{x}}, \quad 3., 28. \sum_{n=1}^{\infty} 8^{-n} n^2 \sin^{3n} x,$$

$$4., 29. \sum_{n=1}^{\infty} \frac{(x+1)^3}{(4n)!}, \quad 5., 30. \sum_{n=1}^{\infty} (-1)^{n+1} e^{-\frac{n}{x}}, \quad 6. \sum_{n=1}^{\infty} \frac{5^{-n} x^n}{n^2},$$

$$7. \sum_{n=1}^{\infty} \frac{x^{2n-1}}{(2n-1)(2n-1)!}, \quad 8. \sum_{n=1}^{\infty} \frac{5^{-n} x^{n-1}}{(2+n)!}, \quad 9. \sum_{n=1}^{\infty} \frac{0,5^n x^{3-n}}{(1+2n)!},$$

$$10. \sum_{n=1}^{\infty} \frac{(x+1)^3}{(4+n)!}, \quad 11. \sum_{n=1}^{\infty} \frac{(n+x)^{-n}}{(2n+1)!}, \quad 12. \sum_{n=1}^{\infty} \frac{(nx)^n}{(2n+1)!},$$

$$13. \sum_{n=1}^{\infty} \frac{x}{(2n-1)!}, \quad 14. \sum_{n=1}^{\infty} \frac{x}{n^{2n}}, \quad 15. \sum_{n=1}^{\infty} (-1)^{n-1} \frac{x}{n^4},$$

$$16. \sum_{n=1}^{\infty} \frac{x^{-n}}{(n+3)!}, \quad 17. \sum_{n=1}^{\infty} (-1)^{n-1} \frac{x^{-n}}{(2n-1)^3 \cdot (n-1)!}, \quad 18. \sum_{n=1}^{\infty} \frac{x^2}{(2n-1)!^2},$$

$$19. \sum_{n=1}^{\infty} \frac{x^{-2n}}{n(n+1)(n+2)!}, \quad 20. \sum_{n=1}^{\infty} \frac{x}{(4n-1)(4n+1)!}, \quad 21. \sum_{n=0}^{\infty} (-1)^n \frac{e^{-n}}{2^n},$$

$$22. \sum_{n=1}^{\infty} \frac{x^n}{(2n-1)(2n+1)!}, \quad 23. \sum_{n=1}^{\infty} \frac{x^n}{n!(n+1)!}, \quad 24. \sum_{n=1}^{\infty} \frac{x^{3-n}}{(2n-1)(n+1)!},$$

$$25. \sum_{n=1}^{\infty} (-1)^n \frac{n^2}{x^3}.$$

Лабораторная работа

Matlab: Дифференцирование и интегрирование

Цель работы: получение навыков дифференцирования и интегрирования функций в пакете Matlab.

Теоретические сведения

Дифференцирование

Аналитическое дифференцирование. Для вычисления производных в символьном виде от выражения f служит функция **diff(f)**.

Создадим символьное выражение:

```
syms a x;  
f = sin(a*x);
```

тогда **diff(f)** приведет к дифференцированию f по ее символьной переменной (в данном случае это x):

```
ans = cos(a*x)*a
```

Для взятия производной по переменной a , явно укажем: **diff(f,a)**. Данное выражение вернет df/da :

```
ans = cos(a*x)*x
```

Чтобы найти производные более высоких порядков, необходимо указать порядок производной. Например, производная второго порядка:

```
diff(f,2); или diff(f,x,2);  
ans = -sin(a*x)*a^2
```

и

```
diff(f,a,2);  
вернет ans = -sin(a*x)*x^2
```

Функция **diff** в качестве входного параметра может принимать символьную матрицу. В таком случае дифференцирование происходит поэлементно:

```
syms a x;  
A = [cos(a*x),sin(a*x);-sin(a*x),cos(a*x)];
```

Т. е. матрица A имеет вид:

```
A =  
[ cos(a*x), sin(a*x)]  
[ -sin(a*x), cos(a*x)]
```

Тогда результат команды **diff(A)** будет:

```
ans =  
[-sin(a*x)*a, cos(a*x)*a]  
[-cos(a*x)*a, -sin(a*x)*a]
```

Численное дифференцирование. Численное дифференцирование строится на использовании аппарата конечных разностей и соответствующего многообразия аппроксимаций. Используются функции:

diff(X) — для вектора X вычисление первых конечных разностей $[X(2)-X(1) X(3)-X(2) \dots X(n)-X(n-1)]$, а для матрицы X вычисление первых конечных разностей между столбцами $[X(2:n,:) - X(1:n-1,:)]$;

diff(X, n) — вычисление конечных разностей n -го порядка;

diff(X, n, dim) — вычисление конечных разностей n -го порядка по указанному измерению.

Интегрирование

Аналитическое интегрирование. Функция **int(f,x)** аналитически вычисляет интеграл от функции f по символьной переменной x :

```
syms x;
int(sin(x)./x.^2,x)
ans =
-sin(x)/x+cosint(x)
```

Функция **int(f,x,a,b)** численно вычисляет интеграл от функции f по символьной переменной x в границах a,b :

```
int(sin(x)./x.^2,x,0.1,pi)
ans =
cosint(pi)+10*sin(1/10)-cosint(1/10)
>> cosint(pi)+10*sin(1/10)-cosint(1/10)
ans =
2.7999
```

Численное интегрирование. Численное интегрирование в Matlab выполняется с помощью функций:

trapz — интегрирование методом трапеций;

quad, quad8 — вычисление интегралов методом квадратур.

Интегрирование методом трапеций.

Здесь используется линейная аппроксимация, т. е. график функции $y = f(x)$ представляется в виде ломаной, соединяющей точки y_i . Формула трапеций

при постоянном шаге $h = \frac{b-a}{n}$, где n – число участков, имеет вид

$$\int_a^b f(x)dx = h \left(\frac{y_0 + y_n}{2} + \sum_{i=1}^{n-1} y_i \right). \quad (2.8)$$

В Matlab данную формулу реализует функция **trapz(x, y)**, которая вычисляет интеграл от функции y по переменной x , используя метод трапеций. Аргументы x и y могут быть одномерными массивами одинакового размера, либо массив y может быть двумерным, но тогда должно выполняться условие $\text{size}(y, 1) = \text{length}(x)$. В последнем случае вычисляется интеграл для каждого столбца.

Функция **trapz(y)** вычисляет интеграл, предполагая, что шаг интегрирования постоянен и равен единице; в случае, когда шаг h отличен от единицы, но постоянен, достаточно вычисленный интеграл умножить на h .

Пример 1. Вычислим интеграл: $I = \int_0^{\pi} \sin(x)dx$. Его точное значение равно

двум.

Выберем равномерную сетку:

```
x = 0:pi/100:pi;
```

```
y = sin(x);
```

тогда оба интеграла

```
I = trapz(x, y)
```

и

```
I = pi/100*trapz(y)
```

дают одинаковый результат:

```
I = 1.9998
```

Образует неравномерную сетку, используя генератор случайных чисел:
x = sort(rand(1,101)*pi);
y = sin(x);
I = trapz(x, y)
I = 1.9987

Результат еще менее точен, поскольку максимальный из шагов **max(diff(x))** равен 0,1810.

Вычисление интегралов методом квадратур. Квадратура — это численный метод вычисления площади под графиком функции, то есть вычисление определенного интеграла вида:

$$I = \int_a^b f(x)dx.$$

Функции **quad** и **quad8** используют разные квадратурные формулы. Функция **quad** использует квадратурные формулы Ньютона-Котеса 2-го порядка (правило Симпсона), а функция **quad8** — формулы 8-го порядка. При наличии в подынтегральной функции особенностей типа:

$$I = \int_0^1 \sqrt{x}dx,$$

предпочтительнее использовать процедуру **quad8**.

Функции **[I, cnt] = quad('имя функции', a, b)** и **[I, cnt] = quad8('имя функции', a, b)** вычисляют интеграл от заданной функции. Переменная **cnt** содержит информацию о том, сколько раз в процессе интегрирования вычислялась подынтегральная функция.

Функции **[I, cnt] = quad('имя функции', a, b, tol)** и **[I, cnt] = quad8('имя функции', a, b, tol)** вычисляют интеграл с заданной относительной погрешностью **tol**. По умолчанию **tol = 1e-3**.

Функции **[I, cnt] = quad('имя функции', a, b, tol, trace)** и **[I, cnt] = quad8('имя функции', a, b, tol, trace)**, когда аргумент **trace** не равен нулю, строят график, показывающий ход вычисления интеграла.

Пример 2. Вычислим интеграл: $I = \int_0^{\pi} \sin(x)dx$. Его точное значение равно

двум.

[I, cnt] = quad('sin', 0, pi)	[I, cnt] = quad('sin', 0, pi, 1e-4, 1)
I = 2.0000	I = 2.0000
cnt = 17	cnt = 33

Как следует из анализа полученных данных, при увеличении точности вычисления на порядок почти вдвое увеличивается трудоемкость расчетов. График с результатами промежуточных расчетов показан на [рис. 1](#).

Информационные технологии в менеджменте

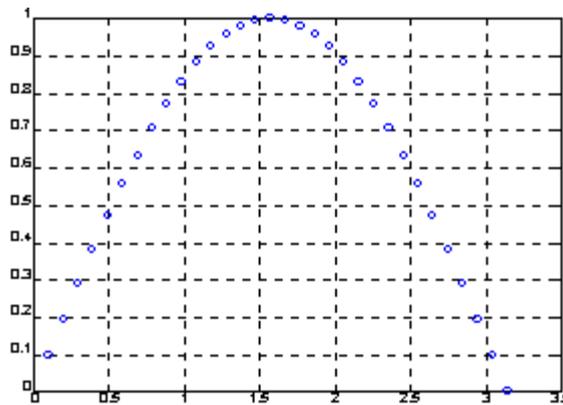


Рис. 1.

Функции **quad** и **quad8** используют рекурсивный вызов. Для того чтобы предотвратить бесконечную рекурсию при вычислении сингулярных интегралов, глубина рекурсии ограничена уровнем 10. При достижении этого ограничения выдается сообщение: «Recursion level limit reached in quad. Singularity likely» (В процедуре **quad** достигнута предельная глубина рекурсии. Функция, возможно, сингулярная).

Функции **quad** и **quad8** не позволяют интегрировать функции с особенностями типа:

$$I = \int_0^1 \frac{1}{\sqrt{x}} dx.$$

В этом случае рекомендуется выделить такие члены и проинтегрировать их аналитически, а к остатку применить процедуры **quad** и **quad8**.

Задания

Все задания рекомендуется выполнять в m-файлах.

Задание 1. Для функции $f(x, y)$, равной следующему выражению:

№	Выражение	№	Выражение
1.	$y \sin x - 3 + \sqrt{6y}(4 + x^2)$	16.	$(1 + e^y)(x^2 + 4) - 8y$
2.	$\sqrt{6x}(4 + x^2) + y + xe^{\frac{3}{y}} - x^2$	17.	$y^7 x - 3x^2 - yx^4$
3.	$y + xe^{\frac{3}{x}} - y^2 - y \sin x - 3$	18.	$(x + \ln y) + (1 + \frac{x}{5} + \sin y)$
4.	$2x^3 - 3y^2 - x^4 + 2tg x - \sin y$	19.	$2y - 5 + 3e^{2x} + xy^2$
5.	$2tg y - \sin x - \sqrt{6y}(4 + x^2)$	20.	$y^6 + x^3 + 45yx + x^2$
6.	$5\cos^2 y + 7 - x^2(y - 1)$	21.	$(1 - x)y^2 + 6x$
7.	$x^2(y - 1) - x^3 - x(y - 2)$	22.	$6y - 2\sqrt{xy} \cdot \ln x$
8.	$\frac{2x(1 - e^{yx})}{(1 + y^2)^2}$	23.	$xy + \frac{x + 3}{x + 2xy^3}$

Информационные технологии в менеджменте

9.	$2 - 9y^2 + (4x^2 - 6x^3)y$	24.	$6yx^4 - 2x + xy^2$
10.	$x(1 + y^2) + 1 + yx^2$	25.	$5 \operatorname{tg}(yx) + \sin((2x) + y^5)$
11.	$\cos y - (x^2 + 1) \cdot \sin(yx)$	26.	$xy^4 + x(2 \ln(xy) + 1)$
12.	$\frac{5}{x(y-1)} + \frac{3y}{x+2}$	27.	$x - 4xy^2 + x^3 e^{-y}$
13.	$y - 45 + \sqrt{x^2 + xy^5}$	28.	$8xy \left(\sqrt[3]{y} - \frac{x^2}{8} \sin(2xy) \right)$
14.	$x \cdot (\ln y^4 - \ln x + 4) - \ln \frac{3}{y}$	29.	$2xy^{-3} + (xy^4 - 3x^2)x^{-4}$
15.	$y^6 + 2x + xy^3$	30.	$xy^3 + \sqrt{2x} - y - 1$

аналитически найдите первую и вторую частные производные по x и по y .

Задание 2. Аналитически найдите производные сложных функций:

1. $z = x \sin y + y \cos x$, где $x = \frac{u}{v}$, $y = u^3 v^2$; $\frac{\partial z}{\partial u}$, $\frac{\partial z}{\partial v} = ?$

2. $z = e^{4xy}$, где $x = \cos(1-t)$, $y = \sin t^2$; $\frac{dz}{dt} = ?$

3. $z = x^2 - y^2 + 2xy$, где $x = \sin t$, $y = \arccos(e^t)$; $\frac{dz}{dt} = ?$

4. $z = (x + y^3) \cdot e^{x^2 + y^2}$, где $x = \cos(t^2)$, $y = \sqrt{t}$; $\frac{dz}{dt} = ?$

5. $z = \arcsin \frac{x^2}{y}$, где $x = \sin t$, $y = \cos^2 \frac{t}{2}$; $\frac{dz}{dt} = ?$

6. $z = \operatorname{tg}(x + 2x^2 - y)$, где $x = \frac{1}{t}$, $y = \sqrt{t}$; $\frac{dz}{dt} = ?$

7. $z = \frac{1}{\sqrt{x^2 + y^2}}$, где $x = v - u^2 v$, $y = u + v^2 u$; $\frac{\partial z}{\partial u}$, $\frac{\partial z}{\partial v} = ?$

$$8. z = \operatorname{arctg} \frac{x}{\sqrt{y}}, \text{ где } y = \sqrt{x^2 + 1}, \quad \frac{dz}{dx} = ?$$

$$9. z = \frac{4y}{\sqrt{y^2 - x}}, \text{ где } x = t \cdot \cos t, y = t \cdot \sin t; \quad \frac{dz}{dt} = ?$$

$$10. z = \arcsin(x - y), \text{ где } x = \ln(\sqrt{t} + 1), y = 4t^3; \quad \frac{dz}{dt} = ?$$

$$11. z = \operatorname{arctg} \frac{x + y}{1 + xy}, \text{ где } x = t^2 + 1, y = \sin t; \quad \frac{dz}{dt} = ?$$

$$12. z = \ln(\sqrt{x} \cdot \ln y), \text{ где } x = \sin t, y = \arccos(t^5); \quad \frac{dz}{dt} = ?$$

$$13. z = \ln(e^{2x} + e^{6y}), \text{ где } y = x\sqrt{x}; \quad \frac{dz}{dx} = ?$$

$$14. z = \frac{1}{2} \ln \frac{x}{y}, \text{ где } x = \operatorname{tg}^2 t, y = \operatorname{ctg}^3 \frac{t}{2}; \quad \frac{dz}{dt} = ?$$

$$15. z = \ln(x^2 y), \text{ где } x = u^v, y = v^u; \quad \frac{\partial z}{\partial u}, \frac{\partial z}{\partial v} = ?$$

$$16. z = x^3 \sin y + y^3 \cos x, \text{ где } x = 3t^2 - \sqrt{t}, y = \frac{t^2}{t-1}; \quad \frac{dz}{dt} = ?$$

$$17. z = \frac{x+2y}{xy}, \text{ где } x = tg(t^2 + 1), y = ctg(t^4 - 1); \frac{dz}{dt} = ?$$

$$18. z = xy^2 + \frac{x}{y}, \text{ где } x = \ln(t^2 + t), y = 10^t; \frac{dz}{dt} = ?$$

$$19. z = \ln(x^2 + y^2) - x\sqrt{x}, \text{ где } x = \sin t, y = \frac{t}{t+1}; \frac{dz}{dt} = ?$$

$$20. z = \frac{x-2y}{x+2y}, \text{ где } x = \frac{u-v}{u}, y = \frac{v}{u+v}; \frac{\partial z}{\partial u}, \frac{\partial z}{\partial v} = ?$$

$$21. z = \frac{x+y}{1-xy}, \text{ где } x = \frac{t}{\sin t}, y = \frac{t}{\cos t}; \frac{dz}{dt} = ?$$

$$22. z = \sqrt{xy + \sin x}, \text{ где } x = tg(3t^2 + t), y = ctg(2t + 1); \frac{dz}{dt} = ?$$

$$23. z = \sqrt{\frac{x+y}{x-y}}, \text{ где } x = tg(e^t + 1), y = ctg \frac{t}{2}; \frac{dz}{dt} = ?$$

$$24. z = x^{\ln y}, \text{ где } x = \sin(uv), y = \cos(v^2 - u); \frac{\partial z}{\partial u}, \frac{\partial z}{\partial v} = ?$$

$$25. z = \frac{\sin^2 x}{\sin^2 y}, \text{ где } x = \ln t, y = te^t; \frac{dz}{dt} = ?$$

$$26. z = \ln x - \frac{y^2}{x}, \text{ где } x = \sin t^2, y = \cos t^2; \frac{dz}{dt} = ?$$

$$27. z = \arccos(xy^2), \text{ где } x = t^2 \ln t, y = te^t; \frac{dz}{dt} = ?$$

$$28. z = \cos(y^{\sin x}), \text{ где } x = t^3, y = 2^t; \frac{dz}{dt} = ?$$

$$29. z = \sin^2 x + \cos^2 y, \text{ где } x = \pi^{\ln t}, y = \ln^\pi t; \frac{dz}{dt} = ?$$

$$30. z = \frac{x^2 + 2y^3}{x + 2y}, \text{ где } x = \frac{u+v^2}{u}, y = \frac{v^2}{u+v}; \frac{\partial z}{\partial u}, \frac{\partial z}{\partial v} = ?$$

При этом необходимо использовать функцию **subs** для замены:

Информационные технологии в менеджменте

$$z1 = \text{subs}(z, [x \ y], [u/v \ u^3*v^2]),$$

где z — исходное выражение, в которое входят символьные переменные x и y , заменяемые на выражения u/v и u^3*v^2 соответственно.

Задание 3. Задайте вектор X размерности 10×1 с произвольными значениями элементов. Численно найдите производные — вычислите первые и третьи конечные разности.

Задание 4. С помощью функции `int` найдите неопределенные интегралы от функции $f(x)$:

Информационные технологии в менеджменте

№	$f(x)$	№	$f(x)$	№	$f(x)$
1	$\frac{1}{\sin^2 x(1 - \cos x)}$	11	$\frac{1 + \cos^2 x}{1 + \cos 2x}$	21	$\frac{\cos x + 1}{(1 + \sin x + \cos x)}$
2	$\frac{\cos x - \sin x}{(1 + \sin x)^2}$	12	$\frac{1}{(\sin x + \cos x)^2}$	22	$\frac{\sin x}{(1 + \sin x + \cos x)}$
3	$\frac{1}{\sin x(1 - \sin x)}$	13	$\frac{1}{(2 \sin x + 3 \cos x)}$	23	$\frac{\cos x}{(1 + \sin x + \cos x)}$
4	$\frac{\cos x}{(5 + 4 \cos x)}$	14	$\frac{1}{(5 - 3 \cos x)}$	24	$\frac{\cos x}{(1 + \sin x + \cos x)^2}$
5	$\frac{\cos x}{(1 + \sin x - \cos x)}$	15	$\frac{2 - \sin x}{(2 + \cos x)}$	25	$\frac{\sin x}{(1 + \sin x)^2}$
6	$\frac{\cos x}{(2 + \sin x)}$	16	$\frac{1}{(5 + 4 \sin x)}$	26	$\frac{\cos x}{(1 - \sin x + \cos x)}$
7	$\frac{\cos x}{(1 - \cos x)^2}$	17	$\frac{1}{\sin^2 x + 1}$	27	$\frac{\sin x}{(1 - \sin x + \cos x)^2}$
8	$\frac{1}{\cos x(1 - \cos x)}$	18	$\frac{1}{1 - \sin^4 x}$	28	$\frac{\sin^2 x}{(1 + \sin x + \cos x)^2}$
9	$\frac{1}{(1 + \sin x - \cos x)^2}$	19	$\frac{1}{1 + \operatorname{tg} x}$	29	$\frac{1}{\sin x(1 + \sin x)}$
10	$\frac{\sin x + 1}{(1 + \sin x - \cos x)}$	20	$\frac{\sin^2 x}{1 - \operatorname{tg} x}$	30	$\frac{\sin x}{(2 + \sin x)}$

Задание 5. Найдите определенные интегралы с помощью функций *int*, *trapz*, *quad*, *quad8*. Сравните результаты.

№	задание	№	задание	№	задание
1	$\int_{1,6}^{2,2} \frac{dx}{\sqrt{x^2 + 2,5}}$;	11	$\int_{0,5}^{1,2} \frac{tg(x^2)}{x+1} dx.$	21	$\int_{1,4}^2 \frac{dx}{\sqrt{2x^2 + 0,7}}$;
2	$\int_{0,6}^{1,6} \frac{dx}{\sqrt{x^2 + 0,8}}$;	12	$\int_{1,3}^{2,1} \frac{\sin(x^2 - 1)}{2\sqrt{x}} dx.$	22	$\int_{3,2}^4 \frac{dx}{\sqrt{0,5x^2 + 1}}$;
3	$\int_{1,2}^2 \frac{dx}{\sqrt{x^2 + 1,2}}$;	13	$\int_{0,2}^{1,0} (x+1) \cos(x^2) dx.$	23	$\int_{1,2}^{2,8} \frac{\lg(1+x^2)}{2x-1} dx.$
4	$\int_{1,2}^2 \frac{\lg(x^2 + 3)}{2x} dx.$	14	$\int_{0,8}^{1,2} \frac{\sin(x^2 - 0,4)}{x+2} dx.$	24	$\int_{0,8}^{4,7} \frac{dx}{\sqrt{2x^2 + 0,3}}$;
5	$\int_{2,5}^{3,3} \frac{\lg(x^2 + 0,8)}{x-1} dx.$	15	$\int_{0,15}^{0,63} \sqrt{x+1} \lg(x+3) dx.$	25	$\int_{1,2}^{2,0} \frac{dx}{\sqrt{0,5x^2 + 1,5}}$;
6	$\int_{2,1}^{3,6} \frac{dx}{\sqrt{x^2 - 3}}$;	16	$\int_{1,3}^{2,5} \frac{dx}{\sqrt{0,2x^2 + 1}}$;	26	$\int_{0,6}^{1,4} \frac{dx}{\sqrt{12x^2 + 0,5}}$;
7	$\int_{0,4}^{1,2} \frac{\cos(x^2)}{x+1} dx.$	17	$\int_{0,6}^{1,4} x^2 \cos x dx.$	27	$\int_{1,3}^{2,1} \frac{dx}{\sqrt{3x^2 - 0,4}}$;
8	$\int_{0,6}^{0,72} (\sqrt{x} + 1) tg 2x dx.$	18	$\int_{0,8}^{1,2} \frac{\cos x}{x^2 + 1} dx.$	28	$\int_{1,4}^{2,6} \frac{dx}{\sqrt{1,5x^2 + 0,7}}$;
9	$\int_{1,6}^{3,2} \frac{x}{2} \lg\left(\frac{x^2}{2}\right) dx.$	19	$\int_{1,2}^{2,8} \left(\frac{x}{2} + 1\right) \sin \frac{x}{2} dx.$	29	$\int_{0,8}^{1,6} \frac{\lg(x^2 + 1)}{x+1} dx.$
10	$\int_{0,32}^{0,66} \frac{dx}{\sqrt{x^2 + 2,3}}$;	20	$\int_{2,3}^{0,5} \frac{dx}{\sqrt{x^2 - 4}}$;	30	$\int_{0,15}^{0,5} \frac{dx}{\sqrt{2x^2 + 1,6}}$;

Лабораторная работа

Matlab: Типовые средства программирования

Цель работы: получение навыков программирования в пакете Matlab.

Теоретические сведения

Назначение языка программирования Matlab

Практически невозможно предусмотреть в одной, даже самой большой и мощной математической системе возможность решения всех задач, которые могут интересовать пользователя. *Программирование* в системе MATLAB является эффективным средством ее расширения и адаптации к решению специфических проблем. Оно реализуется с помощью *языка программирования* системы.

Информационные технологии в менеджменте

Большинство объектов этого языка, в частности все команды, операторы и функции, одновременно являются объектами *входного языка* общения с системой в командном режиме работы.

Так в чем же отличие входного языка от языка программирования? В основном — в способе фиксации создаваемых ими кодов. Сессии в командном режиме работы не сохраняются в памяти компьютера (ведение дневника не в счет). Хранятся только определения созданных в ходе их выполнения переменных и функций. А вот программы на языке программирования MATLAB сохраняются в виде текстовых *m*-файлов. При этом могут сохраняться как целые программы в виде файлов-сценариев, так и отдельные полноценные *программные модули* — функции. Кроме того, важно, что программа может менять структуру алгоритмов вычислений в зависимости от входных данных и данных, создаваемых в ходе вычислений.

С позиций программиста язык программирования системы является типичным *проблемно-ориентированным* языком программирования высокого уровня интерпретирующего типа. Точнее говоря, это даже язык *сверхвысокого* уровня, содержащий сложные операторы и функции, реализация которых на обычных языках (например, Бейсике, Паскале или Си) потребовала бы много усилий и времени. К таким функциям относятся матричные функции, функции быстрого преобразования Фурье (БПФ) и др., а к операторам — операторы построения разнообразных графиков, генерации матриц определенного вида и т.д.

Управляющие структуры

Помимо программ с *линейной структурой*, инструкции которых исполняются строго по порядку, существует множество программ, структура которых *нелинейна*. При этом ветви программ могут выполняться в зависимости от определенных условий, иногда с конечным числом повторений — циклов, иногда в виде циклов, завершаемых при выполнении заданного условия. Практически любая серьезная программа имеет нелинейную структуру. Для создания таких программ необходимы специальные *управляющие структуры*.

