СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
SIBERIAN FEDERAL UNIVERSITY

В. Е. Зализняк

ЧИСЛЕННЫЕ МЕТОДЫ

ОСНОВЫ НАУЧНЫХ ВЫЧИСЛЕНИЙ

УЧЕБНИК И ПРАКТИКУМ
ДЛЯ АКАДЕМИЧЕСКОГО БАКАЛАВРИАТА

2-е издание, переработанное и дополненное

Рекомендовано Учебно-методическим отделом высшего образования в качестве учебника для студентов высших учебных заведений, обучающихся по физико-техническим направлениям и специальностям

Допущено УМО по классическому университетскому образованию в качестве учебного пособия для студентов высших учебных заведений, обучающихся по специальности (направлению) подготовки ВПО 010501 (010500.62) «Прикладная математика и информатика» (ОПД.Ф.09 — Численные методы)

Москва ▪ Юрайт ▪ 2015

УДК 51
ББК 22.193я73
З22

***Автор*:**

Зализняк Виктор Евгеньевич — кандидат физико-математических наук, доктор философии (PhD, Aerospace Engineering), доцент кафедры математического моделирования и процессов управления Института математики Сибирского федерального университета.

Зализняк, В. Е.

З22 Численные методы. Основы научных вычислений : учебник и практикум для академического бакалавриата / В. Е. Зализняк. — 2-е изд., перераб. и доп. — М. : Издательство Юрайт, 2015. — 356 с. — Серия : Бакалавр. Академический курс.

ISBN 978-5-9916-4895-0

Учебник подготовлен в соответствии с Федеральным государственным образовательным стандартом высшего образования четвертого поколения по дисциплине «Численные методы» для студентов высших учебных заведений.

В книге рассмотрены основы классических численных методов. Наряду с теоретическими аспектами вычислений в нем приводится краткий обзор библиотек программ, широко используемых в научных исследованиях.

Для студентов вузов. Книга может быть полезна аспирантам в качестве справочника по численным методам.

УДК 51
ББК 22.193я73

ISBN 978-5-9916-4895-0

Оглавление

Предисловие

Развитие современной науки и технологии в значительной степени основано на компьютерном моделировании. Для того чтобы освоить основные методы и принципы компьютерного моделирования, студенты прежде всего должны овладеть основами классических численных методов, которые и составляют предмет этой книги. В настоящее время различное программное обеспечение широко используется на практике. Его грамотное и эффективное применение невозможно без глубокого понимания численных методик, лежащих в основе этих программ. В этой книге, наряду с теоретическими аспектами вычислений, приводится краткий обзор библиотек программ, широко используемых в научных исследованиях.

Эта книга предназначена для использования в курсе численных методов для студентов физико-технических специальностей, а также она может быть полезна аспирантам в качестве справочника по численным методам. Объем этой книги рассчитан на изучение в течение двух семестров. Уровень изложения материала предполагает, что читатель освоил курсы математического анализа (два семестра), линейной алгебры (один семестр) и дифференциальных уравнений (один семестр).

В небольшой по объему книге невозможно рассмотреть все проблемы классического численного анализа. Поэтому были выбраны ниболее важные, с моей точки зрения, темы, составляющие основу прикладных численных методов.

Виктор Зализняк

Глава 1
ВОЗНИКНОВЕНИЕ ОШИБОК ВЫЧИСЛЕНИЙ ПРИ ВЫПОЛНЕНИИ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ НА КОМПЬЮТЕРЕ

Компьютерная арифметика отличается от обычной арифметики. В традиционной математике числа могут иметь бесконечное число цифр. В компьютерной арифметике каждое допустимое число имеет конечное число цифр. Входные данные и результаты арифметических операций помещаются в оперативную память компьютера в определенной форме.

Вещественные числа представляются в следующей стандартной форме (*форма с плавающей точкой*):

$$x = \pm b^c m, \tag{1.1}$$

здесь x — *вещественное число*, b — основание системы счисления (обычно используются основания 2, 8 и 16), c называется характеристикой, m_b — мантиссой (она определяет дробную часть числа), а ($\pm$) обозначает знак числа x. Мантисса числа $x \neq 0$ удовлетворяет условию

$$\frac{1}{b} < m_b < 1,$$

что обеспечивает единственность представления (1.1). Любое вещественное число можно представить в форме с плавающей точкой, и это может быть сделано в любой системе счисления с основанием $b > 0$. Однако некоторые вещественные числа (например, иррациональные) невозможно записать в оперативную память компьютера точно. Все числа, которые могут быть помещены в оперативную память, представляются в определенном виде. При этом их характеристики ограничены: $c_m \leqslant c \leqslant c_p$, а мантиссы имеют вид конечных дробей по основанию b:

$$m_b = \frac{a_1}{b} + \frac{a_2}{b^2} + \dots + \frac{a_k}{b^k},$$

где коэффициенты a_n удовлетворяют условиям

$$1 \leqslant a_1 \leqslant b - 1, \quad 0 \leqslant a_n \leqslant b - 1, \quad n = 2, \dots, k.$$

