



ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
УПРАВЛЕНИЕ ЦИФРОВЫХ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ
Кафедра «Автоматизация производственных процессов»

Лабораторная работа
«Применение графического интерфейса
Matlab»
по дисциплине

**«Пакеты прикладных про-
грамм»**

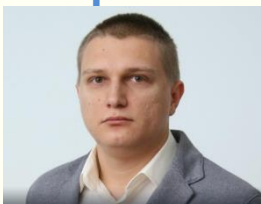
Авторы
Туркин И.А.,
Носачев С.В.,
Зимовнов О.В.,
Лапшин В.П.

Ростов-на-Дону, 2019

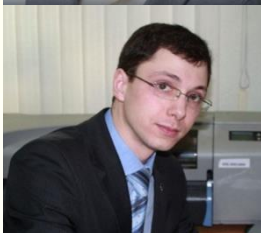
Аннотация

Практикум предназначен для студентов очной, заочной форм обучения направлений 27.03.04 «Управление в технических системах», 15.03.04 «Автоматизация производственных процессов».

Авторы



к.т.н, доцент кафедры «АПП»
Туркин И.А.,



к.т.н., доцент кафедры «АПП»
Носачев С.В.,



к.т.н., доцент кафедры «АПП»
Зимовнов О.В.,



к.т.н., доцент кафедры «АПП»
Лапшин В.П.





Оглавление

Введение	4
1. Цель работы.....	4
2. Создание окон в среде GUIDE	7
3. Программирование интерфейса	20

ВВЕДЕНИЕ

В этом пособии рассказывается о применении пакета MatLab для анализа многомерных данных. В пособии интенсивно используются понятия и методы матричной алгебры – вектор, матрица, и т.п. MatLab является мощным и универсальным средством обработки многомерных данных. Сама структура пакета делает его удобным средством для проведения матричных вычислений. Спектр проблем, исследование которых может, осуществлено при помощи MatLab, охватывает: матричный анализ, обработку сигналов и изображений, нейронные сети и многие другие. MatLab — это язык высокого уровня, имеющий открытый код, что дает возможность опытным пользователям разбираться в запрограммированных алгоритмах. Простой встроенный язык программирования позволяет легко создавать собственные алгоритмы. За много лет использования MatLab создано огромное количество функций и ToolBox (пакетов специализированных средств), которые значительно расширяют функционал пакета.

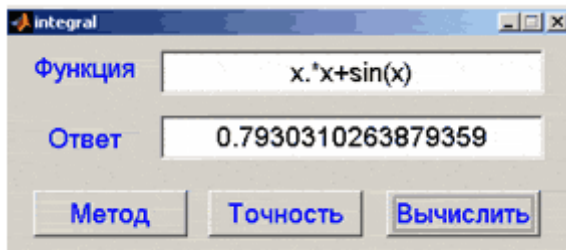
1. ЦЕЛЬ РАБОТЫ

В этой работе рассматривается один из многочисленных способов разработки приложений с графическим интерфейсом пользователя в MATLAB, при котором:

1. основное окно приложения и дочерние окна создаются в среде визуального программирования GUIDE;
2. пишется функция с подфункциями, реализующая совместную работу этих окон и взаимодействие между ними.

В качестве простого примера напишем графический интерфейс к функциям `quad` и `quadl`, предназначенным для численного интегрирования функций одной переменной от 0 до 1 по квадратурным формулам Симпсона и Гаусса-Лобатто, соответственно. На примере этого приложения мы рассмотрим реализацию пунктов 1) и 2).

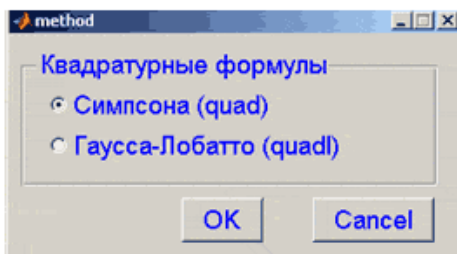
Внешний вид основного окна `integral` приложения приведен ниже на рисунке.



Основное окно `integral` приложения

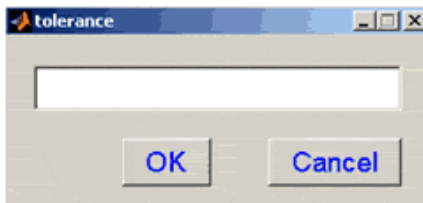
Кнопка `Метод` предназначена для открытия диалогового окна `method`, приведенного ниже на рисунке, в котором можно при помощи переключателей выбрать один из двух методов интегрирования:

- по квадратурным формулам Симпсона;
- по квадратурным формулам Гаусса-Лобатто.



Диалоговое окно `method` для выбора метода интегрирования

Кнопка `Точность` в основном окне приложения предназначена для открытия диалогового окна `tolerance`, приведенного ниже на рисунке, в котором в строке ввода пользователь задает точность интегрирования.



Диалоговое окно `tolerance` для задания точности интегрирования

После нажатия на кнопку Вычислить в основном окне приложения, вычисляется интеграл от функции, заданной в строке ввода Функция основного окна приложения, и результат, полученный при помощи выбранного метода и с заданной точностью выводится в строку Ответ.

