Лабораторная работа №1

Ларин Егор. 4 группа 3 курс

23 сентября 2022 г.

Теория

Задача

$$u' = \frac{u \ln u}{x}, u(1) = e, x \in [0, 1]\,.$$

Точное решение данного уравнения $u(x) = e^x$. Для нахождения численного решения введем сетку узлов $\omega_h = \{x_i = 1 + i * h : i = \overline{0, N}, N = \frac{(b-a)}{N}\}$

Неявный метод трапеций

$$y_{i+1}^0 = y_i$$

$$y_{i+1}^{k+1} = y_{i+1}^k - \frac{y_{i+1}^k - y_i - \frac{1}{2}h(f(x_i, y_i), f(x_{i+1}, y_{i+1}^k))}{1 - \frac{1}{2}hf_u(x_{i+1}, y_{i+1}^k)}$$

Условие остановки итерационного процесса Ньютона:

$$\left|y_i^k - y_i^{k+1}\right| < \varepsilon = 10^{-6}$$

Явный метод Рунге-Кутта 4-го порядка

$$q = 3, y_{i+1} = y_i + \frac{1}{6}h(\varphi_0 + 2\varphi_1 + 2\varphi_2 + \varphi_3)$$

$$\begin{cases} \varphi_0 = f(x_i, y_i) \\ \varphi_1 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}h\varphi_0) \\ \varphi_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}h\varphi_1) \\ \varphi_3 = f(x_i, y_i + h\varphi_2) \end{cases}$$

Явный метод средних прямоугольников

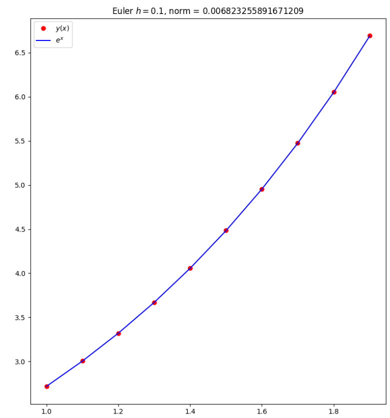
$$y_{i+1} = y_i + hf(x_i + \frac{h}{2}, y_i + \frac{h}{2}f(x_i, y_i)), y_0 = u(1)$$

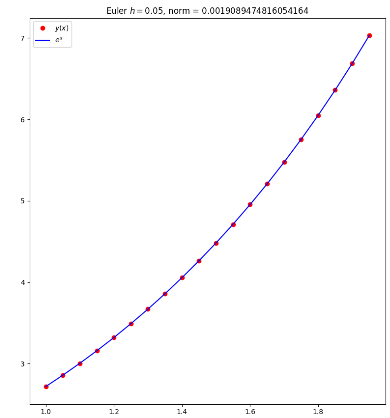
Листинг кода

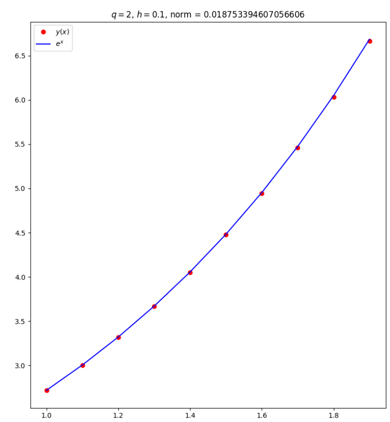
```python
from math import ceil, exp, log
from matplotlib import pyplot as plt

a = 1.
b = 2.
initial = exp(a)
f = lambda x, y: y * log(y) / x
u = lambda x: exp(x)
dudu = lambda x, y: (log(y) + 1) / x
eps = 0.000001

def runge(h, ass, alpha, beta):
    q = len(ass)
    n = ceil((b - a) / h)
    ys = [initial]
    xs = [a + i * h for i in range(n)]
    for i in range(n-1):
        integral = 0.
        y = ys[-1]
        phi = [f(xs[i], y)]
        for k in range(q):
            if k != 0:
                phi.append(f(xs[i] + h * alpha[k-1], y + h * sum([beta[k-1][j] * phi[k-1] for j in r
            integral += ass[k] * phi[k]
        ys.append(y + h * integral)
    norm = max([abs(y - u(x)) for (x, y) in zip(xs, ys)])
    plt.plot(xs, ys, "ro", xs, [u(x) for x in xs], "b")
    plt.legend(["$y(x)$", "$e^x$"])
    plt.title(f"$q = {q}$, $h = {h}$, norm = ${norm}$")
    plt.show()
    print(f"q = {q}, h = {h}, norm = {norm}")

def euler(h):
    n = ceil((b - a) / h)
    ys = [initial]
    xs = [a + i * h for i in range(n)]
    for i in range(n-1):
        last = ys[-1]
        while True:
            current = last - (last - ys[-1] - 0.5 * h * (f(xs[i], ys[-1]) + f(xs[i+1], last)))/(1 -
            if abs(last - current) < eps:
                ys.append(current)
                break
```

