

**Р.Я.Кучумов, В.Р.Сыртланов,
Н.Г.Мусакаев**

**ЛАБОРАТОРНЫЙ ПРАКТИКУМ
ПО КУРСУ
“ЧИСЛЕННЫЕ МЕТОДЫ”**

Тюмень 1999

**Р.Я.Кучумов, В.Р.Сыртланов,
Н.Г.Мусакаев**

**ЛАБОРАТОРНЫЙ ПРАКТИКУМ
ПО КУРСУ
“ЧИСЛЕННЫЕ МЕТОДЫ”**

**Тюмень 1999
«Вектор-Бук»**

УДК 519.6(076.5)

Лабораторный практикум по курсу “Численные методы”:
/Р.Я.Кучумов, В.Р.Сыртланов, Н.Г.Мусакаев /. Под ред. профессора
Р.Я.Кучумова – Тюмень: Издательство «Вектор-Бук», 1999. – 109 с.

Лабораторный практикум представляет собой сборник упражнений по основным разделам курса “Численные методы”. Описываются наиболее распространенные алгоритмы и методы их реализации на ЭВМ.

Предназначено для студентов, обучающихся по специальности 01.02.00 – “Прикладная математика”. Может быть полезен студентам инженерных специальностей, аспирантам и научным работникам, чья работа связана с практическим применением ЭВМ.

Рецензенты: Старший научный сотрудник ТФ ИТПМ СО РАН
Родионов С.П., к.ф.-м.н.; доцент кафедры прикладной
математики ТюмГНГУ Подкорытова С.В., к.т.н.

ПРЕДИСЛОВИЕ

Характерной чертой современного этапа научно-технического прогресса является широкое применение ЭВМ во всех сферах человеческой деятельности и, в первую очередь, в тех отраслях, которые нуждаются в решении сложных инженерно-математических задач.

Разумное использование современной вычислительной техники невозможно без умелого применения численных методов. Для успешного усвоения данного материала и последующего его применения при решении прикладных инженерных задач авторами разработан лабораторный практикум по численным методам.

Содержание практикума традиционное – интерполяция и аппроксимация, численное интегрирование, прямые и итерационные методы решения систем линейных алгебраических уравнений, численное решение нелинейных уравнений, разностные методы решения задачи Коши и краевых задач для обыкновенных дифференциальных уравнений, численные методы решения уравнений математической физики.

Лабораторный практикум имеет следующую структуру. В начале каждой лабораторной работы даются сжатые теоретические сведения: постановка исходной задачи, численные схемы, оценки погрешности. Практически для каждого вычислительного алгоритма приводится принцип его построения и разбирается характерный пример. В каждой лабораторной работе рассматривается, как правило, несколько схем численного решения проблемы; начинается изложение материала с более простых для понимания методов, затем происходит переход к более сложным. Авторами рассматриваются только те методы, которые испытаны практикой и применяются для решения реальных задач. В конце лабораторной работы предлагаются задачи для самостоятельных упражнений. В приложении приведены программы на языке Паскаль, реализующие один из численных методов по каждой лабораторной работе.

Для более полного понимания материала и его успешного применения требуется знание алгебры, математического анализа, обыкновенных дифференциальных уравнений, уравнений математической физики, а также основ информатики.

Выполнение лабораторной работы предусматривает наличие следующих пунктов: название и цель работы, постановка задачи, схемы алгоритмов и программ, протоколы результатов решения задач, интерпретация полученного решения.

Лабораторная работа 1. Интерполирование функций

Цель работы: изучение методов интерполирования функций, сравнительный анализ рассмотренных методов, практическое интерполирование функций на ЭВМ.

Постановка задачи

Пусть функция $y = f(x)$ задана таблицей:

$$y_0 = f(x_0), y_1 = f(x_1), \dots, y_n = f(x_n).$$

Задача интерполирования ставится обычно в следующей форме: найти многочлен $P(x) = P_n(x)$ степени не выше n , значения которого в точках x_i ($i = 0, 1, \dots, n$), называемых узлами интерполяции, совпадают со значениями данной функции, т.е. $P(x_i) = y_i$.

Геометрически это означает, что нужно найти алгебраическую кривую вида $y = a_0x^n + a_1x^{n-1} + \dots + a_n$, проходящую через заданную систему точек $M_i(x_i, y_i)$ ($i = 0, 1, \dots, n$) (рис. 1.1).

В такой постановке задача интерполирования называется параболической. Многочлен $P(x)$ называется интерполяционным многочленом.

Доказано, что в указанной постановке задача интерполирования всегда имеет единственное решение. Интерполяционные формулы обычно используются при нахождении неизвестных значений $f(x)$ для промежуточных значений аргумента. При этом различают интерполирование в узком смысле, когда x находится между x_0 и x_n , и экстраполирование, когда x находится вне отрезка $[x_0, x_n]$.

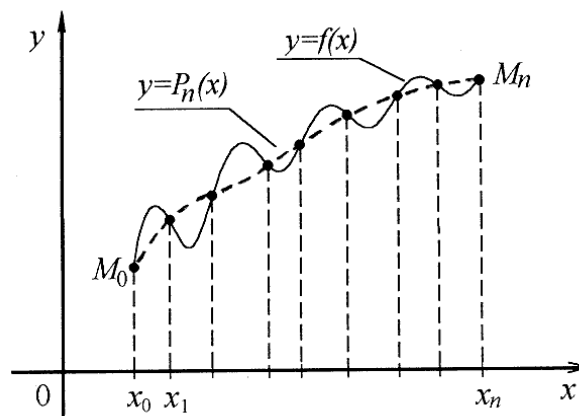


Рис. 1.1

При оценке погрешности результатов должны учитываться как погрешность метода интерполяции (остаточный член), так и погрешности округления при вычислениях.

Интерполяционная формула Лагранжа. Пусть x_i ($i = 0, 1, \dots, n$) – произвольные узлы, а $y_i = f(x_i)$ – значения функции $f(x)$. Многочленом степени n , принимающим в точках x_i значения y_i , является интерполяционный многочлен Лагранжа

$$P_n(x) = \frac{(x-x_1)(x-x_2)\dots(x-x_n)}{(x_0-x_1)(x_0-x_2)\dots(x_0-x_n)} y_0 + \\ + \frac{(x-x_0)(x-x_2)\dots(x-x_n)}{(x_1-x_0)(x_1-x_2)\dots(x_1-x_n)} y_1 + \dots + \\ + \frac{(x-x_0)(x-x_1)\dots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\dots(x_n-x_{n-1})} y_n.$$

Остаточный член равен

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x-x_0)(x-x_1)\dots(x-x_n),$$

где ξ есть некоторая точка наименьшего промежутка, содержащего все узлы x_i ($i = 0, 1, \dots, n$) и точку x .

Выражения

$$P_i^{(n)}(x) = \frac{(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)} \quad (1.1)$$

называются коэффициентами Лагранжа

i	0	1	2	3
x_i	0	0,1	0,3	0,5
y_i	-0,5	0	0,2	1

Пример 1.1. Написать интерполяционный многочлен Лагранжа для функции $f(x)$, значения которой даны таблицей.

Решение. По формуле (1.1)

при $n=3$ получим выражения $P_i^{(3)}(x)$ при $i=0, 2, 3$:

$$P_0^{(3)}(x) = \frac{(x-0,1)(x-0,3)(x-0,5)}{(-0,1)(-0,3)(-0,5)} = -\frac{x^3 - 0,9x^2 + 0,23x - 0,015}{0,015},$$

$$P_2^{(3)}(x) = \frac{x(x-0,1)(x-0,5)}{0,3 \cdot 0,2 \cdot (-0,2)} = -\frac{x^3 - 0,6x^2 + 0,05x}{0,012},$$

$$P_3^{(3)}(x) = \frac{x(x-0,1)(x-0,3)}{0,5 \cdot 0,4 \cdot 0,2} = \frac{x^3 - 0,4x^2 + 0,03x}{0,04}$$

$(P_1^{(3)}(x))$ в данном случае вычислять не следует, т.к. $y_1 = 0$).

Тогда искомым многочлен будет иметь вид

$$P_3(x) = \sum_{i=0}^3 y_i P_i^{(3)}(x) = \frac{125}{3}x^3 - 30x^2 + \frac{73}{12}x - 0,5.$$

Интерполяционные формулы Ньютона. Узлы интерполяции называются равноотстоящими, если

$$x_{i+1} - x_i = \Delta x_i = h, \quad (i = 0, 1, \dots, n-1).$$

Конечными разностями функции $y = f(x)$ называются разности вида

$\Delta y_i = y_{i+1} - y_i$ – конечные разности первого порядка,

$\Delta^2 y_i = \Delta y_{i+1} - \Delta y_i$ – конечные разности второго порядка,

.....

$\Delta^k y_i = \Delta^{k-1} y_{i+1} - \Delta^{k-1} y_i$ – конечные разности k -го порядка.

Ниже дается таблица конечных разностей при $n=5$ (табл. 1.1).

Первая интерполяционная формула Ньютона имеет вид

$$y(x) = P_n(x) = y_0 + q \Delta y_0 + \\ + \frac{q(q-1)}{2!} \Delta^2 y_0 + \dots + \frac{q(q-1) \dots (q-n+1)}{n!} \Delta^n y_0, \quad (1.2)$$

где $q = (x - x_0)/h$.

Заметим, что в формуле используется верхняя горизонтальная строка табл. 1.1. Остаточный член $R_n(x)$ формулы (1.2) имеет вид

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} h^{n+1} q(q-1) \dots (q-n). \quad (1.3)$$

При наличии дополнительного узла x_{n+1} на практике пользуются более удобной приближенной формулой

$$R_n(x) \approx \frac{\Delta^{n+1} y_0}{(n+1)!} q(q-1) \dots (q-n).$$

Последняя формула полезна, например, в случае эмпирически заданных функций.

Число n желательно выбирать так, чтобы разности $\Delta^n y_i$, были практически постоянными (см. пример 1.2).

Формула (1.2) используется для интерполирования и

экстраполирования в точках x , близких к началу таблицы x_0 .

Вторая интерполяционная формула Ньютона имеет вид

$$y(x) = P_n(x) = y_n + q \Delta y_{n-1} + \frac{q(q+1)}{2!} \Delta^2 y_{n-2} + \dots + \frac{q(q+1) \dots (q+n-1)}{n!} \Delta^n y_0, \quad (1.4)$$

где $q = (x - x_n)/h$.

В формуле (1.4) используется нижняя наклонная строка разностей (см. табл. 1.1). Остаточный член $R_n(x)$ формулы (1.4) имеет вид

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} h^{n+1} q(q+1) \dots (q+n).$$

Формула (1.4) используется для интерполирования и экстраполирования в точках x , близких к концу таблицы, т.е. к x_n .

Пример 1.2. По данной таблице значений функции $y = \lg x$ (табл.1.2) найти $\lg 1001$.

Решение. Значение 1001 близко к началу таблицы, поэтому для интерполирования используем формулу (1.2). Составляем таблицу разностей (табл. 1.2), и замечаем, что третьи разности практически постоянны. В этой связи в формуле (1.2) достаточно взять $n=3$.

Таблица 1.1

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
x_0	y_0	Δy_0	$\Delta^2 y_0$	$\Delta^3 y_0$	$\Delta^4 y_0$
x_1	y_1	Δy_1	$\Delta^2 y_1$	$\Delta^3 y_1$	$\Delta^4 y_1$
x_2	y_2	Δy_2	$\Delta^2 y_2$	$\Delta^3 y_2$	
x_3	y_3	Δy_3	$\Delta^2 y_3$		
x_4	y_4	Δy_4			
x_5	y_5				

Таблица 1.2

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$
1000	3,0000000	0,0043241	-0,0000426	0,0000008
1010	3,0043214	0,0042788	-0,0000418	0,0000009
1020	3,0086002	0,0042370	-0,0000409	0,0000008
1030	3,0128372	0,0041961	-0,0000401	
1040	3,0170333	0,0041560		
1050	3,0211893			

$$y(x) = y_0 + q \Delta y_0 + \frac{q(q-1)}{2!} \Delta^2 y_0 + \frac{q(q-1)(q-2)}{3!} \Delta^3 y_0.$$

Для $x=1001$ имеем $q = (1001 - 1000)/10 = 0,1$.

$$\begin{aligned} \lg 1001 &= 3,0000000 + 0,1 \cdot 0,0043214 + \frac{0,1 \cdot 0,9}{2} 0,0000426 + \\ &+ \frac{0,1 \cdot 0,9 \cdot 1,9}{2} 0,0000008 = 3,0004341. \end{aligned}$$

Оценим остаточный член. По формуле (1.3) при $n=3$ имеем

$$R_3(x) = \frac{f^{(4)}(\xi)}{4!} h^4 q(q-1)(q-2)(q-3),$$

где $1000 < \xi < 1030$.

В нашем случае $f^{(4)}(x) = -\frac{3!}{x^4} \lg e$, поэтому $\left| f^{(4)}(\xi) \right| \leq -\frac{3!}{(1000)^4} \lg e$.

Окончательно получаем

$$\left| R_3(1001) \right| \leq \frac{0,1 \cdot 0,9 \cdot 1,9 \cdot 2,9 \cdot 10^4 \lg e}{4 \cdot (1000)^4} \approx 0,5 \cdot 10^{-9}.$$

Остаточный член может повлиять только на девятый десятичный знак. Заметим, что полученное значение $\lg 1001$ полностью совпадает со значением в семизначной таблице логарифмов.

Метод наименьших квадратов. Сущность данного метода состоит в построении полинома $P_m(x) = \sum_{j=0}^m a_j x^j$ степени $m < n$, сумма квадратов отклонений которого от экспериментальных значений функции $y_i = f(x_i)$ минимальна. За меру качества аппроксимации функции $f(x)$ полиномом $P_m(x)$ в точках x_i принимают сумму

$$S(a_0, a_1, \dots, a_m) = \sum_{i=1}^n \omega(x_i) [f(x_i) - P_m(x_i)],$$

где $\omega(x) \geq 0$ – заранее выбранная «весовая» функция.

Для отыскания коэффициентов a_0, a_1, \dots, a_m полинома используется необходимое условие минимума функции многих переменных $\frac{\partial S}{\partial a_j} = 0$, которое при $\omega(x) = 1$ и после некоторых преобразований сводится к нормальной системе уравнений

$$\begin{aligned} a_0 \sum_{i=1}^n x_i^0 + a_1 \sum_{i=1}^n x_i^1 + \dots + a_m \sum_{i=1}^n x_i^m &= \sum_{i=1}^n y_i x_i^0, \\ \dots & \\ a_0 \sum_{i=1}^n x_i^k + a_1 \sum_{i=1}^n x_i^{k+1} + \dots + a_m \sum_{i=1}^n x_i^{k+m} &= \sum_{i=1}^n y_i x_i^k, \\ \dots & \\ a_0 \sum_{i=1}^n x_i^m + a_1 \sum_{i=1}^n x_i^{m+1} + \dots + a_m \sum_{i=1}^n x_i^{2m} &= \sum_{i=1}^n y_i x_i^m. \end{aligned} \quad (1.5)$$

При больших m система (1.5) становится «плохообусловленной», и на результат решения начинают оказывать сильное влияние ошибки округления, неточное задание исходных данных и т.п. Поэтому обычно стандартным методом наименьших квадратов пользуются для того, чтобы аппроксимировать в среднем большой массив экспериментальных данных простым полиномиальным выражением ($m \sim 2 \div 3$).

Пример 1.3. По данной таблице значений функции $y = f(x)$ (табл. 1.3) построить интерполяционную кривую и найти значение функции в точке $x=1,5$.

Таблица 1.3

i	x_i	y_i
1	-9	29
2	-8	22
3	-7	15
4	-6	9
5	-5	5
6	-4	3
7	-3	0
8	-2	-2
9	-1	-3
10	0	-2
11	1	-1
12	2	2
13	3	6
14	4	10
15	5	16
16	6	23
17	7	29
18	8	37
19	9	46
20	10	58

Решение. Построим аппроксимирующую функцию в виде квадратного полинома

$$P_2(x) = a_0 + a_1x + a_2x^2,$$

используя метод наименьших квадратов.

$$\text{Имеем: } n = 20, \quad m = 2, \quad \sum_{i=1}^n x_i^0 = 20,$$

$$\sum_{i=1}^n x_i = 10, \quad \sum_{i=1}^n x_i^2 = 670, \quad \sum_{i=1}^n x_i^3 = 1000,$$

$$\sum_{i=1}^n x_i^4 = 40666, \quad \sum_{i=1}^n y_i x_i^0 = 302,$$

$$\sum_{i=1}^n y_i x_i = 1146, \quad \sum_{i=1}^n y_i x_i^2 = 19742.$$

Тогда система уравнений для определения a_j ($j = 0, 1, 2$) примет вид:

$$\begin{cases} 20a_0 + 10a_1 + 670a_2 = 302, \\ 10a_0 + 670a_1 + 1000a_2 = 1146, \\ 670a_0 + 1000a_1 + 40666a_2 = 19742, \end{cases}$$

откуда

$$a_0 = -1,66; \quad a_1 = 0,99; \quad a_2 = 0,49.$$

Значение функции $y(1,5) \approx P_2(1,5) = 0,915$.

Построенная интерполяционная кривая (сплошная линия) и табличные значения функции (точки) приведены на рис. 1.2. Программа, реализующая данный метод, описана в прил. 1.

Задания к работе

1. Разработать схемы интерполирования функций методами Лагранжа, Ньютона, наименьших квадратов.

2. Написать, отладить и выполнить программы интерполирования функций (табл. 1.4). Интерполирование провести любым из вышеназванных методов. Построить интерполяционную кривую и найти значение функции в указанной точке (в соответствии с вариантом задания).

Таблица 1.4

x_i	Значения $y_i = f(x_i)$
-------	-------------------------

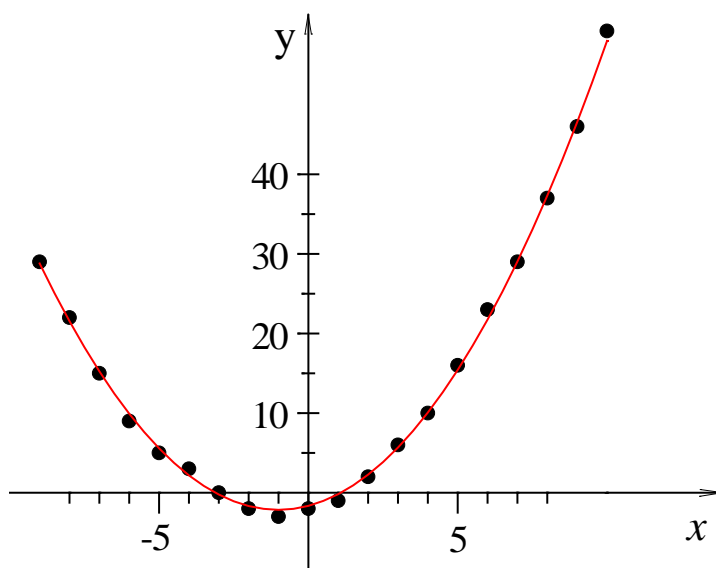


Рис. 1.2

	№1 $x = -3,5$	№2 $x = 0,5$	№3 $x = 1,25$	№4 $x = 0,75$	№5 $x = 3,12$	№6 $x = 8,25$	№7 $x = -7,4$	№8 $x = 1,8$
-9	0,1	120	53	-121	12	-140	14	-170
-8	-0,1	88	44	-90	7	-97	10	-122
-7	-1	63	35	-65	3	-67	7	-82
-6	-1	44	28	-45	0	-44	4	-51
-5	-1,5	28	23	-31	-2	-28	2	-36
-4	-1,1	16	16	-18	-3,3	-15	0	-19
-3	-0,5	8	11	-10	-4	-8	-1	-9
-2	-0,4	2	8	-4	-4,4	-5	-2	-3
-1	0	0	5	-1	-3,6	-3	-1	0
0	1	-1	4	0	-2	-2	-1	1
1	2	-2	3	0	0,4	-0	0	3
2	3	-3	4	1	3,6	2	3	4
3	4	-3	7	2	7	8	4	7
4	6	-4	8	3	12	19	8	14
5	8	-6	13	5	18	32	11	26
6	10	-11	16	9	24	54	15	45
7	12	-18	21	16	31	83	20	67
8	15	-28	28	27	39	123	27	98
9	18	-42	37	41	48	169	32	139
10	21	-61	44	60	58	228	39	191

x_i	Значения $y_i = f(x_i)$							
	№9 $x=7,5$	№10 $x=-1,3$	№11 $x=1,97$	№12 $x=9,14$	№13 $x=3,2$	№14 $x=5,43$	№15 $x=-4,2$	№16 $x=8,4$
-9	64	40	75	-114	52	159	-47	-203
-8	52	42	63	-77	44	122	-38	-147
-7	42	41	49	-54	34	88	-29	-96
-6	33	37	38	-28	26	60	-20	-53
-5	25	31	28	-14	18	40	-13	-31
-4	17	25	20	-4	14	29	-11	-10
-3	11	20	13	-1	8	17	-5	2
-2	7	14	7	0	6	8	-2	10
-1	4	8	6	-1	4	2	0	15
0	2	5	1	-2	2	1	0	14
1	1	3	0	-2	2	1	2	12
2	1	3	0	-1	2	3	3	9
3	2	7	1	3	5	4	1	7
4	7	15	4	11	8	10	-2	7
5	9	27	8	28	10	11	-7	9
6	14	47	14	46	16	12	-10	18
7	20	67	21	74	20	12	-15	26
8	26	96	33	112	28	9	-24	44
9	37	131	39	159	35	4	-30	70
10	47	175	51	218	42	-4	-37	102

Контрольные вопросы

1. Сформулируйте постановку задачи интерполирования функций, назовите области ее применения.
2. Какую функцию называют аппроксимирующей?
3. Охарактеризуйте интерполяционную формулу Лагранжа.
4. Приведите интерполяционные формулы Ньютона.
5. Опишите схему Эйткена.
6. Приведите оценку остаточного члена для каждой из формул.
7. Опишите метод наименьших квадратов.

Лабораторная работа 2. Приближенное вычисление интегралов

Цель работы: изучение различных методов вычисления определенных интегралов, практическое интегрирование функций на ЭВМ.

Постановка задачи

Заменяя подынтегральную функцию каким-либо интерполяционным многочленом, получаем квадратурные формулы вида

$$\int_a^b f(x) dx = \sum_{i=0}^n c_i f(x_i) + \Psi,$$

где x_i – выбранные узлы интерполяции, c_i – коэффициенты, зависящие только от выбора узлов, но не от вида функции ($i = 0, 1, \dots, n$), Ψ – остаточный член, или погрешность квадратурной формулы. Погрешность зависит как от расположения узлов, так и от выбора коэффициентов. Отбрасывая остаточный член Ψ , мы совершаем погрешность усечения. При расчете к ней еще добавляются погрешности округления.

Разобьем отрезок интегрирования $[a, b]$ на n равных частей системой точек

$$\omega_h = \{ x_i = x_0 + h, \quad i = 0, 1, \dots, n, \quad hN = b - a \},$$

и вычислим подынтегральную функцию в полученных узлах $y_i = f(x_i)$.

Квадратурные формулы для равноотстоящих узлов называются формулами Ньютона-Котеса. Чтобы не иметь дело с многочленами высоких степеней, обычно разбивают промежуток интегрирования на отдельные участки, применяют формулы Ньютона-Котеса с невысокими степенями на каждом участке и потом складывают полученные результаты (так называемые составные формулы). Наиболее простые из формул такого типа приведены ниже.

1. *Формула прямоугольников:*

$$\int_a^b f(x) dx \approx \sum_{i=1}^n f(x_{i-1/2}) h,$$

где $x_{i-1/2} = x_i - 0,5h$.

Погрешность этой формулы

$$|\Psi| \leq \frac{h^2(b-a)}{24} M_2, \quad M_2 = \max_{\xi \in [a,b]} |f''(\xi)|.$$

Формула прямоугольников дает точное значение интеграла, когда подынтегральная функция $f(x)$ линейна, ибо тогда $f''(x) \equiv 0$.

2. *Формула трапеций:*

$$\int_a^b f(x) dx \approx h \left(\frac{y_0 + y_n}{2} + y_1 + y_2 + \dots + y_{n-1} \right),$$

где $y_i = f(x_i)$ ($i = 0, 1, \dots, n$).

