

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Томский государственный университет систем управления и радиоэлектроники
(ТУСУР)

Кафедра моделирования и системного анализа (МиСА)

В.Г. Баранник, Е.В. Истигечева

ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА

Учебное пособие

Томск 2014

Баранник В.Г., Истигечева Е.В. Вычислительная математика / Учебное пособие – Томск: Томский государственный университет систем управления и радиоэлектроники. Кафедра моделирования и системного анализа, 2014. – 83 с.

© Баранник В.Г., Истигечева Е.В. 2014.
© ТУСУР, каф. МиСА, 2014.

СОДЕРЖАНИЕ

Введение.....	4
1. Элементы теории погрешностей.....	20
2. Численное интегрирование.....	24
2.1. Постановка задачи.....	24
2.2. Формула прямоугольников	24
2.3. Формула трапеций.....	26
2.4. Формула Симпсона	27
2.5. Метод Монте–Карло	28
3. Численное решение систем линейных алгебраических уравнений (СЛАУ).....	31
3.1. Решение задач линейной алгебры	31
3.2. Метод Гаусса	33
3.3. Схема Гаусса с выбором главного элемента	37
3.4. Вычисление обратной матрицы методом Гаусса.....	39
3.5. Вычисление определителей методом Гаусса	40
3.6. Метод простой итерации (метод Якоби).....	42
3.7. Метод Зейделя	43
3.8. Метод скорейшего спуска для системы линейных алгебраических уравнений	45
4. Приближенное решение нелинейных и трансцендентных уравнений	51
4.1. Постановка задачи.....	51
4.2. Графическое решение уравнений	52
4.3. Метод половинного деления (дихотомии).....	52
4.4. Метод хорд.....	53
4.5. Метод Ньютона (метод касательных)	55
4.6. Комбинированный метод	56
5. Приближенное решение систем нелинейных уравнений	58
5.1. Метод Ньютона	58
5.2. Метод скорейшего спуска	61
6. Решение обыкновенных дифференциальных уравнений	64
6.1. Методы решения задачи Коши	64
6.2. Метод рядов, не требующий вычисления производных правой части уравнения	65
6.3. Метод Рунге-Кутты.....	66
6.4. Многошаговые методы	69
6.5. Экстраполяционные методы Адамса	69
6.6. Интерполяционные методы Адамса.....	70
7. Интерполирование и приближение функций	72
7.1. Задача интерполирования и аппроксимации функций.....	72
7.2. Интерполирование алгебраическими многочленами	73
7.3. Интерполяционная формула Ньютона.....	74
7.4. Задача обратного интерполирования	76
7.5. Оценка параметров эмпирических формул	77
7.6. Метод наименьших квадратов (МНК)	78
Литература	83

Введение

Первое применение вычислительных методов принадлежит древним египтянам, которые умели вычислять диагональ квадрата за конечное количество действий. Они также могли находить квадратный корень из 2, скорее всего, с помощью *алгоритма*, в дальнейшем получившего название формулы Герона, а еще позднее — метода Ньютона:

$$u_{k+1} = \frac{1}{2} \left(u_k + \frac{2}{u_k} \right), \quad u_0 = a.$$

С именем среднеазиатского врача, философа и математика Аль-Хоремзи связано понятие *алгоритма*. Разработкой вычислительных методов занимались Л.Эйлер, которому принадлежит, по-видимому, первый численный метод для решения обыкновенных дифференциальных уравнений, И.Ньютон, О.Л.Коши, Ж.Л.Лагранж, А.М.Лежандр, П.С.Лаплас, А.Пуанкаре, П.Л.Чебышёв и многие другие известные математики. Решающую роль в развитии вычислительной математики как самостоятельной науки сыграли немецкий математик Карл Рунге и русский математик, механик и кораблестроитель А.Н.Крылов.

В наше время вычислительная математика получила значительный импульс в 1950-е годы, что было связано с развитием ядерной физики, механики полета, аэродинамики спускаемых космических аппаратов. В дальнейшем решались задачи, связанные не только с расчетами действия ядерного взрыва и обтеканием боеголовок стратегических ракет. Численные методы нашли свое применение в таких областях как теория и методы оптимизации, математическая экономика, физика атмосферы, физика плазмы, механика горных пород и ледников, синергетика, биомеханика, и др. Наиболее наукоемки и требуют максимальных вычислительных ресурсов задачи физики, механики и электродинамики сплошных сред. К ним относятся системы уравнений в частных производных Эйлера, Лагранжа, Максвелла и др., а также задачи многомерной оптимизации.

Развитие многих вычислительных методов — заслуга многих ученых таких как академиков Дородницын А.А., Белоцерковский О.М., Самарский А.А., Рябенский В.С., Гельфанд И.М., Фаворский А.П..

Вычислительная математика отличается от других математических дисциплин и обладает специфическими особенностями.

1. Вычислительная математика имеет дело не только с непрерывными, но и с дискретными объектами. Так, вместо отрезка прямой часто рассматривается система точек

$$\{t_k\}_{k=0}^K,$$

вместо непрерывной функции $f(x)$ — табличная функция

$$\{f_k\}_{k=0}^K,$$

вместо первой производной — ее разностная аппроксимация, например,

$$\frac{f_{k+1} - f_k}{x_{k+1} - x_k},$$

$$k = 0 \div K, \quad x_{k+1} > x_k.$$

Такие замены, естественно, порождают погрешности метода.

2. В машинных вычислениях присутствуют числа с ограниченным количеством знаков после запятой из-за конечности длины мантииссы при представлении действительного числа в памяти ЭВМ. Другими словами, в вычислениях присутствует машинная погрешность (округления) δ_M . Это приводит к вычислительным эффектам, неизвестным, например, в классической теории обыкновенных дифференциальных уравнений, уравнений математической физики или в математическом анализе.
3. В вычислительной практике большое значение имеет *обусловленность задачи*, т.е. чувствительность ее решения к малым изменениям входных данных.
4. В отличие от "классической" математики выбор вычислительного *алгоритма* влияет на результаты вычислений.
5. Существенная черта численного метода — *экономичность* вычислительного *алгоритма*, т.е. минимизация числа элементарных операций при выполнении его на ЭВМ.
6. *Погрешности при численном решении* задач делятся на две категории — неустранимые и устранимые. К первым относят *погрешности*, связанные с построением математической модели объекта и приближенным заданием входных данных, ко вторым — *погрешности* метода решения задачи и ошибки округления, которые являются источниками малых возмущений, вносимых в решение задачи.

Специфику машинных вычислений можно пояснить на нескольких элементарных примерах.

Обусловленность задачи

Пример 1.1. Вычислить все корни уравнения

$$x^4 - 4x^3 + 8x^2 - 16x + \underbrace{15.99999999}_8 = (x - 2)^4 - 10^{-8} = 0.$$

Точное решение задачи легко найти:

$$(x - 2)^2 = \pm 10^{-4},$$

$$x_1 = 2,01; \quad x_2 = 1,99; \quad x_{3,4} = 2 \pm 0,01i.$$

Если компьютер работает при $\delta_M > 10^{-8}$, то свободный член в исходном уравнении будет округлен до 16,0 и, с точки зрения представления чисел с плавающей точкой, будет решаться уравнение $(x-2)^4 = 0$, т.е. $x_{1,2,3,4} = 2$, что, очевидно, неверно. В данном случае малые погрешности в задании свободного члена $\approx 10^{-8}$ привели, независимо от метода решения, к погрешности в решении $\approx 10^{-2}$.

Пример 1.2. Решается задача Коши для обыкновенного дифференциального уравнения 2-го порядка:

$$u''(t) = u(t), u(0) = 1, u'(0) = -1.$$

Общее решение имеет вид

$$u(t) = 0,5[u(0) + u'(0)]e^t + 0,5[u(0) - u'(0)]e^{-t}.$$

При заданных начальных данных точное решение задачи: $u(x) = e^{-t}$, однако малая погрешность δ в их задании приведет к появлению члена δe^t , который при больших значениях аргумента может существенно исказить решение.

Пример 1.3. Пусть необходимо найти решение обыкновенного дифференциального уравнения

$$\begin{aligned} \dot{u} &= 10u, & u &= u(t), \\ u(t_0) &= u_0, & t &\in [0, 1]. \end{aligned}$$

Его решение: $u(t) = u_0 e^{10(t-t_0)}$, однако значение $u(t_0)$ известно лишь приближенно: $u(t_0) \approx u_0^*$, и на самом деле $u^*(t) = u_0^* e^{10(t-t_0)}$.

Соответственно, разность $u^* - u$ будет

$$u^* - u = (u_0^* - u_0) e^{10(t-t_0)}.$$

Предположим, что необходимо гарантировать некоторую заданную точность вычислений $\varepsilon > 0$ всюду на отрезке $t \in [0, 1]$. Тогда должно выполняться условие

$$|u^*(t) - u(t)| \leq \varepsilon.$$

Очевидно, что

$$\max_{t \in [0, 1]} |u^*(t) - u(t)| = |u^*(1) - u(1)| = |u_0^* - u_0| e^{10(1-t_0)}.$$

Отсюда можно получить требования к точности задания начальных данных $\delta: |u_0^* - u_0| < \delta, \delta \leq \varepsilon e^{-10}$ при $t_0 = 0$.

Таким образом, требование к заданию точности начальных данных оказываются в e^{10} раз выше необходимой точности результата решения задачи. Это требование, скорее всего, окажется нереальным.

Решение оказывается очень чувствительным к заданию начальных данных. Такого рода задачи называются **плохо обусловленными**.

Пример 1.4. Решением системы линейных алгебраических уравнений (СЛАУ)

$$\begin{cases} u + 10v = 11 \\ 100u + 1001v = 1101 \end{cases}$$

является пара чисел $\{1, 1\}$.

Изменив правую часть системы на 0,01, получим возмущенную систему

$$\begin{cases} u + 10v = 11.01 \\ 100u + 1001v = 1101 \end{cases}$$

с решением $\{11.01; 0.00\}$, сильно отличающимся от решения невозмущенной системы. Эта система также плохо обусловлена.

Пример 1.5. Рассмотрим полином

$$(x - 1)(x - 2)\dots(x - 20) = x^{20} - 210x^{19} + \dots,$$

корни которого $x_1 = 1, x_2 = 2, \dots, x_{20} = 20$.

Положим, что коэффициент (-210) при x_{19} увеличен на $\approx 10^{-7}$. В результате вычислений с 11-ю значащими цифрами получим совершенно иные корни: $x_1 = 1,00; x_2 = 2,00; x_3 = 3,00; x_4 = 4,00; x_5 = 5,00; x_6 = 6,00; x_7 = 7,00; x_8 = 8,01; x_9 = 8,92; x_{10,11} = 10,1 \pm 0,644i; x_{12,13} = 11,8 \pm 1,65i; x_{14,15} = 14,0 \pm 2,52i; x_{16,17} = 16,7 \pm 2,81i; x_{18,19} = 19,5 \pm 19,4i; x_{20} = 20,8$.

Причина значительного расхождения также заключается в плохой *обусловленности задачи* вычисления корней рассматриваемого выражения.

Влияние выбора вычислительного алгоритма на результаты вычислений

Пример 1.6. Пусть необходимо вычислить значение выражения

$$\left(\frac{\sqrt{2}-1}{\sqrt{2}+1} \right)^3.$$

Избавившись от знаменателя, получаем $(\sqrt{2}-1)^6 = (3-2\sqrt{2})^3 = 99-70\sqrt{2}$.

Полагая

а)

$$\sqrt{2} \approx \frac{7}{5} = 1,4,$$

в)

$$\sqrt{2} \approx \frac{17}{12} = 1,41(6)$$

и рассматривая эти приближения как разные методы вычисления, получим следующие результаты:

$\sqrt{2}$	$(\sqrt{2} - 1)^6$	$(3 - 2\sqrt{2})^3$	$99 - 70\sqrt{2}$
7/5	0,004096	0,008000	1
17/12	0,005233	0,004630	- 0,1(6)

Очевидно, что столь значительное различие в результатах вызвано влиянием ошибки округления в задании $\sqrt{2}$.

Пример 1.7. Вычисление функции $\sin x$ с помощью ряда Тейлора.

Из курса математического анализа известно, что функция синус представляется своим рядом Тейлора

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots,$$

причем радиус сходимости ряда равен бесконечности — ряд сходится при любых значениях x .

Вычислим значения синуса при двух значениях аргумента. Пусть сначала $x = 0,5236(30^\circ)$. Будем учитывать лишь члены ряда, большие, чем 10^{-4} . Выполнив вычисления с четырьмя значащими цифрами, получим $\sin(0.5236) = 0.5000$, что соответствует принятой точности.

Пусть теперь $x = 25,66(1470^\circ)$. Если вычисления по данной формуле проводить с восемью значащими цифрами, то получим абсурдный результат: $\sin(25,66) \approx 24$ (учитывались члены ряда, большие, чем 10^{-8}).

Разумеется, выходом из создавшейся ситуации может быть использование формул приведения.

Пример 1.8. Вычисление функции e^x с помощью ряда Тейлора.

Из курса математического анализа известно, что экспонента представляется своим рядом Тейлора

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots,$$

радиус сходимости этого ряда также равен бесконечности.

Приведем некоторые результаты расчетов (e^x_M — значения экспоненты, вычисленные на компьютере).

x	e^x	e^x_M
1	2,718282	2,718282
20	$4,8516520 \cdot 10^8$	$4,8516531 \cdot 10^8$
-1	0,3678795	0,3678794

-10	$4,5399930 \cdot 10^{-6}$	$-1,6408609 \cdot 10^{-4}$
-20	$2,0611537 \cdot 10^{-9}$	1,202966

Выходом из этой ситуации может быть использование для отрицательных аргументов экспоненты формулы

$$e^{-x} = \frac{1}{e^x} = \frac{1}{1+x+\frac{x^2}{2!}+\dots}$$

Естественно ожидать рост ошибок округления при вычислении рассматриваемой функции при больших значениях аргумента x . В этом случае можно использовать формулу $e^x = e^{n+a} = e^n e^a$, где $n = [x]$.

Пример 1.9. Рассмотрим следующий метод вычисления интеграла

$$I_n = \int_0^1 x^n e^{1-x} dx, n = 1, 2, \dots$$

Интегрирование по частям дает

$$I_n = \int_0^1 x^n d(-e^{1-x}) - x^{n-1} \cdot e^{1-x} + \int_0^1 e^{1-x} \cdot d(x^n) = -1 + \int_0^1 n x^{n-1} e^{1-x} dx,$$

откуда следует

$$I_0 = \int_0^1 e^{1-x} dx = e - 1 \approx 1,71828.$$

$$I_n = n I_{n-1} - 1, n \geq 1.$$

Тогда

$$I_1 = 1 \cdot I_0 - 1 \approx 0.71828, \quad I_2 = 2I_1 - 1 \approx 0.43656, \quad I_3 = 3I_2 - 1 \approx 0.30968,$$

$$I_4 = 4I_3 - 1 \approx 0.23872, \quad I_5 = 5I_4 - 1 \approx 0.1936, \quad I_6 = 6 \cdot I_5 - 1 \approx 0.16160,$$

$$I_7 = 7I_6 - 1 \approx 0.13120, \quad I_8 = 8I_7 - 1 \approx 0.00496, \quad I_9 = 9I_8 - 1 \approx -0.55360,$$

$$I_{10} = 10 \cdot I_9 - 1 \approx -6.5360$$

Очевидно, что отрицательные значения при $n = 9, 10$ не имеют смысла. Дело в том, что ошибка, сделанная при округлении I_0 до 6-ти значащих цифр сохранилась при вычислении I_1 , умножилась на 2! при вычислении I_2 , на 3! — при вычислении I_3 , и так далее, т.е. ошибка растет очень быстро, пропорционально $n!$.

Пример 1.10. Рассмотрим методический пример вычислений на модельном компьютере, обеспечивающем точность $\delta_M = 0,0005$. Проанализируем причину происхождения ошибки, например, при вычитании двух чисел, взятых с точностью до третьей цифры после десятичной точки $u = 1,001$, $v = 1,002$, разность которых составляет $\Delta = |v_M - u_M| = 0,001$.

В памяти машины эти же числа представляются в виде

$$u_M = u(1 + \delta_M^u), v_M = v(1 + \delta_M^v), \text{ причем } |\delta_M^u| \leq \delta_M \text{ и } |\delta_M^v| \leq \delta_M.$$

Тогда $u_M - u \approx u\delta_M^u, v_M - v \approx v\delta_M^v$.

Относительная ошибка при вычислении разности $u_M - v_M$ будет равна

$$\delta = \frac{(u_M - v_M) - (u - v)}{(u - v)} = \frac{(u_M - u) - (v_M - v)}{(u - v)} = \frac{\delta_M^u - \delta_M^v}{(u - v)}.$$

Очевидно, что

$$\delta = \left| \frac{\delta_M^u - \delta_M^v}{\Delta} \right| \leq \frac{2\delta_M}{0,001} \approx 2000\delta_M = 1,$$

т.е. все значащие цифры могут оказаться неверными.

Пример 1.11. Рассмотрим рекуррентное соотношение $u_{i+1} = qu_i, i \geq 0, u_0 = a, q > 0, u_i > 0$.

Пусть при выполнении реальных вычислений с конечной длиной мантиссы на i -м шаге возникла погрешность округления, и вычисления проводятся с возмущенным значением

$$u_i^M = u_i + \delta_i,$$

тогда вместо u_{i+1} получим

$$u_{i+1}^M = q(u_i + \delta_i) = u_{i+1} + q\delta_i, \text{ т.е. } \delta_{i+1} = q\delta_i, \quad i = 0, 1, \dots$$

Следовательно, если $|q| > 1$, то в процессе вычислений погрешность, связанная с возникшей ошибкой округления, будет возрастать (алгоритм неустойчив). В случае $|q| \leq 1$ погрешность не возрастает и численный алгоритм устойчив.

Экономичность вычислительного метода

Пример 1.12. Пусть требуется вычислить сумму $S = 1 + x + x^2 + \dots + x^{1023}$ при $0 < x < 1$. Для последовательного вычисления

$$x, x^2 = x \cdot x, \dots, x^{1023} = x^{1022} \cdot x$$

необходимо проделать 1022 умножения, а затем столько же сложений.

Однако если заметить, что

$$S = \frac{1-x^{1024}}{1-x},$$

то количество арифметических действий значительно уменьшается; в частности, для вычисления x^{1024} требуется всего 10 умножений:

$$x^2 = x \cdot x; x^4 = (x)^2(x)^2, \dots, x^{1024} = (x)^{512}(x)^{512}.$$

Пример 1.13. Вычисления значений многочленов. Если вычислять значение многочлена $P(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ "в лоб", т.е. вычислять значения каждого члена и суммировать, то окажется, что необходимо выполнить $(n^2 + [n/2])$ умножений и n сложений. Кроме того, такой способ вычислений может привести к накоплению ошибок округления при вычислениях с плавающей точкой.

Его очевидным улучшением является вычисление каждого члена последовательным умножением на x . Такой *алгоритм* требует $(2n - 1)$ умножение и n сложений.

Еще более экономичным *алгоритмом* является хорошо известная в алгебре схема Горнера:

$$P(x) = (((...(a_nx + a_{n-1})x + a_{n-2})x + \dots + a_0),$$

требующая n операций сложения и n операций умножения. Этот метод был известен в средние века в Китае под названием Тянь-Юань и был заново открыт в Европе в начале XIX века англичанином Горнером и итальянцем Руффини.

Пример 1.14. Рассмотрим систему линейных алгебраических уравнений (СЛАУ) вида $\mathbf{A}\mathbf{u} = \mathbf{f}$, $\mathbf{u} = \{u_1, \dots, u_n\}^T$, $\mathbf{f} = \{f_1, \dots, f_n\}^T$, с трехдиагональной матрицей

$$\mathbf{A} = \begin{pmatrix} b_1 & c_1 & 0 & \dots & 0 \\ a_2 & b_2 & c_2 & \dots & 0 \\ 0 & a_3 & b_3 & c_3 & \dots & 0 \\ 0 & \dots & 0 & a_n & b_n \end{pmatrix}.$$

Если проводить решение такой системы без учета специфической структуры матрицы (например, с помощью метода Гаусса), то количество арифметических действий будет порядка n^3 , если же учесть эту структуру, то количество операций можно уменьшить до n .

Погрешность метода

Оценим *погрешность при вычислении* первой производной при помощи соотношения:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h};$$

$$\frac{f(x+h) - f(x)}{h} = \frac{[f(x) + hf'(x) + O(h^2)] - f(x)}{h} = f'(x) + O(h),$$

где $O(h)$ есть погрешность метода. В данном случае под погрешностью метода понимается абсолютная величина разности

$$\left| f'(x) - \frac{f(x+h) - f(x)}{h} \right|,$$

которая составляет величину

$$\frac{h}{2} f''(\xi),$$

где

$$\xi \in [x, x+h].$$

Если же взять другой метод вычисления производной

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h},$$

то получим, что его погрешность составляет $O(h^2)$, это оказывается существенным при малых h . Однако уменьшать h до бесконечности не имеет смысла, что видно из следующего примера. Реальная погрешность при вычислении первой производной будет

$$\Delta = \frac{h}{2} \max_{\xi \in [x, x+h]} |f''(\xi)| + \frac{2\delta_M}{h} = O(h) + O(h^{-1}),$$

поскольку абсолютная погрешность вычисления значения функции за счет машинного округления не превосходит

$$\frac{2\delta_M}{h}.$$

В этом случае можно найти оптимальный шаг h . Будем считать полную погрешность в вычислении производной Δ функцией шага h . Отыщем минимум этой функции. Приравняв производную $\Delta'_h(h)$ к нулю, получим оптимальный шаг численного дифференцирования

$$h_{\text{опт}} = 2 \sqrt{\frac{\delta_M}{\max_{\xi \in [x, x+h]} |f''(\xi)|}}.$$

Выбирать значение h меньше оптимального не имеет смысла, так как при дальнейшем уменьшении шага суммарная погрешность начинает расти из-за возрастания вклада ошибок округления.

Определение. Пусть u и u^* — точное и приближенное значение некоторой величины, соответственно. Тогда **абсолютной погрешностью** приближения u^* называется величина $\Delta(u^*)$, удовлетворяющая неравенству

$$|u - u^*| \leq \Delta(u^*).$$

Определение. **Относительной погрешностью** называется величина $\delta(u^*)$, удовлетворяющая неравенству

$$\left| \frac{u - u^*}{u^*} \right| \leq \delta(u^*).$$

Обычно используется запись

$$u = u^*(1 \pm \delta(u^*)).$$

Определение. Пусть искомая величина u является функцией параметров $t_1, \dots, t_n \in \Omega$, u^* — приближенное значение u . Тогда предельной абсолютной погрешностью называется величина

$$D(u^*) = \sup_{(t_1, \dots, t_n) \in \Omega} |u(t_1, \dots, t_n) - u^*|,$$

Предельной относительной погрешностью называется величина $D(u^*)/|u^*|$.

