Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «МЭИ»

Институт Информационных и Вычислительных Технологий

Кафедра Математического и компьютерного моделирования

ТИПОВЫЕ РАСЧЕТЫ

Вариант: 14

Студент гр. А-02-22 ________________________________ Пивоваров Я.В.

(подпись)

Преподаватель ________________________________ Пепа Р.Ю.

(оценка/зачёт, подпись)

Москва

2024

ТР. 1.14 — Пивоваров Я. В. А-01-22

$z = \sqrt{18.12} + \sqrt[3]{11.12} + \sqrt[4]{88.11}$

Вычислить значение z и оценить абсолютную и относительную погрешности результата. Указать верные цифры

$z(x_1; x_2; x_3) = \sqrt{x_1} + \sqrt[3]{x_2} + \sqrt[4]{x_3}$

$x_1^* = 18.12 \quad \Delta x_1^* = 0{,}005$

$x_2^* = 11.12 \quad \Delta x_2^* = 0{,}005$

$x_3^* = 88.11 \quad \Delta x_3^* = 0{,}005$

$$\Delta(f^*) \leq \sum_{i=1}^{n} \left| \frac{\partial f}{\partial x_i}(x_1^* \dots x_n^*) \right| \cdot \Delta(x_i^*)$$

$$\Delta(z^*) \leq \left| \frac{1}{2\sqrt{x_1^*}} \right| \Delta(x_1^*) + \left| \frac{1}{3\sqrt[3]{(x_2^*)^2}} \right| \cdot \Delta(x_2^*) + \left| \frac{1}{4\sqrt[4]{(x_3^*)^3}} \right| \cdot \Delta(x_3^*) =$$

$$= \frac{1}{2\sqrt{18{,}12}} \cdot 0{,}005 + \frac{1}{3\sqrt[3]{11{,}12^2}} \cdot 0{,}005 + \frac{1}{4\sqrt[4]{88{,}11^3}} \cdot 0{,}005 = 0{,}000965 = 9{,}65 \cdot 10^{-4}$$

абс. погрешность

$\tilde{z} = 9{,}55257$ — значен. $\delta z = \frac{0{,}000965}{9{,}55257} = 0{,}000101 \approx 1{,}01 \cdot 10^{-4}$ — отн. погрешность

$z = \underbrace{9{,}5525}_{\text{верные}}7 + 0{,}000965$

Ответ: $z = 9{,}55257 \pm 0{,}00096$, результат содержит 4 верные цифры

$\delta z = 1{,}01 \cdot 10^{-4}$ — отн. погрешность.

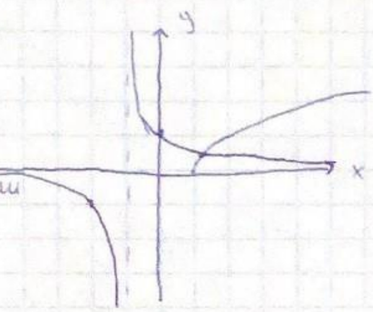
Т.Р. 2.14

$f(x)=\sqrt{x-1}-\frac{1}{x+1}$ $\varepsilon=0{,}01$

Локализовать $f(x)=0$ методом бисекции

$\sqrt{x-1}-\frac{1}{x+1}=0$

$\sqrt{x-1}=\frac{1}{x+1}$



$\Rightarrow$ имеет один корень

Составим таблицу исх. ф-ции

x	0	1	2	3
$f(x)$	$-$	$-\frac{1}{2}$	$\frac{2}{3}$	1,15

Из таблицы видно, что в качестве отрезка локализации можно взять отрезок [1;2], т.к на нем ф-ция меняет знак.

Метод бисекции:

n	a_n	b_n	x_n	знак $f(a_n)$	знак $f(b_n)$	знак $f(x_n)$	b_n-a_n
0	1	2	1,5	$-$	$+$	$+$	1
1	1	1,5	1,25	$-$	$+$	$+$	0,5
2	1	1,25	1,125	$-$	$+$	$-$	0,25
3	1,125	1,25	1,1875	$-$	$+$	$-$	0,125
4	1,1875	1,25	1,21875	$-$	$+$	$+$	0,0625
5	1,1875	1,21875	1,203125	$-$	$+$	$-$	0,03125
6	1,203125	1,21875	1,2109375	$-$	$+$	$+$	$0{,}015625 \le 0{,}02 = 2\varepsilon$

[1,203125 ; 1,21875]

Ответ: $\bar{x}=1{,}20\pm 0{,}01$

Т.р 3.14

$f(x) = \sqrt{x-1} - \frac{1}{x+1}$ $\quad \varepsilon = 0,0001$ методом простой итерации

$[a;b] = [1,203125;\ 1,21875]$

$\varphi(x) = x - \alpha f(x) \Rightarrow \alpha = \frac{2}{M+m}$

$f'(x) = \frac{1}{2\sqrt{x-1}} + \frac{1}{(x+1)^2}$

$M = 1,24218$

$m = 1,31543$

$\alpha = \frac{2}{1,27218 + 1,31543} = 0,772914$

$q = \left|\frac{M-m}{M+m}\right| = 0,0167143$

$D = \frac{q}{1-q} = \frac{0,0167143}{1 - 0,0167143} \approx 0,017$

$\varphi(x) = x - 0,772914\left(\sqrt{x-1} - \frac{1}{x+1}\right)$

$x^{(1)} = \varphi(x^{(0)}) = \varphi(1,2109375) = 1,20554$

$D \cdot |x^{(1)} - x^{(0)}| = 0,017\,|1,20554 - 1,2109375| = 0,0000917 \le 0,0001 = \varepsilon$

Ответ: $\bar{x} = x^{(1)} \pm \varepsilon = 1,2055 \pm 0,0001$

Пр 4.14

$e^x - 2x - 5 = 0$ $[0;3]$; $\varepsilon = 10^{-8}$ методом Ньютона.

$x_{k+1} = x_k - \dfrac{f(x_k)}{f'(x_k)}$; $x_{k+1} = x_k - \dfrac{e^x - 2x - 5}{e^x - 2}$

$x_0 = \dfrac{3-0}{2} = 1{,}5$ $f'(x) = e^x - 2$

$x_1 = x_0 - \dfrac{e^{x_0} - x_0 \cdot 2 - 5}{e^{x_0} - 2}$

$x_1 = 1{,}5 - \dfrac{e^{1{,}5} - 3 - 5}{e^{1{,}5} - 2} \approx 2{,}91770819$ $|x_1 - x_0| > \varepsilon$

$x_2 = 2{,}91771 - \dfrac{e^{2{,}91771} - 2 \cdot 2{,}91771 - 5}{e^{2{,}91771} - 2} \approx 2{,}4532255521$ $|x_2 - x_1| > \varepsilon$

$x_3 = x_2 - \dfrac{e^{x_2} - 2 \cdot x_2 - 5}{e^{x_2} - 2} = 2{,}27460794477$ $|x_3 - x_2| > \varepsilon$

$x_4 = x_3 - \dfrac{e^{x_3} - 2 \cdot x_3 - 5}{e^{x_3} - 2} = 2{,}251965843$ $|x_4 - x_3| > \varepsilon$

$x_5 = x_4 - \dfrac{e^{x_4} - 2x_4 - 5}{e^{x_4} - 2} = 2{,}2516362719$ $|x_5 - x_4| > \varepsilon$

$x_6 = x_5 - \dfrac{e^{x_5} - 2x_5 - 5}{e^{x_5} - 2} = 2{,}25163620.3$ $|x_6 - x_5| > \varepsilon$

$x_7 = x_6 - \dfrac{e^{x_6} - 2x_6 - 5}{e^{x_6} - 2} = 2{,}2516362031$ $|x_7 - x_6| < \varepsilon$

Ответ: $\bar{x} = 2{,}25163620 \pm 10^{-8}$

Пр 5.14 Решить систему ур-ний $Ax = b$ методом Гаусса (схема ед. дел.)