Пример 1.1 (двоичное представление вещественного числа)
Рассмотрим вещественное число $x = 8.75$. Представим это число в двоичной форме ($b = 2$). Сначала мы перепишем число x в виде $0.546875 \cdot 2^4$, тогда характеристика c, согласно (1.1), равна 4 (100 в двоичной форме). Дробная часть числа представляется в виде

$$0.546875 = \frac{1}{2} + \frac{1}{32} + \frac{1}{64},$$

тогда коэффициенты a_n равны

a_1	a_2	a_3	a_4	a_5	a_6	a_7	...	a_k
1	0	0	0	1	1	0	...	0

Для простоты числа, которые могут быть помещены в оперативную память, мы будем называть *машинными числами*, и они образуют конечное подмножество множества вещественных чисел. Для описания машинных чисел мы ввели параметры b, c_m, c_p и k, но использовать эти параметры на практике не очень удобно. Обычно для описания машинных чисел вводится другой набор параметров: b, ε_0, ε_1, и ε_∞. Эти параметры, как и предыдущие, зависят от конкретного компьютера и определяются следующим образом:

$$\varepsilon_0 = b^{c_m - 1}, \quad \varepsilon_1 = b^{1-k}, \quad \varepsilon_\infty = b^{c_p - 1}(1 - b^{-k}).$$

Эти параметры имеют следующие значения: ε_∞ — максимальное машинное число, ε_0 — минимальное машинное число и ε_1 есть минимальное расстояние между двумя машинными числами на интервале от 1 до b. Параметр ε_1 также называют относительной ошибкой определения единицы, так как все числа вида $1 + x$ из интервала $[1, 1 + \varepsilon_1]$ заменяются машинным числом 1 с ошибкой, не превосходящей ε_1. Следовательно, интервалы $(-\infty, -\varepsilon_\infty)$, $(-\varepsilon_0, 0)$, $(0, \varepsilon_0)$, $(1 - \varepsilon_1/\mathrm{b}, 1)$, $(1, 1 + \varepsilon_1)$ и $(\varepsilon_\infty, \infty)$ не содержат машинных чисел. Наряду со стандартной точностью существует возможность более точного представления машинных чисел. Это можно осуществить, увеличив число цифр в мантиссе и расширив область допустимых значений характеристик. Тогда машинные числа с повышенной точностью будут определяться некоторыми параметрами

$$b, c_m^*, c_p^*, k^*.$$

Например, определяющие параметры для PC представлены в табл. 1.1.

Таблица 1.1

Определяющие параметры для PC

	Одинарная точность	Двойная точность
b	2	2
c_m	–125	–1021
c_p	128	1024
k	24	53
ε_0	2^{-126}	2^{-1022}
ε_1	2^{-23}	2^{-52}
ε_∞	$2^{128}(1 - 2^{-24})$	$2^{1024}(1 - 2^{-53})$

Если число $x \in [-\varepsilon_\infty, \varepsilon_\infty]$, то оно заменяется машинным числом x_m. В процессе такой замены используется какой-нибудь способ приближения (усечение или округление) и в качестве x_m выбирается ближайшее к x число. В результате такой замены возникает ошибка $x - x_m$. Для $\varepsilon_0 \leqslant |x| \leqslant \varepsilon_\infty$ эта ошибка оценивается величиной $\varepsilon_1|x|$, т.е. $|x - x_m| \leqslant \varepsilon_1|x|$. Если $|x| \in (0, \varepsilon_0)$, тогда x_m может принимать одно из двух значений: 0 или ε_0, и понятно, что в обоих случаях $|x - x_m| \leqslant \varepsilon_0|x|$. Поэтому параметр ε_0 также называют абсолютной ошибкой определения нуля. Эти оценки можно объединить в следующее выражение:

$$x_m = x(1 + \alpha) + \beta, \quad |\alpha| \leqslant \varepsilon_1, \quad |\beta| \leqslant \varepsilon_0, \tag{1.2}$$

которое справедливо для всех $x \in [-\varepsilon_\infty, \varepsilon_\infty]$.