Пусть

$$|t_j - t_j^*| \leq \Delta(t_j^*), j = 1 \div n$$

– приближенное значение

$$u^* = u(t_1^*, \dots, t_n^*).$$

Предполагаем, что u - непрерывно дифференцируемая функция своих аргументов. Тогда, по формуле Лагранжа,

$$u(t_1, \dots, t_n) - u^* = \sum_{j=1}^n \gamma_j(\alpha)(t_j - t_j^*),$$

где

$$\gamma_j(\alpha) = u'_{t_j}(t_1^* + \alpha(t_1 - t_1^*), \dots, t_n^* + \alpha(t_n - t_n^*)), 0 \leq \alpha \leq 1.$$

Отсюда

$$|u(t_1, \dots, t_n) - u^*| \leq D_1(u^*) = \sum_{j=1}^n b_j \Delta(t_j^*),$$

где

$$b_j = \sup_{\Omega} |u'_{t_j}(t_1, \dots, t_n)|.$$

При малых значениях величины:

$$\rho = \sqrt{(\Delta(t_1^*))^2 + \dots + (\Delta(t_n^*))^2}$$

эта оценка не может быть существенно улучшена. На практике иногда пользуются грубой (линейной) оценкой

$$|u(t_1, \dots, t_n) - u^*| \leq D_2(u^*), \text{ где } D_2(u^*) = \sum_{j=1}^n |\gamma_j(0)| \Delta(t_j^*).$$

Несложно показать, что

а) $\Delta(\pm t_1^* \pm \dots \pm t_n^*) = \Delta(t_1^*) + \dots + \Delta(t_n^*)$, предельная погрешность суммы или разности равна сумме предельных погрешностей.

б) Предельная относительная погрешность произведения или частного приближенного равна сумме предельных относительных погрешностей

$$\delta(t_1^* \cdot \dots \cdot t_m^* \cdot d_1^{*-1} \cdot \dots \cdot d_m^{*-1}) = \delta(t_1^*) + \dots + \delta(t_m^*) + \delta(d_1^*) + \dots + \delta(d_m^*).$$

Задача численного дифференцирования

Пусть задана таблица значений x_j . В дальнейшем совокупность точек на отрезке, на котором проводятся вычисления, иногда будут называться **сеткой**, каждое значение x_j — **узлом сетки**. Пусть сетка — равномерная, и расстояние между узлами равно h — **шагу сетки**. Пусть узлы сетки пронумерованы в порядке возрастания, т.е.

$$\begin{aligned} x_0 &= a, \\ x_j &= a + jh, j = 0, 1, \dots, N \end{aligned}$$

Пусть $f(x_j) = f_j$ — функция, определенная в узлах сетки. Такие функции будут называться табличными, или сеточными функциями. Считаем, кроме того, что рассматриваемая сеточная функция есть проекция (или ограничение) на сетку некоторой гладкой нужное число раз непрерывно дифференцируемой функции $f(x)$. По определению производной

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x+\Delta x) - f(x)}{\Delta x},$$

тогда, если шаг сетки достаточно мал, по аналогии можно написать формулу в конечных разностях, дающую приближенное значение производной сеточной функции:

$$f'(x_j) \approx \frac{f(x_j+h) - f(x_j)}{h}.$$

Если параметр h достаточно мал, то можно считать полученное значение производной достаточно точным. При уменьшении шага сетки h ошибка будет уменьшаться, но при некотором значении h ошибка может возрасти до бесконечности. При оценке погрешности метода обычно считается, что все вычисления были точными. Но существует ошибка округления. При оценке ее большую роль играет машинный ε — мера относительной погрешности машинного округления, возникающей из-за конечной разрядности мантиссы при работе с числами в формате с плавающей точкой. Напомним, что по определению машинным ε называют наибольшее из чисел, для которых в рамках используемой системы вычислений выполнено $1 + \varepsilon = 1$. Тогда абсолютная погрешность при вычислении значения функции определяется как

$$f(x_j) \cdot \varepsilon.$$

Пусть $k = \max |f(x)|$, максимум ищется на отрезке, на котором вычисляются значения производных. Тогда суммарная ошибка, состоящая из погрешности метода и погрешности округления, есть

$$\Delta = \frac{2k\varepsilon}{h} + \frac{M_2 h}{2}, \quad M_2 = \max |f''(x)|.$$

Для вычисления оптимального шага *численного дифференцирования* найдем минимум суммарной ошибки, как функции шага сетки

$$\frac{M_2}{2} - \frac{2k\varepsilon}{h^2} = 0,$$

откуда

$$h_{opt} = 2\sqrt{\frac{k\varepsilon}{M_2}}.$$

Если требуется повысить точность вычисления производных, необходимо воспользоваться формулами, имеющими меньшие погрешности метода. Так, из курсов математического анализа известно, что

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x}.$$

По аналогии напишем конечно-разностную формулу

$$f'(x_j) \approx \frac{f(x_j+h) - f(x_j-h)}{2h},$$

и оценим погрешность метода такой аппроксимации. Предположим, что функция, которую спроектировали на сетку, трижды непрерывно дифференцируема, тогда

$$\begin{aligned} f(x_j+h) &= f(x_j) + f'(x_j)h + f''(x_j)\frac{h^2}{2} + f'''(x_j + \theta_1 h)\frac{h^3}{6}, \\ f(x_j-h) &= f(x_j) - f'(x_j)h + f''(x_j)\frac{h^2}{2} - f'''(x_j - \theta_2 h)\frac{h^3}{6}. \end{aligned}$$

Погрешность метода определяется 3-й производной функции.

Пусть

$$M_3 = \max_{x \in [a, b]} |f'''(x)|,$$

тогда суммарная *погрешность при вычислении* по формуле с центральной разностью есть

$$\Delta = \frac{M_3 h^2}{6} + \frac{k\varepsilon}{h},$$

для вычисления оптимального шага, находя минимум погрешности, как функции шага сетки, имеем

$$\frac{2M_3 h}{6} - \frac{k\varepsilon}{h^2} = 0,$$

откуда

$$h_{opt} = \sqrt[3]{\frac{3k\varepsilon}{M_3}}.$$

Для более точного вычисления производной необходимо использовать разложение более высокого порядка, шаг h_{opt} будет увеличиваться.

При вычислении производной используются точки (узлы) x_j (узел входит с нулевым коэффициентом),

$$x_j + h,$$

$$x_j - h$$

– совокупность узлов, участвующих в каждом вычислении производной, в дальнейшем будем иногда называть **сеточным шаблоном**.

Введем на рассматриваемом отрезке шаблон из нескольких точек.

Считаем, что сетка равномерная — шаг сетки постоянный, расстояния между любыми двумя соседними узлами равны. Используем для вычисления значения первой производной следующую приближенную (конечно-разностную) формулу:

$$f'(x_j) \approx \frac{1}{h} \sum_{k=-l}^m \alpha_k f(x_j + kh),$$

шаблон включает l точек слева от рассматриваемой точки x_j и m справа.

Коэффициенты α_k — **неопределенные коэффициенты**.

Формула дифференцирования может быть и **односторонней** — либо l , либо m могут равняться нулю. В первом случае иногда называют (на наш взгляд, не слишком удачно) такую приближенную формулу формулой дифференцирования вперед, во втором — формулой дифференцирования назад. Потребуем, чтобы (1.3) приближала первую производную с точностью $O(h^{l+m})$. Используя разложения в ряд Тейлора в окрестности точки x_j , получим

$$\begin{aligned} \frac{1}{h} \sum_{k=-l}^m \alpha_k f(x_j + kh) &= \frac{1}{h} f(x_j) \sum \alpha_k + f'(x_j) \sum k \alpha_k + f''(x_j) \sum \frac{k^2}{2} \alpha_k h + \\ &+ f'''(x_j) \sum \frac{k^3}{6} \alpha_k h^2 + \dots + f^{(n)}(x_j) \sum \alpha_k \frac{k^n}{n!} h^{n-1} + \dots \end{aligned}$$

Потребуем выполнение дополнительных условий:

$$\sum \alpha_k = 0, \sum k \alpha_k = 1, \sum \alpha_k \frac{k^2}{2} = 0, \dots, \sum \alpha_k \frac{k^n}{n!} = 0, \dots$$

Получаем систему линейных алгебраических уравнений для неопределенных коэффициентов α . Матрица этой системы есть

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ -l & -l+1 & \dots & m \\ l^2 & (l-1)^2 & \dots & m^2 \\ (-l)^3 & (-l+1)^3 & \dots & m^3 \\ \dots & \dots & \dots & \dots \end{pmatrix}.$$

Вектор правых частей $(0, 1, 0, \dots, 0)^T$.

Определитель данной матрицы — детерминант Вандермонда. Из курса линейной алгебры следует, что он не равен нулю. Тогда существует единственный набор коэффициентов α , который позволяет найти на шаблоне из $(1 + 1 + m)$ точек значение первой производной с точностью $O(h^{1+m})$.

Для нахождения второй производной можно использовать формулу:

$$f'(x_j) \approx \frac{1}{h^2} \sum_{k=-l}^m \alpha_k f(x_j + kh),$$

только теперь

$$\sum k\alpha_k = 0, \sum \alpha_k k^2 = \frac{1}{2}.$$

Очевидно, что и данная система уравнений для нахождения неопределенных коэффициентов имеет единственное решение.

Таким образом, доказано следующее утверждение. На сеточном шаблоне, включающем в себя $N + 1$ точку, с помощью *метода неопределенных коэффициентов* всегда можно построить единственную формулу для вычисления производной от первого до n порядка включительно с точностью $O(h^N)$.

Утверждение доказано для равномерной сетки, но на случай произвольных расстояний между сеточными узлами обобщение проводится легко.

Так как на практике вычисления проводятся с конечной длиной мантииссы, то получить нулевую ошибку невозможно.

Примеры

1. Найти абсолютную предельную погрешность, погрешность по производной, линейную погрешность для функции $u = t^{10}$, если заданы точка приближения $t^* = 1$, значение функции u^* в этой точке и погрешность $\Delta t^* = 10^{-1}$.

Решение : обозначим

$$\begin{aligned} b &= \sup_{|t-1| \leq 0.1} |u'_t(t)| = \sup_{|t-1| \leq 0.1} 10 \cdot t^9 = 10 \cdot (1,1)^9 \approx \\ &\approx 23, \dots \end{aligned}$$

Абсолютная предельная погрешность может быть определена как

$$D(u^*) = \sup_{|t-1| \leq 0,1} |t^{10} - 1| = (1.1)^{10} - 1 \approx 1,5 \dots$$

Оценка *погрешности* u при вычислении значения функции по максимуму производной и линейная оценка соответственно будут $D_1(u^*) = b\Delta(t^*) = 2,3 \dots$; $D_2(u^*) = |\gamma(0)|\Delta(t^*) = 1$.

2. Дать линейную оценку *погрешности* при вычислении неявной функции $\varphi(u, t_1, t_2, \dots, t_n) = 0$, если известны точка приближения $\{t_1^*, \dots, t_n^*\}$, значение функции в точке приближения u^* и погрешность в определении аргументов $\Delta t_1^*, \dots, \Delta t_n^*$.

Решение. Дифференцируя по t_j , получим

$$\frac{\partial \varphi}{\partial u} \frac{\partial u}{\partial t_j} + \frac{\partial \varphi}{\partial t_j} = 0,$$

откуда

$$\frac{\partial u}{\partial t_j} = -\left(\frac{\partial \varphi}{\partial t_j}\right)\left(\frac{\partial \varphi}{\partial u}\right)^{-1}.$$

При заданных $\{t_1^*, \dots, t_n^*\}$, можно найти u^* как корень уравнения $\varphi(u, t_1, t_2, \dots, t_n) = 0$, а затем — значения

$$b_j(0) = -\left(\frac{\partial \varphi}{\partial t_j}\right)\left(\frac{\partial \varphi}{\partial u}\right)^{-1} \Big|_{(u^*, t_1^*, \dots, t_n^*)},$$

откуда можно получить линейную оценку погрешности функции $D_2(u^*)$.

3. Оценить погрешность в определении корней квадратного уравнения

$$\varphi(u, t_1, t_2) = u^2 + t_1 u + t_2 = 0,$$

если заданы приближения

$$t_1^*, t_2^*, \Delta(t_1^*), \Delta(t_2^*).$$

Пусть u^* — решение уравнения

$$u^{*2} + t_1^* u^* + t_2^* = 0.$$

Из формулы

$$b_j(0) = -\left(\frac{d\varphi}{dt_j}\right)\left(\frac{d\varphi}{du}\right)^{-1} \Big|_{(u^*, t_1^*, \dots, t_n^*)}$$

получим

$$b_1(0) = \left. \frac{du}{dt_1} \right|_{(t_1^*, t_2^*)} = -\frac{u^*}{2u^* + t_1^*},$$

$$b_2(0) = \left. \frac{du}{dt_2} \right|_{(t_1^*, t_2^*)} = -\frac{1}{2u^* + t_1^*}.$$

Следовательно, линейная оценка будет

$$D_2(u^*) = \frac{|u^*| \cdot \Delta(t_1^*) + \Delta(t_2^*)}{|2u^* + t_1^*|}.$$

Задачи для самостоятельного решения

1. Найти абсолютную предельную погрешность, погрешность по производной и линейную оценку погрешности для функций $u = \sin t$,

$$u = \frac{1}{t^2 - 5t + 6}.$$

Заданы точка приближения $t = t^*$ и погрешность Δt .

2. Определить шаг τ , при котором погрешность вычисления производной $u'(t)$, приближенно вычисляемой в соответствии с формулами

$$u'(t) \approx \frac{f(t+\tau) - f(t)}{\tau},$$
$$u'(t) \approx \frac{f(t+\tau) - f(t-\tau)}{2\tau},$$

не превосходит 10^3 . Известно, что $|u''(t)| \leq 1$, $|u'''(t)| \leq 1$ для любых t .

3. Пусть для вычисления функции $u = f(t)$ используется частичная сумма ряда Маклорена

$$u(t) \approx u(0) + \frac{u'(0)}{1!}t + \dots + \frac{u^{(n)}(0)}{n!}t^n,$$

причем аргумент задан с погрешностью $\Delta t = 10^{-3}$.

Найти n такое, чтобы погрешность в определении функции $u(y)$ по данной формуле не превышала Δt . Рассмотреть отрезки $t \in [0, 1]$, $t \in [10, 11]$.

Предложить более совершенный алгоритм для вычисления функций $u(t) = \sin t$, $u(t) = e^t$ на отрезке $t \in [10, 11]$.

4. Определить оптимальный шаг численного дифференцирования τ при использовании для вычисления производной приближенной формулы

$$u'(t) \approx \frac{u(t-2\tau) - 8(t-\tau) + 8(t+\tau) - u(t+2\tau)}{12\tau},$$

имеющей четвертый порядок точности, если известно, что

$$|u^{(5)}(t)| \leq M_5,$$

а значения функций вычисляются с точностью ε .

1. ЭЛЕМЕНТЫ ТЕОРИИ ПОГРЕШНОСТЕЙ

Численное решение любой задачи, как правило, осуществляется приближенно, с различной точностью. Это может быть обусловлено неточностью исходных данных, конечной разрядностью вычислений (вручную или на ЭВМ) и т. п.

Главная задача численных методов – фактическое нахождение решения с требуемой или, по крайней мере, оцениваемой точностью.

Отклонение истинного решения от приближенного называется погрешностью.

Полная погрешность вычислений состоит из двух составляющих:

- 1) неустраняемая погрешность;
- 2) устранимая погрешность.

Неустраняемая погрешность обусловлена неточностью исходных данных и никаким образом не может быть уменьшена в процессе вычислений.

Устранимая погрешность состоит из двух составляющих:

- а) погрешность аппроксимации (метода);
- б) погрешность вычислений.

Эти составляющие могут быть уменьшены выбором более точных методов и увеличением разрядности вычислений.

Как правило, в дальнейшем нас будут интересовать корректно поставленные задачи вычисления.

Задача вычисления $y = A(x)$ называется корректно поставленной, если для любых входных данных из некоторого класса решение задачи существует, единственно и устойчиво по входным данным (т. е. непрерывно зависит от входных данных задачи).

В сформулированном понятии **корректности** учтены достаточно естественные требования, т. к. чтобы численно решать задачу, нужно быть уверенным, что ее решение **существует**. Столь же естественны требования **единственности** и **устойчивости** решения.

Рассмотрим подробнее понятие **устойчивости**. Обычно нас интересует решение y , соответствующее входным данным x . Реально мы имеем возмущенные входные данные с погрешностью δx , т. е. $x + \delta x$, и находим возмущенное решение:

$$y + \delta y = A(x + \delta x).$$

Эта погрешность входных данных порождает **неустраняемую** погрешность решения:

$$\delta y = A(x + \delta x) - A(x).$$

Если решение непрерывно зависит от входных данных, то

$$\|\delta y\| \rightarrow 0$$

всегда при

$$\|\delta x\| \rightarrow 0,$$

и задача устойчива по входным данным.

Здесь символ $\| \|$ – норма.

Если X – точное значение величины, а X^* – приближенное значение, то **абсолютная погрешность** приближения определяется выражением

$$\Delta = |X^* - X|.$$

Относительной погрешностью величины называется отношение абсолютной погрешности к модулю ее точного значения: $\delta = \Delta / |X|$.

Достаточно часто точное значение величины неизвестно, поэтому указывают границы погрешности:

$$X^* - \Delta \leq X \leq X^* + \Delta; \quad (1.1)$$

$$X^* (1 - \delta) \leq X \leq X^* (1 + \delta). \quad (1.2)$$

Рассмотрим подробнее погрешность округления чисел, участвующих в вычислениях. В позиционной системе счисления с основанием r запись

$$a = \pm a_n a_{n-1} \dots a_0, a_{-1} a_{-2} \dots \quad (1.3)$$

означает, что

$$a = \pm (a_n r^n + a_{n-1} r^{n-1} + \dots + a_0 r^0 + a_{-1} r^{-1} + a_{-2} r^{-2} + \dots).$$

Здесь r – целое число, большее единицы. Каждое из чисел a_i может принимать одно из значений $\{0, 1, \dots, r-1\}$. Числа a_i называются *разрядами*, например: a_3 – третий разряд перед запятой, a_{-2} – второй разряд после запятой.

Запись вещественного числа в виде (1.3) называется также его представлением в форме *числа с фиксированной запятой*. В ЭВМ чаще всего используется представление *чисел в форме с плавающей запятой*. Так как наиболее часто в компьютерах применяется двоичная система с плавающей запятой, то вещественное число можно представить в виде

$$a = \pm 2^p \sum_{k=1}^t \alpha_k 2^{-k} = \pm 2^p (\alpha_1 \dots \alpha_t), \quad (1.4)$$

где

$$(|p| \leq p_0), \quad \alpha_1 = 1.$$

Здесь p – целое число называется порядком числа a , а $(\alpha_1 \dots \alpha_t)$ – мантиссой.

Если исходная величина или промежуточный результат требуют большего числа разрядов, то производится округление до t – го разряда. Значащие цифры называются верными до t – го разряда, если абсолютная величина разности между a^* и a меньше или равна половине единицы младшего разряда:

$$|a^* - a| \leq \frac{\alpha^{p-t}}{2}. \quad (1.5)$$

Ограничения на порядки чисел, представляемых в ЭВМ $|p| \leq p_0$, порой приводят к прекращению вычислений (так называемое исчезновение порядка); в других случаях относительно небольшая разрядность представления чисел в ЭВМ приводит к недопустимому искажению результата вычислительной погрешностью.

Приведем несколько примеров иллюстрирующих это и способы (приемы) уменьшения вычислительной погрешности за счет несложных алгебраических преобразований.

Рассмотрим типичный пример, в котором порядок выполнения операций существенно влияет на погрешность результата.

Пример 1.1. Необходимо отыскать минимальный корень уравнения. Вычисления производим в десятичной системе счисления, причем в числе после округления оставляем четыре действующие цифры (разряда):

$$\begin{aligned}x^2 - 140x + 1 &= 0; \\x &= 70 - \sqrt{4899}; \\ \sqrt{4899} &= 69.992 \approx 69.99; \\x &= 70 - 69.99 = 0.01.\end{aligned}$$

Рассмотрим другой алгоритм вычисления корня, для чего избавимся от иррациональности в числителе:

$$\begin{aligned}x &= 1/(70 + \sqrt{4899}); \\70 + 69.99 &\approx 140.0; \\x &= 1/140 = 0.00714285 \approx 0.007143.\end{aligned}$$

Как видно из сравнения полученных результатов, применение "неудачного" алгоритма завышает результат на 30 %. Это явление в прикладной математике (в практике вычислений) называется *потерей значащих цифр*, и часто наблюдается при вычитании близких величин. Потеря значащих цифр, например, довольно часто приводит к существенному искажению результатов при решении даже сравнительно небольших систем линейных алгебраических уравнений.

Пример 1.2. На машине с плавающей запятой необходимо вычислить значение суммы

$$S_{1000000} = \sum_{i=1}^{1000000} \frac{1}{i^2}.$$

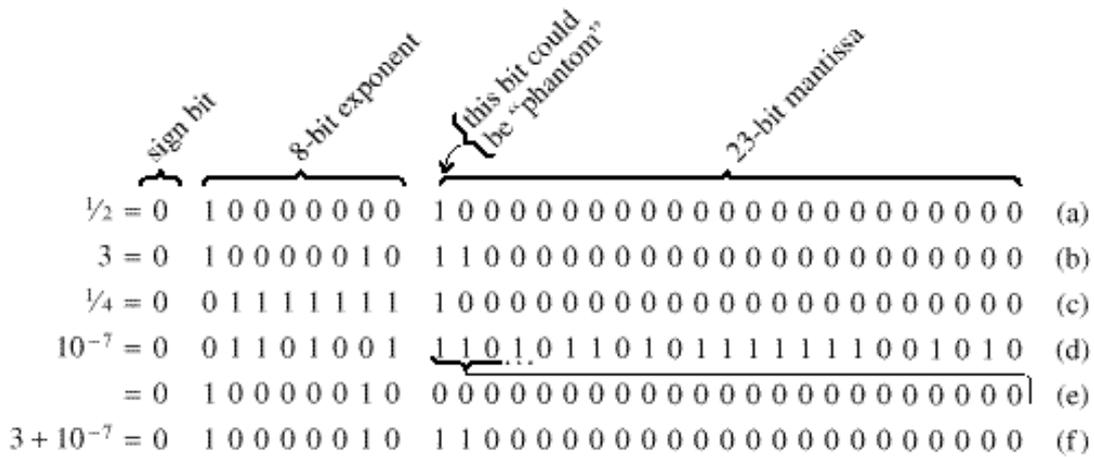
Эту сумму можно вычислить двумя способами:

$$\begin{aligned}1. \quad S_n &= S_{n-1} + \frac{1}{n^2}; & n &= 1, 2, \dots, 1000000; & S_0 &= 0; \\2. \quad \Sigma_{n-1} &= \Sigma_n + \frac{1}{n^2}; & n &= 1000000, \dots, 2, 1; & \Sigma_{1000000} &= 0.\end{aligned}$$

Оказывается, для второго алгоритма вычислительная погрешность будет существенно меньше.

