



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)**

Кафедра «Информационные системы в строительстве»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по дисциплине

ПРИКЛАДНАЯ МАТЕМАТИКА

Ростов-на-Дону
ДГТУ
2024

Составитель: А.А. Ляпин

Методические указания для выполнения контрольной работы по дисциплине «Прикладная математика» для обучающихся заочной формы / сост. А.А. Ляпин – Ростов-на-Дону: Донской государственный технический университет, 2024. – 19 с.

Содержат краткую теорию по изучаемым вопросам, примеры применения математических методов для решения прикладных задач линейной алгебры, теории дифференциальных уравнений и математической физики. Приведены варианты индивидуальных заданий.

Предназначены для обучающихся магистратуры направления подготовки 09.04.02 заочной формы обучения

УДК 004.02

Ответственный за выпуск: зав. кафедрой «Информационные системы в строительстве» д-р ф.-м. наук, профессор А.А. Ляпин

Издательский центр ДГТУ
Адрес университета и полиграфического предприятия:
344000, г. Ростов-на-Дону, пл. Гагарина, 1

© Донской государственный технический университет, 2024

СОДЕРЖАНИЕ

1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ КОНТРОЛЬНОЙ РАБОТЫ	4
2. РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ.....	5
3. СОБСТВЕННЫЕ ЗНАЧЕНИЯ МАТРИЦ. СИНГУЛЯРНОЕ РАЗЛОЖЕНИЕ МАТРИЦЫ.....	9
4. КРАЕВЫЕ ЗАДАЧИ ДЛЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ	12
5. РАЗНОСТНЫЕ МЕТОДЫ РЕШЕНИЯ УРАВНЕНИЙ В ЧАСТНЫХ ПРОИЗВОДНЫХ.....	16

1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ КОНТРОЛЬНОЙ РАБОТЫ

Задания контрольной работы выполняются по индивидуальным вариантам. Вариант соответствует последней цифре в зачетке обучающегося: цифра 1 – вариант 1, ..., цифра 0 – вариант 10.

Отчет по контрольной работе оформляется в виде электронного документа MS Word. Содержит:

- титульный лист;
- постановку задачи, незначительные сведения из теории, код программы или скриншоты выполнения в рекомендованной программе на компьютере, результаты вычислений и сформулированные выводы по каждой задаче;
- список использованной литературы, включая интернет-источники.

Отчет распечатывается и предоставляется на кафедру «Кибербезопасность информационных систем» для проверки и оценивания.

2. РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ

ЦЕЛЬ РАБОТЫ:

усвоить численные методы решения систем линейных алгебраических уравнений (СЛАУ) и обращения матриц, осуществить их практическую реализацию.

ЗАДАНИЕ:

1. Решить СЛАУ $\mathbf{Ax} = \mathbf{b}$ в системе Matlab и на языке Python с использованием библиотеки numpy в соответствии с индивидуальным заданием.
2. Вычислить элементы обратной матрицы $\mathbf{A}^{-1}$. Найти вектор невязок решения СЛАУ $\boldsymbol{\eta} = \mathbf{Ax} - \mathbf{b}$.
3. Вычислить число обусловленности матрицы $\mathbf{A}$.
4. Дать анализ устойчивости и погрешности решения данной СЛАУ.

КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ И ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Решение СЛАУ $\mathbf{Ax} = \mathbf{b}$ в системе Matlab можно осуществить с использованием команды $[\mathbf{x}, r] = \text{linsolve}(\mathbf{A}, \mathbf{b})$. Здесь r – возвращаемое число обусловленности матрицы $\mathbf{A}$. Аналогично на языке Python применим команду: $\mathbf{x} = \text{numpy.linalg.solve}(\mathbf{A}, \mathbf{b})$. При использовании процедуры обращения системы на других языках можно воспользоваться кроссплатформенной библиотекой Alglib (см. <https://www.alglib.net/docs.php>).