Итак, приложение с графическим интерфейсом пользователя состоит из основного окна `integral` и двух дочерних диалоговых окон `method` и `tolerance`, в которых задаются некоторые параметры приложения. Эти диалоговые окна мы сделаем модальными, т.е. после появления одного из них на экране и задания в них нужных параметров необходимо закрыть это окно для продолжения работы с приложением `integral`.

Разумеется, настройки, сделанные в окнах `method` и `tolerance`, должны быть доступны в подфункциях обработки событий элементов интерфейса основного окна. И наоборот, выбранный ранее метод интегрирования и точность должны быть известны при открытии окон `method` и `tolerance` для того, чтобы в окне `method` был установлен переключатель, соответствующий ранее выбранному методу интегрирования, а в строке ввода окна `tolerance` при его открытии была выведена заданная ранее точность.

Для начала в среде GUIDE требуется создать основное окно `integral`. (В разделе Приложения с GUI достаточно подробно рассмотрено создание несложных приложений с графическим интерфейсом в среде GUIDE). В нашем слу-

чае требуется сделать окно с тремя кнопками:

- Метод
- Точность
- Вычислить

и двумя строками ввода, слева от которых размещены текстовые подписи:

- Функция
- Ответ

2. СОЗДАНИЕ ОКОН В СРЕДЕ GUIDE

Сначала мы кратко разберем создание диалоговых окон в среде визуального программирования GUIDE MATLAB для экономии времени читателя и для того, чтобы подчеркнуть необходимые в нашем случае действия. После выполнения в командной строке MATLAB команды `guide`

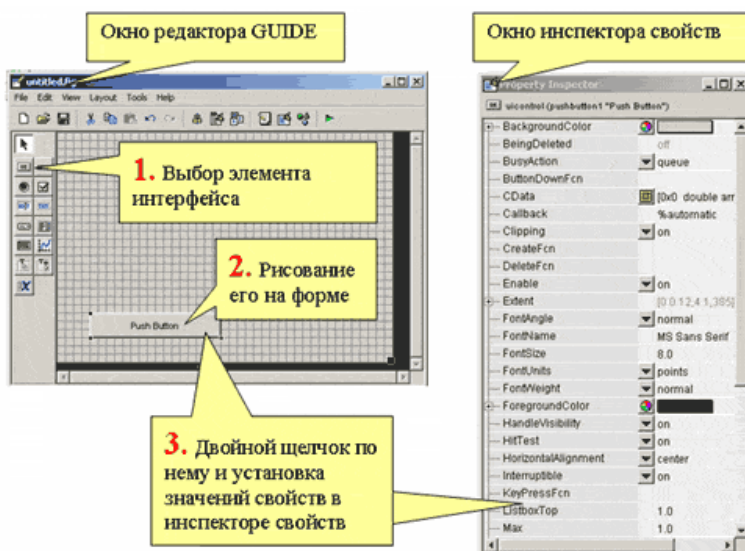
```
>> guide
```

(или нажатии кнопки GUIDE на панели инструментов MATLAB) на экране появляется диалоговое окно GUIDE Quick Start, в котором на вкладке Create New GUI в списке GUIDE Templates требуется выбрать строку Blank GUI (Default) для создания диалогового окна с пустой формы, на которую мы будем добавлять элементы интерфейса, и нажать кнопку ОК. После этого на экране появляется окно редактора GUIDE, содержащее заготовку окна приложения (на ней нанесена сетка).

На панели компонент, размещенной слева по вертикали в окне редактора GUIDE, выбирается нужный элемент интерфейса, дальше он рисуется при помощи мыши на

Пакеты прикладных программ

заготовке окна приложения и делается двойной щелчок по нарисованному элементу интерфейса для изменения его свойств. В результате двойного щелчка открывается окно инспектора свойств, в первом столбике которого приведены названия свойств, а во втором - их значения. Этапы создания элементов интерфейса в редакторе GUIDE и задание значений их свойствам в инспекторе свойств приведены ниже на рисунке.



Этапы создания элементов интерфейса и задания значений их свойствам

Значения либо вводятся в ячейки второго столбика, либо выбираются нажатием на кнопку, расположенную рядом. Если значением свойства является массив, то слева от названия свойства есть кнопка со знаком плюс, щелчок по которой раскрывает массив и позволяет изменить значение его компонент в правом столбике.

На основном окне приложения требуется разместить:

- два текстовых объекта (при помощи кнопки Static Text на панели компонент редактора GUIDE)

- две строки ввода (при помощи кнопки Edit Text на панели компонент редактора GUIDE)
- три кнопки (при помощи кнопки Push Button на панели компонент редактора GUIDE).

Примечания.

1. Если случайно на заготовке окна приложения был нарисован ненужный объект, то его можно удалить, выделив щелчком мыши и нажав клавишу Delete.
2. Нарисованные объекты можно перемещать при помощи мыши и изменять их размеры, потянув курсором мыши за маркеры на границе выделенного объекта.
3. Мы сейчас не будем разбирать все способы и приемы работы с объектами, нарисованными на заготовке окна приложения.