$$A = \begin{pmatrix} 9 & 4 & 2 & 0 \\ 9 & 11 & -2 & -1 \\ -81 & -106 & 12 & 8 \\ -90 & -68 & 6 & 8 \end{pmatrix} \quad b = \begin{pmatrix} -37 \\ -45 \\ 465 \\ 356 \end{pmatrix}$$

Решение: Прямой ход

Шаг 1

$$\mu_{2,1} = \frac{a_{21}}{a_{1,1}} = \frac{9}{9} = 1; \quad \mu_{3,1} = \frac{a_{3,1}}{a_{1,1}} = \frac{-81}{9} = -9$$

$$\mu_{4,1} = \frac{a_{4,1}}{a_{1,1}} = \frac{-90}{9} = -10$$

$$\begin{cases} 9x_1 + 4x_2 + 2x_3 + 0 \cdot x_4 = -37 \\ 7x_2 - 4x_3 - x_4 = -8 \\ -70x_2 + 30x_3 + 8x_4 = 132 \\ -28x_2 + 26x_3 + 8x_4 = -12 \end{cases}$$

Шаг 2:

$$\mu_{3,2} = \frac{a^{(1)}_{3,2}}{a^{(1)}_{2,2}} = \frac{-70}{7} = -10$$

$$\mu_{4,2} = \frac{a^{(1)}_{4,2}}{a^{(1)}_{2,2}} = \frac{-28}{7} = -4$$

$$\begin{cases} 9x_1 + 4x_2 + 2x_3 + 0x_4 = -37 \\ 7x_2 - 4x_3 - x_4 = -8 \\ -10x_3 - 2x_4 = 52 \\ 10x_3 + 4x_4 = -44 \end{cases}$$

Шаг 3:

$$\mu_{4,3} = \frac{a^{(2)}_{4,3}}{a^{(2)}_{3,3}} = \frac{10}{-10} = -1$$

$$\begin{cases} 9x_1 + 4x_2 + 2x_3 + 0x_4 = -37 \\ 7x_2 - 4x_3 - x_4 = -8 \\ -10x_3 - 2x_4 = 52 \\ 2x_4 = 8 \end{cases}$$

Обратный ход

$$\begin{cases} x_4 = 4 \\ x_3 = -\frac{52+8}{10} = -6 \\ x_2 = \frac{-8-24+4}{7} = -4 \\ x_1 = \frac{+16-37+12}{9} = -1 \end{cases}$$

Ответ: $x = \begin{pmatrix} -1 \\ -4 \\ -6 \\ 4 \end{pmatrix}$

П.р № 8.14 Решить систему ур-ний $Ax=b$ методом прогонки.

Промежуточные результаты вычислить с шестью знаками после запятой.

$$A=\begin{pmatrix} 12 & -6 & 0 & 0 & 0 \\ -4 & 8 & -1 & 0 & 0 \\ 0 & -5 & 16 & 3 & 0 \\ 0 & 0 & 3 & 13 & -4 \\ 0 & 0 & 0 & -3 & 6 \end{pmatrix} \quad \vec{b}=\begin{pmatrix} 30 \\ -12 \\ -65 \\ -18 \\ -24 \end{pmatrix} \quad \begin{cases} 12x_1 - 6x_2 = 30 \\ -4x_1 + 8x_2 - x_3 = -12 \\ -5x_2 + 16x_3 + 3x_4 = -65 \\ 3x_3 + 13x_4 - 4x_5 = -18 \\ -3x_4 + 6x_5 = -24 \end{cases}$$

Решение. Прямой ход:

Вычислим прогоночные коэффициенты:

b_i - э-ты главной диагонали d_i - правая ч. ур-ния

a_i - э-ты поддиагонали

c_i - э-ты наддиагонали (i - соответствует номеру уравнения)

$$\alpha_1 = -\frac{c_1}{b_1} = -\frac{-6}{12} = 0{,}5; \quad \beta_1 = \frac{d_1}{b_1} = \frac{30}{12} = \frac{15}{6} = \frac{5}{2} = 2{,}5$$

$$y_2 = b_2 + a_2\alpha_1 = 8 - 4\cdot 0{,}5 = 8 - 2 = 6$$

$$\alpha_2 = -\frac{c_2}{y_2} = \frac{1}{6}; \quad \beta_2 = \frac{d_2 - a_2\beta_1}{y_2} = \frac{-12-(-4)\cdot 2{,}5}{6} = \frac{-12+10}{6} = -\frac{1}{3}$$

$$y_3 = b_3 + a_3\cdot\alpha_2 = 16 - 5\cdot\frac{1}{6} = 15\frac{1}{6} = \frac{91}{6}$$

$$\alpha_3 = \frac{-c_3}{y_3} = \frac{-3}{\frac{91}{6}} = \frac{-18}{91}, \quad \beta_3 = \frac{d_3 - a_3\beta_2}{y_3} = \frac{-65 - 5\cdot\frac{1}{3}}{91/6} = -\frac{400}{91}$$

$$y_4 = b_4 + a_4\alpha_3 = 13 + 3\cdot\frac{(-18)}{91} = \frac{1129}{91}$$

$$\alpha_4 = -\frac{c_4}{y_4} = \frac{4}{1129/91} = \frac{364}{1129}; \quad \beta_4 = \frac{d_4 - a_4\beta_3}{y_4} = \frac{-18 - 3\cdot\left(-\frac{400}{91}\right)}{364/1129} = -\frac{247251}{16562}$$

$$\beta_5 = \frac{d_5 - a_5\cdot\beta_4}{b_5 + a_5\alpha_4} = \frac{-24-(-3)\cdot\left(\frac{-247251}{16562}\right)}{6 - 3\cdot 364/1129} = -\frac{452729}{33124}$$

Обратный ход:

$$x_5 = \beta_5 = -13{,}667703$$

$$x_4 = \alpha_4 x_5 + \beta_4 = -19{,}335406$$

$$x_3 = \alpha_3 x_4 + \beta_3 = -0{,}571019$$

$$x_2 = \alpha_2 x_3 + \beta_2 = -0{,}428503$$

$$x_1 = \alpha_1 x_2 + \beta_1 = 2{,}285749$$

Ответ: $x = (2{,}285749; -0{,}428503; -0{,}571019; -19{,}335406; -13{,}667703)$

Л.р №9.14

Вычислить нормы $\|\circ\|_1$, $\|\circ\|_E$, $\|\circ\|_\infty$ матрицы A и нормы вектора $\vec{b}$, считая, что компоненты вектора $\vec{b}$ получены в результате округ

$$A = \begin{pmatrix} 1,591 & -0,75 & -2,093 \\ -1,863 & -2,892 & 2,026 \\ -2,742 & -0,221 & -2,502 \end{pmatrix} \quad \vec{b} = \begin{pmatrix} -2,851 \\ -0,536 \\ 2.6 \end{pmatrix}$$

Решение. для $\vec{b}$:

$$\|b\|_1 = |-2,851| + |-0,536| + |2,6| = 5,987$$

$$\|b\|_2 = \sqrt{(-2,851)^2 + (-0,536)^2 + 2,6^2} = 3,896$$

$$\|b\|_\infty = \max\{|-2,851|; |-0,534|, |2,6|\} = 2,851$$

$$\text{в } \|b\|_1: \Delta b = |5\cdot 10^{-4}| + |5\cdot 10^{-4}| + |5\cdot 10^{-2}| = 0,051$$

$$\text{в } \|b\|_2: \Delta b = \sqrt{(5\cdot 10^{-2})^2 + 2\cdot(5\cdot 10^{-4})^2} \approx 0,050$$

$$\text{в } \|b\|_\infty: \Delta b = \max\{|5\cdot 10^{-2}|; |5\cdot 10^{-4}|; |5\cdot 10^{-4}|\} = 5\cdot 10^{-2}$$

$$\delta_1 b = \frac{0,051}{5,987} \approx 0,0085; \quad \delta_2 b = \frac{0,050}{3,896} \approx 0,013; \quad \delta_3 b = \frac{5\cdot 10^{-2}}{2,851} \approx 0,018$$