Для невырожденной квадратной матрицы системы уравнений можно единственным образом построить ей обратную $\mathbf{A}^{-1}$. Для этого используем команды:

$\mathbf{A1} = \text{inv}(\mathbf{A})$ - Matlab

$\mathbf{A1} = \text{numpy.linalg.inv}(\mathbf{A})$ - Python

Число обусловленности матрицы СЛАУ. Для получения этой характеристики определяется норма матрицы $\mathbf{A}$, которая может быть найдена по одной из формул:

$$\|\mathbf{A}\|_1 = \max_{1 \leq i \leq N} \sum_{j=1}^N |a_{ij}|, \quad \|\mathbf{A}\|_3 = \max_{1 \leq j \leq N} \sum_{i=1}^N |a_{ij}|,$$

$$\|\mathbf{A}\|_2 = \left(\sum_{i=1}^N \sum_{j=1}^N a_{ij}^2 \right)^{1/2} \quad \text{или} \quad \|\mathbf{A}\|_2 = \max_{1 \leq j \leq N} \sqrt{\nu_j(\mathbf{A}^T \mathbf{A})},$$

где $\nu_j(\mathbf{A}^T \mathbf{A})$ - собственные числа матрицы $\mathbf{A}^T \mathbf{A}$, а $\mathbf{A}^T$ - транспонированная матрица к матрице $\mathbf{A}$.

Числом обусловленности матрицы $\mathbf{A}$ называется величина $\gamma(\mathbf{A}) = \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\|$. На практике, если $\gamma(\mathbf{A})$ большая ($> \gamma_*$), то матрица является плохо обусловленной, если $\gamma(\mathbf{A})$ - невелика ($1 < \gamma(\mathbf{A}) < \gamma_*$), то хорошо обусловленной. Критическое значение γ_* можно оценить как $\frac{10^m}{N^{5/2}} \cdot \varepsilon$, где N - порядок СЛАУ, ε - требуемая точность получения результата, m - число значащих цифр в представлении действительных чисел в программе.

Устойчивость решения СЛАУ. Число обусловленности является важной характеристикой для анализа устойчивости ее решения.

Рассматриваемая СЛАУ $\mathbf{A}\mathbf{x} = \mathbf{b}$ при условии, что $\det \mathbf{A} \neq 0$, имеет единственное решение. Однако при решении СЛАУ, как правило, производится округление чисел, и при этом в вычислительный процесс вносятся погрешности (ошибки округления). Такие погрешности вычислений будем интерпретировать как погрешность правой части $\mathbf{b}$, а само решение $\mathbf{x}$ считать точным. Устойчивость решения к малым его возмущениям $\mathbf{r}$ характеризуется возмущенной СЛАУ $\mathbf{A}(\mathbf{x} + \mathbf{r}) = \mathbf{b} + \boldsymbol{\eta}$ по отношению к исходной $\mathbf{A}\mathbf{x} = \mathbf{b}$.

Рассмотрим СЛАУ $\mathbf{A}(\mathbf{x} + \mathbf{r}) = \mathbf{b} + \boldsymbol{\eta}$, где $\boldsymbol{\eta}$ - погрешность правой части (в нашем случае вектор невязок). Тогда, учитывая, что $\mathbf{A}\mathbf{x} = \mathbf{b}$, приходим к выводу, что $\mathbf{A}\mathbf{r} = \boldsymbol{\eta}$. Устойчивость решения будем характеризовать отношением относительной погрешности решения $\varepsilon = \frac{\|\mathbf{r}\|}{\|\mathbf{x}\|}$ к относительной

погрешности правой части $\delta = \frac{\|\eta\|}{\|b\|}$. Следовательно

$$\frac{\varepsilon}{\delta} = \frac{\|r\| \cdot \|b\|}{\|x\| \cdot \|\eta\|} = \frac{\|Ax\| \cdot \|A^{-1}\eta\|}{\|x\| \cdot \|\eta\|} < \|A\| \cdot \|A^{-1}\| = \gamma(A)$$

Вычисление числовых характеристик матриц в пакете MatLab реализуется функциями:

COND - число обусловленности матрицы

NORM - нормы векторов и матриц

RANK - ранг матрицы

DET - определитель матрицы

Вычисление числовых характеристик матриц в Python определяется набором команд библиотеки numpy:

linalg.eig(a) - собственные значения и собственные векторы;

linalg.norm(x[, ord, axis]) - норма вектора или оператора;

linalg.cond(x[, p]) - число обусловленности;

linalg.det(a) - определитель.