Тестовые расчеты на конкретной ЭВМ по первому и второму алгоритмам показали, что величина погрешности для обоих алгоритмов составляет $2 \cdot 10^{-4}$ и $6 \cdot 10^{-8}$ соответственно. Причина этого ясна, если вспомним, как числа представлены в ЭВМ (см. рис. 1.1).

Рис. 1.1. Представление чисел с плавающей точкой в типичном



32-бит (4 - байт) формате:

а) Число $\frac{1}{2}$; **б)** число 3; **с)** число $\frac{1}{4}$; **д)** число 10^{-7} , представленное в машине с максимальной точностью (нормализованное); **е)** то же самое число 10^{-7} , но представленное с тем же порядком, что и число 3*; **ф)** сумма чисел $3 + 10^{-7}$, которая эквивалентна 3.

Этот пример ярко иллюстрирует тот факт, что даже если оба слагаемых представлены в компьютере точно, то их сумма может быть представлена с погрешностью, особенно если слагаемые различаются на порядки.

При оценке погрешностей арифметических действий следует учитывать следующее:

а) абсолютная погрешность алгебраической суммы

$W = \pm x_1 \pm x_2 \pm \dots \pm x_n$ не превышает суммы абсолютных погрешностей ее членов:

$$|\Delta W| \leq \sum_{i=1}^n |\Delta x_i|; \quad (1.6)$$

б) относительная погрешность произведения $W = x_1 \cdot x_2 \cdot \dots \cdot x_n$ не превышает суммы относительных погрешностей сомножителей:

$$|\Delta \delta| \leq \sum_{i=1}^n |\delta x_i|; \quad (1.7)$$

в) относительная погрешность частного $W = x_1 / x_2$ не превышает суммы относительных погрешностей делимого и делителя:

$$\delta W \leq \delta x_1 + \delta x_2. \quad (1.8)$$

Вопросы для самопроверки

* Из рисунка видно, что для такого представления числа разрядов не хватает, т.е. в машине такое число представляется нулем.

1. Дайте определения и приведите примеры устранимой и неустранимой погрешностей.
2. Что такое погрешность округления? Какова ее связь с разрядностью ЭВМ?
3. Как вычислить относительную погрешность, зная абсолютную?
4. Как по абсолютной погрешности вычислить относительную погрешность?

2. ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ

2.1. Постановка задачи

Задача численного интегрирования функции заключается в вычислении определенного интеграла на основании ряда значений подынтегральной функции. Численное вычисление однократного интеграла называется *механической квадратурой*.

Мы будем рассматривать способы приближенного вычисления определенных интегралов

$$J = \int_a^b f(x) dx, \quad (2.1)$$

основанные на замене интеграла конечной суммой:

$$I_n = \sum_{k=0}^n C_k f(x_k), \quad (2.2)$$

где C_k - числовые коэффициенты, а $x_k \in [a, b]$, $k = 0, 1, \dots, n$.

Приближенное равенство

$$\int_a^b f(x) dx \approx \sum_{k=0}^n C_k f(x_k) \quad (2.3)$$

называется квадратурной формулой, а x_k – узлами квадратурной формулы. Погрешность квадратурной формулы определяется соотношением

$$\Psi_n = \int_a^b f(x) dx - \sum_{k=0}^n C_k f(x_k). \quad (2.4)$$

В общем случае погрешность квадратурной формулы (2.4) зависит как от выбора коэффициентов C_k , так и от расположения узлов x_k . Введем на отрезке $[a, b]$ равномерную сетку с шагом h , тогда $x_i = a + ih$, где ($i = 0, 1, \dots, n$; $h \cdot n = b - a$). Теперь выражение (2.1) можно представить в виде суммы интегралов по частичным отрезкам:

$$J = \int_a^b f(x) dx = \sum_{i=1}^n \int_{x_{i-1}}^{x_i} f(x) dx. \quad (2.5)$$

Таким образом, для построения формулы численного интегрирования на отрезке $[a, b]$ достаточно построить квадратурную формулу на частичном отрезке $[x_{i-1}, x_i]$ и воспользоваться формулой (2.5).

2.2. Формула прямоугольников

На частичном отрезке $[x_{i-1}, x_i]$ заменим подынтегральную функцию полиномом Лагранжа нулевого порядка, построенным в одной точке. Естественно в качестве этой точки выбрать среднюю: $x_{i-0.5} = x_i - 0.5h$. Тогда получим формулу

$$\int_{x_{i-1}}^{x_i} f(x)dx \approx f(x_{i-0.5})h. \quad (2.6)$$

Подставив (2.6) в (2.5), получим составную формулу средних прямоугольников:

$$\int_a^b f(x)dx \approx \sum_{i=1}^n f(x_{i-0.5})h. \quad (2.7)$$

Графическая иллюстрация метода средних прямоугольников представлена на рис. 2.1.

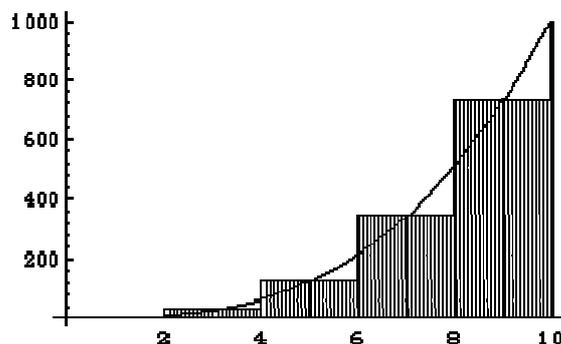


Рис. 2.1. Интегрирование методом средних прямоугольников

Погрешность формулы (2.7) определяется выражением

$$|\psi| \leq \frac{h^2(b-a)}{24} M_2. \quad (2.8)$$

Здесь

$$M_2 = \max_{x \in [a,b]} |f''(x)|.$$

Таким образом, погрешность формулы (2.7) пропорциональна $O(h^2)$.

Замечание. Формулу (2.7) можно представить в ином виде:

$$I \approx \sum_{i=1}^n hf(x_{i-1}); \quad I \approx \sum_{i=1}^n hf(x_i). \quad (2.9)$$

Эти формулы в выражении (2.9) называются формулой левых и правых прямоугольников соответственно. Графически метод левых и правых прямоугольников представлен на рис. 2.2.

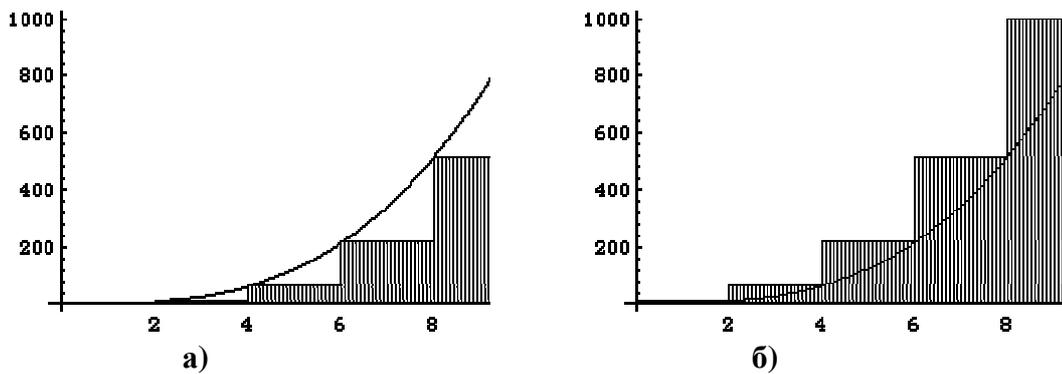


Рис. 2.2. Метод левых (а) и правых (б) прямоугольников

Однако из-за нарушения симметрии в формулах (2.9) их погрешность значительно больше, чем в методе средних прямоугольников и $\sim O(h)$.

2.3. Формула трапеций

Если на частичном отрезке подынтегральную функцию заменить полиномом Лагранжа первой степени, то есть

$$f(x) \approx L_{1,i}(x) = \frac{1}{h} [(x - x_{i-1})f(x_i) - (x - x_i)f(x_{i-1})], \quad (2.10)$$

тогда искомый интеграл запишется следующим образом:

$$\begin{aligned} \int_{x_{i-1}}^{x_i} f(x) dx &\approx \frac{1}{h} [f(x_i) \int_{x_{i-1}}^{x_i} (x - x_{i-1}) dx - \\ &- f(x_{i-1}) \int_{x_{i-1}}^{x_i} (x - x_i) dx] = \dots = \frac{f(x_{i-1}) + f(x_i)}{2} h. \end{aligned} \quad (2.11)$$

После подстановки выражения (2.11) в (2.5) составная формула трапеций примет вид

$$\begin{aligned} \int_a^b f(x) dx &\approx \sum_{i=1}^n \frac{f(x_i) + f(x_{i-1})}{2} h = \\ &= h \left[\frac{1}{2} (f_0 + f_n) + f_1 + \dots + f_{n-1} \right]. \end{aligned} \quad (2.12)$$

Графически метод трапеций представлен на рис. 2.3.

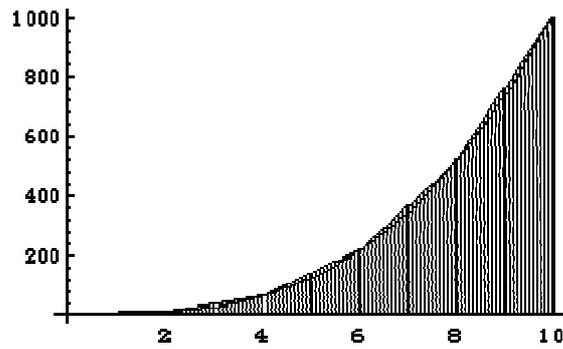


Рис. 2.3. Метод трапеций

Погрешность формулы (2.12) определяется выражением:

$$|\Psi| \leq \frac{h^2(b-a)}{12} M_2. \quad (2.13)$$

Таким образом, погрешность метода трапеций $\Psi \sim O(h^2)$, но она в два раза больше, чем для формулы средних прямоугольников.

2.4. Формула Симпсона

В этом методе предлагается подынтегральную функцию на частичном отрезке аппроксимировать параболой, проходящей через точки

$(x_j, f(x_j))$, то есть подынтегральную функцию аппроксимируем интерполяционным многочленом Лагранжа второй степени:

$$\begin{aligned} f(x) \approx L_{2,i}(x) &= \frac{2}{h^2} [(x - x_{i-1/2})(x - x_i) f(x_{i-1}) - \\ &- 2(x - x_{i-1})(x - x_i) f(x_{i-1/2}) + (x - x_{i-1})(x - x_{i-1/2}) f(x_i)]; \quad (2.14) \\ x &\in [x_{i-1}, x_i]. \end{aligned}$$

Проведя интегрирование, получим:

$$\begin{aligned} \int_{x_{i-1}}^{x_i} f(x) dx &\approx \int_{x_{i-1}}^{x_i} L_{2,i}(x) dx = \frac{h}{6} (f_{i-1} + 4f_{i-1/2} + f_i), \quad (2.15) \\ h &= x_i - x_{i-1}. \end{aligned}$$

Это и есть **формула Симпсона** или формула парабол. На отрезке $[a, b]$ формула Симпсона примет вид

$$\int_a^b f(x)dx \approx \frac{h}{6}[f_0 + f_n + 2(f_1 + f_2 + \dots + f_{n-1}) + 4(f_{1/2} + f_{3/2} + f_{5/2} + \dots + f_{n-1/2})]. \quad (2.16)$$

Графическое представление метода Симпсона показано на рис. 2.4.

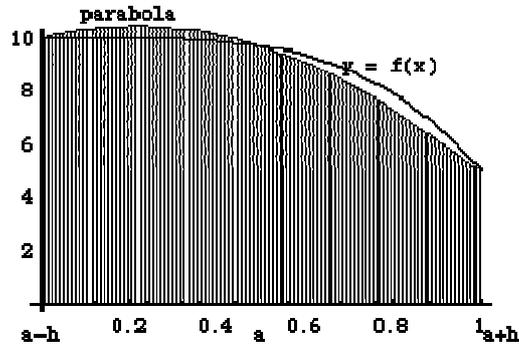


Рис. 2.4. Метод Симпсона

Избавимся в выражении (2.16) от дробных индексов, переобозначив переменные:

$$\begin{aligned} x_i &= a + 0.5h \cdot i; & f_i &= f(x_i); \\ i &= 0, 1, 2, \dots, 2n; & h \cdot n &= b - a. \end{aligned} \quad (2.17)$$

Тогда формула Симпсона примет вид

$$\int_a^b f(x)dx \approx \frac{b-a}{6n}[f_0 + f_{2n} + 2(f_2 + f_4 + \dots + f_{2n-2}) + 4(f_1 + f_3 + f_5 + \dots + f_{2n-1})]. \quad (2.18)$$

Погрешность формулы (2.18) оценивается следующим выражением:

$$|\Psi| \leq \frac{h^4(b-a)}{2880} M_4, \quad (2.19)$$

где $h \cdot n = b - a$,

$$M_4 = \sup_{x \in [a, b]} |f^{IV}(x)|.$$

Таким образом, погрешность формулы Симпсона пропорциональна $O(h^4)$.

Замечание. В формуле Симпсона отрезок интегрирования обязательно разбивается на *четное* число интервалов.

2.5. Метод Монте-Карло

Рассматриваемые ранее методы называются *детерминированными*, то есть лишенными элемента случайности.

Методы Монте-Карло (ММК) – это численные методы решения математических задач с помощью моделирования случайных величин. ММК позволяют успешно решать математические задачи, обусловленные вероятностными процессами. Более того, при решении задач, не связанных с какими-либо вероятностями, можно искусственно придумать вероятностную модель (и даже не одну), позволяющую решать эти задачи. Рассмотрим вычисление определенного интеграла

$$J = \int_a^b f(x)dx. \quad (2.20)$$

При вычислении этого интеграла по формуле прямоугольников интервал $[a, b]$ разбиваем на N одинаковых интервалов, в серединах которых вычислялись значения подынтегральной функции. Вычисляя значения функции в случайных узлах, можно получить более точный результат:

$$J = \int_a^b f(x)dx \approx \frac{b-a}{N} \sum_{i=1}^N f(x_i); \quad (2.21)$$

$$x_i = a + \gamma_i(b-a). \quad (2.22)$$

Здесь γ_i - случайное число, равномерно распределенное на интервале $[0, 1]$.

Погрешность вычисления интеграла ММК $\sim 1/\sqrt{N}$, что значительно больше, чем у ранее изученных детерминированных методов.

На рис. 2.5 представлена графическая реализация метода Монте-Карло вычисления однократного интеграла со случайными узлами (2.21) и (2.22).

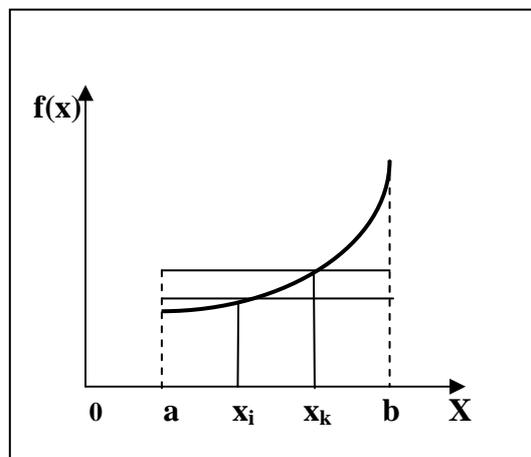


Рис. 2.5. Интегрирование методом Монте-Карло (1-й случай)

Однако при вычислении кратных интегралов детерминированными методами оценка погрешности перерастает в задачу порой более сложную, чем вычисление интеграла. В то же время погрешность вычисления кратных интегралов ММК слабо зависит от кратности и легко вычисляется в каждом конкретном случае практически без дополнительных затрат.

Рассмотрим еще один метод Монте-Карло на примере вычисления однократного интеграла:

$$J = \int_0^1 f(x)dx, \quad 0 \leq J \leq 1. \quad (2.23)$$

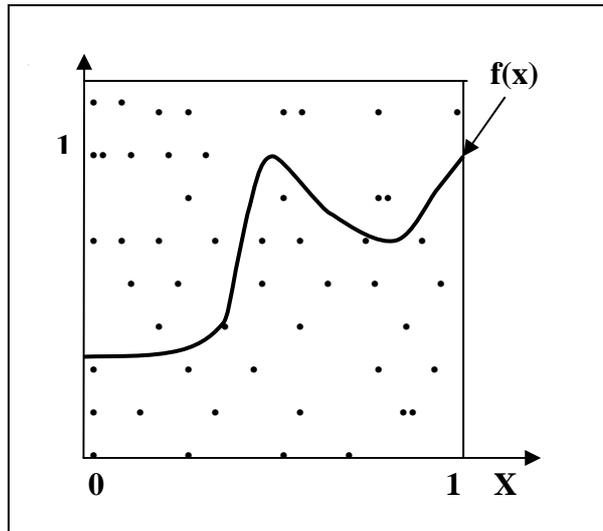


Рис. 2.6. Интегрирование методом Монте-Карло (2-й случай)

Как видно на рис. 2.6, интегральная кривая лежит в единичном квадрате, и если мы сумеем получать пары случайных чисел, равномерно распределенных на интервале $[0, 1]$, то полученные значения (γ_1, γ_2) можно интерпретировать как координаты точки в единичном квадрате. Тогда, если этих пар чисел получено достаточно много, можно приблизительно считать, что

$$J \approx S / N.$$

Здесь S – число пар точек, попавших под кривую, а N – общее число пар чисел.

Пример 2.1. Вычислить следующий интеграл:

$$J = \int_1^2 e^x dx = e^x \Big|_1^2 = e^2 - e^1 = 4.670774270$$

Поставленная задача была решена различными методами. Полученные результаты сведены в табл. 2.1.

Таблица 2.1

Число интервалов (точек)	Метод левых прямоугольников	Метод средних прямоугольников	Метод правых прямоугольников	Метод трапеций	Метод Симпсона	Метод Монте-Карло
10	4.44112722	4.66882868	4.90820465	4.25683746	4.67077443	4.62289422
100	4.64745932	4.67075481	4.69416706	4.62903035	4.67077427	4.69812790

Замечание. Выбор табличного интеграла позволил нам сравнить погрешность каждого метода и выяснить влияние числа разбиений на точность вычислений.

Вопросы для самопроверки

1. Сформулируйте задачу численного интегрирования.
2. Метод средних, левых и правых прямоугольников. Что можно сказать об их погрешности, трудоемкости?
3. Задача численного интегрирования решена методом трапеций. Предложите и обоснуйте пути повышения точности (уменьшения погрешности) расчетов.
4. Сравните метод трапеций и метод Симпсона.
5. Какие методы Монте–Карло численного интегрирования вы знаете? Сравните эти методы с любым детерминированным.
6. Необходимо вычислить интеграл методами трапеций, Симпсона и ММК, разбив область интегрирования на 77 интервалов (точек). Что можно сказать о точности и применимости этих методов?

3. ЧИСЛЕННОЕ РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ (СЛАУ)

3.1. Решение задач линейной алгебры

Линейные системы имеют в вычислениях очень большое значение, так как к ним может быть приведено приближенное решение широкого круга задач. Так, основными источниками возникновения СЛАУ являются теория электрических цепей, уравнения балансов и сохранения в механике, гидравлике и т.п.

Пусть дана система n линейных алгебраических уравнений с n неизвестными:

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

Или в матричной форме:

$$A \cdot x = b; \quad (3.2)$$

где

$$A = \{a_{ij}\} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} - \quad (3.3)$$

- матрица коэффициентов системы (3.1);

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \dots \\ \mathbf{x}_n \end{bmatrix}$$

- вектор неизвестных;

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \dots \\ \mathbf{b}_n \end{bmatrix}$$

- вектор свободных членов.

Если матрица A неособенная, т.е.

$$\det A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} = \Delta \neq 0, \quad (3.4)$$

то система (3.1) или эквивалентное ей матричное уравнение (3.2) имеют единственное решение. Действительно, при условии, что $\det A \neq 0$, существует обратная матрица A^{-1} . Умножая обе части уравнения (3.2) слева на A^{-1} , получим:

$$A^{-1} \cdot A \cdot \mathbf{x} = A^{-1} \cdot \mathbf{b}; \Rightarrow \mathbf{x} = A^{-1} \cdot \mathbf{b}. \quad (3.5)$$

Формула (3.5) даёт решение уравнения (3.2), причём единственное.

Пример 3.1.

$$\begin{cases} 3x_1 - x_2 = 5 \\ -2x_1 + x_2 + x_3 = 0; \\ 2x_1 - x_2 + 4x_3 = 15 \end{cases} \quad A = \begin{bmatrix} 3 & -1 & 0 \\ -2 & 1 & 1 \\ 2 & -1 & 4 \end{bmatrix}.$$

$$\Delta = \begin{vmatrix} 3 & -1 & 0 \\ -2 & 1 & 1 \\ 2 & -1 & 4 \end{vmatrix} = [12 + (-2) + 0] - [(0 \cdot 1 \cdot 2) + 8 - 3] = 5.$$

$$A^{-1} = \begin{bmatrix} 1 & 4/5 & -1/5 \\ 2 & 12/5 & -3/5 \\ 0 & 1/5 & 1/5 \end{bmatrix}; \quad x = \begin{bmatrix} 1 & 4/5 & -1/5 \\ 2 & 12/5 & -3/5 \\ 0 & 1/5 & 1/5 \end{bmatrix} \times \begin{bmatrix} 5 \\ 0 \\ 15 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}.$$

Для матрицы A порядка $n > 4$ непосредственное нахождение обратной матрицы A^{-1} требует много времени (операций). Поэтому формула (3.5) на практике употребляется достаточно редко.

Обычно значения неизвестных x_i ($i = 1, 2, \dots, n$) могут быть получены по известным формулам Крамера:

$$x_i = \det A_i / \det A. \quad (3.6)$$

Здесь матрица A_i получается из матрицы A заменой её i -го столбца столбцом свободных членов.

Пример 3.2. Решим вышеприведенную систему по формулам Крамера:

$$\Delta_1 = \begin{vmatrix} 5 & -1 & 0 \\ 0 & 1 & 1 \\ 15 & -1 & 4 \end{vmatrix} = 20 - 15 + 5 = 10; \quad x_1 = 2.$$

$$\Delta_2 = \begin{vmatrix} 3 & 5 & 0 \\ -2 & 0 & 1 \\ 2 & 15 & 4 \end{vmatrix} = 10 - 45 + 40 = 5; \quad x_2 = 1.$$

$$\Delta_3 = \begin{vmatrix} 3 & -1 & 5 \\ -2 & 1 & 0 \\ 2 & -1 & 15 \end{vmatrix} = 45 + 10 - 10 - 30 = 15; \quad x_3 = 3.$$

Применяемые в настоящее время методы решения СЛАУ можно разбить на две группы: *точные и приближённые*.

Точными методами называются такие методы, которые в предположении, что вычисления ведутся точно (без округлений), за конечное число действий позволяют получить точные значения неизвестных x_i .