Диалоговый ввод. Пример диалоговой программы:

% Вычисление длины окружности с диалоговым вводом радиуса

```
r=0;
while r>=0,
r=input('Введите радиус окружности r=');
if r>=0 disp(' Длина окружности l='); disp(2*pi*r), end;
end;
```

Эта программа служит для многократного вычисления длины окружности по вводимому пользователем значению радиуса *r*. Диалог реализован с помощью команды **input**:

```
r=input('Введите радиус окружности r=');
```

При выполнении этой команды вначале выводится запрос в виде строки, затем происходит остановка работы программы и ожидается ввод значения радиуса *r* (в общем случае числа). Ввод, как обычно, подтверждается нажатием клавиши **Enter**, после чего введенное число присваивается переменной *r*. Следующая строка:

```
if r>=0 disp(' Длина окружности l='); disp(2*pi*r), end
```

с помощью команды **disp** при **r>=0** выводит надпись «**Длина окружности l=**» и

Информационные технологии в менеджменте

вычисленное значение длины окружности. Она представляет собой одну из наиболее простых управляющих структур типа **if...end**. В данном случае она нужна для остановки вычислений, если вводится отрицательное значение **r**.

Приведенные строки включены в управляющую структуру **while...end**. Это необходимо для циклического повторения вычислений с вводом значений **r**. Пока **r >= 0**, цикл повторяется. Но стоит задать **r < 0**, вычисление длины окружности перестает выполняться, а цикл завершается.

Если данная программа записана в виде m-файла **circ.m**, то работа с ней будет выглядеть следующим образом:

```
>> circ
Введите радиус окружности R=1
Длина окружности l=
    6.2832
Введите радиус окружности R=2
Длина окружности l=
   12.5664
Введите радиус окружности R=-1
>>
```

Итак, на примере даже простой программы мы видим пользу применения управляющих структур типа **if...end** и **while...end**, а также функций диалогового ввода **input('String')** и вывода **disp**.

Функция **input** может использоваться и для ввода произвольных строковых выражений. При этом она задается в следующем виде:

```
input('Комментарий','s')
```

При выполнении этой функции она останавливает вычисления и ожидает ввода строкового комментария. После ввода возвращается набранная строка. Это иллюстрирует следующий пример:

```
>> S=input('Введите выражение ','s')
Введите выражение (Вводим) 2*sin(1)
S =
2*sin(1)
>> eval(S)
ans =
1.6829
```

Обратите внимание на то, что функция **eval** позволяет вычислить выражение, заданное (полученное от функции **input**) в символьном виде. Вообще говоря, возможность ввода любого символьного выражения в сочетании с присущими языку программирования MATLAB управляющими структурами открывает путь к созданию диалоговых программ любой сложности.

Условный оператор if...elseif...else...end. Условный оператор **if** в общем виде записывается следующим образом:

```
if Условие
    Инструкции_1
elseif Условие
    Инструкции_2
else
    Инструкции_3
end
```

Эта конструкция допускает несколько частных вариантов. В простейшем, типа **if...end**:

if Условие Инструкции end

пока **Условие** возвращает логическое значение 1 (то есть «истина»), выполняются **Инструкции**, составляющие тело структуры **if...end**. При этом оператор **end** указывает на конец перечня инструкций. Инструкции в списке разделяются оператором **,** (запятая) или **;** (точка с запятой). Если **Условие** не выполняется (дает логическое значение 0, то есть «ложь»), то **Инструкции** также не выполняются.

Еще одна конструкция

if Условие Инструкции_1 else Инструкции_2 end

выполняет **Инструкции_1**, если выполняется **Условие**, или **Инструкции_2** в противном случае.

Условия записываются в виде:

Выражение_1 Оператор_отношения Выражение_2

причем в качестве **Операторов_отношения** используются следующие операторы: **==, <, >, <=, >=** или **~=**. Все эти операторы представляют собой пары символов без пробелов между ними.

Циклы типа for...end. Циклы типа **for...end** обычно используются для организации вычислений с заданным числом повторяющихся циклов. Конструкция такого цикла имеет следующий вид:

for var=Выражение, Инструкция, ..., Инструкция end

Выражение чаще всего записывается в виде **s:d:e**, где **s** — начальное значение переменной цикла **var**, **d** — приращение этой переменной и **e** — конечное значение управляющей переменной, при достижении которого цикл завершается. Возможна и запись в виде **s:e** (в этом случае **d=1**). Список выполняемых в цикле инструкций завершается оператором **end**.

Точка с запятой после выражения в теле цикла предотвращает повторения вывода результатов на экран.

Хорошим стилем являются отступы при использовании циклов для лучшей читаемости, особенно, когда они вложенные.

Пример 1. Применение цикла для получения квадратов значений переменной цикла:

```
for i=1:5
    i^2,
end;
ans = 1
ans = 4
ans = 9
ans = 16
ans = 25
for x=0:.25:1
    x^2,
end;
ans = 0
ans = 0.0625
ans = 0.2500
ans = 0.5625
ans = 1
```

Возможны *вложенные циклы*.

Пример 2. Программа создания вектора *A*:

```
n=10;
```

```

for k = 1:n;
    if k<=3
        A(k)=k+1;
    elseif (k>3 & k<=10)
        A(k)=k+2;
    else
        A(k)=k+3;
    end;
end;

A =

    2  3  4  6  7  8  9  10  11  12

```

Пример 3. Программа создания матрицы H :

```

for i = 1:m
    for j = 1:n
        H(i,j) = 1/(i+j);
    end;
end;

```

Пример 4. Программа создания матрицы A :

```

for i=1:3
    for j=1:3
        A(i,j)=i+j;
    end
end

```

В результате выполнения этого цикла формируется матрица A :

```

>> A
A =
    2  3  4
    3  4  5
    4  5  6

```

Следует отметить, что формирование матриц с помощью оператора `:` (двоеточие) обычно занимает меньше места при записи и дает меньшее время исполнения, чем с помощью цикла. Однако применение цикла нередко оказывается более наглядным и понятным.

MATLAB допускает использование в качестве переменной цикла массива A размера $m \times n$. При этом цикл выполняется столько раз, сколько столбцов в массиве A , и на каждом шаге переменная `var` представляет собой вектор, соответствующий текущему столбцу массива A :

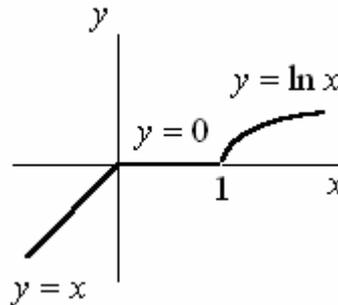
```

>> A=[1 2 3;4 5 6]
A =
    1  2  3
    4  5  6
>> for var=A; var, end
var =
    1
    4
    2
    5
    3

```

6

Пример 5. Для функции, заданной графически, вычислите значения функции y для всех значений аргумента x на заданном интервале $[x_H, x_K]$ с заданным шагом d_x .



%Вычисление функции

```
deltax=0.05;
xn=-2;
xk=2;
i=1;
for xx=xn:deltax:xk;
    if xx<0;
        y(i)=xx;
    end;
    if xx<=1 && xx>=0;
        y(i)=0;
    end;
    if xx>1;
        y(i)=log(xx);
    end;
    x(i)=xx;
    i=i+1;
end;
```

```
plot(x,y);
grid on;
```

Конструкция переключателя switch...case...end. Для осуществления множественного выбора (или ветвления) используется конструкция с переключателем типа **switch**:

```
switch switch_Выражение
case case_Выражение
Список_инструкций
case {case_Выражение1, Case_выражение2, case_Выражение3,...}
Список_инструкций
...
otherwise,
Список_инструкций
end
```

Если выражение после заголовка **switch** имеет значение одного из выражений **case_Выражение...**, то выполняется блок операторов **case**, в противном случае — список инструкций после оператора **otherwise**. При выполнении блока **case** исполняются те списки инструкций, для которых

case_Выражение совпадает с **switch_Выражением**. Обратите внимание на то, что **case_Выражение** может быть числом, константой, переменной, вектором ячеек или даже строчной переменной. В последнем случае оператор **case** истинен, если функция **strcmp(значение, выражение)** возвращает логическое значение «истинно».

Выполняется только первый соответствующий случай.

Замечание для программистов Си. В отличие от языка Си, оператор **switch** в Matlab не «проваливается». Если первый случай является истинным, другие случаи не выполняются. Таким образом, нет необходимости в использовании оператора **break**.

Пример 6.

```
g=3
switch g
  case {1}
    h=10;
  case {2}
    h=20;
  case {3}
    h=30;
end;
```

Циклы типа while...end. Цикл **while** повторяет группу операторов определенное число раз, пока выполняется логическое условие. Ключевое слово **end** очерчивает используемые операторы.

Пример 7. Программа создания меню для выбора пользователем действия:

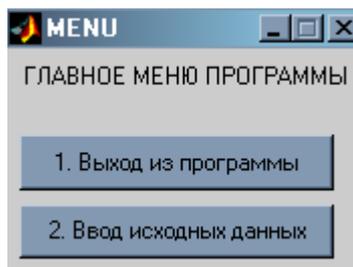
```
clear all;
id=0;
while not(id==1); %выход из программы

id=menu(['ГЛАВНОЕ МЕНЮ ПРОГРАММЫ'];{''},...
'1. Выход из программы',...
'2. Ввод исходных данных');
```

```
switch id

  case {2}
    clc;
    data; %вызов программы ввода исходных данных
  end;
end;

clear id;
```



Пример 8. Программа создания меню для выбора пользователем действия:

```
g=0;
```

```
while not(g==1);
```

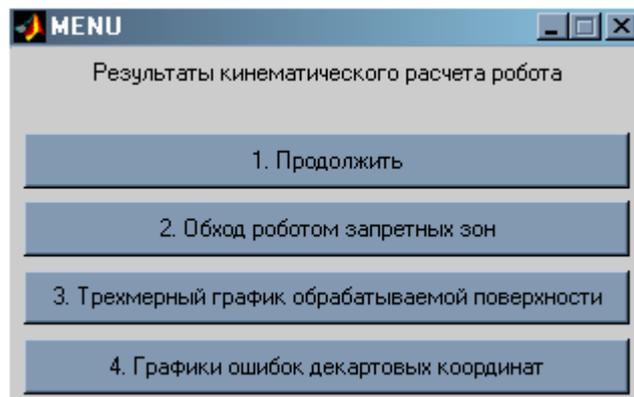
```
g=menu(['Результаты кинематического расчета робота'];{''}),...
'1. Продолжить',...
'2. Обход роботом запретных зон',...
'3. Трехмерный график обрабатываемой поверхности',...
'4. Графики ошибок декартовых координат');
```

```
switch g
```

```
case {1}
dinamica; %вызов файла
```

```
case {2} Инструкции
case {3} Инструкции
case {4} Инструкции
```

```
end;
end;
```



Операторы *break*, *continue* и *return*. В управляющих структурах, в частности в циклах **for** и **while**, часто используются операторы, влияющие на их выполнение. Так, оператор **break** может использоваться для досрочного прерывания выполнения цикла. Как только он встречается в программе, цикл прерывается. Во вложенных циклах **break** осуществляет выход только из самого внутреннего цикла.

Пример 9.

```
for i=1:10
    i,
    if i==5
        break,
    end,
end;
i = 1
i = 2
i = 3
i = 4
i = 5
```

Информационные технологии в менеджменте

Оператор **continue** передает управление в следующую итерацию цикла, пропуская операторы, которые записаны за ним, причем во вложенном цикле он передает управление на следующую итерацию основного цикла.

Оператор **return** обеспечивает нормальный возврат в вызывающую функцию или в режим работы с клавиатурой.

Пример 10.

```
if isempty(A)
```

```
d = 1;
```

```
return
```

```
else
```

```
...
```

```
end
```

В данном примере, если матрица A пустая, будет выведено значение 1, после чего управление будет передано в блок **else...end**.

Векторизация циклов

MATLAB — это язык, специально разработанный для обработки массивов и выполнения матричных операций. Всюду, где это возможно, пользователь должен учитывать это обстоятельство.

Под векторизацией понимается преобразование циклов **for** и **while** к эквивалентным векторным или матричным выражениям. При векторизации алгоритма ускоряется выполнение М-файла.

Пример 1. Один из способов вычислить 1001 значение функции синуса на интервале $[0\ 10]$, используя оператор цикла:

```
i = 0;
```

```
for t = 0:.01:10
```

```
    i = i + 1;
```

```
    y(i) = sin(t);
```

```
end;
```

Эквивалентная векторизованная форма имеет вид:

```
t = 0:.01:10;
```

```
y = sin(t);
```

Пример 2. Программа создания таблицы логарифмов:

```
x = 0;
```

```
for k = 1:1001
```

```
    y(k) = log10(x);
```

```
    x = x + .01;
```

```
end;
```

Векторизованная версия этого кода выглядит следующим образом:

```
x = 0: .10:10;
```

```
y = log10(x);
```

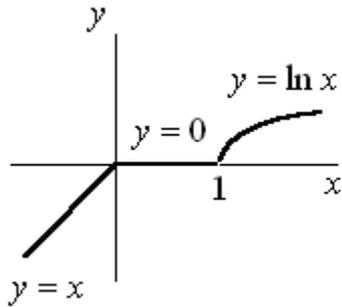
Для более сложных программ возможности векторизации не так очевидны. Однако, когда важна скорость, вы должны всегда искать способы векторизации вашего алгоритма.

Задания

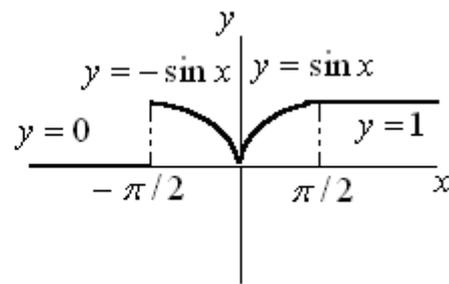
Все задания рекомендуется выполнять в m-файлах.

Задание 1. Напишите программу для вычисления значений функции y для всех значений аргумента x на заданном интервале $[x_H, x_K]$ с заданным шагом d_x с использованием операторов ветвления **if** и оператора цикла **for**.

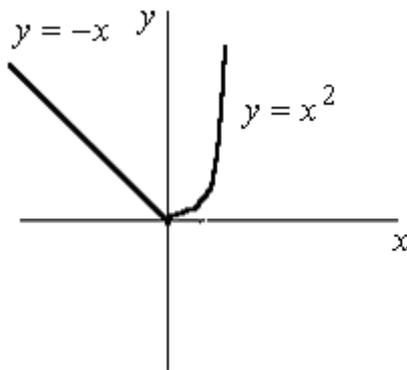
Результат выведите на экран в виде графика.



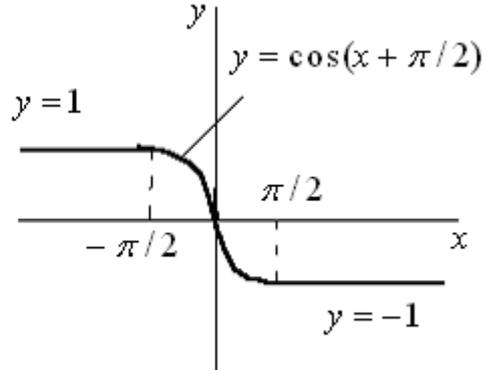
1)



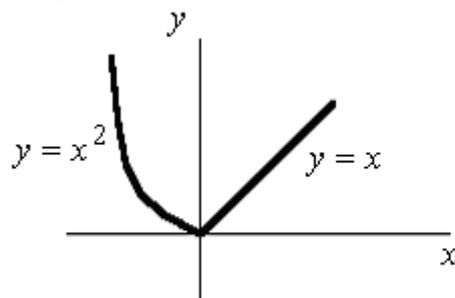
2)



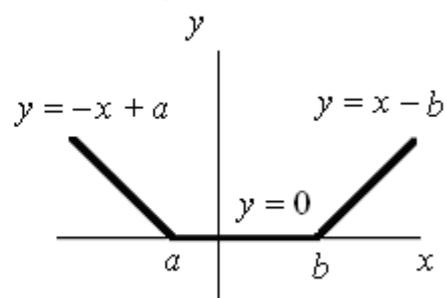
3)



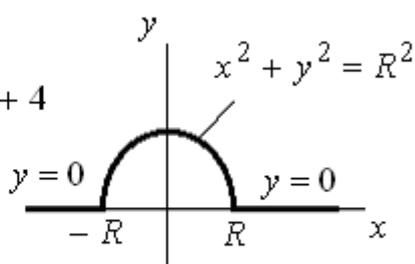
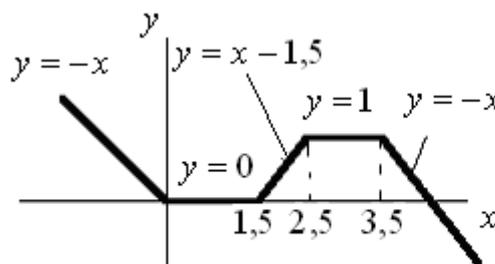
4)



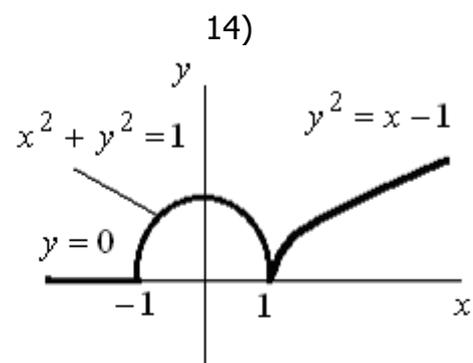
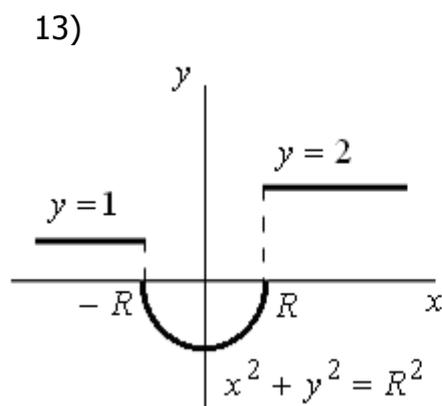
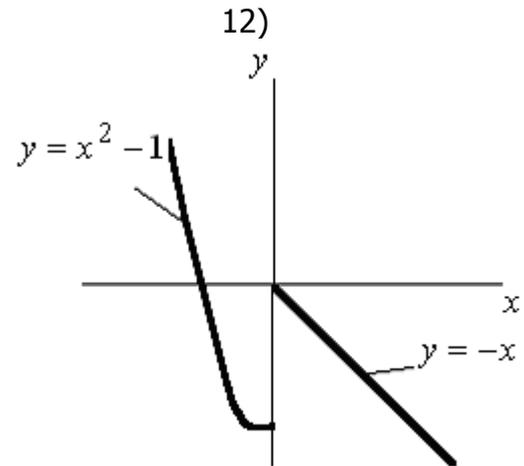
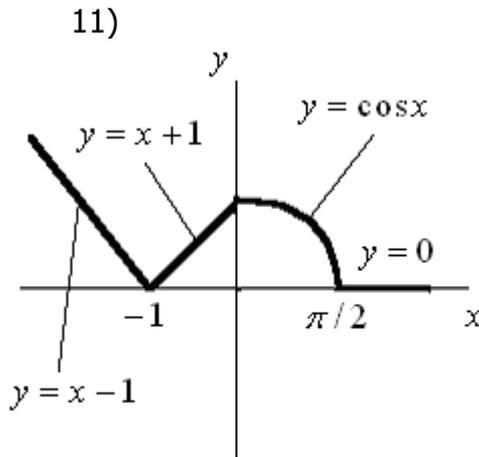
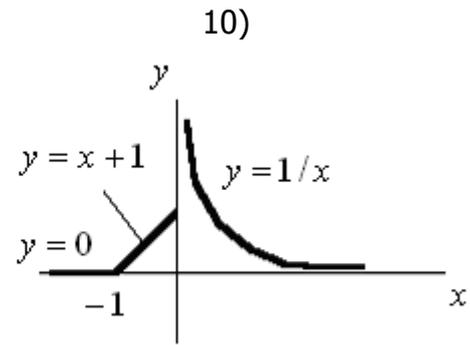
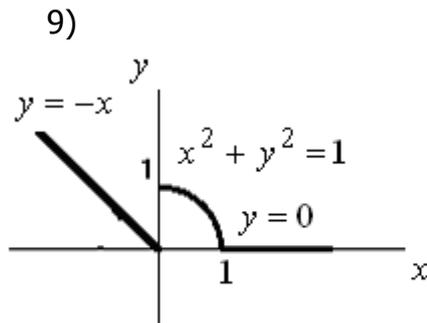
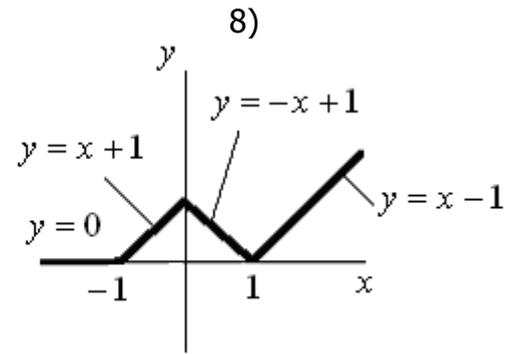
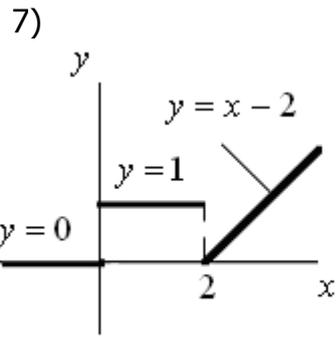
5)



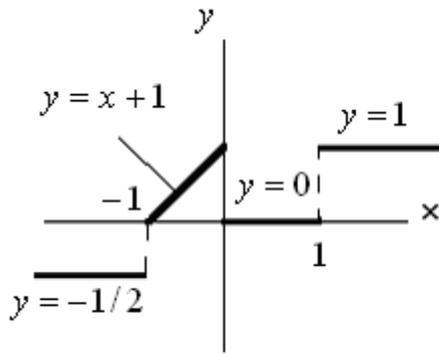
6)



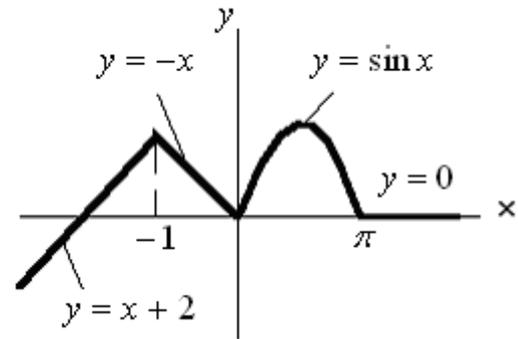
Информационные технологии в менеджменте



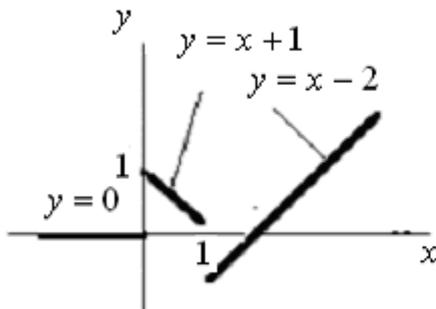
15)



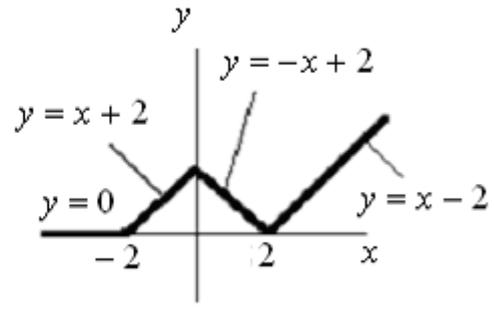
16)



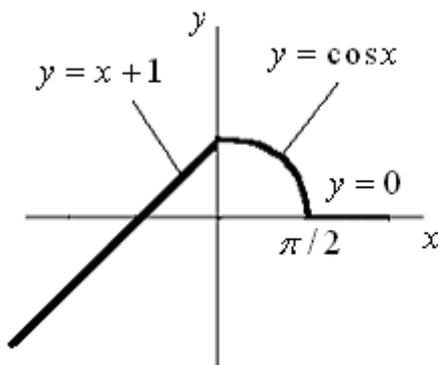
17)



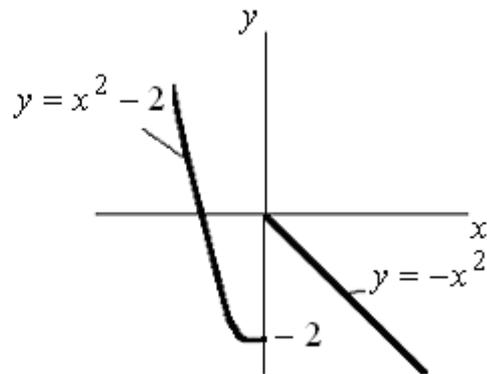
18)



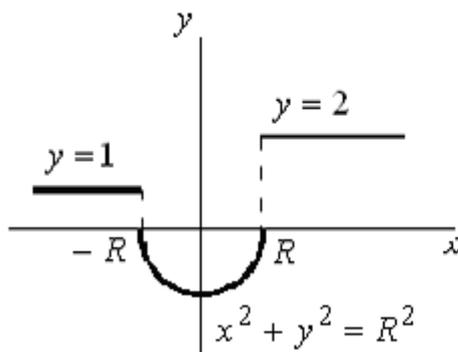
19)



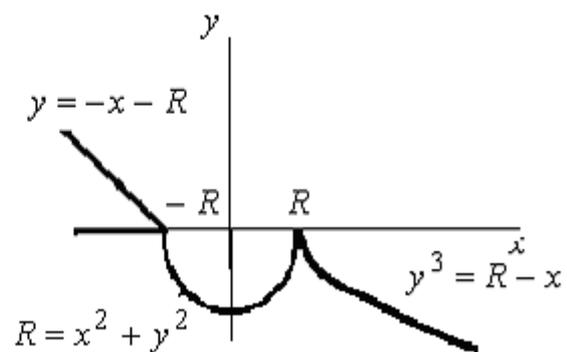
20)



21)

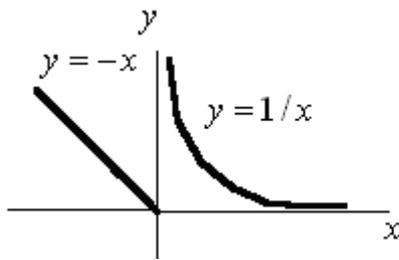


22)



23)

24)



25)

Задание 2. Напишите программу вычисления функции $F(x)$ с использованием операторов ветвления **if** и оператора цикла **for**. Диапазон изменения аргумента задайте самостоятельно.

Результат выведите на экран в виде графика.