Индивидуальные задания:

$$1. \quad A = \begin{pmatrix} 2.3 & 23.25 & 19.16 & 3.29 \\ 31.5 & 23.38 & -33.4 & -1.96 \\ -1.33 & -9.55 & 22.64 & 2.74 \\ -12.36 & 29.45 & -5.14 & -16.28 \end{pmatrix} \quad b = \begin{pmatrix} 14.5 \\ -4.68 \\ -6.21 \\ 2.9 \end{pmatrix}$$

$$2. \quad A = \begin{pmatrix} -2.99 & 45.69 & -27.8 & -27.71 \\ 34.65 & 27.47 & -4.58 & -25.96 \\ -2.85 & -22.29 & -20.72 & 4.74 \\ -24.72 & 62.25 & -9.3 & -23.28 \end{pmatrix} \quad b = \begin{pmatrix} -8.05 \\ 0.4 \\ -12.45 \\ -23.57 \end{pmatrix}$$

$$3. \quad A = \begin{pmatrix} -3.45 & 7.65 & -30.36 & -17.71 \\ -3.15 & -1.45 & 22.1 & -15.96 \\ -0.57 & 11.73 & -25.6 & -5.26 \\ -20.6 & 20.05 & -17.06 & -9.28 \end{pmatrix} \quad b = \begin{pmatrix} -15.45 \\ 3.28 \\ -10.09 \\ -23.35 \end{pmatrix}$$

$$\begin{array}{ll}
4. & \mathbf{A} = \begin{pmatrix} -0.46 & 28.11 & -5.08 & -10.71 \\ 44.1 & 32.54 & -25.32 & -17.96 \\ -1.71 & -30.84 & -4.44 & 11.74 \\ 2.06 & 29.15 & 10.82 & -13.28 \end{pmatrix} & \mathbf{b} = \begin{pmatrix} 4.3 \\ -7.16 \\ 0.73 \\ -0.78 \end{pmatrix} \\
5. & \mathbf{A} = \begin{pmatrix} 2.76 & 4.59 & 23.88 & 11.29 \\ -44.1 & -30.14 & 11.92 & 24.04 \\ -0.19 & -1.06 & 24.44 & 1.74 \\ -2.06 & 4.35 & -1.02 & 12.72 \end{pmatrix} & \mathbf{b} = \begin{pmatrix} 5.7 \\ 14.76 \\ -1.03 \\ 9.68 \end{pmatrix} \\
6. & \mathbf{A} = \begin{pmatrix} 2.53 & 48.57 & 20.2 & -3.71 \\ -28.35 & -15.93 & 3.26 & 18.04 \\ -2.85 & -54.24 & 20.68 & 19.74 \\ 6.18 & 47.25 & 20.7 & 11.72 \end{pmatrix} & \mathbf{b} = \begin{pmatrix} 6.95 \\ 20.4 \\ 2.55 \\ 12.79 \end{pmatrix} \\
7. & \mathbf{A} = \begin{pmatrix} -1.61 & -13.44 & -13.52 & -2.71 \\ -34.65 & -24.83 & 31.38 & 2.04 \\ 0.76 & 2.11 & -16.2 & 2.74 \\ 10.3 & -18.4 & 6.08 & 15.72 \end{pmatrix} & \mathbf{b} = \begin{pmatrix} -11.95 \\ 7.96 \\ 5.12 \\ -1.23 \end{pmatrix} \\
8. & \mathbf{A} = \begin{pmatrix} -3.45 & 7.65 & -30.36 & -17.71 \\ 22.05 & 15.91 & 6.1 & -23.96 \\ -0.57 & -30.87 & -34.4 & 14.74 \\ 20.6 & 0.05 & 22.94 & 2.72 \end{pmatrix} & \mathbf{b} = \begin{pmatrix} -11.85 \\ -4.72 \\ 9.91 \\ -3.35 \end{pmatrix} \\
9. & \mathbf{A} = \begin{pmatrix} 0 & -3.15 & 0.12 & 1.29 \\ -25.2 & -17.6 & 16 & 6.04 \\ 0.19 & -13.85 & -3.52 & 8.74 \\ 16.48 & -11.35 & 15.02 & 15.72 \end{pmatrix} & \mathbf{b} = \begin{pmatrix} -3.6 \\ 7.24 \\ 8.03 \\ 8 \end{pmatrix} \\
10. & \mathbf{A} = \begin{pmatrix} 0.69 & 6.66 & 5.76 & 1.29 \\ -22.05 & -14.71 & 9.98 & 8.04 \\ -0.38 & 17.05 & 10.84 & -7.26 \\ -22.66 & 17.7 & -20.04 & -4.28 \end{pmatrix} & \mathbf{b} = \begin{pmatrix} -0.15 \\ 8.52 \\ -11.06 \\ -8.33 \end{pmatrix}
\end{array}$$

