

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования
**«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»**

Кафедра информатики

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к лабораторным работам по курсу “Методы численного анализа”

для студентов специальности «Информатика» всех форм обучения

Минск 2006

УДК 681.3

Составитель Минченко Л.И.

Методические указания к лабораторным работам по курсу “Методы численного анализа”, для специальности «Информатика» всех форм обучения. – Мн.: БГУИР, 2006. –

СОДЕРЖАНИЕ

Введение	3
1. Задание 1. Решение систем линейных алгебраических уравнений (СЛАУ) методом Гаусса и с помощью его модификаций	4
2. Задание 2. Численное решение систем линейных уравнений методом простых итераций и методом Зейделя.....	4
3. Задание 3. Численное решение нелинейных уравнений	17
4. Задание 4. Решение систем нелинейных уравнений	27
5. Задание 5. Вычисление собственных значений и векторов	28
6. Задание 6. Интерполяционные многочлены	30
7. Задание 7. Интерполяция сплайнами	34
8. Задание 8. Численное дифференцирование и интегрирование функций	39
9. Задание 9. Методы Эйлера и Рунге-Кутты	44
10. Задание 10. Метод Адамса	48

ВВЕДЕНИЕ

При изучении курса «Численные методы» студенты должны освоить методы решения классических элементарных задач. С этой целью необходимо выполнить под руководством преподавателя на лабораторных занятиях предлагаемые индивидуальные задания, охватывающие основные задачи:

- 1) Численное решение систем линейных уравнений (метод Гаусса и его модификации);
- 2) Численное решение систем линейных уравнений (метод простых итераций, метод Зейделя);
- 3) Численное решение нелинейных уравнений;
- 4) Численное решение систем нелинейных уравнений;
- 5) Аппроксимация функций - интерполяция, метод наименьших квадратов;
- 6) Численное дифференцирование и интегрирование;
- 7) Решение задачи Коши для обыкновенных дифференциальных уравнений;
- 8) .

По результатам работы студентом должен быть представлен и защищен отчет. Содержание отчета включает:

1. Введение, содержащее постановку задачи, обзор имеющихся методов ее решения, их сравнительная характеристика.
2. Детальное описание и схема алгоритма выбранного метода.
3. Распечатку программы (каждый метод должен быть реализован в виде отдельной подпрограммы, в головной программе предусмотреть ввод исходных данных и обращение к подпрограмме), таблицы полученных результатов, графики сходимости приближенного решения к точному.
4. Анализ полученных результатов, включающий аналитическую оценку точности метода и ее сравнение с численно полученными результатами (т.е. оценку разности между точным и приближенным решением при различных параметрах метода).

ЗАДАНИЕ 1. Решение систем линейных алгебраических уравнений (СЛАУ) методом Гаусса и с помощью его модификаций

Цель выполнения задания:

- изучить метод Гаусса и его модификации, составить алгоритм метода и программу его реализации, получить численное решение заданной СЛАУ;
- составить алгоритм решения СЛАУ указанными методами, применимый для организации вычислений на ЭВМ;
- составить программу решения СЛАУ по разработанному алгоритму;
- выполнить тестовые примеры и проверить правильность работы программы.

Краткие теоретические сведения. Задача отыскания решения СЛАУ с n неизвестными является одной из наиболее часто встречающихся вычислительных задач. Хотя задача решения системы линейных уравнений сравнительно редко представляет самостоятельный интерес для приложений, от умения эффективно решать такие системы часто зависит сама возможность математического моделирования с применением ЭВМ разнообразных процессов. Значительная часть численных методов решения различных по своей природе задач (в особенности – нелинейных) включает в себя решение систем линейных уравнений как элементарный шаг соответствующего алгоритма.

Одна из трудностей практического решения систем большой размерности связана с ограниченностью оперативной памяти ЭВМ. Хотя объем оперативной памяти вновь создаваемых вычислительных машин растет очень быстро, тем не менее, еще быстрее возрастают потребности практики в решении задач все большей размерности. В значительной степени ограничения на размерность решаемых систем можно снять, если использовать для хранения матрицы внешние запоминающие устройства. Однако в этом случае многократно возрастают как затраты машинного времени, так и сложность соответствующих алгоритмов. Поэтому при создании вычислительных алгоритмов линейной алгебры большое внимание уделяют способам компактного размещения элементов матриц в памяти ЭВМ.

К счастью, приложения очень часто приводят к матрицам, в которых число ненулевых элементов намного меньше общего числа элементов матрицы. Такие матрицы принято называть *разреженными*. Одним из основных источников разреженных матриц являются математические модели

технических устройств, состоящих из большого числа элементов, связи между которыми локальны. Простейшие примеры таких устройств – сложные строительные конструкции и большие электрические цепи.

Известны примеры решенных в последние годы задач, где число неизвестных достигало сотен тысяч. Естественно, это было бы невозможно, если бы соответствующие матрицы не являлись разреженными (матрица системы из 100 тыс. уравнений в формате двойной точности заняла бы около 75 Гбайт).

СЛАУ обычно записывается в виде

$$\sum_{j=1}^n a_{ij}x_j = b_i; i \leq 1 \leq n, \text{ или коротко } \mathbf{Ax} = \mathbf{b}, \quad (1.1)$$

где

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}; \quad \mathbf{a} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}; \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}.$$

Здесь \mathbf{A} и \mathbf{b} заданы, требуется найти \mathbf{x} .

Методы решения СЛАУ делятся на прямые и итерационные.

Прямые методы дают в принципе точное решение (если не учитывать ошибок округления) за конечное число арифметических операций. Они просты и наиболее универсальны. Для хорошо обусловленных систем небольшого порядка $n \leq 200$ применяются практически только прямые методы.

Наибольшее распространение среди прямых методов получили метод Гаусса и его модификации.

Метод Гаусса. Одним из самых распространенных методов решения систем линейных уравнений является метод Гаусса. Этот метод (который также называют *методом последовательного исключения неизвестных*) широко известен в различных вариантах.

Вычисления с помощью метода Гаусса заключаются в последовательном исключении неизвестных из системы для преобразования ее к эквивалентной системе с верхней треугольной матрицей. Вычисления значений неизвестных производят на этапе обратного хода.

Рассмотрим сначала простейший вариант метода Гаусса, называемый *схемой единственного деления*.

Прямой ход состоит из $n - 1$ шагов исключения.

1-й шаг. Целью этого шага является исключение неизвестного x_1 из уравнений с номерами $i = 2, 3, \dots, n$. Предположим, что коэффициент $a_{11} \neq 0$. Будем называть его *главным элементом 1-го шага*.

Найдем величины

$$q_{i1} = a_{i1}/a_{11} \quad (i = 2, 3, \dots, n),$$

называемые *множителями 1-го шага*. Вычтем последовательно из второго, третьего, ..., n -го уравнений системы первое уравнение, умноженное соответственно на $q_{21}, q_{31}, \dots, q_{n1}$. Это позволит обратить в нуль коэффициенты при x_1 во всех уравнениях, кроме первого. В результате получим эквивалентную систему

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n &= b_1, \\ a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \dots + a_{2n}^{(1)}x_n &= b_2^{(1)}, \\ a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 + \dots + a_{3n}^{(1)}x_n &= b_3^{(1)}, \\ \vdots &\vdots \\ a_{n2}^{(1)}x_2 + a_{n3}^{(1)}x_3 + \dots + a_{nn}^{(1)}x_n &= b_n^{(1)}, \end{aligned}$$

в которой $a_{ij}^{(1)}$ и $b_i^{(1)}$ вычисляются по формулам

$$a_{ij}^{(1)} = a_{ij} - q_{i1}a_{1j}, \quad b_i^{(1)} = b_i - q_{i1}b_1.$$

2-й шаг. Целью этого шага является исключение неизвестного x_2 из уравнений с номерами $i = 3, 4, \dots, n$. Пусть $a_{22}^{(1)} \neq 0$, где $a_{22}^{(1)}$ – коэффициент, называемый *главным* (или *ведущим*) *элементом 2-го шага*. Вычислим множители 2-го шага

$$q_{i2} = a_{i2}^{(1)} / a_{22}^{(1)} \quad (i = 3, 4, \dots, n)$$

и вычтем последовательно из третьего, четвертого, ..., n -го уравнений системы второе уравнение, умноженное соответственно на $q_{32}, q_{42}, \dots, q_{n2}$. В результате получим систему

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n &= b_1, \\ a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \dots + a_{2n}^{(1)}x_n &= b_2^{(1)}, \\ a_{33}^{(2)}x_3 + \dots + a_{3n}^{(2)}x_n &= b_3^{(2)}, \\ \vdots &\vdots \\ a_{n3}^{(2)}x_3 + \dots + a_{nn}^{(2)}x_n &= b_n^{(2)}. \end{aligned}$$

Здесь коэффициенты $a_{ij}^{(2)}$ и $b_j^{(2)}$ вычисляются по формулам

$$a_{ij}^{(2)} = a_{ij}^{(1)} - q_{i2}a_{2j}^{(1)}, \quad b_i^{(2)} = b_i^{(1)} - q_{i2}b_2^{(1)}.$$

Аналогично проводятся остальные шаги. Опишем очередной k -й шаг.

k -й шаг. В предположении, что *главный (ведущий) элемент* k -го шага $a_{kk}^{(k-1)}$ отличен от нуля, вычислим *множители* k -го шага

$$q_{ik} = a_{ik}^{(k-1)} / a_{kk}^{(k-1)} \quad (i = k + 1, \dots, n)$$

и вычтем последовательно из $(k + 1)$ -го, ..., n -го уравнений полученной на предыдущем шаге системы k -е уравнение, умноженное соответственно на $q_{k+1,k}, q_{k+2,k}, \dots, q_{nk}$.

После $(n - 1)$ -го шага исключения получим систему уравнений

$$\begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n & = & b_1 \\ a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \dots + a_{2n}^{(1)}x_n & = & b_2^{(1)} \\ a_{33}^{(2)}x_3 + \dots + a_{3n}^{(2)}x_n & = & b_3^{(2)} \\ \dots & & \dots \\ a_{nn}^{(n-1)}x_n & = & b_n^{(n-1)} \end{array},$$

матрица $A^{(n-1)}$ которой является верхней треугольной. На этом вычисления прямого хода заканчиваются.

Обратный ход. Из последнего уравнения системы находим x_n . Подставляя найденное значение x_n в предпоследнее уравнение, получим x_{n-1} . Осуществляя обратную подстановку, далее последовательно находим x_{n-2}, \dots, x_1 . Вычисления неизвестных здесь проводятся по формулам

$$x_n = b_n^{(n-1)} / a_{nn}^{(n-1)},$$

$$x_k = (b_k^{(k-1)} - a_{k,k+1}^{(k-1)}x_{k+1} - \dots - a_{kn}^{(k-1)}x_n) / a_{kk}^{(k-1)}, \quad (k = n - 1, \dots, 1).$$

Необходимость отличия от 0 главных элементов. Заметим, что вычисление множителей, а также обратная подстановка требуют деления на главные элементы $a_{kk}^{(k-1)}$. Поэтому если один из главных элементов оказывается равным нулю, то схема не может быть реализована. Здравый смысл подсказывает, что и в ситуации, когда все главные элементы отличны от нуля, но среди них есть близкие к нулю, возможен неконтролируемый рост погрешности.

Метод Гаусса с выбором главного элемента по столбцу (схема частичного выбора). На k -м шаге прямого хода коэффициенты уравнений системы с номерами $i = k + 1, \dots, n$ преобразуются по формулам

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - q_{ik}a_{kj}, \quad b_i^{(k)} = b_i^{(k-1)} - q_{ik}b_k^{(k-1)}, \quad i = k + 1, \dots, n.$$

Интуитивно ясно, что во избежание сильного роста коэффициентов системы и связанных с этим ошибок нельзя допускать появления больших множителей q_{ik} .

