Министерство образования и науки РФ

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Лабораторная работа № 2

“**Неявные методы Эйлера**”

по дисциплине «Численное моделирование динамических систем, описываемых обыкновенными дифференциальными уравнениями»

**Факультет:** ПМИ

**Группа:**  ПМ-92

**Студенты:**  Иванов В., Кутузов И.

**Преподаватель:**  Вагин Д. В.

Новосибирск

2021

**Условие**

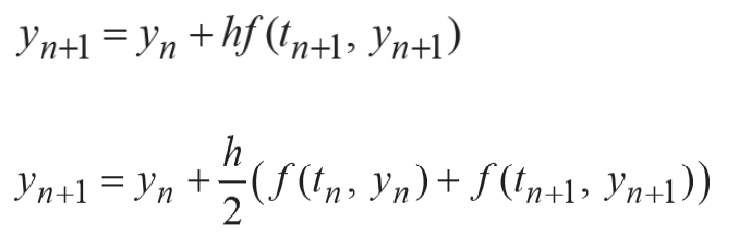
На трех сетках h = [0.1, 0.05, 0.025] решить задачу

y'=2ty

t=[0,1]

y(0)=1

с помощью двух неявных схем Эйлера.



**Шаг 0.1**

|  | | **Метод 1** | | **Метод 2** | |
| --- | --- | --- | --- | --- | --- |
| **t** | **y\_аналит** | **y\_числен** | **погрешн** | **y\_числен** | **погрешн** |
| 0.0 | 1.0 | 1.0 | 0.00e+00 | 1.0 | 0.00e+00 |
| 0.1 | 1.01005 | 1.020408 | 1.04e-02 | 1.010204 | 1.54e-04 |
| 0.2 | 1.040811 | 1.062925 | 2.21e-02 | 1.041352 | 5.41e-04 |
| 0.3 | 1.094174 | 1.130771 | 3.66e-02 | 1.095414 | 1.24e-03 |
| 0.4 | 1.173511 | 1.229099 | 5.56e-02 | 1.175903 | 2.39e-03 |
| 0.5 | 1.284025 | 1.365665 | 8.16e-02 | 1.288266 | 4.24e-03 |
| 0.6 | 1.433329 | 1.551891 | 1.19e-01 | 1.440515 | 7.19e-03 |
| 0.7 | 1.632316 | 1.804524 | 1.72e-01 | 1.644197 | 1.19e-02 |
| 0.8 | 1.896481 | 2.148241 | 2.52e-01 | 1.915881 | 1.94e-02 |
| 0.9 | 2.247908 | 2.619805 | 3.72e-01 | 2.279429 | 3.15e-02 |
| 1.0 | 2.718282 | 3.274754 | 5.56e-01 | 2.769505 | 5.12e-02 |

**Шаг 0.05**

|  | | **Метод 1** | | **Метод 2** | |
| --- | --- | --- | --- | --- | --- |
| **t** | **y\_аналит** | **y\_числен** | **погрешн** | **y\_числен** | **погрешн** |
| 0.0 | 1.0 | 1.0 | 0.00e+00 | 1.0 | 0.00e+00 |
| 0.05 | 1.002503 | 1.005025 | 2.52e-03 | 1.002513 | 9.43e-06 |
| 0.1 | 1.01005 | 1.015177 | 5.13e-03 | 1.010082 | 3.19e-05 |
| 0.15 | 1.022755 | 1.030636 | 7.88e-03 | 1.022823 | 6.84e-05 |
| 0.2 | 1.040811 | 1.05167 | 1.09e-02 | 1.040932 | 1.21e-04 |
| … | | | | | |
| 0.8 | 1.896481 | 2.012766 | 1.16e-01 | 1.901041 | 4.56e-03 |
| 0.85 | 2.059576 | 2.199744 | 1.40e-01 | 2.065382 | 5.81e-03 |
| 0.9 | 2.247908 | 2.417301 | 1.69e-01 | 2.255295 | 7.39e-03 |
| 0.95 | 2.46576 | 2.671051 | 2.05e-01 | 2.475155 | 9.39e-03 |
| 1.0 | 2.718282 | 2.967834 | 2.50e-01 | 2.730233 | 1.20e-02 |