3. СОБСТВЕННЫЕ ЗНАЧЕНИЯ МАТРИЦ. СИНГУЛЯРНОЕ РАЗЛОЖЕНИЕ МАТРИЦЫ

ЦЕЛЬ РАБОТЫ:

усвоить методы определения собственных значений матриц, сингулярных разложений матриц и их использования в задачах линейной алгебры и системах искусственного интеллекта

ЗАДАНИЕ:

1. Найти собственные значения и собственные векторы заданной квадратной матрицы **A** из предыдущей лабораторной работы.
2. Найти левые сингулярные числа и векторы и правые сингулярные числа и векторы матрицы **B** размерности 3 x 4
3. Найти сингулярное разложение матрицы **B**. Сравнить полученные матрицы с матрицами левых и правых сингулярных векторов
4. Построить псевдообратную матрицу
5. С использованием сингулярного разложения и построения обратной (псевдообратной) матрицы решить систему линейных уравнений $\mathbf{Ax} = \mathbf{b}$ из предыдущей лабораторной работы
6. Задание выполняется в системе Matlab или на языке Python

КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ И ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Число λ называется *собственным значением* квадратной матрицы **A**, если существует ненулевой вектор $\mathbf{x} = \{x_1, x_2, \dots, x_N\}^T \in R^N$, такой что выполняется равенство $\mathbf{Ax} = \lambda\mathbf{x}$.

Вектор **x** называется *собственным вектором* матрицы **A**.

Если матрица симметрична, то есть $\mathbf{A} = \mathbf{A}^T$, то все собственные значения матрицы **A** вещественны, и существует ортонормированный базис, составленный из собственных векторов этой матрицы. Данная матрица может соответствовать матрице самосопряженного оператора, действующего в евклидовом пространстве R^N .

Сингулярные числа и сингулярные векторы. Неотрицательное вещественное число σ называется сингулярным числом матрицы $\mathbf{B}$ размерности $M \times N$, если существуют два вектора единичной длины $\mathbf{u} \in R^M$, $\mathbf{v} \in R^N$ такие, что:

$$\mathbf{B}\mathbf{v} = \sigma\mathbf{u}, \quad \mathbf{B}^T\mathbf{u} = \sigma\mathbf{v}$$

Такие векторы $\mathbf{u}$ и $\mathbf{v}$ называются, соответственно, левым сингулярным вектором и правым сингулярным вектором, соответствующим сингулярному числу σ .

При этом левые сингулярные векторы матрицы $\mathbf{B}$ — это собственные векторы матрицы $\mathbf{B}\mathbf{B}^T$, правые сингулярные векторы матрицы $\mathbf{B}$ — это собственные векторы матрицы $\mathbf{B}^T\mathbf{B}$.

Сингулярным разложением матрицы $\mathbf{B}$ размерности $M \times N$ является разложение следующего вида

$$\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*,$$

где $\mathbf{\Sigma}$ — матрица размерности $M \times N$ с неотрицательными элементами, у которой элементы, лежащие на главной диагонали — это сингулярные числа, а все элементы, не лежащие на главной диагонали, нулевые, матрицы $\mathbf{U}$ — порядка M , $\mathbf{V}$ — порядка N — это две унитарные матрицы, состоящие из левых и правых сингулярных векторов соответственно. $\mathbf{V}^*$ — эрмитово-сопряженная к $\mathbf{V}$.

