Численные методы решения задач динамики дисперсных систем

Погрешности численных методов

К.И. Михайленко

ИМех УФИЦ РАН

http://const.uimech.org/sG23df

институт механики им. Р.Р. Мавлютова

УФИЦ РАН

Последовательность слайдов

1. Два примера
   - Вычисление экспоненты
   - Решение СЛАУ
2. Представление вещественных чисел
   - Дискретность
   - Последствия дискретности
   - Машинный эпсилон
   - Последствия погрешности округления
3. Конкуренция погрешностей
   - Погрешность аппроксимации
   - Погрешность округления
   - Полная погрешность

Два примера

Пример 1. Вычисление экспоненты

Вычислим экспоненту, основываясь на разложении в ряд Тейлора

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

Алгоритм

```
get x
n=1; a=1; s=1
while |a| > ε
    a = a * x / n
    n += 1
    s += a
return s
```

Вычисление экспоненты. Fortran

Код

```fortran
program calc_exp
implicit none
real(8), dimension(8) ::
    & x = (/ 20., 10., 5., 2., -2., -5., -10., -20. /)
real(8) :: xx, c_exp, eps
integer :: i
read *, eps
do i = 1, 8
    xx = abs(c_exp(x(i)),eps)
    print *, x(i), abs(xx - exp(x(i))) / xx
end do
end program
```

Вычисление экспоненты. Fortran

Код (продолжение)

```
function c_exp(x,exp)
implicit none
real(8) :: x, eps, c_exp, a, s
integer :: n
a = 1.
s = 1.
n = 1.
do while (a > eps)
     a = a * x / n
     s = s + a
     n = n + 1
end do
c_exp = s
end function
```

Fortran. Результаты

Результаты

```
Δ(ε):    1.e-3        1.e-5        1.e-8

  20   4.235e-13   4.423e-15   1.229e-16
  10   7.372e-09   1.667e-10   4.215e-14
   5   1.402e-06   1.821e-08   5.604e-12
   2   8.308e-06   2.073e-07   5.606e-11
  -2   0.865       0.865       0.865
  -5   0.988       0.988       0.988
 -10   1           1           1
 -20   1           1           1
```

Вычисление экспоненты: Python

Код

```
import math

def c_exp(x,e):
    n=1.0; s=1.0; a=1.0
    while abs(a)>e
        a *= x/n
        s += a
        n += 1
    return abs(s)

e = float(input())
for x in [20,10,5,2,-2,-5,-10,-20]
    xx = c_exp(x,e)
    print (x,abs(xx-math.eps(x))/xx)
```

Вычисление экспоненты: Python

Результаты

```
Δ(ε):    1.e-3      1.e-5      1.e-8
 20  4.230e-13  3.931e-15  3.686e-16
 10  7.372e-09  1.667e-10  4.195e-14
  5  1.402e-06  1.821e-08  5.604e-12
  2  8.308e-06  2.073e-07  5.606e-11
 -2  0.00032    8.496e-06  2.449e-09
 -5  0.01824    0.00026    8.609e-08
-10  0.66184    0.04883    1.271e-05
-20  1          0.99813    0.61682
```

Пример 2. «Неповторимый»

$$\begin{pmatrix} 0.78 & 0.563 \\ 0.457 & 0.33 \end{pmatrix} \cdot \vec{x} = \begin{pmatrix} 0.217 \\ 0.127 \end{pmatrix}$$

Пример 2. «Неповторимый»

$$\begin{pmatrix} 0.78 & 0.563 \\ 0.457 & 0.33 \end{pmatrix} \cdot \vec{x} = \begin{pmatrix} 0.217 \\ 0.127 \end{pmatrix}$$

Решение методом Гаусса–Зейделя «в лоб» может дать результат

$$x = (1.71, -1.98)$$

Пример 2. «Неповторимый»

$$\begin{pmatrix} 0.78 & 0.563 \\ 0.457 & 0.33 \end{pmatrix} \cdot \vec{x} = \begin{pmatrix} 0.217 \\ 0.127 \end{pmatrix}$$