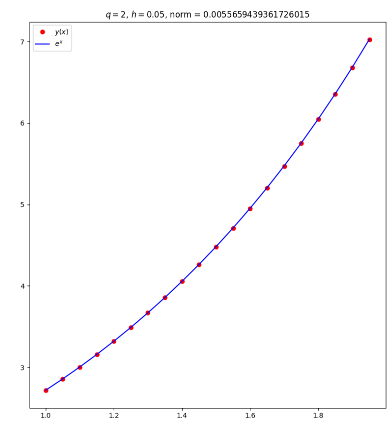
```
        last = current
norm = max([abs(y - u(x)) for (x, y) in zip(xs, ys)])
plt.plot(xs, ys, "ro", xs, [u(x) for x in xs], "b")
plt.legend(["$y(x)$", "$e^x$"])
plt.title(f"Euler $h = {h}$, norm = ${norm}$")
plt.show()
print(f"Euler h = {h}, norm = {norm}")
```

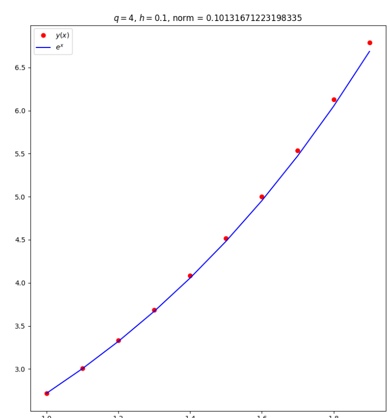
Графики

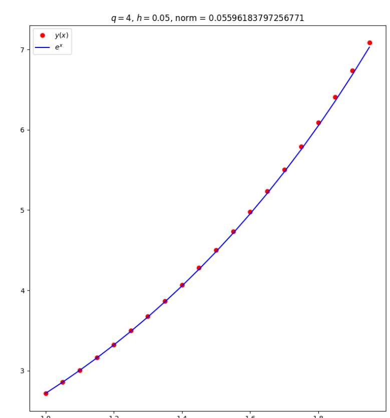

Euler $h = 0.1$, norm = 0.006823255891671209
$y(x)$
e^x


Euler $h = 0.05$, norm = 0.0019089474816054164
$y(x)$
e^x


$q = 2$, $h = 0.1$, norm = 0.018753394607056606
$y(x)$
e^x


$q = 2$, $h = 0.05$, norm = 0.0055659439361726015
$y(x)$
e^x


$q = 4$, $h = 0.1$, norm = 0.10131671223198335
$y(x)$
e^x


$q = 4$, $h = 0.05$, norm = 0.05596183797256771
$y(x)$
e^x

Результаты вычислительного эксперимента

Метод	h	Погрешность
НМТ	0.1	0.006823255891671209
НМТ	0.05	0.0019089474816054164
ЯСП	0.1	0.018753394607056606
ЯСП	0.05	0.0055659439361726015
$q = 4$	0.1	$3.70279931267703 \cdot 10^{-5}$
$q = 4$	0.05	$2.7515251455056955 \cdot 10^{-6}$

Выводы

Вычислительный эксперимент показал, что более высокий порядок аппроксимации метода не гарантирует меньшую погрешность полученного решения и что при достаточно малом количестве вычислений оба метода дают относительно небольшую погрешность.