Остаточный член имеет вид

$$|\Psi| \leq \frac{h^2(b-a)}{12} M_2, \quad M_2 = \max_{\xi \in [a,b]} |f''(\xi)|.$$

Формула трапеций имеет, так же как и формула прямоугольников, второй порядок точности, $\Psi = O(h^2)$.

3. *Формула Симпсона:*

$$\begin{aligned} \int_a^b f(x) dx \approx & \frac{h}{3} (y_0 + y_{2m} + \\ & + 2[y_2 + y_4 + \dots + y_{2m-2}] + 4[y_1 + y_3 + \dots + y_{2m-1}]), \end{aligned}$$

где $h = \frac{b-a}{n} = \frac{b-a}{2m}$.

Остаточный член имеет вид

$$|\Psi| \leq \frac{h^4(b-a)}{180} M_4, \quad M_4 = \max_{\xi \in [a,b]} |f^{(4)}(\xi)|.$$

Формула Симпсона является точной для многочленов до третьей степени включительно, так как в этом случае $f^{(4)}(x) \equiv 0$.

Заметим, что в формуле Симпсона число узлов обязательно нечетное, т.е. n четное, $n=2m$.

Пример 2.1. Вычислить интеграл

$$\int_0^1 e^{-x^2} dx$$

по формуле трапеций при $n=10$ и оценить погрешность вычислений.

Решение. Оценим вначале остаточный член. Для этого находим вторую производную функции $y = e^{-x^2}$:

$$y'' = 2(2x^2 - 1)e^{-x^2}.$$

На отрезке $[0, 1]$ абсолютная величина второй производной $|y''(x)|$ имеет наибольшее значение при $x=0$. Отсюда

$$|\Psi| \leq \frac{h^2(b-a)}{12} \max |y''(x)| = \frac{2 \cdot (0,1)^2}{12} < 0,002.$$

Чтобы погрешности округления не повлияли на точность результата, будем вести вычисления с одним запасным знаком, т.е. с четырьмя знаками после запятой.

По формуле трапеций получим

$$\int_0^1 e^{-x^2} dx \approx 0,1 \cdot 7,4620 = 0,7462.$$

Окончательный ответ округляем до трех знаков:

$$\int_0^1 e^{-x^2} dx \approx 0,746.$$

Выбор шага интегрирования. Пусть нам необходимо выбрать такой шаг h , который обеспечивает заданную точность ε вычисления интеграла по выбранной формуле численного интегрирования.

Рассмотрим два способа решения этой задачи.

1. *Выбор шага по оценке остаточного члена.* Пусть требуется вычислить интеграл с точностью ε . Используя формулу соответствующего остаточного члена Ψ , выбирают h таким, чтобы выполнялось неравенство $|\Psi| < \varepsilon/2$. Затем вычисляют интеграл по приближенной формуле с полученным шагом. При этом вычисления следует производить с таким числом знаков, чтобы погрешность округления не превышала $\varepsilon/2$.

Замечание. Бывают ситуации, когда допустимую погрешность ε делят между погрешностью усечения и погрешностью округления не поровну. Например, если вычисления значений подынтегральной функции очень трудоемки, но могут быть произведены с любой точностью, то может оказаться целесообразным выбирать шаг h из

условия $|\Psi| < \varepsilon$. Другой крайний случай может представиться для функций, задаваемых экспериментально, когда трудно обеспечить большую точность вычисления значений функции.

Пример 2.2. С помощью формулы Симпсона вычислить интеграл

$$\int_{\pi/4}^{\pi/2} \frac{\sin x}{x} dx$$

с точностью до $\varepsilon = 10^{-3}$.

Решение. Выберем шаг h таким, чтобы выполнялось неравенство

$$\frac{h^4(b-a)}{180} \max_{x \in [a,b]} |f^{(4)}(x)| < 0,5 \cdot 10^{-3}.$$

Вычисляем $f^{(4)}(x)$:

$$f^{(4)}(x) = \frac{\sin x}{x} + 4 \frac{\cos x}{x^2} - 12 \frac{\sin x}{x^3} - 24 \frac{\cos x}{x^4} + 24 \frac{\sin x}{x^5}.$$

При оценке $|f^{(4)}(x)|$ на отрезке $[\pi/4, \pi/2]$ воспользуемся тем, что величины $\frac{\sin x}{x} \left(1 - \frac{12}{x^2} + \frac{24}{x^4}\right)$ и $4 \frac{\cos x}{x^2} \left(\frac{6}{x^2} - 1\right)$ на этом отрезке положительны и убывают. Поэтому они достигают наибольшего значения в точке $x = \pi/4$, причем

$$|f^{(4)}(x)| \leq \frac{\sin x}{x} \left(1 - \frac{12}{x^2} + \frac{24}{x^4}\right) + 4 \frac{\cos x}{x^2} \left(\frac{6}{x^2} - 1\right) < 81.$$

Тогда для определения шага расчета h мы получаем неравенство

$$\frac{h^4 \cdot \pi/4}{180} \cdot 81 < 0,5 \cdot 10^{-3},$$

откуда $h < 0,19$.

С другой стороны, шаг расчета h следует выбирать так, чтобы разделить отрезок $[\pi/4, \pi/2]$ на четное число равных частей. Указанным двум условиям отвечает значение $h = \pi/24 = 0,13 < 0,19$, при котором $n=6$. Далее, для того чтобы погрешность вычислений не превышала $0,5 \cdot 10^{-3}$, достаточно вести вычисления с четырьмя знаками после запятой.

По формуле Симпсона при $n=6$ находим

$$\int_{\pi/4}^{\pi/2} \frac{\sin x}{x} dx \approx \frac{h}{3} \{f(x_0) + f(x_6) + 2[f(x_2) + f(x_4)] + 4[f(x_1) + f(x_3) + f(x_5)]\} = 0,6118.$$

Окончательный результат округляем до трех знаков:

$$\int_{\pi/4}^{\pi/2} \frac{\sin x}{x} dx = 0,612.$$

2. *Двойной пересчет.* Поскольку отыскание $\max |f^{(k)}(x)|$ нередко приводит к слишком громоздким вычислениям, на практике обычно используют следующий прием.

Вычисляют интеграл I по выбранной квадратурной формуле дважды: сначала с некоторым шагом h , затем с шагом $h/2$, т.е. удваивают число n .

Обозначив результаты вычислений через I_h и $I_{h/2}$ соответственно, сравнивают их. Если $|I_h - I_{h/2}| < \varepsilon$, где ε - допустимая погрешность, то полагают $I \approx I_{h/2}$.

Если же окажется, что $|I_h - I_{h/2}| \geq \varepsilon$, то расчет повторяют с шагом $h/4$. В качестве начального шага иногда можно рекомендовать число, близкое к $\sqrt[m]{\varepsilon}$, где $m=2$ для формулы трапеций и $m=4$ для формулы Симпсона.

Указанный прием широко используется при вычислении интегралов на ЭВМ, так как он позволяет осуществить автоматический выбор шага при заданной точности с одновременным контролем вычислений.

Отметим, что для приближенной оценки погрешности усечения Δ можно пользоваться принципом Рунге, согласно которому

$$\Delta \approx \frac{1}{3} |I_h - I_{h/2}| \text{ для формулы трапеций,}$$

$$\Delta \approx \frac{1}{15} |I_h - I_{h/2}| \text{ для формулы Симпсона.}$$

Пример 2.3. Вычислить по формуле прямоугольников значение интеграла

$$I = \int_a^b \frac{dx}{x + \cos x}$$

с точностью $\varepsilon = 10^{-4}$ для $a=0$, $b=\pi$.

Решение. Схему вычислений можно записать следующим образом. Выбираем начальный шаг h разбиения отрезка $[a, b]$, например, $h = \sqrt{\varepsilon} = 0,01$. Находим $n = \left\lceil \frac{b-a}{h} \right\rceil = 314$. Определяем приближенное значение интеграла по формуле прямоугольников:

$$I_h \approx \sum_{i=1}^n f(x_{i-1/2}) h.$$

Далее вычисляем приближенное значение интеграла $I_{h/2}$ с шагом $h/2$. Определяем погрешность вычислений $\Delta = |I_h - I_{h/2}|$. Если $\Delta \geq \varepsilon$, шаг интегрирования уменьшаем вдвое, т.е. приближенные значения интеграла определяем для последовательных шагов h , $h/2$, $h/4$, ... Вычисления заканчиваем при выполнении условия $\Delta < \varepsilon$. Окончательный результат равен $I = 2,0431$.

Программа, реализующая данную схему, приведена в прил. 2.

Задания к работе

1. Разработать схемы интегрирования по формулам прямоугольников, трапеций и Симпсона.
2. Написать, отладить и выполнить программы интегрирования функций, приведенных в табл. 2.1. Вычисления значения интеграла на отрезке $[a, b]$ провести с заданной точностью (в соответствии с вариантом задания). Величину шага, обеспечивающего требуемую точность, определить с помощью двойного пересчета.
3. Определить относительную погрешность вычислений по формуле:

$$\delta = \left| \frac{I - I_h}{I} \right| \cdot 100 \% ,$$

где I – точное значение интеграла, вычисленное через первообразную функции; I_h – значение интеграла, полученное в результате применения конкретной формулы интегрирования.

Таблица 2.1

№	Подынтегральная функция $f(x)$	Формулы численного интегрирования	Заданная точность	Интервал $[a, b]$	Первообразная функции $F(x)$
I	II	III	IV	V	VI
1	$\frac{x}{(x+3)^2}$	Трапедий	10^{-3}	$[0; 2]$	$\frac{3}{x+3} + \ln(x+3)$
2	$x \sin 2x$	Симпсона	10^{-4}	$\left[0; \frac{\pi}{4}\right]$	$\frac{\sin 2x}{4} - \frac{x \cos 2x}{2}$
3	2^{3x}	Симпсона	10^{-5}	$[0; 1]$	$\frac{2^{3x}}{3 \ln 2}$
4	$\frac{\ln^2 x}{x}$	Симпсона	10^{-4}	$[1; 5]$	$\frac{\ln^3 x}{3}$
5	$e^{2x} \sin x$	Трапедий	10^{-4}	$\left[0; \frac{\pi}{2}\right]$	$\frac{1}{5} e^{2x} (2 \sin x - \cos x)$
6	$\frac{x}{\sin^2 3x}$	Симпсона	10^{-5}	$[0,2; 1]$	$\frac{\ln(\sin 3x)}{9} - \frac{x}{3} \operatorname{ctg} 3x$
7	$x^2 \sin x$	Симпсона	10^{-4}	$[0; 1]$	$2x \sin x - (x^2 - 2) \cos x$
8	$\frac{1}{x\sqrt{x^2 + 1/4}}$	Трапедий	10^{-3}	$[1; 2]$	$-2 \ln \left(\frac{0,5 + \sqrt{x^2 + 1/4}}{x} \right)$
9	$x e^{0,8x}$	Симпсона	10^{-4}	$[2; 3]$	$\frac{e^{0,8x}}{0,64} (0,8x - 1)$
10	$\operatorname{arctg} x$	Симпсона	10^{-5}	$[0; 1]$	$x \cdot \operatorname{arctg} x - \frac{\ln(1+x^2)}{2}$
11	$x \ln x$	Симпсона	10^{-4}	$[2; 6]$	$\frac{x^2}{2} \ln x - \frac{x^2}{4}$
12	$\frac{1}{1 + \sin x}$	Трапедий	10^{-3}	$\left[0; \frac{\pi}{2}\right]$	$-\frac{2}{1 + \operatorname{tg}(x/2)}$
13	$\frac{e^x - 1}{e^x + 1}$	Симпсона	10^{-4}	$[0; 3]$	$2 \ln(e^x + 1) - x$

I	II	III	IV	V	VI
14	$\sin^3 x \cos x$	Симпсона	10^{-5}	$\left[0; \frac{\pi}{2}\right]$	$\frac{1}{4} \sin^4 x$
15	$\frac{x}{(x^2 + 1)^2}$	Трапеций	10^{-4}	[1; 7]	$-\frac{1}{2(x^2 + 1)}$
16	$x^x (\ln x + 1)$	Трапеций	10^{-4}	[1; 3]	x^x

Контрольные вопросы

1. В чем состоит суть методов численного интегрирования функций?
2. Опишите формулу прямоугольников.
3. Охарактеризуйте метод трапеций.
4. Опишите формулу Симпсона.
5. Приведите оценку погрешности для каждого из методов на частичном отрезке и на всем интервале интегрирования.
6. Проведите сравнение формул численного интегрирования по точности на основании остаточных членов формул.
7. Как оценивается погрешности приближенного вычисления интегралов по правилу Рунге?

Лабораторная работа 3. Решение систем линейных алгебраических уравнений

Цель работы: изучение численных методов решения систем линейных алгебраических уравнений, практическое решение систем на ЭВМ.

Постановка задачи

Пусть дана система n линейных алгебраических уравнений с n неизвестными

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2, \\ \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n. \end{cases} \quad (3.1)$$

или в матричной форме

$$A \bar{x} = \bar{b}, \quad (3.2)$$

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}, \quad \bar{b} = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{pmatrix}, \quad \bar{x} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix},$$

где A – матрица коэффициентов; \bar{b} и \bar{x} – столбец свободных членов и столбец неизвестных соответственно. Если матрица A неособенная, т.е. определитель этой матрицы не равен нулю, то система (3.1) имеет единственное решение.

Применяемые в настоящее время численные методы решения систем линейных алгебраических уравнений можно разбить на две группы: точные (прямые) и приближенные (итерационные).

Точными методами называются такие методы, которые в предположении, что вычисления ведутся точно (без округлений), приводят к точным значениям неизвестных x_i . Поскольку на практике все вычисления ведутся с округлениями, то и значения неизвестных, полученные точным методом, неизбежно будут содержать погрешности.

Наиболее распространенным точным методом решения систем линейных алгебраических уравнений является метод Гаусса, в основе

1) Прямой ход, когда система (3.1) приводится к треугольному виду

где коэффициенты уравнений определяются следующим образом:

Вычисление правых частей уравнений системы (3.3) осуществляется по формулам:

$$b_i^{(k)} = b_i^{(k-1)} - a_{ik}^{(k-1)} d_k, \quad i = k+1, k+2, \dots, n, \quad k = 1, 2, \dots, n.$$

2) Коэффициенты c_{ij} и правые части d_i ($i=1, 2, \dots, n$, $j=i+1, i+2, \dots, n$) хранятся в памяти ЭВМ и используются при осуществлении обратного хода, который заключается в нахождении неизвестных x_i из системы (3.3). Общие формулы обратного хода имеют вид

$$x_n = d_n, \quad x_i = d_i - \sum_{j=i+1}^n c_{ij} x_j, \quad (i = n-1, n-2, \dots, 1). \quad (3.6)$$

Пример 3.1. Методом Гаусса решить систему

$$\begin{cases} 2x_1 + x_2 - 0,1x_3 + x_4 = 2,7, \\ 0,4x_1 + 0,5x_2 + 4x_3 - 8,5x_4 = 21,9, \\ 0,3x_1 - x_2 + x_3 + 5,2x_4 = -3,9, \\ x_1 + 0,2x_2 + 2,5x_3 - x_4 = 9,9. \end{cases}$$

Решение. Прямой ход. Приведем заданную систему к треугольному виду, используя формулы (3.4) и (3.5). Тогда система запишется следующим образом

$$\begin{cases} x_1 + 0,5x_2 - 0,05x_3 + 0,5x_4 = 1,35, \\ x_2 + 13,4x_3 - 29x_4 = 71,2, \\ x_3 - 1,72298x_4 = 4,72298, \\ x_4 = -1. \end{cases}$$

Обратный ход. Последовательно из полученной системы линейных алгебраических уравнений по формуле (3.6) находим неизвестные

$$x_4 = -1,$$

$$x_3 = 4,72298 - 1,72298 = 3,$$

$$x_2 = 71,2 - 13,4 \cdot 3 - 29 = 2,$$

$$x_1 = 1,35 - 0,5 \cdot 2 + 0,05 \cdot 3 + 0,5 = 1.$$

Программа, реализующая метод Гаусса, приведена в прил. 3.

Основным ограничением метода Гаусса является предположение о том, что элемент $a_{kk}^{(k-1)}$, называемый ведущим элементом на k -м шаге исключения и на который проводится деление, отличен от нуля. Даже если какой-то ведущий элемент не равен нулю, а просто близок к нему, то в процессе вычислений может происходить сильное накопление погрешностей. Избежать указанных трудностей позволяет метод Гаусса с выбором главного элемента. Основная идея метода состоит в том, чтобы на очередном шаге исключать не следующее по номеру неизвестное, а то неизвестное, коэффициент при котором является наибольшим по модулю. Таким образом, в качестве ведущего элемента здесь выбирается главный, т.е.

наибольший по модулю элемент. Тем самым, если $\det A \neq 0$, то в процессе вычислений не будет происходить деление на нуль.

Приближенными методами называются такие методы, которые даже в предположении, что вычисления ведутся без округлений, позволяют получить решение системы (3.1) лишь с заданной точностью. Точное решение системы в этих случаях может быть получено теоретически как результат бесконечного процесса. К приближенным методам относятся метод простой итерации, метод Зейделя и др. Каждый из этих методов не всегда является сходящимся в применении к конкретному классу систем линейных алгебраических уравнений.

Метод простой итерации. Пусть система линейных уравнений $A \bar{x} = \bar{b}$ каким-либо образом приведена к виду

$$\bar{x} = C \bar{x} + \bar{d}, \quad (3.7)$$

где C – некоторая матрица, а \bar{d} – вектор-столбец.

Исходя из произвольного вектора $\bar{x}^{(0)} = \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \\ \dots \\ x_n^{(0)} \end{pmatrix},$

строим итерационный процесс

$$\bar{x}^{(k+1)} = C \bar{x}^{(k)} + \bar{d}, \quad (k = 0, 1, 2, \dots),$$

или в развернутой форме

$$\begin{cases} x_1^{(k+1)} = c_{11}x_1^{(k)} + c_{12}x_2^{(k)} + \dots + c_{1n}x_n^{(k)} + d_1, \\ \dots \\ x_n^{(k+1)} = c_{n1}x_1^{(k)} + c_{n2}x_2^{(k)} + \dots + c_{nn}x_n^{(k)} + d_n. \end{cases}$$

Производя итерации, получим последовательность векторов $\bar{x}^{(1)}, \bar{x}^{(2)}, \dots, \bar{x}^{(k)}, \dots$

Доказано, что если элементы матрицы C удовлетворяют одному из условий

$$\sum_{j=1}^n |c_{ij}| \leq \alpha < 1, \quad (i = 1, 2, \dots, n) \quad (3.8)$$

или

$$\sum_{i=1}^n |c_{ij}| \leq \beta < 1, \quad (j=1,2,\dots,n), \quad (3.9)$$

то процесс итерации сходится к точному решению системы \bar{x} при любом начальном векторе $\bar{x}^{(0)}$, т. е.

$$\bar{x} = \lim_{k \rightarrow \infty} \bar{x}^{(k)}.$$

Таким образом, точное решение системы получается лишь в результате бесконечного процесса и всякий вектор $\bar{x}^{(k)}$ из полученной последовательности является приближенным решением. Оценка погрешности этого приближенного решения $\bar{x}^{(k)}$ дается одной из следующих формул:

$$|x_i - x_i^{(k)}| \leq \frac{\alpha}{1 - \alpha} \max_{j=1,2,\dots,n} |x_j^{(k)} - x_j^{(k-1)}|,$$

если выполнено условие (3.8), или

$$|x_i - x_i^{(k)}| \leq \frac{\beta}{1 - \beta} \sum_{j=1}^n |x_j^{(k)} - x_j^{(k-1)}|,$$

если выполнено условие (3.9).

Начальный вектор $\bar{x}^{(0)}$ может быть выбран, вообще говоря, произвольно. Иногда берут $\bar{x}^{(0)} = \bar{d}$. Однако наиболее целесообразно в качестве компонент вектора $\bar{x}^{(0)}$ взять приближенные значения неизвестных, полученные грубой прикидкой.

Приведение системы (3.2) к виду (3.7) можно осуществить различными способами. Важно только, чтобы выполнялось одно из условий (3.8) или (3.9).

Пример 3.2. Решить систему

$$\begin{cases} 1,02x_1 - 0,05x_2 - 0,1x_3 = 0,795, \\ -0,11x_1 + 1,03x_2 - 0,05x_3 = 0,849, \\ -0,11x_1 - 0,12x_2 + 1,04x_3 = 1,398, \end{cases}$$

произведя три итерации. Указать погрешность полученного результата.

Решение. Матрица данной системы такова, что диагональные элементы близки к единице, а все остальные – значительно меньше единицы. Поэтому для применения метода простой итерации запишем данную систему (9.11) в виде

$$\begin{cases} x_1 = 0,795 - 0,02x_1 + 0,05x_2 + 0,1x_3, \\ x_2 = 0,849 + 0,11x_1 - 0,03x_2 + 0,05x_3, \\ x_3 = 1,398 + 0,11x_1 + 0,12x_2 - 0,04x_3. \end{cases}$$

Условия сходимости (3.8) для полученной системы выполнены. Действительно

$$\sum_{j=1}^3 |c_{1j}| = 0,02 + 0,05 + 0,1 = 0,17 < 1,$$

$$\sum_{j=1}^3 |c_{2j}| = 0,11 + 0,03 + 0,05 = 0,19 < 1,$$

$$\sum_{j=1}^3 |c_{3j}| = 0,11 + 0,12 + 0,04 = 0,27 < 1,$$

Берем в качестве начального вектора $\bar{x}^{(0)}$ столбец свободных членов, округлив его элементы до двух знаков после запятой:

$$\bar{x}^{(0)} = \begin{pmatrix} 0,80 \\ 0,85 \\ 1,40 \end{pmatrix}.$$

Далее последовательно находим
при $k=1$

$$x_1^{(1)} = 0,962, \quad x_2^{(1)} = 0,982, \quad x_3^{(1)} = 1,532;$$

при $k=2$

$$x_1^{(2)} = 0,978, \quad x_2^{(2)} = 1,002, \quad x_3^{(2)} = 1,560;$$

при $k=3$

$$x_1^{(3)} = 0,980, \quad x_2^{(3)} = 1,004, \quad x_3^{(3)} = 1,563.$$

Значения неизвестных при $k=2$ и $k=3$ отличаются не более чем на $\varepsilon = 3 \cdot 10^{-3}$; поэтому, взяв в качестве неизвестных

$$x_1 \approx 0,980, \quad x_2 \approx 1,004, \quad x_3 \approx 1,563,$$

погрешность этих приближенных значений не превзойдет

$$\frac{\alpha}{1-\alpha} \varepsilon = \frac{0,27}{1-0,27} \cdot 3 \cdot 10^{-3} < 1,1 \cdot 10^{-3}.$$

Метод Зейделя является модификацией метода простой итерации. Он заключается в том, что при вычислении $(k+1)$ -го приближения неизвестного x_i при $i > 1$ используются уже вычисленные ранее $(k+1)$ -е приближения неизвестных x_1, x_2, \dots, x_{i-1} . Таким образом, для системы (3.7) вычисления по методу Зейделя ведутся по формулам

$$\left\{ \begin{array}{l} x_1^{(k+1)} = c_{11}x_1^{(k)} + c_{12}x_2^{(k)} + \dots + c_{1n}x_n^{(k)} + d_1 , \\ x_2^{(k+1)} = c_{21}x_1^{(k+1)} + c_{22}x_2^{(k)} + \dots + c_{2n}x_n^{(k)} + d_2 , \\ . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \\ x_n^{(k+1)} = c_{n1}x_1^{(k+1)} + c_{n2}x_2^{(k+1)} + \dots + c_{nn}x_n^{(k)} + d_n . \end{array} \right.$$

Указанные условия сходимости для метода простой итерации остаются верными и для метода Зейделя. Обычно метод Зейделя дает лучшую сходимость, чем метод простой итерации, хотя это бывает не всегда. Кроме того, метод Зейделя может оказаться более удобным при программировании, так как при вычислении $x_i^{(k+1)}$ нет необходимости хранить значения $x_1^{(k)}, x_2^{(k)}, \dots, x_{i-1}^{(k)}$.