Итак, опишем кратко этапы создания элементов управления в основном окне приложения и те их свойства, которые необходимо задать в редакторе свойств.

Основное окно приложения *integral*

Перед размещением объектов зададим свойства основного окна *integral* приложения. Для этого надо сделать двойной щелчок по заготовке окна в редакторе GUIDE и в окне инспектора свойств установить следующие значения, которые приведены в таблице

Свойства окна <i>integral</i>		
Свойство	Значение	Комментарии
Units	pixels	единицы измерения размеров и положения окна

Position	[400 400 400 152]	координаты левого нижнего угла, ширина, высота (в единицах измерения, заданных свойством Units), для их задания следует щелкнуть по кнопке со знаком плюс слева от названия свойства Position
Name	integral	текст заголовка окна
Tag	win_main	тэг окна приложения, который понадобится для получения указателя на него в функциях обработки событий

После задания свойств окна можно перейти к размещению на заготовке окна приложения основных элементов интерфейса.

Размещение элементов интерфейса в окне `integral`

Требуется поместить на заготовку окна приложения два текстовых объекта (текстовые объекты добавляются на заготовку окна приложения при помощи кнопки Static Text на панели компонент редактора GUIDE), предназначенных для вывода текста в окно приложения с надписями Функция и Ответ и рядом с ними две строки ввода (при помощи кнопки Edit Text на панели компонент редактора GUIDE). В следующей таблице приведены свойства текстовых объектов, которые следует изменить для того, чтобы окно приложения выглядело так, как на рисунке Основное окно приложения `integral`.

Примечание

Общие свойства можно задавать в инспекторе свойств, предварительно выделив несколько объектов (в нашем случае два текстовых объекта), удерживая клавишу и затем сделав двойной щелчок мышью по любому из выделен-

ных объектов.

Общие свойства текстовых объектов		
Свойство	Значение	Комментарии
Font Size	14	размер шрифта текста
ForegroundColor	синий	цвет шрифта
Свойства текстового объекта Функция		
String	Функция	надпись на объекте
Свойства текстового объекта Ответ		
String	Ответ	надпись на объекте

Тэги задавать не обязательно (они выбираются автоматически и не понадобятся нам, так как ссылаться на текстовые объекты мы не будем). Размеры и положение в таблице тоже не приведены, поскольку в нашем простом примере они могут быть выбраны при помощи мыши.

Теперь рядом с этими текстовыми объектами поместим две строки ввода, которые дальше будем называть строка ввода Функция и строка ввода Ответ (строки ввода добавляются на заготовку окна приложения при помощи кнопки Edit Text на панели компонент редактора GUIDE). В следующей таблице приведены значения свойств строк ввода Функция и Ответ, которые следует изменить в окне инспектора свойств.

Общие свойства строк ввода		
Свойство	Значение	Комментарии
Font Size	14	размер шрифта текста

String	пустая строка	текст в строке ввода, она должна быть пустой, т.е. надо удалить в инспекторе свойств значение Edit Text свойства String
Свойства текстового объекта Функция		
Tag	edt_fun	тэг строки ввода, который понадобится для получения указателя на нее в функциях обработки событий
Свойства текстового объекта Ответ		
Tag	edt_ans	тэг строки ввода, который понадобится для получения указателя на нее в функциях обработки событий

Размеры и положение строк ввода можно установить при помощи мыши.

Далее разместим три кнопки Метод, Точность и Вычислить (кнопки добавляются на заготовку окна приложения при помощи кнопки Push Button на панели компонент редактора GUIDE). Свойства этих кнопок приведены в таблице ниже.

Общие свойства кнопок		
Свойство	Значение	Комментарии
Font Size	14	размер шрифта текста
ForegroundColor	синий	цвет шрифта
Свойства текстового объекта Метод		
Tag	btn_method	тэг кнопки, который понадобится для получения указателя на нее в функциях обработки событий
String	Метод	надпись на кнопке
Свойства текстового объекта Точность		

Tag	btn_tol	тэг кнопки, который понадобится для получения указателя на нее в функциях обработки событий
String	Точность	надпись на кнопке
Свойства текстового объекта Вычислить		
Tag	btn_calc	тэг кнопки, который понадобится для получения указателя на нее в функциях обработки событий
String	Вычислить	надпись на кнопке

Размеры и положение кнопок можно установить при помощи мыши.

Сохранение основного окна *integral* приложения

По умолчанию, приложение, созданное в среде визуального программирования GUIDE MATLAB, сохраняется в двух файлах:

1. Файл с расширением fig - графическое окно, содержащее добавленные на форму элементы интерфейса;
2. Файл с расширением m - генерируемая в среде GUIDE файл-функция, открывающая окно и содержащая подфункции для обработки событий элементов интерфейса, которые возникают, например, при нажатии на кнопку пользователем. Заголовки этих подфункций и комментарии к ним создаются автоматически.