В методе Гаусса с выбором главного элемента по столбцу гарантируется, что $|q_{ik}| \leq 1$ для всех $k = 1, 2, \dots, n-1$ и $i = k+1, \dots, n$. Отличие этого варианта метода Гаусса от схемы единственного деления заключается в том, что на k -м шаге исключения в качестве главного элемента выбирают максимальный по модулю коэффициент a_{ik} при неизвестной x_k в уравнениях с номерами $i = k+1, \dots, n$. Затем соответствующее выбранному коэффициенту уравнение с номером i_k меняют местами с k -м уравнением системы для того, чтобы главный элемент занял место коэффициента $a_{kk}^{(k-1)}$. После этой перестановки исключение неизвестного x_k производят, как в схеме единственного деления.

Метод Гаусса с выбором главного элемента по всей матрице (схема полного выбора). В этой схеме допускается нарушение естественного порядка исключения неизвестных. На 1-м шаге метода среди элементов a_{ij} определяют максимальный по модулю элемент $a_{i_1 j_1}$. Первое уравнение системы и уравнение с номером i_1 меняют местами. Далее стандартным образом производят исключение неизвестного x_{i_1} из всех уравнений, кроме первого.

На k -м шаге метода среди коэффициентов $a_{ij}^{(k-1)}$ при неизвестных в уравнениях системы с номерами $i = k, \dots, n$ выбирают максимальный по модулю коэффициент $a_{i_k j_k}^{(k-1)}$. Затем k -е уравнение и уравнение, содержащее найденный коэффициент, меняют местами и исключают неизвестное x_{j_k} из уравнений с номерами $i = k+1, \dots, n$.

На этапе обратного хода неизвестные вычисляют в следующем порядке: $x_{j_n}, x_{j_{n-1}}, \dots, x_{j_1}$.

ЗАДАНИЕ. Методом Гаусса и методом выбора главного элемента найти с точностью 0,0001 численное решение системы $Ax=b$, где $A = kC + D$, A – исходная матрица для расчёта, k – номер варианта (0-15), матрицы C, D и вектор свободных членов b задаются ниже.

Исходные данные:

$$\text{Вектор } \mathbf{b} = (4,2; 4,2; 4,2; 4,2; 4,2)^T,$$

$$C = \begin{bmatrix} 0,2 & 0 & 0,2 & 0 & 0 \\ 0 & 0,2 & 0 & 0,2 & 0 \\ 0,2 & 0 & 0,2 & 0 & 0,2 \\ 0 & 0,2 & 0 & 0,2 & 0 \\ 0 & 0 & 0,2 & 0 & 0,2 \end{bmatrix}, \quad D = \begin{bmatrix} 2,33 & 0,81 & 0,67 & 0,92 & -0,53 \\ -0,53 & 2,33 & 0,81 & 0,67 & 0,92 \\ 0,92 & -0,53 & 2,33 & 0,81 & 0,67 \\ 0,67 & 0,92 & -0,53 & 2,33 & 0,81 \\ 0,81 & 0,67 & 0,92 & -0,53 & 2,33 \end{bmatrix}.$$

ЗАДАНИЕ 2. Численное решение систем линейных уравнений методом простых итераций и методом Зейделя.

Цель выполнения задания:

- Изучить итерационные методы решения СЛАУ (метод простых итераций, метод Зейделя).
- Составить алгоритм решения СЛАУ указанными методами, применимый для организации вычислений на ЭВМ.
- Составить программу решения СЛАУ по разработанному алгоритму.
- Численно решить тестовые примеры и проверить правильность работы программы. Сравнить трудоемкость решения методом простых итераций и методом Зейделя.

Краткие теоретические сведения. Прямые методы применяют главным образом для решения задач малой размерности, когда нет ограничений в доступной оперативной памяти ЭВМ или необходимости выполнения чрезмерно большого числа арифметических операций. Большие системы уравнений, возникающие в основном в приложениях, как правило, являются разреженными. Методы исключения для систем с разреженным и матрицами неудобны, например, тем, что при их использовании большое число нулевых элементов превращается в ненулевые, и матрица теряет свойство разреженности. В противоположность им при использовании итерационных методов в ходе итерационного процесса матрица не меняется, и она, естественно, остается разреженной. Большая эффективность итерационных методов по сравнению с прямыми методами тесно связана с возможностью существенного использования разреженности матриц.

Итерационные методы основаны на построении сходящейся к точному решению \mathbf{x} рекуррентной последовательности.

Для решения СЛАУ **методом простых итераций** преобразуем систему от первоначальной формы $\mathbf{Ax} = \mathbf{b}$ или

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2, \\ &\dots\dots\dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n \end{aligned} \quad (2.1)$$

к виду

$$\mathbf{x} = \mathbf{Bx} + \mathbf{c}. \quad (2.2)$$

Здесь \mathbf{B} – квадратная матрица с элементами b_{ij} ($i, j = 1, 2, \dots, n$), \mathbf{c} – вектор-столбец с элементами c_i ($i = 1, 2, \dots, n$).

В развернутой форме записи система (2.2) имеет следующий вид:

$$\begin{aligned} x_1 &= b_{11}x_1 + b_{12}x_2 + b_{13}x_3 + \dots + b_{1n}x_n + c_1 \\ x_2 &= b_{21}x_1 + b_{22}x_2 + b_{23}x_3 + \dots + b_{2n}x_n + c_2 \\ &\dots\dots\dots \\ x_n &= b_{n1}x_1 + b_{n2}x_2 + b_{n3}x_3 + \dots + b_{nn}x_n + c_n \end{aligned}$$

Вообще говоря, операция *приведения системы к виду, удобному для итераций*, не является простой и требует специальных знаний, а также существенного использования специфики системы.

Можно, например, преобразовать систему (2.1) следующим образом

$$\begin{aligned} x_1 &= (b_1 - a_{11}x_1 - a_{12}x_2 - \dots - a_{1n}x_n) / a_{11} + x_1, \\ x_2 &= (b_2 - a_{21}x_1 - a_{22}x_2 - \dots - a_{2n}x_n) / a_{22} + x_2, \\ &\dots\dots\dots \\ x_n &= (b_n - a_{n1}x_1 - a_{n2}x_2 - \dots - a_{nn}x_n) / a_{nn} + x_n \end{aligned}$$

если диагональные элементы матрицы \mathbf{A} отличны от нуля.

Можно преобразовать систему (2.1) в эквивалентную ей систему

$$\mathbf{x} = (\mathbf{E} - \mathbf{A})\mathbf{x} + \mathbf{b}.$$

Задав произвольным образом столбец начальных приближений $\mathbf{x}^0 = (x_1^0, x_2^0, \dots, x_n^0)^T$, подставим их в правые части системы (2.2) и вычислим новые приближения $\mathbf{x}^1 = (x_1^1, x_2^1, \dots, x_n^1)^T$, которые опять подставим в систему (2.2) и т.д. Таким образом, организуется итерационный процесс

$\mathbf{x}^k = \mathbf{B}\mathbf{x}^{k-1} + \mathbf{c}$, $k = 1, 2, \dots$. Известно, что система (2.1) имеет единственное решение \mathbf{x}^* и последовательность $\{\mathbf{x}^k\}$ сходится к этому решению со скоростью геометрической прогрессии, если $\|\mathbf{B}\| < 1$ в любой матричной норме. Т.е.

Т.е. для того, чтобы последовательность простых итераций сходилась к единственному решению достаточно, чтобы выполнялось одно из следующих условий:

$$1) \max_i \left(\sum_{j=1}^n |b_{ij}| \right) < 1;$$

$$2) \sum_{i=1}^n \sum_{j=1}^n |b_{ij}|^2 < 1;$$

$$3) \max_j \left(\sum_{i=1}^n |b_{ij}| \right) < 1.$$

Метод Зейделя. Метод Зейделя является модификацией метода простых итераций. Суть его состоит в том, что при вычислении следующего x_i^k : $2 \leq i \leq n$ в формуле $\mathbf{x}^k = \mathbf{B}\mathbf{x}^{k-1} + \mathbf{c}$, $k = 1, 2, \dots$ используются вместо $x_1^{k-1}, \dots, x_{i-1}^{k-1}$ уже вычисленные ранее x_1^k, \dots, x_{i-1}^k , т.е.

$$x_i^k = \sum_{j=1}^{i-1} g_{ij} x_j^k + \sum_{j=i+1}^n g_{ij} x_j^{k-1} + c_i. \quad (2.3) \quad \text{Такое}$$

усовершенствование позволяет ускорить сходимость итераций почти в два раза. Кроме того, данный метод может быть реализован на ЭВМ без привлечения дополнительного массива, т.к. полученное новое x_i^k сразу засылается на место старого.

Схема алгоритма аналогична схеме метода простых итераций.

Самый простой способ приведения системы к виду, удобному для итераций, состоит в следующем. Из первого уравнения системы выразим неизвестное x_1 :

$$x_1 = a_{11}^{-1} (b_1 - a_{12}x_2 - a_{13}x_3 - \dots - a_{1n}x_n),$$

из второго уравнения – неизвестное x_2 :

$$x_2 = a_{21}^{-1} (b_2 - a_{22}x_2 - a_{23}x_3 - \dots - a_{2n}x_n),$$

и т. д. В результате получим систему

$$\begin{array}{ccccccc} x_1 = b_{12}x_2 + & b_{13}x_3 + \dots + & b_{1,n-1}x_{n-1} + & b_{1n}x_n + c_1, \\ x_2 = b_{21}x_1 + & b_{23}x_3 + \dots + & b_{2,n-1}x_{n-1} + & b_{2n}x_n + c_2, \\ x_3 = b_{31}x_1 + & b_{32}x_2 + \dots + & b_{3,n-1}x_{n-1} + & b_{3n}x_n + c_3, \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n = b_{n1}x_1 + & b_{n2}x_2 + b_{n3}x_3 + & \dots + & b_{n,n-1}x_{n-1} + & c_n, \end{array}$$

в которой на главной диагонали матрицы \mathbf{B} находятся нулевые элементы. Остальные элементы выражаются по формулам

$$b_{ij} = -a_{ij} / a_{ii}, c_i = b_i / a_{ii} \quad (i, j = 1, 2, \dots, n, j \neq i) \quad (2.4)$$

Конечно, для возможности выполнения указанного преобразования необходимо, чтобы диагональные элементы матрицы A были ненулевыми.

Введем нижнюю \mathbf{B}_1 (получается из \mathbf{B} заменой нулями элементов стоявших на главной диагонали и выше ее) и верхнюю \mathbf{B}_2 (получается из \mathbf{B} заменой нулями элементов стоявших на главной диагонали и ниже ее) треугольные матрицы.