Пример 3.3. Методом Зейделя решить систему

$$\begin{cases} 20,9x_1 + 1,2x_2 + 2,1x_3 + 0,9x_4 = 21,70, \\ 1,2x_1 + 21,2x_2 + 1,5x_3 + 2,5x_4 = 27,46, \\ 2,1x_1 + 1,5x_2 + 19,8x_3 + 1,3x_4 = 28,76, \\ 0,9x_1 + 2,5x_2 + 1,3x_3 + 32,1x_4 = 49,72. \end{cases}$$

Решение. Приведем исходную систему к виду (3.7):

$$\left\{ \begin{array}{l} x_1 = \frac{1}{20,9}(21,70 - 1,2x_2 - 2,1x_3 - 0,9x_4), \\ x_2 = \frac{1}{21,2}(7,46 - 1,2x_1 - 1,5x_3 - 2,5x_4), \\ x_3 = \frac{1}{19,8}(28,76 - 2,1x_1 - 1,5x_2 - 1,3x_4), \\ x_4 = \frac{1}{32,1}(49,72 - 0,9x_1 - 2,5x_2 - 1,3x_3). \end{array} \right.$$

Коэффициенты полученной системы удовлетворяют условию (3.8)

$$\sum_{j=1}^4 |c_{1j}| \approx 0,20 < 1, \quad \sum_{j=1}^4 |c_{2j}| \approx 0,24 < 1,$$

$$\sum_{j=1}^4 |c_{3j}| \approx 0,25 < 1, \quad \sum_{j=1}^4 |c_{4j}| \approx 0,15 < 1.$$

Поэтому сходимость итераций гарантирована. При этом $\alpha=0,25$.

В качестве начального вектора $\bar{x}^{(0)}$ возьмем элементы столбца свободных членов, округлив их значения до двух знаков после запятой:

$$\bar{x}^{(0)} = \begin{pmatrix} 1,04 \\ 1,30 \\ 1,44 \\ 1,55 \end{pmatrix}.$$

Вычисления будем вести до тех пор, пока величины $|x_j^{(k)} - x_j^{(k-1)}|$ ($j=1,2,3,4$) не станут меньше, например, $\varepsilon = 10^{-3}$.

При $k=1$ $x_1^{(1)} = 0,7512$. При вычислении $x_2^{(1)}$ используем уже полученное значение $x_1^{(1)}$: $x_2^{(1)} = 0,9674$. При вычислении $x_3^{(1)}$ используем значения $x_1^{(1)}$ и $x_2^{(1)}$: $x_3^{(1)} = 1,1977$. Наконец, используя значения $x_1^{(1)}$, $x_2^{(1)}$ и $x_3^{(1)}$, получаем $x_4^{(1)} = 1,4037$.

Аналогичным образом ведем вычисления при последующих k . Получаем

при $k=2$

$$x_1^{(2)} = 0,8019, \quad x_2^{(2)} = 0,9996, \quad x_3^{(2)} = 1,1996, \quad x_4^{(2)} = 1,4000;$$

при $k=3$

$$x_1^{(3)} = 0,8001, \quad x_2^{(3)} = 1,0000, \quad x_3^{(3)} = 1,1999, \quad x_4^{(3)} = 1,4000;$$

при $k=4$

$$x_1^{(4)} = 0,8000, \quad x_2^{(4)} = 1,0000, \quad x_3^{(4)} = 1,1999, \quad x_4^{(4)} = 1,4000.$$

Модули разностей значений $x_i^{(k)}$ при $k=3$ и $k=4$ не превышают заданного числа ε , поэтому в качестве решения системы возьмем

$$x_1 \approx 0,8000, \quad x_2 \approx 1,0000, \quad x_3 \approx 1,1999, \quad x_4 \approx 1,4000.$$

При этом погрешность этих приближенных значений не превзойдет

$$\frac{\alpha}{1-\alpha} \max_{j=1,2,3,4} |x_j^{(4)} - x_j^{(3)}| = \frac{0,25}{1-0,25} \cdot 10^{-4} < 3,3 \cdot 10^{-5}.$$

Задания к работе

1. Составить схемы алгоритмов решения систем линейных алгебраических уравнений методами Гаусса, простой итерации и Зейделя.

2. Написать, отладить и выполнить программы решения систем линейных алгебраических уравнений, записанных в векторно-матричной форме $A\bar{x} = \bar{b}$ и приведенных в табл. 3.1. Четным вариантам решить систему методом Гаусса с выбором главного элемента. Остальным – методом Зейделя.

3. Вычислить точностные оценки методов по координатам:

$$\delta = \max |x_i - x_i^*|, \quad i = 1, \dots, N,$$

где x_i^* – координаты точного решения; x_i – координаты численного решения.

Таблица 3.1

№	Матрица коэффициентов системы A				Столбец свободных членов \bar{b}	Точное решение x^*
I	II				III	IV
1	4,52	−9,11	2,24	1,72	−6,77	0,5
	3,56	−6,75	14,28	2,07	−22,25	1,0
	−9,34	4,13	0,98	3,00	3,99	−1,5
	1,64	2,32	−1,80	7,12	20,08	2,0
2	0,31	0,14	0,30	0,27	1,02	1
	0,26	0,32	0,18	0,24	1,00	1
	0,61	0,22	0,20	0,31	1,34	1
	0,40	0,34	0,36	0,17	1,27	1
3	1,32	2,06	−3,40	7,11	30,17	1
	−9,13	5,84	1,21	0,76	3,62	2
	3,12	−8,14	2,51	−1,13	−19,06	−1
	0,77	0,17	2,32	1,10	2,09	3

Продолжение табл. 3.1

I	II				III	IV
4	2,01	1,00	– 0,24	1,31	1,98	1
	0,45	2,36	0,58	3,22	3,69	2
	0,30	– 1,08	1,00	– 2,34	3,48	3
	1,12	0,24	2,55	– 1,11	10,36	– 1
5	1,14	– 5,03	3,01	0,12	– 10,91	2
	4,77	1,03	0,58	– 1,17	12,19	2
	2,11	1,17	4,89	0,88	0,79	– 1
	0,14	– 0,18	1,28	2,10	– 3,46	– 1
6	0,74	– 0,62	2,11	0,55	3,18	2
	0,50	0,98	1,79	0,09	0,56	– 2
	– 0,73	0,25	2,07	1,00	– 2,89	1
	1,00	– 0,85	1,95	0,15	5,20	– 3
7	3,1	1,5	1,1	– 0,5	9,70	1,3
	– 2,0	5,1	0,4	1,6	4,71	1,7
	0,8	2,2	0,9	4,7	0,18	2,2
	– 0,1	3,2	6,2	1,8	16,43	– 1,4
8	2,12	0,42	1,34	0,88	11,172	3,7
	0,42	3,95	1,87	0,43	0,115	– 1,5
	1,34	1,87	2,98	0,46	9,009	2,1
	0,88	0,43	0,46	4,44	9,349	1,3
9	1,85	0,70	– 0,12	– 0,18	8,41	3
	0,16	0,19	0,79	0,11	– 0,23	4
	1,13	2,77	0,18	– 0,20	13,91	– 2
	1,14	1,01	0,55	3,22	9,58	1
10	6,1	– 2,2	– 1,2	– 3,3	– 0,50	1,5
	7,2	0,9	1,8	– 4,1	– 2,15	2,0
	2,8	3,3	1,1	2,5	14,30	– 2,5
	– 1,5	1,0	6,3	0,8	– 14,00	2,5
11	3,82	1,02	0,75	0,81	16,855	2,5
	1,05	4,53	0,98	1,53	22,705	3,0
	0,73	0,85	4,71	0,81	22,480	3,5
	0,88	0,81	1,28	3,50	16,110	2,0
12	0,15	2,11	3,75	8,14	16,60	– 1
	0,64	1,21	2,05	– 0,99	– 2,25	2
	3,21	1,53	– 1,04	– 3,18	– 5,47	– 1
	0,77	1,22	1,18	2,25	4,99	2
13	1,15	0,42	10,10	4,25	15,08	1
	1,59	0,55	– 0,32	0,29	1,01	– 1
	1,14	3,15	2,05	7,86	7,90	1
	0,77	6,11	– 3,01	0,74	– 7,61	1

Окончание табл. 3.1

I	II				III	IV
14	1,02	– 0,25	– 0,30	– 4,11	7,42	0
	0,41	1,13	– 0,15	0,77	0,57	2
	0,85	2,17	1,21	6,27	– 6,99	1
	3,44	– 4,33	0,50	1,42	– 11,00	– 2
15	5,9	1,2	2,1	0,9	2,0	– 1
	1,2	7,2	1,5	2,5	5,3	0
	2,1	1,5	9,8	1,3	10,3	1
	0,9	2,5	1,3	6,1	12,6	2
16	0,28	10,36	– 0,60	2,70	23,56	1
	– 2,70	– 1,36	– 0,72	3,85	– 11,60	3
	2,92	3,06	1,11	2,57	11,40	4
	0,97	2,94	2,61	– 13,40	47,03	2

Контрольные вопросы

1. Когда система линейных алгебраических уравнений имеет единственное решение?
2. Каковы недостатки решения системы уравнений по правилу Крамера?
3. Охарактеризуйте точные и приближенные численные методы решения систем линейных алгебраических уравнений.
4. Опишите метод Гаусса с выбором главного элемента.
5. Почему метод простой итерации называется самоисправляющимся?
6. Дайте определение сходимости итерационного процесса.
7. Опишите метод Зейделя.

Лабораторная работа 4. Решение нелинейных уравнений

Цель работы: изучение методов решения нелинейных алгебраических и трансцендентных уравнений, практическое решение уравнений на ЭВМ.

Постановка задачи

Проблема решения нелинейного уравнения

$$f(x) = 0, \quad (4.1)$$

где $f(x)$ – заданная функция, возникает в разнообразных практических приложениях.

Задача нахождения корней уравнения (4.1) обычно решается в два этапа. На первом этапе изучается расположение корней и проводится их разделение, т.е. находится какой-либо интервал $[a, b]$ оси x , внутри которого находится один корень, и нет других решений уравнения (4.1). Для выделения интервала, содержащего единственный корень, обычно пытаются использовать всю имеющуюся информацию о функции $f(x)$ и построить эскиз ее графика, позволяющий судить с достоверностью о наличии корней, а также об их примерном расположении на оси x . Кроме того, изучается вопрос о кратности корней. Тем самым находят некоторые начальные приближения для корней уравнения (4.1).

На втором этапе, используя заданное начальное приближение, строится итерационный процесс, позволяющий уточнить значение отыскиваемого корня. Для численного анализа типично использование универсальных или почти универсальных алгоритмов итерационного типа, которые основаны на многократном циклическом повторении определенной совокупности действий, причем повторение цикла должно обеспечивать если не достижение цели, то, по крайней мере, приближение к ней. Итерационные алгоритмы не дают точного значения корня уравнения, но позволяют приблизиться к нему сколь угодно близко.

Метод деления отрезка пополам (метод вилки). Для его применения требуется, чтобы на концах интервала $[a, b]$ функция $f(x)$ принимала ненулевые значения противоположного знака, т.е. чтобы $f(a) \cdot f(b) < 0$. Итерационная процедура состоит в переходе от такого интервала к новому интервалу, совпадающему с одной из половин предыдущего и обладающему тем же свойством.

Пусть для определенности $f(a) > 0$ и $f(b) < 0$. Положим $x_0 = (a + b)/2$ и вычислим $f(x_0)$ (рис. 4.1). Если $f(x_0) = 0$, то x_0

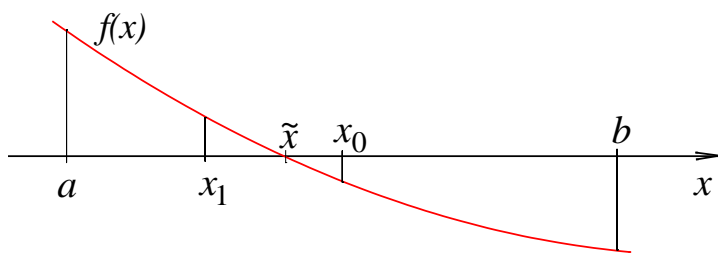


Рис. 4.1

является корнем уравнения. Если $f(x_0) < 0$, то искомый корень \tilde{x} находится на интервале $[a, x_0]$, если же $f(x_0) > 0$, то $\tilde{x} \in [x_0, b]$. Далее, из двух

интервалов $[a, x_0]$ и $[x_0, b]$ выбираем тот, на границах которого функция $f(x)$ имеет различные знаки, находим точку x_1 – середину выбранного интервала, вычисляем $f(x_1)$ и повторяем указанный процесс. В результате получаем последовательность интервалов, содержащих искомый корень \tilde{x} , причем длина каждого последующего интервала вдвое меньше, чем предыдущего. Процесс заканчивается, когда длина вновь полученного интервала станет меньше заданного числа $\varepsilon > 0$, и в качестве корня \tilde{x} приближенно принимается середина этого интервала.

Метод деления отрезка пополам обладает линейной скоростью сходимости. Для достижения точности $\varepsilon > 0$ достаточно выполнить N итераций ($N > \log_2(b - a) / \varepsilon$).

Пример 4.1. Найти корень уравнения $f(x) = \cos x - x = 0$ с точностью $\varepsilon = 10^{-3}$.

Решение. Построим эскиз графиков функций $y = \cos x$ и $y = x$. На рис. 4.2 видно, что искомый корень уравнения лежит между $a = 0$ и $b = \pi/2$, $f(a) > 0$, $f(b) < 0$. Используя метод деления отрезка пополам, получим

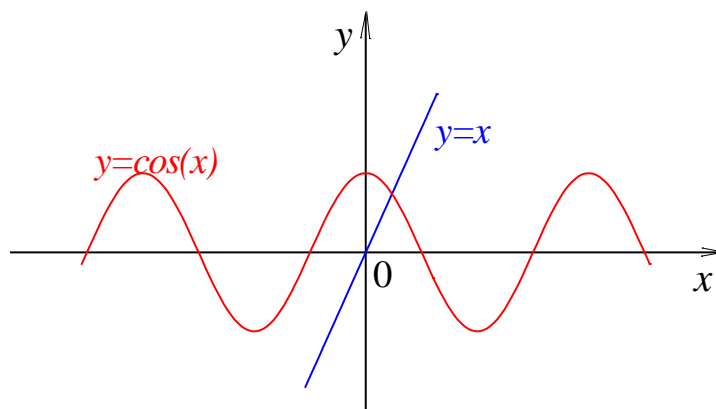


Рис. 4.2

№ итерации	a_n	b_n	$d_n = b_n - a_n $	x_n	$f(x_n)$
0	0,00000	1,57079	1,57079	0,78540	-0,07829
1	0,00000	0,78540	0,78540	0,39270	0,53118
2	0,39270	0,78540	0,39270	0,58905	0,24242
3	0,58905	0,78540	0,19635	0,68722	0,08579
4	0,68722	0,78540	0,09817	0,73631	0,00464
5	0,73631	0,78540	0,04909	0,76085	-0,03661
6	0,73631	0,76085	0,02454	0,74858	-0,01593
7	0,73631	0,74858	0,01227	0,74245	-0,00563
8	0,73631	0,74245	0,00614	0,73938	-0,00049
9	0,73631	0,73938	0,00307	0,73784	0,00208
10	0,73784	0,73938	0,00153	0,73861	0,00079
11	0,73861	0,73938	0,00077	0,73899	

В результате проведенной процедуры корень нелинейного уравнения $\cos x - x = 0$ равен 0,73899. Программа, реализующая метод деления отрезка пополам для данного примера, приведена в прил. 4.

Метод простой итерации (последовательных приближений). Он состоит в том, что уравнение (4.1) заменяется на интервале $[a, b]$ эквивалентным уравнением

$$x = \varphi(x). \quad (4.2)$$

Преобразование исходного уравнения обычно может быть выполнено разными способами, и от выбора способа зачастую во многом зависит эффективность решения.

Итерации образуются по правилу

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

причем задается начальное приближение x_0 .

Если последовательность чисел x_k имеет предел \tilde{x} при $k \rightarrow \infty$, то этот предел является корнем уравнения (4.2).

Метод простой итерации, в отличие от метода деления отрезка пополам, применим не всегда, поскольку последовательность значений x_k может не иметь конечного предела, даже если уравнение (4.2) имеет корень. Достаточное условие сходимости метода простой итерации можно сформулировать следующим образом:

Если функция $\varphi(x)$ отображает интервал $[a, b]$ действительной оси в тот же интервал и удовлетворяет условию Липшица: существует константа $q < 1$ такая, что

$$|\varphi(x') - \varphi(x'')| \leq q |x' - x''|$$

при $x', x'' \in [a, b]$, тогда уравнение (4.2) имеет на интервале $[a, b]$ одно, и только одно, решение \tilde{x} . В этом случае метод простой итерации сходится при любом начальном приближении $x_0 \in [a, b]$.

Скорость сходимости процесса характеризуется неравенством

$$|x_k - \tilde{x}| \leq q |x_{k-1} - \tilde{x}| \leq q^k |x_0 - \tilde{x}|.$$

Для погрешности справедлива оценка

$$|x_k - \tilde{x}| \leq \frac{q}{1-q} |x_k - x_{k-1}|.$$

Если у функции $\varphi(x)$ на интервале $[a, b]$ существует производная, удовлетворяющая неравенству

$$|\varphi'(x)| \leq q < 1,$$

то условие Липшица автоматически оказывается выполненным.

Пример 4.2. Вычислить корни уравнения

$$f(x) = e^x + e^{-x} + x - 4 = 0 \quad (4.3)$$

с точностью $\varepsilon = 10^{-4}$.

Решение. Построив эскиз графика функции $f(x)$, можно увидеть, что искомое уравнение имеет два корня: $x^{(1)} \in [0; 2]$ и $x^{(2)} \in [-2; -1]$ (рис. 4.3).

Для нахождения $x^{(1)}$ представим (4.3) в виде

$$x = \ln(4 - x - e^{-x}), \quad (4.4)$$

т.е. в виде (4.2) с правой частью $\varphi(x) = \ln(4 - x - e^{-x})$. У функции $\varphi(x)$ существует производная $\varphi'(x) = \frac{e^{-x} - 1}{4 - x - e^{-x}}$; на интервале $[0; 2]$ $|\varphi'(x)| \leq 0,464 < 1$. Следовательно, к решению уравнения (4.3) применим метод простой итерации. Взяв в качестве начального приближения $x_0 = (0 + 2)/2 = 1$, получим последовательность решений: $x_1 = 0,96779$; $x_2 = 0,97542$; $x_3 = 0,97363$; $x_4 = 0,97405$; $x_5 = 0,97396$. Заметим, что $|x_5 - x_4| < \varepsilon$, поэтому в качестве корня примем $x^{(1)} = 0,9740$.

Переходя к вычислению второго корня, заметим, что запись уравнения в виде (4.4) уже непригодна, поскольку на интервале $[-2; -1]$ производная $\varphi(x) = \ln(4 - x - e^{-x})$ больше единицы. На этот раз лучше переписать (4.3) в виде $x = -\ln(4 - x - e^x)$. Тогда на втором интервале

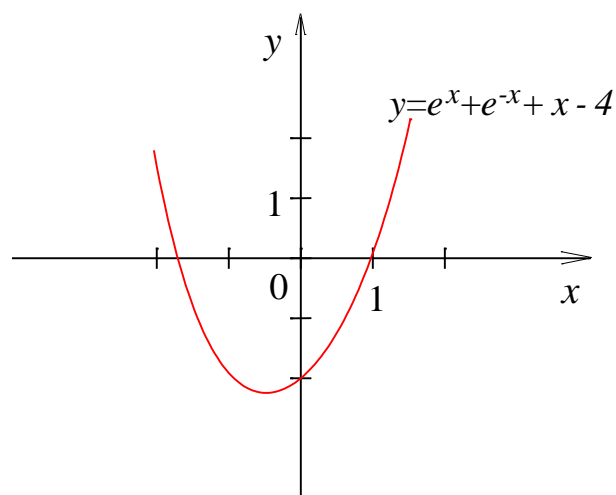


Рис. 4.3

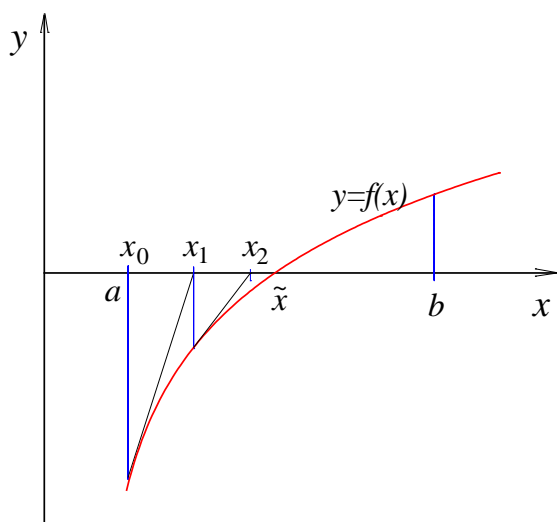


Рис. 4.4

$$|\varphi'(x)| = \left| \left(-\ln(4 - x - e^x) \right)' \right| = \left| \frac{e^x + 1}{4 - x - e^x} \right| \leq$$

и метод простой итерации снова оказывается сходящимся. Прodelывая аналогичную процедуру, что и в случае нахождения $x^{(1)}$, получим $x^{(2)} = -1,7100$.

Разобранный пример показывает, что сведение уравнения (4.3) к виду, удобному для применения

метода простой итерации, требует известного профессионализма и удачи, поскольку связано с довольно искусственными приемами.

Метод Ньютона (метод касательных). Пусть функция $f(x)$ имеет на интервале $[a, b]$ непрерывные производные $f'(x)$ и $f''(x)$, сохраняющие на $[a, b]$ постоянный знак. Для начала вычислений в методе Ньютона требуется задание одного начального приближения x_0 . Последующие приближения определяются по формуле

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \dots \quad (4.5)$$

Метод Ньютона называют также методом касательных, поскольку новое приближение x_{k+1} является абсциссой точки пересечения касательной, проведенной в точке $(x_k, f(x_k))$ к графику функции $f(x)$, с осью абсцисс (рис. 4.4).

Скорость сходимости процесса характеризуется неравенством

$$|x_k - \tilde{x}| \leq \frac{q}{2} |x_{k-1} - \tilde{x}|^2,$$

$$\text{где } q = \frac{\max |f''(x)|}{\min |f'(x)|}, \quad x \in [a, b].$$

Для погрешности справедлива оценка

$$|x_k - \tilde{x}| \leq \frac{2}{q} \cdot \left[\frac{q}{2} |x_k - x_{k-1}| \right]^{2^k}.$$

Метод Ньютона сойдется, если в качестве начального приближения x_0 принять такую точку $x \in [a, b]$, в которой $f(x) \cdot f''(x) > 0$.

Метод Ньютона относится к числу итерационных методов второго порядка и весьма быстро сходится. Точность каждого приближения в этом методе пропорциональна квадрату точности предыдущего.

Основной недостаток метода – необходимость достаточно точного начального приближения. Если начальное приближение выбрано неудачно, то метод может сходиться медленно, либо не сойдется вовсе. Поэтому метод Ньютона обычно применяется в комбинации с другими методами, дающим относительно грубые приближения, например, методом вилки, для того, чтобы уточнить решение за малое время.

Модифицированный метод Ньютона. Данный метод является более простым, чем предыдущий метод, поскольку производная $f'(x)$ вычисляется один раз в начальной точке x_0

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_0)}, \quad k = 0, 1, 2, \dots$$

Однако данная схема обладает лишь линейной сходимостью, т.е. $x_{k+1} - \tilde{x} = O(x_k - \tilde{x})$.

Задания к работе

1. Составить схемы алгоритмов решения нелинейных уравнений методами деления отрезка пополам, простой итерации и Ньютона.

2. Написать, отладить и выполнить программы решения нелинейных уравнений, приведенных в табл. 4.1 (в соответствии с вариантом задания). Нахождение одного из корней уравнения провести любым из вышеназванных методов с точностью до 10^{-4} . Интервал изоляции корня найти путем построения эскиза графика функции $f(x)$.

Вариант	Уравнение $f(x) = 0$
I	II
1	$x^2 - 5 \sin x = 0$
2	$\arcsin(2x + 1) - x^2 = 0$
3	$x - \sqrt{9 + x} + x^2 - 4 = 0$
4	$x^4 - 26x^3 + 131x^2 - 226x + 120 = 0$
5	$2x^2 + x - 7 = 0$
6	$e^x - 2(x - 1)^2 = 0$
7	$\sin x - x + 0,15 = 0$
8	$\sqrt{1 - x} - \operatorname{tg} x = 0$
9	$x^4 - 0,486x^3 - 5,792x^2 + 0,486x + 4,792 = 0$

Окончание табл. 4.1

I	II
10	$0,1e^x - \sin^2 x + 0,5 = 0$
11	$0,1\sin x + x^3 - 1 = 0$
12	$e^x - 10x = 0$
13	$0,1x^2 - x\ln x = 0$
14	$\sin x - 2x - 0,5 = 0$
15	$x - 1,25\ln x - 125 = 0$
16	$e^x - x - 1,25 = 0$

Контрольные вопросы

1. Назовите основные этапы процесса нахождения корня нелинейного уравнения.
2. Опишите алгоритм метода деления отрезка пополам.
3. Охарактеризуйте метод простой итерации. Как формулируется достаточное условие сходимости данного метода?
4. Опишите алгоритм метода Ньютона. В чем достоинство и недостаток этого метода?
5. Проведите сравнение методов по различным критериям.

