Тема : Вводные понятия вычислительной математики

$\mathbf{1}^0$. Предмет и цели курса. $\mathbf{2}^0$. Определение численного метода. Общие свойства численных методов на примере приближенного решения квадратного уравнения. $\mathbf{3}^0$. Компьютер как средство реализации вычислительного алгоритма. Числа с плавающей точкой. Порядок машинного числа, его мантисса и разряды. $\mathbf{4}^0$. Округление числа до нормализованного числа с плавающей точкой. Абсолютная и относительная погрешности. $\mathbf{5}^0$. Абсолютная и относительная погрешности приближения чисел. $\mathbf{6}^0$. Зависимость численного результата от выбора алгоритма. $\mathbf{7}^0$. Понятие обусловленности задачи.

1^0. Предметом курса служат **вычислительные алгоритмы (ВА)**, применяемые к приближенному решению разнообразных задач.

Во всем множестве ВА выделяют несколько больших классов — в зависимости от характера задач, к которым они применяются. В курсе принята следующая классификация вычислительных алгоритмов:

1. Вычислительные методы линейной алгебры, матричные вычисления.

2. Задачи оптимизации, решение нелинейных уравнений и систем.

3. Интерполирование и численное дифференцирование функций.

4. Неполиномиальные численные приближения функций.

5. Численное интегрирование.

6. Численные методы решения задач для обыкновенных дифференциальных уравнений, разностные методы.

Цель курса: познакомить слушателей с наиболее характерными задачами вычислительной математики, относящимися к вышеперечисленным ее разделам, а также с численными методами решения этих задач, дав при этом навыки практического использования этих методов на примере простейших модельных задач.

2^0. Язык любого вычислительного алгоритма — это числа и арифметические действия

над ними. Будучи расположены в строго определенном порядке, эти числа и арифметические действия образуют **численный метод (ЧМ)**.

Простота языка вычислительного алгоритма позволяет реализовывать численные методы на компьютере (ЭВМ), что делает эти методы мощным и универсальным средством исследования.

Задачи, к которым применяются ЧМ формулируются обычно на стандартном математическом языке непрерывной математики (уравнения, функции, дифференцирование и интегрирование, интегральные преобразования Фурье, Лапласа и так далее).

Поэтому разработка ЧМ с необходимостью предполагает **замену исходной задачи другой**, близкой к ней, но сформулированной

уже в терминах чисел и арифметических операций над ними.

Способы такого рода замены могут быть совершенно разными, однако некоторые общие свойства присущи им всем. Поясним, что это за свойства на простом примере.

Задача. *Найти решение уравнения*

$$x^2 - a = 0, \quad \text{где} \quad a > 0. \tag{1}$$

Алгебраически решение этого уравнения записывается в виде формулы $x = \pm\sqrt{a}$, однако символ квадратного корня не дает способа вычисления этого корня из положительного числа a. Если мы хотим численно измерить величину $\sqrt{a}$, то нам необходимо указать предназначенный для этого ЧМ.

Возьмем $x_0 = 1$, а далее поступим следующим образом: будем последовательно вы-

числять значения x_1, x_2, x_3, ..., x_N с помощью рекуррентного равенства

$$x_n = \frac{1}{2}\left(x_{n-1} + \frac{a}{x_{n-1}}\right). \qquad (2)$$

Формула справа предполагает выполнение трех арифметических действий с заданными вещественными числами: деления, сложения и еще одного деления. В результате выполнения этих действий получаем число

слева. Следовательно, формула (2) задает нам некоторый численный метод.

Выполнив предписываемые равенством (2) действия $n = N$ раз, получим в итоге число x_N, которое и объявим приближенным решением исходной задачи, то есть положим

$$\sqrt{a} \approx x_N.$$

Точность этого приближения, то есть точность ЧМ (2), зависит от величины a и от значения N, которое мы выбираем.

