# Оглавление

1. Погрешности арифметических операций над приближенными числами.

## Источники погрешностей результата

1. Математическая модель (приближенное описание реального процесса) + исходные данные (результат эксперимерта/решение вспомогательной задачи) $\to$ погрешность задачи $\delta_1 y$ (неустранимая погрешность, зависит от степени адекватности модели реальному процессу)
2. Численный метод (приближенный) $\to$ погрешность метода $\delta_2 y$
3. Округление $\to$ вычислительная погрешность $\delta_3 y$ (определяется характеристиками ЭВМ)

Полная погрешность результата решения задачи:

$$\delta y = y - y^* = \delta_1 y + \delta_2 y + \delta_3 y \qquad (1)$$

где $y$ и $y^*$ - точное и численное решение соответственно.

$$\delta_1 y > \delta_2 y > \delta_3 y \qquad (2)$$

## Абсолютная и относительная погрешности

Пусть $y$ - точное (неизвестное) значение некоторой величины, $y^*$ - приближенное (известное) значение той же величины.
Абсолютная погрешность приближенного значения $y^*$

$$\Delta(y^*) = |y - y^*| \qquad (3)$$

Относительная погрешность (не зависит от масштаба величины, единиц измерения)

$$\delta(y^*) = \frac{|y - y^*|}{|y|} \qquad (4)$$

Значение $y$ неизвестно $\Rightarrow$ получение оценок погрешностей, или верхних границ абсолютной и относительной погрешностей

$$\Delta(y^*) \leq \overline{\Delta}(y^*), \delta(y^*) \leq \overline{\delta}(y^*) \qquad (5)$$

## Сложение и вычитание приближенных чисел

Пусть $a^*$ и $b^*$ - приближенные значения чисел $a$ и $b$.

**Утверждение**

$$\Delta(a^* \pm b^*) \le \Delta(a^*) + \Delta(b^*) \qquad (6)$$

**Доказательство.**

$\Delta(a^* \pm b^*) = |(a \pm b) - (a^* \pm b^*)| = |(a - a^*) \pm (b - b^*)| \le$
$|(a - a^*)| + |(b - b^*)| = \Delta(a^*) + \Delta(b^*)$ $\square$

При сложении и вычитании приближенных чисел их предельные абсолютные погрешности складываются.

## Сложение и вычитание приближенных чисел

**Утверждение**

Пусть $a$ и $b$ - ненулевые числа одного знака. Тогда

$$\delta(a^* + b^*) \le \delta_{max}, \delta(a^* - b^*) \le \frac{|a+b|}{|a-b|}\delta_{max}, \qquad (7)$$

где $\delta_{max} = \max(\delta(a^*), \delta(b^*))$.

**Доказательство.**

Упражнение. $\square$

При суммировании чисел одного знака не происходит потери точности (в относительных единицах).
При вычитании чисел одного знака возможна существенная потеря точности.

# Умножение и деление приближенных чисел

### Утверждение

$$\delta(a^*b^*) \leq \delta(a^*) + \delta(b^*) + \delta(a^*)\delta(b^*) \approx \delta(a^*) + \delta(b^*) \qquad (8)$$

$$\delta\left(\frac{a^*}{b^*}\right) \leq \frac{\delta(a^*) + \delta(b^*)}{1 - \delta(b^*)} \approx \delta(a^*) + \delta(b^*) \qquad (9)$$

### Доказательство.

Упражнение. □

При умножении и делении приближенных чисел предельные относительные погрешности складываются.