Решение методом Гаусса–Зейделя «в лоб» может дать результат

$$x = (1.71, -1.98)$$

Проверка невязки ($x = (1.71, -1.98)$)

$$|0.78 \cdot 1.71 - 0.563 \cdot 1.98 - 0.217| = 0.00206$$
$$|0.457 \cdot 1.71 - 0.333 \cdot 1.98 - 0.127| = 0.00107$$

Пример 2. «Неповторимый»

$$\begin{pmatrix} 0.78 & 0.563 \\ 0.457 & 0.33 \end{pmatrix} \cdot \vec{x} = \begin{pmatrix} 0.217 \\ 0.127 \end{pmatrix}$$

Решение методом Гаусса «в лоб» может дать результат

$$x = (0.278, 0)$$

Пример 2. «Неповторимый»

$$\begin{pmatrix} 0.78 & 0.563 \\ 0.457 & 0.33 \end{pmatrix} \cdot \vec{x} = \begin{pmatrix} 0.217 \\ 0.127 \end{pmatrix}$$

Решение методом Гаусса «в лоб» может дать результат

$$x = (0.278, 0)$$

Проверка невязки ($x = (0.278, 0)$)

$$|0.78 \cdot 0.278 - 0.563 \cdot 0 - 0.217| = 0.00016$$

$$|0.457 \cdot 0.278 - 0.333 \cdot 0 - 0.127| = 0.000046$$

Пример 2. «Неповторимый»

$$\begin{pmatrix} 0.78 & 0.563 \\ 0.457 & 0.33 \end{pmatrix} \cdot \vec{x} = \begin{pmatrix} 0.217 \\ 0.127 \end{pmatrix}$$

Ещё одно решение (программируемый микрокалькулятор)

$$x = (0.27983, 0.00225)$$

Пример 2. «Неповторимый»

$$\begin{pmatrix} 0.78 & 0.563 \\ 0.457 & 0.33 \end{pmatrix} \cdot \vec{x} = \begin{pmatrix} 0.217 \\ 0.127 \end{pmatrix}$$

Ещё одно решение (программируемый микрокалькулятор)

$$x = (0.27983, 0.00225)$$

Проверка невязки ($x = (0.27983, 0.00225)$)

$$|0.78 \cdot 0.27983 - 0.563 \cdot 0.00225 - 0.217| = 0.00000066$$

$$|0.457 \cdot 0.27983 - 0.333 \cdot 0.00225 - 0.127| = 0.00013981$$

Пример 2. «Неповторимый»

$$\begin{pmatrix} 0.78 & 0.563 \\ 0.457 & 0.33 \end{pmatrix} \cdot \vec{x} = \begin{pmatrix} 0.217 \\ 0.127 \end{pmatrix}$$

$$x = (1.71, -1.98), \qquad x = (0.278, 0), \qquad x = (0.27983, 0.00225)$$

Пример 2. «Неповторимый»

$$\begin{pmatrix} 0.78 & 0.563 \\ 0.457 & 0.33 \end{pmatrix} \cdot \vec{x} = \begin{pmatrix} 0.217 \\ 0.127 \end{pmatrix}$$

$$x = (1.71, -1.98), \qquad x = (0.278, 0), \qquad x = (0.27983, 0.00225)$$

Точное решение

$$x = (1, -1)$$

Почему?

Источник ошибок — компьютер

Подобные ошибки возникают из-за метода представления действительных чисел на вычислительном устройстве.

Представление вещественных чисел

Дискретность

Источник ошибок — компьютер

Подобные ошибки возникают из-за метода представления действительных чисел на вычислительном устройстве.

Дискретность

Источник ошибок — компьютер

Подобные ошибки возникают из-за метода представления действительных чисел на вычислительном устройстве.

Переход от непрерывного ряда к конечной дискретной последовательности

$$\mathbb{R} \to \{\mathbb{R}_n\}$$

Дискретность

Источник ошибок — компьютер
Подобные ошибки возникают из-за метода представления действительных чисел на вычислительном устройстве.