Матрица $\mathbf{B}^+$ называется **псевдообратной матрицей** для матрицы $\mathbf{B}$, если она удовлетворяет следующим критериям:

1. $\mathbf{B}\mathbf{B}^+\mathbf{B} = \mathbf{B}$
2. $\mathbf{B}^+\mathbf{B}\mathbf{B}^+ = \mathbf{B}^+$
3. $(\mathbf{B}\mathbf{B}^+)^* = \mathbf{B}\mathbf{B}^+$
4. $(\mathbf{B}^+\mathbf{B})^* = \mathbf{B}^+\mathbf{B}$

Если известно сингулярное разложение матрицы $\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$, то

$$\mathbf{B}^+ = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^*,$$

Для диагональной матрицы, такой как $\mathbf{\Sigma}$, псевдообратная $\mathbf{\Sigma}^+$ получается из неё заменой каждого ненулевого элемента на диагонали на обратный к нему.

Если $\mathbf{A}$ — невырожденная квадратная, то $\mathbf{A}^{-1} = \mathbf{A}^+$, а следовательно с ее помощью можно решить СЛАУ $\mathbf{A}\mathbf{x} = \mathbf{b}$ в виде $\mathbf{x} = \mathbf{A}^+\mathbf{b}$

Некоторые функции библиотеки numpy:

<code>Bt = B.transpose()</code>	транспонирование матрицы B
<code>lam, vect = np.linalg.eig(A)</code>	нахождение собственных чисел lam и собственных векторов vect матрицы A
<code>U, S, V = np.linalg.svd(B)</code>	сингулярное разложение матрицы B
<code>C = np.dot(A, B)</code>	скалярное произведение матриц A и B
<code>Sig1 = np.append(Sig, Nu, axis=1)</code>	добавление столбца Nu справа с матрице Sig
<code>Diag = np.diag(1./S)</code>	диагональная матрица Diag с элементами из вектора S на главной диагонали

Индивидуальные задания:

Зададим матрицу размерности 3 x 4 вида

$$B = \begin{pmatrix} 1 & -2 * num & 3 & 1 \\ -2 * num & 1 & 2 & 2 \\ 3 & 2 & 4 & 3 \end{pmatrix}, \quad \text{где } num - \text{номер варианта по}$$

последней цифре в зачетке.

- С использованием операции транспонирования матрицы и вычисления собственных значений найти левые сингулярные числа и векторы и правые сингулярные числа и векторы.
- Найти сингулярное разложение матрицы **B**. Сравнить полученные матрицы с матрицами левых и правых сингулярных векторов.
- Сформировать матрицу Σ , добавив к диагональной матрице с сингулярными числами столбец нулей справа
- Проверить правильность сингулярного разложения и обратного к нему:
 $\Sigma = U^* B V$
- Построить псевдообратную матрицу. Проверить ее свойство

$$B B^+ B = B$$

- С использованием сингулярного разложения и построения обратной матрицы (псевдообратной) решить систему линейных уравнений из предыдущей лабораторной работы

4. КРАЕВЫЕ ЗАДАЧИ ДЛЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

ЦЕЛЬ РАБОТЫ:

освоить методы численного решения краевой задачи для обыкновенного дифференциального уравнения (ОДУ) и осуществить их практическую реализацию

ЗАДАНИЕ:

1. Разработать и реализовать алгоритмы решения краевой задачи для линейного ОДУ второго порядка

$$y'' + p(x) y' + g(x) y = f(x) \quad (1)$$

с граничными условиями

$$y(a) = A, \quad y(b) = B \quad (2)$$

методами пристрелки и конечных разностей

2. Задание выполняется в системе Matlab **или** на языке Python

КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ И ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:

1. Метод пристрелки.