Приближенными методами называются такие методы, которые даже в предположении, что вычисления ведутся без округлений, позволяют получить решение системы (x_1, x_2, \dots, x_n) лишь с заданной точностью. Точное решение СЛАУ в этих случаях может быть получено теоретически как результат бесконечного процесса.

К приближенным методам относятся метод простой итерации, метод Зейделя и т.п.

3.2. Метод Гаусса

Наиболее распространенным методом решения СЛАУ является метод Гаусса, в основе которого лежит идея последовательного исключения неизвестных. Существуют различные схемы, реализующие данный метод. Рассмотрим одну из них – *схему единственного деления*.

Для простоты ограничимся рассмотрением СЛАУ с четырьмя неизвестными:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 = a_{15}, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 = a_{25}, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 = a_{35}, \\ a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 = a_{45}. \end{cases} \quad (3.7)$$

Пусть $a_{11} \neq 0$ (ведущий элемент). Разделив первое уравнение на a_{11} , получим *первую главную строку*:

$$x_1 + b_{12}x_2 + b_{13}x_3 + b_{14}x_4 = b_{15}, \quad (3.8)$$

где

$$b_{1j} = a_{1j} / a_{11}; \quad (j = 2, 3, 4, 5).$$

Используя уравнение (3.8), можно исключить неизвестные x_1 из 2-го, 3-го и 4-го уравнений системы (3.7). Для этого последовательно умножаем уравнение (3.8) на a_{21} ; a_{31} ; a_{41} и вычитаем результат из 2-го, 3-го и 4-го уравнений системы (3.7) соответственно.

В результате получим систему из трех уравнений:

$$\begin{cases} a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + a_{24}^{(1)}x_4 = a_{25}^{(1)}, \\ a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 + a_{34}^{(1)}x_4 = a_{35}^{(1)}, \\ a_{42}^{(1)}x_2 + a_{43}^{(1)}x_3 + a_{44}^{(1)}x_4 = a_{45}^{(1)}, \end{cases} \quad (3.9)$$

где коэффициенты $a_{ij}^{(1)}$ вычисляются по формуле

$$a_{ij}^{(1)} = a_{ij} - a_{i1}b_{1j} \quad (i = 2, 3, 4; j = 2, 3, 4, 5). \quad (3.10)$$

Далее первое уравнение системы (3.9) делим на *ведущий элемент* $a_{22}^{(1)} \neq 0$ и получаем

$$x_2 + b_{23}^{(1)}x_3 + b_{24}^{(1)}x_4 = b_{25}^{(1)}, \quad (3.11)$$

где

$$b_{2j}^{(1)} = a_{2j}^{(1)} / a_{22}^{(1)}, \quad (j = 3, 4, 5).$$

Аналогично предыдущему шагу, исключая x_2 , как и x_1 , получим систему

$$\begin{cases} a_{33}^{(2)} x_3 + a_{34}^{(2)} x_4 = a_{35}^{(2)}, \\ a_{43}^{(2)} x_3 + a_{44}^{(2)} x_4 = a_{45}^{(2)}. \end{cases} \quad (3.12)$$

Здесь

$$a_{ij}^{(2)} = a_{ij}^{(1)} - a_{i2}^{(1)} b_{2j}^{(1)} \quad (i = 3, 4; j = 3, 4, 5).$$

Разделив первое уравнение системы (3.12) на $a_{33}^{(2)} \neq 0$, получим:

$$x_3 + b_{34}^{(2)} x_4 = b_{35}^{(2)}, \quad (3.13)$$

где

$$b_{3j}^{(2)} = a_{3j}^{(2)} / a_{33}^{(2)} \quad (j = 4, 5).$$

Теперь с помощью уравнения (3.13) исключим x_3 из второго уравнения системы (3.12), окончательно получим:

$$a_{44}^{(3)} x_4 = a_{45}^{(3)}, \quad (3.14)$$

где

$$a_{4j}^{(3)} = a_{4j}^{(2)} - a_{43}^{(2)} b_{3j}^{(2)} \quad (j=4, 5).$$

Таким образом, исходную систему (3.7) привели к составленной из *главных строк* (3.8), (3.11), (3.13) и (3.14) эквивалентной системе с треугольной матрицей(3.15):

$$\begin{cases} x_1 + b_{12} x_2 + b_{13} x_3 + b_{14} x_4 = b_{15}, \\ x_2 + b_{23}^{(1)} x_3 + b_{24}^{(1)} x_4 = b_{25}^{(1)}, \\ x_3 + b_{34}^{(2)} x_4 = b_{35}^{(2)}, \\ a_{44}^{(3)} x_4 = a_{45}^{(3)}. \end{cases} \quad (3.15)$$

Из (3.15) последовательно находим

$$\begin{cases} x_4 = a_{45}^{(3)} / a_{44}^{(3)}, \\ x_3 = b_{35}^{(2)} - b_{34}^{(2)} x_4, \\ x_2 = b_{25}^{(1)} - b_{23}^{(1)} x_3 - b_{24}^{(1)} x_4, \\ x_1 = b_{15} - b_{12} x_2 - b_{13} x_3 - b_{14} x_4. \end{cases} \quad (3.16)$$

Итак, решение СЛАУ (3.7) распадается на два этапа:

- прямой ход (приведение системы (3.7) к треугольному виду (3.15));
- обратный ход (определение неизвестных по формуле (3.16)).

Пример 3.3.

$$\begin{cases} 2.0x_1 + 1.0x_2 - 1.0x_3 + 1.0x_4 = 2.7, \\ 0.4x_1 + 0.5x_2 + 4.0x_3 - 8.5x_4 = 21.9, \\ 0.3x_1 - 1.0x_2 + 1.0x_3 + 5.2x_4 = -3.9, \\ 1.0x_1 + 0.2x_2 + 2.5x_3 - 1.0x_4 = 9.9. \end{cases}$$

Прямой ход:

$$x_1 + 0.5x_2 - 0.05x_3 + 0.5x_4 = 1.35;$$

$$b_{12} = 0.5; \quad b_{13} = 0.05; \quad b_{14} = 0.5; \quad b_{15} = 1.35.$$

Из выражений (3.10) вычислим коэффициенты $a_{2j}^{(1)}$:

$$a_{22}^{(1)} = a_{22} - a_{21}b_{12} = 0.5 - 0.4 \cdot 0.5 = 0.3;$$

$$a_{23}^{(1)} = a_{23} - a_{21}b_{13} = 4 + 0.4 \cdot 0.05 = 4.02;$$

$$a_{24}^{(1)} = a_{24} - a_{21}b_{14} = -8.5 - 0.4 \cdot 0.5 = -8.7;$$

$$a_{25}^{(1)} = a_{25} - a_{21}b_{15} = 21.9 - 0.4 \cdot 1.35 = 21.36.$$

Аналогично вычислим коэффициенты $a_{ij}^{(1)}$ при $(i = 3, 4)$ и составим систему

$$\begin{cases} 0.3x_2 + 4.02x_3 - 8.7x_4 = 21.36, \\ -1.15x_2 + 1.015x_3 + 5.05x_4 = -4.305, \\ -0.3x_2 + 2.55x_3 - 1.5x_4 = 8.55. \end{cases}$$

Разделив первое уравнение системы на $a_{22}^{(1)} = 0.3$, получим

$$x_2 + 13.40x_3 - 29.00x_4 = 71.20.$$

Значит,

$$b_{23}^{(1)} = 13.40; \quad b_{24}^{(1)} = -29.00; \quad b_{25}^{(1)} = 71.20.$$

Из (3.12) вычислим $a_{ij}^{(2)}$ для $i = 3$ и $j = 3, 4, 5$:

$$a_{33}^{(2)} = a_{33}^{(1)} - a_{32}^{(1)}b_{23}^{(1)} = 1.015 + 1.15 \cdot 13.40 = 16.425;$$

$$a_{34}^{(2)} = a_{34}^{(1)} - a_{32}^{(1)}b_{24}^{(1)} = 5.05 - 1.15 \cdot 29.00 = -28.3;$$

$$a_{35}^{(2)} = a_{35}^{(1)} - a_{32}^{(1)}b_{25}^{(1)} = -4.305 + 1.15 \cdot 71.20 = 77.575.$$

Аналогично, вычислив коэффициенты для $i = 4$, получим:

$$\begin{cases} 16.425x_3 - 28.3x_4 = 77.575, \\ 6.570x_3 - 10.200x_4 = 29.910. \end{cases}$$

Разделив первое уравнение на $a_{33}^{(2)} = 16.425$, получим:

$$x_3 - 1.72298x_4 = 4.72298,$$

где

$$b_{34}^{(2)} = -1.72298; \quad b_{35}^{(2)} = 4.72298.$$

По формуле (3.14) находим коэффициенты $a_{ij}^{(3)}$:

$$a_{44}^{(3)} = a_{44}^{(2)} - a_{43}^{(2)}b_{34}^{(2)} = -10.2 + 6,57 * 1.72298 = 1.1199786;$$

$$a_{45}^{(3)} = a_{45}^{(2)} - a_{43}^{(2)}b_{35}^{(2)} = 29.910 - 6,57 * 4.72298 = -1.1199786$$

и записываем одно уравнение с одним неизвестным:

$$1.1199786x_4 = -1.1199768.$$

Действительно, в этом случае дальнейшие вычисления определяют:

$$x_1 + 0.5x_2 - 0.05x_3 + 0.5x_4 = 1.35;$$

$$x_2 + 13.4x_3 - 29x_4 = 71.2;$$

$$x_3 - 1.72298x_4 = 4.72298.$$

Наконец,

$$1.11998x_4 = -1.11998.$$

На этом закончен прямой ход.

Обратный ход:

$$x_4 = -1.000;$$

$$x_3 = 4.72298 - 1.72298 = 3;$$

$$x_2 = 71.2 - 13.4 * 3 - 29 = 2;$$

$$x_1 = 1.35 - 0.5 * 2 + 0.05 * 3 + 0.5 = 1.$$

3.3. Схема Гаусса с выбором главного элемента

Рассмотрим СЛАУ

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = a_{1n+1}, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = a_{2n+1}, \\ \dots \quad \dots \quad \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = a_{nn+1}. \end{array} \right. \quad (3.17)$$

Запишем расширенную прямоугольную матрицу коэффициентов системы (3.17):

$$M = \begin{bmatrix} a_{11} & \dots & a_{1q} & a_{1n} & a_{1n+1} \\ \dots & & \dots & & \dots \\ a_{p1} & \dots & a_{pq} & \dots & a_{pn+1} \\ \dots & & \dots & & \dots \\ a_{n1} & \dots & \dots & \dots & a_{nn+1} \end{bmatrix}. \quad (3.18)$$

Среди элементов матрицы a_{ij} ($i, j = 1, \dots, n$) выберем наибольший по модулю, называемый *главным элементом*. Пусть им будет, например, элемент a_{pq} . Строка, содержащая главный элемент, называется **главной строкой**.

Далее вычисляем множители $m_i = a_{iq} / a_{pq}$ для всех $i \neq p$. Затем преобразуем матрицу (3.18) следующим образом: из каждой i -ой неглавной строки вычитаем почленно главную строку, умноженную на m_i . В результате получим матрицу, у которой все элементы q -го столбца за исключением a_{pq} , равны 0. Отбрасывая этот столбец и главную строку, получим новую матрицу M_1 с числом строк и столбцов на 1 меньше.

Над матрицей M_1 повторяем те же операции, после чего получим матрицу M_2 и т.д. Таким образом продолжаем до тех пор, пока не получим матрицу, содержащую одну строку из двух элементов, которую тоже считаем главной.

Затем объединим все главные строки, начиная с последней. После некоторой перестановки они образуют треугольную матрицу, эквивалентную исходной. На этом заканчивается этап вычислений, называемый **прямым ходом**. Решив систему с полученной треугольной матрицей коэффициентов, найдём последовательно значения неизвестных x_i ($i = 1, 2, \dots, n$). На этом заканчивается **обратный ход**.

Смысл выбора главного элемента состоит в том, чтобы сделать возможно меньшими числа m_i и тем самым уменьшить погрешность вычислений.

Пример 3.4. Рассмотрим СЛАУ, состоящую из трех уравнений. Запишем расширенную матрицу

$$M = \begin{bmatrix} 2 & 6 & -1 & -12 \\ 5 & -1 & 2 & 29 \\ -3 & -4 & 1 & 5 \end{bmatrix}; \quad a_{12} = 6 \quad - \quad \max.$$

$$m_2 = -1/6; \quad m_3 = -2/3.$$

$$M_1 = \begin{bmatrix} 16/3 & 11/6 & 27 \\ -5/3 & 1/3 & -3 \end{bmatrix}; \quad a_{11}^{(1)} = 16/3 \quad - \quad \max.$$

$$m_2 = -5/16.$$

$$M_2 = [87/96 \quad 174/32].$$

$$x_3 = 6; \quad x_1 = 3; \quad x_2 = -2.$$

3.4. Вычисление обратной матрицы методом Гаусса

Пусть дана неособенная матрица

$$A = [a_{ij}] \quad (i, j = 1, 2, \dots, n). \quad (3.19)$$

Необходимо найти её обратную матрицу

$$A^{-1} = [x_{ij}] \quad (i, j = 1, 2, \dots, n). \quad (3.20)$$

Вспомним основное соотношение линейной алгебры:

$$A \cdot A^{-1} = E, \quad (3.21)$$

где E – единичная матрица.

Перемножая матрицы A и A^{-1} , получаем n^2 уравнений относительно n^2 неизвестных x_{ij} :

$$\sum_{k=1}^n a_{ik} x_{kj} = \delta_{ij} \quad (i, j = 1, 2, \dots, n), \quad (3.22)$$

где

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

Таким образом, получим n систем линейных уравнений для $j = 1, 2, \dots, n$, имеющих одну и ту же матрицу коэффициентов A и различные столбцы - свободные члены, которые можно *одновременно решить методом Гаусса*.

Рассмотрим это подробнее, вычислив матрицу, обратную $A[4 \times 4]$:

$$A = \begin{bmatrix} 2.0 & 1.0 & -0.1 & 1.0 \\ 0.4 & 0.5 & 4.0 & -8.5 \\ 0.3 & -1.0 & 1.0 & 5.2 \\ 1.0 & 0.2 & 2.5 & -1.0 \end{bmatrix} \quad \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Разделив все коэффициенты первой строки на $a_{11} = 2$, получим первую главную строку (*обратите внимание, что с n столбцами свободных членов проводятся те же действия, что и с одним*):

$$\begin{matrix} 1.0 & 0.5 & -0.05 & 0.5 & & & & & 0.5 & 0 & 0 & 0 \\ \left[\begin{matrix} 0.3 & 4.02 & -8.7 \\ -1.15 & 1.015 & 5.05 \\ -0.3 & 2.55 & -1.5 \end{matrix} \right] & & & & -0.2 & 1 & 0 & 0 \\ & & & & -0.15 & 0 & 1 & 0 \\ & & & & -0.5 & 0 & 0 & 1 \\ 1.0 & 13.4 & -29 & & -0.6667 & 3.333 & 0 & 0 \end{matrix}$$

$$\begin{bmatrix} 16.425 & -28.3 \\ 6.57 & -10.2 \end{bmatrix} \begin{bmatrix} -0.91671 & 3.8333 & 1 & 0 \\ -0.7 & 1 & 0 & 1 \end{bmatrix} \\
 1.0 & -1.723 & -0.055812 & 0.2338 & 0.06088 & 0 \\
 & 1.1201 & -0.3333 & -0.53332 & -0.39998 & 1 \\
 A^{-1} = \begin{bmatrix} 1.45931 & 1.51313 & 1.6143 & -3.00844 \\ -1.67791 & -2.60883 & -2.92694 & 5.277793 \\ -0.56851 & -0.58701 & -0.5544 & 1.53826 \\ -0.29756 & -0.47614 & -0.3571 & 0.89278 \end{bmatrix}.$$

Для проверки перемножим полученную обратную матрицу и исходную (должны получить единичную):

$$E = \begin{bmatrix} 0.99972 & -1.13 * 10^{-4} & 2.16 * 10^{-4} & 5.07 * 10^{-4} \\ 5.02 * 10^{-4} & 1.00020 & 3.71 * 10^{-4} & 8.79 * 10^{-4} \\ 1.16 * 10^{-4} & 3.7 * 10^{-5} & 1.00006 & 6.5 * 10^{-5} \\ 7.4 * 10^{-5} & 2.6 * 10^{-5} & 4.6 * 10^{-5} & 0.99993 \end{bmatrix}.$$

Благодаря округлению, убеждаемся, что обратная матрица вычислена неточно. В дальнейшем можно показать, как методом простой итерации можно уточнить A^{-1} .

3.5. Вычисление определителей методом Гаусса

Пусть дана исходная матрица

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & \dots & \dots & a_{nn} \end{bmatrix}. \tag{3.23}$$

Необходимо вычислить $\Delta = \det A$.

Вспомним свойства определителей:

1. Чтобы умножить (разделить) определитель на какое либо число, достаточно умножить (разделить) на это число строку или столбец:

$$\Delta = a_{11} \begin{vmatrix} 1 & b_{12} & \dots & b_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & \dots & \dots & a_{nn} \end{vmatrix}; \quad (3.24)$$

2. Значение определителя не изменится, если его строку заменить суммой этой строки и другой, умноженной на произвольное число.

Учитывая это свойство, умножая первую строку последовательно на a_{21} , a_{31} , ..., a_{n1} и вычитая из второй, третьей и т.д., получим

$$\Delta = a_{11} \begin{vmatrix} 1 & b_{12} & \dots & b_{1n} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} \\ \dots & \dots & \dots & \dots \\ 0 & a_{n2}^{(1)} & \dots & a_{nn}^{(1)} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22}^{(1)} & a_{23}^{(1)} & \dots & a_{2n}^{(1)} \\ a_{32}^{(1)} & a_{33}^{(1)} & \dots & a_{3n}^{(1)} \\ \dots & \dots & \dots & \dots \\ a_{n2}^{(1)} & \dots & \dots & a_{nn}^{(1)} \end{vmatrix}; \quad (3.25)$$

3. Величина определителя равна сумме произведений элементов строки (столбца) на $(-1)^{i+j} |A_{ij}|$, где $|A_{ij}|$ – соответствующие миноры.

Используя это свойство, представим определитель как сумму произведений элементов первого столбца на соответствующие миноры. При этом учтем, что за исключением первого элемента значения остальных элементов столбца равны нулю.

Таким образом, мы понизили порядок определителя на 1. Применим к полученному определителю порядка $n - 1$ такие же преобразования. Выполняя n шагов, найдем определитель Δ как произведение ведущих элементов:

$$\Delta = a_{11} \cdot a_{22}^{(1)} \cdot \dots \cdot a_{nn}^{(n-1)}. \quad (3.26)$$

Пример 3.5.

$$\begin{aligned} \Delta &= \begin{vmatrix} 2 & 1.0 & -0.1 & 1.0 \\ 0.4 & 0.5 & 4.0 & -8.5 \\ 0.3 & -1.0 & 1.0 & 5.2 \\ 1.0 & 0.2 & 2.5 & -1.0 \end{vmatrix} = 2 \begin{vmatrix} 1 & 0.5 & -0.05 & 0.5 \\ 0 & 0.3 & 4.02 & -8.7 \\ 0 & -1.15 & 1.015 & 5.05 \\ 0 & -0.3 & 2.55 & -1.5 \end{vmatrix} = \\ &= 2 \begin{vmatrix} 0.3 & 4.02 & -8.7 \\ -1.15 & 1.015 & 5.05 \\ -0.3 & 2.55 & -1.5 \end{vmatrix} = 2 \cdot 0.3 \begin{vmatrix} 1.0 & 13.4 & -29 \\ 0 & 16.425 & -28.3 \\ 0 & 6.57 & -10.2 \end{vmatrix} = \end{aligned}$$

$$= 2 \cdot 0.3 \begin{vmatrix} 16.425 & -28.3 \\ 6.57 & -10.2 \end{vmatrix} = 2 \cdot 0.3 \cdot 16.425 \begin{vmatrix} 1.0 & -1.72298 \\ 0 & 1.11998 \end{vmatrix} =$$

$$= 2 \cdot 0.3 \cdot 16.425 \cdot 1.11998 = 11.0374.$$

Замечания

1. При наличии решения, точные методы всегда дадут его через конечное число шагов.
2. В рамках точных методов вычислительная погрешность увеличивается с ростом размеров СЛАУ и не может быть уменьшена.

3.6. Метод простой итерации (метод Якоби)

Рассмотрим систему

$$A \cdot x = f, \quad (3.27)$$

где матрица $A = [a_{ij}]$ ($i, j = 1, 2, \dots, m$) имеет обратную матрицу; $x = (x_1, x_2, x_3, \dots, x_m)$ – вектор неизвестных, f – вектор свободных членов.

Преобразуем систему (3.27) к следующему виду:

$$x_i = \beta_i - \sum_{j=1}^{i-1} \alpha_{ij} x_j - \sum_{j=i+1}^m \alpha_{ij} x_j \quad (i = 1, 2, \dots, m), \quad (3.28)$$

где

$$\beta_i = f_i / a_{ii}, \quad \alpha_{ij} = a_{ij} / a_{ii},$$

при этом предполагаем, что $a_{ii} \neq 0$.

Условимся, как обычно, считать значение суммы равным нулю, если верхний предел суммирования меньше нижнего. Тогда при $i = 1$ уравнение (3.28) имеет вид

$$x_1 = \beta_1 - \sum_{j=2}^m \alpha_{1j} x_j. \quad (3.29)$$

В методе простой итерации (методе Якоби) исходят из записи системы в виде (3.28), итерации при этом определяют следующим образом:

$$x_i^{(n+1)} = \beta_i - \sum_{j=1}^{i-1} \alpha_{ij} x_j^{(n)} - \sum_{j=i+1}^m \alpha_{ij} x_j^{(n)} \quad (3.30)$$

$$(n = 0, 1, \dots, n_0, \quad i = 1, 2, \dots, m).$$

Начальные значения $x_i^{(0)}$ – ($i = 0, 1, \dots, m$) задаются произвольно. Окончание итерационного процесса определяют либо заданием максимального числа итераций n_0 , либо следующим условием:

$$\max_{1 \leq i \leq m} |x_i^{(n+1)} - x_i^{(n)}| \leq \varepsilon, \quad (3.31)$$

где $\varepsilon > 0$.

В качестве нулевого приближения в системе (3.30) примем

$$x_i^{(0)} = \frac{f_i}{a_{ii}}. \quad (3.32)$$

Если последовательность приближений $x_1^{(0)}, x_2^{(0)}, \dots, x_m^{(0)}, x_1^{(1)}, x_2^{(1)}, \dots, x_m^{(1)}, \dots, x_1^{(k)}, x_2^{(k)}, \dots, x_m^{(k)}$ имеет предел

$$x_1 = \lim_{k \rightarrow \infty} x_1^k, \quad x_2 = \lim_{k \rightarrow \infty} x_2^k, \dots, \quad x_m = \lim_{k \rightarrow \infty} x_m^k, \quad (3.33)$$

то этот предел является решением системы (3.28).