# Подходы к учету погрешностей действий

1. Аналитический (громоздкий, наихудший случай)
2. Вероятностный, или статистический

### Пример. Среднее арифметическое

Пусть $x = \frac{1}{n}(x_1 + x_2 + \ldots + x_n)$ и все слагаемые имеют одинаковый уровень абсолютных погрешностей. Тогда классическая оценка (6)

$$\Delta(x) \approx \frac{1}{n}\left(\Delta(x_1) + \ldots + \Delta(x_n)\right) = \frac{1}{n} n\Delta(x_i) = \Delta(x_i). \qquad (10)$$

Статистический подход (правило Чеботарева):

$$\Delta(x) \approx \frac{1}{n}\sqrt{3n}\Delta(x_i) = \sqrt{\frac{3}{n}}\Delta(x_i) \xrightarrow[n\to\infty]{} 0. \qquad (11)$$

Арифметическое усреднение увеличивает точность.

2. Представление чисел с плавающей точкой. Множество компьютерных чисел. Особенности машинной арифметики

## Представление вещественных чисел

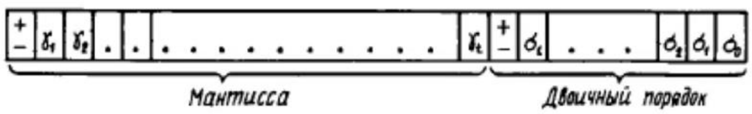
Представление с плавающей точкой (floating point):

$$x = \pm(\gamma_1 2^{-1} + \gamma_2 2^{-2} + \ldots + \gamma_t 2^{-t})2^p, \qquad (12)$$

где $\gamma_1, \ldots \gamma_t$ - двоичные цифры, причем $\gamma_1 = 1$.

$p$ - двоичный порядок, $p = \pm(\sigma_l \sigma_{l-1} \ldots \sigma_0)_2$.

$\mu = \pm(\gamma_1 2^{-1} + \gamma_2 2^{-2} + \ldots + \gamma_t 2^{-t})$ - мантисса числа $x$
$t$ - разрядность мантиссы.




Пример
$x = 20.5 = (10100.1)_2 = (0.101001)_2 \cdot 2^5$

## Представление с плавающей точкой

1. Множество компьютерных чисел float дискретно. Все остальные числа $x$ имеют приближенное представление $x^* = fl(x)$ с *ошибкой округления*. Относительная погрешность представления равна

$$\overline{\delta}(x^*) = 2^{1-t} \qquad (13)$$

Замечание
Почти наверняка во множестве компьютерных чисел нет числа $y$, являющегося решением поставленной задачи.
Лучший результат - найти представление $y^* = fl(y)$ с относительной точностью $\overline{\delta}(y^*)$.

Среди компьютерных чисел нет ни одного иррационального числа (в частности, $\pi$ и $e$).
Число 0.1 также отсутствует: $0.1 = (0.0001100110011\ldots)_2$.

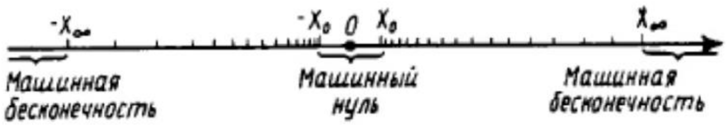
Относительная погрешность представления --???

# Представление с плавающей точкой

2. Диапазон изменения компьютерных чисел ограничен.
   $0.5 \leq |\mu| < 1$, так как $\gamma_1 = 1$
   $p \leq p_{max} = 2^{l+1} - 1 \Rightarrow$

$$0 < X_0 \leq |x| < X_\infty, \tag{14}$$

где $X_0 = 2^{-(p_{max}+1)}, X_\infty = 2^{p_{max}}$.




### Замечание

Диапазон представления компьютерных чисел определяется исключительно разрядностью порядка $l$.

# Представление с плавающей точкой

3. На машинной числовой оси числа расположены неравномерно:

$$\overline{\Delta}(x^*) = |x^*|2^{1-t} \tag{15}$$

4. *Машинный эпсилон*, $\epsilon_M$ - расстояние между единицей и ближайшим следующим за ней числом.
   $\epsilon_M = (1 \cdot 2^{-1} + 0 \cdot 2^{-2} + \ldots + 1 \cdot 2^{-t}) \cdot 2^1 - (1 \cdot 2^{-1} + 0 \cdot 2^{-2} + \ldots + 0 \cdot 2^{-t}) \cdot 2^1 = 1 \cdot 2^{-t} \cdot 2^1 = 2^{1-t}$

### Замечание

Машинный эпсилон $\epsilon_M$ служит мерой относительной точности представления вещественных чисел.