**Шаг 0.025**

|  | | **Метод 1** | | **Метод 2** | |
| --- | --- | --- | --- | --- | --- |
| **t** | **y\_аналит** | **y\_числен** | **погрешн** | **y\_числен** | **погрешн** |
| 0.0 | 1.0 | 1.0 | 0.00e+00 | 1.0 | 0.00e+00 |
| 0.025 | 1.000625 | 1.001252 | 6.26e-04 | 1.000626 | 5.87e-07 |
| 0.05 | 1.002503 | 1.003761 | 1.26e-03 | 1.002505 | 1.96e-06 |
| 0.075 | 1.005641 | 1.007539 | 1.90e-03 | 1.005645 | 4.14e-06 |
| 0.1 | 1.01005 | 1.012602 | 2.55e-03 | 1.010057 | 7.16e-06 |
| … | | | | | |
| 0.9 | 2.247908 | 2.329043 | 8.11e-02 | 2.249704 | 1.80e-03 |
| 0.925 | 2.352844 | 2.441985 | 8.91e-02 | 2.35487 | 2.03e-03 |
| 0.95 | 2.46576 | 2.563763 | 9.80e-02 | 2.468043 | 2.28e-03 |
| 0.975 | 2.587326 | 2.695152 | 1.08e-01 | 2.589901 | 2.57e-03 |
| 1.0 | 2.718282 | 2.837002 | 1.19e-01 | 2.721185 | 2.90e-03 |

Первый метод, в отличии от явного, показал аппроксимацию сверху, что соответствует ожиданиям. Метод трапеций, являясь, по сути, усреднением явного простого и неявного простого методов, показал намного меньшую погрешность, чем какие-либо другие методы из ранее изученных.

**Код программы**

**import numpy as np**

**import pandas as pd**

**h1, h2, h3 = [0.1, 0.05, 0.025]**

**eps = 0.0001**

**def f(t, y):**

**return 2\*t\*y**

**def f\_solve(t):**

**return np.exp(t\*t)**

**def method1(stepsize):**

**y = 1.0**

**num\_result, real\_result = [], []**

**num\_result.append(y + stepsize\*f(0.0, y))**

**for i in np.arange(0, 1, stepsize):**

**y\_k = y + stepsize\*f(i, y)**

**while True:**

**y\_k1 = y + stepsize\*f(i+stepsize, y\_k)**

**if abs(y\_k1-y\_k) < eps:**

**break**

**y\_k = y\_k1**

**y\_n1 = y + stepsize\*f(i+stepsize, y\_k1)**

**num\_result.append(y\_n1)**

**y = y\_n1**

**grid = np.arange(0, 1, stepsize)**

**for i in grid:**

**y\_new\_real = f\_solve(i)**

**real\_result.append(y\_new\_real)**

**real\_result.append(f\_solve(1))**

**error = np.absolute(np.array(num\_result) - np.array(real\_result))**

**error\_scientific = []**

**for x in error:**

**error\_scientific.append('{:.2e}'.format(x))**

**df = pd.DataFrame([grid, real\_result, num\_result, error\_scientific]).T**

**df.columns = ['t', 'y\_real', 'y\_num', '|y\_num-y\_real|']**

**df['t'].iloc[-1] = 1.0**

**df**

**return df**

**def method2(stepsize):**

**y = 1.0**

**num\_result, real\_result = [], []**

**num\_result.append(y + stepsize\*f(0.0, y))**

**for i in np.arange(0, 1, stepsize):**

**y\_k = y + stepsize\*f(i, y)**

**while True:**

**y\_k1 = y + stepsize\*f(i+stepsize, y\_k)**

**if abs(y\_k1-y\_k) < eps:**

**break**

**y\_k = y\_k1**

**y\_n1 = y + (stepsize/2.0)\*(f(i, y)+f(i+stepsize, y\_k1))**

**num\_result.append(y\_n1)**

**y = y\_n1**

**grid = np.arange(0, 1, stepsize)**

**for i in grid:**

**y\_new\_real = f\_solve(i)**

**real\_result.append(y\_new\_real)**

**real\_result.append(f\_solve(1))**

**error = np.absolute(np.array(num\_result) - np.array(real\_result))**

**error\_scientific = []**

**for x in error:**

**error\_scientific.append('{:.2e}'.format(x))**

**df = pd.DataFrame([grid, real\_result, num\_result, error\_scientific]).T**

**df.columns = ['t', 'y\_real', 'y\_num', '|y\_num-y\_real|']**

**df['t'].iloc[-1] = 1.0**

**df**

**return df**