1., 11., 21. $F = \frac{3y + x^2z}{\pi}$	$\begin{cases} y = \frac{-8x \cdot \sin x}{e^{\sqrt{ x }}}, & z = \frac{8}{-x}, & x \leq 0; \\ y = \frac{0,8x}{ \sin x } + \cos(x - \frac{\pi}{3}), & z = \frac{2x}{\sqrt{x^3 - 1}}, & x > 0. \end{cases}$
2., 12., 22. $F = \sqrt{ xyz } - \frac{z}{2+x}$	$\begin{cases} y = \frac{\pi \sin 2x + 2}{\ln(x+2)}, & z = \ln x - 8, & x > 1,5; \\ y = \sqrt{\frac{2x^2 - 1,5}{ \cos 2x - \frac{\pi}{4} }}, & z = \frac{e^{-x}}{x + 2 \ln x }, & x \leq 1,5. \end{cases}$
3., 13., 23. $F = (xy + z)^2$	$\begin{cases} y = \frac{(x^2 + \pi)^3}{2x - \operatorname{ctg} \frac{x}{2}}, & z = \frac{ x - \sin 2x }{\pi}, & x < 0; \\ y = \sqrt{\frac{\ln 2x + 0,5}{15}}, & z = \frac{\pi x}{2x + \cos 3x} + 1, & x > 0. \end{cases}$
4., 14., 24. $F = \ln(y + z)$	$\begin{cases} y = \frac{\ln(2x)}{e^{3x^2}}, & z = \frac{\cos 2x}{\cos^2 x} & x \leq 2,5; \\ y = \frac{2,1x \cdot \lg x}{\sqrt{2x - 3} + 10}, & z = \frac{\sin 2x}{x + \frac{\pi}{3}} & x > 2,5. \end{cases}$
5., 15., 25. $F = x + 3\frac{y}{z}$	$\begin{cases} y = \frac{\operatorname{arctg}(x)}{2 + x^2}, & z = \sin 2x & x < 1; \\ y = \frac{ 2 - x }{3,1 \operatorname{tg}(x) - \pi}, & z = \frac{e^{x+1}}{\sin x - 2 \cos 2x} & x \geq 1. \end{cases}$

$$\begin{aligned}
 6., 16., 26. \quad F = e^{xy} - z & \quad \begin{cases} y = \frac{|\operatorname{tg}x| - 2}{\sqrt{|x|} + x^2}, & z = \frac{1}{\sin(x) - \frac{\pi}{3}}, & x < 0; \\ y = 3\operatorname{ctg}x, & z = \frac{\sin x}{2,15 + \cos 3x}, & x \geq 0. \end{cases} \\
 7., 17., 27. \quad F = 2x^3 + \frac{y}{z} & \quad \begin{cases} y = \frac{x^2 - 3x}{\ln 3x - 2}, & z = \frac{\operatorname{ctg}x - 1,1}{\cos x^2 + \sin^3 x}, & x \geq 3,5; \\ y = \frac{\sin^2 x}{1 + \sqrt{\ln|2x|}}, & z = \sqrt[5]{|2x^3|}, & x < 3,5. \end{cases} \\
 8., 18., 28. \quad F = \frac{\sin x + \cos 2y}{z + \frac{\pi}{4}} & \quad \begin{cases} y = \frac{|x+2| - 2}{3\sqrt{|x^3|}}, & z = \frac{x + \sin x}{\cos x - 1,2} & x < 0; \\ y = \frac{e^{\sqrt{x}} + 1}{\sqrt[3]{x^2} + 2}, & z = \frac{\cos x + 1}{x^3 - \sqrt{3x^2} 3} & x > 0. \end{cases} \\
 9., 19., 29. \quad F = \sqrt{xyz^3} - 1 & \quad \begin{cases} y = \frac{\operatorname{arctg}x + 1}{\operatorname{ctg}(x - \frac{\pi}{2})}, & z = \frac{\sin x + x^2}{\cos(x+1) - 1}, & x > 0,5; \\ y = x^3 + 2x^2 + 5, & z = \frac{1}{|\operatorname{tg}x + 1|}, & x \leq 0,5. \end{cases} \\
 10., 20., 30. \quad F = \frac{x + z^2}{(x + y + z)^2} & \quad \begin{cases} y = \frac{2x^2 - 5}{\sqrt{x^3} - \frac{2}{x+4}}, & z = \sqrt[3]{3x - 2} & x > 0; \\ y = \frac{3x^4 - |5 - x|}{\lg|x| + 3}, & z = \frac{5x + 2}{\operatorname{tg}|x - 2| - 0,3} & x \leq 0. \end{cases}
 \end{aligned}$$

Лабораторная работа

Matlab: Составление функций пользователя

Цель работы: получение навыков составления функций пользователя в пакете Matlab.

Теоретические сведения

Функции, задаваемые пользователем

В пакете Matlab существует три способа задания функций пользователя.

1. Использование команды **inline** приводит к созданию функции, зависящей от указанных аргументов.

Пример 1. Создание функции одной переменной:

```
>> f=inline('x^2','x')
f =
```

Inline function:

$f(x) = x^2$

>> f(5)

ans =

25

Пример 2. Создание функции нескольких переменных:

>> f=inline('x^2+y^5+sin(z)','x','y','z')

f =

Inline function:

$f(x,y,z) = x^2+y^5+\sin(z)$

>> f(5,6,9)

ans =

7.8014e+003

>> f(0.5,0.6,0.9)

ans =

1.1111

Поскольку большинство функций программы Matlab могут оперировать как векторами, так и скалярами, чтобы быть уверенным, что созданная функция может оперировать с векторами, необходимо использовать поэлементные операции умножения, деления и возведения в степень (`.* ./ .^`). Таким образом, получается векторизованная версия функции.

Пример 3. Если ставится задача вычислить значения функции в некотором диапазоне значений аргумента, то необходимо использовать поэлементные операции:

>> f=inline('x.^2','x')

f =

Inline function:

$f(x) = x.^2$

>> f(5:9)

ans =

25 36 49 64 81

%-----

>> f=inline('x.^2+y.^5+sin(z)','x','y','z')

f =

Inline function:

$f(x,y,z) = x.^2+y.^5+\sin(z)$

>> f(0:3,0:3,0:3)

ans =

0 2.8415 36.9093 252.1411

2. Использование оператора @ позволяет создавать так называемые *анонимные функции*. Этот метод является новым в программе Matlab 7, и в настоящее время этому методу отдается предпочтение.

Анонимная (безымянная) функция — в программировании особый вид функций, которые объявляются в месте использования и не получают уникального идентификатора для доступа к ним. Обычно при создании они либо вызываются напрямую, либо ссылка на функцию присваивается переменной, с помощью которой затем можно косвенно вызывать данную функцию.

Пример 4.

>> f=@(x)x.^2

f =

```
@(x)x.^2
>> f=@(x,y,z) x.^2+y.^5+sin(z);
>> f([1 2],[3 4])
```

дает значения функции в точках (1, 3) и (2, 4).

3. Задание функции в отдельных файлах (создание файла-функции). Этот метод подробно рассмотрим в следующем разделе.

Понятие о файлах-функциях

Кроме *m*-файла сценария, в Matlab существует *m*-файл функция. Структура *m*-файла функции имеет вид:

```
function [Список выходных параметров] = funcname(Список входных параметров)
```

Текст программы, реализующей алгоритм функции

Выходной параметр 1 = выражение

Выходной параметр 2 = выражение

...

Выходной параметр N = выражение

Правила создания функций пользователя:

1. Файл, содержащий программу реализации функции, должен иметь расширение «**m**» и называться также как имя функции. Matlab автоматически присваивает данное имя при выполнении команды **Save as**.

2. Первая строка файла должна содержать описание функции в виде:

```
function [x1,x2,...,xN] = funcname(param1, param2, ...,paramM);
```

где **function** — ключевое слово; **x1,x2,...,xN** — переменные, необходимые для сохранения результата функции (выходные параметры функции); **param1, param2, ...,paramM** — параметры, передаваемые в функцию.

3. Далее следует текст программы, реализующей алгоритм функции. Каждое действие в программе должно быть завершено символом точкой с запятой.

4. В основном файле происходит вызов функции путем упоминания имени, например, если необходимо в переменные **y1,y2,...,yN** записать результат работы функции:

```
[y1,y2,...,yN] = funcname(param1, param2, ...,paramM);
```

Причем, имена возвращаемых переменных и имена параметров функции могут быть разными в основной программе и в функции, главное соблюдать порядок их перечисления.

Функция может вызываться в цикле, например:

```
for i=1:10;
    y(i) = funcname(param1(i), param2(i), ...);
end;
```

В этом случае **param1, param2, ...** должны быть векторами соответствующей размерности.

Все переменные, используемые в файле-функции, являются локальными, т. е. действуют только в пределах тела функции.

Файл-функция является самостоятельным программным модулем, который связан с другими модулями и головной программой через входные и выходные параметры. При обнаружении файла-функции он компилируется и затем исполняется.

Таким образом, пользователь Matlab может сам создавать функции, аналогичные тем, которые входят в этот программный пакет. Создав такую функцию, можно пользоваться ею в среде Matlab, набирая ее имя и параметры в скобках.

Пример 1.

Напишите функцию пользователя, которая возвращает вектор, элементы которого являются значениями функции $f = y^2 \cdot t^3 + x \cdot t^2$.

Входными параметрами функции должны быть значения переменных x и y , а также вектор значений моментов времени t .

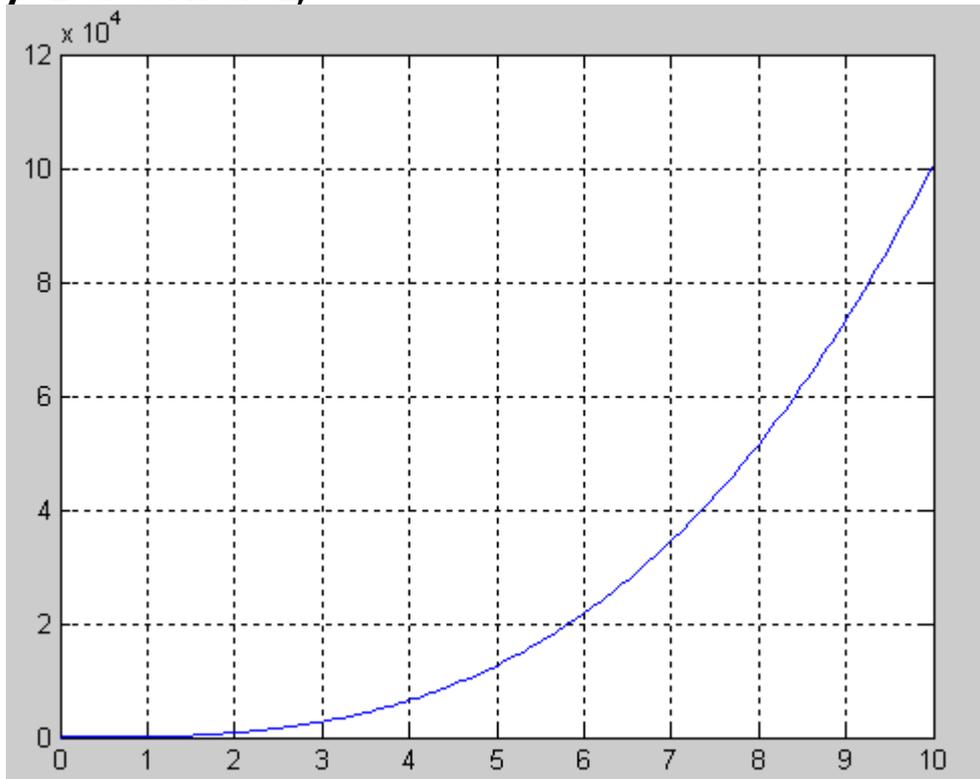
Результат отобразите в виде графика зависимости $f(t)$.

Содержание файла основной программы:

```
x=5;
y=10;
deltat=0.05;
tk=10;
t=0:deltat:tk;
f=fun_07_1(x,y,t); %вызов функции
figure
plot(t,f)
grid on
```

Содержание файла-функции **fun_07_1**:

```
function [f]=fun_07_1(x,y,t);
f=y^2*t.^3+x*t.^2;
```



Пример 2. Напишите функцию пользователя, которая возвращает значения функции $f = y^2 \cdot t^3 + x \cdot t^2$ в дискретных точках t .

Входными параметрами функции должны быть значения переменных x и y , а также дискретные значения переменной t .

Результат отобразите в виде графика зависимости $f(t)$.

Содержание файла основной программы:

```
i=0;
```

```

for t=0:deltat:tk;
    i=i+1;
    f(i)=fun_07_2(x,y,t);
end;

```

```

t=0:deltat:tk;
figure
plot(t,f);
grid on;

```

Содержание файла-функции **fun_07_2**:

```

function [f]=fun_07_2(x,y,t);
f=y^2*t^3+x*t^2;

```

Глобальные переменные

Если необходимо чтобы более одной функции использовали некоторую переменную можно объявить ее как глобальную во всех функциях, например:

```
global XX.
```

Хотя это не обязательно, но использование больших букв для имени глобальной переменной поможет отличить их от других переменных.

Определение **global** должно быть до использования самой переменной. Например, создадим m-файл **falling.m**:

```

function h = falling(t)
global GRAVITY
h = 1/2*GRAVITY*t^2;

```

В основном файле или командном окне запишем:

```

global GRAVITY
GRAVITY = 32;
y = falling((0: .1: 5)');

```

Таким образом, строки определения **GRAVITY** делают ее доступной внутри функции. Можно впоследствии изменить **GRAVITY** и получить новое решение, не редактируя какие-либо файлы-функции.

Задания

Все задания рекомендуется выполнять в m-файлах.

Задание 1. Создайте функцию $f(t)$ с помощью команды **inline** и оператора **@**. Вычислите значения функции в диапазоне изменения аргумента $t = -Nv : 0.1 : Nv$, где Nv — номер варианта.

Результат отобразите в виде графика зависимости $f(t)$.

Задание 2. Напишите функцию пользователя, которая возвращает вектор, элементы которого являются значениями функции f .

Входными параметрами функции должны быть значения переменных x и y , а также вектор значений моментов времени t .

Результат отобразите в виде графика зависимости $f(t)$.

Задание 3. Напишите функцию пользователя, которая возвращает значения функции f в точках дискретизации. По возможности используйте векторизацию.

Входными параметрами функции должны быть значения переменных x и y , а также дискретные значения переменной t .

Результат отобразите в виде графика зависимости $f(t)$.

Информационные технологии в менеджменте

Задание 4. Сравните результаты, полученные в процессе выполнения заданий 1—3. Если графики функции $f(t)$ отличаются, следовательно, в программах имеются ошибки.

Варианты.

№	Выражение $f(t)$	№	Выражение $f(t)$
1.	$y^2 \cdot t^3 + x \cdot t^2 + x^2 y \cdot t - y$	16.	$xyt^3 + \cos^2(xt) - 2y^4$
2.	$y \cos(xt) + \frac{y}{\ln y} t$	17.	$x^2 yt^3 + 2xyt^2 + \ln xt + 10x$
3.	$xyt^2 + yt - 4xt - 1$	18.	$xyt^3 - y(\ln y - \ln x)t^2 + \cos(xt)$
4.	$y \sin^2(x \cdot t) + y \ln y \cdot t$	19.	$y/2 + \operatorname{arctg}(xt)$
5.	$3e^x \operatorname{tg} yt + (1 - e^x) \cos^{-2}(yt)$	20.	$xe^{-t^2/2} + \cos(yt)$
6.	$yt^3 - y \sin(xt) - \frac{1}{\cos(xt)}$	21.	$e^{-1t^2} / y - \cos(xt + 4)$
7.	$(1 + x^2)yt^2 - 2xyt + (1 + x^2)^2$	22.	$- x ^3 e^{-yt^2/2}$
8.	$x(y^2 - 3)t^2 + x^2 yt + 0,9$	23.	$xy^2 \ln(x^2 t)$
9.	$x^2 yt^3 - 2xyt^2 + y^2 t + zx$	24.	$xy \ln^2(yt)$
10.	$(xy - 1)t^3 - 2yt^2 + 45yt + 14$	25.	$x \operatorname{arctg}(yt)$
11.	$y^2 x \cos(xt^2) + (y + 1)t + 12x$	26.	$\sin(xyt) + \cos(6yt)$
12.	$2xy^2 t^2 + (2y - x^2)t + x$	27.	$-y \ln(t) / x^2 + \sin(xt) + \cos(yt)$
13.	$(1 + x^2)yt^2 - 2xy \sin(yt)$	28.	$x^2 \ln^2(yt) + xy^4$
14.	$x^2 yt^3 - 2xyt + 3$	29.	$7y^2 \sin(t) / x$
15.	$\left(1 + e^{\frac{xt}{y}}\right) + e^{\frac{xt}{y}} \left(1 - \frac{xt}{y}\right)$	30.	$-60x^{1/yt} + \sin(xt)$

Переменные x, y задайте произвольно.

Лабораторная работа

Matlab: Решение уравнений и их систем

Цель работы: получение навыков решения уравнений и их систем в пакете Matlab.

Теоретические сведения

Аналитическое решение уравнений и их систем.

Функция решение алгебраических уравнений – *solve*

Для решения систем алгебраических уравнений и одиночных уравнений служит функция *solve*:

solve(*expr1*, *expr2*,... *exprN*, *var1*, *var2*,... *varN*) — возвращает значения переменных *var1*, при которых соблюдаются равенства, заданные выражениями *exprI*. Если в выражениях не используются знаки равенства, то полагается $\text{exprI} = 0$;

Примеры решения уравнений:

```

» syms x y;
» solve(x^3 -1, x)
ans =
[      1]
[-1/2+1/2*i*3^(1/2)]
[-1/2-1/2*i*3^(1/2)]
» syms a b c
» solve(a*x^2+b*x+c)
ans =
[ 1/2/a*(-b+(b^2-4*a*c)^(1/2))]
[ 1/2/a*(-b-(b^2-4*a*c)^(1/2))]
» S = solve('x+y=3', 'x*y^2=4', x, y)
S =
    x : [ 3x1 sym ]
    y : [ 3x1 sym ]
» S.x
ans =
[4]
[1]
[1]

```

Информационные технологии в менеджменте

```

» S.y
ans =
[-1]
[2]
[2]
» solve('sin(x)=0.5', x)
ans =
0.52359877559829887307710723054658

```

Численное решение уравнений и их систем. Задача нахождения корней уравнений встречается в различных областях научно-технических исследований. Проблема формулируется следующим образом. Пусть задана непрерывная функция $f(x)$ и требуется найти корень уравнения:

$$f(x) = 0.$$

Будем предполагать, что имеется интервал изменения x $[a;b]$, на котором необходимо исследовать функцию $f(x)$ и найти значение x_0 , при котором $f(x_0)$ равно или весьма мало отличается от нуля.

Данная задача в системе Matlab может быть решена следующим образом. Вначале необходимо построить график функции $f(x)$ на заданном интервале и убедиться в существовании корня или нескольких корней. Затем применить функции поиска корней.

Если существует один корень и график $f(x)$ пересекает ось Ox , то можно применить функцию:

- `fzero(fun, x)` возвращает уточненное значение x , при котором достигается нуль функции `fun`, представленной строкой, при начальном значении аргумента x . Возвращенное значение близко к точке, где функция меняет знак, или равно NaN, если такая точка не найдена;
- `fzero(fun, [x1 x2])` возвращает значение x , при котором $fun(x) = 0$ с заданием интервала поиска с помощью вектора $x = [x1 \ x2]$, такого что знак $fun(x(1))$ отличается от знака $fun(x(2))$. Если это не так, выдается сообщение об ошибке. Вызов функции `fzero` с интервалом гарантирует, что `fzero` возвратит значение, близкое к точке, где `fun` изменяет знак;
- `fzero(fun, x, tol)` возвращает результат с заданной погрешностью `tol`;
- `fzero(fun, x, tol, trace)` выдает на экран информацию о каждой итерации;

Для функции `fzero` нуль рассматривается как точка, где график функции `fun` пересекает ось x , а не касается ее. В зависимости от формы задания функции `fzero` реализуются следующие хорошо известные численные методы поиска нуля функции: деления отрезка пополам, секущей и обратной квадратичной интерполяции.

Если $f(x)$ имеет больше одного корня и может касаться и пересекать ось Ox , то следует применить более мощную функцию **`fsolve`** из пакета **Optimization Toolbox**, которая решает задачу методом наименьших квадратов:

`fsolve(fun, x0:h:x1),`

поиск корней выполняется на интервале от x_0 до x_1 с шагом h .

Пример 7. Найти корень нелинейного уравнения $10^x + 2x - 100 = 0$ на интервале $[1.0; 2.0]$.

Протокол программы

```
>> % Строим график заданной функции
>> x = 1.0 : 0.001 : 2.0; y = 10.0.^x + 2.0*x - 100.0;
>> plot (x, y) ; grid on
```

Появляется окно с графиком функции $10^x + 2x - 100$ (см. рис. 2.4), из которого следует, что корень функции на заданном интервале существует. Для точного определения корня применяем *fzero* и *fsolve*.

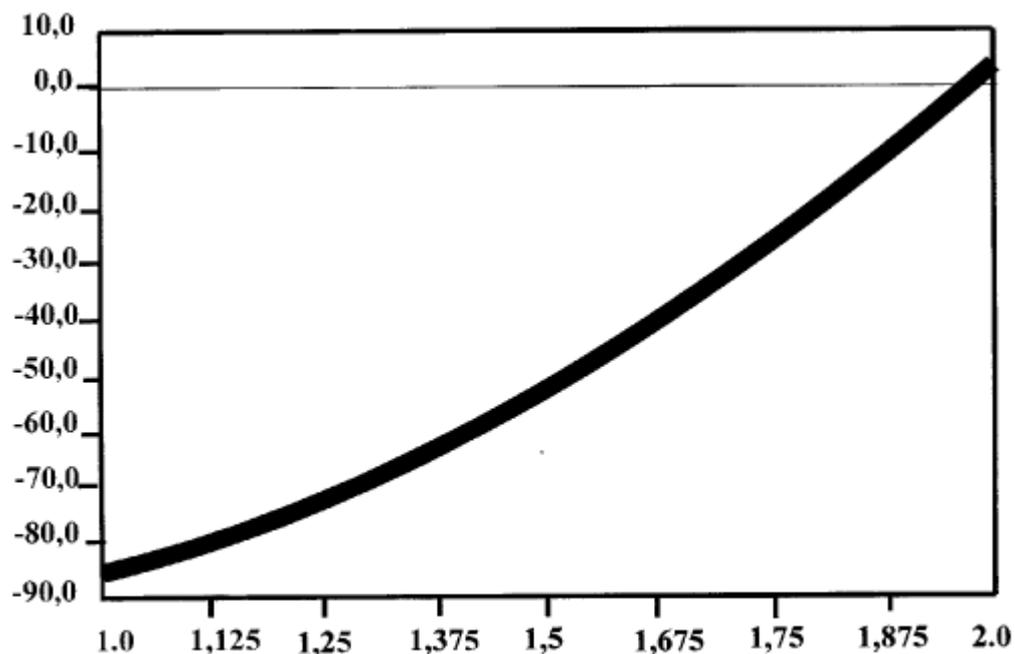


Рис. 2.4

```
>> X1 = fzero (' (10.0.^x + 2.0*x - 100.0) ', [1.0 2.0])
```

Результат решения

```
X1 =
    1.9824
```

```
>> X2 = fsolve (' (10.0.^x + 2.0*x - 100.0) ', 1.0 : 2.0)
```

Результат решения

```
X2 =
    1.9824  1.9824
```

Информационные технологии в менеджменте

Функция `fsolve` может использоваться и для решения систем нелинейных уравнений. В этом случае речь идет о нахождении значений двух и более переменных, удовлетворяющих решению заданной системы нелинейных уравнений.

Пример. Численно решить систему уравнений:

$$\begin{cases} 2x - y - 2z = -1 \\ 2y + z = 3 \\ -2x - 2y + 2z = -2 \end{cases}$$

В этом случае необходимо создать файл-функцию для расчета значений уравнений от x :

```
function F = fun_07(x)
F = [2*x(1)-x(2)-2*x(3)+1;
     2*x(2)+x(3)-3;
     -2*x(1)-2*x(2)+2*x(3)+2];
```

При этом ввели вектор x размерности 3×1 , такой что $x = (x \ y \ z)$.

В основной программе делаем вызов функции `fun_07` в функции `fsolve`:

```
x0=[10;30;-1];
X = fsolve('fun_07',x0)
```

Решение задачи определения корней полинома осуществляется при помощи встроенной функции `roots(v)`, где v — вектор-столбец коэффициентов полинома, первым элементом которого является коэффициент при x^n , вторым — коэффициент при x^{n-1} и т. д.

Задания

Все задания рекомендуется выполнять в m-файлах.

Задание 1. Найдите аналитическое решение уравнения:

№ п/п	Уравнение $f(x) = 0$	Отрезок $[a; b]$
1	$\arctg x - 1 = 0$	$[1.0; \sqrt{3}]$
2	$e^{x-2} - \ln(x+2) = 0$	$[2.0; 3.0]$
3	$x^3 - 9x^2 + 5x - 6 = 0$	$[8.0; 9.0]$
4	$e^x - \frac{1}{x} - 1 = 0$	$[0.5; 1.0]$

Информационные технологии в менеджменте

5	$\operatorname{arctg} 2x - \frac{1}{1+x} = 0$	[0.0; 1.0]
6	$e^x - \ln x - 20 = 0$	[3.0; 3.2]
7	$\sqrt{x} - \operatorname{tg} x(1-x) = 0$	[0.0; 1.0]
8	$\sin \sqrt{x} - \cos \sqrt{x} + 2\sqrt{x} = 0$	[0.0; 0.2]
9	$x^4 + 2x^3 - x - 1 = 0$	[0.8; 1.0]
10	$x^3 - e^{4x} - 5.5 = 0$	[2.6; 3.0]
11	$x^6 - 3x^2 + x - 1 = 0$	[1.0; 1.5]
12	$\sqrt[3]{5-x} - x = 0$	[1.0; 2.0]
13	$x^2 - \ln x = 0$	[0.0; 1.0]
14	$x^2 - \cos x = 0$	[0.0; 1.0]
15	$\ln x - \operatorname{arctg} x = 0$	[3.0; 4.0]
16	$x^2 \operatorname{arctg} x - 1 = 0$	[1.0; 1.2]
17	$x^2 + \ln x - 4 = 0$	[1.0; 2.0]
18	$x - \operatorname{arctg} \sqrt[3]{x} = 0$	[0.0; 1.0]
19	$x^2 - e^x - 2 = 0$	[-0.2; -0.1]
20	$x^2 - \ln(x+1) = 0$	[0.1; 0.9]
21	$x^5 - x - 2 = 0$	[1.0; 1.4]
22	$x - 2 - \sqrt[4]{x} = 0$	[3.0; 4.0]
23	$x - \operatorname{tg} x = 0$	[0.0; 1.5]
24	$x + \ln x - 0.5 = 0$	[0.0; 1.0]
25	$\ln x + \sqrt{x} = 0$	[0.1; 1.0]
26	$\sqrt{x} - \cos \sqrt{x} = 0$	[0.4; 0.6]
27	$x^2 - \ln(1+x^2) - 9.75 = 0$	[3.0; 4.0]
28	$x + \sqrt[3]{x} - 6.09 = 0$	[4.0; 5.0]
29	$x^3 - \sqrt{x} - 9.5 = 0$	[2.0; 3.0]
30	$\arccos 2x - x^2 - 0.35 = 0$	[0.0; 0.48]

Задание 2. Найдите численное решение уравнения с помощью функций `fzero` и `fsolve`.