Если мы хотим иметь решение задачи (1) в общем случае, то нужно убедиться, что при любом $a > 0$ найдется номер N, для которого величина x_N будет находиться в заданной окрестности вещественного числа $\sqrt{a}$. Докажем, что ЧМ (2) удовлетворяет этому условию.

Пусть величина ε_n определяется равенством

$$\frac{x_n}{\sqrt{a}} = 1 + \varepsilon_n. \qquad (3)$$

Разделив равенство (2) на $\sqrt{a}$ и подставив в него (3), получим

$$1 + \varepsilon_n = \frac{1}{2}\left(1 + \varepsilon_{n-1} + \frac{1}{1 + \varepsilon_{n-1}}\right).$$

Таким образом, для погрешностей ε_n при

$n = 1, 2, \ldots$, справедливы соотношения

$$\varepsilon_n = \frac{1}{2}\left(\varepsilon_{n-1} - 1 + \frac{1}{\varepsilon_{n-1}+1}\right) = \frac{1}{2}\left(\frac{\varepsilon_{n-1}^2}{1+\varepsilon_{n-1}}\right). \quad (4)$$

Учитывая, что $1 + \varepsilon_0 = \frac{1}{\sqrt{a}} > 0$, заключаем из равенства (4), что все погрешности ε_j, начиная с номера $j = 1$, строго положительны:

$$\varepsilon_1 > 0, \quad \varepsilon_2 > 0, \quad \varepsilon_3 > 0, \quad \ldots, \quad \varepsilon_n > 0.$$

В частности, получаем неравенства

$$0 < \frac{\varepsilon_{n-1}}{\varepsilon_{n-1}+1} < 1, \quad n = 2, 3, \ldots.$$

Подставляя их в (4), имеем далее

$$0 < \varepsilon_n < \frac{1}{2}\varepsilon_{n-1} < \frac{1}{4}\varepsilon_{n-2} < \cdots < \frac{\varepsilon_1}{2^{n-1}}.$$

Таким образом, последовательность ε_n с ростом n убывает быстрее геометрической прогрессии со знаменателем $\frac{1}{2}$. Следовательно,

$$0 < \frac{x_n}{\sqrt{a}} - 1 = \varepsilon_n \leq \frac{\varepsilon_1}{2^{n-1}} \quad \Rightarrow \lim_{n\to\infty} x_n = \sqrt{a}.$$

Несмотря на всю элементарность проведенных рассуждений, рассмотренный пример наглядно демонстрирует следующие три общие для всех вычислительных алгоритмов принципа:

1. Исходная непрерывная задача (1) заменяется другой, дискретной задачей (2), записанной как BA.

2. Дискретная задача (**2**) содержит новый параметр N, которого нет в (**1**). Этот параметр играет роль дополнительной дискретной переменной.

3. Выбрав N достаточно большим, всегда можно добиться, чтобы результат решения задачи (**2**) — число x_N — был сколь угодно близок к решению $\sqrt{a}$ исходной задачи.

Есть еще одна общая черта вычислительных алгоритмов, на которую по необходимости **приходится** обращать внимание, состоит в следующем.

При исследовании сходимости вычислительного алгоритма (2) **по умолчанию предполагалось**, что все операции в (2) можно реализовать абсолютно точно. Однако в реальности это не так.

Оперировать с представлением вещественного числа в виде бесконечной десятичной дроби не получится. Вычисления всегда производятся с *конечным* числом десятичных знаков, и *точность результата алгоритма не может быть выше точности любого разового расчета*.

Важно если не установить, то хотя бы оценить, как соотносится результирующая точ-

ность ВА с точностью производимых в процессе реализации разовых расчетов, и не будут ли ошибки округления, накапливаясь, лишать результат вычислительного алгоритма всякой математической ценности.

$\mathbf{3^0}$. Как видно из самого определения, реализация любого из численных методов связана с последовательным выполнением целого ряда арифметических операций (иногда очень большого количества операций).