На компьютерах арифметические операции выполняются над машинными числами. Во многих случаях результат этих операций может не быть машинным числом или, другими словами, результат арифметической операции (машинный результат), будучи машинным числом, может не совпадать с точным результатом. Например, для точного представления частного двух машинных чисел a и b часто требуется бесконечное количество цифр мантиссы. Поэтому машинный результат отличается от точного результата деления a и b. Для дальнейшего анализа мы введем следующее правило компьютерной арифметики: машинный результат арифметической операции над двумя машинными числами можно представить как результат записи точного значения этой операции в память компьютера. Для того чтобы отличать машинные операции от обычных (точных) операций, мы будем обозначать их таким же знаком в угловых скобках. Тогда, согласно принятому выше правилу, можно записать

$$\begin{cases} a\langle + \rangle b = (a+b)_m, \\ a\langle - \rangle b = (a-b)_m, \\ a\langle \times \rangle b = (a\times b)_m, \\ a\langle / \rangle b = (a/b)_m. \end{cases} \tag{1.3}$$

Формулы (1.3) имеют следующий смысл: машинный результат арифметической операции есть ближайшее к точному результату машинное число. Машинный результат также зависит от типа операции и чисел a и b. Например, если точный результат лежит вне итервала допустимых в компьютере чисел (результат не принадлежит $[-\varepsilon_\infty, \varepsilon_\infty]$), то мы сталкиваемся с ситуацией, называемой переполнением порядка. Эту ситуацию компьютер трактует как фатальную ошибку, и дальнейшие вычисления прекращаются. В правильно работающей программе такие ситуации должны быть исключены. Если абсолютное значение точного результата меньше чем ε_0, то возникает ситуация, называемая исчезновением порядка, и хотя это не является фатальной ошибкой такая ситуация также нежелательна.

Объединяя выражения (1.2) и (1.3), мы получим формулы для моделирования ошибок арифметических операций

$$\begin{cases} a\langle + \rangle b = (a+b)_m = (a+b)(1+\alpha)+\beta, \\ a\langle - \rangle b = (a-b)_m = (a-b)(1+\alpha)+\beta, \\ a\langle \times \rangle b = (a\times b)_m = (ab)(1+\alpha)+\beta, \\ a\langle / \rangle b = (a/b)_m = (a/b)(1+\alpha)+\beta. \end{cases}$$

Эти формулы приводят к следующей оценке для ошибки, возникающей при выполнении операций с плавающей точкой:

$$\left|a\langle * \rangle b - (a*b)\right| \leqslant \varepsilon_0 + \varepsilon_1 |a*b|,$$

где $(*)$ обозначает любую из четырех арифметических операций. Эта оценка может использоваться для анализа поведения ошибки при реализации различных вычислительных алгоритмов. Это очень важный вопрос, так как часто возникает ситуация, когда при выполнении некоторого вычислительного процесса происходит накапливание ошибки, что приводит к значительному искажению конечного результата. Например, если предположить, что $\beta = 0$, тогда ошибка вычисления

$$\prod_{k=1}^{N} a_k$$

может быть оценена как

$$\left| a_1 \langle \times \rangle a_2 \langle \times \rangle \ldots \langle \times \rangle a_N - \prod_{k=1}^{N} a_k \right| \leqslant \varepsilon_1 (N-1) + \prod_{k=1}^{N} |a_k|.$$

Глава 2
РЕШЕНИЕ УРАВНЕНИЯ
$f(x) = 0$

Список обозначений

x	Точное значение корня ($f(x^*) \equiv 0$)
I	Интервал, содержащий единственный корень
x_k	Приближенное значение корня
ε_p	Требуемая точность вычисления приближенного значения корня
$f', f'', f^{(k)}$	Производные функции $f(x)$
$e_k = x_k - x^*$	Ошибка приближенного значения корня

Решение уравнения $f(x) = 0$ — часто встречающаяся проблема. На первый взгляд она выглядит довольно простой, но точное ее решение возможно, только если $f(x)$ есть полином степени $n \leqslant 4$. Под точным решением понимается некоторая процедура вычисления корня через параметры уравнения (например, для уравнения $ax^2 + bx + c = 0$).