Задание 3. Аналитически решите систему уравнений:

№	Система уравнений	№	Система уравнений
1	$\begin{cases} \sin(x+1) - y = 1,2; \\ 2x + \cos y = 2. \end{cases} \begin{cases} \operatorname{tg}(xy+0,4) = x^2; \\ 0,6x^2 + 2y^2 = 1, x > 0, y > 0. \end{cases}$	16	$\begin{cases} \cos(x-1) + y = 0,5; \\ x - \cos y = 3. \end{cases} \begin{cases} \sin(x+y) - 1,6x = 0; \\ x^2 + y^2 = 1, x > 0, y > 0. \end{cases}$
2	$\begin{cases} \cos(x-1) + y = 0,5; \\ x - \cos y = 3. \end{cases} \begin{cases} \sin(x+y) - 1,6x = 0; \\ x^2 + y^2 = 1, x > 0, y > 0. \end{cases}$	17	$\begin{cases} \sin x + 2y = 2; \\ \cos(y-1) + x = 0,7; \end{cases} \begin{cases} \operatorname{tg}(xy+0,1) = x^2; \\ x^2 + 2y^2 = 1. \end{cases}$
3	$\begin{cases} \sin x + 2y = 2; \\ \cos(y-1) + x = 0,7; \end{cases} \begin{cases} \operatorname{tg}(xy+0,1) = x^2; \\ x^2 + 2y^2 = 1. \end{cases}$	18	$\begin{cases} \cos x + y = 1,5; \\ 2x - \sin(y-0,5) = 1. \end{cases} \begin{cases} \sin(x+y) - 1,2x = 0,2; \\ x^2 + y^2 = 1. \end{cases}$
4	$\begin{cases} \sin(x+0,5) - y = 1; \\ \cos(y-2) + x = 0. \end{cases} \begin{cases} \operatorname{tg}(xy+0,3) = x^2; \\ 0,9x^2 + 2y^2 = 1. \end{cases}$	19	$\begin{cases} \sin(y+1) - x = 1,2; \\ 2y + \cos x = 2. \end{cases} \begin{cases} \operatorname{tg}(xy+0,2) = x^2; \\ 0,6x^2 + 2y^2 = 1. \end{cases}$
5	$\begin{cases} \sin(x+2) - y = 1,5; \\ x + \cos(y-2) = 0,5. \end{cases} \begin{cases} \sin(x+y) - 1,2x = 0,1; \\ x^2 + y^2 = 1. \end{cases}$	20-	$\begin{cases} 2y - \cos(x+1) = 0; \\ x + \sin y + 0,4. \end{cases} \begin{cases} \sin(x+y) - 1,5x = 0,1; \\ x^2 + y^2 = 1. \end{cases}$
6	$\begin{cases} \cos(x+0,5) - y = 2; \\ \sin y - 2x = 1. \end{cases} \begin{cases} \operatorname{tg}xy = x^2; \\ 0,7x^2 + 2y^2 = 1. \end{cases}$	21	$\begin{cases} \sin(x-1) = 1,3 - y; \\ x - \sin(y+1) = 0. \end{cases} \begin{cases} \operatorname{tg}xy = x^2; \\ 0,8x^2 + 2y^2 = 1. \end{cases}$
7	$\begin{cases} \cos(x+0,5) - y = 1; \\ \sin y - 2x = 1,6. \end{cases} \begin{cases} \sin(x+y) - 1,3x = 0; \\ x^2 + y^2 = 1. \end{cases}$	22	$\begin{cases} \cos(y-1) + x = 0,5; \\ y - \cos x = 3. \end{cases} \begin{cases} \sin(x+y) + 1,5x - 0,1; \\ x^2 + y^2 = 1. \end{cases}$
8	$\begin{cases} \sin y + 2x = 2; \\ \cos(x-1) + y = 0,7. \end{cases} \begin{cases} \operatorname{tg}(xy+0,4) = x^2; \\ 0,8x^2 + 2y^2 = 1. \end{cases}$	23	$\begin{cases} \cos y + x + 1,5; \\ 2y - \sin(x-0,5) = 1. \end{cases} \begin{cases} \sin(x+y) = 1,2x - 0,1; \\ x^2 + y^2 = 1. \end{cases}$
9	$\begin{cases} \sin(y+0,5) - x = 1; \\ \cos(x-2) + y = 0. \end{cases} \begin{cases} \operatorname{tg}(xy+0,1) = x^2; \\ 0,9x^2 + 2y^2 = 1. \end{cases}$	24	$\begin{cases} \sin(y-1) + x = 1,3; \\ y - \sin(x+1) = 0,8. \end{cases} \begin{cases} \operatorname{tg}(xy+0,1) = x^2; \\ 0,5x^2 + 2y^2 = 1. \end{cases}$
10	$\begin{cases} \cos(y+0,5) + x = 0,8; \\ \sin x - 2y = 1,6. \end{cases} \begin{cases} \sin(x+y) - 1,4x = 0; \\ x^2 + y^2 = 1. \end{cases}$	25	$\begin{cases} 2x - \cos(y+1) = 0; \\ y + \sin x + 0,4. \end{cases} \begin{cases} \sin(x+y) = 1,1x - 0,1; \\ x^2 + y^2 = 1. \end{cases}$
11	$\begin{cases} \cos(y+0,5) - x = 2; \\ \sin x - 2y = 1. \end{cases} \begin{cases} \operatorname{tg}(x-y) - xy = 0; \\ x^2 + 2y^2 = 1. \end{cases}$	26	$\begin{cases} \sin(y+2) - x = 1,5; \\ y + \cos(x-2) = 0,5. \end{cases} \begin{cases} \sin(x-y) - xy = -1; \\ x^2 - y^2 = \frac{3}{4}. \end{cases}$
12	$\begin{cases} \sin(x+1) - y = 1; \\ 2x + \cos y = 2. \end{cases} \begin{cases} \operatorname{tg}(xy+0,2) = x^2; \\ x^2 + 2y^2 = 1. \end{cases}$	27	$\begin{cases} \cos(x-1) + y = 0,8; \\ x - \cos y + 2. \end{cases} \begin{cases} \sin(x+y) - 1,5x = 0; \\ x^2 + y^2 = 1. \end{cases}$
13	$\begin{cases} \sin x + 2y + 1,6; \\ \cos(y-1) + x = 1. \end{cases} \begin{cases} \operatorname{tg}xy = x^2; \\ 0,5x^2 + 2y^2 = 1. \end{cases}$	28	$\begin{cases} \cos x + y = 1,2; \\ 2x - \sin(y-0,5) = 2. \end{cases} \begin{cases} \sin(x+y) = 1,2x - 0,2; \\ x^2 + y^2 = 1. \end{cases}$
14	$\begin{cases} \sin(x+0,5) - y = 1,2; \\ \cos(y-2) + x = 0. \end{cases} \begin{cases} \operatorname{tg}(xy+0,1) = x^2; \\ 0,7x^2 + 2y^2 = 1. \end{cases}$	29	$\begin{cases} \cos(x+0,5) + y = 1; \\ \sin y - 2x = 2. \end{cases} \begin{cases} \sin(x+y) - 1,5x + 0,2; \\ x^2 + y^2 = 1. \end{cases}$
15	$\begin{cases} \sin(x-1) + y = 1,5; \\ x - \sin(y+1) = 1. \end{cases} \begin{cases} \operatorname{tg}xy = x^2; \\ 0,6x^2 + 2y^2 = 1. \end{cases}$	30	$\begin{cases} \sin(y+1) - x + 1; \\ 2y + \cos x = 2. \end{cases} \begin{cases} \sin(x+y) - 1,2x = 0; \\ x^2 + y^2 = 1. \end{cases}$

Задание 4. С помощью функции **fsolve** найдите численное решение системы уравнений.

Задание 5. Найдите корни полинома:

№	уравнение	№	уравнение
1	$x^4 - x - 1 = 0; x^3 + x - 3 = 0.$	16	$x^4 - x - 1 = 0; x^3 - 3x^2 + 9x - 10 = 0.$
2	$2x^4 - 9x^2 - 60x + 1 = 0; x^3 - 2x + 2 = 0.$	17	$x^4 + 4x^3 - 8x^2 - 17 = 0; x^3 + 3x - 1 = 0.$
3	$3x^4 + 4x^3 - 12x^2 - 5 = 0; x^3 + 2x^2 + 2 = 0.$	18	$3x^4 + 8x^3 + 6x^2 - 10 = 0; x^3 + 0,4x^2 + 0,6x - 1,6 = 0.$
4	$2x^4 - x^2 - 10 = 0; x^3 - 0,2x^2 + 0,4x - 1,4 = 0.$	19	$x^4 - 18x^2 + 6 = 0; x^3 - 0,1x^2 + 0,4x + 2 = 0.$
5	$3x^4 + 8x^3 + 6x^2 - 10 = 0; x^3 + 3x^2 + 12x + 3 = 0.$	20	$2x^4 - x^2 - 10 = 0; x^3 - 0,2x^2 + 0,5x - 1 = 0.$
6	$x^4 - 18x^2 + 6 = 0; x^3 - 0,1x^2 + 0,4x + 1,2 = 0.$	21	$x^4 - x^3 - 2x^2 + 3x - 3 = 0; x^3 - 3x^2 + 6x - 5 = 0.$
7	$x^4 + 4x^3 - 8x^2 - 17 = 0; x^3 - 0,2x^2 + 0,5x - 1,4 = 0.$	22	$3x^4 + 4x^3 - 12x^2 + 1 = 0; x^3 + 0,2x^2 + 0,5x + 0,8 = 0.$
8	$x^4 - x^3 - 2x^2 + 3x - 3 = 0; x^3 + 4x - 6 = 0.$	23	$3x^4 - 8x^3 - 18x^2 + 2 = 0; x^3 + 0,1x^2 + 0,4x - 1,2 = 0.$
9	$3x^4 + 4x^3 - 12x^2 + 1 = 0; x^3 + 3x^2 + 6x - 1 = 0.$	24	$3x^4 + 4x^3 - 12x^2 - 5 = 0; x^3 - 0,1x^2 + 0,4x - 1,5 = 0.$
10	$3x^4 - 8x^3 - 18x^2 + 2 = 0; x^3 - 3x^2 + 6x - 2 = 0.$	25	$2x^3 - 9x^2 - 60x + 1 = 0; x^3 + 2x + 4 = 0.$
11	$2x^4 - 8x^3 + 8x^2 - 1 = 0; x^3 - 3x^2 + 12x - 12 = 0.$	26	$x^4 - x - 1 = 0; x^3 - 0,2x^2 + 0,3x - 1,2 = 0.$
12	$2x^4 + 8x^3 + 8x^2 - 1 = 0; x^3 - 3x^2 + 12x - 9 = 0.$	27	$2x^4 - x^2 - 10 = 0; x^3 + 0,2x^2 + 0,5x - 2 = 0.$
13	$x^4 - 4x^3 - 8x^2 + 1 = 0; x^3 + 3x + 1 = 0.$	28	$3x^4 + 8x^3 + 10 = 0; x^3 + 0,2x^2 + 0,5x - 1,2 = 0.$
14	$3x^4 + 4x^3 - 12x^2 - 5 = 0; x^3 - 3x^2 + 9x + 2 = 0.$	29	$x^3 + 0,1x^2 + 0,4x - 1,2 = 0. x^4 - 18x^2 + 6 = 0;$
15	$2x^3 - 9x^2 - 60x + 1 = 0; x^3 - 3x^2 + 6x + 3 = 0.$	30	$3x^4 + 4x^3 - 12x^2 + 1 = 0; x^3 - 0,1x^2 + 0,3x - 0,6 = 0.$

Лабораторная работа

Matlab: Решение дифференциальных уравнений и их систем

Цель работы: получение навыков решения дифференциальных уравнений и их систем в пакете Matlab.

Теоретические сведения

Анализ поведения многих систем и устройств в динамике, а также решение многих задач в теории колебаний и в поведении упругих оболочек обычно базируется на решении систем *обыкновенных дифференциальных уравнений* (ОДУ), или, в оригинале, *ordinary differential equations* (ODEs). Их, как правило, представляют в виде системы из дифференциальных уравнений первого порядка в форме Коши:

$$\frac{dy}{dt} = y' = f(y, t) \quad (1)$$

с граничными условиями $y(t_0, t_{\text{end}}, p) = b$, где t_{end}, t_0 — начальные и конечные точки интервалов. Параметр t (независимая переменная) необязательно означает время, хотя чаще всего решение дифференциальных уравнений ищется во временной области.

Система дифференциальных уравнений в форме Коши записывается аналогично (1), но под y в этом случае подразумевается вектор-столбец зависимых переменных. Вектор p задает начальные условия.

Для решения дифференциальных уравнений второго и высшего порядка их нужно свести к системе дифференциальных уравнений первого порядка.

Информационные технологии в менеджменте

Возможны дифференциальные уравнения, не разрешенные относительно производной:

$$F(t, y, dy/dt) = 0. \quad (2)$$

Уравнения (2) аналитически к форме (1) обычно привести не удастся.

Однако численное решение особых трудностей не вызывает — достаточно для определения $f(y, t)$ решить (2) численно относительно производной при заданных y и t .

Наряду с ОДУ MATLAB может оперировать с *дифференциальными алгебраическими уравнениями* (ДАУ, или *differential algebraic equations* — DAEs). ОДУ и ДАУ являются основой математического моделирования динамических нелинейных (и линейных) систем. Автоматическое их составление и решение реализованы в специальном расширении **Simulink**.

Аналитическое решение дифференциальных уравнений и их систем

Для решения дифференциальных уравнений в форме Коши MATLAB имеет следующую функцию:

- `dsolve('eqn1', 'eqn2', ...)` — возвращает аналитическое решение системы дифференциальных уравнений с начальными условиями. Они задаются равенствами `eqn1` (вначале задаются уравнения, затем начальные условия).

По умолчанию независимой переменной считается переменная 't', обычно обозначающая время. Можно использовать и другую переменную, добавив ее в конец списка параметров функции `dsolve`. Символ D обозначает производную по независимой переменной, то есть d/dt , при этом D2 означает d^2/dt^2 и т. д. Имя независимой переменной не должно начинаться с буквы D.

Начальные условия задаются в виде равенств ' $y(a)=b$ ' или ' $Dy(a)=b$ ', где y — независимая переменная, a и b — константы. Если число начальных условий меньше, чем число дифференциальных уравнений, то в решении будут присутствовать произвольные постоянные C1, C2 и т. д.

Примеры применения функции `dsolve`:

```
» dsolve('D2x = -2*x')
```

```
ans =
```

```
 C1*cos(2^(1/2)*t) + C2*sin(2^(1/2)*t)
```

```
» dsolve('D2y = -2*x + y', 'y(0) = 1', 'x')
```

```
ans =
```

```
 (2*x*exp(x) + (- C2 + 1)*exp(x)^2 + C2 / exp(x))
```

Пример 1. Аналитически решите дифференциальное уравнение первого порядка $y'(t) = y \cos t + \sin 2t$:

```
dsolve('Dy=y*cos(t)+sin(2*t)')
```

```
ans =
```

```
-2*sin(t)-2+exp(sin(t))*C1
```

Пример 2. Аналитически решите дифференциальное уравнение второго порядка $mx''+Hx'+kx=-\sqrt{t}$:

dsolve('m*D2x+H*Dx+k*x=-sqrt(t)')

ans =

$$-2*m*(2*(-(2*H-2*(H^2-4*m*k)^(1/2))/m)^(1/2)*t^(1/2)*(-2*H+2*(H^2-4*m*k)^(1/2))/m)^(1/2)*(H^2-4*m*k)^(1/2)+exp(-1/2*(H+(H^2-4*m*k)^(1/2))*t/m)*(-(2*H-2*(H^2-4*m*k)^(1/2))/m)^(1/2)*pi^(1/2)*erf(1/2*(-(2*H+2*(H^2-4*m*k)^(1/2))/m)^(1/2)*t^(1/2))*H-exp(-1/2*(H+(H^2-4*m*k)^(1/2))*t/m)*(-(2*H-2*(H^2-4*m*k)^(1/2))/m)^(1/2)*pi^(1/2)*erf(1/2*(-(2*H+2*(H^2-4*m*k)^(1/2))/m)^(1/2)*t^(1/2))*(H^2-4*m*k)^(1/2)-exp(-1/2*(H-(H^2-4*m*k)^(1/2))*t/m)*(-(2*H+2*(H^2-4*m*k)^(1/2))/m)^(1/2)*pi^(1/2)*erf(1/2*(-(2*H-2*(H^2-4*m*k)^(1/2))/m)^(1/2)*t^(1/2))*H-exp(-1/2*(H-(H^2-4*m*k)^(1/2))*t/m)*(-(2*H+2*(H^2-4*m*k)^(1/2))/m)^(1/2)*pi^(1/2)*erf(1/2*(-(2*H-2*(H^2-4*m*k)^(1/2))/m)^(1/2)*t^(1/2))*(H^2-4*m*k)^(1/2))/(H^2-4*m*k)^(1/2)/(H+(H^2-4*m*k)^(1/2))/(-(2*H+2*(H^2-4*m*k)^(1/2))/m)^(1/2)/(H-(H^2-4*m*k)^(1/2))/(-(2*H-2*(H^2-4*m*k)^(1/2))/m)^(1/2)+C1*exp(-1/2*(H+(H^2-4*m*k)^(1/2))*t/m)+C2*exp(-1/2*(H-(H^2-4*m*k)^(1/2))*t/m)$$

Пример 3. Аналитически решите систему дифференциальных уравнений

$$\begin{cases} y_0' = ay_1 \\ y_1' = y_0 + y_1 \end{cases} :$$

F=dsolve('Dy0=a*y1','Dy1=y0+y1')

F =

y0: [1x1 sym]

y1: [1x1 sym]

>> F.y0

ans =

$C1*exp(1/2*t+1/2*t*(1+4*a)^(1/2))+C2*exp(1/2*t-1/2*t*(1+4*a)^(1/2))$

>> F.y1

ans =

$1/2*(C1*exp(1/2*t+1/2*t*(1+4*a)^(1/2))+C1*(1+4*a)^(1/2)*exp(1/2*t+1/2*t*(1+4*a)^(1/2))+C2*exp(1/2*t-1/2*t*(1+4*a)^(1/2))-C2*(1+4*a)^(1/2)*exp(1/2*t-1/2*t*(1+4*a)^(1/2)))/a$

Численное решение дифференциальных уравнений и их систем

Решение дифференциальных уравнений осуществляют функции **ode23** и **ode45**. Они могут применяться как для решения простых дифференциальных уравнений, так и для моделирования сложных динамических систем.

Известно, что система дифференциальных уравнений может быть представлена в форме Коши:

$$\frac{dy}{dt} = f(y,t)$$

Информационные технологии в менеджменте

где y — вектор переменных состояния системы; t — аргумент (обычно время); f — нелинейная вектор-функция от переменных состояния y и аргумента t .

Обращение к процедурам численного интегрирования дифференциальных уравнений имеет вид:

[t,y]=ode23('имя функции',tspan, y0, options)

[t,y]=ode45('имя функции',tspan, y0, options)

где *имя функции* — строка символов, являющаяся именем m-файла, в котором вычисляется вектор-функция $f(y,t)$, т. е. в котором находятся правые части системы дифференциальных уравнений; y_0 — вектор начальных значений переменных состояния; t — массив значений аргумента, соответствующих шагам интегрирования; y — матрица проинтегрированных значений фазовых переменных, в которой каждый столбец соответствует одной из переменных состояния, а строка содержит значения переменных состояния, соответствующих определенному шагу интегрирования; $tspan$ — вектор-строка $[t_0, t_{last}]$, содержащая два значения: t_0 — начальное значение аргумента t ; t_{last} — конечное значение аргумента t ; $options$ — строка параметров, определяющих значения допустимой относительной и абсолютной погрешности интегрирования.

Для получения решений дифференциальных уравнений в определенные моменты времени можно указать явным образом $tspan = [t_0, t_1, t_2, \dots, t_n]$.

Параметр $options$ можно не указывать. Тогда по умолчанию допустимая относительная погрешность интегрирования принимается равной $1.0e-3$, а абсолютная (по каждой из переменных состояния) — $1.0e-6$.

Функция **ode23** осуществляет интегрирование численным методом Рунге-Кутта 2-го порядка, а с помощью метода 3-го порядка контролирует относительные и абсолютные ошибки интегрирования на каждом шаге и изменяет величину шага интегрирования так, чтобы обеспечить заданные пределы ошибок интегрирования.

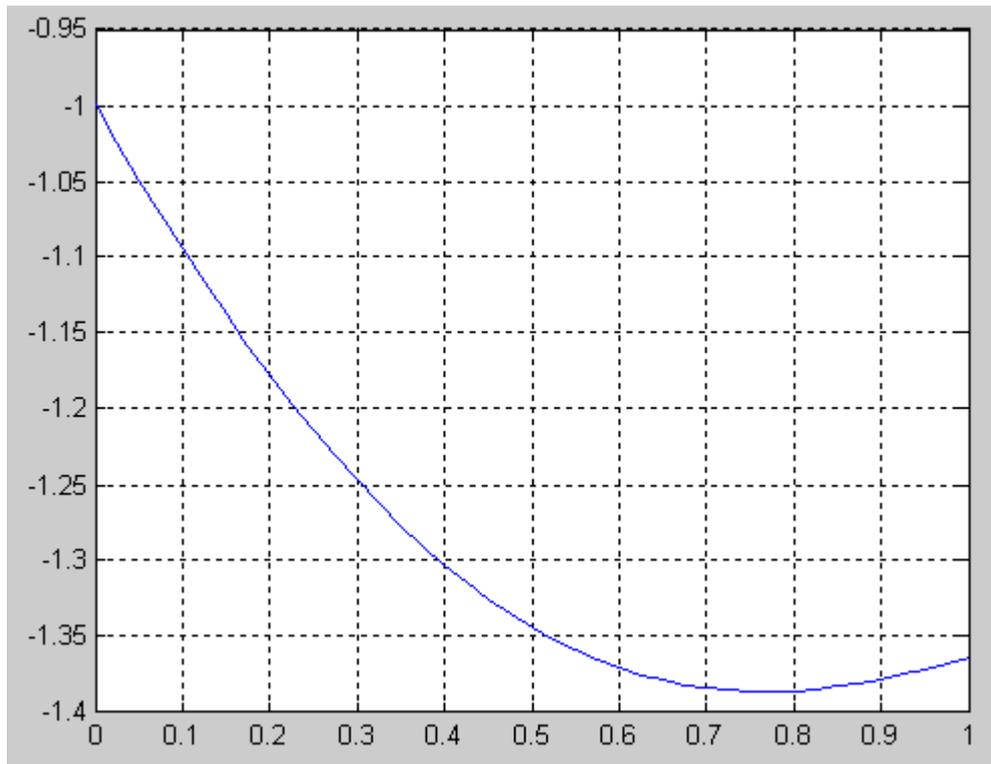
Для **ode45** основным методом является метод Рунге-Кутта 4-го порядка, а величина шага контролируется методом 5-го порядка.

Пример 4. Численно решите дифференциальное уравнение первого порядка $y'(t) = y \cos t + \sin 2t$:

```
tspan=[0 1];
y0=-1;
[T,Y] = ode45('fun_08_1',tspan,y0);
figure
plot(T,Y)
grid on
```

Содержание файла-функции **fun_08_1**:

```
function dy = fun_08_1(t,y)
dy=y*cos(t)+sin(2*t);
```



Пример 5. Численно решите дифференциальное уравнение второго порядка $mx'' + Hx' + kx = -\sqrt{t}$.