Как раз в разбираемом в этом разделе способе создания графического интерфейса пользователя автоматически генерируемая файл-функция не нужна, поскольку созданные в среде GUIDE окна мы будем открывать из основной функции и ее подфункций, которые напишем самостоятельно. Поэтому перед сохранением приложения следует отказаться от автоматического создания m-файла. Для этого в редакторе GUIDE в меню Tools следует выбрать пункт GUI

Options..., в появившемся диалоговом окне GUI Options установить переключатель Generate FIG-file only и нажать кнопку ОК.

Примечание

Перед сохранением окон method и tolerance, создание которых описано далее, следует проделать ту же самую операцию для отказа от автоматического создания m-файла и сохранения диалогового окна только в файле с расширением fig.

После этого следует сохранить созданное основное окно приложения integral. Для этого в редакторе GUDE в меню File следует выбрать пункт Save и в появившемся стандартном диалоговом окне сохранения файла выбрать нужную папку и сохранить окно в файле integral.fig (расширение набирать не нужно).

Желательно сразу же сделать выбранную папку текущим каталогом MATLAB, например при помощи кнопки, расположенной слева от раскрывающегося списка Current Directory на панели инструментов основного окна MATLAB.

Создание диалогового окна method

Для создания новой заготовки для диалогового окна method следует нажать кнопку New на панели инструментов редактора GUIDE (или в меню File перейти к пункту New и в появившемся окне вкладке Create New GUI в списке GUIDE Templates требуется выбрать строку Blank GUI (Default) и нажать кнопку ОК).

Перед размещением объектов (панели и переключателей) зададим свойства диалогового окна method нашего приложения. Для этого надо сделать двойной щелчок

по заготовке окна в редакторе GUIDE и в окне инспектора свойств установить следующие значения, которые приведены в следующей таблице

Свойства окна method		
Свойство	Значение	Комментарии
Units	pixels	единицы измерения размеров и положения окна
Position	[520 366 338 167]	координаты левого нижнего угла, ширина, высота (в единицах измерения, заданных свойством Units), для их задания следует щелкнуть по кнопке со знаком плюс слева от названия свойства Position
Name	method	текст заголовка окна
WindowStyle	modal	окно становится модальным, т.е. для продолжения работы с приложением его требуется закрыть
Tag	win_method	тэг окна приложения, который понадобится для получения указателя на него в функциях обработки событий

Теперь можно переходить к расположению в окне панели переключателей Квадратурные формулы и двух переключателей Симпсона (quad) и Гаусса-Лобатто (quadl) на ней. Для создания панели переключателей служит кнопка Button Group на панели компонент редактора GUIDE. В следующей таблице приведены значения свойств панели переключателей, которые следует изменить в окне инспектора свойств.

Свойства панели переключателей Квадратурные формулы

Свойство	Значение	Комментарии
Title	Квадратурные формулы	Название панели
Tag	pnl_method	тэг панели переключателей, который понадобится для получения указателя на него в функциях обработки событий

После размещения панели переключателей и задания ее свойств следует разместить два переключателя при помощи кнопки Radio Button на панели компонент редактора GUIDE. Переключатели следует размещать именно на панели переключателей для того, чтобы они вели себя согласованно в работающем приложении.

Ниже в таблице приведены общие свойства переключателей и отличающиеся свойства каждого из них

Общие свойства переключателей		
Свойство	Значение	Комментарии
Font Size	14	размер шрифта текста
ForegroundColor	синий	цвет шрифта
Свойства переключателя Симпсона (quad)		
Tag	rb_quad	тэг переключателя, который понадобится для получения указателя на нее в функциях обработки событий
String	Симпсона (quad)	надпись на переключателе
Свойства переключателя Гаусса-Лобатто (quadI)		

Пакеты прикладных программ

Tag	rb_quadl	тэг переключателя, который понадобится для получения указателя на нее в функциях обработки событий
String	Гаусса-Лобатто (quadl)	надпись на переключателе

Положени и размеры как панели переключателей, так и самих переключателей можно выбрать при помощи мыши.

Теперь осталось добавить две кнопки ОК и Cancel на подготовку диалогового окна method и сохранить его. Кнопки добавляются так как описано в разделе Размещение элементов интерфейса в окне integral. Свойства кнопок ОК и Cancel диалогового окна method приведены в следующей таблице.

Общие свойства кнопок		
Свойство	Значение	Комментарии
Font Size	14	размер шрифта текста
ForegroundColor	синий	цвет шрифта
Свойства кнопки OK		
Tag	btn_methodOK	тэг кнопки, который понадобится для получения указателя на нее в функциях обработки событий
String	OK	надпись на кнопке
Свойства кнопки Cancel		
Tag	btn_methodCancel	тэг кнопки, который понадобится для получения указателя на нее в функциях обработки событий
String	Cancel	надпись на кнопке

Сохранение диалогового окна `method` выполняется так же, как и сохранение основного окна `integral` приложения. Важно не забыть отказаться от автоматического создания соответствующего окну `m`-файла (см. разд. Сохранение основного окна `integral` приложения). Сохраним диалоговое окно `method` в файле `method.fig` в том же самом каталоге, где находится файл `integral.fig`.

Создадим теперь последнее диалоговое окно `tolerance` нашего приложения.