Каждая нелинейная задача имеет свои особенности, поэтому численное решение уравнения $f(x) = 0$ невозможно без предварительного анализа функции $f(x)$. Например, дополнительные трудности возникают, когда график функции в точке корня касается оси x, как показано на рис. 2.1. Такие корни трудно обнаружить, потому что график функции $f(x)$ не пересекает ось x. Такое поведение характерно для функций, которые имеют кратные корни. Корень уравнения x^* имеет кратность m, если существует некоторая непрерывная функция $g(x)$ такая, что

1) $g(x^*) \neq 0$;

2) для каждого x $f(x) = (x - x^*)^m g(x)$.

Если $m = 1$, тогда x^* есть простой корень уравнения $f(x) = 0$. Мы ограничимся рассмотрением случая, когда $f(x)$ вещест-

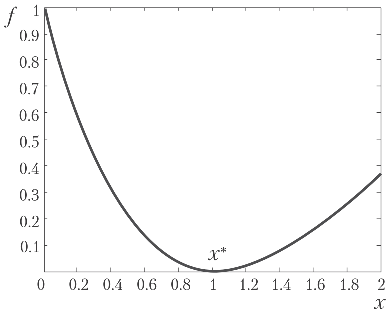

Рис. 2.1. **Ситуация когда $f(x^*) = f'(x^*) = 0$**

венная однозначная функция вещественного аргумента и уравнение $f(x) = 0$ имеет простой корень.

Основой для вычисления корня уравнения является *итерационный процесс*, или, другими словами, процесс последовательных приближений. Итерационный процесс имеет четыре стадии:

1) найдем интервал, который содержит единственный корень (локализация корня);

2) выберем начальное приближение x_0;

3) используя какую-либо итерационную процедуру k раз, получим набор приближений x_k, начиная с x_0;

4) определим, насколько близко каждое приближение x_k к точному значению корня. Если некоторое приближение находится в ε_p-окрестности x^*, тогда итерационный процесс завершен.

Последний пункт будет выполнен, только если

$$\lim_{k\to\infty} x_k = x^*,$$

т.е. приближения x_k сходятся к x^*. Поэтому основное внимание мы будем уделять условиям, которые обеспечивают сходимость итерационного процесса.

2.1. Отделение корней

Корень уравнения $f(x) = 0$ считается отделенным на отрезке $[a, b]$, если на этом отрезке уравнение $f(x) = 0$ не имеет других корней. Отделить корни — значит разбить всю область определения функции на отрезки, в каждом из которых содержится один корень.

Если построить график функции $f(x)$, точки пересечения графика с осью Ox дают значения корней и по графику легко определить два числа a и b, между которыми заключен каждый корень. Аналитически корни уравнения $f(x) = 0$ можно отделить, используя свойства функций. Приведем без доказательства теоремы, знание которых необходимо при отделении корней.

Теорема 2.1. Если функция $f(x)$ непрерывна и монотонна на отрезке $[a, b]$ и принимает на концах этого отрезка значения разных знаков, то внутри отрезка $[a, b]$ содержится единственный корень уравнения $f(x) = 0$.

Теорема 2.2. Если функция $f(x)$ непрерывна на отрезке $[a, b]$ и принимает на концах этого отрезка значения разных знаков, а производная $f'(x)$ сохраняет постоянный знак внутри отрезка, то внутри отрезка $[a, b]$ содержится единственный корень уравнения $f(x) = 0$.

На основании этих теорем можно предложить следующий алгоритм отделения корней:

1) найти производную $f'(x)$;

2) составить таблицу знаков $f(x_n)$, где x_n — точки, в которых производная меняет знаки (или близкие к ним точки);

3) определить отрезки, на концах которых функция принимает значения противоположных знаков. Внутри этих отрезков содержится по одному и только одному корню.

2.2. Метод деления отрезка пополам

Рассмотрим простой, но надежный метод для вычисления корней. Пусть некоторый отрезок $[a, b] = I$ содержит единственный корень. Стратегия этого метода состоит в том, что мы делим интервал I пополам и затем оставляем только половину этого интервала, содержащую корень. Очевидно, что половина, на которой функция $f(x)$ меняет знак, содержит корень. Это дает нам способ определить, какую половину оставить и затем использовать для дальнейших вычислений. Метод деления отрезка пополам можно представить в виде блок-схемы, показанной на рис. 2.2.

Таким образом, на каждом шаге вычислительного процесса x^* лежит внутри интервала $[ak, bk] \to [a_k, b_k]$, и как только длина этого интервала становится меньше чем ε_p, итерационный процесс завершается. В этом методе используется минимальная информация о функции $f(x)$ (знак $f(x)$). Поэтому требуется относительно большое число итераций, но сходимость этого метода гарантирована.