При этом необходимо преобразовать дифференциальное уравнение второго порядка в систему двух дифференциальных уравнений первого порядка (переход к переменным состояния). Для этого введем так называемые *переменные состояния* x_0, x_1, x_2 :

$$x_0 = x, \quad x_1 = x', \quad x_2 = x''.$$

Тогда можно записать:

$$\begin{cases} x_0' = x' = x_1 \\ x_1' = x'' = x_2 = \frac{-Hx_1 - kx_0 - \sqrt{t}}{m} \end{cases}$$

Однако, поскольку в Matlab нумерация индексов массива начинается с 1 (а не с нуля), то систему уравнений необходимо переписать в следующем виде:

$$\begin{cases} x_1' = x' = x_2 \\ x_2' = x'' = x_3 = \frac{-Hx_2 - kx_1 - \sqrt{t}}{m} \end{cases}$$

tspan=[0 15];

x0=[0 -10];

[T,X] = ode45('fun_08_2',tspan,x0);

Для наглядного представления результата строится график зависимостей столбцов возвращенной матрицы решений X , от вектора T :

figure

plot(T,X(:,1),'b',T,X(:,2),'g')

grid on

Содержание файла-функции **fun_08_2**:

function dx = fun_08_2(t,x)

m=0.5;

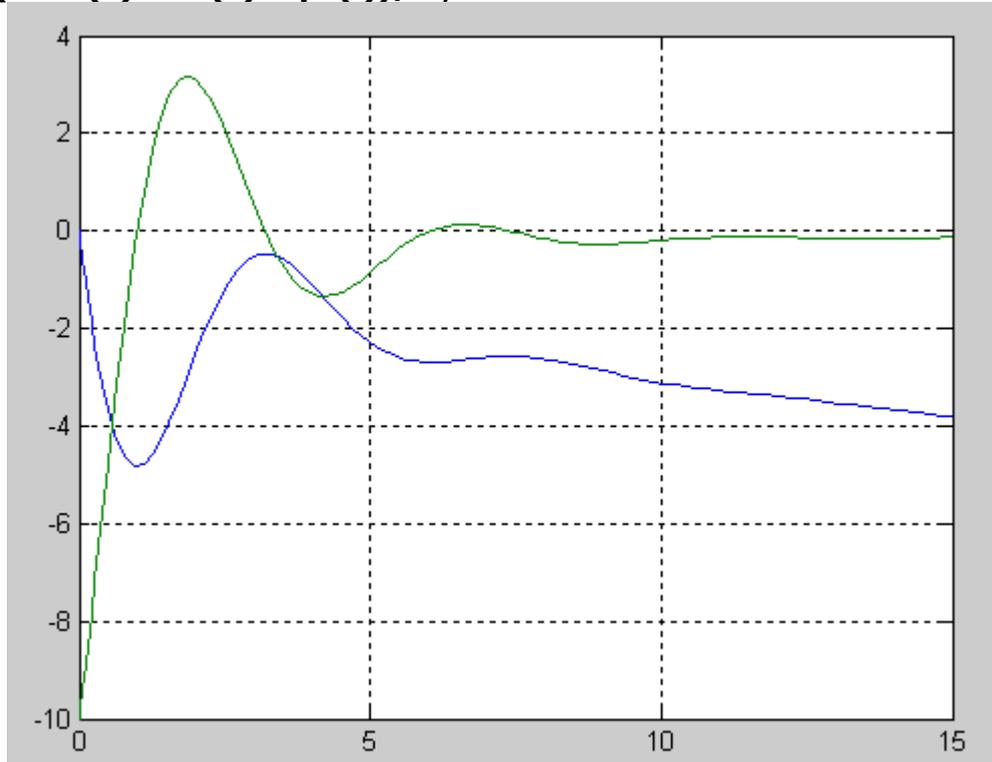
H=0.5;

k=1;

dx = zeros(2,1); % создает нулевой вектор-столбец, где будут храниться полученные на данном этапе значения переменных состояния

dx(1)=x(2);

dx(2)=(-H*x(2)-k*x(1)-sqrt(t))/m;



Пример 6. Решить систему дифференциальных уравнений:

$$\begin{cases} y_1' = y_2 y_3 & y_1(0) = 0 \\ y_2' = -y_1 y_3 & \text{с начальными условиями} \quad y_2(0) = 1 \\ y_3' = -0,51 y_1 y_2 & y_3(0) = 1 \end{cases}$$

Для моделирования данной системы, создаем функцию **rigid**, следующего содержания:

function dy=rigid(t,y)

dy=zeros(3,1);

dy(1)=y(2)*y(3);

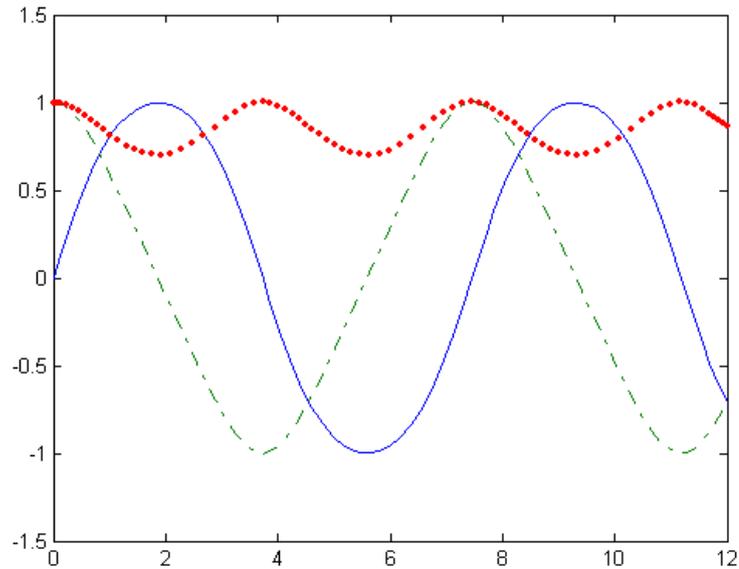
dy(2)=-y(1)*y(3);

dy(3)=-0.51*y(1)*y(2);

Решение находится на интервале [0; 12] с начальными условиями [0 1 1]:

[T,Y] = ode45('rigid',[0 12],[0 1 1]);

plot(T,Y(:,1),'-',T,Y(:,2),'-.',T,Y(:,3),'-.')



Задания

Все задания рекомендуется выполнять в m-файлах.

Задание 1. Аналитически и численно решите дифференциальное уравнение первого порядка $y'(t) = f(t, y(t))$. Постройте график решения.

№	$f(t, y(t))$	t_0	t_1	y_0
31.	$y/t + t^2$	1	2	0
32.	$yctgt + 2t \sin t$	$\frac{\pi}{2}$	$\frac{\pi}{2} + 1$	0
33.	$-y \cos t + \frac{\sin(2t)}{2}$	0	1	0
34.	$-y \operatorname{tg} t + \cos^2 t$	$\frac{\pi}{4}$	$\frac{\pi}{4} + 1$	0,5
35.	$\frac{y}{t+2} + t^2 + 2t$	-1	0	1,5
36.	$\frac{y}{t+1} + e^t(t+1)$	0	1	1
37.	$y/t + t \sin t$	$\frac{\pi}{2}$	$\frac{\pi}{2} + 1$	1
38.	$-y/t + \sin t$	π	$\pi + 1$	$\frac{1}{\pi}$
39.	$-\frac{y}{2t} + t^2$	1	2	1
40.	$-\frac{2t}{1+t^2} y + \frac{2t^2}{1+t^2}$	0	1	$\frac{2}{3}$
41.	$\frac{2t-5}{t^2} y + 5$	2	3	4

Информационные технологии в менеджменте

42.	$-y/t + \frac{t+1}{t}e^t$	1	2	e
43.	$y/t - 2\ln t/t$	1	2	1
44.	$y/t - 12/t^3$	1	2	4
45.	$-2y/t + t^3$	1	2	$-\frac{5}{6}$
46.	$-y/t + 3t$	1	2	1
47.	$\frac{2ty}{1+t^2} + 1 + t^2$	1	2	3
48.	$\frac{2t-1}{t^2}y + 1$	1	2	1
49.	$-\frac{3y}{t} + \frac{2}{t^3}$	1	2	1
50.	$-2ty - 2t^3$	1	2	e^{-1}
51.	$y/t - 2/t^2$	1	1	1
52.	$-ty - t^3$	0	1	3
53.	$\frac{2}{t+1}y + e^t(t+1)^2$	0	1	1
54.	$-2ty + te^{-t^2} \sin t$	0	1	1
55.	$\frac{2y}{t+1} + (t+1)^3$	0	1	0,5
56.	$y \cos t - \sin 2t$	0	1	3
57.	$4ty - 4t^3$	0	1	-0,5
58.	$y/t - \ln t/t$	1	2	1
59.	$3t^2y + t^2(1+t^3)/3$	0	1	0
60.	$y \cos t + \sin 2t$	0	1	-1

Задание 2. Аналитически и численно решите дифференциальное уравнение третьего порядка на интервале $t = [A, B]$:

$$a_0y''' + a_1y'' + a_2y' + a_3y = f(t)$$

$$y(A) = b_1, y'(A) = b_2, y''(A) = b_3$$

Постройте графики решений.

№	A	B	b_1	b_2	b_3	a_0	a_1	a_2	a_3	$f(t)$
1, 11, 21	0	1,5	1	2,5	6	1	-2	0,25	45,75	$e^{-2t} + 3t + 1$
2, 12, 22	0	1,5	1	2,0	4	1	-1,8	0,36	44,28	$e^{-2t} - 1,5t + 1$
3, 13, 23	0	1,5	1	2,5	6	1	-1,4	0,64	41,52	$\cos(2t) + 3t + 1$
4, 14, 24	0	2,0	1	1,5	2	1	-1,4	1,88	45,24	$\sin(2t) + 2t - 1$

Информационные технологии в менеджменте

5, 15, 25	0	1,5	1	3,0	10	1	-2,4	0,09	48,87	$\sin(t) - 7t + 2$
6, 16, 26	0	1,0	1	3,5	9	1	-1	8,8	29,00	$\cos(t) + 5t + 3$
7, 17, 27	0	1,5	1	2,8	5	1	-1,5	- 1,25	53,375	$e^{-t} + \cos(2t)$
8, 18, 28	0	1,5	1,5	4,0	10	1	-4,6	3,94	34,28	$e^{-1.5t} + 2\sin(3t)$
9, 19, 29	0	1,5	0	2,5	8	1	-4,1	0,64	42,85	$e^{-2t} + 3\sin(2.5t)$
10, 20, 30	0	1,5	0	3,1	9	1	-3,9	9,43	26,295	$\sin(2t) + 2\cos(3t)$

Задание 3. Аналитически и численно решите систему дифференциальных уравнений. Постройте графики решений.

№	Система дифференциальных уравнений	№	Система дифференциальных уравнений
31.	$\begin{cases} x' = -x + 4y + e^t \\ y' = x + 2y \end{cases}$	32.	$\begin{cases} x' = 2\sqrt{1+x} - 2e^y, \\ y' = \cos x - 2y - 1. \end{cases}$
33.	$\begin{cases} x' = -4x - 5y + 3e^t \\ y' = x + 2y \end{cases}$	34.	$\begin{cases} x' = y - 4, \\ y' = -2y - x^2 + 3x. \end{cases}$
35.	$\begin{cases} x' = x - y + \frac{1}{\cos t}, \\ y' = 2x - y. \end{cases}$	36.	$\begin{cases} x' = 2x + 2y - 2e^t, \\ y' = x + 3y + e^t \end{cases}$
37.	$\begin{cases} x' = x - 3y + x \sin y, \\ y' = x + 4y + 1 - \cos y. \end{cases}$	38.	$\begin{cases} x' = -x + y + 4t^2 e^{t^2}, \\ y' = x + y. \end{cases}$
39.	$\begin{cases} x' = \frac{x}{x+y}, \\ y' = \frac{y}{x+y}, \end{cases}$	40.	$\begin{cases} x' = y, \\ y' = -y - x^3 + 4x^2 - 3x. \end{cases}$
41.	$\begin{cases} x' = x + 3y, \\ y' = -2x - y \end{cases}$	42.	$\begin{cases} x' = 3x - 4y + x^2, \\ y' = -2x + 2y - \sin y. \end{cases}$
43.	$\begin{cases} x' = -2x + 8 \sin y, \\ y' = -x - 3y + 4x^3. \end{cases}$	44.	$\begin{cases} x' = y, \\ y' = 0,4y - x^3 + 9x. \end{cases}$

45.	$\begin{cases} x' = y + 1, \\ y' = -y + x^2 - x. \end{cases}$	46.	$\begin{cases} x' = 3x - 4y + t \\ y' = 2x - 3y \end{cases}$
47.	$\begin{cases} x' = -x - 6y, \\ y' = x + 4y + \cos t \end{cases}$	48.	$\begin{cases} x' = -2x - 2y, \\ y' = 3x - 10y \end{cases}$
49.	$\begin{cases} x' = x^2 + xy, \\ y' = xy + y^2, \end{cases}$	50.	$\begin{cases} x' = 2x - 5y + 4 \sin t, \\ y' = x - 2y. \end{cases}$
51.	$\begin{cases} x' = 2x - 2y + e^t \sin(e^t), \\ y' = x - y + e^t \sin(e^t). \end{cases}$	52.	$\begin{cases} x' = y - 1 + e^x, \\ y' = -x + \sin 2y. \end{cases}$
53.	$\begin{cases} x' = -2x - 8 \sin y, \\ y' = x - 3y + 4x^3 + x^5. \end{cases}$	54.	$\begin{cases} x' = y, \\ y' = -4y + x^3 - 4x^2 - 5x. \end{cases}$
55.	$\begin{cases} x' = y, \\ y' = -2y - x(x^2 - 1). \end{cases}$	56.	$\begin{cases} x' = 3x + y - \sin 2t \\ y' = x + 3y \end{cases}$
57.	$\begin{cases} x' = -x - 8y + \cos 2t, \\ y' = x + 5y \end{cases}$	58.	$\begin{cases} x' = 3x - 2y, \\ y' = 2x - y + 15e^t \sqrt{t}. \end{cases}$
59.	$\begin{cases} x' = 4x - 3y + 2,5e^{3t}, \\ y' = 2x - y + e^{-3t}. \end{cases}$	60.	$\begin{cases} x' = 3\sqrt{1+x} - 3e^y, \\ y' = -2 \sin x + 0,5y. \end{cases}$

Лабораторная работа

Matlab: Функции для решения задач интерполяции

Цель работы: получение навыков использования функций Matlab для решения задач интерполяции.

Теоретические сведения

В Matlab для решения задач интерполяции имеются следующие функции:
spline — интерполяция функции одной переменной кубическим сплайном;
interp1 — одномерная табличная интерполяция;
interp2 — двумерная табличная интерполяция;
griddata — двумерная табличная интерполяция на неравномерной сетке.

Интерполяция функции одной переменной кубическим сплайном.

Функция $y_i = \text{spline}(x, y, x_i)$ интерполирует значения функции y в точках x_i внутри области определения функции, используя кубические сплайны.

Пример 1. Зададим синусоиду 11 точками и проведем интерполяцию кубическими сплайнами, используя мелкую сетку (рис. 1):

```
x = 0:10;
y = sin(x);
xi = 0:.25:10;
yi = spline(x, y, xi);
plot(x, y, 'o', xi, yi, 'g')
grid on
```

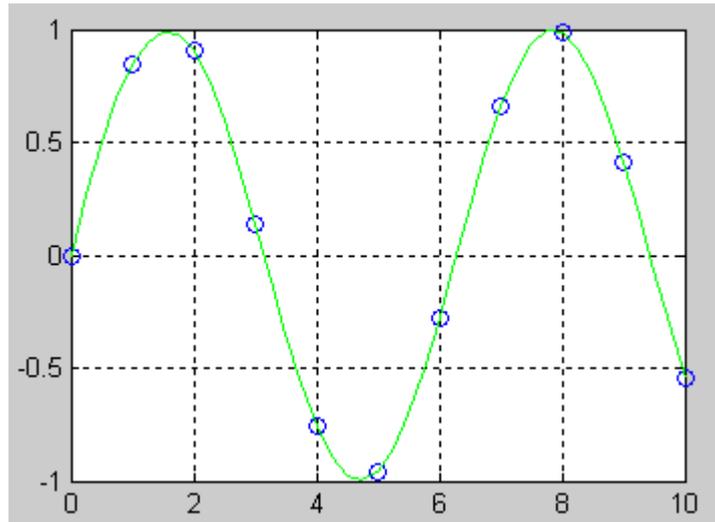


Рис. 1.

Большие возможности для работы со сплайнами предоставляет пользователю специализированный пакет **Spline Toolbox**.

Одномерная табличная интерполяция. Функция $y_i = \text{interp1}(x, y, x_i)$ строит интерполирующую кривую для одномерного массива y , заданного на сетке x ; выходной массив y_i может быть определен на более мелкой сетке x_i . Если Y — двумерный массив, то интерполирующая кривая строится для каждого столбца. По умолчанию реализована линейная интерполяция.

Функция $y_i = \text{interp1}(x, y, x_i, \text{'метод'})$ позволяет задать метод интерполяции:

'linear'	линейная
'cubic'	кубическая
'spline'	кубические сплайны

Принято, что аргумент x изменяется монотонно; кроме того, для кубической интерполяции предполагается, что сетка по x равномерна.

Пример 2. Зададим синусоиду 11 точками и проведем интерполяцию, используя мелкую сетку (рис. 2):

```
x = 0:10;
y = sin(x);
xi = 0:.25:10;
yi = interp1(x, y, xi);
plot(x, y, 'o', xi, yi, 'g')
hold on
yi = interp1(x, y, xi, 'spline');
```

```
plot(x, y, 'ob', xi, yi, 'm')
grid on
```

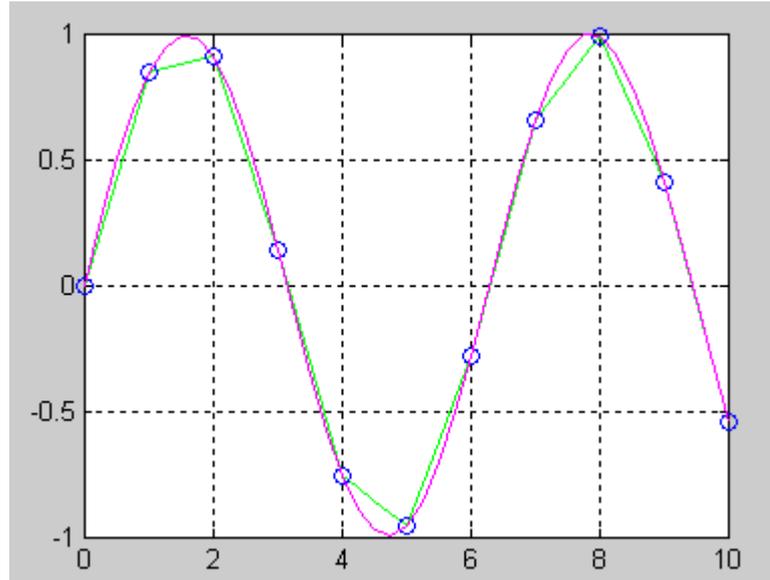


Рис. 2.

Двумерная табличная интерполяция. Функция **ZI = interp2(X, Y, Z, XI, YI)** интерполирует данные, определяющие некоторую поверхность на двумерной сетке (X, Y) ; выходной массив ZI может быть определен на более мелкой сетке (XI, YI) . По умолчанию реализована линейная интерполяция.

Функция **ZI = interp2(X, Y, Z, XI, YI, 'метод')** позволяет задать метод интерполяции:

'linear'	линейная
'cubic'	кубическая

Принято, что аргументы X и Y изменяются монотонно; кроме того, для кубической интерполяции предполагается, что сетка (X, Y) равномерна.

Пример 3. Проведем интерполяцию функции **peaks**, используя мелкую сетку (рис. 3):

```
[X, Y] = meshgrid(-3:0.25:3);
Z = peaks(X, Y);
[XI, YI] = meshgrid(-3:0.125:3);
ZI = interp2(X, Y, Z, XI, YI);
mesh(X, Y, Z)
hold on
mesh(XI, YI, ZI+15)
```

Функция **peaks** осуществляет построение поверхности с нормальным законом распределения значений по координате Z .

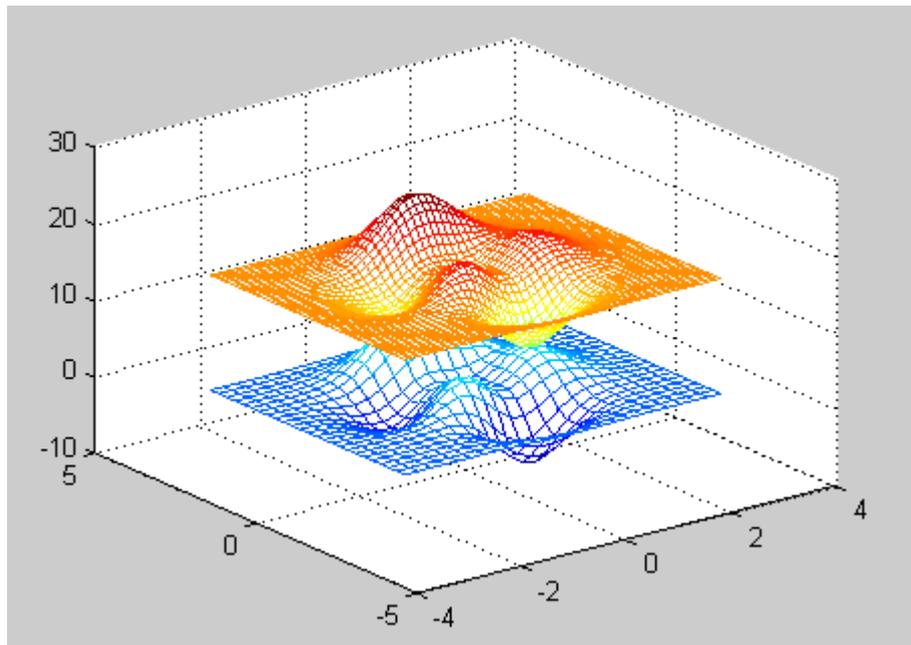


Рис. 3.

Двумерная интерполяция на неравномерной сетке. Функция **ZI = griddata(X, Y, Z, XI, YI)** возвращает массив **ZI**, который определен на новой сетке (**XI, YI**) в результате интерполяции исходной функции **Z**, заданной на неравномерной сетке (**X, Y**).

Функция **[XI, YI, ZI] = griddata(X, Y, Z, XI, YI)** кроме массива **ZI** возвращает массивы **XI, YI**, упорядоченные по аналогии с функцией **meshgrid**.

Пример 4. Определим функцию на сетке, заданной 100 точками, выбранными случайно на отрезке $[-2; 2]$:

```
X = rand(100, 1) * 4 - 2;
Y = rand(100, 1) * 4 - 2;
Z = X.*exp(-X.^2 - Y.^2);
```

При этом функция **X = rand(m, n)** формирует массив размера $m \times n$, элементами которого являются случайные величины, распределенные по равномерному закону в интервале $(0, 1)$.

Векторы **X, Y, Z** определяют 100 случайных точек на поверхности функции **ZI**, которую зададим на следующей равномерной сетке:

```
ti = -2:0.25:2;
[XI, YI] = meshgrid(ti, ti);
ZI = griddata(X, Y, Z, XI, YI);
```

Построим поверхность функции, полученной в результате интерполяции на неравномерной случайной сетке (рис. 4):

```
mesh(XI, YI, ZI)
hold on
plot3(X, Y, Z, 'r o')
```

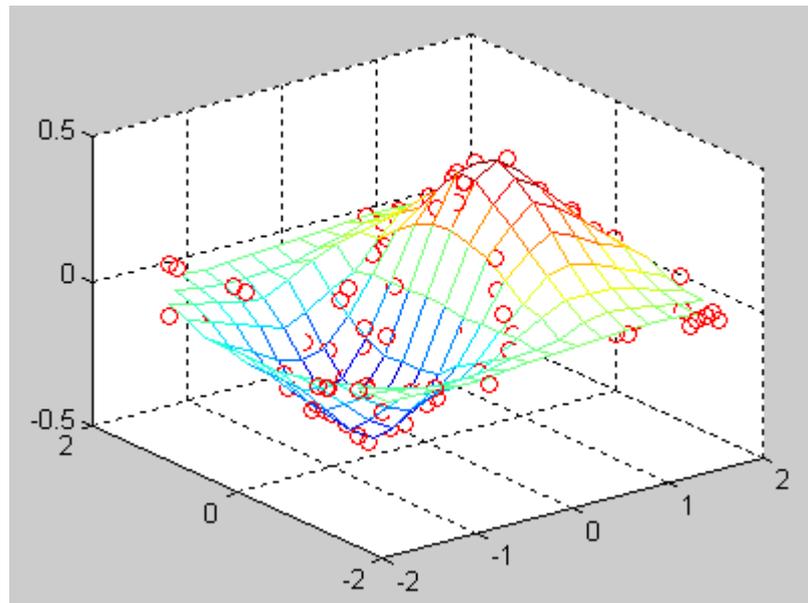


Рис. 4.

Задания

Все задания рекомендуется выполнять в m-файлах.

Задание 1. Проведите одномерную интерполяцию функции $f(x)$ с помощью функции **spline**.

Задание 2. Проведите одномерную интерполяцию функции $f(x)$ с помощью функции **interp1**, используя разные методы. Сравните результаты.

Задание 3. Проведите двумерную интерполяцию поверхности, заданной функцией h с помощью функции **interp2**, используя разные методы. Сравните результаты.

Задание 4. Проведите двумерную интерполяцию поверхности, заданной функцией h с помощью функции **griddata**.

Варианты.

№	Выражения	№	Выражения
1.	$f(x) = x^2 + 2e^x$ $h = (xy^2 + x) + (x^2y - y)$	16.	$f(x) = 3 \cos^2 x - \sqrt{x}$ $h = x^2y + 2xy + \ln x$
2.	$f(x) = e^x \sin x$ $h = y \sin^2 x + x \ln y$	17.	$f(x) = 3 \cos^2 x - \sqrt{x}$ $h = xy + y - 2y^2 \ln x$
3.	$f(x) = 2x + e^{4-x}$ $h = (x + 2y) - \lg(xy)$	18.	$f(x) = \sin^3 x + \cos^2 x$ $h = -y / \sqrt{x^2 - y^2}$
4.	$f(x) = e^x - 2 \sin x$ $h = x^2y - (2xy - y^2)x$	19.	$f(x) = x^2 \cos x$ $h = (y^2 - 3) + (x^2y - 2\sqrt{y})$
5.	$f(x) = x^2 - 2^x$ $h = (1 + x^2)y - 2xy$	20.	$f(x) = 4^x - 8x$ $h = y \cos x + (y + 1) \sin x$
6.	$f(x) = 2^x - \ln x$ $h = (xy^5 - 1) \ln x + 2y$	21.	$f(x) = x^5 - 5^x$ $h = y \operatorname{ctg} x + y + 2$

Информационные технологии в менеджменте

7.	$f(x) = x^3 - e^x$ $h = xy - 3y - x^4 y^2$	22.	$f(x) = \ln x - 5 \sin x$ $h = \cos^2 x \cdot \operatorname{ctg} y + \sin^2 y$
8.	$f(x) = \sin x - 2 \cos x$ $h = 4x^3 e^y + y^4 e^x$	23.	$f(x) = x^3 - e^x$ $h = xy + y(\ln y - \ln x)$
9.	$f(x) = x^2 - 2 \sin x$ $h = \cos y \cdot \sin x$	24.	$f(x) = \sin x + e^{-x}$ $h = (x^2 - 2y^2)x + 2xy$
10.	$f(x) = e^x \cos x$ $h = (x^2 - y^2)y + 2xy$	25.	$f(x) = \sin^2 x - \sqrt{x}$ $h = xy^{2-y} + y - x - 1$
11.	$f(x) = -3x + e^{x-1}$ $h = y \cos x - \frac{y}{\ln y}$	26.	$f(x) = 2^x \cos x$ $h = y - y \operatorname{ctg} x + \frac{1}{\sin x}$
12.	$f(x) = x^3 - 3^x$ $h = 3e^x \operatorname{tg} y + \cos^{-2} y$	27.	$f(x) = e^{x-4} - 4x$ $h = x^2 y - 2xy + y^2$
13.	$f(x) = e^x - \ln x$ $h = xy + y - x e^{\frac{y}{x}}$	28.	$f(x) = \ln x - 4^x$ $h = 2x \cos^2 y + 2y - x^2$
14.	$f(x) = 3 \cos x - \sin x$ $h = y^2 + x^2 y + xy$	29.	$f(x) = 2 \sin x - 3 \cos x$ $h = \left(1 + e^{\frac{x}{y}}\right)x + e^{\frac{x}{y}}(1 - y)$
15.	$f(x) = x^4 - e^x$ $h = x^2 y + 2xy + 3$	30.	$f(x) = x^6 - e^x$ $h = xy + y - y^2$

Лабораторная работа

Matlab: Функции для решения задач аппроксимации

Цель работы: получение навыков использования функций Matlab для решения задач аппроксимации.