Для матрицы A:

$$\|A\|_1 = \max_{1\le j\le n} \sum_{i=1}^{n} |A_{ij}| = \max\{|1,591| + |-1,863| + |-2,442|; |-0,75| + |-2,892| + |-0,22$$

$$|-2,093| + |2,026| + |-2,502|\} = 6,621$$

$$\|A\|_E = \sqrt{\sum_{i=1}^{n} |A_{ij}|^2} = 6,10254$$

$$\|A\|_\infty = \max_{1\le i\le n} \sum_{j=1}^{n} |A_{ij}| = \max\{|1,591| + |-0,75| + |-2,093|; |-1,863| + |-2,892| + |2,026|,$$

$$|-2,742| + |-0,221| + |-2,502|\} = 6,481$$

Ответ: $\|b\|_1 = 5,987$; $\|b\|_2 = 3,896$; $\|b\|_\infty = 2,851$

$\delta_1 b = 0,0085$; $\delta_2 b = 0,013$, $\delta_3 b = 0,018$

$\|A\|_1 = 6,621$; $\|A\|_E = 6,10254$; $\|A\|_\infty = 6,481$

Т.р 11.14

$$A = \begin{pmatrix} 99 & -3 & -9 & -5 \\ 3 & 3 & 4 & 50 \\ -4 & 1 & 93 & -9 \\ -10 & 169 & 9 & 7 \end{pmatrix} \quad b = \begin{pmatrix} -628 \\ -376 \\ -369 \\ 821 \end{pmatrix} \Rightarrow \begin{cases} 99x_1 - 3x_2 - 9x_3 - 5x_4 = -628 \\ -10x_1 + 169x_2 + 9x_3 + 7x_4 = 821 \\ -4x_1 + x_2 + 93x_3 - 9x_4 = -369 \\ 3x_1 + 3x_2 + 4x_3 + 50x_4 = -375 \end{cases}$$

Решение

Метод Якоби

$$\begin{cases} x_1 = \dfrac{-628 + 3x_2 + 9x_3 + 5x_4}{99} \\ x_2 = \dfrac{821 + 10x_1 - 9x_3 - 7x_4}{169} \\ x_3 = \dfrac{-369 + 4x_1 - x_2 + 9x_4}{93} \\ x_4 = \dfrac{-376 - 3x_1 - 3x_2 - 4x_3}{50} \end{cases} \qquad B = \begin{pmatrix} 0 & \frac{3}{99} & \frac{9}{99} & \frac{5}{99} \\ \frac{10}{169} & 0 & -\frac{9}{169} & -\frac{7}{169} \\ \frac{4}{93} & -\frac{1}{93} & 0 & \frac{9}{93} \\ -\frac{3}{50} & -\frac{3}{50} & -\frac{4}{50} & 0 \end{pmatrix}$$

$$\|B\|_\infty = \max\left\{ \frac{3+9+5}{99}, \frac{10+9+7}{169}, \frac{4+1+9}{93}, \frac{10}{50} \right\} = 0,2 < 1$$ – метод сходится

$$x^{(0)} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$n=1$

$$\begin{cases} x_1 = -6,343 + 0,030 + 0,091 + 0,051 = -6,171 \\ x_2 = 4,858 + 0,059 + 0,053 - 0,041 = 4,823 \\ x_3 = -3,968 + 0,043 - 0,011 + 0,097 = -3,839 \\ x_4 = -7,52 - 0,06 - 0,06 - 0,08 = -7,72 \end{cases} \qquad x^{(1)} = \begin{pmatrix} -6,171 \\ 4,823 \\ -3,839 \\ -7,72 \end{pmatrix}$$

$n=2$

$$\begin{cases} x_1 = -6,343 + 0,03\cdot(4,823) + 0,091\cdot(-3,839) + 0,051\cdot(-7,72) = -6,941 \\ x_2 = 4,858 + 0,059\cdot(-6,171) - 0,053\cdot(-3,839) - 0,041\cdot(-7,72) = 5,014 \\ x_3 = -3,968 + 0,043\cdot(-6,171) - 0,011\cdot(4,823) + 0,097\cdot(-7,72) = -5,035 \\ x_4 = -7,52 - 0,06(-6,171) - 0,06\cdot(4,823) - 0,08\cdot(-3,839) = -7,132 \end{cases} \qquad x^{(2)} = \begin{pmatrix} -6,941 \\ 5,014 \\ -5,035 \\ -7,132 \end{pmatrix}$$

$n=3$

$$\begin{cases} x_1 = -6,343 + 0,03(5,014) + 0,091\cdot(-5,035) + 0,051(-7,132) = -7,014 \\ x_2 = 4,858 + 0,059\cdot(-6,941) - 0,053\cdot(-5,035) - 0,041\cdot(-7,132) = 5,008 \\ x_3 = -3,968 + 0,043\cdot(-6,941) - 0,011\cdot(5,014) + 0,097\cdot(-7,132) = -5,013 \\ x_4 = -7,52 - 0,06\cdot(-6,941) - 0,06\cdot(5,014) - 0,08\cdot(-5,035) = -7,002 \end{cases} \qquad x^{(3)} = \begin{pmatrix} -7,014 \\ 5,008 \\ -5,013 \\ -7,002 \end{pmatrix}$$

Норма невязки:

$$r^{(0)} = b - Ax^{(0)}$$

$$r^{0} = \begin{pmatrix} -628 \\ -376 \\ -369 \\ 821 \end{pmatrix} - \begin{pmatrix} 99 & -3 & -9 & -5 \\ 3 & 3 & 4 & 50 \\ -4 & 1 & 93 & -9 \\ -10 & 169 & 9 & 7 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -628 \\ -376 \\ -369 \\ 821 \end{pmatrix} - \begin{pmatrix} 82 \\ 60 \\ 81 \\ 175 \end{pmatrix} = \begin{pmatrix} -710 \\ -436 \\ -450 \\ 646 \end{pmatrix}$$

$$r^{(3)} = \begin{pmatrix} -628 \\ -376 \\ -369 \\ 821 \end{pmatrix} - \begin{pmatrix} 99 & -3 & -9 & -5 \\ 3 & 3 & 4 & 50 \\ -4 & 1 & 93 & -9 \\ -10 & 169 & 9 & 7 \end{pmatrix} \cdot \begin{pmatrix} -7,014 \\ 5,008 \\ -5,013 \\ -7,002 \end{pmatrix} = \begin{pmatrix} 1,283 \\ 0,17 \\ 1,127 \\ -1,361 \end{pmatrix}$$

$$\|r^{(0)}\|_1 = 2242$$

$$\|r^{(3)}\|_1 = 3,941$$

$$\frac{\|r^{0}\|_1}{\|r^{(3)}\|_1} = \frac{2242}{3,941} = 568,9$$

Апостериорная оценка $\|x^{(3)} - \bar{x}\|_\infty \le \frac{\|B\|_\infty}{1-\|B\|_\infty} \cdot \|x^{(3)} - x^{(2)}\|$

$$x^{(3)} - x^{(2)} = \begin{pmatrix} -0,073 \\ -0,006 \\ 0,022 \\ 0,130 \end{pmatrix} \qquad \|x^{(3)} - x^{(2)}\| = 0,13 \qquad \bar{x} = \begin{pmatrix} -7 \\ 5 \\ -5 \\ -7 \end{pmatrix}$$

$$x^{(3)} - \bar{x} = \begin{pmatrix} -0,014 \\ 0,008 \\ -0,013 \\ -0,002 \end{pmatrix} \qquad \|x^{(3)} - \bar{x}\| = 0,014$$

$0,014 \le \frac{0,2}{1-0,2} \cdot 0,13 = 0,0325 \Rightarrow$ оценка верна.