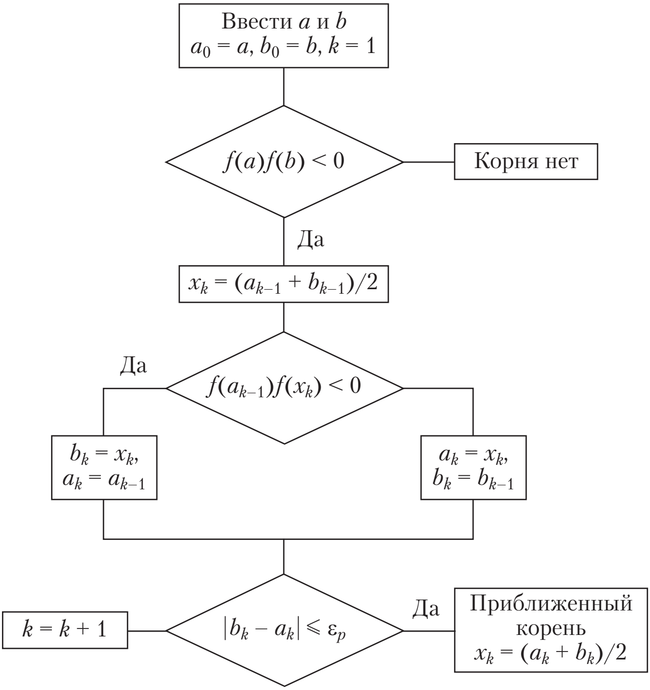

Рис. 2.2. **Блок-схема, представляющая метод деления отрезка пополам**

Оценим количество итераций, необходимых для вычисления корня. Легко видеть, что длина интервала $[a_k, b_k]$ уменьшается в два раза на каждом шаге итерационного процесса. Вычисления завершаются, когда

$$a_k - b_k = (a - b)/2^k \leqslant \varepsilon_p.$$

Для оценки числа итераций k выберем знак « = », тогда

$$k = \left[\frac{\ln(b-a) - \ln \varepsilon_p}{\ln 2}\right] + 1,$$

где $[y]$ означает целую часть y. Например, если $b - a = 1$ и $\varepsilon_p = 10^{-5}$, мы получим $k = 17$.

Пример 2.1 (метод деления отрезка пополам)

$$f(x) = \exp(-x) - \sin x = 0, \quad x^* \in [0; 1] = I, \quad \varepsilon_p = 10^{-5}.$$

Результаты расчетов представлены в табл. 2.1.

Таблица 2.1

k	x_k	$b_k - a_k$
1	0.5	5.00000e–01
2	0.75	2.50000e–01
3	0.625	1.25000e–01
4	0.5625	6.25000e–02
...	...	...
17	0.58854	7.62939e–06

2.3. Вычисление корней с использованием итерационных функций

В дальнейшем мы будем рассматривать другой подход для расчета корней. Уравнение $f(x) = 0$ может быть представлено в следующей эквивалентной форме:

$$x = g(x), \tag{2.1}$$

где $g(x)$ — некоторая итерационная функция и $f(x^*) = 0$ соответствует $x^* = g(x^*)$. Оказалось, что итерационные алгоритмы проще конструировать на основе эквивалентной формы (2.1). Итерационный процесс определяется следующим образом: зададим начальное значение x_0 и будем вычислять последующие приближения по формуле

$$x_{k+1} = g(x_k), \quad k = 0, 1, \ldots \tag{2.2}$$

На рис. 2.3 показана геометрическая интерпретация сходящегося итерационного процесса (2.2). Следующая теорема объясняет условия, которые обеспечивают сходимость итераций (2.2).

Теорема 2.3. Пусть $I = [a, b]$ ограниченный интервал и $g(x)$ функция, действующая из I в I. В дополнение $g(x)$ удовлетворяет условию Липшица

$$|g(y) - g(z)| \leqslant L|y - z|, \quad 0 \leqslant L \leqslant 1$$

для произвольных точек y и z в I. Пусть $x_0 \in I$ и $x_{k+1} = g(x_k)$, тогда последовательность $\{x_k\}$ сходится к единственному решению $x^* \in I$ уравнения $x = g(x)$.

На основе этой теоремы можно получить оценку ошибки k-го приближения, которая зависит от двух последовательных приближений и константы Липшица L. Так как $|x_{k+1} - x_k| = |g(x_k) - g(x_{k-1})|$, мы можем заключить из условия Липши-