Заметим, что $\mathbf{B} = \mathbf{B}_1 + \mathbf{B}_2$ и поэтому решение \mathbf{x} исходной системы удовлетворяет равенству

$$\mathbf{x} = \mathbf{B}_1 \mathbf{x} + \mathbf{B}_2 \mathbf{x} + \mathbf{c} \quad (2.5)$$

Выберем начальное приближение $\mathbf{x}^{(0)} = [x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}]^T$. Подставляя его в правую часть равенства при верхней треугольной матрице \mathbf{B}_2 и вычисляя полученное выражение, находим первое приближение

$$\mathbf{x}^{(1)} = \mathbf{B}_1 \mathbf{x}^{(0)} + \mathbf{B}_2 \mathbf{x}^{(1)} \quad (2.6)$$

Подставляя приближение $\mathbf{x}^{(1)}$, получим

$$\mathbf{x}^{(2)} = \mathbf{B}_1 \mathbf{x}^{(1)} + \mathbf{B}_2 \mathbf{x}^{(2)} \quad (2.7)$$

Продолжая этот процесс далее, получим последовательность $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}$, ... приближений к вычисляемым по формуле

$$\mathbf{x}^{(k+1)} = \mathbf{B}_1 \mathbf{x}^{(k+1)} + \mathbf{B}_2 \mathbf{x}^{(k)} + \mathbf{c} \quad (2.8)$$

или в развернутой форме записи

$$x_1^{(k+1)} = b_{12}x_2^{(k)} + b_{13}x_3^{(k)} + \dots + b_{1n}x_n^{(k)} + c_1,$$

$$x_2^{(k+1)} = b_{21}x_1^{(k+1)} + b_{23}x_3^{(k)} + \dots + b_{2n}x_n^{(k)} + c_2,$$

$$x_3^{(k+1)} = b_{31}x_1^{(k+1)} + b_{32}x_2^{(k+1)} + \dots + b_{3n}x_n^{(k)} + c_3,$$

$$x_n^{(k+1)} = b_{n1}x_1^{(k+1)} + b_{n2}x_2^{(k+1)} + b_{n3}x_3^{(k+1)} + \dots + c_n.$$

Объединив приведение системы к виду, удобному для итераций и метод Зейделя в одну формулу, получим

$$x_i^{(k+1)} = x_i^{(k)} - a_{ii}^{-1} (\sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} + \sum_{j=1}^n a_{ij} x_j^{(k)} - b_i). \quad (2.9)$$

Тогда достаточным условием сходимости метода Зейделя будет условие доминирования диагональных элементов в строках или столбцах матрицы A , т.е.

$$a_{ii} > a_{i1} + \dots + a_{in} \quad \text{для всех } i=1, \dots, n,$$

$$\text{или } a_{jj} > a_{1j} + \dots + a_{nj} \quad \text{для всех } j=1, \dots, n.$$

Методы простой итерации и Зейделя сходятся примерно так же, как геометрическая прогрессия со знаменателем $\|\mathbf{B}\|$.

ЗАДАНИЕ. Методом простых итераций и методом Зейделя найти с точностью 0,0001 численное решение системы $A\mathbf{x}=\mathbf{b}$, где $A = kC + D$, A – исходная матрица для расчёта, k – номер варианта (0-15), матрицы C, D и вектор свободных членов \mathbf{b} задаются ниже.

$$C = \begin{bmatrix} 0,01 & 0 & -0,02 & 0 & 0 \\ 0,01 & 0,01 & -0,02 & 0 & 0 \\ 0 & 0,01 & 0,01 & 0 & -0,02 \\ 0 & 0 & 0,01 & 0,01 & 0 \\ 0 & 0 & 0 & 0,01 & 0,01 \end{bmatrix}, \quad D = \begin{bmatrix} 1,33 & 0,21 & 0,17 & 0,12 & -0,13 \\ -0,13 & -1,33 & 0,11 & 0,17 & 0,12 \\ 0,12 & -0,13 & -1,33 & 0,11 & 0,17 \\ 0,17 & 0,12 & -0,13 & -1,33 & 0,11 \\ 0,11 & 0,67 & 0,12 & -0,13 & -1,33 \end{bmatrix}.$$

Вектор $\mathbf{b} = (1,2; 2,2; 4,0; 0,0; -1,2)^T$.

ЗАДАНИЕ 3. Численное решение нелинейных уравнений

Цель работы: Изучение методов численного решения нелинейных уравнений - методов бисекции, хорд, простой итерации, релаксации, метода Ньютона и его модификаций; исследование скорости сходимости итерационных процедур; изучение метода Эйткена ускорения сходимости; сравнение числа итераций, необходимого для достижения заданной точности вычисления разными методами.

Краткие теоретические сведения

Численное решение нелинейного уравнения $f(x)=0$ заключается в вычислении с заданной точностью значения всех или некоторых корней уравнения и распадается на несколько задач: *во-первых*, надо исследовать количество и характер корней (вещественные или комплексные, простые или кратные), *во-вторых*, определить их приближенное расположение, т.е. значения начала и конца отрезка, на котором лежит только один корень, *в-третьих*, выбрать интересующие нас корни и вычислить их с требуемой точностью. Вторая задача называется **отделением корней**. Решив ее, по сути дела, находят приближенные значения корней с погрешностью, не превосходящей длины отрезка, содержащего корень. Отметим два простых приема отделения действительных корней уравнения - *табличный* и *графический*. Первый прием состоит в вычислении таблицы значений функции $f(x)$ в заданных точках x_i и использовании следующих теорем математического анализа:

1. Если функция $y=f(x)$ непрерывна на отрезке $[a,b]$ и $f(a)f(b)<0$, то внутри отрезка $[a,b]$ существует по крайней мере один корень уравнения $f(x)=0$.
2. Если функция $y=f(x)$ непрерывна на отрезке $[a,b]$, $f(a)f(b) < 0$ и $f'(x)$ на интервале (a,b) сохраняет знак, то внутри отрезка $[a,b]$ существует единственный корень уравнения $f(x)=0$.

Таким образом, если при некотором k числа $f(x_k)$ и $f(x_{k+1})$ имеют разные знаки, то это означает, что на интервале (x_k, x_{k+1}) уравнение имеет по крайней мере один действительный корень нечетной кратности (точнее - нечетное число корней). Выявить по таблице корень четной кратности очень сложно.

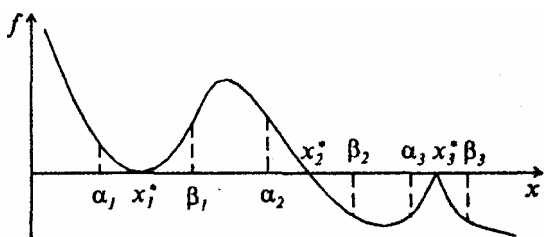


Рис. 5.1

На рис. 3.1 представлены три наиболее часто встречающиеся ситуации:

- а) кратный корень: $f'(x^*)=0, f(a_1) \cdot f(b_1) > 0$;
- б) простой корень: $f'(x^*) \neq 0, f(a_2) \cdot f(b_2) < 0$;
- в) вырожденный корень: $f'(x^*)$ не существует, $f(a_3) \cdot f(b_3) > 0$.

Как видно из рис. 3.1, в первых двух случаях значение корня совпадает с точкой экстремума функции и для нахождения таких корней рекомендуется использовать методы поиска минимума функции.

Для определения числа корней на заданном промежутке используется *Теорема Штурма*: Если $f(x)$ многочлен и уравнение не имеет кратных корней на промежутке $[a, b]$, то число корней уравнения $f(x) = 0$, лежащих на

промежутке $[a, b]$, совпадает с числом $N(a) - N(b)$, которое определяется из следующей процедуры.

Строим ряд Штурма $f_0(x), f_1(x), f_2(x), \dots, f_m(x)$, где

$$f_0(x) = f(x);$$

$$f_1(x) = f'(x);$$

$f_0(x)$ делим на $f_1(x)$ и в качестве $f_2(x)$ берем остаток от деления, взятый с обратным знаком;

$f_1(x)$ делим на $f_2(x)$ и в качестве $f_3(x)$ берем остаток от деления, взятый с обратным знаком;

и т.д.

Полагаем $N(a)$ – число перемен знака в ряде Штурма, если вместо x подставлена точка a , $N(b)$ – число перемен знака в ряде Штурма, если вместо x подставлена точка b .

Для отделения корней можно использовать график функции $y=f(x)$. Корнями уравнения являются те значения x , при которых график функции пересекает ось абсцисс. Построение графика функции даже с малой точностью обычно дает представление о расположении и характере корней уравнения (иногда позволяет выявить даже корни четной кратности). Если построение графика функции $y=f(x)$ вызывает затруднение, следует преобразовать исходное уравнение к виду $\varphi_1(x)=\varphi_2(x)$ таким образом, чтобы графики функций $y=\varphi_1(x)$ и $y=\varphi_2(x)$ были достаточно просты. Абсциссы точек пересечения этих графиков и будут корнями уравнения.

Допустим, что искомый корень уравнения отделен, т.е. найден отрезок $[a,b]$, на котором имеется только один корень уравнения. Для вычисления корня

с требуемой точностью ε обычно применяют какую-либо итерационную процедуру **уточнения корня**, строящую числовую последовательность значений x_n , сходящуюся к искомому корню уравнения. Начальное приближение x_0 выбирают на отрезке $[a,b]$, продолжают вычисления, пока не выполнится неравенство $|x_{n-1} - x_n| < \varepsilon$, и считают, что x_n – есть корень уравнения, найденный с заданной точностью. Имеется множество различных методов построения таких последовательностей и выбор алгоритма – весьма важный момент при практическом решении задачи. Немалую роль при этом играют такие свойства метода, как простота, надежность, экономичность, важнейшей характеристикой является его *скорость сходимости*. Последовательность x_n , сходящаяся к пределу x^* , имеет скорость сходимости

порядка α , если при $n \rightarrow \infty$ $|x_{n+1} - x^*| = O(|x_n - x^*|^\alpha)$. При $\alpha=1$ сходимость называется линейной, при $1 < \alpha < 2$ - сверхлинейной, при $\alpha=2$ - квадратичной. С ростом α алгоритм, как правило, усложняется и условия сходимости становятся более жесткими. Рассмотрим наиболее распространенные итерационные методы уточнения корня.

Метод простых итераций. Вначале уравнение $f(x)=0$ преобразуется к эквивалентному уравнению вида $x=\varphi(x)$. Это можно сделать многими способами, например, положив $\varphi(x)=x+\psi(x)f(x)$, где $\psi(x)$ -произвольная непрерывная знакопостоянная функция. Выбираем некоторое начальное приближение x_0 и вычисляем дальнейшие приближения по формуле

$$x_k = \varphi(x_{k-1}), \quad k=0,1,\dots$$

Метод простых итераций не всегда обеспечивает сходимость к корню уравнения. Достаточным условием сходимости этого метода является выполнение неравенства $|\varphi'(x)| \leq q < 1$ на отрезке, содержащем корень и все приближения x_n . Метод имеет линейную скорость сходимости и справедливы следующие оценки:

$$\begin{aligned} |x_n - x^{**}| &< \frac{q}{1-q} |x_n - x_{n-1}|, \text{ если } \varphi'(x) > 0, \\ |x_n - x^{**}| &< |x_n - x_{n-1}|, \text{ если } \varphi'(x) < 0. \end{aligned}$$

Метод имеет простую геометрическую интерпретацию: нахождение корня уравнения $f(x)=0$ равносильно обнаружению неподвижной точки функции $x=\varphi(x)$, т.е. точки пересечения графиков функций $y=\varphi(x)$ и $y=x$. Если производная $\varphi'(x) < 0$, то последовательные приближения колеблются около корня, если же производная $\varphi'(x) > 0$, то последовательные приближения сходятся к корню монотонно.