Лабораторная работа 5. Численное решение задачи Коши для обыкновенных дифференциальных уравнений

Цель работы: изучение методов численного интегрирования обыкновенных дифференциальных уравнений, практическое решение уравнений на ЭВМ, сравнительный анализ рассмотренных методов.

Постановка задачи

Задача Коши для обыкновенного дифференциального уравнения n -го порядка

$$y^{(n)} = f(x, y, y', y'', \dots, y^{(n-1)}) \quad (5.1)$$

заключается в отыскании функции $y = y(x)$, удовлетворяющей этому уравнению и начальным условиям

$$y(x_0) = y_0, \quad y'(x_0) = y'_0, \dots, \quad y^{(n-1)}(x_0) = y_0^{(n-1)}, \quad (5.2)$$

где $x_0, y_0, y'_0, \dots, y_0^{(n-1)}$ – заданные числа.

Задача Коши для системы обыкновенных дифференциальных уравнений первого порядка

$$\left\{ \begin{array}{l} \frac{dy_1}{dx} = f_1(x, y_1, y_2, \dots, y_n), \\ \frac{dy_2}{dx} = f_2(x, y_1, y_2, \dots, y_n), \\ \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ \frac{dy_n}{dx} = f_n(x, y_1, y_2, \dots, y_n) \end{array} \right. \quad (5.3)$$

заключается в отыскании функций y_1, y_2, \dots, y_n , удовлетворяющих этой системе и начальным условиям

$$y_1(x_0) = y_{10}, y_2(x_0) = y_{20}, \dots, y_n(x_0) = y_{n0}. \quad (5.4)$$

Если удастся найти общее решение уравнения (5.1) или системы (5.3), то задача Коши сводится к отысканию значений произвольных постоянных. Но найти общее решение задачи Коши удастся в редких случаях; чаще всего приходится решать задачу Коши приближенно. Численные методы решения задачи Коши в зависимости от количества используемых для вычисления значения y_{r+1} предыдущих значений $y_r, y_{r-1}, \dots, y_{r-k+1}$ можно разделить на две группы:

одношаговые ($k=1$) и многошаговые ($k>1$) разностные методы. Данные методы обычно приводятся для решения одного обыкновенного дифференциального уравнения первого порядка, поскольку они легко распространяются на систему таких уравнений первого порядка.

Систему, содержащую производные высших порядков и разрешенную относительно старших производных искомых функций, путем введения новых неизвестных функций можно привести к виду (5.3). В частности, обыкновенное дифференциальное уравнение (5.1) приводится к виду (5.3) путем замены переменных

$$y_1 = y', \quad y_2 = y'', \quad \dots, \quad y_{n-1} = y^{(n-1)},$$

что дает следующую систему обыкновенных дифференциальных уравнений первого порядка

$$\left\{ \begin{array}{l} \frac{dy}{dx} = y_1, \\ \frac{dy_1}{dx} = y_2, \\ \dots \\ \frac{dy_{n-2}}{dx} = y_{n-1}, \\ \frac{dy_{n-1}}{dx} = f(x, y_1, y_2, \dots, y_{n-1}). \end{array} \right.$$

Метод Эйлера (метод Рунге-Кутты первого порядка точности). Рассмотрим дифференциальное уравнение первого порядка

$$y' = f(x, y) \quad (5.5)$$

с начальным условием

$$y(x_0) = y_0. \quad (5.6)$$

Выбрав достаточно малый шаг h , построим систему равноотстоящих точек $x_i = x_0 + i h$ ($i = 0, 1, 2, \dots$).

В методе Эйлера приближенные значения $y(x_i) \approx y_i$ вычисляются последовательно по формулам

$$y_{i+1} = y_i + h f(x_i, y_i), \quad (i = 0, 1, 2, \dots).$$

Если функция $f(x, y)$ удовлетворяет в некоторой области $D = \{ |x - x_0| \leq a, |y - y_0| \leq b \}$ условиям

$$|f(x, y_1) - f(x, y_2)| \leq K|y_1 - y_2|, \quad (K = \text{const} > 0),$$

$$\left| \frac{df}{dx} \right| = \left| \frac{\partial f}{\partial x} + f \frac{\partial f}{\partial y} \right| \leq M, \quad (M = \text{const} > 0),$$

то имеет место следующая оценка погрешности:

$$|\tilde{y}(x_i) - y_i| \leq \frac{hM}{2K} |(1+hK)^i - 1|,$$

где $\tilde{y}(x_i)$ – значение точного решения уравнения при $x = x_i$; y_i – приближенное значение, полученное на i -ом шаге.

Данная формула практически малоприменима. Обычно применяют двойной просчет: расчет производят с шагом h и $h/2$ и погрешность значения y_i^* при шаге $h/2$ оценивают, исходя из принципа Рунге, следующим образом:

$$\left| y_i^* - \tilde{y}(x_i) \right| \approx \left| y_i^* - y_i \right|.$$

Метод Эйлера легко распространяются на системы обыкновенных дифференциальных уравнений и на дифференциальные уравнения высших порядков. Последние должны быть предварительно приведены к системе дифференциальных уравнений первого порядка.

Приближенные значения $y_1(x_i) \approx y_{1,i}, y_2(x_i) \approx y_{2,i}, \dots, y_n(x_i) \approx y_{n,i}$ системы (5.3) с начальными условиями (5.4) вычисляются последовательно по формулам

[illegible]

Пример 5.1. Применяя метод Эйлера, найти на отрезке $[1;4]$ решение уравнения

$$y'' + \frac{2y'}{x} - y = 0$$

с начальными условиями $y(1) = 2,718$, $y'(1) = 0$, выбрав шаг $h = 0,1$.

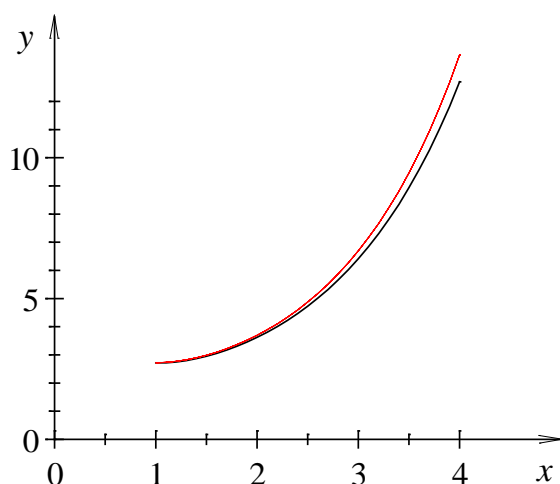


Рис. 5.1

Решение. С помощью замены переменных $y' = y_1$, $y'' = y_1'$ приведем исходное уравнение к системе уравнений

$$\begin{cases} y' = y_1, \\ y_1' = -\frac{2y_1}{x} + y \end{cases}$$

с начальными условиями $y(1) = 2,718$, $y_1(1) = 0$.

Таким образом,

$$f_1(x, y, y_1) = y_1,$$

$$f_2(x, y, y_1) = -\frac{2y_1}{x} + y.$$

Вычисления будем производить по формулам (5.7). Результаты расчетов представлены на рис. 5.1, для сравнения приведен график точного решения $y = \frac{e^x}{x}$ (штриховая линия). Программа, реализующая метод Эйлера, приведена в прил. 5.

Методы Рунге-Кутты. По-прежнему, рассматриваем дифференциальное уравнение (5.5) с начальным условием (5.6). Сущность явного m -этапного метода Рунге-Кутты состоит в том, что сначала вычисляются некоторые промежуточные значения

$$k_1 = (x_i, y_i),$$

$$k_2 = (x_i + \alpha_2 h, y_i + b_{21} h k_1),$$

$$\dots$$

$$k_m = (x_i + \alpha_m h, y_i + b_{m1} h k_1 + b_{m2} h k_2 + \dots + b_{m,m-1} h k_{m-1}),$$

а затем находится новое значение y_{i+1} из формулы

$$y_{i+1} = y_i + h \sum_{j=1}^m \sigma_j k_j. \quad (5.8)$$

Коэффициенты $\alpha_j, b_{js}, \sigma_j$ выбираются из соображений точности. Наибольшее распространение среди методов Рунге-Кутты получили методы второго и четвертого порядков точности (методы Рунге-Кутты при $m > 4$ не используются).

Методы второго порядка:

$$1) k_1 = f(x_i, y_i), \quad k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{hk_1}{2}\right), \quad (5.9)$$

$$y_{i+1} = y_i + hk_2;$$

$$2) k_1 = f(x_i, y_i), \quad k_2 = f(x_i + h, y_i + hk_1),$$

$$y_{i+1} = y_i + \frac{h}{2}(k_1 + k_2). \quad (5.10)$$

Оценку погрешности в точке x_i может быть получена с помощью двойного просчета: расчет повторяют с шагом $h/2$ и погрешность более точного значения y_i^* (при шаге $h/2$) оценивают приближенно так:

$$\left| y_i^* - \tilde{y}(x_i) \right| \approx \frac{1}{3} \left| y_i^* - y_i \right|.$$

Пример 5.2. Применяя метод Рунге-Кутты второго порядка точности, проинтегрировать уравнение

$$y' - y + \frac{2x}{y} = 0$$

с начальным условием $y(0) = 1$, выбрав шаг $h = 0,2$.

Решение. По формуле (5.9). Результаты вычислений занесем в табл. 5.1. Заполнение таблицы производится следующим образом. Записываем в первом столбце при $i=0$, $x_0 = 0$, $y_0 = 1$.

Вычисляем $k_1 = f(x_0, y_0) = y_0 - \frac{2x_0}{y_0} = 1,$

$k_2 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{hk_1}{2}\right) = y_0 + \frac{hk_1}{2} - \frac{2(x_0 + h/2)}{y_0 + hk_1/2} = 0,9182$. Тогда по формуле (5.9) при $i=0$ получаем $y_1 = y_0 + hk_2 = 1 + 0,2 \cdot 0,9182 = 1,1836$. Используя этот результат, записываем во втором столбце $i=1$, $x_1 = 0,2$, $y_1 = 1,1836$ и находим при этих значениях $k_1 = f(x_1, y_1) = 0,8456$,

Таблица 5.1

i	0	1	2	3	4	5
x	0	0,2	0,4	0,6	0,8	1,0
Точное решение	1,0	1,1832	1,3416	1,4832	1,6124	1,7320
Метод Эйлера	1,0	1,2000	1,3733	1,5294	1,6786	1,8237
Метод Рунге-Кутта (5.9)	1,0	1,1836	1,3426	1,4850	1,6152	1,7362
Метод Рунге-Кутта (5.10)	1,0	1,1867	1,3484	1,4938	1,6272	1,7542

$k_2 = f\left(x_1 + \frac{h}{2}, y_1 + \frac{hk_1}{2}\right) = 0,7950$ и $y_2 = y_1 + hk_2 = 1,3426$. При $i=2,3,4,5$ вычисления ведутся аналогично.

По формуле (5.10). Также при $i=0$ записываем $x_0 = 0$, $y_0 = 1$. Находим $k_1 = f(x_0, y_0) = 1$ и $k_2 = f(x_0 + h, y_0 + hk_1) = 0,8667$. Тогда по формуле (5.10) получаем $y_1 = y_0 + \frac{h}{2}(k_1 + k_2) = 1,1867$. После этого приступаем к заполнению второго столбца таблицы: $i=1$, $x_1 = 0,2$, $y_1 = 1,1867$. Затем вычисляем $k_1 = f(x_1, y_1) = 0,8497$, $k_2 = f(x_1 + h, y_1 + hk_1) = 0,7669$ и $y_2 = y_1 + \frac{h}{2}(k_1 + k_2) = 1,3484$. Заполнение таблицы при $i=2,3,4,5$ производится аналогично.

Более высокая точность методов Рунге-Кутта второго порядка по сравнению с методом Эйлера видна из табл. 5.1, в которой приведены значения точного решения $y = \sqrt{2x+1}$ и значения приближенных решений, полученных по формулам (5.9) и (5.10).

Методы четвертого порядка:

$$\begin{aligned}
1) \quad k_1 &= f(x_i, y_i), \quad k_2 = f\left(x_i + \frac{h}{4}, y_i + \frac{hk_1}{4}\right), \\
k_3 &= f\left(x_i + \frac{h}{2}, y_i + \frac{hk_2}{2}\right), \\
k_4 &= f(x_i + h, y_i + hk_1 - 2hk_2 + 2hk_3), \\
y_{i+1} &= y_i + \frac{h}{6}(k_1 + 4k_3 + k_4);
\end{aligned} \tag{5.11}$$

$$\begin{aligned}
2) \quad k_1 &= f(x_i, y_i), \quad k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{hk_1}{2}\right), \\
k_3 &= f\left(x_i + \frac{h}{2}, y_i + \frac{hk_2}{2}\right), \quad k_4 = f(x_i + h, y_i + hk_3), \\
y_{i+1} &= y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4).
\end{aligned} \tag{5.12}$$

Наиболее употребительна последняя расчетная схема.

Шаг расчета можно менять при переходе от одной точки к другой. Для контроля правильности выбора шага h по схеме (5.12) иногда рекомендуют вычислять дробь

$$\theta = \left| \frac{k_2 - k_3}{k_1 - k_2} \right|.$$

Величина θ не должна превышать нескольких сотых. В противном случае шаг h следует уменьшить. Метод Рунге-Кутты имеет порядок точности h^4 на всем интервале интегрирования. Оценка погрешности метода весьма затруднительна. Грубую оценку погрешности можно получить с помощью двойного просчета по формуле

$$\left| y_i^* - \tilde{y}(x_i) \right| \approx \frac{1}{15} \left| y_i^* - y_i \right|$$

При реализации метода Рунге-Кутта на ЭВМ с автоматическим выбором шага обычно в каждой точке x_i делают двойной просчет – сначала с шагом h , затем с шагом $h/2$. Если полученные при этом значения y_i различаются в пределах допустимой точности, то шаг h для следующей точки x_{i+1} удваивают, в противном случае берут половинный шаг.

Таблица 5.2

$h=0,05$			$h=0,1$			$h=0,2$		
x	y	k_j $j=1,2,3,4$	x	y	k_j $j=1,2,3,4$	x	y	k_j $j=1,2,3,4$
0	0	0	0	0	0	0	0	0
0,025	0	0,016668	0,05	0	0,033347	0,1	0	0,066778
0,025	0,000417	0,017016	0,05	0,001667	0,034737	0,1	0,006678	0,072354
0,05	0,000851	0,034057	0,1	0,003473	0,069679	0,2	0,014470	0,146383
0,05	0,000846		0,1	0,003432		0,2	0,014155	
0,05	0,000846	0,034057	0,1	0,003432	0,069644			
0,075	0,001697	0,051463	0,15	0,006914	0,106164			
0,075	0,002132	0,051826	0,15	0,008740	0,107693			
0,1	0,003438	0,069648	0,2	0,014202	0,146157			
0,1	0,003432		0,2	0,014158				

Пример 5.3. Методом Рунге-Кутты четвертого порядка (5.12) найти с точностью до $5 \cdot 10^{-6}$ решение дифференциального уравнения

$$y' = \frac{sh(0,5y + x)}{1,5} + 0,5y$$

с начальным условием $y(0) = 0$ на отрезке $[0; 0,2]$.

Решение. Для выбора шага вычислим решение в точке $x=0,1$ с шагом $h=0,1$ и с шагом $h=0,05$.

При вычислении с шагом $h=0,1$ последовательно имеем

$$k_1 = 0,$$

$$k_2 = \frac{sh(0,05)}{1,5} = 0,033347,$$

$$k_3 = \frac{sh(0,5 \cdot 0,001667 + 0,05)}{1,5} + 0,5 \cdot 0,001667 = 0,034737,$$

$$k_4 = \frac{sh(0,5 \cdot 0,003474 + 0,1)}{1,5} + 0,5 \cdot 0,003474 = 0,069679.$$

$$\text{Отсюда } y_1 = y_0 + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 0,003432.$$

Далее находим значение $y(0,1)$ как результат вычислений с шагом $h=0,05$ (см. табл. 5.2).

Поскольку полученные результаты совпадают в пределах заданной точности, вычисления продолжим с шагом $h=0,1$ и шагом $h=0,2$. Результаты дальнейших вычислений помещены в табл. 5.2. Сравнение результатов, полученных с шагом $h=0,1$ и с шагом $h=0,2$, показывает, что с точностью до $5 \cdot 10^{-6}$ можно принять $y(0,2) \approx 0,014158$ и что в дальнейшем шаг расчета следовало бы снова удвоить.

Задания к работе

1. Составить схемы алгоритмов решения задачи Коши для обыкновенных дифференциальных уравнений методами Эйлера и Рунге-Кутты.

2. Написать, отладить и выполнить программы решения дифференциальных уравнений, приведенных в табл. 5.3 (в соответствии с вариантом задания), методом Рунге-Кутты четвертого порядка точности. Предусмотреть в программе вычисление значений функции по заданному в таблице точному решению.

3. Результаты счета численным методом и по точному решению оформить в виде графика или таблицы.

4. Определить близость полученного заданным методом решения к точному значению с помощью оценок:

$$\delta_1 = \max |y_i - \tilde{y}_i|, \quad i = 1, 2, \dots, n,$$

$$\delta_2 = \sqrt{\sum_{i=1}^n (y_i - \tilde{y}_i)^2} / \sqrt{\sum_{i=1}^n (\tilde{y}_i)^2} - \text{интегральная оценка.}$$

Здесь \tilde{y}_i – точное решение, y_i – полученное приближенное решение.

Таблица 5.3

№	Дифференциальное уравнение	Начальные условия	Шаг h	Интервал интегрирования	Точное решение
I	II	III	IV	V	VI
1	$y'' - 2y' + y = 0$	$y(2) = 1$ $y'(2) = -2$	0,2	[2; 10]	$y = (7 - 3x)e^{x-2}$
2	$y'' - 3y' + 2y - 2x + 3 = 0$	$y(0) = 1$ $y'(0) = 2$	0,2	[0; 8]	$y = e^x + x$

3	$y'' + y = 4e^x$	$y(0) = 4$ $y'(0) = -3$	0,1	[0; 4]	$y = 2 \cos x + 2e^x - 5 \sin x$
4	$x^2 y'' + xy' = 0$	$y(1) = 5$ $y'(1) = -1$	0,05	[1; 3]	$y = 5 - \ln x$
5	$y'' - 2y' = 2e^x$	$y(1) = -1$ $y'(1) = 0$	0,1	[1; 5]	$y = e^{2x-1} - 2e^x + e - 1$
6	$y'' + 4y = \cos 3x$	$y(0) = 0,8$ $y'(0) = 2$	0,1	[0; 4]	$y = \cos 2x + \sin 2x - 0,2 \cos 3x$
7	$y'' + 2y' + 2y = xe^{-x}$	$y(0) = 0$ $y'(0) = 0$	0,1	[0; 4]	$y = e^{-x}(x - \sin x)$
8	$(1+x^2)y'' + (y')^2 + 1 = 0$	$y(0) = 1$ $y'(0) = 1$	0,05	[0; 2]	$y = 1 - x + 2 \cdot \ln(1+x)$

I	II	III	IV	V	VI
9	$y'' + 4y' + 4y = 0$	$y(0) = 1$ $y'(0) = -1$	0,1	[0; 4]	$y = (1 + x) e^{-2x}$
10	$y'' - 3y' = e^{5x}$	$y(0) = 2,2$ $y'(0) = 0,8$	0,02	[0; 0,8]	$y = 2 + 0,1(e^{3x} + e^{5x})$
11	$x^2 y'' - 2y = 0$	$y(1) = 0,83$ $y'(1) = 0,66$	0,1	[1; 5]	$y = 0,5x^2 + \frac{1}{3x}$
12	$y'' - 5y' + 6y = e^x$	$y(0) = 0$ $y'(0) = 0$	0,02	[0; 0,8]	$y = 0,5(e^{3x} + e^x) - e^{2x}$
13	$y'' + y = 1 + e^x$	$y(0) = 2,5$ $y'(0) = 1,5$	0,1	[0; 4]	$y = \cos x + \sin x +$ $+ 0,5e^x + 1$
14	$x^2 y'' + 2,5xy' - y = 0$	$y(1) = 2$ $y'(1) = 3,5$	0,1	[1; 5]	$y = 3\sqrt{x} - \frac{1}{x^2}$
15	$y'' + y = x^2 - x + 2$	$y(0) = 1$ $y'(0) = 0$	0,1	[0; 4]	$y = \cos x + \sin x +$ $+ x^2 - x$
16	$y'' - \frac{3}{x}y' = x$	$y(1) = 0$ $y'(1) = 0$	0,05	[1; 3]	$y = \frac{x^4}{4} - \frac{x^3}{3} + \frac{1}{12}$

Контрольные вопросы

1. Дайте определение обыкновенного дифференциального уравнения. Что значит решить дифференциальное уравнение?
2. Сформулируйте задачу Коши для одного дифференциального уравнения и для системы дифференциальных уравнений.
3. В чем состоит суть численных методов решения обыкновенных дифференциальных уравнений?
4. Охарактеризуйте метод Эйлера.
5. Опишите методы Рунге-Кутты.

Лабораторная работа 6. Численное решение краевой задачи для обыкновенных дифференциальных уравнений

Цель работы: изучение разностных методов решения краевой задачи для обыкновенных дифференциальных уравнений, практическое решение уравнений на ЭВМ.

Постановка задачи

Пусть дано дифференциальное уравнение второго порядка

$$F(x, y, y', y'') = 0. \quad (6.1)$$

Краевая задача для данного уравнения ставится следующим образом: найти функцию $y = y(x)$, которая внутри отрезка $[a, b]$ удовлетворяет уравнению (6.1), а на концах отрезка – краевым условиям

$$\begin{cases} \varphi_1[y(a), y'(a)] = 0, \\ \varphi_2[y(b), y'(b)] = 0. \end{cases} \quad (6.2)$$

Рассмотрим случай, когда уравнение (6.1) и граничные условия (6.2) линейны. Такая краевая задача называется линейной краевой задачей. В этом случае дифференциальное уравнение и краевые условия записываются в виде:

$$y'' + p(x)y' + q(x)y = f(x), \quad (6.3)$$

$$\begin{cases} \alpha_0 y(a) + \alpha_1 y'(a) = A, \\ \beta_0 y(b) + \beta_1 y'(b) = B, \end{cases} \quad (6.4)$$

где $p(x), q(x), f(x)$ – функции, определенные и непрерывные на интервале $[a, b]$; $\alpha_0, \alpha_1, \beta_0, \beta_1, A, B$ – заданные постоянные, причем $|\alpha_0| + |\alpha_1| \neq 0$ и $|\beta_0| + |\beta_1| \neq 0$.

Если $A = B = 0$, то краевые условия (6.4) называются однородными. Методы приближенного решения поставленных краевых задач можно разбить на две группы: разностные методы и аналитические методы. Рассмотрим методы первой группы.

Метод конечных разностей. Пусть $x_0 = a$, $x_n = b$, $x_i = x_0 + ih$ ($i = 1, 2, \dots, n-1$) – система равноотстоящих узлов с некоторым шагом $h = \frac{b-a}{n}$.

Обозначим функции $p(x)$, $q(x)$ и $f(x)$ в узлах x_i через p_i , q_i и f_i соответственно, а получаемые в результате расчета приближенные значения искомой функции $y(x)$ и ее производных $y'(x)$, $y''(x)$ через y_i , y'_i , y''_i . Заменим приближенно в каждом внутреннем узле производные $y'(x_i)$, $y''(x_i)$ конечно-разностными отношениями

$$y'_i = \frac{y_{i+1} - y_i}{h},$$

$$y''_i = \frac{y_{i+2} - 2y_{i+1} + y_i}{h^2},$$

а на концах положим

$$y'_0 = \frac{y_1 - y_0}{h}, \quad y'_n = \frac{y_n - y_{n-1}}{h}.$$

Используя эти формулы, приближенно заменим уравнение (6.3) и краевые условия (6.4) системой уравнений

$$\frac{y_{i+2} - 2y_{i+1} + y_i}{h^2} + p_i \frac{y_{i+1} - y_i}{h} + q_i y_i = f_i, \quad (i = 0, 1, \dots, n-2), \quad (6.5)$$

$$\begin{cases} \alpha_0 y_0 + \alpha_1 \frac{y_1 - y_0}{h} = A, \\ \beta_0 y_n + \beta_1 \frac{y_n - y_{n-1}}{h} = B. \end{cases}$$

Получим линейную алгебраическую систему $n+1$ уравнений с $n+1$ неизвестными, причем матрица коэффициентов этой систем является трехдиагональной. Специально для решения систем такого вида разработан достаточно простой метод – метод прогонки. Сущность его заключается в следующем.