Переход от непрерывного ряда к конечной дискретной последовательности

$$\mathbb{R} \to \{\mathbb{R}_n\}$$

Хранение в виде мантиссы фиксированной конечной длины n и показателя степени фиксированной длины m.

$$\underbrace{a.aa\ldots a}_{n} \cdot 2^{\overbrace{b\ldots b}^{m}} \qquad a, b \in [0, 1]$$

Дискретность

Источник ошибок — компьютер

Подобные ошибки возникают из-за метода представления действительных чисел на вычислительном устройстве.

Переход от непрерывного ряда к конечной дискретной последовательности

$$\mathbb{R} \to \{\mathbb{R}_n\}$$

Хранение в виде мантиссы фиксированной конечной длины n и показателя степени фиксированной длины m.

$$\underbrace{a.aa\ldots a}_{n} \cdot 2^{\overbrace{b\ldots b}^{m}} \qquad a, b \in [0, 1]$$

$$0.\underbrace{aa\ldots a}_{n} \cdot 2^{\overbrace{b\ldots b}^{m}} \qquad a, b \in [0, 1]$$

Модельное дискретное представление

$$n = 4, \quad m = 2, \quad a, b \in [0, \ldots 9] \quad \rightarrow \quad a.aaa \cdot 10^{bb}$$

Модельное дискретное представление

$$n = 4, \quad m = 2, \quad a, b \in [0, \ldots 9] \quad \rightarrow \quad a.aaa \cdot 10^{bb}$$

1. Существует наибольшее число ($9.999 \cdot 10^{99} \sim 10^{100}$).

Модельное дискретное представление

$$n = 4, \quad m = 2, \quad a, b \in [0, \ldots 9] \quad \rightarrow \quad a.aaa \cdot 10^{bb}$$

1. Существует наибольшее число ($9.999 \cdot 10^{99} \sim 10^{100}$).
2. Существует наименьшее число ($-9.999 \cdot 10^{99} \sim -10^{100}$).

Модельное дискретное представление

$$n = 4, \quad m = 2, \quad a, b \in [0, \ldots 9] \quad \rightarrow \quad a.aaa \cdot 10^{bb}$$

1. Существует наибольшее число ($9.999 \cdot 10^{99} \sim 10^{100}$).
2. Существует наименьшее число ($-9.999 \cdot 10^{99} \sim -10^{100}$).
3. Существует разрыв возле нуля ($0 \ldots 0.001 \cdot 10^{-99}; \quad \Delta_0 \sim 10^{-102}$).

Последствия дискретности

1. **Ошибка переполнения (overflow).**
   Результат арифметической операции превышает наибольшее число.

Последствия дискретности

1. **Ошибка переполнения (overflow).**
   Результат арифметической операции превышает наибольшее число.
2. **Ошибка исчезновения порядка.**
   Результат арифметической операции **не** превышает Δ_0.
   В этом случае результат приравнивается нулю.

Последствия дискретности

1. **Ошибка переполнения (overflow).**
   Результат арифметической операции превышает наибольшее число.
2. **Ошибка исчезновения порядка.**
   Результат арифметической операции **не** превышает Δ_0.
   В этом случае результат приравнивается нулю.
3. **Ошибка округления.**
   Операнды арифметической операции различаются показателем степени.
   Если разность показателей степени превышает длину мантиссы, ошибка округления может быть равна одному из операндов.

$$1.234 \cdot 10^{00} + 5.678 \cdot 10^{-04} = 1.234\underbrace{5678}_{?}$$

Последствия дискретности

1. **Ошибка переполнения (overflow).**
   Результат арифметической операции превышает наибольшее число.
2. **Ошибка исчезновения порядка.**
   Результат арифметической операции **не** превышает Δ_0.
   В этом случае результат приравнивается нулю.
3. **Ошибка округления.**
   Операнды арифметической операции различаются показателем степени.
   Если разность показателей степени превышает длину мантиссы, ошибка округления может быть равна одному из операндов.