Для разработки алгоритма метода пристрелки решения краевой задачи для линейного ОДУ второго порядка (1),(2) ее необходимо заменить последовательным решением двух задач Коши ($k = 1, 2$) для уравнения (1) с начальными условиями в точке $x = a$:

$$y(a) = A, \quad y'(a) = \alpha_k, \quad (3)$$

где α_k - произвольные не равные друг другу числа.

При решении задачи Коши для уравнения (1) с начальными условиями (3) используется один из пошаговых методов, например метод Рунге-Кутты. Для этого уравнение второго порядка (1) с помощью замен $y = y_1, \quad y' = y_2$ сводится к решению системы двух линейных дифференциальных уравнений вида

$$\begin{cases} y_1' = y_2 \\ y_2' = -p(x)y_2 - g(x)y_1 + f(x) \end{cases} \quad (4)$$

с начальными условиями $y_1(a) = A, \quad y_2(a) = \alpha_k$.

В общем случае подобная система

$$\begin{cases} y_1' = f_1(x, y_1, y_2) \\ y_2' = f_2(x, y_1, y_2) \end{cases} \quad (5)$$

с начальными условиями $y_1(x_0) = y_{10}$, $y_2(x_0) = y_{20}$ может быть записана в векторной форме

$$y' = f(x, y) \text{ при } y(x_0) = y_0, \text{ где } y_0 = (y_{10}, y_{20}) \quad (6)$$

В результате решения двух задач Коши для ОДУ (1) с начальными условиями (3) на равномерной сетке $\omega_h: x_j = x_0 + jh, j = 1, 2, \dots, N$, $x_0 = a, x_N = b$ в интервале интегрирования получим два решения $y_1(x_j)$ и $y_2(x_j)$.

Пусть в конечной точке отрезка интегрирования $x_N = b$ получились значения $y_1(b) = \beta_1$ и $y_2(b) = \beta_2$. В этом случае решение линейной краевой задачи (1),(2) дается формулой

$$y(x_j) = \frac{1}{\beta_1 - \beta_2} \left[(B - \beta_2)y_1(x_j) - (B - \beta_1)y_2(x_j) \right] \quad (7)$$

2. Для алгоритмизации конечно-разностного метода решения краевой задачи (1),(2) уравнения второго порядка (2) заменяются конечно-разностной аппроксимацией на равномерной сетке $\omega_h: x_j = x_0 + jh, j = 1, 2, \dots, N$, $x_0 = a, x_N = b$ вида

$$\begin{aligned} \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p_i \frac{y_{i+1} - y_{i-1}}{2h} + g_i y_i &= f_i \\ i = 1, 2, \dots, N-1; \quad y_0 = y(a) = A, \quad y_N = y(b) = B, \\ p_i &= p(x_i), \quad g_i = g(x_i), \quad f_i = f(x_i). \end{aligned} \quad (8)$$

После этого решение краевой задачи (1),(2) переходит в задачу решения системы линейных алгебраических уравнений (8) для определения y_i .

3. Оценка допущенной погрешности осуществляется по правилу Рунге согласно формуле

$$\Delta_h = y - y_{h/2} \approx \frac{y_{h/2} - y_h}{2^k - 1}$$

и условию, что $|\Delta_h| < \varepsilon$.

При этом для метода пристрелки данная оценка закладывается при решении каждой из задач Коши.

Для конечно-разностного метода условие $|\Delta_h| < \varepsilon$ проверяется в какой-либо из средних точек интервала $[a, b]$. При невыполнении условия последний шаг разбиения $h/2$ делится пополам, и процесс поиска решения повторяется.