Запишем сначала первые $n-1$ уравнений системы (6.5) в виде

$$y_{i+2} + v_i y_{i+1} + w_i y_i = \Psi_i,$$

где

$$v_i = -2 + h p_i, \quad w_i = 1 - h p_i + h^2 q_i, \quad \psi_i = h^2 f_i, \quad (i = 0, 1, \dots, n-2).$$

Затем написанная выше система приводится к виду

$$y_{i+1} = c_i y_{i+2} + d_i, \quad (i = 0, 1, \dots, n-2). \quad (6.6)$$

Числа c_i, d_i , называемые прогоночными коэффициентами, последовательно вычисляются по формулам:
при $i = 0$

$$c_0 = \left(\frac{w_0 \alpha_1}{\alpha_0 h - \alpha_1} - v_0 \right)^{-1},$$
$$d_0 = \frac{\psi_0 (\alpha_0 h - \alpha_1) - w_0 A h}{v_0 (\alpha_0 h - \alpha_1) - w_0 \alpha_1};$$

при $i = 1, 2, \dots, n-2$

$$c_i = -\frac{1}{v_i + w_i c_{i-1}},$$
$$d_i = \frac{\psi_i - w_i d_{i-1}}{v_i + w_i c_{i-1}}. \quad (6.7)$$

Вычисления производятся в следующем порядке.

Прямой ход. Вначале находим c_0, d_0 и затем, применяя последовательно рекуррентные формулы (6.7), получаем значения c_i, d_i при $i = 1, 2, \dots, n-2$.

Обратный ход. Из уравнения (6.6) при $i = n-2$ и из последнего уравнения системы (6.6) получаем

$$y_{n-1} = c_{n-2} y_n + d_{n-2},$$
$$\beta_0 y_n + \beta_1 \frac{y_n - y_{n-1}}{h} = B.$$

Разрешив эту систему относительно y_n , будем иметь

$$y_n = \frac{\beta_1 d_{n-2} + B h}{\beta_1 (1 - c_{n-2}) + \beta_0 h}. \quad (6.8)$$

Используя уже известные числа c_{n-2} , d_{n-2} находим y_n . Затем вычисляем значения y_i ($i = n-1, \dots, 1$), последовательно применяя рекуррентные формулы (6.6):

$$\begin{aligned} y_{n-1} &= c_{n-2} y_n + d_{n-2}, \\ y_{n-2} &= c_{n-3} y_{n-1} + d_{n-3}, \\ &\vdots \\ y_1 &= c_0 y_2 + d_0. \end{aligned} \tag{6.9}$$

Значение y_0 находим из предпоследнего уравнения системы (6.5):

$$y_0 = \frac{\alpha_1 y_1 - Ah}{\alpha_1 - \alpha_0 h}. \quad (6.10)$$

Таким образом, все вычисления как бы «прогоняются» два раза. Вычисления прямого хода заготавливают прогоночные коэффициенты c_i, d_i в порядке возрастания индекса i . При этом для вычисления значений c_0, d_0 используется краевое условие на левом конце отрезка интегрирования $[a, b]$. Затем на первом шаге обратного хода происходит согласование полученных чисел c_{n-2}, d_{n-2} с краевым условием на правом конце отрезка $[a, b]$, после чего последовательно получают значения искомой функции y_i в порядке убывания индекса i .

Пример 6.1. Найти приближенное решение уравнения

$$y'' - 2xy' - 2y = -4x, \quad (6.11)$$

удовлетворяющее краевым условиям

$$y(0) - y'(0) = 0, \quad y(1) = 1 + e = 3,718. \quad (6.12)$$

Решение. Возьмем $h=0,1$ и заменим исходное дифференциальное уравнение и краевые условия системой конечно-разностных отношений

$$\frac{y_{i+2} - 2y_{i+1} + y_i}{0.01} - 2x_i \frac{y_{i+1} - y_i}{0.1} - 2y_i = -4x_i, \quad (i = 0, 1, \dots, 8),$$

$$y_0 - \frac{y_1 - y_0}{0.1} = 0, \quad y_{10} = 3,718.$$

После приведения подобных членов получим

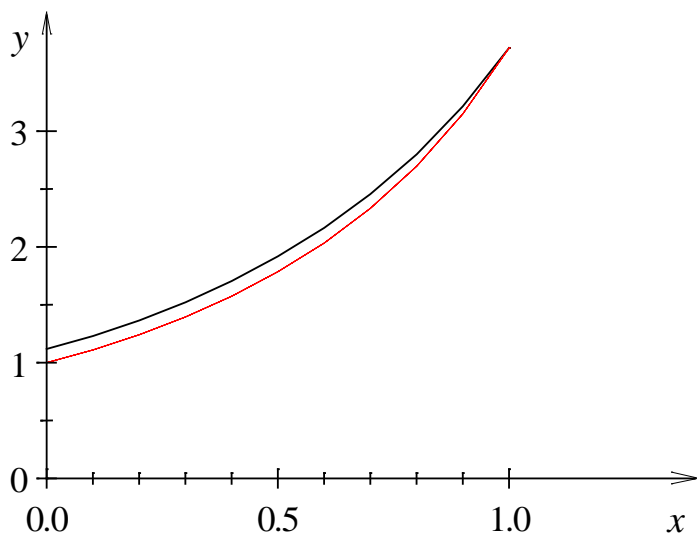


Рис. 6.1

$$y_{i+2} + (-2 - 0,2x_i)y_{i+1} + (0,98 +$$

Таким образом, имеем

$$v_i = -2 - 0,2x_i ,$$

$$w_i = 0,98 + 0,2x_i ,$$

$$\psi_i = -0,04x_i ,$$

$$\alpha_0 = 1,$$

$$\alpha_1 = -1,$$

$$A = 0,$$

$$\beta_0 = 1,$$

$$\beta_1 = 0,$$

$$B = 3,718.$$

Прямой ход. Сначала находим

$$c_0 = \left(\frac{0,98 \cdot (-1)}{0,1 + 1} + 2 \right)^{-1} = 0,902, \quad d_0 = 0.$$

и затем по формулам (6.7) приступаем к последовательному вычислению c_i, d_i при $i = 1, 2, \dots, 8$.

При $i = 1$

$$c_1 = -\frac{1}{v_1 + w_1 c_0} = -\frac{1}{-2,02 + 0,902} = 0,894,$$

$$d_1 = \frac{\psi_1 - w_1 d_0}{v_1 + w_1 c_0} = 0,004.$$

При $i = 2$

$$c_2 = -\frac{1}{v_2 + w_2 c_1} = 0,887, \quad d_2 = \frac{\psi_2 - w_2 d_1}{v_2 + w_2 c_1} = 0,010.$$

При $i = 3, 4, \dots, 8$ вычисления ведутся аналогично.

Обратный ход. По формуле (6.8) находим

$$y_{10} = \frac{B}{\beta_0} = 3,718.$$

После этого приступаем к последовательному вычислению значений y_i ($i = 9, 8, \dots, 1$) из соотношений (6.9). Значение y_0 находим по формуле (6.10)

$$y_0 = \frac{1,230}{1 + 0,1} = 1,118.$$

Результаты расчетов представлены на рис. 6.1, для сравнения приведен график точного решения $y = x + e^{x^2}$ (штриховая линия). Программа, реализующая данную схему, приведена в прил. 6.

Более точные формулы получаются, если заменить $y'(x_i)$, $y''(x_i)$ центрально-разностными отношениями

$$y'_i = \frac{y_{i+1} - y_{i-1}}{2h},$$

$$y''_i = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}.$$

Тогда получаем систему

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p_i \frac{y_{i+1} - y_{i-1}}{2h} + q_i y_i = f_i,$$

$$(i = 1, 2, \dots, n-1), \quad (6.13)$$

$$\begin{cases} \alpha_0 y_0 + \alpha_1 \frac{y_1 - y_0}{h} = A, \\ \beta_0 y_n + \beta_1 \frac{y_n - y_{n-1}}{h} = B. \end{cases}$$

Рассмотрим метод прогонки для решения данной системы уравнений. Запишем первые $n-1$ уравнений системы в виде

$$y_{i+1} + v_i y_i + w_i y_{i-1} = \psi_i,$$

где

$$v_i = \frac{2h^2 q_i - 4}{2 + h p_i}, \quad w_i = \frac{2 - h p_i}{2 + h p_i}, \quad \psi_i = \frac{2 f_i h^2}{2 + h p_i}, \quad (i = 1, 2, \dots, n-1).$$

Затем приводим эти уравнения к виду

$$y_i = c_i y_{i+1} + d_i \quad (i=1, 2, \dots, n-1). \quad (6.14)$$

где коэффициенты c_i , d_i вычисляются по формулам:
при $i=1$

$$c_1 = -\left(\frac{w_1 \alpha_1}{\alpha_1 - \alpha_0 h} + v_1 \right)^{-1},$$

$$d_1 = \frac{\psi_1(\alpha_1 - \alpha_0 h) + w_1 A h}{v_1(\alpha_1 - \alpha_0 h) + w_1 \alpha_1};$$

при $i=2, 3, \dots, n-1$ получаем выражения, аналогичные (6.7)

Вычисления производятся в следующем порядке.

Прямой ход. Вычисляем c_1 , d_1 , а затем по рекуррентным формулам (6.7) находим последовательно c_i , d_i ($i=2, 3, \dots, n-1$).

Обратный ход. Запишем уравнение (6.14) при $i=n-1$ и последнее уравнение системы (6.13):

$$y_{n-1} = c_{n-1} y_n + d_{n-1},$$

$$\beta_0 y_n + \beta_1 \frac{y_n - y_{n-1}}{h} = B.$$

Решая эту систему относительно y_n , будем иметь

$$y_n = \frac{\beta_1 d_{n-1} + B h}{\beta_1 (1 - c_{n-1}) + \beta_0 h}. \quad (6.15)$$

Используя уже известные числа c_{n-1} , d_{n-1} находим y_n . Значения y_i ($i=n-1, \dots, 1$) получаем из рекуррентных формул (6.14). Для вычисления y_0 используем предпоследнее уравнение системы (6.13).

Пример 6.2. Найти приближенное решение уравнения (6.11) с краевыми условиями (6.12).

Решение. Примем $h=0,1$ и заменим исходное дифференциальное уравнение и краевые условия системой центральных конечно-разностных отношений

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{0,01} - 2x_i \frac{y_{i+1} - y_{i-1}}{0,2} - 2y_i = -4x_i, \quad (i=1, 2, \dots, 9),$$

$$y_0 - \frac{y_1 - y_0}{0,1} = 0, \quad 2y_{10} - \frac{y_{10} - y_9}{0,1} = 1.$$

После приведения подобных членов получим

$$y_{i+1} + \left(-\frac{2,02}{1-0,1x_i} \right) y_i + \left(\frac{1+0,1x_i}{1-0,1x_i} \right) y_{i-1} = -\frac{0,04x_i}{1-0,1x_i}.$$

Таким образом, имеем

$$v_i = -\frac{2,02}{1-0,1x_i}, \quad w_i = \frac{1+0,1x_i}{1-0,1x_i}, \quad \psi_i = -\frac{0,04x_i}{1-0,1x_i},$$

$$\alpha_0 = 1, \quad \alpha_1 = -1, \quad A = 0,$$

$$\beta_0 = 1, \quad \beta_1 = 0, \quad B = 3,718.$$

Прямой ход. Записываем в табл. 6.1 числа $x_i = 0,1i$ и вычисляем значения v_i, w_i, ψ_i . Затем находим

$$c_1 = -\left(\frac{1,020 \cdot (-1)}{-1-0,1} - 2,040 \right)^{-1} = 0,899,$$

$$d_1 = \frac{-0,004 \cdot (-1-0,1)}{-2,040 \cdot (-1-0,1) + 1,020 \cdot (-1)} = 0,004.$$

Записываем полученные числа в табл. 6.1 числа и приступаем к последовательному определению c_i, d_i ($i=2,3,\dots,n-1$) по формулам (6.7). Все результаты записываем в столбцы прямого хода табл. 6.1.

Обратный ход. По формуле (6.15) находим

$$y_{10} = \frac{\beta_1 d_9 + B h}{\beta_1 (1 - c_9) + \beta_0 h} = 3,718.$$

Записываем полученное значение в последней строке табл. 6.1 при $i=10$. Затем последовательно находим y_i ($i = 9, \dots, 1$) по формулам (6.14); y_0 определяем из предпоследнего уравнения исходной системы.

В предпоследнем и последнем столбцах табл. 6.1 приводятся для сравнения значения приближенного решения y_i^* по формулам (6.5) – (6.10) и точного решения $\tilde{y}(x_i)$.

Оценка погрешности метода конечных разностей для задачи (6.3), (6.4) имеет вид

$$|y_i - \tilde{y}(x_i)| \leq \frac{h^2 M_4}{96} (b - a)^2,$$

где $\tilde{y}(x_i)$ – значение точного решения; $M_4 = \max_{[a, b]} |y^{(4)}(x)|$.

Данная формула практически малоприменима. Обычно применяют двойной пересчет и, исходя из принципа Рунге, получают приближенную оценку погрешности значения y_i :

$$|y_i^* - \tilde{y}(x_i)| \approx \frac{1}{3} |y_i^* - y_i|,$$

y_i и y_i^* – значения приближенных решений в точке $x = x_i$, полученных с шагами h и $h/2$ соответственно.

Точность разностного метода можно значительно повысить, если при замене производных использовать многоточечные

Таблица 6.1

i	x_i	Прямой ход		Обратный ход	y_i^*	\tilde{y}_i
		c_i	d_i	y_i		
0	0,0			1,047	1,118	1,000
1	0,1	0,899	0,004	1,151	1,230	1,110
2	0,2	0,888	0,011	1,227	1,364	1,241
3	0,3	0,878	0,021	1,426	1,521	1,394
4	0,4	0,867	0,034	1,601	1,704	1,574
5	0,5	0,856	0,050	1,808	1,917	1,784
6	0,6	0,845	0,069	2,053	2,164	2,033
7	0,7	0,833	0,092	2,348	2,455	2,332
8	0,8	0,821	0,117	2,708	2,800	2,696
9	0,9	0,809	0,145	3,154	3,214	3,148
10	1,0			3,718	3,718	3,718

разностные схемы.

В практических задачах часто встречаются уравнения, в которых функции $p(x)$, $q(x)$ и $f(x)$ заданы таблично с некоторым шагом h . Совершенно естественно такие уравнения решать разностным методом с данным шагом.

Задания к работе

1. Составить схемы алгоритмов решения краевой задачи для обыкновенных дифференциальных уравнений методом конечных разностей.

2. Написать, отладить и выполнить программы решения дифференциального уравнения с указанными краевыми условиями методом конечных разностей. Исходное уравнение заменить центрально-разностными отношениями. Полученную систему решить методом прогонки.

3. Результаты расчетов оформить в виде графика или таблицы.

Варианты заданий

1. $x^2 y'' + x y' = 1, \quad y(1) = 0, \quad y(2) = 0,2402.$

2. $y'' + f(x) y' + y \cos(0,7x) = 2x^2 + 2x - 4, \quad y(0) = 0, \quad y(1) = 0.$

x_i	0,0	0,2	0,4	0,6	0,8	1,0
$f(x_i)$	-1,7930	-1,7863	-1,7832	-1,7838	-1,7878	-1,7953

3. $y'' + 0,5x y' + (1 + 2\pi^2 x^2) y = 4x, \quad y(0) = 1, \quad y(1) = 1,367.$

4. $y'' + (1,4 + x^3) y' + (1 - x^2) y = e^{1-3x^2}, \quad y(0) = 0, \quad y(1) = 0.$

5. $y'' + (x - 1) y' + 3,125 y = 4x, \quad y(0) = 1, \quad y(1) = 1,367.$

6. $y'' + x^3 y' + (1,4 - x) y = \frac{x}{x^2 + 2,5}, \quad y(0) = 0, \quad y(1) = 0.$

$$7. \quad y'' + y' \sin 2,2x + y = \frac{1}{3,5 + \sin^2 2,2x}, \quad y(0) = 0, \quad y(1) = 0.$$

$$8. \quad y'' + y = 1, \quad y(0) = y(\pi) = 0.$$

$$9. \quad y'' + 2x y' + 2y = \frac{10 - 4x}{(2 - x)^3}, \quad y(0) = 1, \quad y(1) = 1,367.$$

$$10. \quad y'' + 2x y' + 2y = 4x, \quad y(0) = 1, \quad y(0,5) = 1,279.$$

$$11. \quad y'' + \frac{y'}{\sqrt{x^2 + 4}} + 1,8y = x, \quad y(0) = y(1) = 0.$$

$$12. \quad y'' + y = 2x - \pi, \quad y(0) = y(\pi) = 0.$$

$$13. \quad y'' + f(x) y' + y \cos(0,75x) = 2x^2 + 2x - 4, \quad y(0) = 0, \quad y(1) = 0.$$

x_i	0,0	0,2	0,4	0,6	0,8	1,0
$f(x_i)$	-1,6184	-1,5994	-1,5838	-1,5714	-1,5630	-1,5555

$$14. \quad y'' + 4y' + 4y = xe^{2x}, \quad y(0) = 0,9688, \quad y(1,5) = 1,3798.$$

$$15. \quad y'' - 5y' = 3x^2 + \sin 5x, \quad y(0) = 3,0200, \quad y(0,5) = 14,0755.$$

$$16. \quad y'' + f(x) y' + y \cos(0,85x) = 2x^2 + 2x - 4, \quad y(0) = 0, \quad y(1) = 0.$$

x_i	0,0	0,2	0,4	0,6	0,8	1,0
$f(x_i)$	-1,4747	-1,4480	-1,4246	-1,4043	-1,3869	-1,3722

Контрольные вопросы

1. Сформулируйте краевую задачу для обыкновенного дифференциального уравнения.

2. Дайте определение краевым условиям 1-го рода, 2-го рода и 3-го рода. Какие краевые условия называются однородными?
3. Охарактеризуйте метод конечных разностей.
4. В чем состоит суть метода прогонки?
5. Дайте оценку погрешности метода конечных разностей для краевой задачи. Как определяется приближенная оценка погрешности?

Лабораторная работа 7. Численное решение уравнений в частных производных гиперболического типа

Цель работы: изучение методов решения волнового уравнения, практическое решение уравнения на ЭВМ.

Постановка задачи

Рассмотрим смешанную задачу для волнового уравнения, заключающуюся в отыскании функции $u(x, t)$, удовлетворяющей уравнению

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}, \quad (7.1)$$

а также начальным условиям

$$u(x, 0) = f(x), \quad u_t(x, 0) = \Phi(x), \quad (0 \leq x \leq L) \quad (7.2)$$

и краевым условиям

$$u(0, t) = \varphi(t), \quad u(L, t) = \psi(t). \quad (7.3)$$

Построим в полуполосе $t \geq 0, 0 \leq x \leq L$ (рис. 7.1) два семейства параллельных прямых:

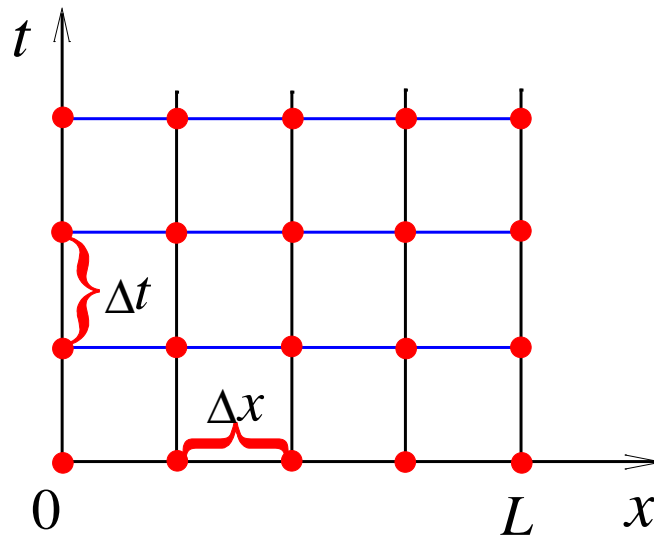


Рис. 7.1

$$x = j \Delta x \quad (j = 0, 1, 2, \dots, m),$$

$$t = n \Delta t \quad (n = 0, 1, 2, \dots).$$

Заменим производные в уравнении (7.1) конечно-разностными отношениями. Пользуясь симметричными формулами для производных, будем иметь

$$\frac{u_j^{n+1} - 2u_j^n + u_j^{n-1}}{(\Delta t)^2} = c^2 \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2}. \quad (7.4)$$

Обозначив $v = \frac{c \Delta t}{\Delta x}$, получим конечно-разностное уравнение

$$u_j^{n+1} = 2u_j^n - u_j^{n-1} + v^2 (u_{j+1}^n - 2u_j^n + u_{j-1}^n). \quad (7.5)$$

Доказано, что при $|v| \leq 1$ это конечно-разностное уравнение устойчиво.

В частности, при $v = 1$ уравнение (7.5) имеет наиболее простой вид:

$$u_j^{n+1} = u_{j+1}^n + u_{j-1}^n - u_j^{n-1}. \quad (7.6)$$

Оценка погрешности приближенного решения, полученного из уравнения (7.5) в полосе $0 \leq x \leq L$, $0 \leq t \leq T$, имеет вид

$$|u - \tilde{u}| \leq \frac{(\Delta x)^2}{12} \left[(M_4 \Delta x + 2M_3)T + T^2 M_4 \right],$$

где \tilde{u} – точное решение, $M_k = \max \left(\left| \frac{\partial^k u}{\partial t^k} \right|, \left| \frac{\partial^k u}{\partial x^k} \right| \right)$ ($k = 3, 4$).

Схема (7.5) является явной, поскольку позволяет найти значения функции $u(x, t)$ на $(n+1)$ -м временном слое, если известны значения на двух предыдущих слоях. Для того чтобы найти приближенное решение задачи (7.1) – (7.3), необходимо знать значения решения на двух начальных слоях. Их можно найти из начальных условий одним из следующих способов.

Первый способ. Заменим в начальном условии (7.2) производную $u_t(x, 0)$ конечно-разностным отношением

$$\frac{u_j^1 - u_j^0}{\Delta t} = \Phi(x_j) = \Phi_j ,$$

тогда для определения значений $u(x, t)$ на слоях $n = 0, n = 1$ получаем

$$u_j^0 = f_j, \quad u_j^1 = f_j + \Delta t \cdot \Phi_j.$$

Оценка погрешности значений u_j^1 в этом случае имеет вид

$$\left| \tilde{u}_j^1 - u_j^1 \right| \leq \frac{\alpha \Delta x}{2} M_2 ,$$

где

$$M_2 = \max \left\{ \left| \frac{\partial^2 u}{\partial x^2} \right|, \left| \frac{\partial^2 u}{\partial t^2} \right| \right\}.$$

Второй способ. Если функция $f(x)$ имеет конечную вторую производную, то значения u_j^1 можно определить с помощью формулы Тейлора

$$u_j^1 \approx u_j^0 + \frac{\partial u_j^0}{\partial t} \Delta t + \frac{\partial^2 u_j^0}{\partial t^2} \frac{(\Delta t)^2}{2}.$$

Используя уравнение (7.1) и начальные условия (7.2), можем записать

$$u_j^0 = f_j, \quad \frac{\partial u_j^0}{\partial t} = \Phi_j, \quad \frac{\partial^2 u_j^0}{\partial t^2} = f_j''.$$

Тогда будем иметь

$$u_j^1 \approx f_j + \Phi_j \Delta t + f_j'' \frac{(\Delta t)^2}{2}.$$

Погрешность значений u_j^1 , полученных по этой формуле, имеет порядок $O((\Delta t)^3)$.