Достаточным условием сходимости решения системы (3.27) является то, что матрица A является матрицей с преобладающими диагональными элементами, то есть

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \quad (i = 1, 2, \dots, m). \quad (3.34)$$

3.7. Метод Зейделя

Этот метод представляет собой некоторую модификацию метода простой итерации. Основная его идея заключается в том, что при вычислении $(k+1)$ -го приближения неизвестной x_i учитываются уже вычисленные ранее $(k+1)$ -е приближения $(x_1, x_2, \dots, x_{i-1})$.

Пусть дана приведенная линейная система:

$$x_i = \beta_i - \sum_{j=1}^n \alpha_{ij} x_j \quad (i = 1, 2, \dots, n). \quad (3.35)$$

Выберем произвольно начальные приближения корней $x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}$, стараясь, конечно, чтобы они в какой-то мере соответствовали неизвестным $x_1, x_2, x_3, \dots, x_n$.

Предположим, что k -е приближение $x_i^{(k)}$ корней известно, тогда в соответствии с идеей метода будем строить $(k+1)$ -е приближение по следующим формулам:

$$\begin{cases}
 x_1^{(k+1)} = \beta_1 - \sum_{j=2}^n \alpha_{1j} x_j^{(k)}, \\
 x_2^{(k+1)} = \beta_2 - \alpha_{21} x_1^{(k+1)} - \sum_{j=3}^n \alpha_{2j} x_j^{(k)}, \\
 \dots\dots\dots \\
 x_i^{(k+1)} = \beta_i - \sum_{j=1}^{i-1} \alpha_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n \alpha_{ij} x_j^{(k)}, \\
 \dots\dots\dots \\
 x_n^{(k+1)} = \beta_n - \sum_{j=1}^{n-1} \alpha_{nj} x_j^{(k+1)},
 \end{cases} \quad (3.36)$$

$(k = 0, 1, 2, \dots)$.

Обычно процесс Зейделя сходится быстрее, чем метод Якоби. Бывает, что процесс Зейделя сходится, когда простая итерация расходится и, т.п. Правда, бывает и наоборот. Во всяком случае, достаточные условия сходимости для метода Якоби достаточны и для сходимости метода Зейделя. Если выполняется достаточное условие сходимости для системы (3.35) – по строкам, то в методе Зейделя выгодно расположить уравнения (3.36) так, чтобы первое уравнение системы имело наименьшую сумму модулей коэффициентов:

$$q_1 = \sum_{j=2}^n |\alpha_{1j}|. \quad (3.37)$$

Пример 3.6.

$$\begin{cases}
 2x_1 + 3x_2 - 4x_3 + x_4 = 3, \\
 x_1 - 2x_2 - 5x_3 + x_4 = 2, \\
 5x_1 - 3x_2 + x_3 - 4x_4 = 1, \\
 10x_1 + 2x_2 - x_3 + 2x_4 = 4.
 \end{cases}$$

Для того чтобы обеспечить достаточные условия сходимости итерационного процесса (преобладающие значения диагональных элементов), преобразуем исходную систему и приведем к удобному виду. Чтобы дальнейшие преобразования были понятны, обозначим уравнения исходной системы буквами А, Б, В и Г соответственно:

$$\begin{aligned}
 x_1 &= -0.2x_2 + 0.1x_3 - 0.2x_4 - 0.4; & (\Gamma) \\
 x_2 &= -0.2x_1 - 0.2x_3 + 0.2; & (A - B) \\
 x_3 &= 0.2x_1 - 0.4x_2 + 0.2x_4 - 0.4; & (B) \\
 x_4 &= 0.333x_1 - 1.111. & (2A - B + 2B - \Gamma)
 \end{aligned}$$

Преобразованную систему будем решать методом Зейделя, тогда, с учетом требования (3.37), окончательно получим:

$$x_4^{(k+1)} = 0.333x_1^{(k)} - 1.111;$$

$$\begin{aligned}
 x_2^{(k+1)} &= -0.2x_1^{(k)} + 0.2x_3^{(k)} + 0.2; \\
 x_1^{(k+1)} &= -0.2x_2^{(k+1)} + 0.1x_3^{(k)} - 0.2x_4^{(k+1)} - 0.4; \\
 x_3^{(k+1)} &= 0.2x_1^{(k+1)} - 0.4x_2^{(k+1)} + 0.2x_4^{(k+1)} - 0.4.
 \end{aligned}$$

В качестве нулевого приближения ($k = 0$) возьмем $x_i^{(0)} = \beta_i$. Зададим количество итераций $k = 2$ и все результаты вычислений сведем в табл. 3.1.

Таблица 3.1

Итерация, k	Значения неизвестных				Невязки			
	x ₁	x ₂	x ₃	x ₄	ε ₁	ε ₂	ε ₃	ε ₄
0	-0.4	0.2	0.4	1.111	2.711	1.911	.444	1.422
1	0.263	.36	0.846	1.244	0.309	.0	.734	0.446
2	0.329	.422	0.874	1.199	.095	0.000	.009	0.029

В приведенной таблице кроме значений неизвестных на каждом шаге оценивались **невязки**. Вспомним, что корнями уравнения $f(\bar{x}) = 0$ называются такие значения неизвестных, которые превращают его в тождество. Так как мы используем итерационный (приближенный) метод, значения неизвестных вычисляем приближенно (три, четыре знака после десятичной точки), то, подставляя значения неизвестных в *исходную* систему, справа получим не ноль, а некоторые значения, называемые **невязкой** первого, второго, ... уравнений на k -ом шаге.

Анализ данных, приведенных в табл. 3.1, показывает, что итерационный процесс быстро сходится, о чем свидетельствуют как быстрое уменьшение невязок, так и уменьшение изменений неизвестных (см. формулу (3.31) метода Якоби).

3.8. Метод скорейшего спуска для системы линейных алгебраических уравнений

В рассматриваемом ниже итерационном методе вычислительный алгоритм строится таким образом, чтобы обеспечить минимальную погрешность на шаге (максимально приблизиться к корню).

Представим систему линейных уравнений в следующем виде:

$$\begin{cases}
 f_1 = \sum_{j=1}^n a_{1j}x_j - b_1, \\
 f_2 = \sum_{j=1}^n a_{2j}x_j - b_2, \\
 \dots \\
 f_n = \sum_{j=1}^n a_{nj}x_j - b_n.
 \end{cases} \quad (3.38)$$

Запишем выражение (3.38) в операторной форме:

$$f = A \cdot x - b. \quad (3.39)$$

Здесь приняты следующие обозначения:

$$f = \begin{bmatrix} f_1 \\ f_2 \\ \dots \\ f_n \end{bmatrix}; \quad A = [a_{ij}]; \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}. \quad (3.40)$$

В методе скорейшего спуска решение ищут в виде

$$x^{(p+1)} = x^{(p)} - \mu_p W_p' r_p, \quad (3.41)$$

где $x^{(p)}$ и $x^{(p+1)}$ - векторы неизвестных на p и $p+1$ шагах итераций; вектор невязок на p -ом шаге определяется выражением

$$r_p = A \cdot x^{(p)} - b, \quad (3.42)$$

а

$$\mu_p = \frac{(r_p, WW' r_p)}{(WW' r_p, WW' r_p)}. \quad (3.43)$$

В формуле (3.43) используется скалярное произведение двух векторов, которое определяется следующей формулой:

$$\begin{aligned} (f(x), \varphi(x)) &= \sum_{i=1}^n f_i(x) \varphi_i(x); \\ (f(x), f(x)) &= \sum_{i=1}^n [f_i(x)]^2. \end{aligned} \quad (3.44)$$

В формуле (3.43) W_p' - транспонированная матрица Якоби, вычисленная на p -ом шаге. Матрица Якоби вектор – функции $f(x)$ определяется как

$$W = \frac{df}{dx} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}. \quad (3.45)$$

Нетрудно убедиться, что для системы (3.39) матрица Якоби равна

$$W = \frac{df}{dx} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} = A. \quad (3.46)$$

Как и для метода простой итерации, достаточным условием сходимости метода градиента является преобладание диагональных элементов. В качестве нулевого приближения можно взять $x_i^{(0)} = b_i/a_{ii}$.

Замечания

1. Как видно из выражения (3.45), матрица Якоби не зависит от шага итерации.
2. Требования минимизации погрешности на каждом шаге обусловили то, что метод градиента более сложен (трудоемок), чем методы Якоби и Зейделя.
3. В методе градиента итерационный процесс естественно закончить при достижении $|r_p| \leq \varepsilon$, вектор невязок входит в вычислительную формулу.
4. В приближенных методах можно обеспечить практически любую погрешность, если итерационный процесс сходится.
5. Итерационный процесс можно прервать на любом k -ом шаге и продолжить позднее, введя в качестве нулевого шага значения $x^{(k)}$.
6. В качестве недостатка приближенных методов можно отметить то, что они часто расходятся, достаточные условия сходимости (преобладание диагональных элементов) можно обеспечить только для небольших систем из 3 – 6 уравнений.

Пример 3.7. Методом скорейшего спуска решим систему уравнений

$$\begin{bmatrix} 8 & -1 & -2 & 0 \\ 0 & 10 & -2 & 2 \\ -1 & 0 & 6 & 2 \\ 3 & -1 & 2 & 12 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} - \begin{bmatrix} -2.3 \\ 0.5 \\ 1.2 \\ -3.7 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Так как диагональные элементы матрицы являются преобладающими, то в качестве начального приближения выберем:

$$x^{(0)} = \begin{bmatrix} 0.3 \\ -0.05 \\ -0.2 \\ 0.3 \end{bmatrix}.$$

Следовательно, вектор невязок на нулевом шаге равен

$$r_0 = Ax^0 - b = \begin{bmatrix} 8 & -1 & -2 & 0 \\ 0 & 10 & -2 & 2 \\ -1 & 0 & 6 & 2 \\ 3 & -1 & 2 & 12 \end{bmatrix} \begin{bmatrix} 0.3 \\ -0.05 \\ -0.2 \\ 0.3 \end{bmatrix} - \begin{bmatrix} -2.3 \\ 0.5 \\ 1.2 \\ -3.7 \end{bmatrix} = \begin{bmatrix} 0.55 \\ 0.4 \\ 0.3 \\ 0.45 \end{bmatrix}.$$

Далее последовательно вычисляем

$$A'r_0 = \begin{bmatrix} 5.45 \\ 3.0 \\ 2.0 \\ 6.8 \end{bmatrix}; \quad AA'r_0 = \begin{bmatrix} 36.6 \\ 45.6 \\ 20.15 \\ 98.95 \end{bmatrix};$$

$$\mu_0 = \frac{0.55 \cdot 36.6 + 0.4 \cdot 45.6 + 0.3 \cdot 20.15 + 0.45 \cdot 98.95}{36.6^2 + 45.6^2 + 20.15^2 + 98.95^2} = 0.006532.$$

$$\text{Отсюда } x^{(1)} = x^{(0)} - \mu_0 A'r_0 = \begin{bmatrix} 0.3 \\ -0.05 \\ -0.2 \\ 0.3 \end{bmatrix} - 0.006532 \begin{bmatrix} 5.45 \\ 3.0 \\ 2.0 \\ 6.8 \end{bmatrix} = \begin{bmatrix} 0.2644 \\ -0.0696 \\ -0.2131 \\ 0.2556 \end{bmatrix},$$

причем

$$r_1 = A \cdot x^{(1)} - b = \begin{bmatrix} 0.3109 \\ 0.1020 \\ 0.1684 \\ -0.1966 \end{bmatrix}.$$

Аналогично находятся последующие приближения и оцениваются невязки. Что касается данного примера, можно отметить, что итерационный процесс сходится достаточно медленно (невязки уменьшаются).

3.9. Факторизация обратной матрицы клетками меньших размерностей

Пусть теперь Ω и Ω^{-1} - матрицы размерности $n \times n$ каждая, определяемые четверками клеток A, B, C, D и K, L, M, N как

$$\Omega = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \quad \Omega^{-1} = \begin{pmatrix} K & L \\ M & N \end{pmatrix} \quad (3.47)$$

При этом, Ω^{-1} является обратной матрицей для Ω , т.е.

$$\Omega\Omega^{-1} = E \quad \Omega^{-1}\Omega = E \quad (3.48)$$

а клетки A, B, C, D и K, L, M, N являются матрицами вдвое меньшей размерности, чем матрицы Ω и Ω^{-1} .

Подставляя в (3.48) вместо матриц Ω и Ω^{-1} соответственно их выражения через клетки A, B, C, D и K, L, M, N из (3.47) получаем следующие соотношения:

$$\Omega\Omega^{-1} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} K & L \\ M & N \end{pmatrix}; \quad \Omega^{-1}\Omega = \begin{pmatrix} K & L \\ M & N \end{pmatrix} \begin{pmatrix} A & B \\ C & D \end{pmatrix} \quad (3.49)$$

Матричные соотношения (3.48) и (3.49) порождают восемь уравнений, которые можно представить в виде двух эквивалентных систем матричных уравнений относительно неизвестных матриц K, L, M, N .

При этом, первая система уравнений имеет вид:

$$\begin{cases} AK + BM = E; \\ AL + BN = 0; \\ MA + NC = 0; \\ MB + ND = E. \end{cases} \quad (3.50)$$

Аналогично, вторая группа соотношений их указанной восьмерки, порождает вторую систему матричных уравнений относительно тех же неизвестных матриц K, L, M, N . В этом случае система уравнений имеет вид:

$$\begin{cases} KA + LC = E; \\ KB + LD = 0; \\ CK + DM = 0; \\ CL + DN = E. \end{cases} \quad (3.51)$$

Две системы уравнений (3.50) и (3.51) являются эквивалентными в том смысле, что если $\det A \neq 0$,

то указанные две системы уравнений единственным образом порождают одно и то же решение.

Легко видеть, что система (3.50) исключительно с использованием клетки A^{-1} порождает искомое решение в виде:

$$\begin{cases} N = (D - CA^{-1}B)^{-1}; \\ M = -NCA^{-1}; \\ L = -A^{-1}BN; \\ K = A^{-1} - A^{-1}BM. \end{cases} \quad (3.52)$$

Соответственно эквивалентная система (3.51) исключительно через клетку D^{-1} определяет то же самое решение в следующем виде:

$$\begin{cases} K = (A - BD^{-1}C)^{-1}; \\ L = -KBD^{-1}; \\ M = -D^{-1}CK; \\ N = D^{-1} - D^{-1}CL. \end{cases} \quad (3.53)$$

Вычисление элементов обратной матрицы Ω^{-1} с использованием, например, клетки A^{-1} сводится к выполнению следующих операций:

1. Вычислить матрицу A^{-1} (3.54)

2. Определить матрицу $P = -A^{-1}B$ (3.55)

3. Определить матрицу $R = -CA^{-1}$ (3.56)

4. Вычислить матрицу $N = (D + CP)^{-1}$ (3.57)

5. Вычислить матрицу $M = NR$ (3.58)

6. Вычислить матрицу $L = PN$ (3.59)

7. Вычислить матрицу $K = A^{-1} + PM$ (3.60)

Преимущества алгоритма с использованием вычислительной схемы (3.54) – (3.60) очевидно определяются двумя следующими факторами:

- размерность матричных вычислений вдвое меньше в сравнении с требованиями известных стандартных процедур типа метода Гаусса, что на практике при больших значениях n может иметь решающее значение;
- кроме вычислений типа (8) и (11), алгоритм не содержит других матричных обращений.

Пример. Пусть матрицы Ω и Ω^{-1} определены как

$$\Omega = \begin{pmatrix} 1 & 0 & 1 & 2 \\ -1 & 2 & 3 & 1 \\ 4 & 0 & -2 & 1 \\ 0 & 2 & 1 & 2 \end{pmatrix}; \quad \Omega^{-1} = \begin{pmatrix} 0 & 0.333333 & 0.333333 & -0.333333 \\ -0.5 & 0.166667 & 0.166667 & 0.333333 \\ 0.2 & 0.466667 & 0.066667 & -0.466667 \\ 0.4 & -0.4 & -0.2 & 0.4 \end{pmatrix} \quad (3.61)$$

ЗДЕСЬ ОБРАТНАЯ МАТРИЦА Ω^{-1} ПОЛУЧЕНА ОБЫЧНЫМ ОБРАЗОМ С ИСПОЛЬЗОВАНИЕМ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОБРАЩЕНИЯ НА ОСНОВЕ АЛГОРИТМА ГАУССА ДЛЯ МАТРИЦЫ Ω РАЗМЕРНОСТИ $N \times N$.

Разбивая очевидным образом матрицу Ω на четыре клетки вдвое меньшей размерности, как:

$$A = \begin{pmatrix} 1 & 0 \\ -1 & 2 \end{pmatrix}; \quad B = \begin{pmatrix} 1 & 2 \\ 3 & 1 \end{pmatrix}; \quad C = \begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix}; \quad D = \begin{pmatrix} -2 & 1 \\ 1 & 2 \end{pmatrix}.$$

в соответствии с предложенным алгоритмом можно получить:

$$A^{-1} = \begin{pmatrix} 1 & 0 \\ 0.5 & 0.5 \end{pmatrix};$$

$$P = \begin{pmatrix} -1 & -2 \\ -2 & -1.5 \end{pmatrix}; \quad R = \begin{pmatrix} -4 & 0 \\ -1 & -1 \end{pmatrix};$$

и также четверку матриц:

$$K = \begin{pmatrix} 0 & 0.333333 \\ -0.5 & 0.166667 \end{pmatrix}, \quad L = \begin{pmatrix} 0.333333 & -0.333333 \\ 0.166667 & 0.333333 \end{pmatrix}; \quad (3.62)$$

$$M = \begin{pmatrix} 0.2 & 0.466667 \\ 0.4 & -0.4 \end{pmatrix}; \quad N = \begin{pmatrix} 0.066667 & -0.466667 \\ -0.2 & 0.4 \end{pmatrix}; \quad (3.63)$$

которые при подстановке в (3.47) полученных клеток K, L, M, N из (3.62) и (3.63) порождает обратную матрицу Ω^{-1} как

$$\Omega^{-1} = \begin{pmatrix} 0 & 0.333333 & 0.333333 & -0.333333 \\ -0.5 & 0.166667 & 0.166667 & 0.333333 \\ 0.2 & 0.466667 & 0.066667 & -0.466667 \\ 0.4 & -0.4 & -0.2 & 0.4 \end{pmatrix}$$

полностью совпадающем с матрицей (3.61), полученной обычным образом с использованием обращения на основе алгоритма Гаусса.

Вопросы для самоконтроля

1. Назовите известные вам методы решения СЛАУ.
2. Чем точные методы отличаются от приближенных?
3. Что такое прямой и обратный ход в методе Гаусса?
4. Нужен ли обратный ход при вычислении методом Гаусса а) обратной матрицы; б) определителя?
5. Что такое невязка?
6. Сравните достоинства и недостатки точных и приближенных методов.
7. Что такое матрица Якоби?
8. Надо ли пересчитывать матрицу Якоби на каждом шаге итерации в методе градиента?
9. Исходная СЛАУ решается независимо тремя методами – методом Якоби, методом Зейделя и методом градиента. Будут ли равны значения
10. а) начального приближения (нулевой итерации);
11. б) первой итерации?
12. При решении СЛАУ ($n > 100$) итерационными методами решение расходится. Как найти начальное приближение?

4. ПРИБЛИЖЕННОЕ РЕШЕНИЕ НЕЛИНЕЙНЫХ И ТРАНСЦЕНДЕНТНЫХ УРАВНЕНИЙ

4.1. Постановка задачи

Пусть дано уравнение

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

где функция $f(x)$ определена и непрерывна в конечном или бесконечном интервале $a < x < b$.

Всякое значение ξ , обращающее функцию $f(x)$ в нуль, то есть такое, что $f(\xi) = 0$, называется корнем уравнения (4.1) или нулем функции $f(x)$. Предположим, что уравнение (4.1) имеет лишь **изолированные** корни, то есть для каждого корня существует окрестность, не содержащая других корней этого уравнения.

Приближенное нахождение изолированных действительных корней уравнения (4.1) складывается обычно из двух этапов:

1. **Отделение корней**, то есть установление возможно тесных промежутков $[\alpha, \beta]$, в которых содержится один и только один корень исходного уравнения (4.1).

2. **Уточнение приближенных корней**, то есть доведение их до заданной степени точности.

4.2. Графическое решение уравнений

Действительные корни уравнения $f(x) = 0$ приближенно можно определить как абсциссы точек пересечения графика функции $y = f(x)$ с осью Ox (см. рис. 4.1, *a*). На практике часто бывает удобнее уравнение (4.1) заменить равносильным ему уравнением

$$\varphi(x) = \psi(x), \quad (4.2)$$

где функции $\varphi(x)$ и $\psi(x)$ более простые, чем функция $f(x)$. Тогда, построив графики этих функций, искомые корни получим как абсциссы точек пересечения этих графиков (смотри рис. 4.1, *b*).

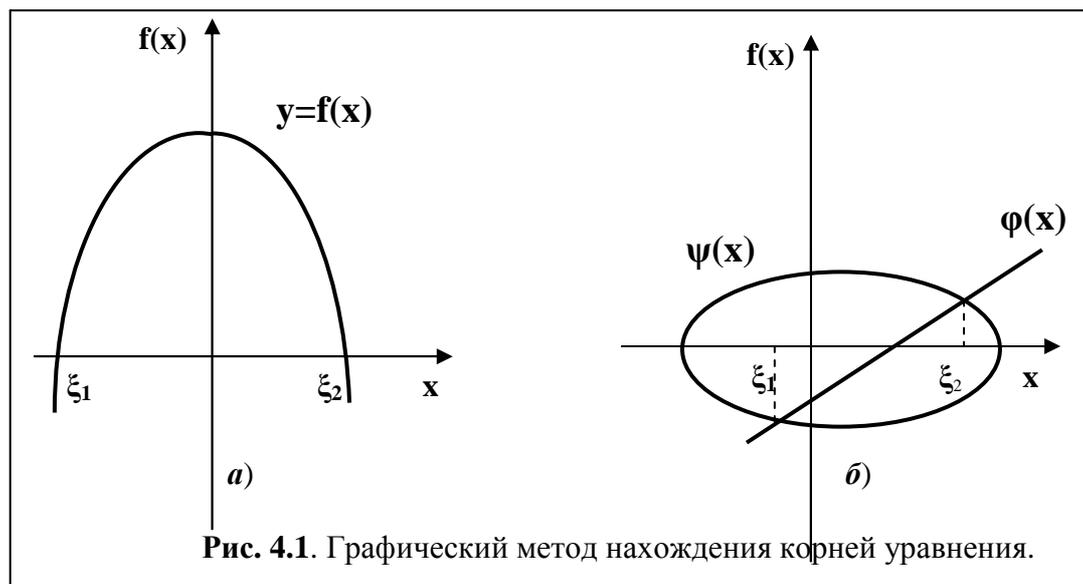


Рис. 4.1. Графический метод нахождения корней уравнения.

4.3. Метод половинного деления (дихотомии)

Сформулируем без доказательства очень важную для рассмотрения дальнейших вопросов теорему.