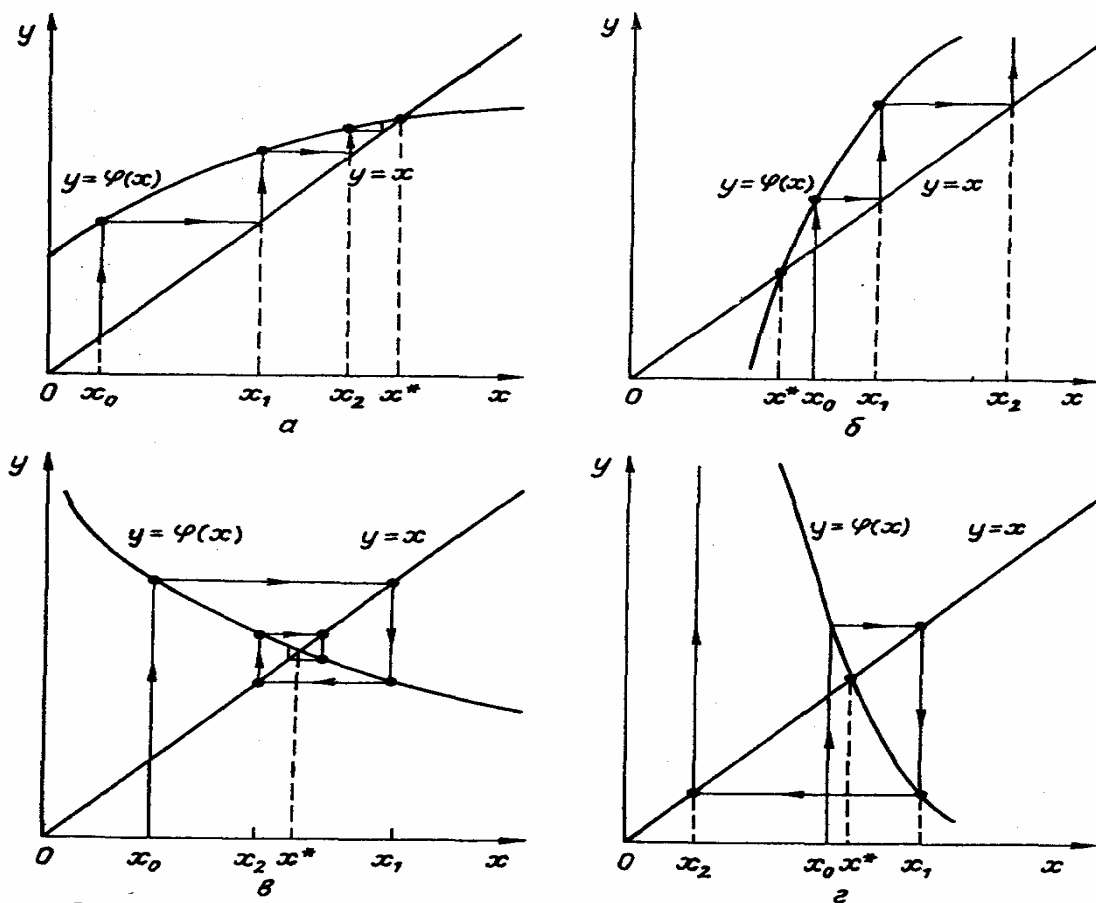


Рис. 3.2. Метод простых итераций: а - односторонний сходящийся процесс; б - односторонний расходящийся процесс; в - двухсторонний сходящийся процесс; г- двухсторонний расходящийся процесс

Рассмотрим процесс графически (рис. 3.2). Из графиков видно, что при $\phi'(x) > 0$ и при $\phi'(x) < 0$ возможны как сходящиеся, так и расходящиеся итерационные процессы. Скорость сходимости зависит от абсолютной величины производной $\phi(x)$. Чем меньше $|\phi'(x)|$ вблизи корня, тем быстрее сходится процесс.

Метод хорд. Пусть дано уравнение $f(x) = 0$, $a \leq x \leq b$, где $f(x)$ — дважды непрерывно дифференцируемая функция.

Пусть выполняется условие $f(a) \cdot f(b) < 0$ и проведено отделение корней, то есть на данном интервале (a, b) находится один корень уравнения. При этом, не ограничивая общности, можно считать, что $f(b) > 0$.

Пусть функция f выпукла на интервале (a, b) (см. рис. 3.3).

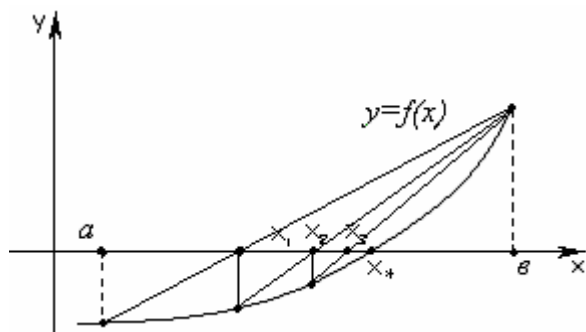


Рис. 3.3

Заменим график функции хордой (прямой), проходящей через точки $M_0(a, f(a))$ и $M_1(b, f(b))$.

Уравнение прямой, проходящей через две заданные точки, можно записать в виде $\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1}$. В нашем случае получим: $\frac{y - f(a)}{f(b) - f(a)} = \frac{x - a}{b - a}$.

Найдем точку пересечения хорды с осью Ox .

Полагая $y = 0$, получаем из предыдущего уравнения:

$$x_1 = a - \frac{f(a)}{f(b) - f(a)} \cdot (b - a).$$

Теперь возьмем интервал (x_1, b) в качестве исходного и повторим вышеописанную процедуру (см. рис. 6.3). Получим

$$x_2 = x_1 - \frac{f(x_1)}{f(b) - f(x_1)} \cdot (b - x_1).$$

Продолжим процесс. Каждое последующее приближение вычисляется по рекуррентной формуле

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f(b) - f(x_{n-1})} \cdot (b - x_{n-1}) \quad n = 1, 2, \dots, \quad (3.1)$$

$$x_0 = a.$$

Если же функция вогнута (см. рис. 3.4),

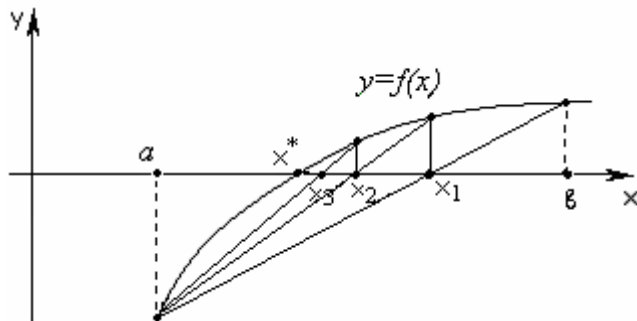


Рис. 3.4

уравнение прямой соединяющей точки $M_0(a, f(a))$ и $M_1(b, f(b))$ запишем в виде

$$\frac{y - f(b)}{f(a) - f(b)} = \frac{x - b}{a - b}.$$

Найдем точку пересечения хорды с осью Ох:

$$x_1 = b - \frac{f(b)}{f(a) - f(b)} \cdot (a - b).$$

Теперь возьмем интервал (a, x_1) в качестве исходного и найдем точки пересечения хорды, соединяющей точки $(a, f(a))$ и $(x_1, f(x_1))$, с осью абсцисс (см. рис. 3.4). Получим

$$x_2 = x_1 - \frac{f(x_1)}{f(a) - f(x_1)} \cdot (a - x_1).$$

Повторяя данную процедуру, получаем рекуррентную формулу:

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f(a) - f(x_{n-1})} \cdot (a - x_{n-1}) \quad n = 1, 2, \dots \quad (3.2)$$

$$x_0 = b.$$

Описанный выше метод построения рекуррентных последовательностей (3.1) и (3.2) называется методом хорд. Для использования метода хорд нужно было бы предварительно найти точки перегиба и выделить участки, на которых функция не меняет характер выпуклости. Однако на практике поступают проще: в случае $f(b)f''(b) > 0$ для построения рекуррентной последовательности применяются формулы (3.1), а в случае, когда $f(a)f''(a) > 0$, применяют формулы (3.2).

Метод Ньютона (касательных). Для начала вычислений требуется задание одного начального приближения x_0 , последующие приближения вычисляются по формуле

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad f'(x_n) \neq 0.$$

Метод имеет квадратичную скорость сходимости для простого корня, но очень чувствителен к выбору начального приближения. При произвольном начальном приближении итерации сходятся, если всюду $|f(x)f''(x)| < (f'(x))^2$, в противном случае сходимость будет только при x_0 , достаточно близком к корню. Существует несколько достаточных условий сходимости. Если производные $f'(x)$ и $f''(x)$ сохраняют знак в окрестности корня, рекомендуется выбирать x_0 так, чтобы $f(x_0)f''(x_0) > 0$. Если, кроме этого, для отрезка $[a, b]$, содержащего корень, выполняются условия

$$\left| \frac{f(a)}{f'(a)} \right| < b - a, \quad \left| \frac{f(b)}{f'(b)} \right| < b - a,$$

то метод сходится для любых $a \leq x_0 \leq b$.

Метод Ньютона получил также второе название *метод касательных* благодаря геометрической иллюстрации его сходимости, представленной на рис. 3.4

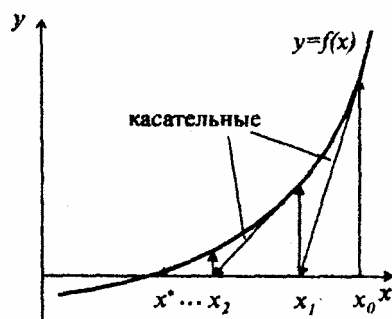


Рис. 5.4

Метод Ньютона позволяет находить как простые, так и кратные корни. сновной его недостаток - малая область сходимости и необходимость вычисления производной.

ЗАДАНИЕ.

- 1) Используя теорему Штурма определить число корней уравнения: $x^3 + ax^2 + bx + c = 0$ на отрезке $[-10, 10]$. Значения коэффициентов уравнения взять из таблицы.
- 2) Отделить все корни, лежащие на данном отрезке.
- 3) Вычислить наименьший из корней сначала методом половинного деления, а затем методом хорд и методом Ньютона. Сравнить число необходимых итераций в обоих методах. Точность до 0.0001.

a	b	c	№ вариан-та
-14,4621	60,6959	-70,9238	1
-10,2374	-91,2105	492,560	2
-19,7997	28,9378	562,833	3
-5,5796	-193,022	-633,105	4
9,57496	-243,672	773,65	5
20,2374	-131,210	-843,923	6
38,4621	364,594	914,196	7
-13,3667	39,8645	-20,6282	8
2,65804	-28,0640	21,9032	9
-6,4951	-31,2543	23,1782	10
9,9296	17,8390	-24,4532	11
6,0951	-35,3942	-25,7283	12

ЗАДАНИЕ 4. Решение систем нелинейных уравнений

Цель задания: изучить численное решение систем нелинейных уравнений методами простых итераций и Ньютона. Провести отделение решений, построить и запрограммировать алгоритмы методов, численно решить тестовое задание, сравнить трудоемкость методов.

Краткие теоретические сведения. Пусть дана система нелинейных уравнений (система не линейна, если хотя бы одно из входящих в нее уравнений не линейно):

$$\begin{cases} f_1(x_1, \dots, x_n) = 0 \\ f_2(x_1, \dots, x_n) = 0 \\ \dots\dots\dots \\ f_n(x_1, \dots, x_n) = 0 \end{cases}$$

Мы можем записать систему в более компактной векторной форме

$$f(x) = 0, \tag{4.1}$$

где $f = (f, \dots, f)$, $x = (x, \dots, x)^T$.

Для решения системы (4.1) иногда можно применить метод последовательного исключения неизвестных, который приводит решение системы к решению одного уравнения с одним неизвестным. Однако в подавляющем большинстве случаев систему уравнений (4.1) решают итерационными методами. Для решения системы (4.1) существует набор методов, из которых рассмотрим простейшие методы: метод простых итераций, базирующийся на принципе сжимающих отображений и метод Ньютона (многомерный аналог метода касательных).

Метод простых итераций. Чтобы воспользоваться методом простых итераций, необходимо предварительно привести систему к следующему виду:

$$\begin{cases} x_1 = \varphi_1(x_1, \dots, x_n) \\ x_2 = \varphi_2(x_1, \dots, x_n) \\ \dots\dots\dots \\ x_n = \varphi_n(x_1, \dots, x_n) \end{cases}$$

Или в векторной форме

$$\bar{x} = \varphi(\bar{x}) \quad (4.2)$$

где $\vec{\varphi} = (\varphi_1, \varphi_2, \dots, \varphi_n)$.