$$1.234 \cdot 10^{00} + 5.678 \cdot 10^{-04} = 1.234\underbrace{5678}_{?}$$

$= 1.234$ (усечение)
$= 1.235$ (округление)

Машинный эпсилон

Наименьшее число с плавающей запятой, которое при сложении с вещественным числом 1.0 даст результат бóльший 1.0 называют **машинным эпсилоном.**

Машинный эпсилон

Наименьшее число с плавающей запятой, которое при сложении с вещественным числом 1.0 даст результат бóльший 1.0 называют **машинным эпсилоном.**

Тип float:

$\varepsilon_{\text{comp}} = 2^{-23} \approx 1.2 \cdot 10^{-7}$ (усечение) или

$\varepsilon_{\text{comp}} = 2^{-24} \approx 6 \cdot 10^{-8}$ (округление)

Машинный эпсилон

Наименьшее число с плавающей запятой, которое при сложении с вещественным числом 1.0 даст результат бóльший 1.0 называют **машинным эпсилоном.**

Тип float:

$\varepsilon_{\text{comp}} = 2^{-23} \approx 1.2 \cdot 10^{-7}$ (усечение) или
$\varepsilon_{\text{comp}} = 2^{-24} \approx 6 \cdot 10^{-8}$ (округление)

Тип double (precision):

$\varepsilon_{\text{comp}} = 2^{-47} \approx 7.1 \cdot 10^{-15}$ (усечение) или
$\varepsilon_{\text{comp}} = 2^{-48} \approx 3.6 \cdot 10^{-15}$ (округление)

Машинный эпсилон

Наименьшее число с плавающей запятой, которое при сложении с вещественным числом 1.0 даст результат бо́льший 1.0 называют **машинным эпсилоном.**

Тип float:

$\varepsilon_{\text{comp}} = 2^{-23} \approx 1.2 \cdot 10^{-7}$ (усечение) или

$\varepsilon_{\text{comp}} = 2^{-24} \approx 6 \cdot 10^{-8}$ (округление)

Тип double (precision):

$\varepsilon_{\text{comp}} = 2^{-47} \approx 7.1 \cdot 10^{-15}$ (усечение) или

$\varepsilon_{\text{comp}} = 2^{-48} \approx 3.6 \cdot 10^{-15}$ (округление)

Тип «специальный»:

$\varepsilon_{\text{comp}} = 0.001$ (усечение) или

$\varepsilon_{\text{comp}} = 0.0005$ (округление)

Машинный эпсилон (интерпретация)

Машинный эпсилон определяет фактическую относительную точность машинной арифметики.

Машинный эпсилон (интерпретация)

Машинный эпсилон определяет фактическую относительную точность машинной арифметики.
Для $x, y > 0$ и $x > y$ запишем

$$x + y = x\left(1 + \frac{x}{y}\right)$$

Из определения $\varepsilon_{\text{comp}}$ следует, что если отношение $\dfrac{x}{y} < \varepsilon_{\text{comp}}$,
то вычисление даст $x + y = x$

Машинный эпсилон (интерпретация)

Машинный эпсилон определяет фактическую относительную точность машинной арифметики.
Для $x, y > 0$ и $x > y$ запишем

$$x + y = x\left(1 + \frac{x}{y}\right)$$

Из определения $\varepsilon_{\text{comp}}$ следует, что если отношение $\dfrac{x}{y} < \varepsilon_{\text{comp}}$,
то вычисление даст $x + y = x$

> Относительная погрешность сложения чисел с плавающей запятой ограничена величиной $\varepsilon_{\text{comp}}$

Машинный эпсилон (выводы)

1. При использовании арифметики типа float точность представления результата операции **не превышает 7 верных знаков** после запятой.

Машинный эпсилон (выводы)

1. При использовании арифметики типа float точность представления результата операции **не превышает 7 верных знаков** после запятой.
2. При использовании арифметики типа double точность представления результата операции **не превышает 15 верных знаков** после запятой.