В этой связи возникает проблема **средств реализации** численного метода.

Исторически изобретенный людьми арсенал средств для операций с числами весьма обширен и разнообразен. Итогом многолетней эволюции приборов для осуществления вычислений в современном мире служат компьютеры (электронно вычислительные машины, ЭВМ).

Уместно отметить, что с момента появления первых ЭВМ (50-е годы двадцатого века) менее чем за 50 лет скорость выполнения арифметических операций на компьютере возросла от 0.1 операции в секунду при ручном счете до 10^{12} операций/сек на серийной ЭВМ. Этот взрывной рост технических возможностей привел к появлению качественно новых социальных технологий.

Когда мы говорим о реализации ЧМ на ЭВМ, нам прежде всего необходимо разобраться **как именно представляются числа в современном компьютере**, а также каким образом над ними производятся "компьютерные" арифметические операции.

Наиболее распространенная форма представления вещественных чисел в компьютере — *числа с плавающей точкой (ЧПТ)*.

Множество $\mathbb{F}$ чисел с плавающей точкой в существующих стандартах машинной арифметики определяется следующими четырьмя числовыми параметрами:

- p — основание системы счисления,

- t — разрядность числа,

- L — нижняя граница порядка,

- U — верхняя граница порядка.

Вещественное число x принадлежит множеству $\mathbb{F} = \mathbb{F}(p, t, L, U)$ в том и только том случае, если x представимо в виде

$$x = \pm d_0.d_1 d_2 d_3 \dots d_{t-1} \cdot p^e,$$

или, что то же самое, в виде

$$x = \pm \left(d_0 + \frac{d_1}{p} + \frac{d_2}{p^2} + \frac{d_3}{p^3} + \ldots + \frac{d_{t-1}}{p^{t-1}} \right) \cdot p^e,$$

где d_i — цифры, $0 \leq d_i \leq p-1$; e — целое число, $L \leq e \leq U$.

Число e в приведенных выше формулах называют **показателем** (или **экспонентой**).

Число $m = \pm d_0.d_1 d_2 d_3 \ldots d_{t-1}$ называют **мантиссой** (или **значащей** частью) числа x.

Представление числа x в указанном виде не единственно: положение точки в его записи может варьироваться, а само число при этом никак не изменяется. Например, справедливы равенства

$$\mathbf{0.0001234} = \mathbf{0.0012340} \cdot \mathbf{10}^{\mathbf{-1}} = \mathbf{1.2340000} \cdot \mathbf{10}^{\mathbf{-4}}.$$

Для того чтобы избежать этой неединственности в представлении числа, используется

нормализованная форма его записи, в которой точка ставится после первой значащей цифры.

Определение. *Число с плавающей точкой*

$$x = \pm d_0.d_1 d_2 d_3 \dots d_{t-1} \cdot p^e$$

с ненулевой цифрой d_0 *называется* ***нормализованным****.*

Множество всех нормализованных ЧПТ условимся обозначать как $\mathbb{F}_1 = \mathbb{F}_1(p, t, L, U)$.

Числовые множества $\mathbb{F}$ и $\mathbb{F}_1$ конечны, $\mathbb{F}_1 \subset \mathbb{F}$. При этом $\mathbb{F}_1 \neq \mathbb{F}$ (к примеру, нуль принадлежит $\mathbb{F}$, но не принадлежит $\mathbb{F}_1$).

Примеры записи ЧПТ в нормализованной форме приведены в следующей таблице (в скобках приводятся значения x в десятичной системе счисления):

p	t	x	m	e
10	6	0.000031415	3.14150	-5
2	10	$10001.101\ (17.625)$	1.000110100	$100\ (4)$
16	5	$ABC.D\ (2748.8125)$	$A.BCD0$	$2\ (2)$

Множество нормализованных ЧПТ дискретно и обладают следующими свойствами:

1. Числа из $\mathbb{F}_1$ распределены на числовой оси неравномерно.