Исходя из некоторого начального приближения \bar{x}^{-0} , построим итерационную последовательность точек

$$\bar{x}^{-k} = \varphi(\bar{x}^{-k-1}), \quad k=1, 2, \dots$$

Пусть точка \bar{x}^{-0} есть некоторое начальное приближение к решению. Рассмотрим δ -окрестность $U_\delta(\bar{x})$ этой точки. В силу принципа сжимающих отображений итерационная последовательность

$$\bar{x}^{-k} = \varphi(\bar{x}^{-k-1}), \quad k=1, 2, \dots$$

будет сходиться, если существует число $q < 1$ такое, что выполнено условие:

$$\|\varphi(\bar{x}^{-1}) - \varphi(\bar{x}^{-2})\| \leq q \|\bar{x}^{-1} - \bar{x}^{-2}\|, \quad \forall \bar{x}^{-1}, \bar{x}^{-2} \in U_\delta(\bar{x}^{-0}), \quad (4.3)$$

называемое условием сжатия для отображения φ . В частности, это условие всегда выполняется, если векторная функция φ непрерывно дифференцируема и норма матрицы производных функции φ удовлетворяет неравенству:

$$\left\| \frac{\partial \varphi}{\partial x}(\bar{x}) \right\| \leq q$$

во всех точках \bar{x} из δ -окрестности $U_\delta(\bar{x})$ точки \bar{x}^{-0} .

Теорема. Пусть отображение φ является сжатием в $U_\delta(\overline{x^0})$ и пусть

Тогда итерационная последовательность:

с начальной точкой x^{-0} , сходится к решению x^{*-} системы (1). При этом справедлива следующая оценка погрешности:

Отметим, что начальное приближение $x^{(0)}$ выбирают экспериментально. (Например, на основе грубого графического решения системы, если порядок системы не высок. По точкам строят график первого уравнения, потом второго и ищут приблизительно точку их пересечения).

Пусть дана система нелинейных уравнений :

Запишем ее в векторной форме:

Найдем начальное приближение $x^{(0)}$

26

Вместо системы (4.1) будем искать решение соответствующей ей линеаризованной системы

$$f(\bar{x}^0) + \frac{\partial f}{\partial x}(\bar{x}^0)(\bar{x} - \bar{x}^0) = 0 \Rightarrow f(\bar{x}^0) + J(\bar{x}^0)(\bar{x} - \bar{x}^0) = 0,$$

где через $J(\bar{x}^0)$ обозначена для удобства записи матрица производных векторной функции f в точке \bar{x}^0 (матрица Якоби системы (4.1) в этой точке).

При этом при применении метода Ньютона предполагается, что $\det J(\bar{x}^0) \neq 0$ в окрестности точки \bar{x}^0 .

Тогда из линеаризованной системы, которая линейна относительно переменных x , можно найти первое приближение

$$\bar{x}^k = \bar{x}^{k-1} - J^{-1}(\bar{x}^{k-1})f(\bar{x}^{k-1}), k = 1, \dots$$

Рассматривая линеаризованную систему в точках при $k=1, 2, \dots$, найдем k -ое приближение

$$\bar{x}^k = \bar{x}^{k-1} - J^{-1}(\bar{x}^{k-1})f(\bar{x}^{k-1}), k = 1, \dots$$

Построенная таким способом рекуррентная последовательность Ньютона сходится при определенных дополнительных условиях к решению системы (4.1). Легко видеть, что рассматриваемый метод совпадает с методом касательных в случае $n=1$, т.е. является многомерным вариантом метода касательных.

На практике обратную матрицу не считают, а на каждом шаге решают линеаризованную систему:

$$f(\bar{x}^{k-1}) + J(\bar{x}^{k-1})(\bar{x} - \bar{x}^{k-1}) = 0 \Rightarrow \bar{x} = \bar{x}^k$$

Теорема. При сделанных выше предположениях, последовательность Ньютона сходится к решению системы (4.1), если начальное

приближение выбрано достаточно близко к решению.

Отметим в заключение, что метод Ньютона сходится достаточно быстро (скорость сходимости квадратичная), если начальное приближение выбрано удачно. На практике итерационный процесс заканчивают, когда норма разности двух последовательных приближений меньше заданной точности вычисления решения.

ЗАДАНИЕ. Решить систему нелинейных уравнений:

$$\begin{cases} \operatorname{tg}(xy + m) = x \\ ax^2 + 2y^2 = 1, \end{cases} \quad \text{где } x > 0, y > 0,$$

с точностью до 0,0001 методами простых итераций и Ньютона, принимая для номера варианта k значения параметров a и m из таблицы:

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14
m	0,0	0,1	0,1	0,2	0,2	0,3	0,3	0,4	0,4	0,1	0,2	0,3	0,2	0,2
a	0,5	0,6	0,7	0,8	0,9	1,0	0,5	0,6	0,7	0,8	0,9	1,0	0,7	0,5

Начальные приближения найти графически. Сравнить скорость сходимости методов.

ЗАДАНИЕ 5. Вычисление собственных значений и векторов

Цель выполнения задания: освоить методы вычисления собственных значений и векторов.

Краткие теоретические сведения.

Метод Якоби (вращений) использует итерационный процесс, который приводит исходную симметрическую матрицу A к диагональному виду с помощью последовательности элементарных ортогональных преобразований (в дальнейшем называемых вращениями Якоби или плоскими вращениями). Процедура построена таким образом, что на $(k+1)$ -ом шаге осуществляется преобразование вида

$$A^{(k)} \rightarrow A^{(k+1)} = V^{(k)*} A^{(k)} V^{(k)} = V^{(k)*} \dots V^{(0)*} A^{(0)} V^{(0)} \dots V^{(k)}, \quad k=0,1,2,\dots, \quad (5.1)$$

где $A^{(0)} = A$, $V^{(k)} = V^{(k)}_{ij}(\varphi)$ — ортогональная матрица, отличающаяся от единичной матрицы только элементами

$$v_{ii} = v_{jj} = \cos \varphi \quad v_{ij} = -v_{ji} = -\sin \varphi, \quad (5.2)$$

значение φ выбирается при этом таким образом, чтобы обратить в 0 наибольший по модулю недиагональный элемент матрицы $A^{(k)}$. Итерационный процесс постепенно приводит к матрице со значениями недиагональных элементов, которыми можно пренебречь, т.е. матрица $A^{(k)}$ все более похожа на диагональную, а диагональная матрица Λ является пределом последовательности $A^{(k)}$ при $k \rightarrow \infty$.

Алгоритм метода вращений.

1) В матрице $A^{(k)}$ ($k=0,1,2,\dots$) среди всех недиагональных элементов выбираем максимальный по абсолютной величине элемент, стоящий выше главной диагонали; определяем его номера i и j строки и столбца, в которых он стоит (если максимальных элементов несколько, можно взять любой из них);

2) По формулам

$$\cos \varphi_k = \sqrt{\frac{1}{2}(1 + (1 + p_k^2))^{-1}}, \quad \sin \varphi_k = \operatorname{sgn} p_k \sqrt{\frac{1}{2}(1 - (1 + p_k^2))^{-1}},$$

где

$$p_k = 2a_{ij}^{(k)} / (a_{ii}^{(k)} - a_{jj}^{(k)}),$$

вычисляем $\cos \varphi_k$ и $\sin \varphi_k$, получаем матрицу $V^{(k)} = V^{(k)}_{ij}(\varphi_k)$.

3) По формулам

$$\begin{aligned} b_{si} &= a_{si}^{(k)} \cos \varphi_k + a_{sj}^{(k)} \sin \varphi_k, \\ b_{sj} &= -a_{si}^{(k)} \sin \varphi_k + a_{sj}^{(k)} \cos \varphi_k, \quad s = 1, 2, \dots, n, \\ a_{is}^{(k+1)} &= b_{is} \cos \varphi_k + b_{js} \sin \varphi_k, \\ a_{js}^{(k+1)} &= -b_{is} \sin \varphi_k + b_{js} \cos \varphi_k, \quad s = 1, 2, \dots, n. \end{aligned}$$

находим элементы матрицы $A^{(k+1)}$.

3) Итерационный процесс останавливаем, когда в пределах принятой точности суммой квадратов всех недиагональных элементов матрицы $A^{(k+1)}$, обозначаемой $t(A^{(k+1)})$, можно пренебречь.

4) В качестве собственных значений матрицы A берем диагональные элементы матрицы $A^{(k+1)}$, в качестве собственных векторов – соответствующие столбцы матрицы $V = V^{(0)} V^{(1)} \dots V^{(k)}$.

Основное достоинство метода Якоби заключается в том, что при выполнении каждого плоского вращения уменьшается сумма квадратов недиагональных элементов; сходимость этой суммы к нулю по мере увеличения числа шагов гарантирует сходимость процесса диагонализации.

ЗАДАНИЕ 5. С точностью 0,0001 вычислить собственные значения и собственные векторы матрицы A ,
где $A = kC + D$, A – исходная матрица для расчёта, k – номер варианта (0-15), матрицы C, D заданы ниже:

$$C = \begin{bmatrix} 0,2 & 0 & 0,2 & 0 & 0 \\ 0 & 0,2 & 0 & 0,2 & 0 \\ 0,2 & 0 & 0,2 & 0 & 0,2 \\ 0 & 0,2 & 0 & 0,2 & 0 \\ 0 & 0 & 0,2 & 0 & 0,2 \end{bmatrix}, \quad D = \begin{bmatrix} 2,33 & 0,81 & 0,67 & 0,92 & -0,53 \\ 0,81 & 2,33 & 0,81 & 0,67 & 0,92 \\ 0,67 & 0,81 & 2,33 & 0,81 & 0,92 \\ 0,92 & 0,67 & 0,81 & 2,33 & -0,53 \\ -0,53 & 0,92 & 0,92 & -0,53 & 2,33 \end{bmatrix}.$$

ЗАДАНИЕ 6. Интерполяция с помощью интерполяционных многочленов Лагранжа и Ньютона.

Цель задания: изучить интерполяцию функций с помощью интерполяционных многочленов Лагранжа и Ньютона.

Пусть $f(x)$ – функция, непрерывная на отрезке $[a, b]$. Выберем на этом отрезке точки, называемые *узлами интерполяции*:

$$a \leq x_0 < x_1 < \dots < x_n \leq b.$$

Предположим, что известны значения функции в узлах интерполяции:

$$f(x_k) = y_k, \quad k = 0, 1, \dots, n.$$

Ставится задача найти многочлен $P_n(x)$ такой, что

$$P_n(x_k) = y_k, \quad \forall k = 0, 1, \dots, n. \quad (6.1)$$

Такой многочлен $P_n(x)$ называется *интерполяционным многочленом*, а задача его нахождения – *задачей интерполяции*.

Можно показать, что задача интерполяции всегда имеет решение, причем единственное.

Обозначим

$R_n(x) = f(x) - P_n(x)$. Пусть $f(x) \in C^{n+1}[a, b]$. Тогда погрешность интерполяции оценивается по формуле

$$|R_n(x)| \leq \frac{M_{n+1}}{(n+1)!} |\omega(x)|, \quad \text{где} \quad M_{n+1} = \max_{a \leq x \leq b} |f^{(n+1)}(x)|.$$

1) Интерполяционный многочлен Лагранжа

Пусть $\omega(x) = (x - x_0)(x - x_1) \cdot \dots \cdot (x - x_n)$,

$$\omega_j(x) = (x - x_0) \cdot \dots \cdot (x - x_{j-1})(x - x_{j+1}) \cdot \dots \cdot (x - x_n).$$

Положим $l_j(x) = \frac{\omega_j(x)}{\omega_j(x_j)}$,

$$\text{т. е.} \quad l_j(x) = \frac{(x - x_0) \cdot \dots \cdot (x - x_{j-1})(x - x_{j+1}) \cdot \dots \cdot (x - x_n)}{(x_j - x_0) \cdot \dots \cdot (x_j - x_{j-1})(x_j - x_{j+1}) \cdot \dots \cdot (x_j - x_n)}.$$

$$\text{Очевидно} \quad l_j(x_i) = \begin{cases} 0, & \text{при } i \neq j \\ 1, & \text{при } i = j. \end{cases}$$

Построим многочлен $L_n(x) = \sum_{j=0}^n l_j(x) y_j$.