Теорема: Если непрерывная функция $f(x)$ принимает значения разных знаков на концах отрезка $[\alpha, \beta]$, то есть $f(\alpha) \cdot f(\beta) < 0$, то внутри этого отрезка содержится по

меньшей мере один корень уравнения $f(x) = 0$, а именно: найдётся хотя бы одно число $\xi \in [\alpha, \beta]$ такое, что $f(\xi) = 0$.

Пусть дано уравнение

$$f(x) = 0, \quad (4.3)$$

где функция $f(x)$ определена и непрерывна на интервале $[a, b]$ и $f(a) \cdot f(b) < 0$. Для нахождения корня уравнения делим отрезок $[a, b]$ пополам:

- если $f((a + b)/2) = 0$, то $\xi = (a + b)/2$ является корнем уравнения (4.3);
- если $f((a + b)/2) \neq 0$, то выбираем ту половину отрезка $[a, (a + b)/2]$ или $[(a + b)/2, b]$, на концах которого функция $f(x)$ имеет противоположные знаки.

Новый суженный отрезок $[a_1, b_1]$ снова делим пополам и проводим тот же анализ и т.д.

Очевидно, что закончить уточнение значения корня можно при достижении условия $|a_j - b_j| < \varepsilon$, где $\varepsilon > 0$ - сколь угодно малое число. Второй способ закончить вычисления - задать максимальное значение невязки:

$$f((a_j + b_j)/2) < \varepsilon.$$

Замечания

1. Метод половинного деления очень прост, здесь нет вычислительной формулы и можно обеспечить практически любую точность.
2. Как недостаток метода можно отметить его медленную сходимость (за один шаг интервал, где находится корень, сужается всего в два раза).

4.4. Метод хорд

Пусть дано уравнение

$$f(x) = 0, \quad (4.4)$$

где функция $f(x)$ определена и непрерывна на интервале $[a, b]$ и выполняется соотношение $f(a) \cdot f(b) < 0$.

Пусть для определенности $f(a) < 0, f(b) > 0$. Тогда вместо того, чтобы делить отрезок $[a, b]$ пополам, более естественно разделить его в отношении: $f(a)/f(b)$. При этом новое значение корня определяется из соотношения

$$x_1 = a + h_1, \quad (4.5)$$

где

$$h_1 = \frac{-f(a)}{-f(a) + f(b)}(b - a) = \frac{-f(a)}{f(b) - f(a)}(b - a). \quad (4.6)$$

Далее этот прием применяем к одному из отрезков $[a, x_1]$ или $[x_1, b]$, на концах которого функция $f(x)$ имеет противоположные знаки. Аналогично находим второе приближение x_2 и т.д. (см. рис. 4.2.).

Геометрически этот способ эквивалентен замене кривой $y = f(x)$ хордой, проходящей через точки $A(a, f(a))$ и $B(b, f(b))$.

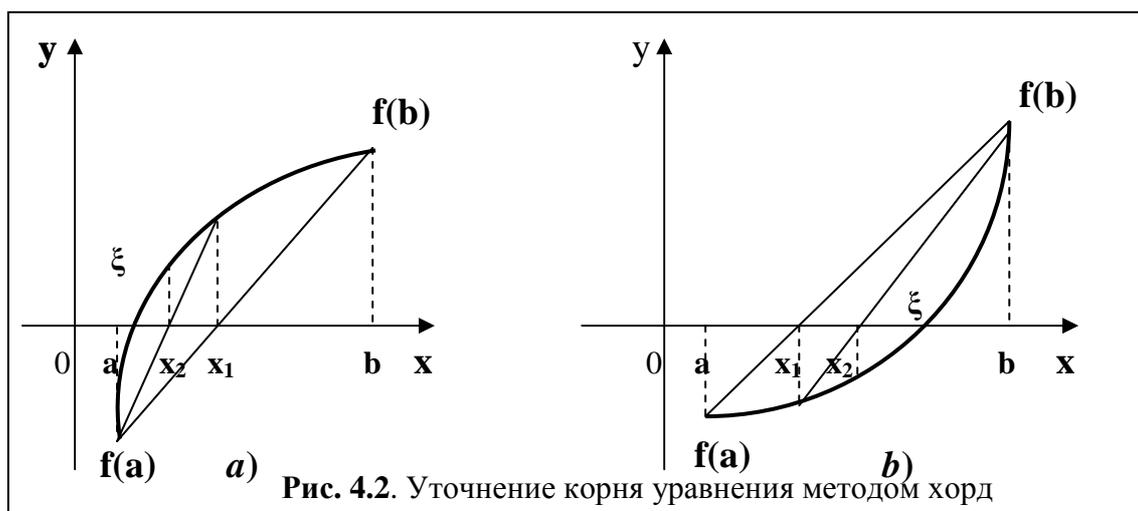


Рис. 4.2. Уточнение корня уравнения методом хорд

Действительно, уравнение хорды AB имеет вид

$$\frac{x-a}{b-a} = \frac{y-f(a)}{f(a)-f(b)}. \quad (4.7)$$

Учитывая, что при $x = x_1 \Rightarrow y = 0$, получим

$$x_1 = a - \frac{f(a)}{f(b)-f(a)}(b-a). \quad (4.8)$$

Полагая, что на отрезке $[a, b]$ вторая производная $f''(x)$ сохраняет постоянный знак, метод хорд сводится к двум различным вариантам:

1. Из рис. 4.2,а видно, что неподвижна точка a , а точка b приближается к ξ , то есть

$$X_{n+1} = a - \frac{f(a)}{f(X_n)-f(a)}(X_n-a). \quad (4.9)$$

Преобразовав выражение (4.9), окончательно получим

$$X_{n+1} = X_n - \frac{f(X_n)}{f(X_n)-f(a)}(X_n-a). \quad (4.10)$$

2. Из рис. 4.2,б видно, что точка b остается неподвижной, а точка a приближается к ξ , тогда вычислительная формула примет вид

$$X_{n+1} = X_n - \frac{f(X_n)}{f(b)-f(X_n)}(b-X_n). \quad (4.11)$$

Таким образом, для вычисления корня уравнения имеем две различные вычислительные формулы (4.10) и (4.11).

Какую точку брать за неподвижную?

Рекомендуется в качестве неподвижной выбирать ту точку, в которой выполняется соотношение

$$f(x) \cdot f''(x) > 0. \quad (4.12)$$

4.5. Метод Ньютона (метод касательных)

Пусть корень ξ уравнения

$$f(x) = 0, \quad (4.13)$$

отделен на отрезке $[a, b]$, причем первая и вторая производные $f'(x)$ и $f''(x)$ непрерывны и сохраняют определенные знаки при $a \leq x \leq b$. Найдя какое-нибудь n -ое приближение корня $x_n \approx \xi$ ($a \leq x_n \leq b$), мы можем уточнить его по методу Ньютона следующим образом. Пусть

$$\xi = x_n + h_n, \quad (4.14)$$

где h_n - величина малая. Отсюда по формуле Тейлора получим (ограничиваясь первым порядком малости относительно h_n)

$$f(x_n + h_n) = f(x_n) + h_n f'(x_n) = 0. \quad (4.15)$$

Следовательно,

$$h_n = -f(x_n) / f'(x_n). \quad (4.16)$$

Подставив полученное выражение в формулу (4.14), найдем следующее (по порядку) значение корня:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (n = 0, 1, \dots). \quad (4.17)$$

Проиллюстрируем графически нахождение корня методом Ньютона (рис. 4.3.).

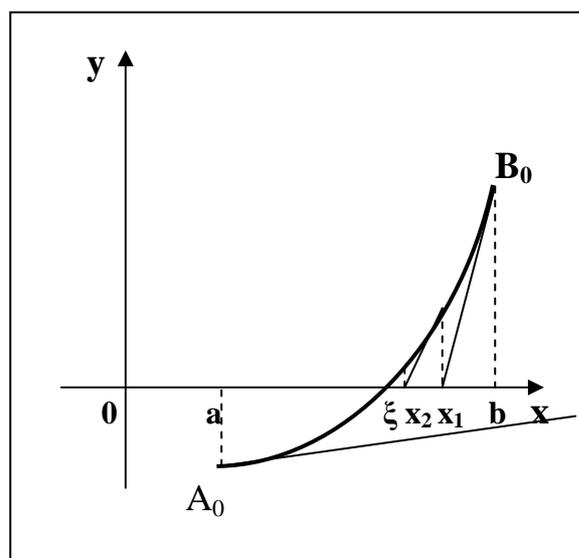


Рис. 4.3. Уточнение корня методом касательных

Если в качестве начального приближения выбрать точку $x_0 = B_0$, то процесс быстро сходится. Если же выбрать точку $x_0 = A_0$, то $x_1 \notin [a, b]$, и процесс нахождения корня расходится. Рекомендуется: в качестве x_0 выбрать точку, где $f(x) \cdot f''(x) > 0$.

4.6. Комбинированный метод

Пусть $f(a) \cdot f(b) < 0$, а $f'(x)$ и $f''(x)$ сохраняют постоянные знаки на отрезке $[a, b]$. Соединяя метод хорд и метод касательных, получаем метод, на каждом шаге которого находим значения по недостатку и значения по избытку точного корня ξ уравнения $f(x) = 0$. Теоретически здесь возможны четыре случая:

- $f'(x) > 0; f''(x) > 0;$
- $f'(x) > 0; f''(x) < 0;$
- $f'(x) < 0; f''(x) > 0;$
- $f'(x) < 0; f''(x) < 0.$

Рассмотрим только первый случай, так как остальные три ведут себя аналогично и могут быть сведены к первому.

Итак, пусть $f'(x) > 0$ и $f''(x) > 0$ при $a \leq x \leq b$. Полагаем, что $x_0 = a$ (для метода хорд), $\bar{x}_0 = b$ (для метода касательных). Тогда новые значения корня вычисляем по формулам

$$\begin{aligned} x_{n+1} &= x_n - \frac{f(x_n)}{f(\bar{x}_n) - f(x_n)} (\bar{x}_n - x_n); \\ \bar{x}_{n+1} &= \bar{x}_n - \frac{f(\bar{x}_n)}{f'(\bar{x}_n)} \quad (n = 0, 1, 2, \dots). \end{aligned} \tag{4.18}$$

Рис. 4.4 наглядно иллюстрирует суть комбинированного метода.

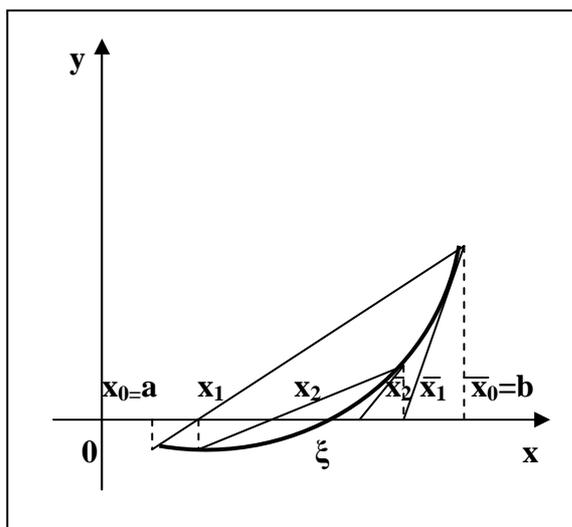


Рис. 4.4. Уточнение корня комбинированным методом

Доказано, что $x_n < \xi < \bar{x}_n$. Следует обратить внимание на то, что на каждом шаге метод хорд применяется к новому отрезку $[x_n, \bar{x}_n]$. Если задать максимальное значение погрешности $\varepsilon > 0$, процесс уточнения значения корня продолжаем до тех пор, пока не выполнится условие

$$f((x_n + \bar{x}_n)/2) \leq \varepsilon. \quad (4.19)$$

Пример 4.1. Вычислить с точностью до 0.0005 положительный корень уравнения

$$f(x) = x^5 - x - 0.2 = 0.$$

На первом этапе отделения корней выбрали интервал $[1.0, 1.1]$, на концах которого функция имеет противоположные знаки. Действительно,

$$f(1) = -0.2 < 0,$$

$$f(1.1) = 0.31051 > 0.$$

В выбранном нами интервале $f'(x) > 0$, $f''(x) > 0$, то есть знаки производных сохраняются.

Применим комбинированный метод, приняв $x_0 = 1.0$, $\bar{x}_0 = 1.1$. По формулам (4.18) вычислим

$$x_1 = 1.03917651; \quad \bar{x}_1 = 1.050872558;$$

$$f((x_1 + \bar{x}_1)/2) = 0.0013037 > 0.0005.$$

Так как точность недостаточная (погрешность велика), вычислим следующие значения:

$$x_2 = 1.044683244; \quad \bar{x}_2 = 1.044846218;$$

$$f((x_2 + \bar{x}_2)/2) = 0.0000150248 < 0.0005.$$

Таким образом, за два шага мы обеспечили требуемую точность.

Замечания

1. Метод, как и метод Ньютона не всегда сходится (почему?).
2. Комбинированный метод сходится быстрее всех ранее рассмотренных, (если он сходится).
3. Комбинированный метод наиболее трудоемок.

$$f(x) = f(x^{(p)} + \varepsilon^{(p)}) = 0. \quad (5.4)$$

Предположим, что функция $f(x)$ - непрерывно дифференцируема в некоторой выпуклой области, содержащей x и $x^{(p)}$. Тогда левую часть уравнения (5.4) разложим в ряд Тейлора по степеням малого вектора $\varepsilon^{(p)}$, ограничиваясь линейными членами:

$$f(x^{(p)} + \varepsilon^{(p)}) = f(x^{(p)}) + f'(x^{(p)})\varepsilon^{(p)} = 0, \quad (5.5)$$

или в развернутом виде:

$$\left\{ \begin{array}{l} f_1(x_1^{(p)} + \varepsilon_1^{(p)}, x_2^{(p)} + \varepsilon_2^{(p)}, \dots, x_n^{(p)} + \varepsilon_n^{(p)}) = f_1(x_1^{(p)}, x_2^{(p)}, \dots, x_n^{(p)}) + \\ + f'_{1,x_1}(x_1^{(p)}, x_2^{(p)}, \dots, x_n^{(p)})\varepsilon_1^{(p)} + f'_{1,x_2}(x_1^{(p)}, x_2^{(p)}, \dots, x_n^{(p)})\varepsilon_2^{(p)} + \dots + \\ + f'_{1,x_n}(x_1^{(p)}, x_2^{(p)}, \dots, x_n^{(p)})\varepsilon_n^{(p)} = 0, \\ \dots\dots\dots \\ f_n(x_1^{(p)} + \varepsilon_1^{(p)}, x_2^{(p)} + \varepsilon_2^{(p)}, \dots, x_n^{(p)} + \varepsilon_n^{(p)}) = f_n(x_1^{(p)}, x_2^{(p)}, \dots, x_n^{(p)}) + \\ + f'_{n,x_1}(x_1^{(p)}, x_2^{(p)}, \dots, x_n^{(p)})\varepsilon_1^{(p)} + \dots + f'_{n,x_n}(x_1^{(p)}, x_2^{(p)}, \dots, x_n^{(p)})\varepsilon_n^{(p)} = 0. \end{array} \right. \quad (5.6)$$

Из анализа формул (5.5) и (5.6) следует, что под производной $f'(x)$ следует понимать матрицу Якоби системы функций f_1, f_2, \dots, f_n , относительно переменных $x_1, x_2, x_3, \dots, x_n$, то есть:

$$f'(x) = W(x) = \begin{vmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{vmatrix}. \quad (5.7)$$

Выражение (5.7) в краткой записи можно представить:

$$f'(x) = W(x) = \left[\frac{\partial f_i}{\partial x_j} \right] \quad (i, j = 1, 2, \dots, n). \quad (5.8)$$

Выражение (5.6) представляет собой линейную систему относительно поправок $\varepsilon_i^{(p)}$ ($i = 1, 2, \dots, n$) с матрицей $W(x)$, поэтому формула (5.5) может быть записана в следующем виде:

$$f(x^{(p)}) + W(x^{(p)})\epsilon^{(p)} = 0. \quad (5.9)$$

Отсюда, предполагая, что матрица $W(x^{(p)})$ - неособенная, получим:

$$\epsilon^{(p)} = -W^{-1}(x^{(p)})f(x^{(p)}). \quad (5.10)$$

Теперь, подставив выражение (5.10) в формулу (5.3), окончательно получим:

$$x^{(p+1)} = x^{(p)} - W^{-1}(x^{(p)})f(x^{(p)}) \quad (p = 0, 1, \dots, k). \quad (5.11)$$

Таким образом, получили вычислительную формулу (метод Ньютона), где в качестве нулевого приближения $x^{(0)}$ можно взять приближенное (грубое) значение искомого корня.

Пример 5.1. Рассмотрим применение метода Ньютона на примере системы двух нелинейных уравнений

$$\begin{cases} f_1(x_1, x_2) = x_1 + 3 \cdot \lg x_1 - x_2^2 = 0, \\ f_2(x_1, x_2) = 2x_1^2 - x_1x_2 - 5x_1 + 1 = 0. \end{cases} \quad (5.12)$$

Прежде чем разбирать конкретные шаги по решению системы (5.12), распишем в общем виде якобиан для системы из двух уравнений

$$W = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}.$$

Здесь A, B, C, D – функционалы от переменных x_1, x_2 . Нас фактически интересует W^{-1} . Пусть матрица W - неособенная, тогда обратная матрица вычисляется

$$W^{-1} = \frac{1}{\Delta} \begin{bmatrix} D & -B \\ -C & A \end{bmatrix}; \quad \Delta = |AD - BC|.$$

Теперь вернемся к системе (5.12). Графическое решение этой системы дает две точки пересечения: $M_1 (1.4; -1.5)$ и $M_2 (3.4; 2.2)$. Зададим начальное приближение:

$$x^{(0)} = \begin{bmatrix} 3.4 \\ 2.2 \end{bmatrix}; \quad W = \begin{bmatrix} 1 + \frac{3}{x_1 \ln 10} & -2x_2 \\ 4x_1 - x_2 - 5 & -x_1 \end{bmatrix} = \begin{bmatrix} 1.3832 & -4.4 \\ 6.4 & -3.4 \end{bmatrix}.$$

$$\Delta = \det W(x^{(0)}) = 23.457; \quad W^{-1}(x^{(0)}) = \frac{1}{\Delta} \begin{bmatrix} -3.4 & 4.4 \\ -6.4 & 1.3832 \end{bmatrix}.$$

Здесь $x^{(p)}$ и $x^{(p+1)}$ - векторы неизвестных на p и $p+1$ шагах итераций; вектор невязок на p -ом шаге $-f^{(p)} = f(x^{(p)})$; W'_p - транспонированная матрица Якоби на p -ом шаге;

$$\mu_p = \frac{(f^{(p)}, W_p W'_p f^{(p)})}{(W_p W'_p f^{(p)}, W_p W'_p f^{(p)})};$$

$$W_p = \left[\frac{\partial f_i^{(p)}}{\partial x_j^{(p)}} \right] \quad (i, j = 1, 2, \dots, n).$$

Пример 5.2. Методом скорейшего спуска вычислим приближенно корни системы

$$\begin{cases} x + x^2 - 2yz = 0.1, \\ y - y^2 + 3xz = -0.2, \\ z + z^2 + 2xy = 0.3, \end{cases}$$

расположенные в окрестности начала координат.

Имеем:

$$f = \begin{bmatrix} x + x^2 - 2yz - 0.1 \\ y - y^2 + 3xz + 0.2 \\ z + z^2 + 2xy - 0.3 \end{bmatrix}; \quad W = \begin{bmatrix} 1 + 2x & -2z & -2y \\ 3z & 1 - 2y & 3x \\ 2y & 2x & 1 + 2z \end{bmatrix}.$$

Выберем начальное приближение:

$$x^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}; \quad \Rightarrow \quad f^{(0)} = \begin{bmatrix} -0.1 \\ 0.2 \\ -0.3 \end{bmatrix}; \quad W_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = E.$$

По вышеприведенным формулам найдем первое приближение:

$$\mu_0 = \frac{(f^{(0)}, f^{(0)})}{(f^{(0)}, f^{(0)})} = 1; \quad x^{(1)} = x^{(0)} - 1 \cdot E f^{(0)} = \begin{bmatrix} 0.1 \\ -0.2 \\ 0.3 \end{bmatrix}.$$

Аналогичным образом находим следующее приближение:

$$f^{(1)} = \begin{bmatrix} 0.13 \\ 0.05 \\ 0.05 \end{bmatrix}; \quad W_1 = \begin{bmatrix} 1.2 & -0.6 & 0.4 \\ 0.9 & 1.4 & 0.3 \\ -0.4 & 0.2 & 1.6 \end{bmatrix}; \quad W_1' f^{(1)} = \begin{bmatrix} 0.181 \\ 0.002 \\ 0.147 \end{bmatrix};$$

$$W_1 W_1' f^{(1)} = \begin{bmatrix} 0.2748 \\ 0.2098 \\ 0.1632 \end{bmatrix}; \quad \mu_1 = \frac{0.054374}{0.14619797} = 0.3719;$$

$$x^{(2)} = \begin{bmatrix} 0.1 \\ -0.2 \\ 0.3 \end{bmatrix} - 0.3719 \cdot \begin{bmatrix} 0.181 \\ 0.002 \\ 0.147 \end{bmatrix} = \begin{bmatrix} 0.0327 \\ -0.2007 \\ 0.2453 \end{bmatrix}.$$

Ограничимся двумя итерациями (шагами), и оценим невязку:

$$f^{(2)} = \begin{bmatrix} 0.032 \\ -0.017 \\ -0.007 \end{bmatrix}.$$

Замечания

1. Как видно из примера, решение достаточно быстро сходится, невязка быстро убывает.
2. При решении системы нелинейных уравнений методом градиента матрицу Якоби необходимо пересчитывать на каждом шаге (итерации).

Вопросы для самопроверки

1. Как найти начальное приближение:
 - а) для метода Ньютона;
 - б) для метода градиента?
2. В методе скорейшего спуска вычисляется Якобиан (матрица Якоби). Чем отличается Якобиан, вычисленный для СЛАУ, от Якобиана, вычисленного для нелинейной системы уравнений?
3. Каков критерий остановки итерационного процесса при решении системы нелинейных уравнений:
 - а) методом Ньютона;
 - б) методом скорейшего спуска?

6. РЕШЕНИЕ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

6.1. Методы решения задачи Коши

Среди задач, с которыми приходится иметь дело в вычислительной практике, значительную часть составляют различные задачи, сводящиеся к решению обыкновенных дифференциальных уравнений. Обычно приходится прибегать к помощи приближенных методов решения подобных задач. В случае обыкновенных дифференциальных уравнений в зависимости от того, ставятся ли дополнительные условия в одной или нескольких точках отрезка изменения независимой переменной, задачи обычно подразделяются на *одноточечные* (задачи с начальными условиями или *задачи Коши*) и *многоточечные*. Среди многоточечных задач наиболее часто в прикладных вопросах встречаются так называемые *граничные задачи*, когда дополнительные условия ставятся на концах рассматриваемого отрезка.