2. Чем больше модуль числа x из $\mathbb{F}_1$, тем больше и расстояние между x и соседними с ним элементами из $\mathbb{F}_1$.

3. Между нулем из $\mathbb{F}$ и минимальным положительным $x_{\mathbf{min}}$ из $\mathbb{F}_1$ имеется “просвет”

— интервал, ширина которого больше нуля и больше расстояния от $x_{\min}$ до следующего за ним ЧПТ в k раз.

Отметим, что двоичные представления ЧПТ выгодно отличаются от всех остальных тем, что в нормализованной записи первый разряд d_0 всегда равен 1 и его можно не хранить в памяти.

В настоящее время общепринят стандарт двоичной арифметики IEEE 754. В нем под запись отводится 32 бита:

- 1 бит — под знак числа (s),
- 8 бит — под запись порядка числа (e),
- 23 бита — под мантиссу числа (m).

Заданным значениям s, e, m в рассматриваемом стандарте соответствует вещественное число, задаваемое равенством

$$x = (-1)^s 2^{e-127}(1+m).$$

4^0. Произвольное вещественное число далеко не всегда можно точно представить как ЧПТ с заданным числом t значащих цифр.

В общем случае число x **приходится округлять**. На практике x округляется до числа с плавающей точкой $R(x)$ по одному из следующих правил.

1. **Отбрасываются** лишние (избыточные в рамках используемого формата) знаки в мантиссе. В этом случае правило округления обозначается как R_d.

Пример: Пусть $x = 0.12345$, $p = 10$, $t = 3$.

Тогда округление задается равенством

$$R_d(0.12345) = 1.23 \cdot 10^{-1}.$$

2. Округление **вверх** (правило R_u).

Пример. Пусть $p = 10$, $t = 3$. Тогда

$$R_u(543.21) = 5.43 \cdot 10^2, \qquad R_u(5678) = 5.68 \cdot 10^3.$$

В случае, когда запись числа заканчивается на цифру ≥ 5, оно округляется до большего ближайшего к нему числа с плавающей точкой. Например, $R_u(23.45) = 2.35 \cdot 10^1$.

3. Округление **до четного** (правило R_e). От округления вверх отличается лишь действием в пограничном случае, когда x находится между двумя соседними ЧПТ $\underline{x}$ и $\overline{x}$ ровно посередине.

В этом случае оба возможных приближения $R_e(x) = \underline{x}$ и $R_e(x) = \overline{x}$ равноправны. Поэтому из двух вариантов округления выбирается тот, мантисса которого заканчивается на **четную** цифру.

Пример. Пусть $p = 10$, $t = 3$ и $x = 23.45$. Тогда $R_e(23.45) = 2.34 \cdot 10^1 (= \underline{x})$. Если же $x = 23.55$, то $R_e(23.55) = 2.36 \cdot 10^1$ $(= \overline{x})$.

Следует обратить внимание, что для каждого ненулевого x из множества $\mathbb{F}(p, t, L, U)$ справедливы неравенства

$m \leq |x| \leq M$, где $m = p^{L-1}$, $M = p^U \cdot (1 - p^{-t})$.

Рассмотрим числовое множество

$$\mathbb{G} = \left\{ x \in \mathbb{R} \mid m \leq x \leq M \right\} \cup \{0\},$$

на котором определим функцию

$$fl \colon \mathbb{G} \to \mathbb{F}(p, t, L, U),$$

задав ее значения равенством $\boldsymbol{fl(0) = 0}$, и полагая для $\boldsymbol{x \neq 0}$, что

$\boldsymbol{fl(x)}$ равно ближайшему к $\boldsymbol{x}$ числу $\boldsymbol{\widetilde{x}}$ из $\mathbb{F}$, если $\boldsymbol{\widetilde{x} = R_e(x)}$;

$\boldsymbol{fl(x)}$ равно ближайшему к $\boldsymbol{x}$ числу $\boldsymbol{\widetilde{x}}$ из $\mathbb{F}$, $|\widetilde{x}| \leq |x|$, если $\boldsymbol{\widetilde{x} = R_d(x)}$.