Метод Зейделя

$$B_1 = \begin{pmatrix} 0 & \frac{3}{99} & \frac{9}{99} & \frac{5}{99} \\ 0 & 0 & -\frac{9}{169} & -\frac{7}{169} \\ 0 & 0 & 0 & \frac{9}{93} \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad B_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{10}{169} & 0 & 0 & 0 \\ \frac{4}{93} & -\frac{1}{93} & 0 & 0 \\ -\frac{3}{50} & -\frac{3}{50} & -\frac{4}{50} & 0 \end{pmatrix}$$

$$\|B_1\|_\infty = \max\{0,171;\ 0,095;\ 0,097\} = 0,171$$

$$\|B_2\|_\infty = \max\left\{\frac{10}{169};\ \frac{5}{93};\ \frac{10}{50}\right\} = 0,2$$

$\|B_1\|_\infty + \|B_2\|_\infty = 0,171 + 0,2 = 0,371 < 1 \Rightarrow$ метод сходится

$$x^{(n+1)} = B_1 \cdot x^{(n+1)} + B_2 x^{(n)} + C$$

$$x^{(0)} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$n=1$

$$\begin{cases} x_1 = -6,343 + 0,030 + 0,091 + 0,051 = -6,171 \\ x_2 = 0,059(-6,171) - 0,053 - 0,041 + 4,858 = 4,4 \\ x_3 = -3,968 + 0,043(-6,171) - 0,011 \cdot (4,4) + 0,097 = -4,185 \\ x_4 = -7,52 - 0,06(-6,171) - 0,06(4,4) - 0,08(-4,185) = -7,079 \end{cases} \quad x^{(1)} = \begin{pmatrix} -6,171 \\ 4,4 \\ -4,185 \\ -7,079 \end{pmatrix}$$

$n=2$

$$\begin{cases} x_1 = -6,343 + 0,03(4,4) + 0,091(-4,185) + 0,051(-7,079) = -6,953 \\ x_2 = 4,858 + 0,059(-6,953) - 0,053(-4,185) - 0,041(-7,079) = 4,96 \\ x_3 = -3,968 + 0,043(-6,953) - 0,011(4,96) + 0,097(-7,079) = -5,008 \\ x_4 = -7,52 - 0,06(-6,953) - 0,06(4,96) - 0,08(-5,008) = -6,999 \end{cases}$$

$$x^{(2)} = (-6,953; 4,96; -5,008; -6,999)$$

$n=3$

$$\begin{cases} x_1 = -6,343 + 0,03(4,96) + 0,091(-5,008) + 0,051(-6,999) = -7,007 \\ x_2 = 4,858 + 0,059(-7,007) - 0,053(-5,008) - 0,041(-6,999) = 4,997 \\ x_3 = -3,968 + 0,043(-7,007) - 0,011(4,997) + 0,097(-6,999) = -5,003 \\ x_4 = -7,52 - 0,06(-7,007) - 0,06(4,997) + 0,08(-5,003) = -6,999 \end{cases}$$

$$x^{(3)} = (-7,007; 4,997; -5,003; -6,999)$$

Норма невязки:

$$r^{(0)} = \begin{pmatrix} -710 \\ -436 \\ -430 \\ 646 \end{pmatrix} \quad r^{(3)} = \begin{pmatrix} -628 \\ -376 \\ -369 \\ 821 \end{pmatrix} - \begin{pmatrix} 99 & -3 & -9 & -5 \\ 3 & 3 & 4 & 50 \\ -4 & 1 & 93 & -9 \\ 70 & 169 & 9 & 7 \end{pmatrix} \cdot \begin{pmatrix} -7,007 \\ 4,997 \\ -5,003 \\ -6,999 \end{pmatrix} = \begin{pmatrix} 0,662 \\ -0,008 \\ 0,263 \\ 0,457 \end{pmatrix}$$

$$\|r^{(0)}\|_1 = 2242 \quad \|r^{(3)}\|_1 = 1,39$$

$$\frac{\|r^{(0)}\|}{\|r^{3}\|} = \frac{2242}{1,39} \approx 1613$$

Апостериорная оценка: $\|x^{(3)}-\bar{x}\|_\infty \le \dfrac{\|B_2\|_\infty}{1-\|B\|_\infty}\cdot\|x^{(3)}-x^{(2)}\|_\infty$

$$x^{(3)}-x^{(2)}=\begin{pmatrix}-0{,}054\\0{,}037\\0{,}005\\0\end{pmatrix}\qquad \|x^{(3)}-x^{(2)}\|_\infty=0{,}054$$

$$\bar{x}=\begin{pmatrix}-7\\5\\-5\\-7\end{pmatrix}$$

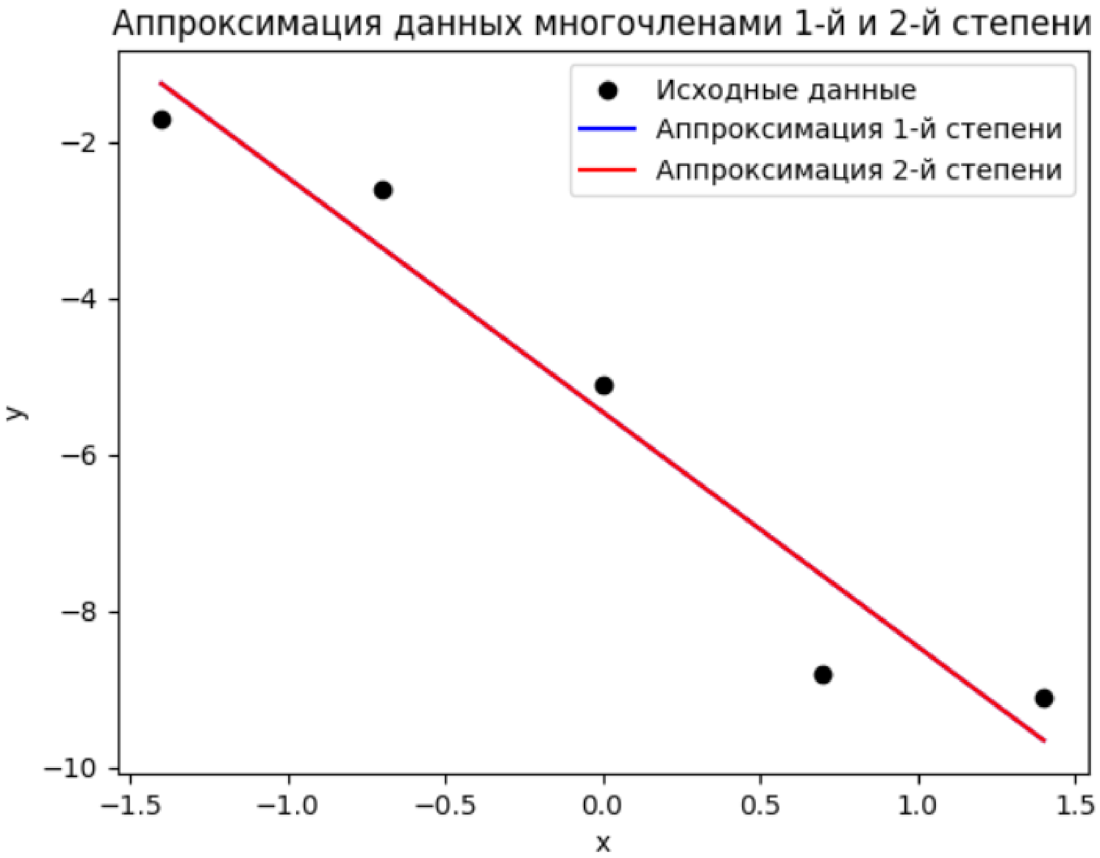
$$x^{(3)}-\bar{x}=\begin{pmatrix}-0{,}004\\-0{,}003\\-0{,}003\\0{,}001\end{pmatrix}\qquad \|x^{(3)}-\bar{x}\|=0{,}007$$

$$0{,}007\le\frac{0{,}2}{1-0{,}2}\cdot0{,}054=0{,}0135 \Rightarrow \text{оценка верна}$$