Машинный эпсилон (выводы)

1. При использовании арифметики типа float точность представления результата операции **не превышает 7 верных знаков** после запятой.
2. При использовании арифметики типа double точность представления результата операции **не превышает 15 верных знаков** после запятой.
3. При итерационных расчетах «до сходимости» вида

$$||A_{n+1} - A_n|| \leqslant \varepsilon$$

необходимое требование $\varepsilon \geqslant \varepsilon_{\text{comp}}$
Замечание: на самом деле необходимо $\varepsilon \gg \varepsilon_{\text{comp}}$

Дискретность и двоичное представление

Переход $\mathbb{R} \to \{\mathbb{R}_n\}$ с учетом преобразования в двоичное представление: не все числа являются тем, чем кажутся.

Дискретность и двоичное представление

Переход $\mathbb{R} \to \{\mathbb{R}_n\}$ с учетом преобразования в двоичное представление: не все числа являются тем, чем кажутся.

$$0.2 + 0.1 \neq 0.3$$

Дискретность и двоичное представление

Переход $\mathbb{R} \to \{\mathbb{R}_n\}$ с учетом преобразования в двоичное представление: не все числа являются тем, чем кажутся.

$$0.2 + 0.1 \neq 0.3$$

```
$ python
Python 3.13.0 (main, Oct  8 2024, 00:00:00) [GCC 14.2.1 2024091
Type "help", "copyright", "credits" or "license" for more inforn
>>> a=0.2+0.1
>>> a
0.30000000000000004
>>>
```

Дискретность и двоичное представление

Представим хранимое число в виде

$$x_{\text{comp}} = x(1+\delta_x)$$

где δ_x – погрешность хранимого на компьютере приближения исходного числа x.

Дискретность и двоичное представление

Представим хранимое число в виде

$$x_{\text{comp}} = x(1 + \delta_x)$$

где δ_x – погрешность хранимого на компьютере приближения исходного числа x.

Утверждение

$$|\delta_x| \leqslant \varepsilon_{\text{comp}}$$

Дискретность и двоичное представление

Представим хранимое число в виде

$$x_{\text{comp}} = x(1 + \delta_x)$$

где δ_x — погрешность хранимого на компьютере приближения исходного числа x.

Утверждение

$$|\delta_x| \leqslant \varepsilon_{\text{comp}}$$

Именно наличие этой погрешности приводит к результатам, показанным в исходных примерах.

Катастрофическая потеря верных цифр

Наиболее выражено проявление погрешности хранимого значения проявляется при вычитании близких значений. Для этого случая придумано название: **катастрофическая потеря верных цифр.**

Рассмотрим на примере ранее введённой модельной четырёхразрядной десятичной дискретной арифметики.

Катастрофическая потеря верных цифр

Имеем два числа x и y таких, что

$$x_{\text{comp}} = 1.001 \cdot 10^{00}, \qquad y_{\text{comp}} = 1.002 \cdot 10^{00}$$

Вычислим разность $z = y - x$.

Катастрофическая потеря верных цифр

Имеем два числа x и y таких, что

$$x_{\text{comp}} = 1.001 \cdot 10^{00}, \qquad y_{\text{comp}} = 1.002 \cdot 10^{00}$$

Вычислим разность $z = y - x$.

При вычислении разности, если учесть погрешности хранимого значения, выражение для относительной погрешности арифметической операции:

$$\delta_{\text{relative}} = \frac{(y_{\text{comp}} - x_{\text{comp}}) - (y - x)}{y_{\text{comp}} - x_{\text{comp}}} = \frac{|\delta_y| + |\delta_x|}{z_{\text{comp}}}$$

Катастрофическая потеря верных цифр

Имеем два числа x и y таких, что

$$x_{\text{comp}} = 1.001 \cdot 10^{00}, \qquad y_{\text{comp}} = 1.002 \cdot 10^{00}$$

Вычислим разность $z = y - x$.