Создание диалогового окна `tolerance`

Диалоговое окно `tolerance` создается точно таким же способом, как и диалоговое окно `method` (см. разд. Создание диалогового окна `method`). В редакторе `GUIDE` создается новая заготовка для диалогового окна, в инспекторе свойств его свойствам устанавливаются следующие значения, приведенные ниже в таблице

Свойства окна <code>tolerance</code>		
Свойство	Значение	Комментарии
Units	pixels	единицы измерения размеров и положения окна
Position	[520 366 338 167]	координаты левого нижнего угла, ширина, высота (в единицах измерения, заданных свойством Units)), для их задания следует щелкнуть по кнопке со знаком плюс слева от названия свойства Position
Name	tolerance	текст заголовка окна

Пакеты прикладных программ

WindowState	modal	окно становится модальным, т.е. для продолжения работы с приложением его требуется закрыть
Tag	win_tol	тэг окна приложения, который понадобится для получения указателя на него в функциях обработки событий

После установки этих свойств для самого диалогового окна tolerance следует разместить на заготовке для него строку ввода и две кнопки ОК и Cancel. Значения их свойств собраны в следующих таблицах

Общие свойства кнопок		
Свойство	Значение	Комментарии
Font Size	14	размер шрифта текста
ForegroundColor	синий	цвет шрифта
Свойства кнопки OK		
Tag	btn_tolOK	тэг кнопки, который понадобится для получения указателя на нее в функциях обработки событий
String	OK	надпись на кнопке
Свойства кнопки Cancel		
Tag	btn_tolCancel	тэг кнопки, который понадобится для получения указателя на нее в функциях обработки событий
String	Cancel	надпись на кнопке

Свойства строки ввода		
Свойство	Значение	Комментарии
Font Size	14	размер шрифта текста

String	пустая строка	текст в строке ввода, она должна быть пустой, т.е. надо удалить в инспекторе свойств значение Edit Text свойства String
Tag	edt_tol	тэг строки ввода, который понадобится для получения указателя на нее в функциях обработки событий

Созданное диалоговое окно следует сохранить в файле `tol.fig`, предварительно отказавшись от способа хранения в двух файлах: с расширением `fig` и с расширением `m`.

Теперь, когда все три окна `intehral`, `method` и `tolerance` созданы, мы перейдем к программированию файл-функции с подфункциями для того, чтобы заставить интерфейс работать.

3. ПРОГРАММИРОВАНИЕ ИНТЕРФЕЙСА

В целом, идея очень простая. Если требуется отобразить диалоговое окно на экране, то мы делаем это при помощи функции `open` (можно использовать и функцию `openfig`). Функция `open` возвращает указатель на диалоговое окно.

Далее при помощи функции `guihandles` мы получаем указатели на объекты диалогового окна (элементы интерфейса). Функция `guihandles` возвращает структуру, имена полей которой совпадают с тэгами объектов (которые задавались в инспекторе свойств при создании объектов в редакторе среды визуального программирования GUIDE), а сами значения полей являются указателями на соответствующие объекты диалоговых окон.

После получения указателей на объекты диалогового окна для некоторых объектов (кнопок, панели переключателей) указываются функции обработки их событий, возникающих, например, при нажатии пользователем на кнопку, или

выборе переключателя. Сами эти функции являются заранее запрограммированными подфункциями в `m`-файле с основной функцией `integral`, которая открывает основное окно `integral` нашего приложения с графическим интерфейсом пользователя.

Примечание

Для того, чтобы запрограммировать событие, возникающее, например при нажатии на кнопку, указатель которой хранится в `handles.btn` (в поле `btn` структуры `handles`), следует выполнить два приведенных ниже действия:

1. Связать с этой кнопкой подфункцию обработки события (при нажатии пользователем кнопки возникает ее событие `Callback`) при помощи функции `set`
2. `set(handles.btn, 'Callback', @btn_Callback)`

Здесь `btn_Callback` - имя подфункции, которую надо запрограммировать. Это можно делать только тогда, когда объект с указателем `handles.btn` (кнопка) существует, т.е. когда открыто диалоговое окно с этой кнопкой.

3. Запрограммировать в подфункции `btn_Callback` действия, которые должны быть выполнены после нажатия пользователем на кнопку. Заголовок подфункции `btn_Callback` должен иметь следующий вид:
4. `function btn_Callback(src, evt)`

где `src` - указатель на объект, событие которого выполняется, а `evt` - аргумент, который используется при обработке некоторых событий.

При открытии диалоговых окон `method` и `tolerance` в них должны быть сделанные ранее пользователем настройки, т.е. выбран тот же самый переключатель (в окне `method`) и в строке ввода должно находиться введенное ранее пользователем число (в окне `tolerance`). Обмен параметрами между

основным окном приложения и его дочерними окнами производится при помощи структуры `par`, которая является данными основного окна `integral` приложения, она всегда может быть получена в подфункциях при помощи функции `MATLAB guidata`, при необходимости изменена и сохранена обновленной при помощи этой же самой функции.