Аналогичным образом применяется конечно-разностный метод при решении смешанной краевой задачи для неоднородного волнового уравнения

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = F(x, t).$$

В этом случае конечно-разностное уравнение имеет вид

$$u_j^{n+1} = 2u_j^n - u_j^{n-1} + v^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n) + F_j^n \cdot (\Delta t)^2.$$

Пример 7.1. Конечно-разностным методом найти решение задачи

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2},$$

$$u(x, 0) = x(\pi - x), \quad u_t(x, 0) = 0,$$

$$u(0, t) = u(\pi, t) = 0.$$

Решение. Выбираем шаг $\Delta x = \Delta t = \pi/18$. Значения $u(x, t)$ на первых двух временных слоях найдем вторым способом, используя формулу Тейлора. Все вычисления занесем в табл. 7.1.

1) Вычисляем значения $u_j^0 = f_j = x_j(\pi - x_j)$ ($j = 0, 1, \dots, 18$) и записываем в первую строку табл. 7.1. В силу симметрии задачи заполняем таблицу для $0 \leq x \leq \pi/2$. В первый столбец таблицы записываем краевые значения.

2) Определяем u_j^1 . В нашем примере $\Phi_j = 0$, $f_j'' = -2$. Таким образом, будем иметь

Таблица 7.1

$x_j \backslash t_n$	0	$\frac{\pi}{18}$	$\frac{\pi}{9}$	$\frac{\pi}{6}$	$\frac{2\pi}{9}$	$\frac{5\pi}{18}$	$\frac{\pi}{3}$	$\frac{7\pi}{18}$	$\frac{4\pi}{9}$	$\frac{\pi}{2}$
0	0	0,518	0,975	1,371	1,706	1,980	2,193	2,346	2,437	2,467
Δt	0	0,487	0,944	1,340	1,675	1,950	2,163	2,315	2,406	2,437
$2\Delta t$	0	0,426	0,853	1,249	1,584	1,858	2,071	2,224	2,315	2,346
$3\Delta t$	0	0,366	0,731	1,097	1,432	1,706	1,919	2,071	2,163	2,193
$4\Delta t$	0	0,305	0,609	0,914	1,218	1,493	1,706	1,858	1,950	1,980
$5\Delta t$	0	0,244	0,487	0,731	0,975	1,218	1,432	1,584	1,675	1,706

$$u_j^1 = u_j^0 - (\Delta t)^2 = u_j^0 - 0,03048.$$

Отсюда находим значения u_j^1 и записываем во вторую строку табл. 7.1.

3) Вычисляем значения u_j^{n+1} при $n=1,2,\dots$ по формуле (7.6).
При $n=1$ получаем

$$\begin{aligned} u_1^2 &= u_2^1 + u_0^1 - u_1^0 = 0,944 + 0 - 0,518 = 0,426, \\ u_2^2 &= u_3^1 + u_1^1 - u_2^0 = 1,340 + 0,487 - 0,975 = 0,853, \\ . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \\ u_9^2 &= u_{10}^1 + u_8^1 - u_9^0 = 2,406 + 2,406 - 2,467 = 2,346. \end{aligned}$$

Вычисления на последующих временных слоях проводятся аналогично.

Программа, реализующая схему (7.6) для данного примера, приведена в прил. 7.

Рассмотрим волновое уравнение первого порядка

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0, \quad c > 0. \quad (7.7)$$

Поскольку свойства решений данного соотношения близки к свойствам решения уравнения (7.1), то именно уравнение (7.7) выберем в качестве модельного при дальнейшем изучении конечно-разностных схем.

Метод Лакса. Известно, что явная конечно-разностная схема Эйлера является неустойчивой. Однако если заменить u_j^n на пространственное среднее $\frac{u_{j+1}^n + u_{j-1}^n}{2}$, то в результате получим схему Лакса

$$\frac{u_j^{n+1} - (u_{j+1}^n + u_{j-1}^n)/2}{\Delta t} + c \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = 0, \quad (7.8)$$

которая устойчива при $|v| \leq 1$.

Схема Лакса (7.8) является явной одношаговой схемой первого порядка точности с погрешностью аппроксимации $O((\Delta t, (\Delta x)^2 / \Delta t))$. Эта схема не всегда удовлетворяет условию согласованности, поскольку отношение $(\Delta x)^2 / \Delta t$ может и не стремиться к нулю при $\Delta t, \Delta x \rightarrow 0$. Однако если при $\Delta t, \Delta x \rightarrow 0$ число Куранта v сохраняется постоянным, то условие согласованности выполняется. При $v \neq 1$ схема Лакса дает высокий уровень диссипации, поэтому обычно в этой схеме полагают $v = 1$.

Метод Лакса-Вендроффа. Схему Лакса-Вендроффа можно построить, исходя из разложения в ряд Тейлора:

$$u_j^{n+1} = u_j^n + \Delta t \cdot u_t + \frac{(\Delta t)^2}{2} u_{tt} + O((\Delta t)^3).$$

Из волнового уравнения следует

$$u_t = -cu_x, \quad u_{tt} = c^2 u_{xx}.$$

Тогда предыдущее выражение можно переписать в виде

$$u_j^{n+1} = u_j^n - c \Delta t \cdot u_x + \frac{(\Delta t)^2}{2} c^2 u_{xx} + O((\Delta t)^3)$$

и заменим производные u_x и u_{xx} , используя центральные разности. В результате получим схему Лакса-Вендроффа

$$u_j^{n+1} = u_j^n - \frac{v}{2} (u_{j+1}^n - u_{j-1}^n) + \frac{v^2}{2} (u_{j+1}^n - 2u_j^n + u_{j-1}^n).$$

Это явная одношаговая конечно-разностная схема второго порядка точности с погрешностью аппроксимации $O((\Delta x)^2, (\Delta t)^2)$, устойчивая при $|v| \leq 1$.

Пример 7.2. Методом Лакса-Вендроффа найти решение задачи

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0,$$

$$u(x, 0) = x(\pi - x), \quad u(0, t) = 0, \quad 0 \leq x \leq \pi.$$

Решение. Выбираем шаг $\Delta x = \Delta t = \pi/18$, тогда схема Лакса-Вендроффа запишется в виде

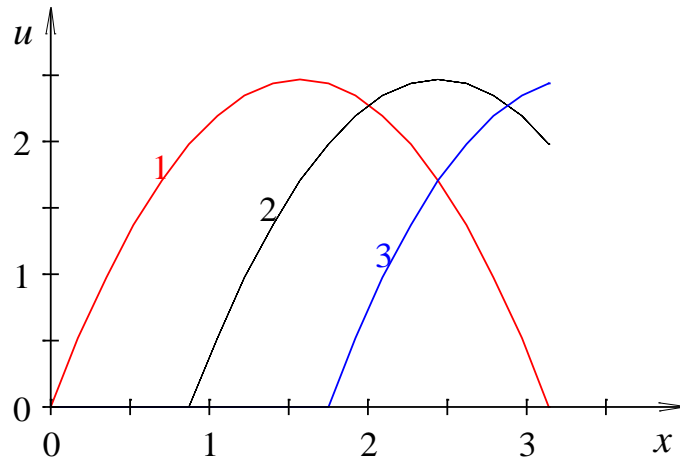


Рис. 7.2

$$u_j^{n+1} = u_{j-1}^n. \quad (7.9)$$

Распределение функции $u(x, t)$ при $n=0$ найдем из начального условия, т.е. $u_j^0 = x_j(\pi - x_j)$ ($j=0, 1, \dots, 18$). На левой границе искомая функция имеет постоянное во времени значение $u_0^n = 0$. Значения u_j^n при $n=1, 2, \dots$ вычисляем по формуле (7.9). Результаты расчетов в некоторые моменты времени представлены на рис. 7.2, кривые 1, 2 и 3 соответствуют $t = 0$, $\frac{5\pi}{18}$ и $\frac{5\pi}{9}$.

Метод Мак-Кормака. Применяя явный метод предиктор – корректор к линейному волновому уравнению, получим конечно-разностную схему Мак-Кормака:

$$\text{Предиктор:} \quad \overline{u_j^{n+1}} = u_j^n - v(u_{j+1}^n - u_j^n),$$

$$\text{Корректор:} \quad u_j^{n+1} = \frac{1}{2} \left[u_j^n + \overline{u_j^{n+1}} - v(\overline{u_j^{n+1}} - \overline{u_{j-1}^{n+1}}) \right].$$

Схема Мак-Кормака является явной двухшаговой конечно-разностной схемой второго порядка точности с погрешностью аппроксимации $O((\Delta x)^2, (\Delta t)^2)$, устойчивой при $|v| \leq 1$. При решении задач по этой схеме сначала находится некоторое предварительное

значение $\overline{u_j^{n+1}}$ величины u , а затем определяется окончательное значение u_j^{n+1} на $(n+1)$ -м шаге по времени (корректор).

Неявный метод Эйлера. Неявная конечно-разностная схема Эйлера имеет вид

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + \frac{c}{2\Delta x}(u_{j+1}^{n+1} - u_{j-1}^{n+1}) = 0.$$

Это схема первого порядка точности с погрешностью аппроксимации $O(\Delta t, (\Delta x)^2)$. Недостатком неявного метода Эйлера является необходимость на каждом шаге по времени решения системы алгебраических уравнений. Однако, метод абсолютно устойчивый и позволяет вести расчеты с существенно большим шагом по времени в отличие от явных методов.

Задания к работе

1. Составить схемы алгоритмов решения волнового уравнения первого порядка методами Лакса, Лакса-Вендроффа, Мак-Кормака и неявным методом Эйлера.

2. Написать, отладить и выполнить программы решения волнового уравнения

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0,$$

удовлетворяющего условиям

$$u(x, 0) = f(x), \quad u(0, t) = \varphi(t).$$

для значений $0 \leq x \leq 1$, взяв по аргументу x шаг $\Delta x = 0,05$. Выражения для функций $f(x)$ и $\varphi(t)$ приведены в табл. 7.2.

3. Результаты расчетов оформить в виде графика или таблицы.

Таблица 7.2

Вариант	Начальные и граничные условия	Метод решения
I	II	III
1	$f(x) = \frac{1}{x + 0,5}, \quad \varphi(t) = \frac{1}{t + 0,5}$	Неявный метод Эйлера

2	$f(x) = e^{-2,5x}, \quad \varphi(t) = e^{-2,5t}$	— // —
3	$f(x) = (1,1x^2 + 1,1)\sin \pi x, \quad \varphi(t) = 0$	— // —

I	II	III
4	$f(x) = x(1-x)\sin \pi x, \quad \varphi(t) = 0$	Неявный метод Эйлера
5	$f(x) = \frac{1}{x+1}, \quad \varphi(t) = \frac{1}{t+1}$	Метод Лакса-Вендроффа
6	$f(x) = e^{-x}, \quad \varphi(t) = e^{-t}$	— // —
7	$f(x) = (1,5x^2 + 1,3)\sin \pi x, \quad \varphi(t) = 0$	— // —
8	$f(x) = 0,75x(1-x), \quad \varphi(t) = 0$	— // —
9	$f(x) = \frac{1}{x+0,7}, \quad \varphi(t) = \frac{1}{t+0,7}$	Метод Мак-Кормака
10	$f(x) = e^{0,5x}, \quad \varphi(t) = e^{0,5t}$	— // —
11	$f(x) = (x+2)\sin \pi x, \quad \varphi(t) = 0$	— // —
12	$f(x) = x^2(1-x^2), \quad \varphi(t) = 0$	— // —
13	$f(x) = \frac{1}{x+1,5}, \quad \varphi(t) = \frac{1}{t+1,5}$	Метод Лакса
14	$f(x) = e^{2x}, \quad \varphi(t) = e^{2t}$	— // —
15	$f(x) = (1,4x^2 + 1,1)\sin \pi x, \quad \varphi(t) = 0$	— // —
16	$f(x) = 0,2x(1-x)\sin \pi x, \quad \varphi(t) = 0$	— // —

Контрольные вопросы

1. Какие физические процессы описывают уравнения в частных производных гиперболического типа?
2. Какая задача называется корректно поставленной?
3. Укажите методы построения конечно-разностных схем.
4. Какая конечно-разностная схема называется сильно неустойчивой (устойчивой)? Слабо неустойчивой (устойчивой)?
5. Дайте определение условия Куранта-Фридрихса-Леви. Каков физический и геометрический смысл данного условия?

Лабораторная работа 8. Численное решение уравнений в частных производных параболического типа

Цель работы: изучение методов решения уравнения теплопроводности, сравнительный анализ рассмотренных методов, практическое решение уравнения на ЭВМ.

Постановка задачи

Рассмотрим смешанную задачу или задачу Робина для уравнения теплопроводности, а именно: найти функцию $u = u(x, t)$, удовлетворяющую уравнению

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, \quad (8.1)$$

начальному условию

$$u(x, 0) = f(x), \quad (0 < x < L) \quad (8.2)$$

и краевым условиям

$$u(0, t) = \varphi(t), \quad u(L, t) = \psi(t). \quad (8.3)$$

К задаче (8.1) – (8.3) приводит, в частности, задача о распространении тепла в однородном стержне длины L .

Путем введения новой переменной $\tau = \sqrt{\alpha} t$ уравнение (8.1) приводится к виду

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2},$$

поэтому в дальнейшем примем $\alpha=1$.

Построим в полуполосе $t \geq 0$, $0 \leq x \leq L$ (рис. 7.1) два семейства параллельных прямых:

$$x = j \Delta x \quad (j = 0, 1, 2, \dots),$$

$$t = n \Delta t \quad (n = 0, 1, 2, \dots).$$

Обозначим $x_j = j \Delta x$, $t_n = n \Delta t$, $u_j^n = u(x_j, t_n)$ и заменим в каждом внутреннем узле (x_j, t_n) производную $\frac{\partial^2 u}{\partial x^2}$ конечно-разностным отношением

$$\left(\frac{\partial^2 u}{\partial x^2} \right)_{(x_j, t_n)} \approx \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2},$$

а производную $\frac{\partial u}{\partial t}$ одним из двух разностных отношений

$$\left(\frac{\partial u}{\partial t} \right)_{(x_j, t_n)} \approx \frac{u_j^{n+1} - u_j^n}{\Delta t},$$

$$\left(\frac{\partial u}{\partial t} \right)_{(x_j, t_n)} \approx \frac{u_j^n - u_j^{n-1}}{\Delta t}.$$

Тогда для уравнения (8.1) при $\alpha=1$ получаем два типа конечно - разностных уравнений:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2}, \quad (8.4)$$

$$\frac{u_j^n - u_j^{n-1}}{\Delta t} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2}. \quad (8.5)$$

Обозначив $r = \frac{\Delta t}{(\Delta x)^2}$, приведем эти уравнения к виду

$$u_j^{n+1} = (1 - 2r)u_j^n + r(u_{j+1}^n + u_{j-1}^n), \quad (8.6)$$

$$(1 + 2r)u_j^n - r(u_{j+1}^n + u_{j-1}^n) - u_j^{n-1} = 0. \quad (8.7)$$

Отметим, что для составления уравнения (8.4) была использована схема узлов, данная на рис. 8.1 – явная схема; для уравнения (8.5) – схема узлов, данная на рис. 8.2 – неявная схема. В этой связи зачастую метод (8.4) называют простым явным методом, а метод (8.5) называют простым неявным методом.

При выборе числа r в уравнениях (8.6) и (8.7) следует учитывать два обстоятельства:

1) погрешность замены дифференциального уравнения конечно-разностным должна быть наименьшей;

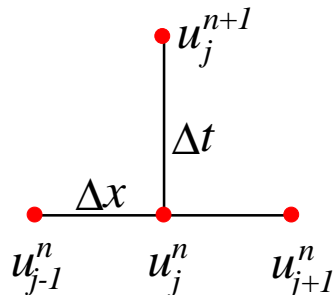


Рис. 8.1

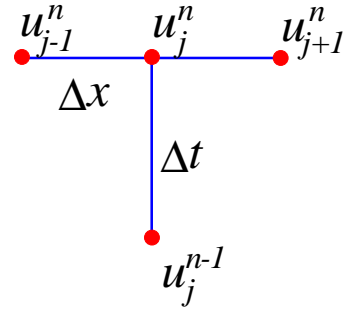


Рис. 8.2

2) конечно-разностное уравнение должно быть устойчивым. Доказано, что уравнение (8.6) будет устойчивым при $0 \leq r \leq 1/2$, а уравнение (8.7) – при любом r . Наиболее удобный вид уравнение (8.6) имеет при $r=1/2$:

$$u_j^{n+1} = \frac{1}{2}(u_{j+1}^n + u_{j-1}^n), \quad (8.8)$$

и при $r=1/6$:

$$u_j^{n+1} = \frac{1}{6}(u_{j+1}^n + 4u_j^n + u_{j-1}^n). \quad (8.9)$$

Оценки погрешностей приближенных решений, полученных из уравнений (8.8), (8.9) и (8.7) в полосе $0 \leq x \leq L$, $0 \leq t \leq T$, соответственно имеют вид

$$|u - \tilde{u}| \leq \frac{1}{3} T M_1 (\Delta x)^3,$$

$$|u - \tilde{u}| \leq \frac{1}{135} T M_2 (\Delta x)^4,$$

$$|u - \tilde{u}| \leq T \left(\frac{\Delta t}{2} + \frac{(\Delta x)^2}{12} \right) M_1,$$

где \tilde{u} – точное решение задачи (8.1) – (8.3),

$$M_1 = \max \left(|f^{(4)}(x)|, |\phi''(t)|, |\psi''(t)| \right) \text{ при } 0 \leq x \leq L, 0 \leq t \leq T,$$

$$M_2 = \max \left(|f^{(6)}(x)|, |\phi^{(4)}(t)|, |\psi^{(4)}(t)| \right) \text{ при } 0 \leq x \leq L, 0 \leq t \leq T.$$

Из приведенных оценок погрешностей видно, что уравнение (8.9) дает более высокую точность решения по сравнению с

уравнением (8.8). Но уравнение (8.8) имеет более простой вид, а, кроме того, шаг Δt для уравнения (8.9) должен быть значительно меньше, что приводит к большему объему вычислений. Уравнение (8.7) дает меньшую точность, но при этом шаги Δt и Δx выбираются независимо друг от друга. Уравнения (8.8) и (8.9) позволяют вычислить значения функции $u(x, t)$ на каждом слое по явным формулам через значения на предыдущем слое; уравнение (8.7) (неявная схема) этим свойством не обладает.

Конечно-разностным методом можно решать смешанную краевую задачу для неоднородного параболического уравнения

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2} + F(x, t).$$

Тогда соответствующее конечно-разностное уравнение, использующее явную схему узлов, имеет вид

$$u_j^{n+1} = (1 - 2r)u_j^n + r(u_{j+1}^n + u_{j-1}^n) + \Delta t \cdot F_j^n.$$

Отсюда получаем при $r=1/2$:

$$u_j^{n+1} = \frac{1}{2}(u_{j+1}^n + u_{j-1}^n) + \Delta t \cdot F_j^n, \quad (8.10)$$

при $r=1/6$:

$$u_j^{n+1} = \frac{1}{6}(u_{j+1}^n + 4u_j^n + u_{j-1}^n) + \Delta t \cdot F_j^n. \quad (8.11)$$

В этом случае имеют место следующие оценки погрешности: для уравнения (8.10)

$$|u - \tilde{u}| \leq \frac{1}{4} T \left(M_2 + \frac{1}{3} M_4 \right) (\Delta x)^2,$$

для уравнения (8.11)

$$|u - \tilde{u}| \leq \frac{1}{3} T \left(\frac{1}{3} M_3 + \frac{1}{5} M_6 \right) (\Delta x)^4,$$

где

$$M_2 = \max \left| \frac{\partial^2 u}{\partial t^2} \right|, \quad M_3 = \max \left| \frac{\partial^3 u}{\partial t^3} \right|, \quad M_4 = \max \left| \frac{\partial^4 u}{\partial x^4} \right|, \quad M_6 = \max \left| \frac{\partial^6 u}{\partial x^6} \right|.$$

Пример 8.1. Используя конечно-разностное уравнение (8.8), найти приближённое решение уравнения теплопроводности

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad (8.12)$$

удовлетворяющего условиям

$$u(x, 0) = \sin \pi x, \quad (0 \leq x \leq 1), \quad (8.13)$$

$$u(0, t) = u(1, t) = 0, \quad (0 \leq t \leq 0,025). \quad (8.14)$$

Решение. Выберем по аргументу x шаг $\Delta x = 0,1$. Поскольку $r = \frac{\Delta t}{(\Delta x)^2} = \frac{1}{2}$, получаем по аргументу t шаг $\Delta t = \frac{(\Delta x)^2}{2} = 0,005$.

Записываем в табл. 8.1 начальные и краевые значения. Учитывая их симметрию, заполняем таблицу только для $x=0; 0,1; 0,2; 0,3; 0,4; 0,5$. Значения функции $u(x, t)$ на первом слое находим, используя значения на начальном слое и краевые условия, по формуле (8.8) при $n=0$:

$$u_j^1 = \frac{u_{j+1}^0 + u_{j-1}^0}{2}.$$

Таблица 8.1

n	x t	0	0,1	0,2	0,3	0,4	0,5
0	0	0	0,3090	0,5878	0,8090	0,9511	1,0000
1	0,005	0	0,2939	0,5590	0,7694	0,9045	0,9511
2	0,010	0	0,2795	0,5317	0,7318	0,8602	0,9045
3	0,015	0	0,2658	0,5056	0,6959	0,8181	0,8602
4	0,020	0	0,2528	0,4809	0,6619	0,7781	0,8181
5	0,025	0	0,2404	0,4574	0,6295	0,7400	0,7781
$\tilde{u}(x, t)$	0,025	0	0,2414	0,4593	0,6321	0,7431	0,7813
$ u - \tilde{u} $	0,025	0	0,010	0,0019	0,0026	0,0031	0,0032

Таким образом, получаем

$$u_1^1 = \frac{u_2^0 + u_0^0}{2} = \frac{0,5878 + 0}{2} = 0,2939,$$

$$u_2^1 = \frac{u_3^0 + u_{10}^0}{2} = \frac{0,8090 + 0,3090}{2} = 0,5590$$

и т. д.

Записываем полученные значения u_j^1 ($j=1,2,3,4,5$) во вторую строку табл. 8.1. После этого переходим к вычислению значений на втором слое по формуле (8.8) при $n=1$:

$$u_j^2 = \frac{u_{j+1}^1 + u_{j-1}^1}{2}.$$

Подобным образом определяем последовательно значения u_j^n при $t = 0,005; 0,010; 0,015; 0,020; 0,025$. Результаты расчетов в некоторые моменты времени представлены на рис. 8.3, кривые 1, 2 и 3 соответствуют $t = 0, 0,005$ и $0,025$.

В двух последних строках табл. 8.1 приведены значения точного решения задачи $\tilde{u}(x,t) = e^{-\pi^2 t} \sin \pi x$ и модуля разности $|u - \tilde{u}|$ при $t=0,025$.

Для сравнения приведем соответствующую формуле (8.8) оценку погрешности. Для данной задачи

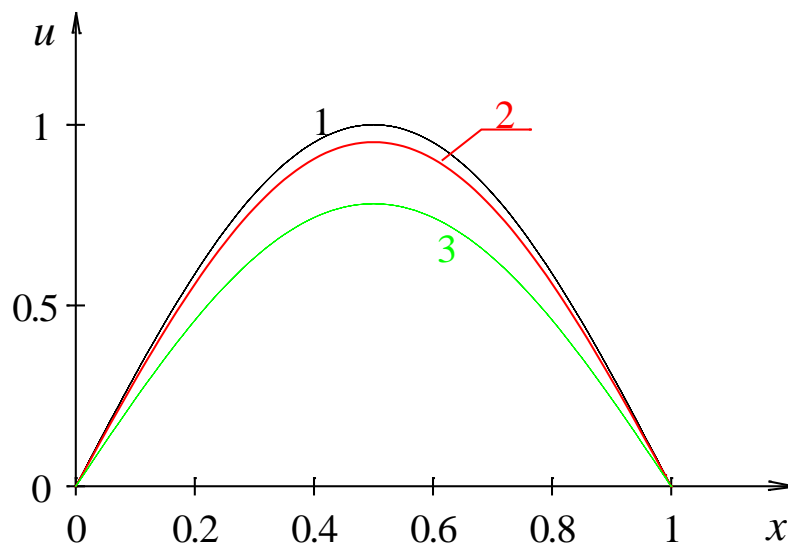


Рис. 8.3

$$\varphi(t) = \psi(t) = 0, \quad f^{(4)}(x) = \pi^4 \sin \pi x, \quad M_1 = \pi^4.$$

Таким образом, получаем

$$|u - \tilde{u}| \leq \frac{0,025}{3} \pi^4 (\Delta x)^2 = \frac{0,025}{3} \cdot 97,22 \cdot 0,01 = 0,0081.$$

Программа, реализующая схему (8.8) для данного примера, приведена в прил. 8.