При вычислении разности, если учесть погрешности хранимого значения, выражение для относительной погрешности арифметической операции:

$$\delta_{\text{relative}} = \frac{(y_{\text{comp}} - x_{\text{comp}}) - (y - x)}{y_{\text{comp}} - x_{\text{comp}}} = \frac{|\delta_y| + |\delta_x|}{z_{\text{comp}}}$$

Подставим значения:

$$z_{\text{comp}} = 0.001, \qquad |\delta_x| = |\delta_y| \leqslant \varepsilon_{\text{comp}}$$

Катастрофическая потеря верных цифр

Имеем два числа x и y таких, что

$$x_{\text{comp}} = 1.001 \cdot 10^{00}, \qquad y_{\text{comp}} = 1.002 \cdot 10^{00}$$

Вычислим разность $z = y - x$.

При вычислении разности, если учесть погрешности хранимого значения, выражение для относительной погрешности арифметической операции:

$$\delta_{\text{relative}} = \frac{(y_{\text{comp}} - x_{\text{comp}}) - (y - x)}{y_{\text{comp}} - x_{\text{comp}}} = \frac{|\delta_y| + |\delta_x|}{z_{\text{comp}}}$$

Подставим значения:

$$z_{\text{comp}} = 0.001, \qquad |\delta_x| = |\delta_y| \leqslant \varepsilon_{\text{comp}}$$

$$\delta_{\text{relative}} \leqslant \frac{2\varepsilon_{\text{comp}}}{z_{\text{comp}}}$$

Катастрофическая потеря верных цифр

Имеем два числа x и y таких, что

$$x_{\text{comp}} = 1.001 \cdot 10^{00}, \qquad y_{\text{comp}} = 1.002 \cdot 10^{00}$$

Вычислим разность $z = y - x$.

При вычислении разности, если учесть погрешности хранимого значения, выражение для относительной погрешности арифметической операции:

$$\delta_{\text{relative}} = \frac{(y_{\text{comp}} - x_{\text{comp}}) - (y - x)}{y_{\text{comp}} - x_{\text{comp}}} = \frac{|\delta_y| + |\delta_x|}{z_{\text{comp}}}$$

Подставим значения:

$$z_{\text{comp}} = 0.001, \qquad |\delta_x| = |\delta_y| \leqslant \varepsilon_{\text{comp}}$$

$$\delta_{\text{relative}} \leqslant \frac{2\varepsilon_{\text{comp}}}{z_{\text{comp}}} \approx 1$$

Конкуренция погрешностей

Погрешность аппроксимации

Рассмотрим аппроксимацию первой производной

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Погрешность аппроксимации

Рассмотрим аппроксимацию первой производной

$$f'(x) = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h} \approx \frac{f(x+h) - f(x)}{h} + \Delta_{\text{appr}}(h)$$

Оценим $\Delta_{\text{appr}}(h)$.

Погрешность аппроксимации

Рассмотрим аппроксимацию первой производной

$$f'(x) = \lim_{h\to 0} \frac{f(x+h)-f(x)}{h} \approx \frac{f(x+h)-f(x)}{h} + \Delta_{\text{appr}}(h)$$

Оценим $\Delta_{\text{appr}}(h)$.
Предположим, что вторая производная ограничена сверху:

$$|f''(x)| \leqslant M$$

Тогда из определения производной можно записать

$$|\Delta_{\text{appr}}(h)| = \left| f'(x) - \frac{f(x+h)-f(x)}{h} \right|$$

Погрешность аппроксимации

Рассмотрим аппроксимацию первой производной

$$f'(x) = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h} \approx \frac{f(x+h) - f(x)}{h} + \Delta_{\text{appr}}(h)$$