В дальнейшем ограничимся рассмотрением численных методов решения задачи Коши. Для простоты изложения методов решения задачи будем рассматривать случай одного обыкновенного дифференциального уравнения первого порядка.

Пусть на отрезке $x_0 \leq x \leq b$ требуется найти решение $y(x)$ дифференциального уравнения:

$$y' = f(x, y), \quad (6.1)$$

удовлетворяющее при $x = x_0$ начальному условию

$$y(x_0) = y_0. \quad (6.2)$$

Будем считать, что условия существования и единственности решения поставленной задачи Коши выполнены.

На практике найти общее либо частное решение задачи Коши удается крайне редко, поэтому приходится решать эту задачу приближенно. Отрезок $[x_0, b]$ покрывается сеткой (разбивается на интервалы) чаще всего с постоянным шагом h ($h = x_{n+1} - x_n$), и по какому-то решающему правилу находится значение $y_{n+1} = y(x_{n+1})$. Таким образом, в качестве решения задачи Коши численными методами мы получаем таблицу, состоящую из двух векторов: $x = (x_0, x_1, \dots, x_n)$ – вектора аргументов и соответствующего ему вектора функции $y = (y_0, y_1, \dots, y_n)$.

Численные методы (правила), в которых для нахождения значения функции в новой точке используется информация только об одной (предыдущей) точке, называются **одношаговыми**.

Численные методы (правила), в которых для нахождения значения функции в новой точке используется информация о нескольких (предыдущих) точках, называются **многшаговыми**.

Из общего курса обыкновенных дифференциальных уравнений широкое распространение получил аналитический метод, основанный на идее разложения в ряд решения рассматриваемой задачи Коши. Особенно часто для этих целей используется ряд Тейлора. В этом случае вычислительные правила строятся особенно просто. При этом приближенное решение $y_m(x)$ исходной задачи ищут в виде

$$y_m(x) = \sum_{i=0}^m \frac{(x-x_0)^i}{i!} y^{(i)}(x_0), \quad (6.3)$$

$$x_0 \leq x \leq b.$$

Здесь $y^{(0)}(x_0) = y(x_0)$, $y^{(1)}(x_0) = y'(x_0) = f(x_0, y_0)$, а значения $y^{(i)}(x_0)$, $i = 2, 3, \dots, m$ находят по формулам, полученным последовательным дифференцированием уравнения (6.1):

$$y^{(2)}(x_0) = y''(x_0) = f_x(x_0, y_0) + f(x_0, y_0) f_y(x_0, y_0);$$

$$y^{(3)}(x_0) = y'''(x_0) = f_{x^2}(x_0, y_0) + 2f(x_0, y_0) f_y(x_0, y_0) +$$

$$+ f^2(x_0, y_0) f_{y^2}(x_0, y_0) + f_y(x_0, y_0) [f_x(x_0, y_0) +$$

$$+ f(x_0, y_0) f_y(x_0, y_0)]; \quad (6.4)$$

...

$$y^{(m)}(x_0) = F_m(f; f_x; f_{x^2}; f_{xy}; f_{y^2}; \dots; f_{x^{m-1}}; f_{y^{m-1}}) x = x_0, y = y_0.$$

Для значений x , близких к x_0 , метод рядов (6.3) при достаточно большом значении m дает обычно хорошее приближение к точному решению $y(x)$ задачи (6.1). Однако с ростом расстояния $|x - x_0|$ погрешность приближенного равенства $y(x) \approx y_m(x)$, вообще говоря, возрастает по абсолютной величине, и правило (6.3) становится вовсе неприемлемым, когда x выходит из области сходимости соответствующего ряда (6.3) Тейлора.

Если в выражении (6.3) ограничиться $m = 1$, то для вычисления новых значений $y(x)$ нет необходимости пересчитывать значение производной, правда и точность решения будет невысока. Графическая интерпретация этого метода приведена на рис. 6.1.

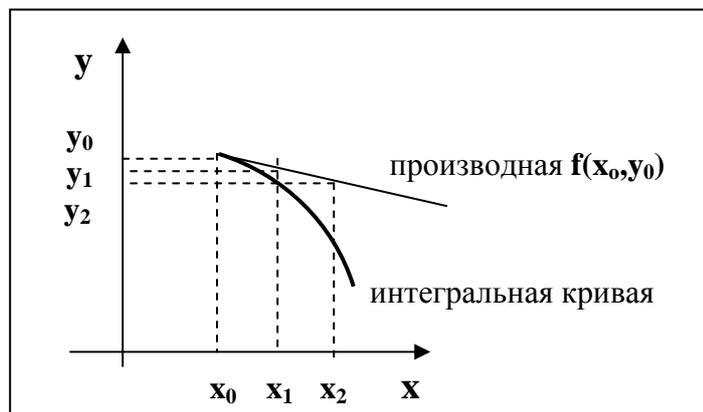


Рис. 6.1. Разложение функции в ряд Тейлора ($m=1$)

6.2. Метод рядов, не требующий вычисления производных правой части уравнения

Естественно поставить задачу о таком усовершенствовании приведенного выше одношагового метода, которое сохраняло бы основные его достоинства, но не было бы связано с нахождением значений производных правой части уравнения

$$y_m(x_{n+1}) \approx \sum_{i=0}^m \frac{h^i}{i!} y^{(i)}(x_n), \quad (6.5)$$

где

$$x_{n+1} = x_n + h.$$

Чтобы выполнить это условие (последнее), производные $y^{(i)}(x)$, $i = 2, 3, \dots, m$, входящие в правую часть уравнения (6.5), можно заменить по формулам численного дифференцирования их приближенными выражениями через значение функции y' и учесть, что $y'(x) = f[x, y(x)]$.

В случае $m = 1$ приближенное равенство (6.5) не требует вычисления производных правой части уравнения и позволяет с погрешностью порядка h^2 находить значение $y(x_n + h)$ решения этого уравнения по известному его значению $y(x_n)$. Соответствующее одношаговое правило можно записать в виде

$$y_{n+1} = y_n + hf_n. \quad (6.6)$$

Это правило (6.6) впервые было построено Эйлером и носит его имя. Иногда его называют также правилом *ломаных* или методом *касательных*. Метод Эйлера имеет простую геометрическую интерпретацию (см. рис. 6.2).

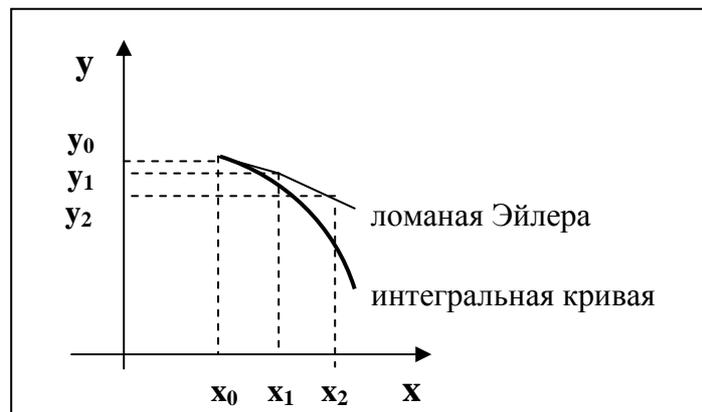


Рис. 6.2. Нахождение решения методом Эйлера

Замечание Метод Эйлера имеет порядок точности $\sim h^2$ на одном шаге. Практическая оценка погрешности приближенного решения может быть получена по правилу Рунге.

6.3. Метод Рунге-Кутты

Изложим идею метода на примере задачи Коши:

$$\begin{aligned}
 y' &= f(x, y); \\
 x_0 &\leq x \leq b; \\
 y(x_0) &= y_0.
 \end{aligned}
 \tag{6.7}$$

Интегрируя это уравнение в пределах от x до $x + h$ ($0 < h < 1$), получим равенство

$$y(x + h) = y(x) + \int_x^{x+h} f[t, y(t)] dt,
 \tag{6.8}$$

которое посредством последнего интеграла связывает значения решения рассматриваемого уравнения в двух точках, удаленных друг от друга на расстояние шага h .

Для удобства записи выражения (6.8) используем обозначение:

$$\Delta y = y(x + h) - y(x)$$

и замену переменной интегрирования:

$$t = x + \alpha h.$$

Окончательно получим:

$$\Delta y = h \int_0^1 f[x + \alpha h, y(x + \alpha h)] d\alpha.
 \tag{6.9}$$

Указав эффективный метод приближенного вычисления интеграла в выражении (6.9), мы получим при этом одно из правил численного интегрирования уравнения (6.7).

Постараемся составить линейную комбинацию величин φ_i , $i = 0, 1, \dots, q$, которая будет являться аналогом квадратурной суммы и позволит вычислить приближенное значение приращения Δy :

$$\Delta y \approx \sum_{i=0}^q a_i \varphi_i,
 \tag{6.10}$$

где

$$\varphi_0 = hf(x, y);$$

$$\varphi_1 = hf(x + \alpha_1 h; y + \beta_{10} \varphi_0);$$

$$\varphi_2 = hf(x + \alpha_2 h; y + \beta_{20} \varphi_0 + \beta_{21} \varphi_1);$$

...

Метод четвертого порядка для $q = 3$, являющийся аналогом широко известной в литературе четырехточечной квадратурной формулы "трех восьмых", имеет вид

$$\Delta y \approx \frac{1}{8} (\varphi_0 + 3\varphi_1 + 3\varphi_2 + \varphi_3),
 \tag{6.11}$$

где

$$\begin{aligned}\varphi_0 &= hf(x_n, y_n); \\ \varphi_1 &= hf\left(x_n + \frac{h}{3}, y_n + \frac{\varphi_0}{3}\right); \\ \varphi_2 &= hf\left(x_n + \frac{2}{3}h, y_n - \frac{\varphi_0}{3} - \varphi_1\right); \\ \varphi_3 &= hf(x_n + h, y_n + \varphi_0 - \varphi_1 + \varphi_2).\end{aligned}$$

Особо широко известно другое вычислительное правило типа Рунге-Кутты четвертого порядка точности:

$$\Delta y = \frac{1}{6}(\varphi_0 + 2\varphi_1 + 2\varphi_2 + \varphi_3), \quad (6.12)$$

где

$$\begin{aligned}\varphi_0 &= hf(x_n, y_n), \\ \varphi_1 &= hf\left(x_n + \frac{h}{2}, y_n + \frac{\varphi_0}{2}\right), \\ \varphi_2 &= hf\left(x_n + \frac{h}{2}, y_n + \frac{\varphi_1}{2}\right), \\ \varphi_3 &= hf(x_n + h, y_n + \varphi_2).\end{aligned}$$

Метод Рунге-Кутты имеет погрешность четвертого порядка ($\sim h^4$).

Правило Рунге. Если приближенный метод имеет порядок погрешности m , то погрешность можно приближенно оценить по формуле

$$\varepsilon(x_i) \approx h^m O(x_i) \approx \frac{y_i^h - y_i^{2h}}{2^m - 1}. \quad (6.13)$$

В формуле (6.13) $O(x_i)$ – главный член погрешности, y_i^h и y_i^{2h} – приближенные решения в точке x_i , найденные с шагом h и $2h$ соответственно.

Пример 6.1. Решить дифференциальное уравнение $y' = \frac{y-x}{y+x}$ на отрезке $[0, 1]$ с начальным условием $y(x=0) = 1$. Найти первые три точки, приняв шаг $h = 0.05$.

Решение. Поставленная задача была решена методом разложения в ряд Тейлора (6.3); методом Эйлера (6.6) и методом Рунге-Кутты (6.12). Для наглядности все полученные результаты сведем в табл. 6.1.

Таблица 6.1

x_i	Ряд Тейлора ($m=1$)	Метод Эйлера	Метод Рунге-Кутты
-------	--------------------------	-----------------	-------------------

	y_i	y_i	y_i	$f(x_i, y_i)$	Φ_0	Φ_1	Φ_2	Φ_3
0	1	1	1	1	-	-	-	-
0.05	1.05	1.05	1.0477	0.9089	0.05	0.0477	0.0476	0.0454
0.1	1.1	1.0931	1.0912	0.8321	0.0454	0.0435	0.0434	0.0416
0.15	1.15	1.1347	1.1311	0.7658	0.0416	0.0399	0.0399	0.0383

6.4. Многошаговые методы

Ранее нами были рассмотрены одношаговые методы решения задачи Коши. Эти методы, обладая рядом удобных для практики вычислений особенностей, страдают одним существенным недостатком. При построении этих методов привлекается информация о решаемой задаче только на отрезке длиной в один шаг, поэтому подобная информация на каждом этапе процесса должна быть получена заново, что предопределяет большую трудоемкость соответствующих вычислительных правил.

Если отказаться от требования одношаговости, можно вычислительные методы строить таким образом, чтобы часть получаемой информации могла быть использована повторно на нескольких следующих шагах вычислительного процесса. Такие методы, использующие информацию о решаемой задаче на отрезке длиной более одного шага, и называются многошаговыми.

Будем, как и раньше рассматривать задачу Коши:

$$\begin{aligned} y' &= f(x, y); \\ x_0 &\leq x \leq b; \\ y(x_0) &= y_0. \end{aligned} \quad (6.14)$$

Ограничимся рассмотрением многошаговых методов с равномерной сеткой:

$$x_i = x_0 + ih; \quad i = 0, 1, \dots, n; \quad n \cdot h = b - x_0. \quad (6.15)$$

Рассмотрим вычислительные правила вида

$$\begin{aligned} y_{n+1} &= y_n + h \sum_{i=-s}^q A_i f(x_{n-i}, y_{n-i}); \\ \sum_{i=0}^q A_i &= 1; \quad \sum_{i=0}^q A_i (-i)^j = \frac{1}{j+1} \quad (j = 1, 2, \dots, q). \end{aligned} \quad (6.16)$$

Среди вычислительных правил вида (6.16) особенно широко известны *экстраполяционные* (при $s = 0$) и *интерполяционные* (при $s = 1, A_{-1} \neq 0$).

6.5. Экстраполяционные методы Адамса

Экстраполяционные формулы Адамса получаются из (6.16) при $s = 0$. Если же предположим при этом, что $q = 0$, то получим уже знакомый нам метод Эйлера:

$$y_{n+1} = y_n + h f(x_n, y_n). \quad (6.17)$$

При $q = 3$ из (6.16) получим следующий вид формулы Адамса:

$$y_{n+1} = y_n + \varphi_n + \frac{1}{2} \Delta \varphi_{n-1} + \frac{5}{12} \Delta^2 \varphi_{n-2} + \frac{3}{8} \Delta^3 \varphi_{n-3}. \quad (6.18)$$

Здесь приняты следующие обозначения:

$$\varphi_i = y'(x_i) = f(x_i, y_i). \quad (6.19)$$

Рекуррентная формула для определения конечных разностей j -го порядка имеет вид:

$$\Delta^j \varphi_i = \Delta^{j-1} \varphi_{i+1} - \Delta^{j-1} \varphi_i. \quad (6.20)$$

Учитывая (6.20), получим:

$$\begin{aligned} \Delta \varphi_{n-1} &= \varphi_n - \varphi_{n-1} = h(f(x_n, y_n) - f(x_{n-1}, y_{n-1})); \\ \Delta^2 \varphi_{n-2} &= \Delta \varphi_{n-1} - \Delta \varphi_{n-2} = \Delta \varphi_{n-1} - (\varphi_{n-1} - \varphi_{n-2}) = \\ &= \Delta \varphi_{n-1} - h[f(x_{n-1}, y_{n-1}) - f(x_{n-2}, y_{n-2})]; \\ \Delta^3 \varphi_{n-3} &= \Delta^2 \varphi_{n-2} - \Delta^2 \varphi_{n-3} = h[f(x_n, y_n) - 2f(x_{n-1}, y_{n-1}) + \\ &+ f(x_{n-2}, y_{n-2})] - h[f(x_{n-1}, y_{n-1}) - 2f(x_{n-2}, y_{n-2}) + \\ &+ f(x_{n-3}, y_{n-3})] = h[f(x_n, y_n) - 3f(x_{n-1}, y_{n-1}) + \\ &+ 3f(x_{n-2}, y_{n-2}) - f(x_{n-3}, y_{n-3})]. \end{aligned} \quad (6.21)$$

6.6. Интерполяционные методы Адамса

При $s = 1$ формула (6.16) примет вид

$$y_{n+1} = y_n + h \sum_{i=-1}^q A_i f(x_{n-i}, y_{n-i}). \quad (6.22)$$

Если $q = 2$, получим следующее вычислительное правило:

$$y_{n+1} = y_n + \Phi_{n+1} + \frac{1}{2} \Delta \Phi_n - \frac{1}{12} \Delta^2 \Phi_{n-1} - \frac{1}{24} \Delta^3 \Phi_{n-2}. \quad (6.23)$$

Обычно на практике используют экстраполяционную формулу (6.18), а затем корректируют полученное значение по формуле (6.23). И если результат уточненного значения не превышает допустимую погрешность расчета, то шаг h считается допустимым

$$|y_{n+1}^{kopp} - y_{n+1}^{эктр}| \leq \varepsilon.$$

Для расчетов на компьютере формулы (6.18) и (6.23) в конечно-разностном виде неудобны. С учетом (6.21) их можно представить в виде

$$\begin{aligned} y_{n+1}^э &= y_n + \frac{h}{24} (55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}); \\ y_{n+1}^у &= y_n + \frac{h}{24} (9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}). \end{aligned} \quad (6.24)$$

Приведенные формулы имеют достаточно большую точность. Они дают погрешность порядка $\sim O(h^4)$, но сами формулы оценки погрешности достаточно сложны. Приближенно погрешность можно оценить по правилу Рунге.

Пример 6.2. Решить дифференциальное уравнение

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

на отрезке $[0, 1]$ с начальным условием

$$y(x=0) = 1.$$

Найти решение методом Адамса (с коррекцией) в точке x_4 , решение в трех первых точках найти методом Рунге-Кутты, приняв шаг $h = 0.05$; $\varepsilon \approx 0.0001$.

Решение. Значения функции в четырех первых точках возьмем из табл. 6.1 (см. пример в предыдущем разделе). Теперь стало понятно, зачем мы сохраняли значения первой производной в этих точках (см. формулы (6.24)).

$$x_4 = x_3 + h = 0.15 + 0.05 = 0.2;$$

$$\begin{aligned} y_4^э &= y_3 + \frac{h}{24} (55f_3 - 59f_2 + 37f_1 - 9f_0) = 1.1311 + \frac{0.05}{24} (55 * 0.7658 - \\ &- 59 * 0.8321 + 37 * 0.9089 - 9 * 1) = 1.1679. \end{aligned}$$

Для того чтобы скорректировать полученный результат, необходимо вычислить значение производной в этой точке:

$$f(x_4, y_4) = \frac{1.1679 - 0.2}{1.1679 + 0.2} = 0.7076.$$

Теперь уточним значение по интерполяционной формуле (а можно этого и не делать, тогда погрешность метода будет больше):

$$y_4^u = y_3 + \frac{h}{24}(9f_4 + 19f_3 - 5f_2 + f_1) = 1.1311 + \frac{0.05}{24}(9 * 0.7076 + 19 * 0.7658 - 5 * 0.8321 + 0.9089) = 1.1679.$$

Так как в качестве нового значения функции принято скорректированное, то *обязательно* следует пересчитать значение производной. В нашем случае модуль разности экстраполяционной и интерполяционной формул меньше ϵ , что позволяет продолжить вычисления с тем же шагом.

Вопросы для самоконтроля

1. Сформулируйте задачу Коши для обыкновенных дифференциальных уравнений первого порядка.
2. Что является решением дифференциального уравнения: а) в высшей математике, б) в прикладной математике?
3. Какие методы решения дифференциальных уравнений называются одношаговыми, многошаговыми? Приведите примеры.
4. Сравните решения, полученные на первом, втором шаге методами Эйлера, Рунге-Кутты и разложением в ряд Тейлора (трудоемкость, погрешность...).
5. Как оценить погрешность применяемого метода? Как ее уменьшить?
6. Сравните одношаговые и многошаговые методы решения дифференциальных уравнений, указав достоинства и недостатки первых и вторых.
7. Что такое экстраполяционные и интерполяционные методы (формулы) Адамса?
8. Можно ли применять: а) только экстраполяционные методы Адамса, б) только интерполяционные?
9. Можно ли использовать: а) многошаговые методы без одношаговых; б) одношаговые методы без многошаговых?
10. При решении дифференциального уравнения методом Адамса на 27-м шаге необходимо сменить шаг. Как это сделать?

7. ИНТЕРПОЛИРОВАНИЕ И ПРИБЛИЖЕНИЕ ФУНКЦИЙ

7.1. Задача интерполирования и аппроксимации функций

Задача интерполирования состоит в том, чтобы по значениям функции $f(x)$ в нескольких точках отрезка восстановить ее значения в остальных точках данного отрезка. Разумеется, такая постановка задачи допускает сколь угодно много решений.

Задача интерполирования возникает, например, в том случае, когда известны результаты измерений $y_k = f(x_k)$ некоторой физической величины $f(x)$ в точках $x_k, k = 0, 1, \dots, n$ и требуется определить ее значение в других точках. Интерполирование используется также при необходимости сгущения таблиц, когда вычисление значений $f(x)$ по точным формулам трудоемко.

Иногда возникает необходимость *приближенной замены (аппроксимации)* данной функции (обычно заданной таблицей) другими функциями, которые легче вычислить. При обработке эмпирических (экспериментальных) зависимостей, результаты обычно представлены в табличном или графическом виде. Задача заключается в аналитическом представлении искомой функциональной зависимости, то есть в подборе формулы, корректно описывающей экспериментальные данные.

7.2. Интерполирование алгебраическими многочленами

Пусть функциональная зависимость задана таблицей $y_0 = f(x_0); \dots, y_1 = f(x_1); \dots, y_n = f(x_n)$. Обычно задача интерполирования формулируется так: найти многочлен $P(x) = P_n(x)$ степени не выше n , значения которого в точках $x_i (i = 0, 1, 2, \dots, n)$ совпадают со значениями данной функции, то есть $P(x_i) = y_i$.

Геометрически это означает, что нужно найти алгебраическую кривую вида

$$y = a_0 + a_1x + \dots + a_nx^n, \quad (7.1)$$

проходящую через заданную систему точек $M_i(x_i, y_i)$ (см. рис. 7.1). Многочлен $P(x)$ называется *интерполяционным многочленом*. Точки $x_i (i = 0, 1, 2, \dots, n)$ называются *узлами интерполяции*.