Метод Кранка - Николсона. В данном методе уравнение теплопроводности аппроксимируется следующим выражением

$$\frac{u_j^n - u_j^{n-1}}{\Delta t} = \frac{(u_{j+1}^n - 2u_j^n + u_{j-1}^n) + (u_{j+1}^{n-1} - 2u_j^{n-1} + u_{j-1}^{n-1})}{2(\Delta x)^2}.$$

Разностная схема Кранка - Николсона является неявной и абсолютно устойчивой. Шаги Δt и Δx выбираются независимо друг от друга; метод имеет второй порядок точности с погрешностью аппроксимации $O((\Delta t)^2, (\Delta x)^2)$.

Обозначив $r = \frac{\Delta t}{(\Delta x)^2}$, приведем конечно-разностное уравнение к

виду

$$(1+r)u_j^n - \frac{r}{2}(u_{j+1}^n + u_{j-1}^n) = (1-r)u_j^{n-1} + \frac{r}{2}(u_{j+1}^{n-1} + u_{j-1}^{n-1}). \quad (8.15)$$

При использовании схемы (8.15) на каждом шаге по времени необходимо решать систему алгебраических уравнений с трехдиагональной матрицей. Системы уравнений такого вида эффективно решаются методом прогонки.

Пример 8.2. Используя разностную схему Кранка - Николсона (8.15), найти приближенное решение задачи (8.12) – (8.14) при $0 \leq t \leq 0,5$.

Решение. Выберем по пространственной координате x шаг $\Delta x = 0,02$, по времени шаг $\Delta t = 0,025$. Используя значения на начальном слое, на первом временном слое получим систему алгебраических уравнений.

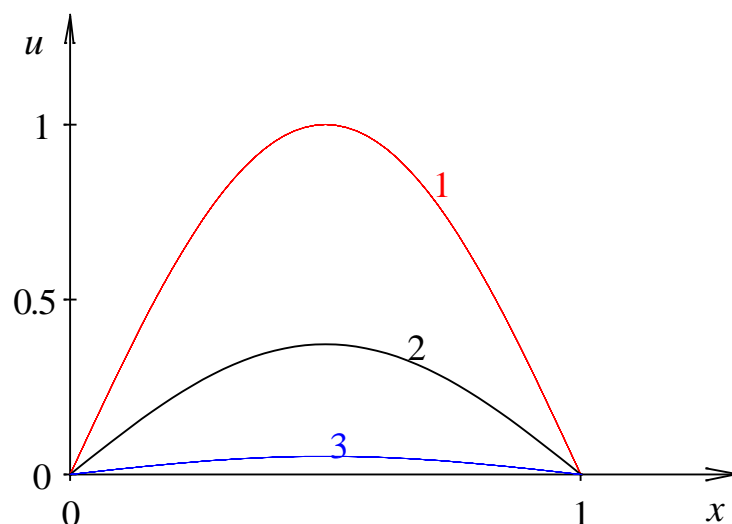


Рис. 8.4

$$\left\{ \begin{array}{l} (1+r)u_0^1 - \frac{r}{2}u_1^1 \dots\dots\dots = 1,9622, \\ -\frac{r}{2}u_0^1 + (1+r)u_1^1 - \frac{r}{2}u_2^1 \dots\dots\dots = 0,0550, \\ \dots\dots\dots \\ \dots\dots\dots - \frac{r}{2}u_{m-2}^1 + (1+r)u_{m-1}^1 - \frac{r}{2}u_m^1 = 0,0550, \\ \dots\dots\dots - \frac{r}{2}u_{m-1}^1 + (1+r)u_m^1 = 1,9622, \end{array} \right.$$

где $m = 1/\Delta x = 50$.

На левой и правой границах величины u_0^n и u_m^n заданы и равны 0. Решая систему методом прогонки, получим распределение величины u_j^n на первом временном слое. Используя эти значения, находим распределение величины u_j^n на втором временном слое и т.д. На рис. 8.4 представлены результаты расчетов в некоторые моменты времени; кривые 1, 2 и 3 соответствуют $t = 0, 0,1$ и $0,3$.

Задания к работе

1. Составить схемы алгоритмов решения уравнения теплопроводности простыми явным и неявным методами, методом Кранка - Николсона.

2. Написать, отладить и выполнить программы решения уравнения теплопроводности

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2},$$

удовлетворяющего условиям

$$u(x, 0) = f(x), \quad u(0, t) = \varphi(t), \quad u(1, t) = \psi(t).$$

для значений $0 \leq t \leq T$, взяв по аргументу x шаг $\Delta x = 0,05$. Выражения для функций $f(x)$, $\varphi(t)$ и $\psi(t)$ приведены в табл. 8.2.

3. Результаты расчетов оформить в виде графика или таблицы.

Таблица 8.2

Вариант	Начальные и граничные условия	Метод решения
I	II	III
1	$f(x) = (1,1x^2 + 1,5) \sin \pi x,$ $\varphi(t) = \psi(t) = 0, \quad T = 0,02$	Простой неявный метод
2	$f(x) = e^{-0,5x} \sin \frac{\pi x}{12}, \quad \varphi(t) = 0,$ $\psi(t) = e^{-0,5} \sin \frac{\pi}{12}, \quad T = 0,03$	— // —
3	$f(x) = (1,1x^2 + 2,5) e^{-x}, \quad \varphi(t) = 2,5,$ $\psi(t) = 3,6 e^{-1}, \quad T = 0,025$	— // —
4	$f(x) = x(1-x)(0,5x^4 + 0,9),$ $\varphi(t) = \psi(t) = 0, \quad T = 0,01$	— // —

Продолжение табл. 8.2

I	II	III
5	$f(x) = x \cos \pi x, \quad \varphi(t) = 0,$ $\psi(t) = -1, \quad T = 0,025$	Простой неявный метод
6	$f(x) = (1,3x^2 + 1,4) \sin \pi x,$ $\varphi(t) = \psi(t) = 0, \quad T = 0,025$	Метод Кранка - Николсона
7	$f(x) = e^{-0,3x} \sin \frac{\pi x}{4}, \quad \varphi(t) = 0,$ $\psi(t) = e^{-0,3} \sin \frac{\pi}{4}, \quad T = 0,035$	— // —
8	$f(x) = (1,3x^2 + 2,4) e^{-x}, \quad \varphi(t) = 2,4,$ $\psi(t) = 3,7 e^{-1}, \quad T = 0,035$	— // —
9	$f(x) = x(1-x)(0,7x^4 + 0,8),$ $\varphi(t) = \psi(t) = 0, \quad T = 0,015$	— // —
10	$f(x) = x^2 \cos \pi x, \quad \varphi(t) = 0,$ $\psi(t) = -1, \quad T = 0,025$	— // —
11	$f(x) = (1,5x^2 + 1,2) \sin \pi x,$ $\varphi(t) = \psi(t) = 0, \quad T = 0,03$	Простой явный метод, $r=1/6$
12	$f(x) = e^{-0,2x} \sin \frac{\pi x}{3}, \quad \varphi(t) = 0,$ $\psi(t) = e^{-0,2} \sin \frac{\pi}{3}, \quad T = 0,04$	— // —
13	$f(x) = (1,5x^2 + 2,3) e^{-x}, \quad \varphi(t) = 2,3,$ $\psi(t) = 3,8 e^{-1}, \quad T = 0,04$	— // —

I	II	III
14	$f(x) = x(1-x)(0,9x^4 + 0,6),$ $\varphi(t) = \psi(t) = 0, \quad T = 0,02$	Простой явный метод, $r=1/6$
15	$f(x) = 2x \cos \pi x, \quad \varphi(t) = 0,$ $\psi(t) = -2, \quad T = 0,025$	— // —
16	$f(x) = (1,1x^2 + 1,3) \sin \pi x,$ $\varphi(t) = \psi(t) = 0, \quad T = 0,015$	Простой явный метод, $r=1/2$

Контрольные вопросы

1. Какая задача называется маршевой?
2. Какие физические процессы описывают уравнения в частных производных параболического типа?
3. Какая конечно-разностная схема называется согласованной?
4. Дайте формулировку теоремы Лакса об эквивалентности.
5. Охарактеризуйте простой явный метод.
6. Опишите простой неявный метод.
7. Охарактеризуйте метод Кранка - Николсона.

Лабораторная работа 9. Численное решение уравнений в частных производных эллиптического типа

Цель работы: изучение методов решения уравнений Лапласа и Пуассона, практическое решение уравнений на ЭВМ.

Постановка задачи

Уравнение Пуассона является уравнением в частных производных эллиптического типа. В декартовой системе координат двумерное уравнение Пуассона имеет вид

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y). \quad (9.1)$$

При $f(x, y) \equiv 0$ выражение (9.1) носит название уравнения Лапласа, которое является модельным уравнением для уравнений в частных производных эллиптического типа.

Задача Дирихле для уравнения Пуассона ставится следующим образом: найти функцию $u = u(x, y)$, удовлетворяющую внутри некоторой области G уравнению (9.1), а на границе Γ – условию

$$u|_{\Gamma} = \varphi(x, y),$$

где $\varphi(x, y)$ – заданная непрерывная функция.

Выбрав шаги Δx и Δy по x и y соответственно, строим сетку

$$x_i = x_0 + i \Delta x \quad (i = 0, \pm 1, \pm 2, \dots),$$

$$y_j = y_0 + j \Delta y \quad (j = 0, \pm 1, \pm 2, \dots)$$

и заменяем в каждом внутреннем узле (x_i, y_j) соотношение (9.1) конечно-разностными уравнениями

$$\frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{(\Delta x)^2} + \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{(\Delta y)^2} = f_{ij}, \quad (9.2)$$

где $f_{ij} = f(x_i, y_j)$.

Правая часть уравнения (9.2) называется пятиточечным разностным оператором Лапласа, т.к. она содержит значения функции $u(x, y)$ в пяти точках, а именно в точках (x_i, y_j) , $(x_{i\pm 1}, y_j)$, $(x_i, y_{j\pm 1})$ (см. рис. 9.1). Данное множество точек носит название шаблона

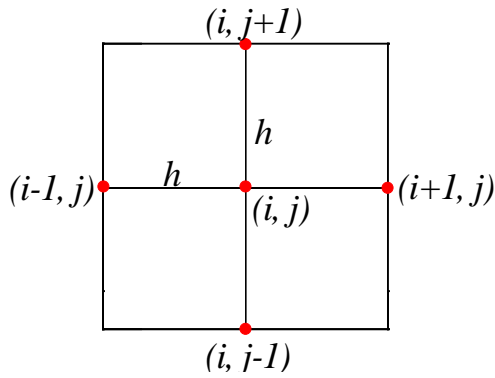


Рис. 9.1

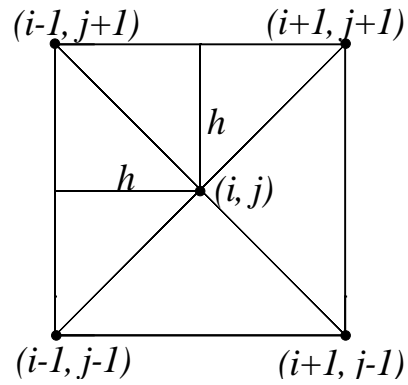


Рис. 9.2

разностного оператора. Возможны разностные аппроксимации оператором Лапласа и на шаблонах, содержащих большее количество точек.

Уравнение (9.2) вместе со значениями u_{ij} в граничных узлах образуют систему линейных алгебраических уравнений относительно значений функции $u(x, y)$ в узлах (x_i, y_j) . Наиболее простой вид эта система имеет для прямоугольной области и для $\Delta x = \Delta y = h$. В этом случае уравнения (9.2) записываются следующим образом:

$$u_{i+1, j} + u_{i-1, j} + u_{i, j+1} + u_{i, j-1} - 4u_{ij} = h^2 f_{ij}, \quad (9.3)$$

а значения в граничных узлах в точности равны значениям граничной функции. Для уравнения Лапласа и соответствующие конечно-разностные уравнения имеют вид

$$u_{ij} = \frac{1}{4} (u_{i+1, j} + u_{i-1, j} + u_{i, j+1} + u_{i, j-1}). \quad (9.4)$$

При составлении уравнений (9.3) и (9.4) была использована схема узлов, изображенная на рис. 9.1. Иногда бывает удобнее использовать схему узлов, показанную на рис. 9.2. В этом случае уравнению Лапласа соответствуют следующие конечно-разностные уравнения:

$$u_{ij} = \frac{1}{4} (u_{i-1, j-1} + u_{i+1, j-1} + u_{i-1, j+1} + u_{i+1, j+1}),$$

а для уравнения Пуассона будем иметь

$$u_{ij} = \frac{1}{4}(u_{i-1,j-1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i+1,j+1}) + \frac{1}{2}h^2 f_{ij}.$$

Погрешность замены дифференциального уравнения разностным, т.е. остаточный член ψ_{ij} для уравнения Лапласа, оценивается неравенством

$$|\psi_{ij}| \leq \frac{1}{6}h^2 M_4,$$

где

$$M_4 = \max_G \left\{ \left| \frac{\partial^4 u}{\partial x^4} \right|, \left| \frac{\partial^4 u}{\partial y^4} \right| \right\}.$$

Погрешность приближенного решения, полученного разностным методом, складывается из трех погрешностей:

- 1) погрешности замены дифференциального уравнения разностным;
- 2) погрешности аппроксимации краевых условий;
- 3) погрешности, получаемой в результате того, что система разностных уравнений решается приближенным методом.

Пример 9.1. Рассмотрим задачу о стационарном распределении тепла в плоской квадратной изолированной пластинке со стороной 1, если на границе пластинки поддерживается постоянная температура.

Известно, что функция $u(x, y)$, дающая распределение температуры, является решением уравнения Лапласа

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

при соответствующих краевых условиях. Для данной задачи краевые условия приведены на рис. 9.3.

Решение. Строим сетку с шагом $h=1/4$. Получим девять внутренних узлов (рис. 9.3). Записываем в этих узлах конечно-разностные уравнения.

В силу симметрии граничных условий имеем

$$u_{11} = u_{31}, \quad u_{12} = u_{32}, \quad u_{13} = u_{33}. \quad (9.5)$$

Это сокращает число неизвестных значений функции u во внутренних узлах до шести. Таким образом, в узлах (3, 1), (3, 2), (3, 3) конечно-разностные уравнения писать не нужно, в остальных шести внутренних узлах (1,1), (2,1), (1, 2), (2, 2), (1, 3), (2, 3) получаем соответственно шесть уравнений:

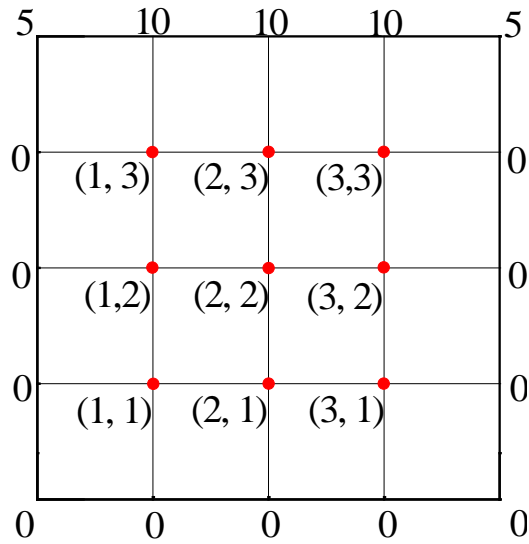


Рис. 9.3

$$\left\{ \begin{array}{l} u_{01} + u_{21} + u_{10} + u_{12} - 4u_{11} = 0, \\ u_{02} + u_{22} + u_{11} + u_{13} - 4u_{12} = 0, \\ u_{03} + u_{23} + u_{12} + u_{14} - 4u_{13} = 0, \\ u_{11} + u_{31} + u_{20} + u_{22} - 4u_{21} = 0, \\ u_{12} + u_{32} + u_{21} + u_{23} - 4u_{22} = 0, \\ u_{13} + u_{33} + u_{22} + u_{24} - 4u_{23} = 0. \end{array} \right.$$

В эти уравнения входят еще 12 значений функции в граничных точках. Эти значения мы берем из краевых условий

$$\begin{aligned} u_{i0} &= 0 \quad (i=1,2,3), \quad u_{0j} = 0 \quad (j=1,2,3), \\ u_{14} &= u_{24} = u_{34} = 10. \end{aligned} \quad (9.6)$$

Заметим, что в остальных узлах краевые условия не используются. Окончательно, учитывая условия (9.5) и (9.6), получим систему

$$\left\{ \begin{array}{ll} u_{21} + u_{12} - 4u_{11} = 0, & 2u_{11} + u_{22} - 4u_{21} = 0, \\ u_{22} + u_{11} + u_{13} - 4u_{12} = 0, & 2u_{12} + u_{21} + u_{23} - 4u_{22} = 0, \\ u_{23} + u_{12} - 4u_{13} = -10, & 2u_{13} + u_{22} - 4u_{23} = -10. \end{array} \right.$$

Разрешив эту систему методом Гаусса, будем иметь

$$u_{11} = 0,714, \quad u_{12} = 1,875, \quad u_{13} = 4,286,$$

$$u_{21} = 0,982, \quad u_{22} = 2,500, \quad u_{13} = 5,268.$$

Непосредственное решение системы конечно-разностных уравнений методами последовательного исключения при большом числе узлов, как видно из примера 9.1, оказывается слишком громоздким. Здесь более удобны итерационные методы решения, которые учитывают специальный вид таких систем и оказываются удобными для реализации на ЭВМ. Один из наиболее простых методов – процесс усреднения Либмана для систем (9.3), (9.4).

Согласно методу Либмана вычисления ведутся так: выбрав начальные приближения $u_{ij}^{(0)}$, последовательные приближения $u_{ij}^{(k+1)}$ для внутренних узлов сеточной области определяем по формуле

$$u_{ij}^{(k+1)} = \frac{u_{i+1,j}^{(k)} + u_{i-1,j}^{(k)} + u_{i,j+1}^{(k)} + u_{i,j-1}^{(k)}}{4} \quad (k = 0, 1, 2, \dots). \quad (9.7)$$

Для получения начальных приближений можно указать два способа:

- 1) значения $u_{ij}^{(0)}$ во внутренних узлах получают путем интерполяции, использующей известные граничные значения;
- 2) составляют систему конечно-разностных уравнений для сетки с более крупным шагом и решают ее методом исключения, а затем полученные значения интерполируют на узлы данной сетки.

Доказано, что для любого шага h процесс Либмана сходится к точному решению независимо от выбора начальных значений, т.е. существует предел

$$\lim_{k \rightarrow \infty} u_{ij}^{(k)} = u_{ij}.$$

Итерационный процесс будет сходиться значительно быстрее, если при вычислении последующих средних арифметических использовать не только значения предыдущего приближения, но и вновь найденные значения (метод Зейделя). Обычно итерации продолжают до тех пор, пока в двух последовательных приближениях не совпадет требуемое количество десятичных знаков.

Пример 9.2. Найти решение уравнения Лапласа для квадрата при краевых условиях, указанных на рис. 9.4.

Решение. 1. Вычисление начального приближения. Интерполируем граничные значения на внутренние узлы следующим образом. Начнем с верхней строки. Будем считать, что функция

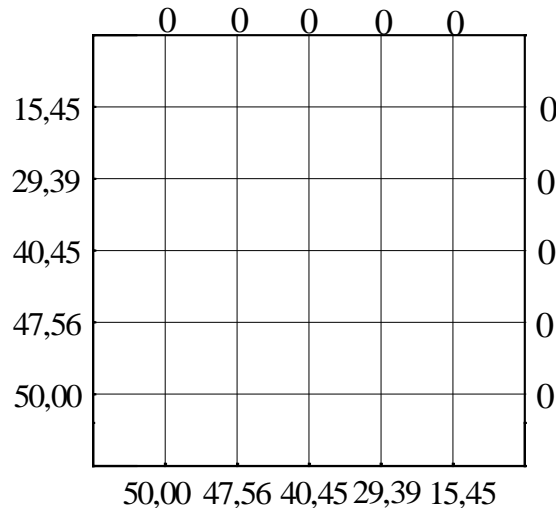


Рис. 9.4

$u(x, y)$ убывает линейно от значения 15,45 до 0. Это означает, что в качестве начального значения $u_{i5}^{(0)}$ возьмем

$$u_{i5}^{(0)} = \frac{15,45}{6}(6-i) \quad (i=1,2,3,4,5),$$

т.е. $u_{15}^{(0)} = 12,88$, $u_{25}^{(0)} = 10,30$, $u_{35}^{(0)} = 7,72$, $u_{45}^{(0)} = 5,15$, $u_{55}^{(0)} = 2,58$.

Аналогично поступим в правом столбце, принимая $u_{5j}^{(0)} = u_{i5}^{(0)}$.

Затем рассмотрим вторую (сверху) строку. Будем считать, что функция $u(x, y)$ убывает линейно от 29,39 до 5,15. Рассуждая также, как и в предыдущем случае, получаем значения $u_{i4}^{(0)}$ ($i=1,2,3,4$), а следовательно, и $u_{4j}^{(0)}$ ($j=1,2,3,4$). Этот процесс продолжаем до заполнения всей таблицы начального приближения (табл. 9.1).

Таблица 9.1

$j \backslash i$	1	2	3	4	5
5	12,88	10,30	7,72	5,15	2,58
4	24,54	19,69	14,85	10,00	5,15
3	34,05	27,65	21,25	14,85	7,72
2	40,92	34,29	27,65	19,69	10,30
1	45,46	40,92	34,05	24,54	12,88

2. Вычисление последовательных приближений. По формуле (9.7) при $k=1$ последовательно вычисляем

$$\begin{aligned} u_{15}^{(1)} &= \frac{1}{4} (u_{25}^{(1)} + u_{05}^{(1)} + u_{16}^{(1)} + u_{14}^{(1)}) = \\ &= \frac{1}{4} (10,30 + 15,45 + 0 + 24,54) = 12,57, \\ u_{14}^{(1)} &= \frac{1}{4} (u_{24}^{(1)} + u_{04}^{(1)} + u_{15}^{(1)} + u_{13}^{(1)}) = \\ &= \frac{1}{4} (19,69 + 29,39 + 34,05 + 12,88) = 24,00 \end{aligned}$$

и т.д.

Вычисления ведем до тех пор, пока значения двух последовательных итераций будут отличаться не более, чем, например, на 0,05, т.е. $|u_{ij}^{(k)} - u_{ij}^{(k-1)}| \leq \varepsilon = 0,05$ ($i, j = 1, 2, 3, 4, 5$).

Результаты 17-й итерации этому условию удовлетворяют. Окончательный результат приведен в табл. 9.2. Программа, реализующая данную схему, приведена в прил. 9.

Таблица 9.2

$j \backslash i$	1	2	3	4	5
5	11,78	8,98	6,62	4,41	2,21
4	22,64	17,49	13,02	8,75	4,41
3	31,84	25,18	19,14	13,02	6,22
2	39,03	32,13	25,18	17,49	8,98
1	44,52	39,03	31,84	22,64	11,78

Для оценки погрешности приближенного решения уравнения Лапласа можно использовать принцип Рунге, согласно которому погрешность ε приближенного решения u_h , полученного с шагом h , дается приближенной формулой

$$\varepsilon \approx \frac{|u_h - u_{2h}|}{3}, \quad (9.8)$$

Таблица 9.3

	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
6,75	5,86	5,06	4,34	3,68	3,04	2,42	1,82	1,21	0,60	0,00
13,38	11,60	10,04	8,63	7,32	6,06	4,84	3,63	2,42		0,00
19,70	17,12	14,86	12,81	10,89	9,04	7,23	5,43			0,00
25,60	22,34	19,47	16,85	14,38	11,98	9,60				0,00
30,95	27,17	23,82	20,73	17,78	14,88					0,00
35,66	31,56	27,89	24,46	21,12						0,00
39,67	35,50	31,72	28,09							0,00
42,98	39,05	35,38								0,00
45,63	42,34									0,00
	45,63	42,98	39,67	35,66	30,95	25,60	19,70	13,38	6,75	

где u_{2h} – приближенное решение, полученное с шагом $2h$. Можно отметить, что метод итераций приводит к выполнению стандартной операции усреднения в каждом внутреннем узле, поэтому он оказывается очень удобным для программирования на ЭВМ.