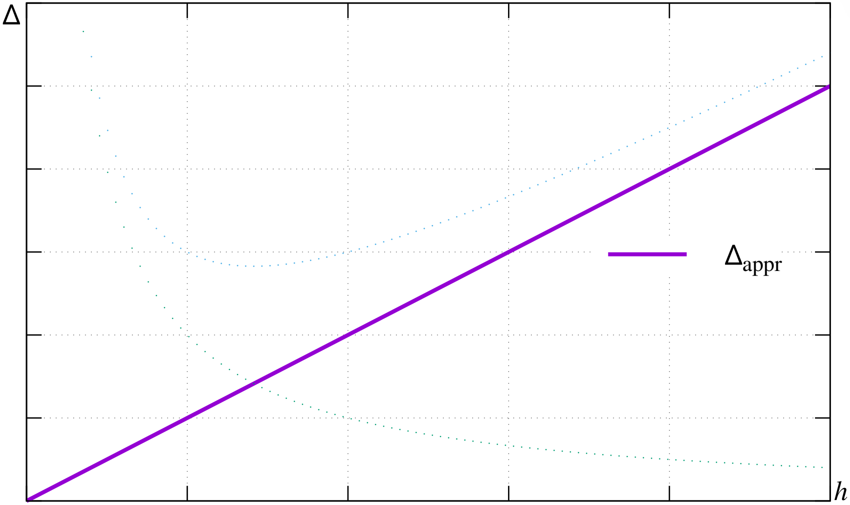
Оценим $\Delta_{\text{appr}}(h)$.
Предположим, что вторая производная ограничена сверху:

$$|f''(x)| \leqslant M$$

Тогда из определения производной можно записать

$$|\Delta_{\text{appr}}(h)| = \left| f'(x) - \frac{f(x+h) - f(x)}{h} \right| \leqslant \frac{Mh}{2}$$

Погрешность аппроксимации



Погрешность округления

Погрешность, определяемая хранимыми приближениями
также может быть оценена.

Погрешность округления

Погрешность, определяемая хранимыми приближениями также может быть оценена.

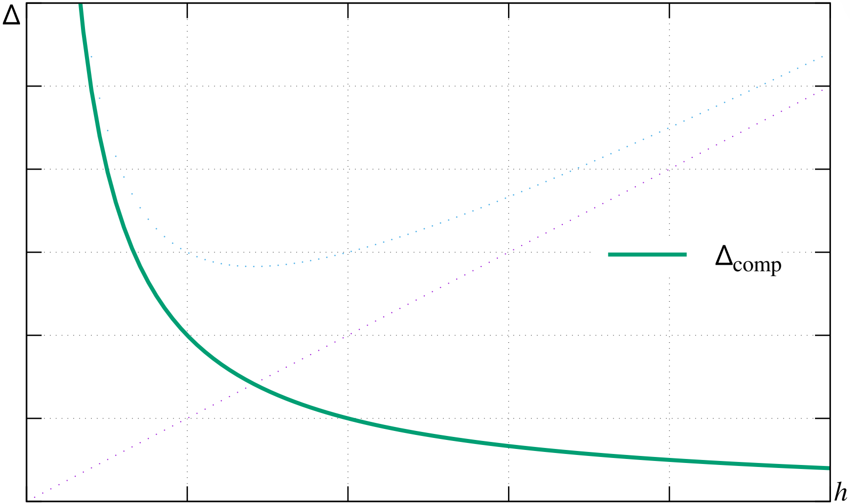
$$|\Delta_{\text{comp}}| = \left| \frac{f(x+h) - f(x)}{h} - f'(x) \right|$$

Погрешность округления

Погрешность, определяемая хранимыми приближениями также может быть оценена.

$$\left|\Delta_{\text{comp}}\right| = \left|\frac{f(x+h)-f(x)}{h} - f'(x)\right| \leqslant \frac{4\varepsilon_{\text{comp}}}{h}$$