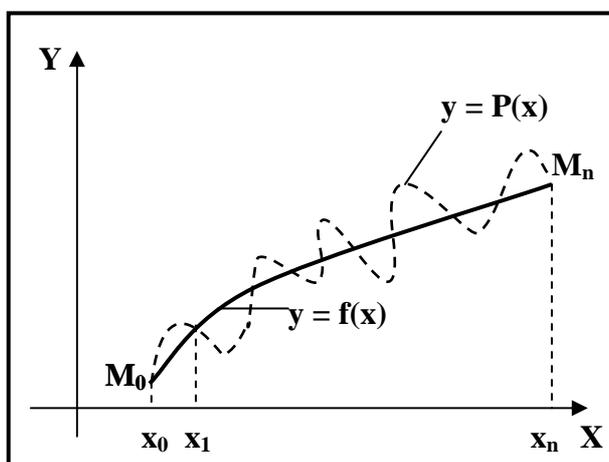


Рис. 7.1. Интерполирование алгебраическим многочленом

Для любой непрерывной функции $f(x)$ сформулированная задача имеет единственное решение. Действительно, для отыскания коэффициентов $a_0, a_1, a_2, \dots, a_n$ получаем систему линейных уравнений

$$a_0 + a_1x_i + a_2x_i^2 + \dots + a_nx_i^n = f(x_i)_{i=0, \overline{n}}, \quad (7.2)$$

определитель которой (определитель Вандермонда) отличен от нуля, если среди точек x_i ($i = 0, 1, 2, \dots, n$) нет совпадающих.

Решение системы (7.2) можно записать различным образом. Однако наиболее употребительна запись интерполяционного многочлена в форме Лагранжа и в форме Ньютона.

Запишем без вывода *интерполяционный многочлен Лагранжа*:

$$L_n(x) = \sum_{k=0}^n \frac{\prod_{j \neq k} (x - x_j)}{\prod_{j \neq k} (x_k - x_j)} f(x_k). \quad (7.3)$$

Нетрудно заметить, что старшая степень аргумента x в многочлене Лагранжа равна n . Кроме этого, несложно показать, что в узловых точках значение интерполяционного многочлена Лагранжа соответствует заданным значениям $f(x_i)$.

7.3. Интерполяционная формула Ньютона

Интерполяционная формула Ньютона позволяет выразить интерполяционный многочлен $P_n(x)$ через значение $f(x)$ в одном из узлов и через разделенные разности функции $f(x)$, построенные по узлам x_0, x_1, \dots, x_n . Эта формула является разностным аналогом формулы Тейлора:

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x - x_0)^2}{2!} f''(x_0) + \dots \quad (7.4)$$

Прежде чем приводить формулу Ньютона, рассмотрим сведения о разделенных разностях. Пусть в узлах $x_k \in [a, b]$, $k = 0, 1, \dots, n$ известны значения функции $f(x)$. Предполагаем, что среди точек x_k , $k = 0, 1, \dots, n$ нет совпадающих. Тогда разделенными разностями первого порядка называются отношения

$$f(x_i, x_j) = \frac{f(x_j) - f(x_i)}{x_j - x_i}, \quad i, j = 0, 1, \dots, n; \quad i \neq j. \quad (7.5)$$

Будем рассматривать разделенные разности, составленные по соседним узлам, то есть выражения $f(x_0, x_1)$, $f(x_1, x_2), \dots, f(x_{n-1}, x_n)$. По этим разделенным разностям первого порядка можно построить разделенные разности второго порядка:

$$\begin{aligned}
f(x_0, x_1, x_2) &= \frac{f(x_1, x_2) - f(x_0, x_1)}{x_2 - x_0}; \\
f(x_1, x_2, x_3) &= \frac{f(x_2, x_3) - f(x_1, x_2)}{x_3 - x_1}; \\
&\dots \\
f(x_{n-2}, x_{n-1}, x_n) &= \frac{f(x_{n-1}, x_n) - f(x_{n-2}, x_{n-1})}{x_n - x_{n-2}}.
\end{aligned} \tag{7.6}$$

Аналогично определяются разности более высокого порядка. То есть пусть известны разделенные разности k -го порядка $f(x_j, x_{j+1}, \dots, x_{j+k})$, $f(x_{j+1}, x_{j+2}, \dots, x_{j+k+1})$, тогда разделенная разность $k+1$ -го порядка определяется как

$$\begin{aligned}
f(x_j, x_{j+1}, \dots, x_{j+k+1}) &= \\
&= \frac{f(x_{j+1}, x_{j+2}, \dots, x_{j+k+1}) - f(x_j, x_{j+1}, \dots, x_{j+k})}{x_{j+k+1} - x_j}.
\end{aligned} \tag{7.7}$$

Интерполяционным многочленом Ньютона называется многочлен

$$\begin{aligned}
P_n(x) &= f(x_0) + (x - x_0)f(x_0, x_1) + (x - x_0)(x - x_1)f(x_0, x_1, x_2) + \dots \\
&\dots + (x - x_0)(x - x_1) \dots (x - x_{n-1})f(x_0, x_1, \dots, x_n).
\end{aligned} \tag{7.8}$$

Показано, что интерполяционный многочлен Лагранжа (7.3) совпадает с интерполяционным многочленом Ньютона (7.8).

На практике часто возникает следующий вопрос: будет ли стремиться к нулю погрешность интерполирования $f(x) - L_n(x)$, если число узлов n неограниченно увеличивать:

1. Свойства сходимости или расходимости интерполяционного процесса зависят как от выбора последовательности сеток, так и от гладкости функции $f(x)$.

2. Известны примеры несложных функций, для которых интерполяционный процесс расходится.

Так последовательность интерполяционных многочленов, построенных для непрерывной функции $f(x) = |x|$ по равноотстоящим узлам на отрезке $[-1, 1]$, не сходится к функции $|x|$ ни в одной точке отрезка $[-1, 1]$, кроме точек $-1, 0, 1$. На рис. 7.2 в качестве иллюстрации изображен график многочлена $L_9(x)$ при $0 \leq x \leq 1$, построенного для функции $|x|$ по равноотстоящим узлам на отрезке $[-1, 1]$.

3. Чтобы избежать этих некорректностей, в практике вычислений обычно избегают пользоваться интерполяционными многочленами высокой степени.

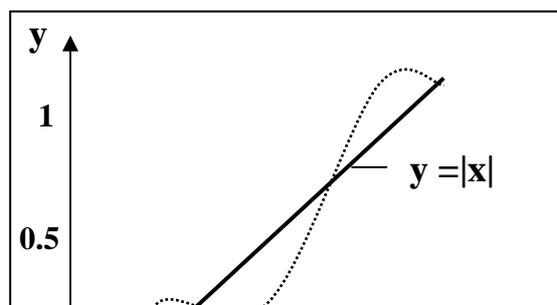


Рис. 7.2. Сходимость интерполяционных многочленов

Замечания

1. В формуле (7.8) не предполагалось, что узлы x_0, x_1, \dots, x_n расположены в каком-то определенном порядке. Поэтому роль точки x_0 в формуле (7.8) может играть любая из точек x_0, x_1, \dots, x_n . Соответствующее множество интерполяционных формул можно получить из (7.8), перенумеровав узлы. Например, тот же самый многочлен $P_n(x)$ можно представить в виде

$$\begin{aligned} P_n(x) = & f(x_n) + (x - x_n)f(x_n, x_{n-1}) + \\ & + (x - x_n)(x - x_{n-1})f(x_n, x_{n-1}, x_{n-2}) + \dots \\ & \dots + (x - x_n)(x - x_{n-1}) \dots (x - x_1)f(x_n, \dots, x_0). \end{aligned} \quad (7.9)$$

2. Если $x_0 < x_1 < x_2 < \dots < x_n$, то (7.8) называется формулой интерполирования вперед, а (7.9) - формулой интерполирования назад.
3. Интерполяционную формулу Ньютона удобнее применять в том случае, когда интерполируется одна и та же функция $f(x)$, но число узлов интерполяции постепенно увеличивается. Если узлы интерполяции фиксированы и интерполируется не одна, а несколько функций, то удобнее пользоваться формулой Лагранжа.

7.4. Задача обратного интерполирования

Пусть функция $y = f(x)$ задана таблично. Задача обратного интерполирования заключается в том, чтобы по заданному значению функции y определить соответствующее значение аргумента x .

Для случая неравноотстоящих значений аргумента x_0, x_1, \dots, x_n задача может быть непосредственно решена с помощью интерполяционного многочлена Лагранжа. В этом случае достаточно принять переменную y за независимую и написать формулу (аналог выражения (7.3)), выражающую x как функцию y :

$$x = \sum_{k=0}^n \frac{\prod_{j \neq k} (y - y_j)}{\prod_{j \neq k} (y_k - y_j)} x_k. \quad (7.10)$$

Можно также, считая y аргументом, использовать формулу Ньютона:

$$x = x_0 + (y - y_0)f(y_0, y_1) + (y - y_0)(y - y_1)f(y_0, y_1, y_2) + \dots \\ \dots + (y - y_0)(y - y_1) \dots (y - y_{n-1})f(y_0, y_1, \dots, y_{n-1}, y_n). \quad (7.11)$$

Замечание. Обратное интерполирование корректно только для взаимно однозначных функций.

Пример 7.1. Исходная функция $y = f(x)$ задана табл. 7.1:

Таблица 7.1

x	10	15	17	20
y	3	7	11	17

Необходимо вычислить значение функции y при $x = 12$.

Кроме того, необходимо вычислить значение x , для которого $y = 10$.

Решение. В качестве примера задачу прямого интерполирования в начале таблицы с неравноотстоящими узлами решим по формулам Ньютона (7.8); для обратного интерполирования применим формулу Лагранжа (7.10).

$$y(12) = f(x_0) + (x - x_0)f(x_0, x_1) + (x - x_0)(x - x_1)f(x_0, x_1, x_2) + \\ + (x - x_0)(x - x_1)(x - x_2)f(x_0, x_1, x_2, x_3) = 3 + 2 \cdot 0.8 + \\ + 2 \cdot (-3) \cdot 0.02857 + 2 \cdot (-3) \cdot (-5) \cdot (-0.002857) = 4.3429.$$

$$x(10) = 10 \cdot 3 \cdot (-1) \cdot (-7) / [(-4) \cdot (-8) \cdot (-14)] + 15 \cdot 7 \cdot (-1) \cdot (-7) / [4 \cdot (-4) \cdot (-10)] + \\ + 17 \cdot 7 \cdot 3 \cdot (-7) / [8 \cdot 4 \cdot (-6)] + 20 \cdot 7 \cdot 4 \cdot 1 / [14 \cdot 10 \cdot 6] = 16.641.$$

7.5. Оценка параметров эмпирических формул

При эмпирическом (экспериментальном) изучении функциональной зависимости одной величины y от другой x производят ряд измерений величины y при различных значениях величины x . Полученные результаты можно представить в виде таблицы, графика:

X	x_1	x_2	...	x_n
Y	y_1	y_2	...	y_n

Задача заключается в аналитическом представлении искомой функциональной зависимости, то есть в подборе функции, описывающей результаты эксперимента.

Особенность задачи состоит в том, что наличие случайных ошибок измерений делает неразумным подбор такой формулы, которая точно описывала бы все опытные значения, то есть график искомой функции не должен проходить через все экспериментальные точки. Эмпирическую формулу обычно выбирают из формул определенного типа:

$$y = ax + b; \quad y = ae^{bx} + c; \quad y = a + h \sin(\omega x + \varphi). \quad (7.12)$$

Таким образом, задача сводится к определению параметров a, b, c, \dots формулы, в то время как вид формулы известен заранее из каких-либо теоретических соображений или из соображения простоты аналитического представления эмпирического материала. Пусть выбранная эмпирическая зависимость имеет вид

$$y = f(x, a_0, a_1, \dots, a_n) \quad (7.13)$$

с явным указанием всех параметров, подлежащих определению. Эти параметры $a_0, a_1, a_2, \dots, a_n$ нельзя определить точно по эмпирическим значениям функции $y_0, y_1, y_2, \dots, y_k$, так как последние содержат случайные ошибки.

Таким образом, речь может идти только о получении достаточно хороших оценок искомых параметров. *Метод наименьших квадратов* (МНК) позволяет получить несмещенные и состоятельные оценки всех параметров $a_0, a_1, a_2, \dots, a_n$.

7.6. Метод наименьших квадратов (МНК)

Дальнейшие рассуждения будем проводить в предположении, что все измерения значений функции $y_0, y_1, y_2, \dots, y_n$ произведены с одинаковой точностью. Тогда оценка параметров $a_0, a_1, a_2, \dots, a_n$ определяется из условия минимума суммы квадратов отклонений измеренных значений y_k от расчетных $f(x_k; a_0, a_1, a_2, \dots, a_n)$:

$$S = \sum_{N=1}^k [y_N - f(x_N; a_0, a_1, \dots, a_n)]^2. \quad (7.14)$$

Отыскание же значений параметров $a_0, a_1, a_2, \dots, a_n$, которые доставляют *min* значение функции

$$S = S(a_0, a_1, \dots, a_n), \quad (7.15)$$

сводится к решению системы уравнений

$$\frac{\partial S}{\partial a_0} = 0; \quad \frac{\partial S}{\partial a_1} = 0; \dots; \quad \frac{\partial S}{\partial a_n} = 0. \quad (7.16)$$

Наиболее распространен способ выбора функции $f(x_k; a_0, a_1, a_2, \dots, a_n)$ в виде линейной комбинации:

$$f = a_0 \Phi_0 + a_1 \Phi_1 + \dots + a_n \Phi_n. \quad (7.17)$$

Здесь $\varphi_0, \varphi_1, \dots, \varphi_n$ – базисные функции (известные); $n \ll k$; $a_0, a_1, a_2, \dots, a_n$ – коэффициенты, определяемые методом наименьших квадратов. Запишем в явном виде условие (7.16), учитывая выражение (7.17):

$$\begin{aligned} \frac{\partial S}{\partial a_0} &= 2 \sum_{i=0}^k [a_0 \varphi_0(x_i) + a_1 \varphi_1(x_i) + \dots + a_n \varphi_n(x_i) - y_i] \varphi_0(x_i) = 0; \\ \frac{\partial S}{\partial a_1} &= 2 \sum_{i=0}^k [a_0 \varphi_0(x_i) + a_1 \varphi_1(x_i) + \dots + a_n \varphi_n(x_i) - y_i] \varphi_1(x_i) = 0; \\ &\dots \\ \frac{\partial S}{\partial a_n} &= 2 \sum_{i=0}^k [a_0 \varphi_0(x_i) + a_1 \varphi_1(x_i) + \dots + a_n \varphi_n(x_i) - y_i] \varphi_n(x_i) = 0. \end{aligned} \quad (7.18)$$

Из системы линейных уравнений (7.18) определяются все коэффициенты a_k . Система (7.18) называется системой нормальных уравнений, матрица которой имеет вид

$$\begin{vmatrix} (\varphi_0, \varphi_0) & (\varphi_0, \varphi_1) & \dots & (\varphi_0, \varphi_n) \\ (\varphi_0, \varphi_1) & (\varphi_1, \varphi_1) & \dots & (\varphi_1, \varphi_n) \\ \dots & \dots & \dots & \dots \\ (\varphi_0, \varphi_n) & (\varphi_1, \varphi_n) & \dots & (\varphi_n, \varphi_n) \end{vmatrix}. \quad (7.19)$$

Здесь

$$(\varphi_j, \varphi_k) = \sum_{i=0}^k \varphi_j(x_i) \varphi_k(x_i). \quad (7.20)$$

Матрица (7.19) называется матрицей Грамма. Расширенную матрицу получим добавлением справа столбца свободных членов:

$$\begin{vmatrix} (\varphi_0, Y) \\ (\varphi_1, Y) \\ \dots \\ (\varphi_n, Y) \end{vmatrix}, \quad (7.21),$$

где
$$(\varphi_j, Y) = \sum_{i=0}^k \varphi_j(x_i) Y_i. \quad (7.22)$$

Основные свойства матрицы Грамма

1. Матрица симметрична относительно главной диагонали, то есть

$$a_{ij} = a_{ji}.$$

2. Матрица является положительно определенной. Следовательно, при решении методом Гаусса можно воспользоваться схемой единственного деления.

3. Определитель матрицы будет отличен от нуля, если в качестве базиса выбраны линейно независимые функции $\Phi_j(x)$; в этом случае система (7.18) имеет единственное решение.

В качестве базисных можно выбрать линейно независимые степенные функции

$$\Phi_0(x) = x^0 = 1; \quad \Phi_1(x) = x^1 = x; \quad \dots; \quad \Phi_n(x) = x^n. \quad (7.23)$$

Следует учесть, что $n \ll k$. Тогда для этих функций расширенная матрица Грамма примет вид

$$\begin{bmatrix} k+1 & \sum_{i=0}^k x_i & \sum_{i=0}^k x_i^2 & \dots & \sum_{i=0}^k x_i^n & \sum_{i=0}^k y_i \\ \sum_{i=0}^k x_i & \sum_{i=0}^k x_i^2 & \sum_{i=0}^k x_i^3 & \dots & \sum_{i=0}^k x_i^{n+1} & \sum_{i=0}^k x_i y_i \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \sum_{i=0}^k x_i^n & \sum_{i=0}^k x_i^{n+1} & \dots & \dots & \sum_{i=0}^k x_i^{2n} & \sum_{i=0}^k x_i^n y_i \end{bmatrix}. \quad (7.24)$$

Если выбрать $n = k$, то на основании единственности интерполяционного полинома получим функцию $\Phi(x)$, совпадающую с каноническим интерполяционным полиномом степени k . При этом аппроксимирующая кривая пройдет через все экспериментальные точки, и функция S будет равна нулю.

Пример 7.2. Исходная функция $y = f(x)$ задана в виде табл. 7.2:

Таблица 7.2

x	10	15	17	20
y	3	7	11	17

Аппроксимируем экспериментальные данные линейной либо квадратичной функцией. Методом наименьших квадратов необходимо уточнить коэффициенты аппроксимирующего полинома.

Решение

1. При линейной аппроксимации исходную зависимость представим в виде $f = a_0 \cdot \Phi_0 + a_1 \cdot \Phi_1$, где $\Phi_0 = x^0 = 1$; $\Phi_1 = x^1 = x$. Методом наименьших квадратов определим a_0 и a_1 . Расширенная матрица Грама в нашем случае имеет вид

$$\begin{bmatrix} 4 & 62 & 38 \\ 62 & 1014 & 662 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 15.5 & 9.5 \\ 0 & 53 & 73 \end{bmatrix}; \quad a_1 = 1.3774; \quad a_0 = -11.8491.$$

Таким образом, аппроксимирующая функция равна

$$f = -11.8491 + 1.3774x.$$

Оценим погрешность формулы, и результаты этой оценки сведем в табл. 7.3:

Таблица 7.3

x	y	f	y - f	y-f / y
10	3	1.9249	1.0751	0.3584
15	7	8.8119	-1.8119	0.2588
17	11	11.5667	-0.5667	0.0515
20	17	15.6989	1.3011	0.07654

Для нашей линейной функции $S_1 = 6.4528$.

2. Решим ту же задачу, аппроксимировав эмпирические данные полиномом второй степени: $f = a_0 + a_1x + a_2x^2$.

Матрица Грамма в этом случае имеет вид

$$\begin{bmatrix} 4 & 62 & 1014 & 38 \\ 62 & 1014 & 17288 & 662 \\ 1014 & 17288 & 304146 & 11854 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 15.5 & 253.5 & 9.5 \\ 0 & 53 & 1571 & 73 \\ 0 & 1571 & 47097 & 2221 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 15.5 & 253.5 & 9.5 \\ 0 & 1 & 29.6415 & 1.3774 \\ 0 & 530.2035 & 57.1046 \end{bmatrix};$$

$$a_2 = 0.1077; \quad a_1 = -1.8151; \quad a_0 = 10.3321.$$

Все результаты сведены в табл. 7.4.

Таблица 7.4

x	y	f	y - f	y-f / y
10	3	2.9511	0.0489	0.0163
15	7	7.3381	-0.3381	0.0483
17	11	10.6007	0.3993	0.0363
20	17	17.1101	-0.1101	0.0065

$$S_2 = 0.2883.$$

Обсуждение результатов

1. Аппроксимировав эмпирические результаты более простой функцией (линейной), мы получили погрешность в различных узловых точках, лежащую в пределах от 5 до 35 %.
2. Более сложная формула квадратичной интерполяции обеспечивает погрешность не более 5 %.
3. Косвенную оценку погрешности можно провести, сравнив значения S_1 и S_2 .
4. Матрица Грама для полинома второй, третьей степени имеет простой вид и может быть решена, например, методом Гаусса.

Вопросы для самоконтроля

1. В чем заключается задача интерполирования и аппроксимации?
2. Запишите интерполяционные формулы Лагранжа и Ньютона.
3. Какие требования предъявляются
 - а) к интерполяционным полиномам;
 - б) к аппроксимационным полиномам?
- 4.
5. Что такое разделенные разности?
6. В каких случаях применяются формулы Ньютона для интерполирования
 - а) вперед,
 - б) назад?
7. Что можно сказать о сходимости интерполяционных полиномов?
8. Что такое обратное интерполирование, при каких условиях оно возможно ?
9. В чем заключается идея метода наименьших квадратов?
10. Что такое матрица Грама, каковы ее свойства?
11. Что такое базисные функции? Можно ли в качестве базисных функций выбрать
 - а) линейно независимые функции;
 - б) линейно зависимые функции?

ЛИТЕРАТУРА

1. Амосов А.А., Дубинский Ю.А., Копченова Н.В. Вычислительные методы для инженеров. – М.: Высшая школа, 1994. – 544 с.
2. Бахвалов Н.В., Жидков Н.П., Кобельков Г.М. Численные методы. – М.: Лаборатория Базовых Знаний, 2002. – 632 с.
3. Бобенко А.В., Федоров М.Д. Параллельные вычисления в задаче факторизации обратной матрицы клетками меньших размерностей//Научная сессия ТУСУР: Материалы докладов Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых. Томск, 5 – 8 мая 2008 г., – Томск: Изд-во «В-Спектр», 2008. – Ч.5. – С.175 – 178.
4. Вержбицкий В.М. Численные методы. Линейная алгебра и нелинейные уравнения. – М.: Высшая школа, 2000. – 266 с.
5. Демидович Б.П., Марон И.А. Основы вычислительной математики. - М.: Наука, 1970. – 664 с.
6. Истигечева Е.В., Федоров М.Д. Факторизация обратной матрицы клетками меньших размерностей//Научная сессия ТУСУР: Материалы докладов Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых. Томск, 3 – 7 мая 2007 г., – Томск: Изд-во «В-Спектр», 2007. – Ч.5. – С.237 – 239.
7. Калиткин Н.Н. Численные методы. – М.: Наука, 1978. – 512 с.
8. Марчук Г.И. Методы вычислительной математики М.: Наука, 1989. – 608 с.
9. На Ц. Вычислительные методы решения прикладных граничных задач. – М.: Мир, 1982. – 294 с.
10. Рябенский В.С. Введение в вычислительную математику – М.: Физматлит, 2000. – 294 с.
11. Самарский А.А. Введение в численные методы. – М.: Наука, 1997. – 234 с.
12. Тихонов А.Н., Васильева А.Б., Свешников А.Г. Дифференциальные уравнения. – М.: Наука. Физматлит, 1998. – 232 с.
13. Фадеев А.К., Фадеева В.Н. Вычислительные методы линейной алгебры. – СПб.: Лань, 2002. – 736 с.