Теоретические сведения

В Matlab для решения задач аппроксимации имеются следующие функции:
polyfit — аппроксимация данных полиномом;
interpft — аппроксимация периодической функции на основе быстрого преобразования Фурье.

Аппроксимация данных полиномом. Функция **p = polyfit(x, y, n)** находит коэффициенты полинома степени n , который аппроксимирует функцию $y(x)$ методом наименьших квадратов. Результатом является строка p длины $n + 1$, содержащая коэффициенты аппроксимирующего полинома.

Пример 1. Рассмотрим аппроксимацию функции $y = \sin(x) + \cos(x)$:

Информационные технологии в менеджменте

```
x = (0:0.1:10)';
```

```
y = sin(x)+cos(x);
```

вычислим коэффициенты аппроксимирующего полинома степени 6:

```
p = polyfit(x, y, 6)
```

```
p = 0.0004 -0.0079 0.0323 0.2245 -1.6581 2.0673 0.8047
```

вычислим значения полинома в точках сетки:

```
f = polyval(p, x);
```

Построим графики функции и аппроксимирующего полинома (рис. 1):

```
plot(x, y, 'b o', x, f, 'r')
```

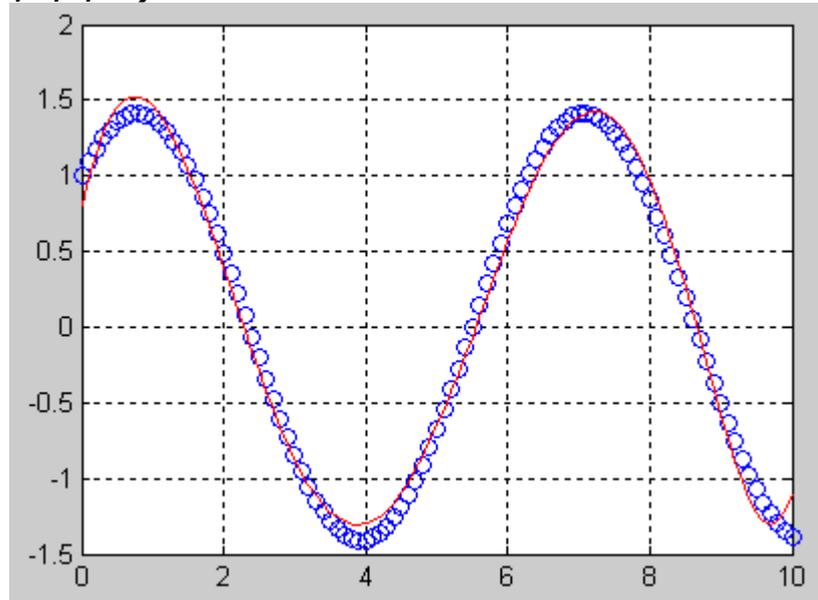


Рис. 1.

Рассмотрим аппроксимацию функции $y = \sin(2x) + \cos(3x)$ (рис. 2). В этом случае для качественной аппроксимации необходимо использовать полином 30 степени:

```
x = (0:0.1:10)';
```

```
y = sin(2*x)+cos(3*x);
```

```
p = polyfit(x, y, 30)
```

```
f = polyval(p, x);
```

```
plot(x, y, 'b o', x, f, 'r')
```

```
grid on
```

Информационные технологии в менеджменте

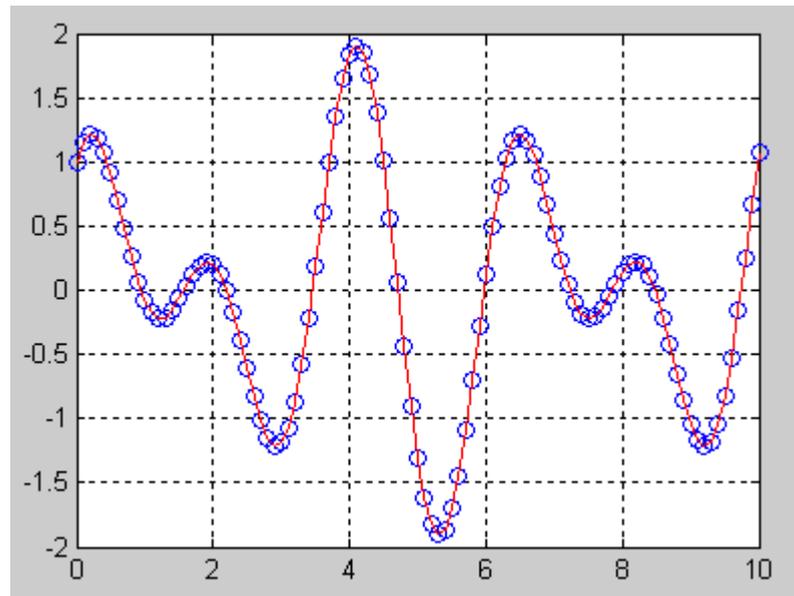


Рис. 2.

Аппроксимация периодической функции на основе быстрого преобразования Фурье. Функция `yp = interpft(y, n)` возвращает одномерный массив чисел, который является периодической функцией, определенной в n точках и аппроксимирующей одномерный массив y . Если $\text{length}(x) = m$, а интервал дискретности dx , то интервал дискретности для y определяется по формуле

$$dy = dx \cdot \frac{m}{n}, \text{ причем } n \text{ всегда превышает } m.$$

Пример 2. Рассмотрим аппроксимацию функции $y = \sin(x)$, которая задана 11 точками на интервале $[0; 10]$ (рис. 3):

```
x = 0:10;
y = sin(x);
xp = 0:0.25:10;
yp = interpft(y, length(xp));
plot(x, y, 'b o -', xp, yp, 'r')
grid on
```

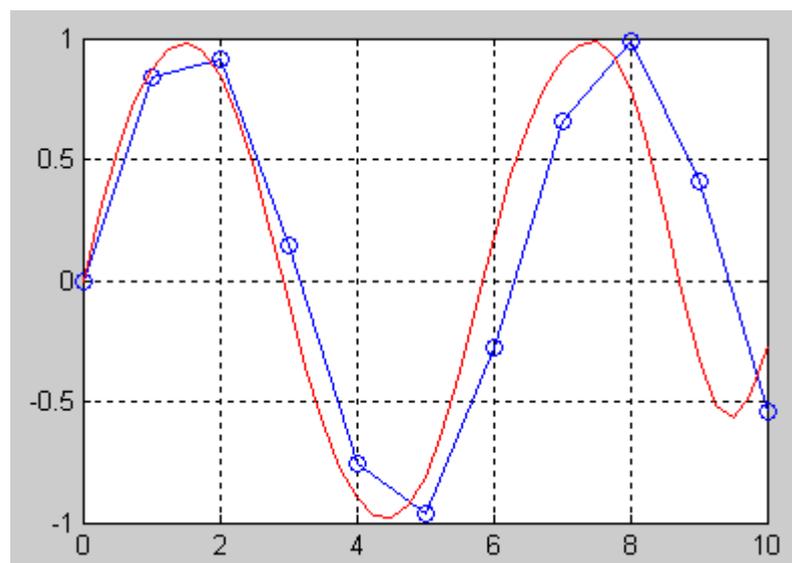


Рис. 3.

Задания

Все задания рекомендуется выполнять в m-файлах.

Задание 1. Проведите аппроксимацию функций $f(x)$ и h с помощью функции **polyfit** (аргумент $y = const$, диапазон и шаг изменения аргумента x задайте самостоятельно). Сделайте выводы о влиянии степени аппроксимирующего полинома на качество аппроксимации.

Задание 2. Проведите аппроксимацию функций $f(x)$ и h с помощью функции **interpft**.

Варианты.

№	Выражения	№	Выражения
1.	$f(x) = x^2 + 2e^x$ $h = (xy^2 + x) + (x^2y - y)$	16.	$f(x) = 3 \cos^2 x - \sqrt{x}$ $h = x^2y + 2xy + \ln x$
2.	$f(x) = e^x \sin x$ $h = y \sin^2 x + x \ln y$	17.	$f(x) = 3 \cos^2 x - \sqrt{x}$ $h = xy + y - 2y^2 \ln x$
3.	$f(x) = 2x + e^{4-x}$ $h = (x + 2y) - \lg(xy)$	18.	$f(x) = \sin^3 x + \cos^2 x$ $h = -y / \sqrt{x^2 - y^2}$
4.	$f(x) = e^x - 2 \sin x$ $h = x^2y - (2xy - y^2)x$	19.	$f(x) = x^2 \cos x$ $h = (y^2 - 3) + (x^2y - 2\sqrt{y})$
5.	$f(x) = x^2 - 2^x$ $h = (1 + x^2)y - 2xy$	20.	$f(x) = 4^x - 8x$ $h = y \cos x + (y + 1) \sin x$
6.	$f(x) = 2^x - \ln x$ $h = (xy^5 - 1) \ln x + 2y$	21.	$f(x) = x^5 - 5^x$ $h = y \operatorname{ctg} x + y + 2$
7.	$f(x) = x^3 - e^x$ $h = xy - 3y - x^4y^2$	22.	$f(x) = \ln x - 5 \sin x$ $h = \cos^2 x \cdot \operatorname{ctg} y + \sin^2 y$
8.	$f(x) = \sin x - 2 \cos x$ $h = 4x^3e^y + y^4e^x$	23.	$f(x) = x^3 - e^x$ $h = xy + y(\ln y - \ln x)$
9.	$f(x) = x^2 - 2 \sin x$ $h = \cos y \cdot \sin x$	24.	$f(x) = \sin x + e^{-x}$ $h = (x^2 - 2y^2)x + 2xy$
10.	$f(x) = e^x \cos x$ $h = (x^2 - y^2)y + 2xy$	25.	$f(x) = \sin^2 x - \sqrt{x}$ $h = xy^{2-y} + y - x - 1$
11.	$f(x) = -3x + e^{x-1}$ $h = y \cos x - \frac{y}{\ln y}$	26.	$f(x) = 2^x \cos x$ $h = y - y \operatorname{ctg} x + \frac{1}{\sin x}$
12.	$f(x) = x^3 - 3^x$ $h = 3e^x \operatorname{tg} y + \cos^{-2} y$	27.	$f(x) = e^{x-4} - 4x$ $h = x^2y - 2xy + y^2$

13.	$f(x) = e^x - \ln x$ $h = xy + y - x e^{\frac{y}{x}}$	28.	$f(x) = \ln x - 4^x$ $h = 2x \cos^2 y + 2y - x^2$
14.	$f(x) = 3 \cos x - \sin x$ $h = y^2 + x^2 y + xy$	29.	$f(x) = 2 \sin x - 3 \cos x$ $h = \left(1 + e^{\frac{x}{y}}\right) x + e^{\frac{x}{y}} (1 - y)$
15.	$f(x) = x^4 - e^x$ $h = x^2 y + 2xy + 3$	30.	$f(x) = x^6 - e^x$ $h = xy + y - y^2$

Лабораторная работа

Matlab: Создание пользовательского интерфейса

Цель работы: получение навыков создания пользовательского интерфейса в пакете Matlab.

Теоретические сведения

Пример 1. Главное меню программы:

```
clear all;
id=0;
while not(id==2); % выход из программы
id=menu(['Интерполяция и аппроксимация функций'];{''},...
        '1. Ввод исходных данных',...
        '2. Выход из программы');
```

switch id

```
case {1}
clc; % очистка командного окна
id=1;
data_fun; % запуск файла ввода исходных данных
end;
end;
```

Пример 2. Организация ввода исходных данных с клавиатуры:

```
disp(' '); % пустая строка
disp('Начальное значение:');
xn=input([sprintf(' xn%g' ) '=']); % в переменную xn заносим начальное значение
диапазона изменения аргумента, введенное с клавиатуры
disp(' ');
disp('Функция:');
y=input([sprintf(' y%g' ) '=']); % в переменную y заносим значение функции,
введенное с клавиатуры
```

Задания

Все задания рекомендуется выполнять в m-файлах.

Задание 1. Создайте пользовательское меню для решения задач интерполяции и аппроксимации функции одной переменной $y = f(x)$.

При этом пользователь в командном окне задает интервалы изменения аргумента x и функцию $y = f(x)$. В пунктах меню выбирает операцию, которую необходимо провести над функцией. К списку выполняемых операций относятся: интерполяция функцией **spline**, интерполяция функцией **interp1**, аппроксимация функцией **polyfit**, аппроксимация функцией **interpft**.

Результатом работы программы должны быть графики исходной и интерполированной (аппроксимированной) функций.

Лабораторная работа Matlab: Simulink

Цель работы: получение навыков моделирования динамических систем и устройств в среде Simulink.

Теоретические сведения

Назначение пакета Simulink

Пакет **Simulink** предназначен для моделирования линейных и нелинейных динамических систем и устройств, представленных своей функциональной схемой, в итоге строится *модель*. При этом возможны различные варианты моделирования: во временной области, в частотной области, с событийным управлением, на основе спектральных преобразований Фурье, с использованием метода Монте-Карло и т. д.

Simulink автоматизирует наиболее трудоемкий этап моделирования: он составляет и решает сложные системы алгебраических и дифференциальных уравнений, описывающих заданную функциональную схему (модель), обеспечивая удобный и наглядный визуальный контроль за поведением созданного пользователем *виртуального устройства*. Достаточно уточнить (если нужно) вид анализа и запустить **Simulink** в режиме *симуляции* (откуда и название пакета — **Simulink**) созданной модели системы или устройства. В русском языке слово «симуляция» носит несколько нарицательный оттенок, поэтому чаще используют термин «моделирование».

Для построения функциональной блок-схемы моделируемых устройств **Simulink** имеет обширную *библиотеку блочных компонентов*. Используя *палитры компонентов* (наборы) блок-схем, пользователь с помощью мыши переносит нужные компоненты с палитр на рабочий лист пакета **Simulink** и соединяет линиями входы и выходы блоков. Таким образом, создается блок-схема моделирования системы или устройства.

Simulink практически мгновенно меняет математическое описание модели по мере ввода новых блоков даже в том случае, когда этот процесс сопровождается сменой порядка системы уравнений и ведет к существенному

Информационные технологии в менеджменте

качественному изменению поведения системы. Это следует отнести к одной из главных целей пакета **Simulink**.

Библиотека компонентов включает в себя источники сигналов с практически любыми временными зависимостями, масштабирующие, линейные и нелинейные преобразователи с разнообразными формами передаточных характеристик, квантующее устройство, интегрирующие и дифференцирующие блоки и так далее.

В библиотеке имеется целый набор виртуальных регистрирующих устройств — от простых измерителей типа вольтметра или амперметра до универсальных осциллоскопов, позволяющих просматривать временные зависимости выходных параметров моделируемых систем — например, токов и напряжений, перемещений, давлений и тому подобного. Имеется даже графопостроитель для построения фигур в полярной системе координат, например, фигур Лиссажу и фазовых портретов колебаний. **Simulink** имеет средства анимации и звукового сопровождения. А в дополнительных библиотеках можно отыскать и такие «дорогие приборы», как анализаторы спектра сложных сигналов, многоканальные самописцы и средства анимации графиков.

Средства анимации **Simulink** позволяют строить виртуальные физические лаборатории с наглядным представлением результатов моделирования. Возможности **Simulink** охватывают задачи математического моделирования сложных динамических систем в физике, электро- и радиотехнике, в биологии и химии, словом, — во всех областях науки и техники. Этим объясняется популярность данного пакета как в университетах и институтах, так и в научных лабораториях.

Как программное средство **Simulink** — типичный представитель *визуально-ориентированного языка программирования*. На всех этапах работы, особенно при подготовке моделей схем, пользователь практически не имеет дела с обычным программированием. Программа автоматически генерируется в процессе ввода выбранных блоков компонентов, их соединений и задания параметров компонентов.

Важное достоинство **Simulink** — это интеграция с системой Matlab и рядом других пакетов расширения, что обеспечивает, по существу, неограниченные возможности в применении **Simulink** для решения практически любых задач имитационного моделирования.

Понятие об S-функциях. Своими возможностями пакет **Simulink** во многом обязан специальному аппарату создания и применения так называемых системных *S-функций* (System Functions). Эти функции позволяют в ходе решения осуществлять сложные функциональные преобразования по различным математическим алгоритмам, например, алгоритмам решения систем дифференциальных уравнений.

Для разработки S-функций **Simulink** имеет специальный редактор. Создав S-функцию, пользователь фактически создает блок библиотеки, который может использоваться по всем правилам применения блоков. Блок можно переносить в окно редактирования с помощью мыши, менять заданную S-функцию, использовать необходимые связи между блоками и т. д.

Библиотека компонентов пакета Simulink

Запуск Simulink. Для запуска программы необходимо предварительно запустить пакет Matlab. После открытия основного окна программы Matlab нужно запустить программу **Simulink**. Это можно сделать одним из трех способов:

— нажать кнопку  (**Simulink**) на панели инструментов командного окна Matlab;

Информационные технологии в менеджменте

— в командной строке главного окна Matlab напечатать **Simulink** и нажать клавишу Enter на клавиатуре;

— выполнить команду **Open...** в меню **File** и открыть файл модели (mdl-файл). Этот вариант удобно использовать для запуска уже готовой и отлаженной модели, когда требуется лишь провести расчеты и не нужно добавлять новые блоки в модель.

Использование первого и второго способов приводит к открытию окна обозревателя разделов библиотеки **Simulink** (рис. 1).

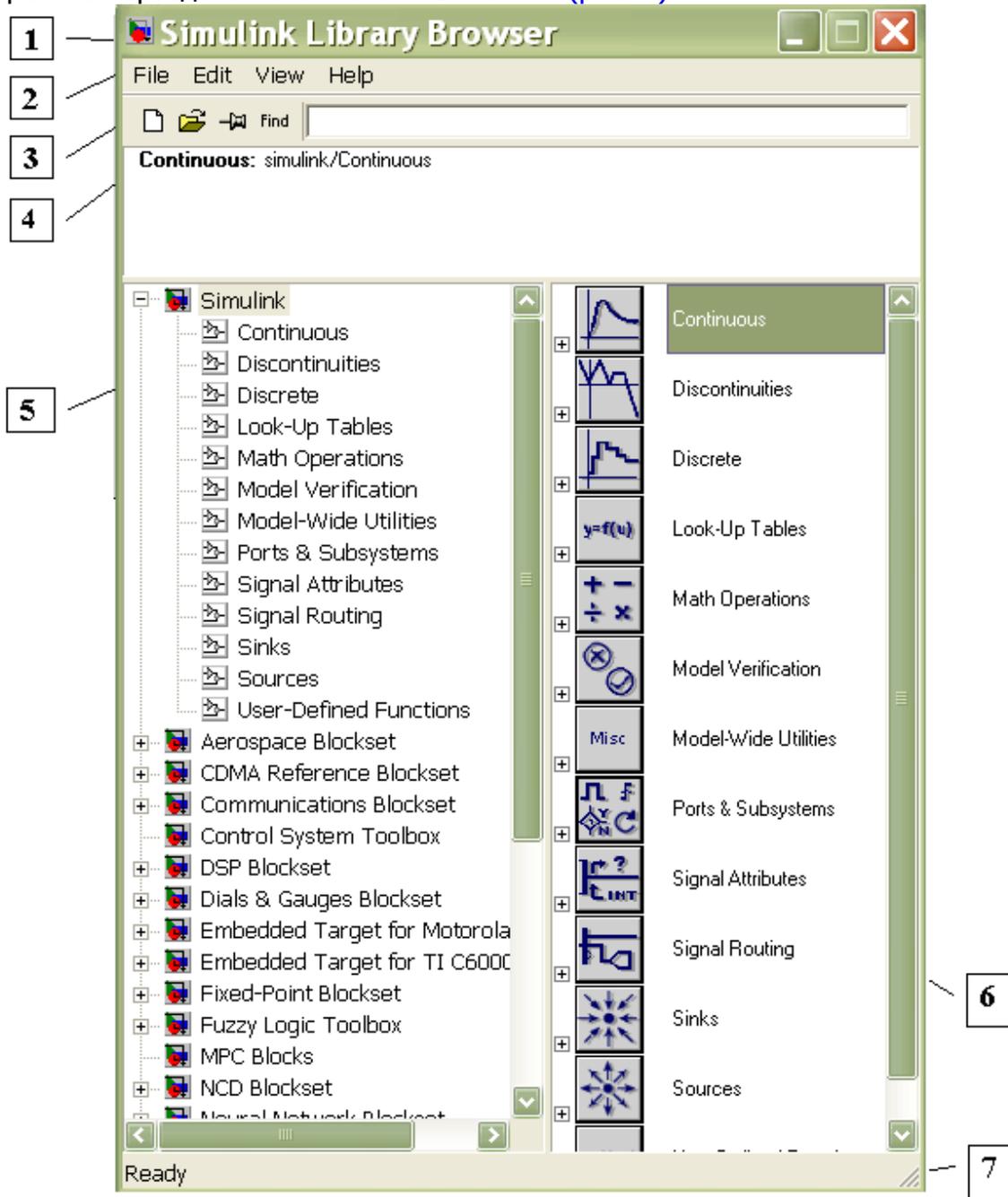


Рис. 1. Окно обозревателя разделов библиотеки **Simulink**
Обозреватель разделов библиотеки Simulink. Окно обозревателя библиотеки блоков содержит следующие элементы:

1. Заголовок, с названием окна — **Simulink Library Browser**.
2. Меню, с командами **File, Edit, View, Help**.

Информационные технологии в менеджменте

3. Панель инструментов, с ярлыками наиболее часто используемых команд.
 4. Окно комментария для вывода поясняющего сообщения о выбранном блоке.
 5. Список разделов библиотеки, реализованный в виде дерева.
 6. Окно содержимого раздела библиотеки (список вложенных разделов библиотеки или блоков)
 7. Строка состояния, содержащая подсказку по выполняемому действию.
- При выборе соответствующего раздела библиотеки в правой части окна отображается его содержимое (рис. 2).

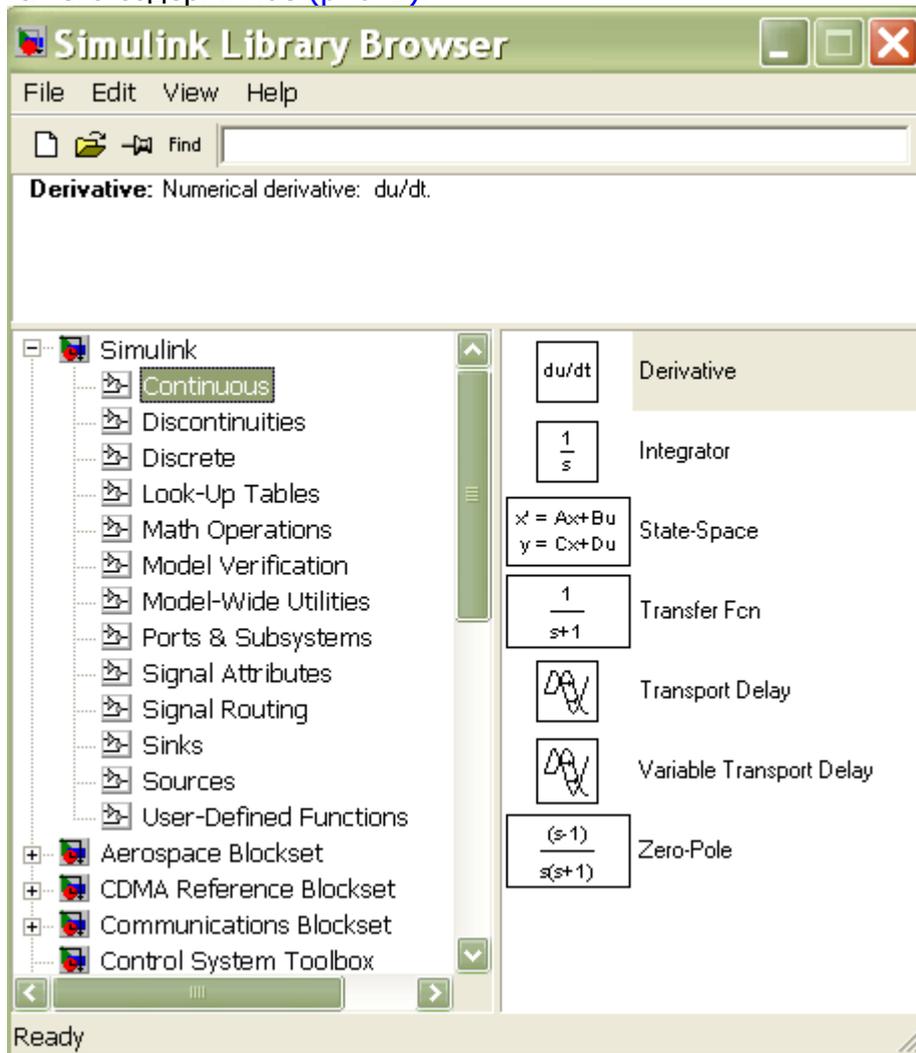


Рис. 2. Окно обозревателя с набором блоков раздела библиотеки

Основные разделы библиотеки:

Continuous — линейные непрерывные компоненты;

Discontinuous — нелинейные компоненты;

Discrete — дискретных компоненты;

Math Operation — математические компоненты;

Ports & Subsystems — компоненты для создания подсистем;

Signal Routing — компоненты для разделения сигналов;

Sinks — регистрирующие компоненты;

Sources — источники сигналов и воздействий;

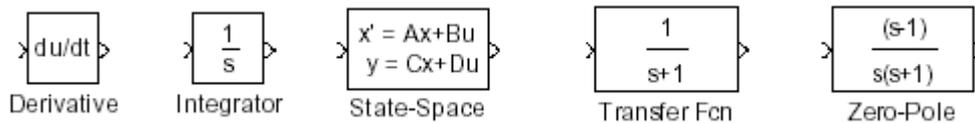
User-Defined Function — компоненты для создания функций пользователя.