Легко видеть, что $L_n(x_i) = l_i(x_i) y_i = 1 \cdot y_i = y_i$, $i = \overline{0, n}$, т. е. это интерполяционный многочлен. Его называют *интерполяционным многочленом Лагранжа*.

2) Интерполяционный многочлен Ньютона

Пусть x_0, x_1, \dots, x_n – набор узлов интерполирования,

y_0, y_1, \dots, y_n – значения функции $f(x)$ в узлах.

Величину $\Delta y_k = y_{k+1} - y_k$ называют конечной разностью первого порядка в k -ом узле.

Аналогично определяются конечные разности высших порядков.

$$\Delta^2 y_k = \Delta y_{k+1} - \Delta y_k = y_{k+2} - y_{k+1} - (y_{k+1} - y_k) = y_{k+2} - 2y_{k+1} + y_k$$

.....

$$\Delta^i y_k = \Delta^{i-1} y_{k+1} - \Delta^{i-1} y_k = \sum_{i=0}^n (-1)^{n-i} C_n^i y_{k+i} \quad \Delta^i y_k = \Delta^{i-1} y_{k+1} - \Delta^{i-1} y_k = \sum_{i=0}^n (-1)^{n-i} C_n^i y_{k+i}.$$

Конечные разности обычно считают по схеме:

x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$
x_0	y_0	$\Delta y_0 = y_1 - y_0$	$\Delta^2 y_0 = \Delta y_1 - \Delta y_0$	$\Delta^3 y_0 = \Delta^2 y_1 - \Delta^2 y_0$
x_1	y_1	$\Delta y_1 = y_2 - y_1$	$\Delta^2 y_1 = \Delta y_2 - \Delta y_1$	
x_2	y_2	$\Delta y_2 = y_3 - y_2$		
x_3	y_3			

Разделенной разностью первого порядка называется выражение

$$f_1(x_k, x_{k+1}) = \frac{y_{k+1} - y_k}{x_{k+1} - x_k} = \frac{\Delta y_k}{\Delta x_k}.$$

Разделенной разностью второго порядка называется выражение

$$f_2(x_k, x_{k+1}, x_{k+2}) = \frac{f_1(x_{k+1}, x_{k+2}) - f_1(x_k, x_{k+1})}{x_{k+2} - x_k} \text{ и т. д.}$$

Пусть x – любая точка отрезка, не совпадающая с узлами. Тогда

$$f_1(x, x_0) = \frac{y_0 - f(x)}{x_0 - x},$$

$$\text{откуда } f(x) = y_0 + f_1(x, x_0)(x - x_0). \quad (6.2)$$

$$\text{Далее } f_2(x, x_0, x_1) = \frac{f_1(x_0, x_1) - f_1(x, x_0)}{x_1 - x},$$

$$\text{откуда } f_1(x, x_0) = f_1(x_0, x_1) + f_2(x, x_0, x_1)(x - x_1).$$

Подставляя в (6.2), получаем:

$$f(x) = y_0 + f_1(x_0, x_1)(x - x_0) + f_2(x, x_0, x_1)(x - x_0)(x - x_1). \quad (6.3)$$

$$\text{Далее } f_3(x, x_0, x_1, x_2) = \frac{f_2(x_0, x_1, x_2) - f_2(x, x_0, x_1)}{x_2 - x},$$

$$\text{откуда } f_2(x, x_0, x_1) = f_2(x_0, x_1, x_2) + f_3(x, x_0, x_1, x_2)(x - x_2).$$

Подставляя в (6.3), имеем:

$$f(x) = y_0 + f_1(x_0, x_1)(x - x_0) + f_2(x, x_0, x_2)(x - x_0)(x - x_1) + f_3(x, x_0, x_1, x_2)(x - x_0)(x - x_1)(x - x_2). \quad (6.4)$$

Продолжая процесс, получим:

$$f(x) = N_n(x) + f_{n+1}(x, x_0, \dots, x_n)(x - x_0) \dots (x - x_n),$$

$$\text{где } N_n(x) = y_0 + f_1(x_0, x_1)(x - x_0) + \dots + f_n(x_0, \dots, x_n)(x - x_0) \dots (x - x_{n-1}).$$

$$\text{Очевидно при } x = x_i, \quad \forall i = \overline{0, n}, \quad f(x_i) = N_n(x_i), \quad i = \overline{0, n},$$

т. е. $N_n(x)$ – интерполяционный многочлен. Его называют *интерполяционным многочленом Ньютона*.

Достоинство интерполяционного многочлена Ньютона: он удобен при расширении интерполяции и добавлении узлов.

Недостаток: в какой-то степени он сложнее в подсчете конечных разностей по сравнению с многочленом Лагранжа.

ЗАДАНИЕ. Построить интерполяционные многочлены в форме Лагранжа и Ньютона, используя номер варианта к, соответствующие значения параметров m и p_i и значения x_i, y_i из таблиц:

x_i	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
p_i	0.0	0.41	0.79	1.13	1.46	1.76	2.04	2.3	2.55	2.79	3.01

$$y_i = p_i + (-1)^k m$$

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14
m	0	1	1.5	2	2.5	3	3.5	4	4.5	1.8	2.53	3.96	5.33	1.96

Оценить погрешность. Вычислить значение f -ии в точке 0.47 с помощью интерполяционного многочлена и многочлена наилучшего приближения. Сравнить значения.

ЗАДАНИЕ 7. Интерполяция сплайнами

Цель занятия: изучить построение кубических интерполяционных сплайнов.
Краткие теоретические сведения.

Рассмотрим задачу интерполяции функции $f(x)$ на отрезке $[a, b]$. Пусть мы имеем узлы $a = x_0 < x_1 < \dots < x_n = b$ и значения функции y_0, \dots, y_n в данных узлах. Отрезок разбивается узлами на n элементарных отрезков $[x_{i-1}, x_i]$, где $h_i = x_i - x_{i-1}$ — длина элементарного отрезка, $i = \overline{1, n}$.

Сплайном называется функция $S(x)$, которая на каждом элементарном отрезке является многочленом и непрерывна на всем отрезке $[a, b]$, вместе со своими производными до некоторого порядка.

Степенью сплайна называется наивысший порядок степени многочлена.

Дефектом сплайна называется разность между его степенью и наивысшим порядком непрерывной на $[a, b]$ производной.

Пример. Рассмотрим функцию

$$S(x) = \begin{cases} x^2, & 0 \leq x < 1 \\ -x^2 + 4x - 2, & 1 \leq x < 2 \\ 2, & 2 \leq x < 3 \\ \frac{x^3}{27} - x + 4, & 3 \leq x < 4. \end{cases}$$

Очевидно, функция $S(x)$ является кубическим сплайном на отрезке $[0, 4]$, так как она непрерывна в узловых точках.

Действительно,

$$S(1-0) = S(1+0) = 1, \quad S(2-0) = S(2+0) = 2, \quad S(3-0) = S(3+0) = 2.$$

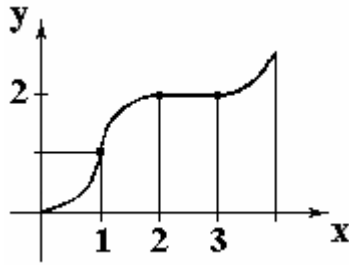


Рис. 7.1.

Найдем дефект сплайна.

$$S'(1-0) = S'(1+0) = 2, \quad S'(2-0) = S'(2+0) = 0, \quad S'(3-0) = S'(3+0) = 0.$$

$$\text{В то же время } S''(2-0) = -2, \quad S''(2+0) = 0.$$

Таким образом, наибольший порядок непрерывной производной функции S на отрезке $[0,4]$ равен 1 и, следовательно, дефект сплайна равен 2. Смотри рисунок 7.1.

Отметим, что в общем случае сам сплайн многочленом не является. Чтобы он был многочленом, необходимо и достаточно, чтобы его дефект равнялся нулю.

Будем рассматривать кубические сплайны, у которых непрерывны первая и вторая производные.

Тогда на отрезке $[x_{i-1}, x_i]$ сплайн $S(x)$ имеет вид

$$S(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3, \quad i = \overline{1, n}.$$

(7.1)

Таким образом, для построения кубического сплайна необходимо найти $4n$ неизвестных коэффициентов многочленов (7.1).

Очевидно $S(x_i) = y_i$, $i = \overline{0, n}$. Найдем $S(x)$. Для этого требуется определить значения $4n$ неизвестных коэффициентов. Очевидно для этого необходимо иметь $4n$ уравнений для определения коэффициентов.

Подставим левый конец отрезка (x_{i-1}) в уравнение:

$$S(x_{i-1}) = y_{i-1} = a_i, \quad i = \overline{1, n}$$

$$S(x_{i+1}) = y_i = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3, \quad i = \overline{1, n}.$$

В итоге получаем $2n$ уравнений:

$$\begin{cases} y_{i-1} = a_i & i = \overline{1, n} \\ a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 = y_i & i = \overline{1, n}. \end{cases}$$

Далее во всех внутренних узлах должны совпадать первая и вторая производные $S(x)$. Имеем

$$S'(x) = b_i + 2c_i(x - x_{i-1}) + 3d_i(x - x_{i-1})^2,$$

$$S''(x) = 2c_i + 6d_i(x - x_{i-1}), \quad i = \overline{1, n-1}.$$

Приравниваем во внутренних узлах значения левых и правых производных. Получим:

$$\begin{cases} b_i + 2c_i h_i + 3d_i h_i^2 = b_{i+1} & i = \overline{1, n}, \\ c_i + 3d_i h_i = c_{i+1}, \end{cases}$$

т. е. $(2n-2)$ уравнений.

Недостающие 2 уравнения можно задать разными способами. Обычно берут $S''(x_0) = S''(x_n) = 0$.

Отсюда

$$2c_1 = 0, \quad 2c_n + 6d_n h_n = 0.$$

Для удобства положим еще $c_{n+1} = 0$.

Объединяя все уравнения, получим систему

$$\begin{cases} y_{i-1} = a_i & i = \overline{1, n} \\ a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 = y_i & i = \overline{1, n} \\ b_i + 2c_i h_i + 3d_i h_i^2 = b_{i+1} & i = \overline{1, n-1} \\ c_i + 3d_i h_i = c_{i+1} & i = \overline{1, n-1} \\ c_n + 3d_n h_n = 0 \\ c_1 = 0 \\ c_{n+1} = 0. \end{cases}$$

Решая систему, получим

$$\begin{cases} b_i h_i + c_i h_i^2 + d_i h_i^3 = y_i - y_{i-1} & i = \overline{1, n} \\ 2c_i h_i + 3d_i h_i^2 = b_{i+1} - b_i & i = \overline{1, n-1} \\ d = \frac{c_{i+1} - c_i}{3h_i} \\ c_1 = c_{n+1} = 0, \end{cases}$$

далее

$$\begin{cases} a_i = y_{i-1} & i = \overline{1, n} \\ b_i h_i + c_i h_i^2 + \left(\frac{(c_{i+1} - c_i) h_i^2}{3} \right) = y_i - y_{i-1} & i = \overline{1, n} \\ 2c_i h_i + (c_{i+1} - c_i) h_i = b_{i+1} - b_i & i = \overline{1, n-1} \\ d = \frac{c_{i+1} - c_i}{3h_i} & i = \overline{1, n} \\ c_1 = c_{n+1} = 0. \end{cases}$$

Откуда

$$b_i = \frac{y_i - y_{i-1}}{h_i} - c_i h_i - \frac{(c_{i+1} - c_i) h_i}{3}, \quad i = \overline{1, n}.$$

$$2c_i h_i + (c_{i+1} - c_i) h_i = \frac{y_{i+1} - y_i}{h_{i+1}} - c_{i+1} h_{i+1} - \frac{(c_{i+2} - c_{i+1}) h_{i+1}}{3} - \frac{y_i - y_{i-1}}{h_i} + c_i h_i + \frac{(c_{i+1} - c_i) h_i}{3}.$$

Таким образом, задача определения коэффициентов сплайна свелась к решению системы

$$c_i \left(\frac{h_i}{3} \right) + c_{i+1} \left(\frac{2}{3} h_i + \frac{2}{3} h_{i+1} \right) + c_{i+2} \left(\frac{h_{i+1}}{3} \right) = \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}, \quad i = \overline{1, n-1} \quad (7.2)$$

$$c_1 = c_{n+1} = 0.$$

Это трехдиагональная система и ее целесообразно решать методом прогонки. Поскольку для матрицы системы выполнено условие доминирования диагональных элементов

$$\frac{2}{3} h_i + \frac{2}{3} h_{i+1} > \frac{h_i}{3} + \frac{h_{i+1}}{3},$$

Проводя обратный ход метода прогонки, последовательно найдем значения неизвестных x_n, x_{n-1}, \dots, x_1 .