Для получения структуры с данными `par` следует в качестве входного аргумента функции `guidata` задавать указатель на диалоговое окно (в нашем случае это окно `integral`)

```
par = guidata(handles.win_main)
```

Здесь считается, что указатель на окно `integral` хранится в поле `win_main` структуры `handles`. Для сохранения структуры с данными `par` используется следующий вызов функции `guidata` с двумя входными аргументами - указателем на окно (или на объект, расположенный в окне) и самой структурой:

```
guidata(handles.win_main, par)
```

Эти обращения нам понадобится делать в подфункциях обработки событий элементов интерфейса, расположенных не в основном окне, а в дочерних, поэтому в эти подфункции придется передавать дополнительный аргумент - структуру с указателями на объекты основного окна `integral` нашего приложения.

Подфункции обработки событий по умолчанию имеют два входных аргумента. В подфункции обработки событий удобно передавать дополнительные параметры: структуру указателей на основное окно приложения и на дочерние окна. Это делается в массиве ячеек, соответствующие параметры пишутся в фигурных скобках через запятую. Поэтому при связывании события с обрабатывающей его подфункцией мы будем использовать такой вариант вызова функции `set`, когда значением свойства с названием события

будет массив ячеек, содержащий указатель на подфункцию и дополнительные входные аргументы:

```
set(handles.btn_calc, 'Callback',
    {@btn_calc_Callback, handles})
```

Если подфункция обработки события имеет дополнительный входной аргумент, то тогда ее заголовок выглядит следующим образом:

```
function btn_calc_Callback(src,evt,handles)
```

Соответственно, если мы указали подфункцию обработки события с двумя дополнительными входными аргументами:

```
set(handles_method.btn_methodOK, 'Callback', {@btn_methodOK_Callback, handles, handles_method})
```

то ее заголовок должен выглядеть так:

```
function
btn_methodOK_Callback(src,evt,handles,handles_method)
```

Для программирования нашего приложения с графическим интерфейсом пользователя понадобится основная функция, которую мы назовем `integral` и сохраним в файле `integral.m`, содержащая семь подфункций

- `btn_method_Callback` - подфункция обработки события Callback кнопки Метод основного окна `integral`;
- `btn_tol_Callback` - подфункция обработки события Callback кнопки Точность основного окна `integral`;
- `btn_calc_Callback` - подфункция обработки события Callback кнопки Вычислить основного окна `integral`;
- `btn_methodOK_Callback` - подфункция обработки события Callback кнопки ОК дочернего окна `method`;
- `btn_methodCancel_Callback` - подфункция обработки

события Callback кнопки Cancel дочернего окна method;

- btn_tolOK_Callback - подфункция обработки события Callback кнопки ОК дочернего окна tolerance;
- btn_tolCancel_Callback - подфункция обработки события Callback кнопки Cancel дочернего окна tolerance.

Ниже приведен текст основной функции integral с подфункциями. Функция integral.m, окна integral.fig, method.fig и tol.fig должны быть в одном каталоге, путь к которому известен MATLAB (например в текущем каталоге). Для запуска нашего приложения с графическим интерфейсом достаточно выполнить

```
>> integral
```

В основной функции производятся следующие действия:

1. открывается основное диалоговое окно integral.fig;
2. его указатели записываются в структуру handles (поля структуры совпадают с тэгами объектов, расположенных в диалоговом окне integral);
3. инициализируется структура данных par основного окна integral с двумя полями: method (для запоминания выбранного метода интегрирования) и tol (для запоминания выбранной точности интегрирования), в поле method записывается 'quad', а в поле tol заносится 1e-3, сама структура сохраняется при помощи функции guidata;
4. с событием Callback кнопок Метод, Точность и Вычислить связываются, соответственно, подфункции btn_method_Callback, btn_tol_Callback, btn_calc_Callback, в которые передается дополнительный входной аргумент - структура handles.

В подфункции btn_method_Callback, кото-

рая вызывается при нажатии пользователем на кнопку Метод, выполняются следующие действия:

1. открывается дочернее диалоговое окно `method.fig`;
2. его указатели записываются в структуру `handles_method` (поля структуры совпадают с тэгами объектов, расположенных в диалоговом окне `method`);
3. в переменную `par` записывается структура данных окна `integral`;
4. в зависимости от значения `par.method` (т.е. выбранного метода инегрирования) устанавливается соответствующий переключатель на панели переключателей в окне `method`;
5. с событием Callback кнопок ОК и Cancel связываются, соответственно, подфункции `btn_methodOK_Callback` и `btn_methodCancel_Callback`, в которые передается дополнительный входной аргумент - структура `handles_method`.

В подфункции `btn_tol_Callback`, которая вызывается при нажатии пользователем на кнопку Точность, выполняются следующие действия:

1. открывается дочернее диалоговое окно `tol.fig`;
2. его указатели записываются в структуру `handles_tol` (поля структуры совпадают с тэгами объектов, расположенных в диалоговом окне `tol`);
3. в переменную `par` записывается структура данных окна `integral`;
4. в строку ввода окна `tolerance` записывается установленная ранее точность, значение которой записано в поле `tol` структуры `par`;
5. с событием Callback кнопок ОК и Cancel связываются, соответственно, подфункции `btn_tolOK_Callback`

и `btn_tolCancel_Callback`, в которые передается дополнительный входной аргумент - структура `handles_tol`.