Пример 9.3. В табл. 9.3 дано приближенное решение уравнения Лапласа для единичного квадрата при указанных краевых значениях с шагом $h=0,1$ (в виду симметрии таблица заполнена наполовину). Требуется оценить погрешность этого решения по методу Рунге.

Решение. Заново решаем задачу с шагом $2h=0,2$, взяв за начальное приближение значения из табл. 9.3. Результаты вычислений с шагом $2h$ приведены в табл. 9.4. Затем находим разности $u_h - u_{2h}$ значений искомого решения, полученных с шагами $h=0,1$ и $2h=0,2$ и вычисляем погрешность ε по формуле (9.8), $\varepsilon \approx \frac{|35,38 - 35,46|}{3} \approx 0,03$.

Таблица 9.4

	0,00	0,00	0,00	0,00	
13,38	10,05	7,32	4,84	2,42	0,00
25,60	19,50	14,40	9,62		0,00
35,66	27,95	21,17			0,00
42,98	35,46				0,00
	42,98	35,66	25,60	13,38	

Задания к работе

1. Составить схемы алгоритмов решения уравнений Лапласа и Пуассона методом конечных разностей.

2. Написать, отладить и выполнить программы решения уравнения Лапласа методом конечных разностей с шагом $\Delta x = \Delta y = h = 0,1$ в квадрате с вершинами $A(0, 0)$, $B(0, 1)$, $C(1, 1)$, $D(1, 0)$. Краевые условия приведены в табл. 9.5. Полученную систему конечно-разностных уравнений решить методом Зейделя. Итерации проводить с точностью до 10^{-2} .

3. Результаты расчетов оформить в виде графика или таблицы.

Таблица 9.5

№	$u _{AB}$	$u _{BC}$	$u _{CD}$	$u _{AD}$
I	II	III	IV	V
1	$30y$	$30(1 - x^2)$	0	0
2	$30y$	$30\cos\frac{\pi x}{2}$	$30\cos\frac{\pi y}{2}$	0
3	$50y(1 - y^2)$	0	0	$50\sin\pi x$
4	$20y$	20	$20y^2$	$50x(1 - x)$
5	0	$50x(1 - x)$	$50y(1 - y^2)$	$50x(1 - x)$
6	$30\sin\pi y$	$20x$	$20y$	$30x(1 - x)$
7	$30(1 - y)$	$20\sqrt{x}$	$20y$	$30(1 - x)$
8	$50\sin\pi y$	$30\sqrt{x}$	$30y^2$	$50\sin\pi x$
9	$40y^2$	40	40	$40\cos\frac{\pi x}{2}$
10	$50y$	$50(1 - x)$	0	$60x, 0 \leq x < 0,5$ $60(1 - x), 0,5 \leq x \leq 1$
11	0	0	$10\sqrt{1 - y}$	$10\sin\frac{\pi x}{2}$

Окончание табл. 9.5

I	II	III	IV	V
12	0	$15 \sin \pi x$	0	$10x, 0 \leq x < 0,5$ $10(1-x), 0,5 \leq x \leq 1$
13	$20 \sin \frac{\pi y}{2}$	20	$20y$	$x(1-x)$
14	$\sqrt[3]{y}$	x	\sqrt{y}	$x(1-x)$
15	10	$10 \cos 2\pi x$	10	$10(1-x), 0 \leq x < 0,5$ $10x, 0,5 \leq x \leq 1$
16	0	$7x(1-x)$	0	$7 \sin \pi x$

Контрольные вопросы

1. Какие физические процессы описывают уравнения в частных производных эллиптического типа?
2. Сформулируйте задачу Дирихле для уравнения Лапласа.
3. Какому типу граничных условий соответствует задача Неймана? Задача Робина?
4. В чем состоит суть метода конечных разностей для уравнений в частных производных?
5. Из чего складывается погрешность решения разностным методом уравнения в частных производных?

Аппроксимация функции методом наименьших квадратов

Program Metod_nk;

Uses Crt;

Const m1=3;

n=20;

Var b :array [0..m1,0..m1] of real;

a, d :array [0..m1] of real;

x, y :array [0..n] of real;

S, P :real;

file1 :text;

i, j, k, m :integer;

Function Step(x:real;n:integer):real;

Begin

If x < > 0

Then Odd(n)

Then

Step:=x/abs(x)*exp(n*ln(abs(x)))

Else

Step:=exp(n*ln(abs(x)))

Else

If n=0 Then Step:=1 Else Step:=0;

End;

Procedure Coef;

Begin

Write(' Введите степень полинома m=');

ReadLn(m);

For j:=0 to m do {*** Начало цикла по j ***}

Begin

For k:=0 to m do

Begin

b[j,k]:=0;

For i:=1 to n do

b[j,k]:=b[j,k]+Step(x[i],j+k);

End;

d[j]:=0;

For i:=1 to n do

```

        d[j]:=d[j]+y[i]*Step(x[i],j);
End;          {*** Конец цикла по j ***}

        {*** Вывод коэффициентов системы уравнений ***}
For j:=0 to m do
    Begin
        For k:=0 to m do
            Write(' ',b[j,k]:5:0);
            Write(' ',d[j]:5:0);
            WriteLn
        End;
End;          {*** Конец Procedure Coef ***}

```

```

Procedure Gaus;
Begin
    For i:=1 to m do
        For j:=i to m do
            Begin
                For k:=0 to m do
                    b[j,k]:=b[j,k]-b[j,i-1]*b[i-1,k]/b[i-1,i-1];
                    d[j]:=d[j]-b[j,i-1]*d[i-1]/b[i-1,i-1];
                End;
            End;
        End;
    End;

    For i:=0 to m do
        a[i]:=0;
    End;
    For i:=m downto 0 do
        Begin
            S:=0;
            For k:=0 to m do
                S:=S+b[i,k]*a[k];
            End;
            a[i]:=(d[i]-S)/b[i,i];
        End;
    End;
End;          {*** Конец Procedure Gaus ***}

```

```

        {*** Начало программы ***}
BEGIN
    ClrScr;
    {*** Ввод данных ***}
    Assign(file1,'dann.dat');
    Reset(file1);
    If IOResult=0 Then
        Begin

```

```

        For i:=1 to n do
            Read(file1,x[i],y[i]);
        End;
    Close(file1);
    {*** Расчет коэффициентов полинома ***}
    Coef;
    Gaus;
    {*** Вывод коэффициентов полинома ***}
    For i:=m downto 0 do
        write(' a',i,'=',a[i]:3:3);
    ReadLn;

    Write(' Введите значение аргумента s=');
    ReadLn(s);
    P:=0;
    For i:=0 to m do
        P:=P+a[i]*Step(s,i);
    WriteLn(' Значение функции y=',P:5:4);
    ReadLn

END.

```

Приложение 2

Вычисление определенного интеграла по формуле прямоугольников

```

Program Integral;

Uses Crt;
Const pi=3.14159;

Var  a,b,I1,I2,h,eps,delta    :real;
     j,n                      :longint;

Function F(x:real) :real;
Begin
    F:=1/(x+cos(x))    {*** вид подынтегральной функции ***}

```

End;

Procedure PRJAM(var I :real);

Begin

I:=0;

h:=abs((b-a)/n);

For j:=1 to n do

I:=I+F(a+(j-1/2)*h); {*** формула прямоугольников ***}

I:=I*h

End;

Procedure INTEGR;

Begin

PRJAM(I1);

Repeat

n:=2*n;

PRJAM(I2);

delta:=abs(I1-I2);

I1:=I2

Until delta<eps

End;

{***** Начало программы *****}

BEGIN

ClrScr;

Write(' Введите значения a, b: ');

ReadLn(a,b);

Write(' Введите точность вычисления интеграла, eps= ');

ReadLn(eps);

h:=sqrt(eps);

n:=trunc((b-a)/h);

INTEGR;

WriteLn(' Приближенное значение интеграла = ',I1:6:5);

ReadLn

END.

Решение системы линейных алгебраических уравнений методом Гаусса

Program Gaus;

Const m=5;

Type matr = array [0..m,0..m] of real;
vect = array [0..m] of real;

Var a,a1,c :matr;
x,b,b1,d :vect;
file1 :text;
i,j,k,n :integer;

Procedure Gauss;

var S:real;

Begin

For k:=1 to n do {*** Прямой ход ***}

For i:=k to n do

Begin

b1[i]:=b[i];

For j:=k to n do

a1[i,j]:=a[i,j];

For j:=k to n do

c[i,j]:=a1[i,j]/a1[i,i];

For j:=k to n do

a[i,j]:=a1[i,j]-a1[i,k]*c[k,j];

d[i]:=b1[i]/a1[i,i];

b[i]:=b1[i]-a1[i,k]*d[k];

End;

For i:=1 to n do

For j:=1 to i do

If j<i Then c[i,j]:=0;

{*** Вывод системы уравнений в треугольном виде ***}

For i:=1 to n do

Begin

for j:=1 to n do

```

        Write(' ',c[i,j]:7:3);
        Write(' ',d[i]:7:3);
        WriteLn;
    End;
ReadLn;

    For i:=n downto 1 do      {*** Обратный ход ***}
    Begin
        S:=0;
        For k:=i+1 to n do
            S:=S+c[i,k]*x[k];
            x[i]:=d[i]-S;
        End;
    End;      {*** Конец procedure Gauss ***}

```

BEGIN

```

Write(' Введите число неизвестных : ');
ReadLn(n);

```

```

Assign(file1,'dann.dat');      {*** Ввод коэффициентов ***}
Reset(file1);
    For i:=1 to n do
    Begin
        for j:=1 to n do
            Read(file1,a[i,j]);
            Read(file1,b[i])
        End;
    Close(file1);

```

Gauss;

```

        {*** ВЫВОД значений неизвестных ***}
    For i:=1 to n do
        WriteLn(' x['i,']=',x[i]:3:1);
    ReadLn

```

END.

**Нахождение корня нелинейного уравнения
методом деления отрезка пополам**

```

Program Dichotom;

Uses Crt;
Const pi=3.14159;
      eps=1e-3;

Var  a,b,a1,b1,d,x :real;

      {*** Искомая функция ***}
Function F(x:real):real;
Begin
  F:=cos(x)-x
End;

Procedure Koren;
Begin
  If (F(a)>0) and (F(b)<0)
  Then
    Begin a1:=a; b1:=b End;
  If (F(a)<0) and (F(b)>0)
  Then
    Begin a1:=b; b1:=a End;

Repeat
  d:=abs(b1-a1);
  x:=a1+abs(b1-a1)/2;
  If F(x)=0
  Then
    Begin
      WriteLn(' корень уравнения = ',x:7:5);
      Exit
    End;
  If F(x)>0
  Then a1:=x
  Else b1:=x;
Until d<eps;
WriteLn(' корень уравнения = ',x:7:5)
End;

```

```

        {***** Начало программы *****}
BEGIN
  ClrScr;
  Write(' Введите интервал изоляции корня : a и b - ');
  ReadLn(a,b);
  Koren;
  ReadLn
END.

```

Приложение 5

Решение задачи Коши для обыкновенных дифференциальных уравнений методом Эйлера

```

Program Euler;

Uses Crt;
Const n=30;

Var  y,y1,ys      :array[0..n] of real;
     k1,k2        :real;
     x,x0,h        :real;
     i             :integer;

Function f1(x,y,y1:real):real;
Begin
  f1:=y1           {*** Первая функция ***}
End;

Function f2(x,y,y1:real):real;
Begin
  f2:=-2*y1/x+y    {*** Вторая функция ***}
End;

Procedure Vvod;
Begin
  x0:=1;
  y[0]:=2.718;

```



```

    y1[0]:=0;           {*** начальные значения ***}
    ys[0]:=y1[0];
    h:=0.1
End;    {*** конец процедуры Vvod ***}

```

```

    {*** Разностная схема Эйлера ***}
Procedure Mrk1;
Begin
    k1:=h*f1(x,y[i],y1[i]);
    k2:=h*f2(x,y[i],y1[i])
End;

```

```

    {*** Вычисление значений ***}
Procedure Calculation;
Begin
    For i:=0 to n-1 do
        Begin
            x:=x0+i*h;
            Mrk1;
            y[i+1]:=y[i]+k1;
            y1[i+1]:=y1[i]+k2
        End
    End;    {*** конец процедуры Calculation ***}

```

```

    {***** Начало программы *****}
BEGIN
ClrScr;
    Vvod;
    Calculation;
    For i:=0 to n do
        Begin
            x:=x0+i*h;
            ys[i]:=exp(x)/x;           {*** точное решение ***}
            WriteLn(' ',x:3:1,' ',y[i]:8:3,' ',ys[i]:8:3)
        End;
    ReadLn
END.

```

**Решение краевой задачи для обыкновенных
дифференциальных уравнений методом конечных разностей**

Program Rasnost;

Uses Crt;

Const n=10;

Var y,ys,v,w,ksi,c,d :array[0..n] of real;
 alf1,alf0,A,B,G :real;
 bet1,bet0,x,x0,h :real;
 i :integer;

Procedure Vvod;

Begin

Write(' Введите шаг h = ');

ReadLn(h);;

alf0:=1; alf1:=-1;

bet0:=1; bet1:=0;

A:=0; B:=3.718;

x0:=0;

End; {*** конец процедуры Vvod ***}

Procedure Calculation;

Begin

For i:=0 to n-2 do

Begin

x:=x0+i*h;

v[i]:=-2-0.2*x;

w[i]:=0.98+0.2*x;

ksi[i]:=-4*h*h*x

End;

{*** прогоночные коэффициенты ***}

G:=w[0]/(alf0*h-alf1);

c[0]:=1/(alf1*G-v[0]);

d[0]:=(ksi[0]-A*h*G)/(v[0]-alf1*G);

For i:=1 to n-2 do

Begin

c[i]:=-1/(v[i]+w[i]*c[i-1]);

d[i]:=(ksi[i]-w[i]*d[i-1])/(v[i]+w[i]*c[i-1])

End;

```

        {*** вычисление значений приближенного решения ***}
y[n]:=(B*h+d[n-2]*bet1)/(bet1*(1-c[n-2])+bet0*h);
For i:=n-1 downto 1 do
    y[i]:=c[i-1]*y[i+1]+d[i-1];
y[0]:=(alf1*y[1]-A*h)/(alf1-alf0*h)
End;    {*** конец процедуры Calculation ***}

        {*** Начало программы ***}
BEGIN
    ClrScr;
    Vvod;
    Calculation;
    For i:=0 to n do
        Begin
            x:=x0+i*h;
            ys[i]:=x+exp(x*x); {*** точное решение ***}
            WriteLn(' ',x:3:1,' ',y[i]:8:3,' ',ys[i]:8:3)
        End;
    ReadLn
END.

```

Приложение 7

Решение волнового уравнения простым явным методом

```

Program Voln;

Uses Crt;
Const n=18;
      pi=3.14159;
Var   u,u1,u2,f,f1,f2      :array[0..n] of real;
      dx,dt,x,T             :real;
      j,k                   :integer;

Procedure Nach;
Begin
    For j:=1 to n-1 do
        Begin
            x:=j*dx;
            f[j]:=x*(pi-x);

```

```

    f1[j]:=0;
    f2[j]:=-2;
    u[j]:=f[j];
    u1[j]:=f[j]+f1[j]*dt+f2[j]*dt*dt/2
End;

```

```

    u[0]:=0; u[n]:=0;
    u1[0]:=0; u1[n]:=0
End;    {*** конец процедуры Nach ***}

```

Procedure Calculation;

Begin

k:=1;

Repeat

k:=k+1;

For j:=1 to n-1 do

u2[j]:=u1[j+1]+u1[j-1]-u[j];

For j:=0 to n do

Begin

u[j]:=u1[j];

u1[j]:=u2[j];

WriteLn(' Значение функции

u('j*dx:4:3,',',k*dt:4:3,')=',u1[j]:4:3);

End;

ReadLn

Until k*dt>T;

End; {*** конец процедуры Calculation ***}

{*** Начало программы ***}

BEGIN

ClrScr;

Write(' Введите шаг по пространственной координате = ');

ReadLn(dx);

Write(' Введите шаг по времени = ');

ReadLn(dt);

Write(' Введите верхнее значение интервала по времени = ');

ReadLn(T);

Nach;

Calculation

END.

Приложение 8

**Решение уравнения теплопроводности при заданных
начальных и граничных условиях
простым явным методом**

Program Robin;

Uses Crt;

Const m=20;

 pi=3.14159;

Var u,u1,us :array[0..m] of real;

 dx,dt,T,x :real;

 i,k :integer;

 file1 :text;

 s,fname :string;

Procedure Nach;

 Begin

 For i:=0 to m do

 Begin

 x:=i*dx;

 u[i]:=x*cos(pi*x) {*** начальные условия ***}

 End

 End;

 {*** Расчет значений физической величины во времени ***}

Procedure Calculation;

 Begin

 k:=0;

 {*** Запись значений физической величины во внешний файл ***}

 str(k:1,s);

 fname:='dann'+s+'.dat';

 Assign(file1,fname);

 Rewrite(file1);

 If IOResult=0 Then

 For i:=0 to m do

 Begin

 x:=dx*i;

 Writeln(file1,x:3:2,' ',u[i]:6:4);

 End;

```

Close(file1);

While k*dt<T do
  Begin
    k:=k+1;

    u[0]:=0;  {*** условия на левой границе ***}
    u[m]:=-1; {*** условия на правой границе ***}

    For i:=1 to m-1 do
      u1[i]:=(u[i-1]+u[i+1])/2;

    For i:=1 to m-1 do
      u[i]:=u1[i];

    {*** Запись значений физической величины во внешний файл ***}
    str(k:1,s);
    fname:='dann'+s+'.dat';
    Assign(file1,fname);
    Rewrite(file1);
    If IOResult=0 Then
      For i:=0 to m do
        Begin
          x:=dx*i;
          Writeln(file1,x:3:2,' ',u[i]:6:4);
        End;
      Close(file1);
    End;

  End;  {*** конец procedure Calculation ***}

  {*** Начало программы ***}
BEGIN
  ClrScr;
  Write(' Задайте шаг по пространственной координате dx= ');
  ReadLn(dx);
  dt:=dx*dx/2;
  Write(' Задайте верхнюю границу во времени T= ');
  ReadLn(T);
  Nach;
  Calculation;
END.

```

Решение уравнения Лапласа при заданных краевых условиях методом конечных разностей

Program Laplac;

Uses Crt;

Const n=6;

Var u,u1 :array[0..n,0..n] of real;
 eps :real;
 i,j,m,k :integer;
 file1 :text;

 {*** Задаются граничные значения ***}

Procedure Gran;

 Begin

 Assign(file1,'dan94.dat');

 Reset(file1);

 If IOResult=0 Then

 Begin

 For i:=1 to n-1 do

 Begin

 u[i,n]:=0; {*** условия на верхней границе ***}

 ReadLn(file1,u[i,0]) {*** условия на нижней границе ***}

 End;

 For i:=n-1 downto 1 do

 Begin

 u[0,i]:=u[i,0]; {*** условия на левой границе ***}

 u[n,i]:=0 {*** условия на правой границе ***}

 End;

 End;

 Close(file1)

 End;

 {*** Задается начальное приближение ***}

Procedure Nach;

 Begin

```

For m:=n-1 downto 1 do
  For i:=1 to m do
    Begin
       $u[i,m]:=u[m+1,m]+(u[0,m]-u[m+1,m])*(m-i+1)/(m+1);$ 
       $u[m,i]:=u[i,m]$ 
    End;

    {*** Начальное распределение физической величины
    ***}
  WriteLn(' Начальное распределение');
  For j:=n-1 downto 1 do
    Begin
      For i:=1 to n-1 do
        Write(u[i,j]:5:2,' ');
      WriteLn
    End;
  ReadLn
End;

```

```

    {*** Расчет значений физической величины внутри области
    ***}

```

```

Procedure Calculation;
Begin
  k:=0;
  Repeat
    k:=k+1;
    m:=0;
    For i:=1 to n-1 do
      For j:=1 to n-1 do
         $u1[i,j]:=(u[i+1,j]+u[i-1,j]+u[i,j+1]+u[i,j-1])/4;$ 

      For i:=1 to n-1 do
        For j:=1 to n-1 do
          Begin
            If  $abs(u1[i,j]-u[i,j])<eps$  Then m:=m+1;
             $u[i,j]:=u1[i,j]$ 
          End
        End
      Until m=(n-1)*(n-1)
    End;

```

```

    {*** Полученное распределение физической величины ***}

```



```

Procedure Otvet;
Begin
  WriteLn(' Прошло ',k,' итераций: окончательный результат');
  For j:=n-1 downto 1 do
    Begin
      For i:=1 to n-1 do
        Write(u[i,j]:5:2,' ');
      WriteLn
    End;
  ReadLn
End;

```

```

      {*** Начало программы ***}
BEGIN
  ClrScr;
  Write(' Задайте точность расчетов; eps= ');
  ReadLn(eps);
  Gran;
  Nach;
  Calculation;
  Otvet
END.

```

ЛИТЕРАТУРА

1. Андерсон Д., Таннехилл Дж., Плетчер Р. *Вычислительная гидромеханика и теплообмен. Т.1.* – М.: Мир, 1990.
2. Бабенко К.И. *Основы численного анализа.* – М.: Наука, 1986.
3. Бахвалов Н.С. *Численные методы.* – М.: Наука, 1975.
4. Бахвалов Н.С., Жидков Н.П., Кобельков Г.Н. *Численные методы.* – М.: Наука, 1987.
5. Березин И.С., Жидков Н.П. *Методы вычислений. Т.1.* – М.: Наука, 1966.
6. Березин И.С., Жидков Н.П. *Методы вычислений. Т.2.* – М.: Физматгиз, 1962.
7. Волков Е.А. *Численные методы.* – М.: Наука, 1982.
8. *Вычислительная математика и техника в разведочной геофизике.* Под ред. Дмитриева В.И. – М.: Недра, 1990.
9. Годунов С.К., Рябенский В.С. *Разностные схемы.* – М.: Наука, 1973.
10. Гутер Р.С., Овчинский Б.В. *Элементы численного анализа и математической обработки результатов опыта.* – М.: Наука, 1970.
11. Калиткин Н.Н. *Численные методы.* – М.: Наука, 1978.
12. Копченова Н.В., Марон И.А. *Вычислительная математика в примерах и задачах.* – М.: Наука, 1972.
13. Кучумов Р.Я., Сыртланов В.Р., Мусакаев Н.Г. *Методы вычислений.* – Тюмень: Изд-во «Вектор Бук», 1998.
14. Марчук Г.И. *Методы вычислительной математики.* – М.: Наука, 1977.
15. Ортега Дж., Пул У. *Введение в численные методы решения.* – М.: Наука, 1986.
16. Рихтмайер Р., Мортон К. *Разностные методы решения краевых задач.* – М.: Мир, 1972.
17. Самарский А.А. *Введение в численные методы.* – М.: Наука, 1987.
18. Самарский А.А., Гулин А.В. *Численные методы.* – М.: Наука, 1989.
19. *Сборник задач по методам вычислений.* Под ред. Монастырного П.И. – Минск: Изд-во БГУ, 1983.
20. Тихонов А.Н., Костомаров Д.П. *Вводные лекции по прикладной математике.* – М.: Наука, 1984.

СОДЕРЖАНИЕ

Предисловие	3
Лабораторная работа 1. Интерполирование функций	4
Лабораторная работа 2. Приближенное вычисление интегралов	13
Лабораторная работа 3. Решение систем линейных алгебраических уравнений	21
Лабораторная работа 4. Решение нелинейных уравнений	32
Лабораторная работа 5. Численное решение задачи Коши для обыкновенных дифференциальных уравнений	40
Лабораторная работа 6. Численное решение краевой задачи для обыкновенных дифференциальных уравнений ..	50
Лабораторная работа 7. Численное решение уравнений в частных производных гиперболического типа	61
Лабораторная работа 8. Численное решение уравнений в частных производных параболического типа	70
Лабораторная работа 9. Численное решение уравнений в частных производных эллиптического типа	81
Приложение	91
Литература	108

Рашит Ямгитдинович Кучумов
Виль Ромэнович Сыртланов
Наиль Габсалямович Мусакаев

ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО КУРСУ
“ЧИСЛЕННЫЕ МЕТОДЫ”

Лицензия ЛРН№015071 от 28.12.93

Формат 60x84/16. Тираж 250 экз.

Отпечатано с готового набора в типографии издательства
«Вектор-БУК».

Россия, г. Тюмень,
ул. Володарского, 45.
Тел. (3452) 29-33-57