ЗАДАНИЕ. Произвести интерполирование кубическими сплайнами приведенных в таблице функций. Вычислить значение сплайна в точке $x = 0.5 * (b - a)$.

Значение сплайна в точке $x = 0.5 * (b - a)$ записать в качестве ответа. Сравнить его со значением функции в соответствующей точке.

№ варианта	Функция $f(x)$	Интервал $[a, b]$	Число узлов	Значение в точке $x = 0.5 * (b - a)$
1.	e^{-x}	[0,4]	5	0,1372
2.	$\ln(x)$	[1,3]	6	0
3.	\sqrt{x}	[0,4]	5	1,4065
4.	$1/x$	[1,2]	6	1
5.	$sh(x)$	[0,2]	6	1,1752
6.	$arctg(x)$	[0,2]	6	0,7854
7.	$ch(x)$	[0,2]	6	1,5431
8.	$th(x)$	[0,2]	6	0,7616
9.	$1/\sqrt{x}$	[1,3]	6	1
10.	$tg(x)$	[0,1.5]	6	0,9316

ЗАДАНИЕ 8. Численное дифференцирование и интегрирование функций

Цель работы: Изучить методы численного вычисления производных и методы численного интегрирования. Сравнить методы по трудоемкости, точности. Выполнить тестовое задание по численному дифференцированию и интегрированию.

Краткие теоретические сведения.

1) Численное дифференцирование. Для получения формул вычисления производной разобьем отрезок $[a, b]$ на n частей следующим образом:

$$h = \frac{b-a}{n}, \quad a = x_0 < x_1 < \dots < x_n = b.$$

Тогда $y_k = f(x_k)$, $y'_k = f'(x_k)$, и по формуле Тейлора (считая функцию дважды непрерывно дифференцируемой) получаем

$$y_{k+1} = y_k + y'_k h + f''(\xi) h^2 \frac{1}{2},$$

или

$$y'_k = \frac{y_{k+1} - y_k}{h} - \frac{f''(\xi)h}{2};$$

где ξ – некоторая точка на $[x_k, x_{k+1}]$.

Таким образом получаем формулу для приближенного вычисления производной :

$$y'_k \approx \frac{y_{k+1} - y_k}{h}, \text{ с погрешностью } R \leq \frac{M_2 h}{2},$$

где $M_2 = \max_{a \leq x \leq b} |f''(x)|$.

Таким образом, обеспечивается точность $O(h)$.

Далее воспользуемся следующей теоремой.

Теорема(о среднем).

Пусть функция $f(x)$ непрерывна на отрезке $[a, b]$ и $x_1, \dots, x_n \in [a, b]$. Тогда \exists точка $\xi \in [a, b]$ такая, что $\frac{f(x_1) + \dots + f(x_n)}{n} = f(\xi)$.

Считая функцию трижды непрерывно дифференцируемой, получим:

$$y_{k+1} = y_k + y'_k h + \frac{y''_k}{2} h^2 + \frac{f'''(\xi_1)}{6} h^3;$$

$$y_{k-1} = y_k - y'_k h + \frac{y''_k}{2} h^2 - \frac{f'''(\xi_2)}{6} h^3.$$

Отсюда можем определить производную как

$$y'_k = \frac{y_{k+1} - y_{k-1}}{2h} + \frac{f'''(\xi_1) + f'''(\xi_2)}{12} h^2 \text{ и, применяя теорему о среднем, получаем:}$$

$$y'_k = \frac{y_{k+1} - y_{k-1}}{2h} + \frac{f'''(\xi_k)}{6} h^2.$$

Т.е. имеет место формула для приближенного вычисления производной:

$$y'_k \approx \frac{y_{k+1} - y_{k-1}}{2h}, \text{ } R \leq \frac{M_3 h^2}{6}, \text{ где } M_3 = \max_{a \leq x \leq b} |f'''(x)|.$$

Точность вычисления производной в этом случае имеет порядок $O(h^2)$.

Для того чтобы найти формулу для вычисления второй производной будем считать функцию $f(x)$ четырежды непрерывно дифференцируемой, тогда:

$$y_{k+1} = y_k + y'_k h + \frac{y''_k}{2} h^2 + \frac{y'''_k}{6} h^3 + \frac{f^{IV}(\xi_1)}{24} h^4;$$

$$y_{k-1} = y_k - y'_k h + \frac{y''_k}{2} h^2 - \frac{y'''_k}{6} h^3 + \frac{f^{IV}(\xi_2)}{24} h^4,$$

отсюда $\frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} = y''_k + \frac{f^{IV}(\xi_1) + f^{IV}(\xi_2)}{24} h^2,$

значит $y''_k \approx \frac{y_{k+1} - 2y_k + y_{k-1}}{h^2}, \quad R = \frac{|f^{IV}(\xi_k)|}{12} h^2 \leq \frac{M_4 h^2}{12}$

При этом обеспечивается точность $O(h^2)$.

2) Интегрирование функций. Пусть дана функция $f(x)$, которую необходимо проинтегрировать на отрезке $[a, b]$. Разобьем этот отрезок на n частей следующим образом:

$$h = \frac{b-a}{n}, \quad a = x_0 < x_1 < \dots < x_n = b, \quad \text{и зафиксируем значения функции в точках}$$

разбиения y_0, y_1, \dots, y_n .

Тогда $\int_a^b f(x) dx = \sum_{k=1}^n \int_{x_{k-1}}^{x_k} f(x) dx$ и, полагая $\int_{x_{k-1}}^{x_k} f(x) dx \approx y_{k-1} h$,

можно получить формулы:

$$\int_a^b f(x) dx \approx h(y_1 + y_2 + \dots + y_n) \quad (\text{правых прямоугольников});$$

$$\int_a^b f(x) dx \approx h(y_0 + y_1 + \dots + y_{n-1}) \quad (\text{левых прямоугольников});$$

$$\int_a^b f(x) dx \approx h(f(x_0 + h/2) + \dots + f(x_{n+1} + h/2)) \quad (\text{средних прямоугольников}).$$

Проанализируем точность наиболее точной из них формулы средних прямоугольников.

Пусть $\Phi(x) = \int_{x_{k-1}+h/2}^{x_k} f(x) dx \Rightarrow \int_{x_{k-1}}^{x_k} f(x) dx = \Phi(x_k) - \Phi(x_{k-1}). \quad (*)$

Считая исходную функцию дважды непрерывно дифференцируемой, получаем

$$\begin{aligned}\Phi(x_k) &= \Phi(x_{k-1} + h/2) + \Phi'(x_{k-1} + h/2) \frac{h}{2} + \Phi''(x_{k-1} + h/2) \frac{h^2}{8} + \frac{\Phi'''(\xi_1)}{6} \frac{h^3}{8} = \\ &= f(x_{k-1} + h/2) \frac{h}{2} + f'(x_{k-1} + h/2) \frac{h^2}{8} + f'''(\xi_1) \frac{h^3}{48}\end{aligned}$$

$$\text{Значит } \Phi(x_{k-1}) = -f(x_{k-1} + h/2) \frac{h}{2} + f'(x_{k-1} + h/2) \frac{h^2}{8} - f'''(\xi_2) \frac{h^3}{48}$$

Полученные значения подставим в (*) и приведем подобные:

$$\int_{x_{k-1}}^{x_k} f(x) dx = \Phi(x_k) - \Phi(x_{k-1}) = f(x_{k-1} + h/2)h + \frac{f'''(\xi_1) + f'''(\xi_2)}{48} h^3;$$

Таким образом

$$\int_a^b f(x) dx = \sum_{k=1}^n f(x_{k-1} + h/2)h + \sum_{k=1}^n f'''(\xi_k) \frac{h^3}{24} = h \sum_{k=1}^n f(x_{k-1} + h/2) + \frac{f'''(\xi)nh^3}{24}, \quad R \leq \frac{M_2 nh^3}{24};$$

То есть оценка точности для данного метода $O(h^2)$.

Используя формулы правых и левых прямоугольников (взяв их среднее арифметическое) получим *формулу трапеций*:

$$\int_a^b f(x) dx \approx h \left(\frac{y_0 + y_n}{2} + y_1 + \dots + y_{n-1} \right), \quad R \leq \frac{M_2 nh^2}{12}.$$

Можно показать, что ее точность тоже $O(h^2)$.

Формула Симпсона.

Дана функция $f(x)$, которую необходимо проинтегрировать на отрезке $[a, b]$. Разобьем этот отрезок на $2n$ частей следующим образом:

$$h = \frac{b-a}{2n}, \quad a = x_0 < x_1 < \dots < x_n = b. \quad \int_a^b f(x) dx = \sum_{k=0}^{n-1} \int_{x_{2k}}^{x_{2k+2}} f(x) dx$$

Будем аппроксимировать элементарную трапецию некоторой параболической трапецией (например $y = ax^2 + bx + c$) через точки вида

$$A(x_{2k}, y_{2k}), B(x_{2k+1}, y_{2k+1}), C(x_{2k+2}, y_{2k+2});$$

$$\text{Составим систему: } \begin{cases} y_{2k} = ax_{2k}^2 + bx_{2k} + c \\ y_{2k+1} = ax_{2k+1}^2 + bx_{2k+1} + c \\ y_{2k+2} = ax_{2k+2}^2 + bx_{2k+2} + c \end{cases} \quad (**)$$

Посчитаем определитель: $\Delta = \begin{vmatrix} x_{2k}^2 & x_{2k} & 1 \\ x_{2k+1}^2 & x_{2k+1} & 1 \\ x_{2k+2}^2 & x_{2k+2} & 1 \end{vmatrix} \neq 0$, значит данная система имеет

решение a, b, c и причем единственное.

Решая ее мы получим так называемую малую формулу Симпсона:

$$\int_{x_{2k}}^{x_{2k+2}} (ax^2 + bx + c)dx = \dots = \frac{h}{3}(y_{2k} + 4y_{2k+1} + y_{2k+2}), \text{ где } a, b \text{ и } c \text{ берутся из}$$

системы (**).

По сдвоенному элементарному промежутку запишем

$$\begin{aligned} \int_a^b f(x)dx &= \sum_{k=0}^{n-1} \int_{x_{2k}}^{x_{2k+2}} f(x)dx = \sum_{k=0}^{n-1} \frac{h}{3}(y_{2k} + 4y_{2k+1} + y_{2k+2}) = \\ &= \frac{h}{3}(y_0 + 2(y_2 + \dots + y_{2n-2}) + 4(y_1 + \dots + y_{2n-1}) + y_{2n}) \end{aligned}$$

Таким образом, получается формула Симпсона:

$$\int_a^b f(x)dx \approx \frac{h}{3}[(y_0 + y_{2n}) + 2(y_2 + \dots + y_{2n-2}) + 4(y_1 + \dots + y_{2n-1})], \quad R \leq \frac{(b-a)^4 h^4 M_4}{180};$$

Данная формула обеспечивает точность вычислений $O(h^4)$.