ВАРИАНТЫ ИНДИВИДУАЛЬНЫХ ЗАДАНИЙ

Вариант	Уравнение, краевые условия
1	$y'' + (c \cos 2x + d)y = 0, \quad y(a) = 0.5, \quad y(b) = -0.5$
2	$y'' + cy' + (de^x + 1)y = 0, \quad y(a) = 1.2, \quad y(b) = 3.8$
3	$y'' + c \operatorname{tg}(x) y' + dy = 0, \quad y(a) = 2.3, \quad y(b) = 1.5$
4	$xy'' + (d - x) y' - cy = 0, \quad y(a) = 0.25, \quad y(b) = 0.5$
5	$x^2 y'' + xy' + (x^2 - c^2)y = \sin dx, \quad y(a) = -1.24, \quad y(b) = 0.51$
6	$(x^2 + 1)y'' + 4xy' + 2cy = 2d(\cos x - x), \quad y(a) = 0.8, \quad y(b) = 8$
7	$y'' \cos^2 x + cy' \sin 2x + dy = 0, \quad y(a) = 2.2, \quad y(b) = -0.5$
8	$y'' \sin 2x - cy' \cos 2x + 2dy \sin 2x = 0, \quad y(a) = 0.15, \quad y(b) = 8.7$
9	$y'' + 2cy' \operatorname{ctg} cx + (d^2 - c^2)y = 0, \quad y(a) = -1.2, \quad y(b) = 1.31$
10	$(x^2 - 1)y'' + cxy' + dy = 0, \quad y(a) = 4.56, \quad y(b) = 1$

Набор констант	Значения констант
1	$a = 0.5, b = 2.5, c = 2, d = 5$

5. РАЗНОСТНЫЕ МЕТОДЫ РЕШЕНИЯ УРАВНЕНИЙ В ЧАСТНЫХ ПРОИЗВОДНЫХ

ЦЕЛЬ РАБОТЫ:

усвоить алгоритм численного решения задачи Дирихле для уравнения Лапласа методом сеток и осуществить его практическую реализацию.

ЗАДАНИЕ:

1. Разработать алгоритм численного решения методом сеток задачи Дирихле для уравнения Лапласа

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad (1)$$

в прямоугольной области ABCD с вершинами A(0;0), B(0;1), C(1;1), D(1;0), на границе Γ которой задана искомая функция

$$u(x, y)|_{\Gamma} = f(x, y). \quad (2)$$

Алгоритм представить в виде блок-схемы.

2. Реализовать алгоритм решения задачи в виде программы на алгоритмическом языке Python.

3. Задание является дополнительным и в отчет по контрольной работе может не входить.

КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ И ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Сведение задачи Дирихле к разностному уравнению.

Задачи Дирихле для уравнения Лапласа (1) в плоской области $D \in R^2$ с условием на границе области вида (2) возникают во многих задачах физики, механики и т.д. Например, в задаче расчета напряженно-деформированного состояния мембраны (абсолютно тонкой пленки), занимающей область D в плоскости xOy , на границе Γ которой заданы вертикальные смещения мембраны $u(x, y)|_{\Gamma} = f(x, y)$.

Пусть область D является прямоугольником со сторонами, параллельными координатным осям. В области D строится сетка с шагом h по оси Ox и с шагом H по оси Oy : $x_i = x_0 + ih$, $y_j = y_0 + jH$, где (x_0, y_0) - координаты точки A .

Вторые производные в уравнении Лапласа (1) заменяются разделенными разностями во всех внутренних точках (x_i, y_j) $1 \leq i \leq n-1$, $1 \leq j \leq m-1$ по формулам

$$\begin{aligned} \frac{\partial^2 u(ih, jH)}{\partial x^2} &\approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \\ \frac{\partial^2 u(ih, jH)}{\partial y^2} &\approx \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{H^2} \end{aligned} \quad (3)$$

Здесь введены обозначения для искомой функции $u(x, y)$ и известной функции на границе ABCD $f(x, y)$ вида $u(ih, jh) = u_{i,j}$, $f(ih, jh) = f_{i,j}$.