Ответы:

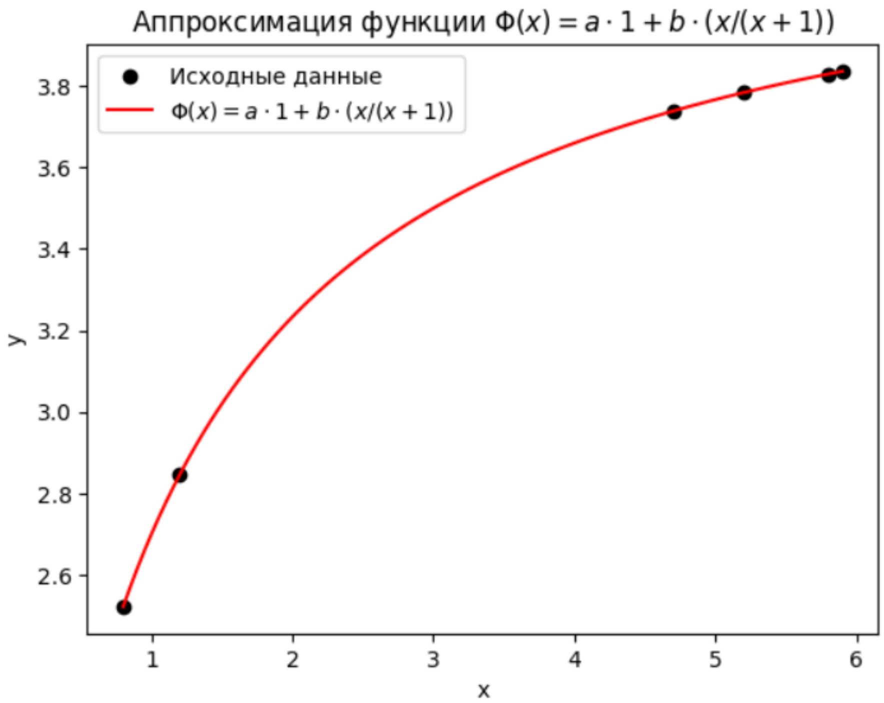
Задание 13

```
Формула для аппроксимации многочленом 1-й степени: y = -3.0000x + -5.4600
Формула для аппроксимации многочленом 2-й степени: y = -0.0000x^2 + -3.0000x + -5.4600
Среднеквадратичная ошибка для аппроксимации многочленом 1-й степени:0.5504
Среднеквадратичная ошибка для аппроксимации многочленом 2-й степени:0.5504
```



Задание 14

```
Коэффициенты аппроксимации: a = 1.0994, b = 3.2007
Среднеквадратичная ошибка: 0.0000
Исходные значения y: [2.522 2.845 3.739 3.784 3.829 3.836]
Аппроксимированные значения y_approx: [2.52189239 2.84519572 3.73853385 3.78381827 3.82936907 3.83619069]
```



Задание 15

```
Интерполяционный многочлен Лагранжа: 5*x**3/3 - 45*x**2/2 + 593*x/6 - 139
Интерполяционный многочлен Ньютона: 1.66666666666667*x**3 - 22.5*x**2 + 98.8333333333333*x - 139.0
Приближенное значение функции в точке 3.6 по методу Лагранжа: 2.95999999999998
Приближенное значение функции в точке 3.6 по методу Ньютона: 2.96000000000004
```

Задание 16

```
Значение интерполяционного многочлена первой степени в точке x~ = 0.6: 2.15
Значение интерполяционного многочлена второй степени в точке x~ = 0.6: 2.1625
Значение интерполяционного многочлена третьей степени в точке x~ = 0.6: 2.14375
Погрешность для многочлена первой степени: 0.006250000000000089
Погрешность для многочлена второй степени: 0.018750000000000266
Погрешность для многочлена третьей степени: 12.6
```

Задание 20

```
Метод центральных прямоугольников с h = 0.4: 11.989961
Метод трапеций с h = 0.4: 11.990036
Метод трапеций с h = 0.2: 11.989998
Оценка погрешности по правилу Рунге для метода трапеций: 0.000013
Уточненный результат для метода трапеций: 11.989986
Метод Симпсона с h = 0.4: 11.989986
```

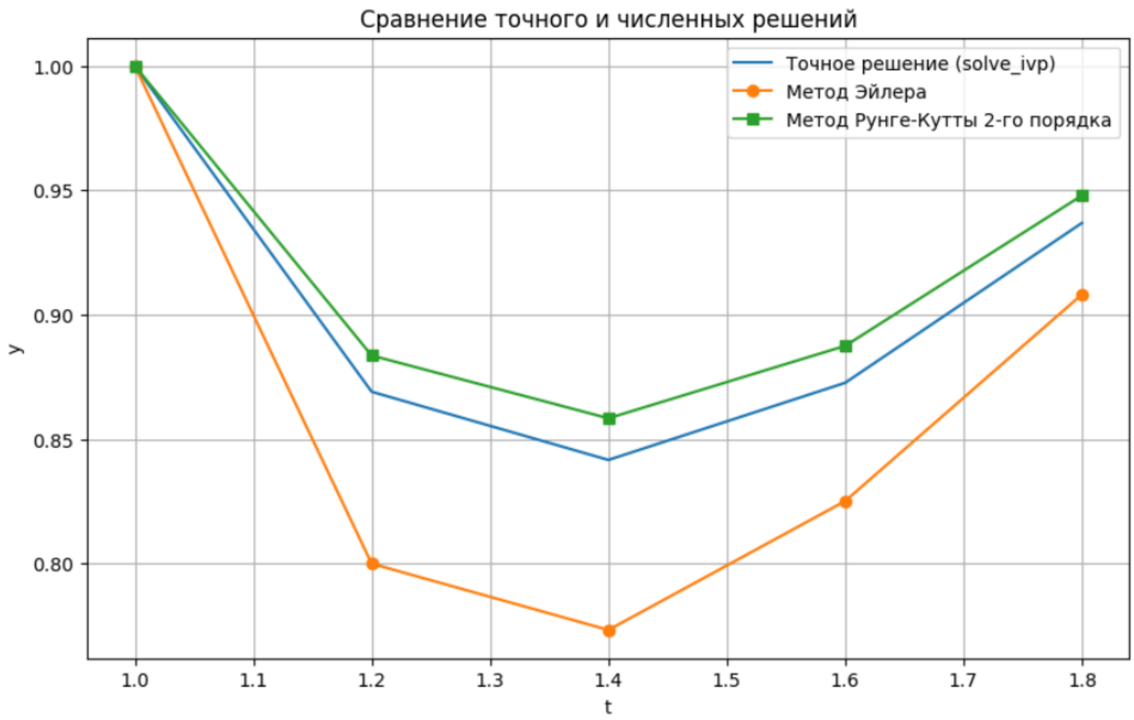
Задание 21

```
Точное значение интеграла: 1.59069166666667
Оптимальный шаг h для достижения точности ε = 0.01: 0.1
Приближенное значение интеграла методом трапеций с шагом h = 0.1: 1.593420
Погрешность приближенного значения: 0.002728
```