### Упражнение

Написать программу, которая вычисляет машинный эпсилон $\epsilon_M$.

## Особенности машинных арифметических операций

$\oplus, \otimes$ - машинные операции, соответствующие математическим операциям $+, \times$.

### Пример. Ассоциативность сложения

Пусть числа представляются с 6 двоичными разрядами мантиссы, округление производится по дополнению.
$a = (1.)_2, b = c = (0.000001)_2$
$a \oplus b = (1.00001)_2$ и $(a \oplus b) \oplus c = (1.00010)_2$
$c \oplus b = (0.000010)_2$ и $(c \oplus b) \oplus a = (1.00001)_2$

### Пример. Ассоциативность умножения

Пусть $D = p_{max}$, т.е. $2^D$ - правая граница числового диапазона.
$(2^{\frac{D}{2}} \otimes 2^{\frac{3D}{4}}) \otimes 2^{-\frac{D}{2}}$ - переполнение.
$2^{\frac{D}{2}} \otimes (2^{\frac{3D}{4}} \otimes 2^{-\frac{D}{2}})$ - правильный результат.

## Погрешности машинных арифметических операций

Воспользуемся следующим представлением

$$fl(a) = a(1 + \delta), |\delta| \leq \epsilon_M. \quad (16)$$

Тогда результат любой арифметической операции $\odot$

$$fl(a \odot b) = (a \odot b)(1 + \delta_1), |\delta_1| \leq \epsilon_M. \quad (17)$$

Рассмотрим сложение трех положительных чисел $a_1, a_2$ и $a_3$

$$fl(a_1 + a_2 + a_3) = ((a_1 + a_2)(1 + \delta_1) + a_3)(1 + \delta_2) \quad (18)$$

Абсолютная погрешность ($\epsilon = \max(\delta_1, \delta_2)$)

$$2(a_1 + a_2)\epsilon + a_3\epsilon + (a_1 + a_2)\epsilon^2 \approx 2(a_1 + a_2)\epsilon + a_3\epsilon. \quad (19)$$

Меньшую роль в (19) играет последнее слагаемое.

## Сумма $n$ положительных чисел

Обобщим (18) на случай $n$ положительных слагаемых.
Приближенная оценка абсолютной погрешности

$$\left| fl\left(\sum_{i=1}^{n} a_i\right) - \sum_{i=1}^{n} a_i \right| \le \big((n-1)(a_1+a_2) + (n-2)a_3 + \dots \\ + 2a_{n-1} + a_n\big)\,\epsilon. \qquad (20)$$

### Упражнение

Вывести оценку (20).

Чтобы погрешность была минимальной, последовательность чисел нужно суммировать в порядке возрастания членов.

3. Корректность вычислительной задачи и вычислительного алгоритма. Примеры неустойчивых задач и неустойчивых алгоритмов

## Корректность вычислительной задачи

Вычислительная задача называется корректной (по Адамару), если

1. ее решение $y$ существует при любых входных данных $x$
2. это решение единственно
3. решение устойчиво по отношению к малым возмущениям входных данных.