Подставив (3) в (1), получаем разностное уравнение

$$\lambda^2(u_{i+1,j} + u_{i-1,j}) + u_{i,j+1} + u_{i,j-1} - 2(1 + \lambda^2)u_{i,j} = 0 \quad (4)$$

$$(\lambda = H/h, \quad i = 1, 2, \dots, n-1, \quad j = 1, 2, \dots, m-1).$$

2. Для разработки алгоритма решения задачи Дирихле для уравнения Лапласа (1),(2) в прямоугольной области ABCD с вершинами A(0;0), B(0;1), C(1;1), D(1;0) (рис. 1)

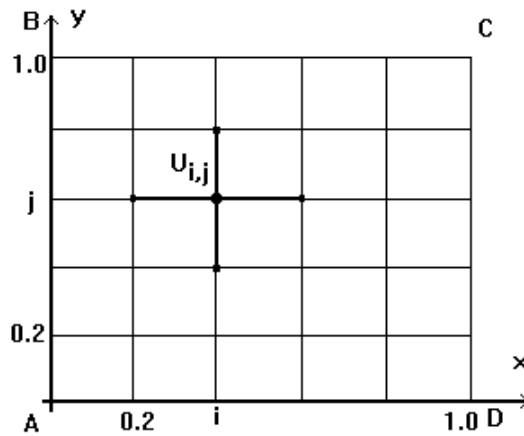


Рисунок 1

строится сетка с начальным шагом $h=0.2$ по осям Ox, Oy :

$$x_i = ih, y_i = ih, \quad i = 0, 1, 2, \dots, n \quad (n = 5).$$

Выражая $u_{i,j}$ из соотношения (4) при $\lambda = 1$, получим разностное уравнение

$$u_{i,j} = \frac{1}{4} [u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}] \quad (5)$$

при выполнении граничных условий

$$\begin{aligned} u_{i,0} &= f_{i,0} & (AD) \\ u_{i,n} &= f_{i,n} & (BC) \\ u_{0,j} &= f_{0,j} & (AB) \\ u_{n,j} &= f_{n,j} & (CD) \end{aligned} \quad (6)$$

$$(i, j = 0, 1, 2, \dots, n).$$

Подставляя из системы (6) значения $u_{i,j}$ в уравнения (5), получим систему $(n-1)$ -го линейного алгебраического уравнения с $(n-1)$ -м неизвестным.

3. Оценка допущенной погрешности. Сеточный метод решения задачи

Дирихле для уравнения Лапласа обладает точностью второго порядка (h^2, H^2)

по каждой из переменных, т.е. если предположить, что $u(x, y) \in C_4(D)$, то имеет место оценка

$$\max_{D_h} |u - u^h| = O(h^2),$$

где D_h множество всех внутренних узлов области D ; $u(x, y)$ - точное решение задачи Дирихле, $u^h(x, y)$ - приближенное решение, найденное с шагом h .

На практике для оценки погрешности используются расчеты решения $u_{i,j}^h$ и $u_{i,j}^{h/2}$ с шагами h и $h/2$. Если соответствующие результаты совпадают с заданной точностью

$$\left| u_{i,j}^h - u_{i,j}^{h/2} \right| < \varepsilon, \quad \text{для всех } i, j = 1, 2, \dots, n-1, \quad (7)$$

то считают, что искомое решение $u_{i,j}^{h/2}$ задачи найдено правильно. В противном случае последний шаг $h/2$ уменьшают вдвое, делают пересчет и сравнивают полученный новый результат $u_{i,j}^{h/2}$ с прежним результатом по формуле (7), и так далее, пока условие (7) не выполнится.

ВАРИАНТЫ ИНДИВИДУАЛЬНЫХ ЗАДАНИЙ

Вариант	u на АВ	u на ВС	u на CD	u на AD
1	y	1	y^2	0
2	$2(1-y^2)$	$2 \sin(\pi x)$	0	$2(1-x)$
3	$3y^{1/2}$	$3 \cos(\pi x)$	-3	$-3x$
4	0	$4x^2$	4	$4 \sin(\pi x/2)$
5	0	$5x(1-x)$	$5 \sin(\pi y)$	$5 \sin(\pi x)$
6	$6 \sin \pi y$	0	$6(y-1)^2$	$6x$
7	$7(1-y)^{1/2}$	$7(1-x)$	0	$7 \cos(\pi x/2)$
8	0	$8x$	$8 \sin(\pi y/2)$	$8x(1-x^2)$
9	$9y^2$	$9 \cos(\pi x)$	$-9y$	0
10	10	$10(1-x)$	0	$10(1-x^2)^2$