Задание 23

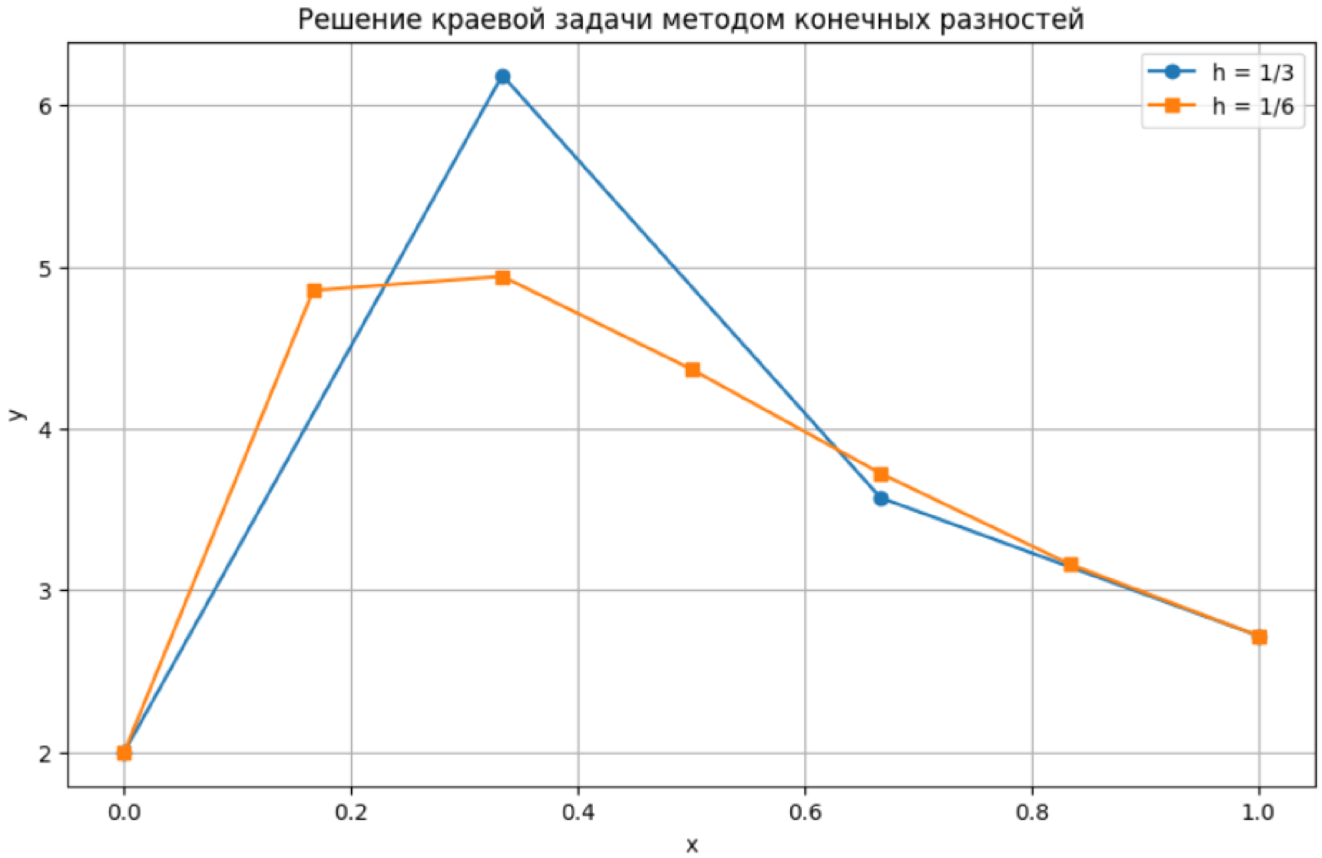
```
Центральная разностная производная в точке x0 = -0.05: 3.440000
Левая разностная производная в точке x0 = -0.05: 3.887500
Вторая разностная производная в точке x0 = -0.05: -8.950000
Точное значение первой производной в точке x0 = -0.05: 3.476000
Точное значение второй производной в точке x0 = -0.05: -9.010000
Погрешность центральной разностной производной: 0.036000
Погрешность левой разностной производной: 0.411500
Погрешность второй разностной производной: 0.060000
```

Задание 24




```
Погрешность метода Эйлера: [0.         0.04181818 0.03852308 0.0249262  0.01368224]
Погрешность метода Рунге-Кутты 2-го порядка: [0.         0.0037373  0.00458153 0.00403796 0.00307207]
```

Задание 27




```
Погрешность по правилу Рунге: [2.14643118e-15 4.14694419e-01 5.02468387e-02 0.00000000e+00]
```

Выполнение:

Задание 13

```
import numpy as np
import matplotlib.pyplot as plt
# Данные для 14 варианта
x = np.array([-1.4, -0.7, 0, 0.7, 1.4])
y = np.array([-1.7, -2.6, -5.1, -8.8, -9.1])
# Аппроксимация многочленом первой степени
coeffs_1 = np.polyfit(x, y, 1)
poly_1 = np.poly1d(coeffs_1)
y_approx_1 = poly_1(x)
# Аппроксимация многочленом второй степени
coeffs_2 = np.polyfit(x, y, 2)
poly_2 = np.poly1d(coeffs_2)
y_approx_2 = poly_2(x)
# Среднеквадратичная ошибка
mse_1 = np.mean((y - y_approx_1)**2)
mse_2 = np.mean((y - y_approx_2)**2)
# Вывод формул многочленов
print(f'Формула для аппроксимации многочленом 1-й степени: y =
{coeffs_1[0]:.4f}x + {coeffs_1[1]:.4f}')
print(f'Формула для аппроксимации многочленом 2-й степени: y =
```

```
{coeffs_2[0]:.4f}x^2 + {coeffs_2[1]:.4f}x + {coeffs_2[2]:.4f}')
# Вывод среднеквадратичных ошибок
print(f'Среднеквадратичная ошибка для аппроксимации многочленом 1-й
степени:{mse_1:.4f}')
print(f'Среднеквадратичная ошибка для аппроксимации многочленом 2-й
степени:{mse_2:.4f}')
# Построение графиков
plt.scatter(x, y, color='black', label='Исходные данные')
x_line = np.linspace(min(x), max(x), 100)
plt.plot(x_line, poly_1(x_line), label='Аппроксимация 1-й степени', color='blue')
plt.plot(x_line, poly_2(x_line), label='Аппроксимация 2-й степени', color='red')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.title('Аппроксимация данных многочленами 1-й и 2-й степени')
plt.show()
```

Задание 14

```
import numpy as np
import matplotlib.pyplot as plt

# Данные для 14 варианта
x = np.array([0.8, 1.2, 4.7, 5.2, 5.8, 5.9])
y = np.array([2.522, 2.845, 3.739, 3.784, 3.829, 3.836])

# Построение матрицы для метода наименьших квадратов
A = np.vstack([np.ones_like(x), x/(x+1)]).T
coeffs, residuals, _, _ = np.linalg.lstsq(A, y, rcond=None)

# Коэффициенты аппроксимации
a, b = coeffs

# Функция аппроксимации
def phi(x):
    return a * np.ones_like(x) + b * x/(x+1)

# Вычисление значений аппроксимированной функции
y_approx = phi(x)

# Среднеквадратичная ошибка
mse = np.mean((y - y_approx) ** 2)

print(f'Коэффициенты аппроксимации: a = {a:.4f}, b = {b:.4f}')
print(f'Среднеквадратичная ошибка: {mse:.4f}')

# Проверка, что значения y_approx не совпадают идеально с y
print(f'Исходные значения y: {y}')
```

```
print(f'Аппроксимированные значения y_approx: {y_approx}')

# Построение графиков
plt.scatter(x, y, color='black', label='Исходные данные')
x_line = np.linspace(min(x), max(x), 100)
plt.plot(x_line, phi(x_line), label='$\Phi(x) = a \cdot 1 + b \cdot (x/(x+1))$', color='red')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.title('Аппроксимация функции $\Phi(x) = a \cdot 1 + b \cdot (x/(x+1))$')
plt.show()
```