В подфункции `btn_calc_Callback`, которая вызывается при нажатии пользователем на кнопку Вычислить, выполняются следующие действия:

1. считывается введенная пользователем формула из строки ввода. Функция основного диалогового окна `integral`;
2. в переменную `par` записывается структура данных окна `integral`;
3. проверяется, какой метод интегрирования был выбран пользователем в диалоговом окне `method` (в поле `method` структуры `par` записано либо 'quad', либо 'quadl') и в зависимости от этого вызывается либо функция `quad`, либо `quadl`, в которых указывается также точность интегрирования, записанная в `par.tol`;
4. результат преобразуется в строковую переменную `str` и выводится в строку Ответ основного диалогового окна `integral`;

В подфункции `btn_methodOK_Callback`, которая вызывается при нажатии пользователем на кнопку ОК в диалоговом окне `method`, выполняются следующие действия:

1. в переменную `par` записывается структура данных окна `integral`;
2. проверяется, какой метод интегрирования был выбран пользователем переключателями в диалоговом окне `method` и в зависимости от этого в поле `method` структуры `par` заносится либо 'quad', либо 'quadl';
3. сохраняется структура данных `par` окна `integral`;
4. окно `method` закрывается;

В подфункции `btn_methodCancel_Callback`, которая вызывается при нажатии пользователем на кнопку `Cancel` в диалоговом окне `method`, закрывается окно `method` без изменения структуры данных `par` окна `integral`.

В подфункции `btn_tolOK_Callback`, которая вызывается при нажатии пользователем на кнопку `OK` в диалоговом окне `tolerance`, выполняются следующие действия:

1. в переменную `str` заносится содержимое строки ввода окна `tolerance`, т.е. введенная пользователем точность;
2. в переменную `par` записывается структура данных окна `integral`;
3. содержимое строки `str` преобразуется в числовое значение, которое заносится в поле `tol` структуры `par`;
4. сохраняется структура данных `par` окна `integral`;
5. окно `tolerance` закрывается;

В подфункции `btn_tolCancel_Callback`, которая вызывается при нажатии пользователем на кнопку `Cancel` в диалоговом окне `tolerance`, закрывается окно `tolerance` без изменения структуры данных `par` окна `integral`.

```
function integral
% основная функция

% открываем основное окно integral
H=open('integral.fig');
% указатели на объекты основного окна integral за-
писываем в структуру handles
handles=guihandles(H);
% инициализируем структуру данных par основного ок-
на integral
par.method='quad';
par.tol=1e-3;
% сохраняем структуру данных par основного окна
integral
guidata(handles.win_main,par)
```

Пакеты прикладных программ

```

% с событием Callback кнопки Метод связываем под-
функцию btn_method_Callback
set(handles.btn_method, 'Callback', {@btn_method_Call
back, handles})
% с событием Callback кнопки Точность связываем
подфункцию btn_tol_Callback
set(handles.btn_tol, 'Callback', {@btn_tol_Callback, h
andles})
% с событием Callback кнопки Вычислить связываем
подфункцию btn_calc_Callback
set(handles.btn_calc, 'Callback', {@btn_calc_Callback
, handles})

function btn_method_Callback(src, evt, handles)
% подфункция обработки события Callback кнопки Ме-
тод

% открываем диалоговое окно method
h=open('method.fig');
% указатели на объекты окна method записываем в
структуру handles_method
handles_method=guihandles(h);
% получаем структуру данных par основного окна
integral
par=guidata(handles.win_main);
% включаем нужный переключатель в окне method
if isequal(par.method, 'quad')
    % выбраны квадратурные формулы Симпсона
    set(handles_method.rb_quad, 'Value', 1)
else
    % выбраны квадратурные формулы Гаусса-Лобатто
    set(handles_method.rb_quad1, 'Value', 1)
end
% с событием Callback кнопки ОК окна method связы-
ваем подфункцию btn_methodOK_Callback
set(handles_method.btn_methodOK, 'Callback', {@btn_me
thodOK_Callback, handles, handles_method})
% с событием Callback кнопки Cancel окна method
связываем подфункцию btn_methodCancel_Callback
set(handles_method.btn_methodCancel, 'Callback', {@bt
n_methodCancel_Callback, handles_method})

function
btn_methodOK_Callback(src, evt, handles, handles_metho
d)
    
```

Пакеты прикладных программ

```

% подфункция обработки события Callback кнопки ОК
окна method

% получаем структуру данных par основного окна
integral
par=guidata(handles.win_main);
% в зависимости от включенного переключателя запол-
няем поле method структуры par
if get(handles_method.rb_quad,'Value')
    % включен переключатель Симпсона (quad)
    par.method='quad';
else
    % включен переключатель Гаусса_Лобатто (quadl)
    par.method='quadl';
end
% сохраняем структуру данных par основного окна
integral
guidata(handles.win_main,par)
% удаляем диалоговое окно method
delete(handles_method.win_method)

function
btn_methodCancel_Callback(src,evt,handles_method)
% подфункция обработки события Callback кнопки Can-
cel окна method

% удаляем диалоговое окно method
delete(handles_method.win_method)

function btn_tol_Callback(src,evt,handles)
% подфункция обработки события Callback кнопки Точ-
ность

% открываем диалоговое окно tolerance
h=open('tol.fig');
% указатели на объекты окна tolerance записываем в
структуру handles_tolerance
handles_tol=guihandles(h);
% получаем структуру данных par основного окна
integral
par=guidata(handles.win_main)
% записываем в строку ввода окна tolerance значение
точности, которое хранится в поле tol структуры par
set(handles_tol.edt_tol,'String',num2str(par.tol))
% с событием Callback      кнопки ОК окна tolerance
    
```