Так определенная функция $fl(x)$ удовлетворяет соотношению

$$fl(x) = x \cdot (1 + \varepsilon),$$

где $|\varepsilon| \leq u = \begin{cases} \frac{1}{2}p^{1-t} & \text{для округления } R_e; \\ p^{1-t} & \text{для округления } R_d. \end{cases}$

5^0. Пусть a — вещественное число и a^* — известное приближение к нему, $a \approx a^*$.

Определение. ***Абсолютной*** *погрешностью приближения* $a \approx a^*$ *называется такая положительная величина* $\Delta(a^*)$, *для которой*

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

Значащими цифрами числа называют все цифры в его записи, начиная с первой ненулевой слева. Нуль может быть значащей цифрой в двух случаях: когда он присутствует в

записи между двумя значащими цифрами и когда он стоит в конце числа и при этом известно, что единиц соответствующего разряда в данном числе не имеется.

Пример. $a^* = \mathbf{0.0\underline{3045}}$ и $a^* = \mathbf{0.0\underline{3045000}}$ (значащие цифры подчеркнуты).

Значащую цифру в приближении числа называют верной, если абсолютная погреш-

ность этого приближения не превосходит половины единицы разряда, соответствующего этой цифре.

Пример. Пусть $a^* = 0.03045$ и $\Delta(a^*) = 0.3 \cdot 10^{-5}$. Тогда $\Delta(a^*) < \frac{1}{2}10^{-5}$. Поэтому все значащие цифры в записи a^* верны.

Числа a^* и $\Delta(a^*)$ принято записывать с одинаковым числом знаков после точки.

Пример. Пусть $a = 1.123$ и $\Delta(a) = 0.004$. Тогда

$$1.123 - 0.004 \leq a \leq 1.123 + 0.004.$$

Определение. ***Относительной*** *погрешностью приближения* $a \approx a^*$, *где* $a^* \neq 0$, *называют такую положительную величину* $\delta(a^*)$, *для которой*

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

Относительную погрешность зачастую выражают в процентах.

Если, например, относительная погрешность приближения $a \approx a^*$ равна одному проценту, то это означает, что приближаемое число a принадлежит промежутку числовой оси с центром в точке a^* и шириной $0.02a^*$.

Тот факт, что a^* является приближением числа a с относительной погрешностью $\delta(a^*)$ записывают в виде $a = a^* \cdot \big(1 \pm \delta(a^*)\big)$.

Пример. Пусть $a = 1.123(1 \pm 0.003)$. Тогда

$$(1 - 0.003) \cdot 1.123 \leq a \leq (1 + 0.003) \cdot 1.123.$$

Абсолютную и относительную погрешности обычно записывают в виде числа, содержащего одну или две значащие цифры.

При переходе от одной формы записи числа к другой необходимо следить, чтобы допускаемый интервал изменения, указываемый новой формой, был **шире** интервала изменения, указываемого старой формой.

Если в постановке задачи говорится, что решение требуется найти с точностью 10^{-2}, то это означает, что погрешность должна

иметь порядок -2, то есть результат с погрешностью $2 \cdot 10^{-2}$ удовлетворителен.

6^0. Вычислительных алгоритмов, предназначенных для решения одной и той же задачи (как правило, в области непрерывного анализа), различными теоретическими руководствами предлагается достаточно много.

При выборе какого-либо одного из этих алгоритмов следует иметь ввиду, что численный результат может оказаться сильно зависящим от реализуемого вычислительного алгоритма.

Приведем пример, показывающий, что увеличения точности результата (то есть уменьшения погрешности метода) можно достичь

за счет несложных алгебраических преобразований.