Информационные технологии в менеджменте

Если два раза щелкнуть левой кнопкой мыши на каком-нибудь блоке в окне библиотек, то открывается окно параметров блока, где отображаются установки параметров по умолчанию. Как правило, они нормализованы — например, задана единичная частота, единичная амплитуда, нулевая фаза и так далее. Возможность изменения параметров в этом случае отсутствует. Она появляется после переноса графических элементов в окно подготовки и редактирования функциональных схем (для его открытия необходимо выбрать иконку  на панели инструментов или **File** → **New** → **Model**).

Подробно рассмотрим наиболее используемые разделы библиотеки.

Линейные непрерывные компоненты (Continuous). Линейные компоненты играют важную роль в создании математических моделей многих устройств. Имеются следующие типы линейных компонентов: **Derivative** — аналоговое дифференцирующее устройство; **Integrator** — аналоговый интегратор; **State-Space** — модель в пространстве состояний; **Transfer Fcn** — передаточные функции, и ряд других устройств.



Название блока	Тип блока	Параметры	
Derivative	Производная	нет	
Integrator	Интегратор	External reset: <ul style="list-style-type: none"> • none • rising • falling • either • level Initial condition	Внешний сброс: <ul style="list-style-type: none"> • отсутствует • по растущему фронту • по падающему фронту • по любому фронту • по уровню Начальные условия
State-space	Описание в пространстве состояний (ОПС)	Limit output: <ul style="list-style-type: none"> • upper saturation limit • lower saturation limit Initial conditions	Ограничения по выходу: <ul style="list-style-type: none"> • верхний предел насыщения • нижний предел насыщения Матрицы системы Вектор начальных условий
Transfer Fcn.	Передаточная функция	Numerator Denominator	Массив коэффициентов числителя Массив коэффициентов знаменателя
Zero-pole	Нуль-полосное описание системы	Zeros Poles Gain	Массив нулей Массив полюсов Коэффициент усиления

Нелинейные компоненты (Discontinuous). **Simulink** предназначен главным образом для моделирования нелинейных динамических систем. Среди нелинейных компонентов следует отметить компоненты с типичными нелинейностями, например, вида **abs(u)**, с характеристиками, описанными типовыми математическими функциями, компонентами вида идеальных и неидеальных ограничителей и так далее. Представлены и такие сложные компоненты, как квантователи, блоки нелинейности, моделирующие нелинейные петли гистерезиса, временные задержки и ключи-переключатели.



Saturation – насыщение, в параметрах задаются верхний и нижний пределы (**Upper limit** и **Lower limit**).

Информационные технологии в менеджменте



Dead zone – нечувствительность, «мертвая зона». В параметрах задаются пределы нечувствительности (**Start of dead zone** и **End of dead zone**).



Rate Limiter – ограничитель скорости изменения сигнала, в параметрах задаются пределы на скорость увеличения (**Rising slew rate**) и на скорость уменьшения (**Falling slew rate**).



Relay – реле, в параметрах задаются точки переключения (**Switch on point** и **Switch off point**), в также величины сигналов в режимах «включено» (**Output when on**) и «выключено» (**Output when off**).

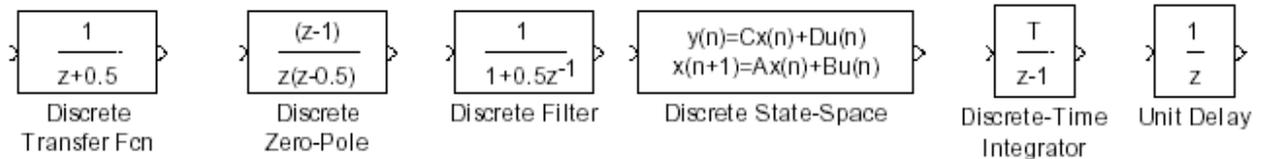


Backlash – люфт, «мертвый ход». В параметрах задаются величина мертвого хода (**Deadband width**) и начальное значение выхода (**Initial output**).



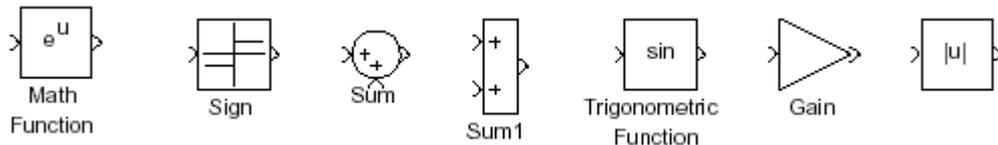
Coulomb and Viscous Friction – кулоновское и вязкое трение.

Дискретные компоненты (Discrete). Дискретные компоненты включают в себя устройства задержки, дискретно-временной интегратор, дискретный фильтр и т. д. Дискретные блоки аналогичны непрерывным блокам:



Среди них есть два блока, не имеющих аналога среди непрерывных блоков. Это блок задержки **Unit delay** и блок **Discrete Filter** — дискретный фильтр, который определяет рациональную функцию от оператора задержки $z / 1$.

Математические компоненты (Math Operation). Окно **Math Operation** позволяет задавать множество математических компонентов с описываемыми пользователями свойствами, имеет решающее значение для выполнения прозрачного для пользователя математического моделирования как простых, так и сложных устройств и систем. Некоторые из них:



Информационные технологии в менеджменте

Название блока	Тип блока	Параметры	
Math function	Математическая функция	Function: exp, log, 10^u , log10, sqrt, Output signal type	Вид функции Тип выходного сигнала
Sign	Знак	• Auto • Real • Complex	• Не определен • Вещественный • Комплексный нет
Sum	Сумма	Icon shape: • Rectangular • Round List of signs	Изображение блока: • Прямоугольное • Круглое Знаки при входах
Trigonometric function	Тригонометрическая функция	Function: sin, cos, tan, asin, acos, sinh, cosh, Output signal type	Вид функции Тип выходного сигнала
Gain	Усилитель	Gain Multiplication: • Element-wise ($K \cdot u$) • Matrix ($K \cdot u$) • Matrix ($u \cdot K$)	Коэффициент усиления Способ умножения: • Поэлементное • Матричное ($K \cdot u$) • Матричное ($u \cdot K$)
Abs	Модуль		нет

Компоненты для создания подсистем (Ports & Subsystems). В случае если моделируемая система имеет сложную схему, удобным может быть объединение нескольких элементов в один с помощью блока **Subsystem**. При этом для задания требуемого количества входов и выходов нового элемента используются блоки **In1** и **Out1**.

Компоненты для разделения сигналов (Signal Routing). Иногда для большей наглядности необходимо отобразить несколько графиков в одном окне, для этого может быть использован блок **Mux** из библиотеки **Signal Routing**, который объединяет любое количество входных воздействий в один выходной сигнал, представляющий собой вектор. С помощью блока **Demux** выполняется обратная операция — разъединение одного сигнала (вектора) на несколько.



Manual Switch — ручной переключатель, позволяет двойным щелчком переключать выход на один из двух входных сигналов.



Mux — мультиплексор, объединяет несколько сигналов в один «жгут» (векторный сигнал), в параметрах задается число входов (**Number of Inputs**).



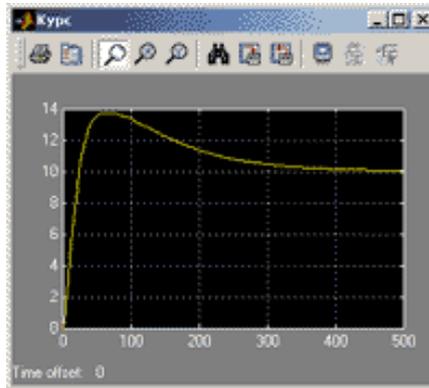
Demux — демультиплексор, позволяет «разбить» векторный сигнал на несколько скалярных, в параметрах задается число выходов (**Number of Outputs**).

Регистрирующие элементы (Sinks). Устройства наблюдения за процессами на выходах звеньев схемы сосредоточены в библиотеке приемников сигналов **Sinks**. Приемники сигналов оформлены в виде блоков, имеющих только входы (один или несколько).

На схеме можно разместить несколько приемников, воспринимающих сигналы на выходах различных звеньев, в том числе и непосредственно от источников сигналов.

Информационные технологии в менеджменте

Наиболее простым устройством для отображения процесса является **Scope**, имитирующий осциллограф. Он имеет один вход, и соответственно одно окно для отображения результата. Если необходимо вывести несколько графиков, то на панели инструментов графика выбирается **Parameters** → **General** и в поле **Number of axes** выставляется количество осей, которому будет соответствовать количество входов блока **Scope**.



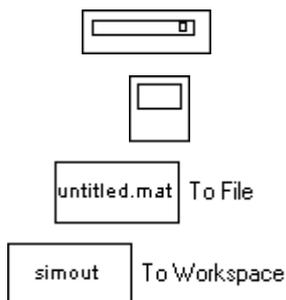
В окне блока **Scope** изображается график изменения входного сигнала. Если вход соединен с выходом мультиплексора, сразу строится несколько графиков (по размерности входного «жгута»).

По умолчанию на оси ординат используется диапазон от -5 до 5. Если этот вариант не подходит, выбрать масштаб автоматически (так, чтобы весь график был виден) можно с помощью кнопки . Соседняя кнопка  сохраняет эти настройки для следующих запусков.

Кнопка  открывает окно настроек, причем наиболее важные данные содержатся на вкладке **Data history**. Если не сбросить флажок **Limit data points**, в памяти будет сохраняться только заданное число точек графика, то есть, при большом времени моделирования начало графика будет потеряно.

Отметив на этой же странице флажок **Save data to workspace** можно сразу передать результаты моделирования в рабочую область MATLAB для того, чтобы их можно было дальше обрабатывать, выводить на графики и сохранять в файле. Поле **Variable name** задает имя переменной в рабочей области, в которой сохраняются данные. В простейшем случае выбирается формат **Array** (в списке **Format**). Это означает, что данные будут сохраняться в массиве из нескольких столбцов (первый столбец – время, второй – первый сигнал, третий – второй сигнал и т.д., по порядку входов мультиплексора).

Блок **XY Scope** применяется при необходимости построения фазовых портретов и статических характеристик.



Display – цифровой дисплей, показывает изменение входного сигнала в цифровом виде.

Scope – осциллограф, показывает изменение сигнала в виде графика.

To File — передача данных в файл **untitled.mat**.

To Workspace — сохранение данных в переменной **simout** и передача в **Workspace**.

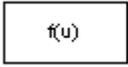
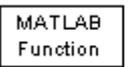
Источники сигналов и воздействий (Sources). Окно **Sources** содержит графические элементы — источники воздействий. В электро- и радиотехнике их принято называть источниками сигналов, но в механике и в других областях науки и техники такое название не очень подходит — природа воздействий может быть самой разнообразной, например, в виде перепада давления или температуры, механического перемещения, звуковой волны и так далее.

Большинство элементов содержит рисунок, представляющий временную зависимость воздействий, например, перепад для блока **Step**, синусоиду для блока **Sine Wave** и т. д.

Набор блоков содержит практически все часто используемые при моделировании источники воздействий с самой различной функциональной и временной зависимостью. Возможно задание произвольного воздействия из файла — блок **From File**.

- 
Constant – сигнал постоянной величины.
- 
Step – ступенчатый сигнал, меняется время скачка (**Step Time**), начальное (**Initial Value**) и конечное значение (**Final Value**).
- 
Ramp – линейно возрастающий сигнал с заданным наклоном (**Slope**). Можно задать также время начала изменения сигнала (**Start Time**) и начальное значение (**Initial Value**).
- 
Pulse Generator – генератор прямоугольных импульсов, задаются амплитуда (**Amplitude**), период (**Period**), ширина (**Pulse Width**, в процентах от периода), фаза (**Phase Delay**).
- 
Repeating Sequence – последовательность импульсов, их форма задается в виде пар чисел (время; величина сигнала)
- 
Sine Wave – синусоидальный сигнал, задается амплитуда (**Amplitude**), частота (**Frequency**), фаза (**Phase**) и среднее значение (**Bias**).
- 
Signal Builder – построитель сигналов, позволяющий задавать форму сигнала, перетаскивая мышью опорные точки.
- 
Random Number – случайные числа с нормальным (гауссовым) распределением. Можно задать среднее значение (**Mean Value**), дисперсию (**Variance**), период изменения сигнала (**Sample Time**).
- 
Uniform Random Number – случайные числа с равномерным распределением в заданном интервале от **Minimum** до **Maximum**.
- 
Band Limited White Noise – случайный сигнал, ограниченный по полосе белый шум (имеющий равномерный спектр до некоторой частоты). Блок используется как источник белого шума для моделей непрерывных систем. Задается интенсивность (**Noise Power**) и интервал дискретизации (**Sample Time**), в течение которого удерживается постоянное значение сигнала. Чем меньше интервал, тем точнее моделирование, однако больше вычислительные затраты.

Компоненты для создания функций пользователя (User-Defined Function).

 Fcn	Fcn — задание функции преобразования входного сигнала.
 MATLAB Fcn	Matlab Fcn — вызов файла-функции Matlab (в параметрах блока указывается имя файла).
 S-Function  S-Function Builder	S-function S-Function Builder — создание S-функции.

Построение схемы и проведение имитации. После того как все необходимые для построения модели устройства блоки перенесены на рабочий лист пакета **Simulink**, их соединяют линиями связи, по которым распространяются сигналы. Для этого надо щелкнуть левой кнопкой мыши по источнику сигнала и затем, при нажатой клавише **Ctrl**, по блоку-приемнику. Можно также протянуть мышкой линию связи между нужными выходом и входом.

Для выделения одного блока или соединительной линии надо щелкнуть левой кнопкой мыши по нужному элементу. Для того чтобы выделить несколько блоков, надо «обвести» их при нажатой левой кнопки мыши. Клавиша **Delete** удаляют выделенную часть. Чтобы скопировать блок (или выделенную часть), надо перетащить его при нажатой правой кнопке мыши.

Для изменения оформления выделенного блока предназначено меню **Format** на верхней панели инструментов. Также для этой цели можно использовать контекстное меню **Format** при нажатии на блоке правой кнопкой мыши. Для выделенного блока можно изменить цвет текста и линий (**Foreground color**), цвет фона (**Background color**), вывести тень (**Show drop shadow**), переместить название на другую сторону (**Flip name**).

Для имитационного моделирования следует отредактировать «параметры» имитации и запустить процесс.

Команда из меню **Simulation** → **Parameters** раскрывает диалоговое поле редактирования параметров имитации, что подразумевает выбор алгоритма численного интегрирования уравнений и задание числовых параметров:

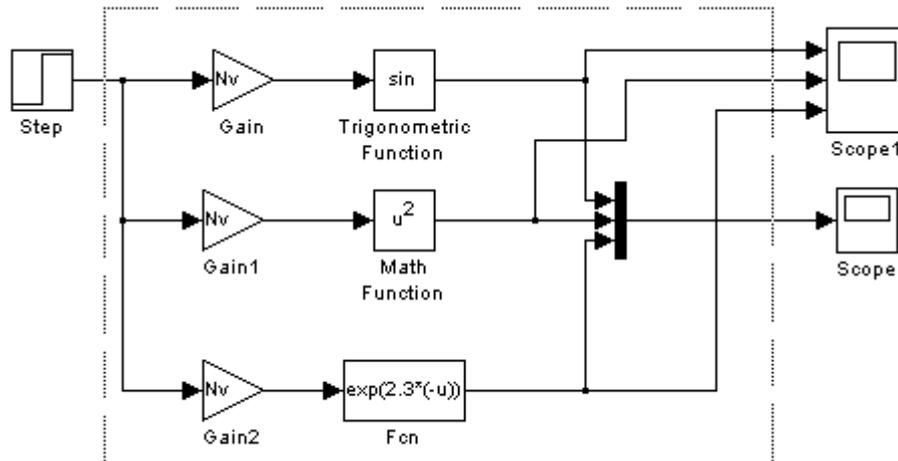
- **Start Time** — время начала процесса (обычно равно 0);
- **Stop Time** — время окончания процесса имитации;
- **Min Step Size** — минимальная величина шага интегрирования;
- **Max Step Size** — максимальная величина шага интегрирования;
- **Tolerance** — допуск (допустимая погрешность).

Величину минимального шага следует назначать с учетом скорости протекания процессов в системе. Максимальный шаг интегрирования можно принять на один или два порядка большим.

Использование переменных для задания параметров блоков. Параметры блоков **Simulink** можно задавать не только как числовые параметры, но и как имена переменных. Значения переменных необходимо задать перед началом расчета в командном окне Matlab или в m-файле. Такой способ задания параметров удобен, если требуется проводить расчеты для множества вариантов параметров блоков модели.

Задания

Задание 1. Постройте функциональную схему и проведите моделирование процесса работы системы:



Здесь Nv — номер варианта; блок **Fcn** должен содержать функцию $f(x)$, заданную в индивидуальном варианте; в блоках **Trigonometric Function** и **Math Function** самостоятельно выберете функции.

Параметры блока **Step** задайте самостоятельно.

Задание 2. Проведите моделирование при подаче на вход сигналов из блоков **Ramp** и **SineWave**. Параметры блоков **Ramp** и **SineWave** задайте самостоятельно.

При этом целесообразно всю неизменяемую часть схемы (на рис. обозначена пунктиром) объединить в один элемент с помощью блока **Subsystem** (раздел **Ports & Subsystems**). В данном случае новый элемент будет иметь один вход и четыре выхода.

Задание 3. Проведите моделирование с добавлением интеграторов **Integrator** (затем дифференциаторов **Derivative**) после блока функций.

Задание 4. Зарегистрируйте сигнал равный сумме выходных сигналов блоков функций (блок **Sum**). Зарегистрируйте сигнал равный произведению выходных сигналов блоков функций (блок **Product**).

Задание 5. Запишите выходной сигнал модели в файл с помощью блока **To File**. Чтобы просмотреть содержимое файла, необходимо в окне **Current Directory** выбрать соответствующее имя файла и два раза нажать на него левой кнопкой мыши. В итоге в окне **Workspace** появится переменная **ans**, представляющая собой матрицу, первая строка которой содержит значения моментов времени моделирования, а последующие строки значения регистрируемых сигналов.

Варианты.

№	Выражения	№	Выражения
1.	$f(x) = x^2 + 2e^{\frac{x}{10}}$	16.	$f(x) = 3 \cos^2 x - \sqrt{x}$
2.	$f(x) = e^x \sin x$	17.	$f(x) = 3 \cos^2 x - \sqrt{x}$
3.	$f(x) = 2x + e^{4-x}$	18.	$f(x) = \sin^3 x + \cos^2 x$
4.	$f(x) = e^x - 2 \sin x$	19.	$f(x) = x^2 \cos x$

Информационные технологии в менеджменте

5.	$f(x) = x^2 - 2^x$	20.	$f(x) = 4^x - 8x$
6.	$f(x) = 2^x - \ln x$	21.	$f(x) = x^5 - 5^x$
7.	$f(x) = x^3 - e^x$	22.	$f(x) = \ln x - 5 \sin x$
8.	$f(x) = \sin x - 2 \cos x$	23.	$f(x) = x^3 - e^x$
9.	$f(x) = x^2 - 2 \sin x$	24.	$f(x) = \sin x + e^{-x}$
10.	$f(x) = e^x \cos x$	25.	$f(x) = \sin^2 x - \sqrt{x}$
11.	$f(x) = -3x + e^{x-1}$	26.	$f(x) = 2^x \cos x$
12.	$f(x) = x^3 - 3^x$	27.	$f(x) = e^{x-4} - 4x$
13.	$f(x) = e^x - \ln x$	28.	$f(x) = \ln x - 4^x$
14.	$f(x) = 3 \cos x - \sin x$	29.	$f(x) = 2 \sin x - 3 \cos x$
15.	$f(x) = x^4 - e^x$	30.	$f(x) = x^6 - e^x$

Задание 6. Проведите моделирование работу системы — гидравлический резервуар в случаях различных типов входных воздействий:

- одиночный перепад;
- линейно нарастающее воздействие.

Сделать выводы по результатам моделирования.

Дифференциальное уравнение процесса имеет вид:

$$S_p \cdot \frac{dH}{dt} = Q - G,$$

где Q — приток воды в резервуар (входная величина) ($\text{м}^3/\text{с}$); G — расход воды из резервуара (внешнее воздействие) ($\text{м}^3/\text{с}$); H — высота столба воды в резервуаре (выходная величина) (м); S_p — площадь резервуара (м^2).

Итоговые вопросы по дисциплине



**Вопросы
по направлению 090900 «Информационная
безопасность»**

Раздел 1. Введение и ознакомление с прикладным программным пакетом Mathcad

<p>Структура и основные компоненты ППП Основные группы пакетов прикладных программ Эволюция ППП</p>
<p>Основные возможности пакета Mathcad Сравнительная характеристика пакета Mathcad с аналогичными системами Расширение функциональности пакета Mathcad Взаимодействие Mathcad с другими программами Структура Mathcad-документа и состав панели инструментов Math (Математика)</p>
<p>Выражения и стандартные функции Определение переменных (присвоение значений переменным) Размерности физических величин Функции, определяемые пользователем Запись логических выражений. Функция if Переменные диапазона (ранжированные переменные)</p>
<p>Состав панели инструментов Math → Graph (Математика → График) Построение X-Y-графиков Построение графиков в полярной системе координат Форматирование двумерных графиков Построение трехмерных графиков Форматирование трехмерных графиков Создание анимации</p>

Раздел 2. Прикладной программный пакет Mathcad

<p>Задание индексированной переменной Операторы суммы Σ и произведения Π Задание и обработка массивов Символьные вычисления</p>
<p>Дифференцирование и интегрирование Решение уравнений и их систем Решение дифференциальных уравнений и их систем</p>
<p>Элементы программирования</p>
<p>Функции для решения задач интерполяции Функции для решения задач аппроксимации</p>
<p>Функции для решения задач сглаживания Функции для решения задач экстраполяции (предсказания)</p>

Раздел 3. Прикладной программный пакет Matlab

<p>Основные возможности пакета Matlab Наборы инструментов пакета Matlab Структура и рабочие окна пакета Matlab Работа в командном режиме Основные элементы языка программирования Matlab</p>
<p>Ввод матриц Индексы элементов матриц Матричные функции линейной алгебры Массивы</p>
<p>Создание графика Окна изображений. Добавление кривых на существующий график. Подграфики Управление осями. Подписи к осям и заголовки Трехмерные графики</p>
<p>Создание символьных переменных и выражений Символьные операции с выражениями</p>
<p>Дифференцирование Интегрирование Решение уравнений и их систем</p>
<p>Назначение языка программирования Matlab Управляющие структуры</p>
<p>Понятие о файлах-функциях Глобальные переменные Векторизация</p>
<p>Решение дифференциальных уравнений и их систем</p>
<p>Функции для решения задач интерполяции Функции для решения задач аппроксимации</p>
<p>Создание пользовательского интерфейса</p>
<p><i>Вывод списка пакетов расширения</i></p> <p><i>Обзор пакетов расширения Matlab</i></p>
<p>Назначение пакета Simulink Библиотека компонентов пакета Simulink</p>

Методические указания к курсовой работе



Тема: «Компьютерное моделирование процессов»

1. Цель курсовой работы

Развитие способностей применять программные средства прикладного назначения и системы программирования для решения математических задач.

2. Задание на курсовую работу

В процессе выполнения курсовой работы с помощью пакетов прикладных программ Mathcad и Matlab провести компьютерное моделирование процессов, описываемых математическими моделями:

1. Математические модели в физике:

- движение снаряда;
- гармонический осциллятор;
- движение груза, подвешенного к концу пружины (уточненная модель гармонического осциллятора).

2. Математические модели в биологии:

- модель Мальтуса;
- логистическая модель (уточненная модель Мальтуса);
- система хищник-жертва.

При этом необходимо:

- в пакете Matlab найти аналитическое решение уравнений математической модели;
- в пакетах Mathcad и Matlab найти численное решение уравнений при заданных параметрах с использованием всех функций пакетов предназначенных для решения дифференциальных уравнений;
- результаты моделирования отобразить графически;
- провести сравнение результатов, полученных разными методами в пакетах Mathcad и Matlab.

Такие параметры, как начальные условия (смещение груза x_0 и скорость груза v_0 в начальный момент времени $t=0$, численность популяции N_0 в начальный момент времени $t=0$, численность популяции жертв x_0 и хищников y_0 в начальный момент времени $t=0$), число шагов интегрирования, временной интервал моделирования процесса студенты выбирают самостоятельно в соответствии с особенностями конкретного процесса.