Задание 15

```
import numpy as np
import sympy as sp

# Данные для 14 варианта
x = np.array([3, 4, 5, 6])
y = np.array([0, 3, 1, 4])
x_approx = 3.6

# Интерполяционный многочлен Лагранжа
def lagrange_polynomial(x, y):
    n = len(x)
    X = sp.symbols('x')
    L = 0
    for i in range(n):
        term = y[i]
        for j in range(n):
            if j != i:
                term *= (X - x[j]) / (x[i] - x[j])
        L += term
    return sp.simplify(L)

# Интерполяционный многочлен Ньютона
def newton_polynomial(x, y):
    def divided_differences(x, y):
        n = len(y)
        coef = np.zeros([n, n])
        coef[:,0] = y
        for j in range(1,n):
            for i in range(n-j):
                coef[i][j] = (coef[i+1][j-1] - coef[i][j-1]) / (x[i+j] - x[i])
        return coef[0, :]

    coef = divided_differences(x, y)
    X = sp.symbols('x')
```

```
    n = len(coef) - 1
    P = coef[0]
    for k in range(1, n + 1):
        term = coef[k]
        for j in range(k):
            term *= (X - x[j])
        P += term
    return sp.simplify(P)

# Вычисление многочленов
L_poly = lagrange_polynomial(x, y)
N_poly = newton_polynomial(x, y)

# Вывод многочленов
print(f'Интерполяционный многочлен Лагранжа: {L_poly}')
print(f'Интерполяционный многочлен Ньютона: {N_poly}')

# Вычисление значений
y_lagrange = L_poly.subs('x', x_approx)
y_newton = N_poly.subs('x', x_approx)
print(f'Приближенное значение функции в точке {x_approx} по методу Лагранжа: {y_lagrange}')
print(f'Приближенное значение функции в точке {x_approx} по методу Ньютона: {y_newton}')
```

Задание 16

```
import numpy as np

# Данные
x = np.array([0, 0.4, 0.8, 1.6, 2])
y = np.array([1, 1.8, 2.5, 4.8, 6.4])
x_tilde = 0.6

# Сортировка данных по расстоянию до x_tilde
distances = np.abs(x - x_tilde)
sorted_indices = np.argsort(distances)
x = x[sorted_indices]
y = y[sorted_indices]

# Вычисление разделенных разностей для многочлена Ньютона
def divided_differences(x, y):
    n = len(y)
    coef = np.zeros([n, n])
    coef[:, 0] = y
    for j in range(1, n):
        for i in range(n - j):
            coef[i][j] = (coef[i + 1][j - 1] - coef[i][j - 1]) / (x[i + j] - x[i])
```

```
    return coef[0, :]

# Интерполяционный многочлен Ньютона
def newton_polynomial(x, y, x_tilde):
    coef = divided_differences(x, y)
    n = len(x) - 1
    p = coef[n]
    for k in range(1, n + 1):
        p = coef[n - k] + (x_tilde - x[n - k]) * p
    return p

# Многочлены Ньютона первой, второй и третьей степени
P1 = newton_polynomial(x[:2], y[:2], x_tilde)
P2 = newton_polynomial(x[:3], y[:3], x_tilde)
P3 = newton_polynomial(x[:4], y[:4], x_tilde)

# Печать результатов
print(f"Значение интерполяционного многочлена первой степени в точке x~ =
{x_tilde}: {P1}")
print(f"Значение интерполяционного многочлена второй степени в точке x~ =
{x_tilde}: {P2}")
print(f"Значение интерполяционного многочлена третьей степени в точке x~ =
{x_tilde}: {P3}")

# Оценка погрешности
def estimate_error(actual, interpolated):
    return abs(actual - interpolated)

# Предположим, что значение функции в x_tilde можно оценить с помощью
интерполяции
# Для оценки погрешности используем значение P3 как более точное
error_P1 = estimate_error(P3, P1)
error_P2 = estimate_error(P3, P2)

# Если известное точное значение функции в x_tilde неизвестно, оценим
погрешность для многочлена третьей степени
# как наибольшее отклонение от известных значений
actual_values = [8.3, 10.5, 6.4, 15.9, 19]
predicted_values = [newton_polynomial(x, y, xi) for xi in x]
error_P3 = max([estimate_error(av, pv) for av, pv in zip(actual_values,
predicted_values)])

print(f"Погрешность для многочлена первой степени: {error_P1}")
print(f"Погрешность для многочлена второй степени: {error_P2}")
print(f"Погрешность для многочлена третьей степени: {error_P3}")
```

Задание 20

```
import numpy as np

# Функция для интегрирования
def f(x):
    return x * np.arctan(x)

# Параметры задачи
a = 4.6
b = 6.2

# Метод центральных прямоугольников
def central_rectangles(f, a, b, h):
    n = int((b - a) / h)
    result = 0.0
    for i in range(n):
        x_i = a + i * h
        x_i_star = x_i + h / 2
        result += f(x_i_star)
    result *= h
    return result

# Метод трапеций
def trapezoidal(f, a, b, h):
    n = int((b - a) / h)
    result = (f(a) + f(b)) / 2
    for i in range(1, n):
        result += f(a + i * h)
    result *= h
    return result

# Метод Симпсона
def simpson(f, a, b, h):
    n = int((b - a) / h)
    if n % 2 != 0:  # n должно быть четным для метода Симпсона
        n += 1
    h = (b - a) / n
    result = f(a) + f(b)
    for i in range(1, n):
        x_i = a + i * h
        if i % 2 == 0:
            result += 2 * f(x_i)
        else:
            result += 4 * f(x_i)
    result *= h / 3
    return result

# Вычисление интегралов
h1 = 0.4
```

```
h2 = 0.2
I_central_rectangles = central_rectangles(f, a, b, h1)
I_trapezoidal_h1 = trapezoidal(f, a, b, h1)
I_trapezoidal_h2 = trapezoidal(f, a, b, h2)
I_simpson = simpson(f, a, b, h1)

# Оценка погрешности по правилу Рунге для метода трапеций
def runge_error(I_h1, I_h2, k):
    return abs(I_h2 - I_h1) / (2 ** k - 1)

k = 2  # Порядок метода трапеций
error_trapezoidal = runge_error(I_trapezoidal_h1, I_trapezoidal_h2, k)

# Уточненный результат для метода трапеций
I_trapezoidal_refined = I_trapezoidal_h2 + (I_trapezoidal_h2 - I_trapezoidal_h1) / (2 **
k - 1)

# Печать результатов
print(f"Метод центральных прямоугольников с h = 0.4: {I_central_rectangles:.6f}")
print(f"Метод трапеций с h = 0.4: {I_trapezoidal_h1:.6f}")
print(f"Метод трапеций с h = 0.2: {I_trapezoidal_h2:.6f}")
print(f"Оценка погрешности по правилу Рунге для метода трапеций:
{error_trapezoidal:.6f}")
print(f"Уточненный результат для метода трапеций: {I_trapezoidal_refined:.6f}")
print(f"Метод Симпсона с h = 0.4: {I_simpson:.6f}")
```

Задание 21

```
import numpy as np
from sympy import symbols, diff, integrate

# Заданные параметры
a = -0.3
b = 0.2
c0 = 3
c1 = -2
c2 = 4
c3 = 3
c4 = -2

# Функция для интегрирования
def f(x):
    return c0 + c1*x + c2*x**2 + c3*x**3 + c4*x**4

# Точное значение интеграла с использованием SymPy
x = symbols('x')
exact_integral = integrate(c0 + c1*x + c2*x**2 + c3*x**3 + c4*x**4, (x, a, b))
exact_value = exact_integral.evalf()
```

```
print(f"Точное значение интеграла: {exact_value}")

# Метод трапеций
def trapezoidal(f, a, b, h):
    n = int((b - a) / h)
    result = (f(a) + f(b)) / 2
    for i in range(1, n):
        result += f(a + i * h)
    result *= h
    return result

# Оценка погрешности априорно по второй производной
x = symbols('x')
polynomial = c0 + c1*x + c2*x**2 + c3*x**3 + c4*x**4
second_derivative = diff(polynomial, x, x)
max_second_derivative = max(abs(second_derivative.subs(x, a)),
abs(second_derivative.subs(x, b)))

# Априорная оценка погрешности
def trapezoidal_error(b, a, h, max_second_derivative):
    return ((b - a) * h**2 / 12) * max_second_derivative

# Подбор шага h для достижения погрешности не более 0.01
epsilon = 0.01
h = 0.1  # Начальное значение шага
error = trapezoidal_error(b, a, h, max_second_derivative)
while error > epsilon:
    h /= 2
    error = trapezoidal_error(b, a, h, max_second_derivative)

# Вычисление интеграла с найденным шагом
I_trapezoidal = trapezoidal(f, a, b, h)

# Печать результатов
print(f"Оптимальный шаг h для достижения точности ε = {epsilon}: {h}")
print(f"Приближенное значение интеграла методом трапеций с шагом h = {h}:
{I_trapezoidal:.6f}")
print(f"Погрешность приближенного значения: {abs(exact_value -
I_trapezoidal):.6f}")
```