Задача. *Найти* ***наименьший*** *корень квадратного уравнения*

$$y^2 - 140y + 1 = 0.$$

Пусть вычисления производятся в десятичной системе счисления, причем в мантиссе

числа после округления удерживается 4 **разряда**. Имеем по известной формуле для корней квадратного уравнения:

$$y = 70 - \sqrt{4899}, \qquad \sqrt{4899} \approx 69.992.$$

Округляя корень, получаем $\sqrt{4899} \approx 69.99$, и далее

$$y \approx 70 - 69.99 = 0.0100.$$

Преобразуем использованную формулу для корня с помощью известного разложения на

множители разности квадратов двух чисел. Тогда получим

$$y = \frac{(70 - \sqrt{4899})(70 + \sqrt{4899})}{70 + \sqrt{4899}} = \frac{1}{70 + \sqrt{4899}}.$$

Пользуясь все тем же приближением корня $\sqrt{4899} \approx 69.99$, получаем далее $70+69.99 = 139.99$. Округляя число справа до четырех цифр в мантиссе, получаем

$$70+69.99 \approx 140.0 \Rightarrow \frac{1}{140.0} \approx 0.00714285 \Rightarrow y \approx 0.007143\,.$$

Проведя вычисления с дополнительными разрядами, как это делает, например, встроенный в ноутбук калькулятор, можно получить приближенное значение искомого корня с 16 знаками после десятичной точки:

$$y_* \approx \mathbf{0.0071428571428571}.$$

Как легко увидеть, во втором случае абсолютная погрешность существенно меньше чем в первом случае. Таким образом, второй

вычислительный алгоритм дает существенно более точный результат.

Объяснение. В первом случае пришлось **вычитать близкие большие** числа, что привело к потере точности.

Этот эффект, называемый **потерей значащих цифр**, часто приводит к существенному искажению результата при решении систем линейных уравнений.

В настоящее время проблемы такого рода (правда, в простейших задачах) удается решить, производя вычисления с **двойной точностью**.

При реализации ЧМ на ЭВМ помимо прочего следует иметь ввиду, что ограничение на порядок машинных чисел может привести к переполнению, а в некоторых случаях недостаточно большая разрядность чисел в

ЭВМ может привести к сильному искажению результата вычислений.

Если реализуется хотя бы один из указанных сценариев, то говорят, что вычислительный алгоритм **неустойчив**. Построение устойчивых ВА составляет существенную часть вычислительной математики.

7^0. В вычислительной математике существенное значение имеет **обусловленность** задачи — чувствительность ее решения к малым изменениям входных данных.

Пример. *Вычислить **все** корни уравнения*

$$x^4 - 4x^3 + 8x^2 - 16x + 15.99999999 = 0,$$

или, в эквивалентной записи:

$$(x - 2)^4 - 10^8 = 0.$$

Точное решение уравнения легко получается по формулам

$$(x-2)^2 = \pm 10^{-4} \Rightarrow \quad x_1 = 2.01; \quad x_2 = 1.99;$$

$$x_3 = 2 + 0.01i; \quad x_4 = 2 - 0.01i.$$

Если точность ввода данных в компьютер меньше, чем 10^{-8}, то свободный член в исходном уравнении будет округлен до числа 16.0. При этом решаться будет возмущенное

уравнение $(x-2)^4 = 0$. Все его корни в точности равны 2. Таким образом, справедливы неравенства

$$|x_j - 2| \leqslant 0.01, \quad j = 1, 2, 3, 4.$$

Это означает, что результат работы компьютера приводит к приближению искомых корней с абсолютной погрешностью 10^{-2}.

В то же время абсолютная погрешность ввода значения последнего коэффициента уравнения в компьютер равна $\approx 10^{-8}$. Таким образом, точность результата больше погрешности ввода данных в миллион раз. Это означает, что исходная задача является плохо обусловленной.