3. Варианты

Математические модели в физике

№	Гармонический осциллятор		Движение груза, подвешенного к концу пружины				Движение снаряда	
	m	k	m	k	H	$f(t)$	v_0	α
1.	1	1	1	1	0,5	\sqrt{t}	37	$\pi/2$
2.	0,5	4	0,5	4	1	$t\sin(t)$	35	$\pi/3$
3.	0,75	5	0,75	5	1	$-\sqrt{t}$	45	$\pi/4$

Информационные технологии в менеджменте

4.	3	1	3	1	1	$\cos(t)$	49	$\pi/5$
5.	1	5	1	5	0,5	$\frac{5e^{\cos t}}{2 + \cos^3 t}$	55	$\pi/2$
6.	1	5	1	5	1	$\arctg(2e^{\cos t})$	43	$\pi/3$
7.	5	1	5	1	1	$\frac{e^{\sin 2t}}{\sqrt{2 + \cos t}}$	59	$\pi/4$
8.	0,5	3	0,5	3	0,25	$\cos^3 t + e^{\sin(t)}$	34	$\pi/5$
9.	2	1	2	1	1	$\cos(0,5t)$	32	$\pi/2$
10.	0,5	2	0,5	2	0,5	$-\cos(0,5t)$	40	$\pi/3$
11.	1	1	1	1	0,5	$\sqrt{2t} + t$	68	$\pi/4$
12.	0,5	1	0,5	1	1	$t \sin(t)$	47	$\pi/5$
13.	0,75	5	0,75	5	1	$-\sqrt{2t} + \sin(t)$	65	$\pi/2$
14.	6	3	6	3	1	$\cos(t)$	33	$\pi/3$
15.	0,5	5	0,5	5	0,5	$\frac{\cos t}{1 + \cos^2 2t}$	50	$\pi/4$
16.	0,4	1	0,4	1	1	$\frac{\sin t}{1 + \cos^2 2t}$	32	$\pi/5$
17.	0,5	4	0,5	4	1	$\frac{\sin 2t}{3 + \cos 2t}$	47	$\pi/2$
18.	0,4	5	0,4	5	1	$\frac{\sin 3t}{ \sin t + \cos t }$	37	$\pi/3$
19.	0,4	1	0,4	1	0,5	$\cos(e^{\sin 3t})$	46	$\pi/4$
20.	0,5	5	0,5	5	1	$\cos t \cos \sin t $	35	$\pi/5$
21.	0,45	5	0,45	5	1	$\arctg(\cos t)$	58	$\pi/2$
22.	0,5	1	0,5	1	0,25	$e^{\cos 2t} + e^{\cos 3t}$	47	$\pi/3$
23.	0,45	3	0,45	3	1	$ \sin t + \cos 2t$	45	$\pi/4$
24.	0,4	1	0,4	1	0,5	$\sin^3 t + e^{\sin(t)}$	43	$\pi/5$
25.	0,45	2	0,45	2	0,5	$\frac{\cos t}{1 + \sin^2 2t}$	42	$\pi/2$
26.	0,4	1	0,4	1	1	$\frac{\sin t}{1 + \sin^2 2t}$	34	$\pi/3$
27.	0,4	1	0,4	1	1	$e^{10 \cos t}$	65	$\pi/4$
28.	0,45	5	0,45	5	1	$e^{\cos 2t} \sin t$	78	$\pi/5$
29.	0,4	3	0,4	3	0,5	$e^{-\cos t} \sin t$	66	$\pi/2$
30.	0,45	3	0,45	3	1	$\frac{\sin 3t}{2 + \cos t}$	23	$\pi/3$

Информационные технологии в менеджменте

Математические модели в биологии

№	Модель Мальтуса	Логистическая модель		Система хищник-жертва		
	a	a	K	c	r	d
1.	6	25	100	0,25	0,03	0,07
2.	7	26	101	0,27	0,1	0,05
3.	8	27	102	0,3	0,3	0,07
4.	9	28	103	0,33	0,1	0,05
5.	10	29	104	0,36	0,1	0,2
6.	11	30	105	0,39	0,3	0,07
7.	12	31	106	0,41	0,05	0,06
8.	13	32	107	0,43	0,3	0,07
9.	14	33	108	0,45	0,08	0,09
10.	15	34	109	0,46	0,09	0,3
11.	16	35	110	0,47	0,08	0,05
12.	17	15	111	0,48	0,09	0,3
13.	18	16	112	0,49	0,04	0,08
14.	19	17	113	0,5	0,08	0,3
15.	20	18	114	0,51	0,09	0,09
16.	21	19	115	0,52	0,08	0,3
17.	22	20	116	0,53	0,09	0,06
18.	23	21	117	0,54	0,08	0,3
19.	24	22	118	0,55	0,09	0,06
20.	25	23	119	0,56	0,08	0,3
21.	26	24	120	0,57	0,92	0,064
22.	27	25	121	0,58	0,36	0,09
23.	28	26	122	0,59	0,35	0,067
24.	29	29	123	0,6	0,05	0,4
25.	30	30	124	0,61	0,4	0,07
26.	31	31	125	0,57	0,46	0,06
27.	32	32	126	0,58	0,47	0,07
28.	33	33	127	0,59	0,48	0,09
29.	34	34	128	0,6	0,49	0,3
30.	35	35	129	0,61	0,35	0,05

4. График выполнения курсовой работы

№	Наименование работы	Срок выполнения	Количество потраченного студентом времени (час)
1.	Выдача тем курсовой работы	2 неделя	0
2.	Написание 1 раздела пояснительной записки	3—6 неделя	10
3.	Написание 2 раздела пояснительной записки	7—10 неделя	15

Информационные технологии в менеджменте

4.	Написание 3 раздела пояснительной записки	11—14 неделя	15
5.	Полное оформление пояснительной записки (введение, оглавление, литературы)	15 неделя	9,5
6.	Защита курсовой работы	16—17 неделя	0,5
Итого			50

5. Требования к содержанию пояснительной записки к курсовой работе

Для защиты курсовой работы необходимо представить пояснительную записку, включающую следующие разделы:

Аннотация

Содержание

Введение

Раздел 1. Теоретические сведения о методах и системах компьютерного моделирования

Раздел 2. Компьютерное моделирование процессов в пакете Mathcad

Раздел 3. Компьютерное моделирование процессов в пакете Matlab

Заключение

Список литературы

В аннотации приводится краткое содержание работы.

Во введении необходимо описать значение математического и компьютерного моделирования для проведения исследований в различных областях деятельности.

Каждый раздел разделяется на подразделы, содержание которых определяется заданием на курсовую работу. Названия подразделов студент придумывает самостоятельно, как краткую формулировку темы подраздела.

Результаты проведенных исследований в виде расчетов в Mathcad или Matlab приводятся непосредственно в конце соответствующего подраздела после необходимых теоретических сведений и пояснений.

В заключении необходимо привести анализ результатов, полученных в ходе выполнения курсовой работы, и выводы, сделанные на их основе. Заключение должно содержать только те выводы, которые согласуются с целью работы, и должны быть изложены таким образом, чтобы их содержание было понятно без чтения текста работы.

Список литературы должен содержать перечень библиографических описаний документов (книги, статьи, нормативно-технические документы и т. п.), использованных при выполнении работы. Описание документов в списке следует располагать в порядке появления ссылок на них в тексте пояснительной записки. В тексте пояснительной записки должны быть ссылки на все источники. Ссылки размещаются в квадратных скобках, например [1], [3, 5—6].

6. Требования к оформлению пояснительной записки к курсовой работе

Оформление титульного листа, листа задания на курсовую работу заглавного и последующих листов пояснительной записки к курсовой работе необходимо выполнить в соответствии с формами, приведенными в «СТАНДАРТ ПРЕДПРИЯТИЯ Правила оформления дипломных и курсовых работ (проектов)».

1. Построение документа.

1.1. Нумерацию разделов, рисунков и таблиц необходимо произвести в пределах каждого раздела.

1.2. Отдельная нумерация страниц документа в пределах раздела и подраздела не допускается.

1.3. Содержание документа размещают на отдельной (пронумерованной) странице (страницах) после аннотации, снабжают заголовком «СОДЕРЖАНИЕ», не нумеруют как раздел.

В содержание документа включают номера разделов, подразделов, пунктов и подпунктов, имеющих заголовки, их наименование и номера страниц; номера и наименование (при наличии) приложений.

Наименования, включенные в содержание, записывают строчными буквами. Прописными должны печататься заглавные буквы и аббревиатуры.

1.4. Структурными элементами текста документа являются разделы, подразделы, пункты, подпункты и перечисления.

Раздел — первая ступень деления, обозначенная номером и снабженная заголовком.

Подраздел — часть раздела, обозначенная номером и имеющая заголовок.

Пункт — часть раздела или подраздела, обозначенная номером. Может иметь заголовок.

Подпункт — часть пункта, обозначенная номером, может иметь заголовок.

Абзац — логически выделенная часть текста, не имеющая номера.

Допускается помещать текст между заголовками раздела и подраздела, между заголовками подраздела и пункта.

Внутри подразделов, пунктов и подпунктов могут быть даны перечисления, которые рекомендуется обозначать арабскими цифрами со скобкой: 1), 2) и т. д. Допускается выделять перечисления простановкой дефиса перед текстом.

1.5. Заголовки разделов пишут прописными буквами и размещают симметрично относительно правой и левой границ текста.

Заголовки подразделов записывают с абзаца строчными буквами (кроме первой прописной).

Переносы слов в заголовках не допускаются. Точку в конце заголовка не ставят.

Если заголовок состоит из двух предложений, их разделяют точкой.

Каждый раздел рекомендуется начинать с нового листа.

1.6. Разделы, подразделы, пункты и подпункты следует нумеровать арабскими цифрами с точкой.

Разделы должны иметь порядковый номер (1, 2 и т. д.). В пределах раздела должна быть сквозная нумерация по всем подразделам, пунктам и подпунктам, входящим в данный раздел.

Нумерация подразделов включает номер раздела и порядковый номер подраздела, входящего в данный раздел, разделенные точкой (2.1, 3.1 и т. д.).

Информационные технологии в менеджменте

При наличии пунктов и подпунктов к номеру подраздела после точки добавляют порядковый номер пункта и подпункта (3.1.1, 3.1.1.1 и т. д.).

2. Текст документа.

2.1. Текст документа должен быть кратким, четким, исключая возможность неоднозначного толкования.

2.2. Термины и определения должны быть едиными и соответствовать установленным стандартам, а или общепринятым в научно-технической литературе.

2.3. Для выделения отдельных понятий допускается изменять интервалы между словами, а также печатать отдельные слова или части текста шрифтом, отличным от печати основного текста.

2.4. Если в документе принята особая система сокращения слов или наименований, то в нем должен быть приведен перечень принятых сокращений.

3. Иллюстрации.

3.1. Иллюстрации могут быть расположены в тексте документа и/или в приложениях.

Иллюстрации, если их в данном документе более одной, нумеруют арабскими цифрами в пределах раздела, таким образом, что первая цифра обозначает номер раздела, а вторая — сквозная нумерация в пределах раздела (например, рис.1.2).

В приложениях иллюстрации нумеруются в пределах каждого приложения в порядке, установленном для текста документа.

Ссылки на иллюстрации дают по типу:

рис. 1.2

или

(рис. 1.2).

Ссылки на ранее упомянутые иллюстрации дают с сокращенным словом «смотри», например,

см. рис. 12.

Иллюстрации могут иметь тематический заголовок и подрисуночный текст, поясняющий содержание иллюстрации.

Тематический заголовок (наименование) помещают над иллюстрацией, подрисуночный текст — под ней. Номер иллюстрации помещают под поясняющими данными.

4. Формулы.

4.1. Формулы в документе, если их более одной, нумеруются арабскими цифрами, номер ставят с правой стороны страницы, в скобках, на уровне формулы.

Нумерация формул проводится аналогично нумерации иллюстраций.

Ссылки в тексте на порядковый номер формулы дают в скобках, например: в формуле (1.4).

4.2. Значения символов и числовых коэффициентов, входящих в формулу, должны быть приведены непосредственно под формулой. Значение каждого символа печатают с новой строки в той последовательности, в какой они приведены в формуле. Первая строка расшифровки должна начинаться со слова «где», без двоеточия после него.

4.3. Размерность одного и того же параметра в пределах одного документа должна быть постоянной.

5. Таблицы.

5.1. Цифровой материал для достижения лучшей наглядности и сравнимости показателей, как правило, следует оформлять в виде таблицы.

5.2. Нумерация таблиц проводится аналогично нумерации иллюстраций.

Ссылки в тексте на порядковый номер таблицы дают в виде:

в табл. 1.4.

5.3. Таблица может иметь заголовок, который следует выполнять строчными буквами. Прописными должны печататься заглавные буквы и аббревиатуры.

6. Приложения.

6.1. Иллюстрированный материал, таблицы и текст вспомогательного характера допускается оформлять в виде приложений.

6.2. Каждое приложение должно начинаться с новой страницы с указанием в правом верхнем углу слова «ПРИЛОЖЕНИЕ» прописными буквами и иметь тематический заголовок, который записывают симметрично тексту прописными буквами.

При наличии в документе более одного приложения все приложения нумеруют арабскими цифрами (без знака №), например, ПРИЛОЖЕНИЕ 1, ПРИЛОЖЕНИЕ 2 и т. д.

6.3. Содержание каждого приложения, при необходимости, разбивают на разделы, подразделы, пункты, нумеруемые отдельно по каждому приложению.

6.4. Нумерация страниц документа и приложений, входящих в состав документа, должна быть сквозная.

Иллюстрации и таблицы в приложениях нумеруют в пределах каждого приложения.

6.5. На приложения должны быть даны ссылки в основном тексте документа.

6.6. Все приложения должны быть перечислены в содержании.

Теоретические сведения к курсовой работе



Тема: «Компьютерное моделирование процессов»

1. Понятия «математическое моделирование» и «математическая модель»

С середины XX в. в самых различных областях человеческой деятельности стали широко применять математические методы и ЭВМ. Возникли такие новые дисциплины, как «математическая экономика», «математическая химия», «математическая лингвистика» и т.д., изучающие математические модели соответствующих объектов и явлений, а также методы исследования этих моделей.

Математическая модель — это приближенное описание какого-либо класса явлений или объектов реального мира на языке математики. Процесс построения и изучения математических моделей называется *математическим моделированием*.

Математическое моделирование — это метод познания окружающего мира, исследования объектов, дающий возможность управлять ими.

Все естественные и общественные науки, использующие математический аппарат, по сути, занимаются математическим моделированием: заменяют объект его математической моделью и затем изучают последнюю. Связь математической модели с реальностью осуществляется с помощью гипотез, идеализаций и упрощений.

Математическое моделирование и связанный с ним компьютерный эксперимент незаменимы в тех случаях, когда натурный эксперимент невозможен или затруднен по тем или иным причинам. Например, нельзя поставить натурный эксперимент в истории, чтобы проверить, «что было бы, если бы...» Невозможно проверить правильность той или иной космологической теории. В принципе возможно, но вряд ли разумно, поставить эксперимент по распространению какой-либо болезни, например чумы, или осуществить ядерный взрыв, чтобы изучить его последствия. Однако все это вполне можно сделать на компьютере, построив предварительно математические модели изучаемых явлений.

2. Основные этапы математического моделирования

1) Построение модели. На этом этапе задается некоторый «нематематический» объект — явление природы, конструкция, экономический план, производственный процесс и т.д. При этом, как правило, четкое описание ситуации затруднено. Сначала выявляются основные особенности явления и связи между ними на качественном уровне. Затем найденные качественные зависимости формулируются на языке математики, т.е. строится математическая модель. Это самая трудная стадия моделирования.

2) Решение математической задачи, к которой приводит модель. На этом этапе большое внимание уделяется разработке алгоритмов и численных методов решения задачи на ЭВМ, при помощи которых результат может быть найден с необходимой точностью и за допустимое время.

3) Интерпретация полученных следствий из математической модели. Следствия, выведенные из модели на языке математики, интерпретируются на языке, принятом в данной предметной области.

4) Проверка адекватности модели. На этом этапе выясняется, согласуются ли результаты эксперимента с теоретическими следствиями из модели в пределах определенной точности.

5) Модификация модели. На этом этапе происходит либо усложнение модели, чтобы она была более адекватной действительности, либо ее упрощение ради достижения практически приемлемого решения.

3. Классификация математических моделей

Классифицировать модели можно по разным критериям. Например, по способу представления объекта модели могут быть разделены на структурные и функциональные.

Структурные модели представляют объект как систему со своим устройством и механизмом функционирования. *Функциональные модели* не используют таких представлений и отражают только внешне воспринимаемое поведение (функционирование) объекта.

Также применяется формальная классификация моделей, которая основывается на классификации используемых математических средств. Такая классификация обычно строится в форме дихотомий. Например, один из популярных наборов дихотомий:

- линейные или нелинейные модели;
- сосредоточенные или распределенные системы;
- детерминированные или стохастические;
- статические или динамические;
- дискретные или непрерывные.

Примечание. *Дихотомия* (греч. διχοτομία: διχῆ, «надвое» + τομή, «деление») — это раздвоенность, последовательное деление на две части, не связанные между собой. Способ логического деления класса на подклассы, который состоит в том, что делимое понятие полностью делится на два взаимоисключающих понятия.

Каждая построенная модель является линейной или нелинейной, детерминированной или стохастической, ... Естественно, что возможны и смешанные типы: в одном отношении сосредоточенные (по части параметров), в другом — распределенные модели и т.д.

4. Прямая и обратная задачи математического моделирования

Традиционно выделяют два основных класса задач, связанных с математическими моделями: прямые и обратные.

Прямая задача: структура модели и все ее параметры считаются известными, главная задача — провести исследование модели для извлечения полезного знания об объекте. Какую статическую нагрузку выдержит мост? Как он будет реагировать на динамическую нагрузку (например, на марш роты солдат, или на прохождение поезда на различной скорости), как самолет преодолет звуковой барьер, не развалится ли он от флаттера, — вот типичные примеры прямой задачи.

Примечание. *Флаттер* (от англ. *flutter* — дрожание, вибрация) — это сочетание самовозбуждающихся незатухающих изгибающих и крутящих автоколебаний элементов конструкции летательного аппарата — главным образом крыла самолета либо несущего винта вертолета. Как правило, флаттер проявляется при достижении некоторой критической скорости, зависящей от

Информационные технологии в менеджменте

характеристик конструкции летательного аппарата; возникающий резонанс может привести к его разрушению.

Постановка правильной прямой задачи (задание правильного вопроса) требует специального мастерства. Если не заданы правильные вопросы, то мост может обрушиться, даже если была построена хорошая модель его поведения. Так, в 1879 г. в Великобритании обрушился металлический мост через реку Тей, конструкторы которого построили модель моста, рассчитали его на 20-кратный запас прочности на действие полезной нагрузки, но забыли о постоянно дуящих в тех местах ветрах. И через полтора года он рухнул.

В простейшем случае (когда модель представлена одним уравнением) прямая задача очень проста и сводится к явному решению этого уравнения.

Обратная задача: известно множество возможных моделей, надо выбрать конкретную модель на основании дополнительных данных об объекте. Чаще всего структура модели известна, и необходимо определить некоторые неизвестные параметры. Дополнительная информация может состоять в дополнительных эмпирических данных, или в требованиях к объекту (*задача проектирования*). Дополнительные данные могут поступать независимо от процесса решения обратной задачи (*пассивное наблюдение*) или быть результатом специально планируемого в ходе решения эксперимента (*активное наблюдение*).

Одним из первых примеров эффективного решения обратной задачи с максимально полным использованием доступных данных был построенный И. Ньютоном метод восстановления сил трения по наблюдаемым затухающим колебаниям.

В качестве другого примера можно привести математическую статистику. Задача этой науки заключается в разработке методов регистрации, описания и анализа данных наблюдений и экспериментов с целью построения вероятностных моделей массовых случайных явлений. Т.е. множество возможных моделей ограничено вероятностными моделями. В конкретных задачах множество моделей ограничено сильнее.

5. Компьютерные системы моделирования

Для поддержки математического моделирования разработаны системы компьютерной математики, например, Maple, Mathematica, Mathcad, Matlab, VisSim и др. Они позволяют создавать формальные и блочные модели как простых, так и сложных процессов и устройств и легко менять параметры моделей в ходе моделирования. *Блочные модели* представлены блоками (чаще всего графическими), набор и соединение которых задаются диаграммой модели.

6. Примеры математических моделей

6.1 Математические модели в физике

Движение снаряда. Рассмотрим следующую задачу механики. Снаряд пущен с Земли с начальной скоростью v_0 под углом α к ее поверхности; требуется найти траекторию его движения и расстояние S между начальной и конечной точкой этой траектории.

Пренебрегая размерами снаряда, будем считать его материальной точкой. Введем систему координат xOy , совместив ее начало O с исходной точкой, из которой пущен снаряд, ось x направим горизонтально, а ось y — вертикально (рис. 1).

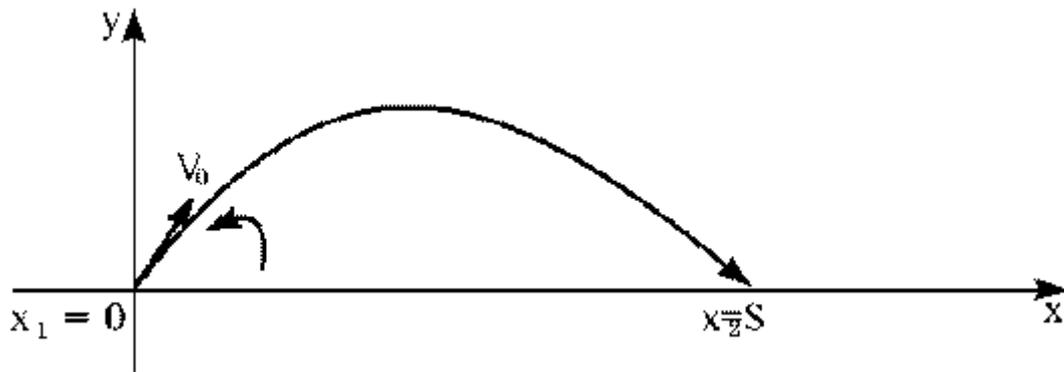


Рис. 1.

Из курса физики известно, что движение снаряда описывается формулами:

$$x = tv_0 \cos \alpha, \quad y = tv_0 \sin \alpha - \frac{gt^2}{2},$$

где t — время; g — ускорение свободного падения.

Эти формулы и дают математическую модель поставленной задачи. Выражая t через x из первого уравнения и подставляя во второе, получим уравнение траектории движения снаряда:

$$y = x \operatorname{tg} \alpha - \frac{x^2 g}{2v_0^2 \cos^2 \alpha}.$$

Эта кривая (парабола) пересекает ось x в двух точках: $x_1 = 0$ (начало

траектории) и $x_2 = S = \frac{v_0^2}{g} \sin 2\alpha$ (место падения снаряда).

При построении этой модели использован ряд предположений: например, считается, что Земля плоская, а воздух и вращение Земли не влияют на движение снаряда.

Гармонический осциллятор. Рассмотрим механическую систему, состоящую из пружины, закрепленной с одного конца, и груза массой m , прикрепленного к свободному концу пружины. Будем считать, что груз может двигаться только в направлении оси пружины (например, движение происходит вдоль стержня). Построим математическую модель этой системы. Будем описывать состояние системы расстоянием x от центра груза до его положения равновесия. Опишем взаимодействие пружины и груза с помощью *закона Гука* $F = -kx$, после чего воспользуемся вторым законом Ньютона, чтобы выразить его в форме дифференциального уравнения:

$$m \frac{d^2 x(t)}{dt^2} = -kx(t).$$

Полученное уравнение описывает математическую модель рассмотренной физической системы. Эта модель называется *гармоническим осциллятором*.

По формальной классификации эта модель линейная, детерминированная, динамическая, сосредоточенная, непрерывная. В процессе ее построения было сделано множество допущений (об отсутствии внешних сил, отсутствии трения, малости отклонений и т.д.), которые в реальности могут не выполняться.

В некотором приближении (скажем, пока отклонение груза от равновесия невелико, при малом трении, в течение не слишком большого времени и при соблюдении некоторых других условий), такая модель достаточно хорошо описывает реальную механическую систему, поскольку отброшенные факторы

оказывают пренебрежимо малое влияние на ее поведение. Однако модель можно уточнить, приняв во внимание какие-то из этих факторов. Это приведет к новой модели, с более широкой (хотя и снова ограниченной) областью применимости.

Впрочем, при уточнении модели сложность ее математического исследования может существенно возрасти и сделать модель фактически бесполезной. Зачастую более простая модель позволяет лучше и глубже исследовать реальную систему, чем более сложная (и, формально, «более правильная»).

Движение груза, подвешенного к концу пружины (уточненная модель гармонического осциллятора). Одним из уточнений модели гармонического осциллятора является дифференциальное уравнение, описывающее движение груза массы m , подвешенного к концу пружины:

$$m \frac{d^2 x(t)}{dt^2} + H \frac{dx(t)}{dt} + kx(t) = f(t),$$

где $x(t)$ — смещение груза от положения равновесия; H — константа, характеризующая силу сопротивления среды; k — коэффициент упругости пружины; $f(t)$ — внешняя сила.

6.2 Математические модели в биологии

При использовании таких моделей предполагаются выполненными следующие гипотезы:

1. Динамика популяции может быть адекватно описана посредством единственной переменной.

2. Численность популяции достаточно велика для того, чтобы использовать детерминированные модели.

Модель Мальтуса. Пусть $N(t)$ — численность популяции в момент времени t , b — ее рождаемость (темп воспроизводства в расчете на одну особь), d — смертность (темп вымирания), $a = b - d$ — мальтузианский параметр, определяемый разностью между рождаемостью и смертностью. Прирост популяции ΔN за время Δt определяется как разница между числом родившихся и умерших особей: $\Delta N = bN\Delta t - dN\Delta t$, так что динамика (скорость роста) численности популяции описывается дифференциальным уравнением:

$$\frac{dN}{dt} = aN, \quad N(0) = N_0,$$

Решением этого уравнения является экспоненциальная функция:

$$N(t) = N_0 e^{at}.$$

Если рождаемость превосходит смертность ($a > 0$), размер популяции неограниченно и очень быстро возрастает. Понятно, что в действительности этого не может происходить из-за ограниченности ресурсов. При достижении некоторого критического объема популяции модель перестает быть адекватной, поскольку не учитывает ограниченность ресурсов.

Логистическая модель (уточненная модель Мальтуса). Уточнением модели Мальтуса может служить логистическая модель, которая описывается дифференциальным уравнением Ферхюльста:

$$\frac{dN}{dt} = aN \left(1 - \frac{N}{K} \right).$$

где $a > 0, K > 0$ — емкость среды, определяющая предельное значение для численности популяции.

Это уравнение интегрируется разделением переменных, и его решение с начальными условиями $N(0) = N_0$ имеет вид:

$$N = \frac{KN_0e^{at}}{K + N_0(e^{at} - 1)}$$

Система хищник-жертва. Простейшей моделью, описывающей взаимодействие двух видов по типу «хищник – жертва», является модель, предложенная В. Вольтерра. В основе этой модели лежат гипотезы о том, что динамика участвующих во взаимодействии видов при их изолированном рассмотрении происходит по линейным законам, в то время как само их взаимодействие осуществляется в соответствии с гипотезой встреч. Эта гипотеза, восходящая к законам химической кинетики, заключается в том, что число встреч, приводящих к пожиранию одного вида другим, пропорционально численности каждого из видов. При этом считается, что фиксированная доля биомассы съеденных жертв, пропорциональная их численности, переходит в биомассу хищника.

Дифференциальные уравнения, моделирующие временную динамику численности двух биологических популяций жертв x и хищников y , имеют вид:

$$\begin{cases} \frac{dx(t)}{dt} = c \cdot x(t) - r \cdot x(t) \cdot y(t), \\ \frac{dy(t)}{dt} = -d \cdot y(t) + r \cdot x(t) \cdot y(t). \end{cases}$$

Предполагается, что жертвы размножаются с постоянной скоростью c , а их численность убывает вследствие того, что их поедают хищники.

Хищники же размножаются со скоростью, пропорциональной количеству пищи (с коэффициентом r), и умирают естественным образом (смертность определяется константой d).

Эта система имеет равновесное состояние, когда число жертв и хищников постоянно. Отклонение от этого состояния приводит к колебаниям численности жертв и хищников, аналогичным колебаниям гармонического осциллятора.

Как и в случае гармонического осциллятора, это поведение не является структурно устойчивым: малое изменение модели (например, учитывающее ограниченность ресурсов, необходимых жертвам) может привести к качественному изменению поведения. Например, равновесное состояние может стать устойчивым, и колебания численности будут затухать. Возможна и противоположная ситуация, когда любое малое отклонение от положения равновесия приведет к катастрофическим последствиям, вплоть до полного вымирания одного из видов. На вопрос о том, какой из этих сценариев реализуется, модель Вольтера ответа не дает, здесь требуются дополнительные исследования.