Анализируя рассмотренные методы численного интегрирования мы можем сделать вывод, что расчет по формуле Симпсона является наиболее точным.

Задание. В каждом варианте найти численное значение первой и второй производной в точке, являющейся серединой заданного интервала, с точностью до 0,01. Вычислить с заданной точностью интегралы по формулам прямоугольников, трапеций, Симпсона. Сравнить методы по точности.

№ вар и- ант а	Функция $f(x)$	Интервал	Метод	Точность	Значение интеграла
1.	$\lg x$	[1 ; 3]	Трапеций	0,000001	0,5627743
2.	$\arctg x$	[0 ; 2]	Симпсона	0,000001	1,4095786

3.	$ch x$	$[0 ; 2]$	Средних	0,000001	0,4995949
4.	\sqrt{x}	$[0 ; 4]$	Симпсона	0,000001	5,3333335
5.	$\sqrt{tg x}$	$[0 ; 1.5]$	Трапеций	0,000001	1,6893633
6.	$\sqrt{1 - lg^2 x}$	$[1 ; 10]$	Средних	0,000001	5,7980142
7.	$arctg \sqrt{x}$	$[0 ; 2]$	Симпсона	0,000001	1,4517355
8.	$\frac{\sin x}{x}$	$[1 ; 2]$	Трапеций	0,000001	0,6593294
9.	$\frac{\cos x}{x}$	$[1 ; 2]$	Симпсона	0,000001	0,0855770
10.	$sh(1/x)$	$[1 ; 2]$	Средних	0,000001	0,7576327

ЗАДАНИЕ 9. Методы Эйлера и Рунге-Кутта

Цель выполнения задания: Изучить решение задачи Коши для обыкновенных дифференциальных уравнений методом Эйлера и методом Рунге-Кутта.

Краткие теоретические сведения.

Рассмотрим дифференциальное уравнение $y' = f(x, y)$ с начальным условием $y(x_0) = y_0$. Будем предполагать, что $f(x, y)$ непрерывная и непрерывно дифференцируемая по y функция в окрестности замкнутой области

$$D = \{(x, y) \mid a \leq x \leq b, c \leq y \leq d\},$$

содержащей внутри себя точку (x_0, y_0) .

Требуется решить задачу Коши: найти непрерывно дифференцируемую функцию $y=y(x)$, такую что $y'(x) = f(x, y(x))$ при всех $x \in [a, b]$ и $y(x_0) = y_0$.

Разобьем отрезок $[a, b]$ с помощью точек разбиения $a = x_0, x_1, \dots, x_n = b$ с шагом $h = (b - a) / n$. Тогда узлы разбиения имеют вид $x_k = x_0 + kh, k = \overline{0, n}$.

Пусть $y(x_0), y(x_1), \dots, y(x_n)$ – значения функции в точках разбиения.

1) Метод Эйлера

Построим рекуррентную последовательность:

$$y_{k+1} = y_k + hf(x_k, y_k), \quad k = 0, 1, \dots$$

(9.1)

$$y_0 = y(x_0),$$

которую называют последовательностью Эйлера. Соединяя ломаными все точки (x_k, y_k) , полученные из рекуррентной последовательности Эйлера, получим ломаную линию, приближающую график решения $y = y(x)$.

Функция, график которой совпадает с ломаной Эйлера, принимается за приближенное решение задачи Коши.

Точность метода Эйлера на всем отрезке $[a, b]$ будет $O(h)$.

Для повышения точности вычислений иногда используется модифицированный метод Эйлера, в котором рекуррентная последовательность Эйлера вычисляется по формулам

$$y_{k+1} = y_k + hf\left(x_k + \frac{h}{2}, y_k + \frac{h}{2}f(x_k, y_k)\right), \quad k = 0, 1, \dots, n-1. \quad (9.2)$$

Модифицированный метод Эйлера обычно дает более точное приближение решения.

Пример. Пусть требуется решить задачу Коши:

$$\begin{cases} y' = -y, & x \in [0, 1] \\ y(0) = 1. \end{cases}$$

Полагая $h = 0,2$ и используя метод Эйлера, получим, как легко убедиться, из формулы Эйлера (9.1)

$$y_{k+1} = y_k + 0.2 \cdot (-y_k) = 0.8 \cdot y_k.$$

С другой стороны, используя модифицированный метод Эйлера, получим в силу формулы (2) рекуррентную последовательность

$$y_{k+1} = y_k + 0.2 \cdot (-y_k) = 0.82 \cdot y_k.$$

Поскольку точным решением задачи Коши, как легко проверить, является функция $y = e^{-x}$, можно сравнить точность обоих методов.

	0	1	2	3	4	5
x_k	0	0.2	0.4	0.6	0.8	1
y_k	1	0.8	0.64	0.572	0.4086	0.3277
$y_k^{\text{модиф}}$	1	0.82	0.6724	0.5514	0.4521	0.3708
e^{-x}	1	0.8187	0.6703	0.5488	0.4493	0.3679

Общепризнанным недостатком метода Эйлера является его не достаточно высокая точность. Несомненным достоинством метода Эйлера является его простота.

2) Метод Рунге-Кутты четвертого порядка.

На каждом шаге производится вычисление коэффициентов K_1, K_2, K_3, K_4 :

$$K_1 = hf(x_k, y_k);$$

$$K_2 = hf(x_k + \frac{h}{2}, y_k + \frac{K_1}{2});$$

$$K_3 = hf(x_k + \frac{h}{2}, y_k + \frac{K_2}{2});$$

$$K_4 = hf(x_k + h, y_k + K_3).$$

Затем вычисляем

$$y_{k+1} = y_k + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4).$$

Данный метод имеет точность $O(h^4)$ на $[a, b]$.

Рассмотрим пример, который мы использовали для иллюстрации точности метода Эйлера.

Пример. Требуется решить задачу Коши:

$$\begin{cases} y' = -y \\ y(0) = 1 \end{cases} \text{ на отрезке } [0, 1].$$

Выберем шаг $h = 0,2$. Результат вычислений поместим в таблицу.

	0	1	2	3	4	5
x_k	0	0.2	0.4	0.6	0.8	1
y_k	1	0.8187	0.6703	0.5487	0.4493	0.3678
e^{-x}	1	0.8187	0.6703	0.5488	0.4493	0.3679

Таким образом, метод Рунге-Кутта 4-го порядка отличается очень высокой точностью. К определенным его недостаткам относится большая сложность и трудоемкость (на каждом шаге необходимо четырежды вычислять значения функции f вместо одного раза в методе Эйлера).

Отметим, что на практике выбирают начальную длину шага h таким образом, чтобы $h^4 < \varepsilon$, где ε – заданная точность вычисления решения. Затем шаг выбирают вдвое меньшим и останавливают вычисления, если разность полученных значений y_k со значениями, полученными при начальном выборе шага меньше ε . В противном случае шаг еще раз уменьшают вдвое и т.д.

ЗАДАНИЕ. С помощью метода Эйлера, а затем метода Рунге-Кутта найти с точностью до 0.001 решения следующих уравнений на отрезке $[0; 1]$.

$$y' = \frac{a(1-y^2)}{(1+m)x^2 + y^2 + 1}, \quad y(0) = 0,$$

где значения параметров a и m принимают следующие значения для вариантов k .

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14
<i>m</i>	1.0	1.5	2.0	1.0	1.5	2.0	1.0	1.5	2.0	1.0	1.5	2.0	1.0	2.0
<i>a</i>	0.5	0.7	0.9	1.1	1.3	0.5	0.7	0.9	1.1	1.3	0.5	0.7	0.9	1.0

Шаг интегрирования h , обеспечивающий требуемую точность, выбирать в процессе вычисления из сравнения результатов, полученных с h и $h/2$. В случае необходимости шаг h должен быть уменьшен. Сравнить результаты.

ЗАДАНИЕ 10. Метод Адамса 2-го порядка

Цель выполнения задания: Изучить численное решение Задачи Коши для обыкновенных дифференциальных уравнений методом Адамса.

Краткие теоретические сведения.

Пусть есть дифференциальное уравнение

$$y' = f(x, y),$$

с начальным условием

$$y(x_0) = y_0.$$

Разбиваем отрезок $[a, b]$ с шагом h на n частей. То есть, получаем узлы $x_k = x_0 + kh$, $k = \overline{0, n}$, где $x_0 = a$.

Пусть $y = y(x)$ – решение. Тогда на $[x_k, x_{k+1}]$ справедливо равенство

$$y(x_{k+1}) = y(x_k) + \int_{x_k}^{x_{k+1}} f(x, y(x)) dx.$$

Применим формулу левых прямоугольников для вычисления интеграла. Получим

$$y_{k+1} = y_k + hf(x_k, y_k), \text{ то есть формулу Эйлера.}$$

Очевидно, то не самый точный метод вычисления интеграла.

Используем интерполяционную квадратурную формулу Лагранжа для вычисления интеграла, т.е.

$$\int_{x_K}^{x_{K+1}} f(x, y(x)) dx = A_0 f(x_k, y_k) + A_1 f(x_{k-1}, y_{k-1}), \text{ где}$$

$$\int_a^b f(x) dx = \sum_{i=0}^m f(x_i) A_i, \quad A_i = \int_a^b l_i(x) dx; \quad l_i(x) = \frac{\varpi_i(x)}{\varpi_i(x_i)}.$$

Найдем коэффициенты A_i методом неопределенных коэффициентов:

$$\int_{x_K}^{x_{K+1}} dx = A_0 + A_1;$$

$$\int_{x_K}^{x_{K+1}} x dx = A_0 x_k + A_1 x_{k-1}.$$

Получаем систему двух уравнений с двумя неизвестными

$$\begin{cases} A_0 = h - A_1 \\ \frac{h(x_{k+1} + x_k)}{2} = A_0 x_k + A_1 x_{k-1}. \end{cases}$$

Откуда

$$\frac{h(x_{k+1} + x_k)}{2} = (h - A_1)x_k + A_1 x_{k-1};$$

$$\frac{h(x_{k+1} + x_k)}{2} = hx_k - A_1(x_k - x_{k-1});$$

$$\frac{h(x_{k+1} + x_k)}{2} = hx_k - A_1 h;$$

$$A_1 h = hx_k - \frac{h(x_{k+1} + x_k)}{2} = \frac{h(x_k - x_{k+1})}{2}.$$

В итоге получим:

$$A_1 = -\frac{h}{2};$$

$$A_0 = h - A_1 = \frac{3h}{2}.$$

Откуда

$$\int_{x_k}^{x_{k+1}} f(x, y(x)) dx = \frac{3}{2} h f_k - \frac{h}{2} f_{k-1}, \quad \text{где } f_k = f(x_k, y_k).$$

Следовательно, получим

$$y_{k+1} = y_k + h \left(\frac{3}{2} f(x_k, y_k) - \frac{1}{2} f(x_{k-1}, y_{k-1}) \right) \quad k = \overline{1, n}.$$

Это формула Адамса второго порядка, которая используется для выполнения задания.

Существенным недостатком метода Адамса второго порядка является то обстоятельство, что для его применения надо знать дополнительно к начальному условию еще

$$y_{-1} = y(x_0 - h) \text{ или } y_1 = y(x_0 + h).$$

Достоинством метода является то, что значение функции f в каждой точке (x_k, y_k) вычисляется только один раз.

ЗАДАНИЕ. Найти с точностью до 0.001 решения следующих уравнений на отрезке $[0; 1]$

$$y' = \frac{a(1-y^2)}{(1+m)x^2 + y^2 + 1}, \quad y(0) = 0,$$

где значения параметров a и m принимают следующие значения для вариантов k .

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14
m	1.0	1.5	2.0	1.0	1.5	2.0	1.0	1.5	2.0	1.0	1.5	2.0	1.0	2.0
a	0.5	0.7	0.9	1.1	1.3	0.5	0.7	0.9	1.1	1.3	0.5	0.7	0.9	1.0