Задание 23

```
import numpy as np
from sympy import symbols, diff

# Заданные параметры
a = -0.3
b = 0.2
```

```
h = 0.1
x0 = round((a + b) / 2, 2)

# Функция для интегрирования
def f(x):
    return 3 + 3*x - 5*x**2 - 3*x**3 + 3*x**4

# Центральная разностная производная
def central_difference(f, x, h):
    return (f(x + h) - f(x - h)) / (2 * h)

# Левая разностная производная
def left_difference(f, x, h):
    return (f(x) - f(x - h)) / h

# Вторая разностная производная
def second_difference(f, x, h):
    return (f(x + h) - 2 * f(x) + f(x - h)) / h**2

# Вычисление производных
f_prime_central = central_difference(f, x0, h)
f_prime_left = left_difference(f, x0, h)
f_double_prime = second_difference(f, x0, h)

# Вычисление точного значения производных с использованием SymPy
x = symbols('x')
polynomial = 3 + 3*x - 5*x**2 - 3*x**3 + 3*x**4
exact_f_prime = diff(polynomial, x)
exact_f_double_prime = diff(polynomial, x, x)
exact_f_prime_value = exact_f_prime.subs(x, x0).evalf()
exact_f_double_prime_value = exact_f_double_prime.subs(x, x0).evalf()

# Печать результатов
print(f"Центральная разностная производная в точке x0 = {x0}:
{f_prime_central:.6f}")
print(f"Левая разностная производная в точке x0 = {x0}: {f_prime_left:.6f}")
print(f"Вторая разностная производная в точке x0 = {x0}: {f_double_prime:.6f}")
print(f"Точное значение первой производной в точке x0 = {x0}:
{exact_f_prime_value:.6f}")
print(f"Точное значение второй производной в точке x0 = {x0}:
{exact_f_double_prime_value:.6f}")

# Сравнение точных и приближенных значений
error_central = abs(exact_f_prime_value - f_prime_central)
error_left = abs(exact_f_prime_value - f_prime_left)
error_second = abs(exact_f_double_prime_value - f_double_prime)
print(f"Погрешность центральной разностной производной: {error_central:.6f}")
print(f"Погрешность левой разностной производной: {error_left:.6f}")
```

```
print(f"Погрешность второй разностной производной: {error_second:.6f}")
```

Задание 24

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp

# Заданные параметры
t0 = 1
y0 = 1
h = 0.2
t_end = t0 + 0.8

# Определение функции
def f(t, y):
    return -((4*t-1)/t)*y + 2*t

# Метод Эйлера
def euler_method(f, t0, y0, h, t_end):
    t_values = np.arange(t0, t_end + h, h)
    y_values = np.zeros(len(t_values))
    y_values[0] = y0
    for i in range(1, len(t_values)):
        y_values[i] = y_values[i - 1] + h * f(t_values[i - 1], y_values[i - 1])
    return t_values, y_values

# Метод Рунге-Кутты 2-го порядка
def runge_kutta_2nd_order(f, t0, y0, h, t_end):
    t_values = np.arange(t0, t_end + h, h)
    y_values = np.zeros(len(t_values))
    y_values[0] = y0
    for i in range(1, len(t_values)):
        k1 = f(t_values[i - 1], y_values[i - 1])
        k2 = f(t_values[i - 1] + h / 2, y_values[i - 1] + h / 2 * k1)
        y_values[i] = y_values[i - 1] + h * k2
    return t_values, y_values

# Решение точное с использованием solve_ivp
sol = solve_ivp(f, [t0, t_end], [y0], method='RK45', t_eval=np.arange(t0, t_end + h, h))

# Решения методами Эйлера и Рунге-Кутты
t_euler, y_euler = euler_method(f, t0, y0, h, t_end)
t_rk2, y_rk2 = runge_kutta_2nd_order(f, t0, y0, h, t_end)

# Решения для шага h/2
_, y_euler_half = euler_method(f, t0, y0, h/2, t_end)
_, y_rk2_half = runge_kutta_2nd_order(f, t0, y0, h/2, t_end)
```

```
# Оценка погрешности по правилу Рунге
def runge_error(y_half, y_full, k):
    return np.abs(y_half[::2] - y_full) / (2**k - 1)

# Погрешности
error_euler = runge_error(y_euler_half, y_euler, 1)
error_rk2 = runge_error(y_rk2_half, y_rk2, 2)

# Графики решений
plt.figure(figsize=(10, 6))
plt.plot(sol.t, sol.y[0], label='Точное решение (solve_ivp)')
plt.plot(t_euler, y_euler, 'o-', label='Метод Эйлера')
plt.plot(t_rk2, y_rk2, 's-', label='Метод Рунге-Кутты 2-го порядка')
plt.xlabel('t')
plt.ylabel('y')
plt.legend()
plt.title('Сравнение точного и численных решений')
plt.grid(True)
plt.show()

# Печать результатов
print(f"Погрешность метода Эйлера: {error_euler}")
print(f"Погрешность метода Рунге-Кутты 2-го порядка: {error_rk2}")
```

Задание 27

```
import numpy as np
import matplotlib.pyplot as plt

# Заданные параметры
a = 0
b = 1
y_a = 2
y_b = np.exp(1)

# Функции q(x) и f(x)
def q(x):
    return np.exp(2)

def f(x):
    return np.exp(2*x)

# Метод конечных разностей
def finite_difference_method(a, b, y_a, y_b, q, f, h):
    n = int((b - a) / h)
    x = np.linspace(a, b, n + 1)
    A = np.zeros((n + 1, n + 1))
```

```
    B = np.zeros(n + 1)
    A[0, 0] = 1
    A[n, n] = 1
    B[0] = y_a
    B[n] = y_b
    for i in range(1, n):
        A[i, i - 1] = 1 / h ** 2 - q(x[i]) / (2 * h)
        A[i, i] = -2 / h ** 2 + q(x[i])
        A[i, i + 1] = 1 / h ** 2 + q(x[i]) / (2 * h)
        B[i] = f(x[i])
    y = np.linalg.solve(A, B)
    return x, y

# Вычисление решений
h1 = 1 / 3
h2 = 1 / 6
x1, y1 = finite_difference_method(a, b, y_a, y_b, q, f, h1)
x2, y2 = finite_difference_method(a, b, y_a, y_b, q, f, h2)

# Оценка погрешности по правилу Рунге
def runge_error(y1, y2, k):
    return np.abs(y1 - y2[::2]) / (2 ** k - 1)

error = runge_error(y1, y2, 2)

# Графики решений
plt.figure(figsize=(10, 6))
plt.plot(x1, y1, 'o-', label='h = 1/3')
plt.plot(x2, y2, 's-', label='h = 1/6')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.title('Решение краевой задачи методом конечных разностей')
plt.grid(True)
plt.show()

# Печать результатов
print(f"Погрешность по правилу Рунге: {error}")
```