Решение $y$ вычислительной задачи называется *устойчивым по входным данным* $x$, если оно зависит от входных данных непрерывным образом.

$$\forall \epsilon > 0 \, \exists \delta = \delta(\epsilon) > 0 : \forall x^*, \Delta(x^*) < \delta \, \exists y^*, \Delta(y^*) < \epsilon. \qquad (21)$$

## Пример неустойчивой задачи

Рассмотрим задачу отыскания корней многочлена

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

Допустим, что в коэффициенте при $x^{19}$ сделана ошибка порядка $\epsilon_M$. Тогда возмущенный многочлен будет иметь следующие корни:

$$\begin{array}{lll}
x_1 \approx 1.000, & x_6 \approx 6.000, & x_{12,13} \approx 11.794 \pm 1.652i, \\
x_2 \approx 2.000, & x_7 \approx 7.000, & x_{14,15} \approx 13.992 \pm 2.519i, \\
x_3 \approx 3.000, & x_8 \approx 8.007, & x_{16,17} \approx 16.731 \pm 2.813i, \\
x_4 \approx 4.000, & x_9 \approx 8.917, & x_{18,19} \approx 19.502 \pm 1.940i, \\
x_5 \approx 5.000, & x_{10,11} \approx 10.095 \pm 0.644i, & x_{20} \approx 20.847.
\end{array}$$

Малое возмущение в одном коэффициенте качественно изменило набор корней многочлена.

# Пример неустойчивой задачи

Рассмотрим задачу вычисления производной приближенно заданной функции

$$u(x) = f'(x).$$

Пусть $f^*(x) = f(x) + \alpha \sin(x/\alpha^2)$ - приближенно заданная функция $(0 < \alpha \ll 1)$ на отрезке $[a, b]$.
Тогда $u^*(x) = u(x) + \alpha^{-1} \cos(x/\alpha^2)$.
Абсолютная погрешность $f^*$

$$\Delta(f^*) = \max_{[a,b]} |f(x) - f^*(x)| = \alpha,$$

в то время как абсолютная погрешность производной $u^*$

$$\Delta(u^*) = \max_{[a,b]} |u(x) - u^*(x)| = \alpha^{-1}.$$

# Корректность вычислительного алгоритма

Вычислительный алгоритм называется корректным, если

1. он позволяет получить результат $y$ за конечное число операций
2. результат $y$ устойчив по отношению к малым возмущениям входных данных
3. результат $y$ обладает вычислительной устойчивостью.

Алгоритм называется *вычислительно устойчивым*, если вычислительная погрешность результата стремится к 0 при $\epsilon_M \to 0$.

## Пример вычислительно неустойчивого алгоритма

Пусть требуется вычислить таблицу значений интегралов

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

Справедлива формула (правило интегрирования по частям)

$$I_n = nI_{n-1} - 1, n \geq 1 \qquad (22)$$

и $I_0 = \int_0^1 e^{1-x} = e - 1 \approx I_0^* = 1.71828$.
Последовательное вычисление приближенных значений интегралов

$I_1 \approx I_1^* = 1I_0^* - 1 = 0.71828$; $\quad I_2 \approx I_2^* = 2I_1^* - 1 = 0.43656$;
$I_3 \approx I_3^* = 3I_2^* - 1 = 0.30968$; $\quad I_4 \approx I_4^* = 4I_3^* - 1 = 0.23872$;
$I_5 \approx I_5^* = 5I_4^* - 1 = 0.19360$; $\quad I_6 \approx I_6^* = 6I_5^* - 1 = 0.16160$;
$I_7 \approx I_7^* = 7I_6^* - 1 = 0.13120$; $\quad I_8 \approx I_8^* = 8I_7^* - 1 = 0.00496$;
$I_9 \approx I_9^* = 9I_8^* - 1 = -0.55360$; $\quad I_{10} \approx I_{10}^* = 10I_9^* - 1 = -6.5360$ .

## Как изменить алгоритм, чтобы сделать его устойчивым?

Перепишем формулу (22)

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

и будем вести вычисления значений $I_n$ в обратном порядке, например, начиная с $n = 100$.
Положим $I_{100} \approx I_{100}^* = 0$. Тогда абсолютная погрешность $\Delta(I_{100}^*) \leq e/101$.

В данном случае вычислительные погрешности не растут, а затухают.
Модифицированный алгоритм вычислительно устойчив.