Погрешность округления

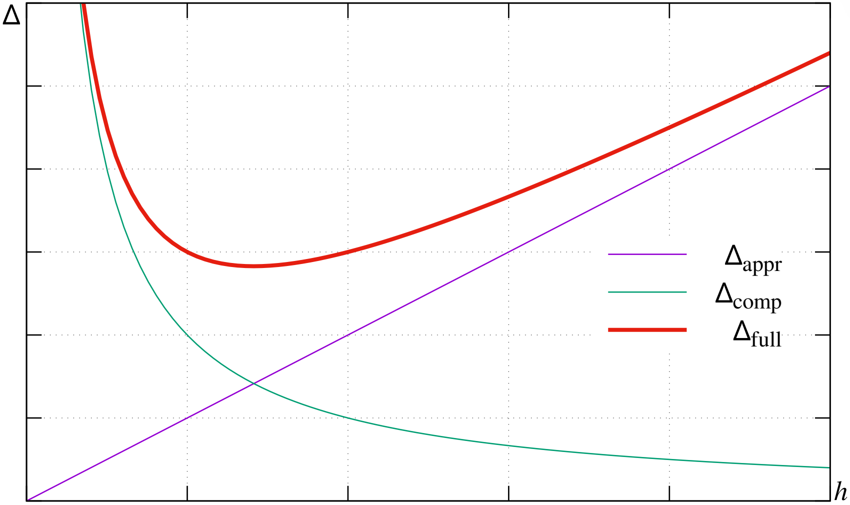


Полная погрешность

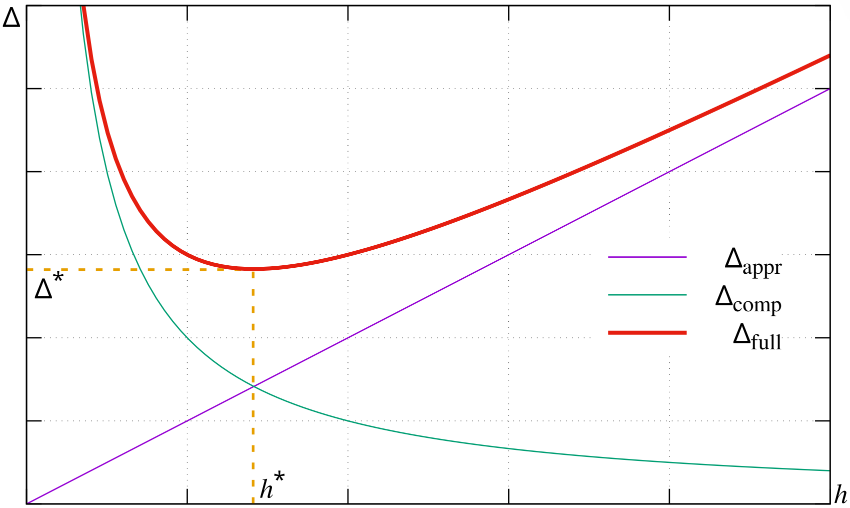
Полная погрешность вычисляемой аппроксимации производной складывается из суммы погрешности аппроксимации и погрешности округления:

$$|\Delta_{\text{full}}| = |\Delta_{\text{appr}}| + |\Delta_{\text{comp}}| \leqslant \frac{Mh}{2} + \frac{4\varepsilon_{\text{comp}}}{h}$$

Полная погрешность



Полная погрешность



Полная погрешность

Величина $h = h^*$, при котором достигается минимальная погрешность, может быть оценена

Полная погрешность

Величина $h = h^*$, при котором достигается минимальная погрешность, может быть оценена

$$h^* = 2\sqrt{\frac{2\varepsilon_{\text{comp}}}{M}}$$

Полная погрешность

Величина $h = h^*$, при котором достигается минимальная погрешность, может быть оценена

$$h^* = 2\sqrt{\frac{2\varepsilon_{\text{comp}}}{M}}$$

Значение минимально достижимой погрешности менее интересно, но также может быть оценено

Полная погрешность

Величина $h = h^*$, при котором достигается минимальная погрешность, может быть оценена

$$h^* = 2\sqrt{\frac{2\varepsilon_{\text{comp}}}{M}}$$

Значение минимально достижимой погрешности менее интересно, но также может быть оценено

$$\Delta^* = 2\sqrt{2M\varepsilon_{\text{comp}}}$$

Задание на лабораторную работу

http://const.uimech.org/De38cY