Пакеты прикладных программ

```

связываем подфункцию btn_tolOK_Callback
set(handles_tol.btn_tolOK, 'Callback', {@btn_tolOK_Callback, handles, handles_tol})
% с событием Callback кнопки Cancel окна tolerance
связываем подфункцию btn_tolCancel_Callback
set(handles_tol.btn_tolCancel, 'Callback', {@btn_tolCancel_Callback, handles_tol})
    
```

```

function
btn_tolOK_Callback(src, evt, handles, handles_tol)
% подфункция обработки события Callback кнопки ОК
окна tolerance
    
```

```

% записываем в строковую переменную str содержимое
строки ввода окна tolerance
str=get(handles_tol.edt_tol, 'String');
% получаем структуру данных par основного окна
integral
par=guidata(handles.win_main);
% преобразуем str в число и записываем в поле tol
структуры par
par.tol=str2num(str);
% сохраняем структуру данных par основного окна
integral
guidata(handles.win_main, par)
% удаляем диалоговое окно tolerance
delete(handles_tol.win_tol)
    
```

```

function
btn_tolCancel_Callback(src, evt, handles_tol)
% подфункция обработки события Callback кнопки Cancel
окна tolerance
    
```

```

% удаляем диалоговое окно tolerance
delete(handles_tol.win_tol)
    
```

```

function btn_calc_Callback(src, evt, handles)
% подфункция обработки события Callback кнопки Вы-
числить
    
```

```

% в строковую переменную funstr записываем содержи-
мое строки ввода Функция основного окна integral
funstr=get(handles.edt_fun, 'String')
% получаем структуру данных par основного окна
integral
    
```

```
par=guidata(handles.win_main);
% в зависимости от выбранного метода интегрирования
используем функции MATLAB quad или quadl
% для вычисления определенного интеграла от 0 до 1
if isequal(par.method,'quad')
    % выбраны квадратурные формулы Симпсона
    I=quad(funstr,0,1,par.tol);
else
    % выбраны квадратурные формулы Гаусса-Лобатто
    I=quadl(funstr,0,1,par.tol);
end
% преобразуем значение интеграла и заносим его в
строковую переменную str
str=num2str(I,16);
% записываем строку str в строку Ответ основного
окна integral
set(handles.edt_ans,'String',str)
```

Приведенный текст функции не содержит

1. проверки на правильность ввода функции в строку ввода Функция основного окна `integral`;
2. проверки на правильность ввода точности в строку ввода окна `tolerance`.

Кроме того, можно было выводить в строку ответ не 16 цифр, а в соответствии с заданной точностью, или вводить функцию без поэлементных операций, заменив потом символы `*`, `/` и `^` на поэлементные `.*`, `./` и `.^`. при помощи функции `strrep`, но все эти улучшения увеличили бы объем `m`-файла.

Список литературы

В.И.Горбаченко "Вычислительная линейная алгебра с примерами на MATLAB. СПб.: БХВ-Петербург, 2011. – 320с. (http://www.bhv.ru/books/full_contents.php?id=189162)

Шампайн Л. Ф., Гладвел И., Томпсон С. Решение обыкновенных дифференциальных уравнений с использованием MATLAB: Учебное пособие. 1-е изд. СПб.: Лань, 2009, 304 с.

В.Ф. Худяков, В.А. Хабuzов Моделирование источников вторичного электропитания в среде MATLAB 7.x: учебное пособие. СПб.: ГУАП, 2008, 382 с.

И.Ф. Цисарь, В.Г. Нейман. Компьютерное моделирование экономики. М.: ДИАЛОГ-МИФИ, 2008.-384с.

И.Ф. Цисарь. MATLAB Simulink. Компьютерное моделирование экономики. М.: Солон-Пресс, 2008.-256с.

Чарльз Генри Эдвардс , Дэвид Э. Пенни. Дифференциальные уравнения и краевые задачи: моделирование и вычисление с помощью Mathematica, Maple и MATLAB. 3-е издание. Киев.: Диалектика-Вильямс, 2007. ISBN 978-5-8459-1166-7.

И. Черных. Моделирование электротехнических устройств в MATLAB, SimPowerSystems и Simulink. М.: ИД Питер, 2007, 288 с. ISBN 978-5-388-00020-0.

С.П. Иглин Теория вероятностей и математическая статистика на базе MATLAB. Харьков: НТУ "ХПИ", 2006,-612 стр.

С.Д.Штовба Проектирование нечетких систем средствами MATLAB. М.: Горячая Линия - Телеком, 2007,-288 стр. ISBN: 5-93517-359-X.