Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
Высшего образования
Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа 6 по вычислительной математике

Численное решение обыкновенных дифференциальных уравнений

Вариант №13

Группа: P3216
Выполнил:
Сиразетдинов А.Н.
Проверил:
Малышева Т.А.

Г. Санкт-Петербург
2024

Оглавление

Цель работы

Решить задачу Коши для обыкновенных дифференциальных уравнений численными методами

Описание алгоритма

Метод Эйлера

Метод Э й л е р а (1707–1783) основан на разложении искомой функции $y(x)$в ряд Тейлора в окрестностях узлов $x = x_i$ $(i = 0, 1, \dots)$, в котором отбрасываются все члены, содержащие производные второго и более высоких порядков:

$$Y(x_i + h) = Y(x_i) + Y'(x_i) \cdot h + O(h^2)$$

Полагаем: $Y'(x_i) = f(x_i, Y(x_i)) = f(x_i, y_i) = \frac{y_{i+1} - y_i}{h}$

Введем последовательность равноотстоящих точек $x_0, x_1, \dots, x_n$ (узлов), выбрав малый шаг $h = x_{i+1} - x_i = const$. Тогда получаем **формулу Эйлера**:

$$\boldsymbol{y_{i+1} = y_i + hf(x_i, y_i)} \qquad (7)$$

При i=0 находим значение сеточной функции y_1 при $x = x_1$:

$$y_1 = y_0 + hf(x_0, y_0)$$

Значение y_0 задано начальным условием:

$$\boldsymbol{y_0 = Y_0} \qquad (8)$$

Аналогично могут быть найдены значения сеточной функции в других узлах:

$$y_2 = y_1 + hf(x_1, y_1)$$

$$\tilde{y}_{i+1} = y_i + hf(x_i, y_i)$$

Вычисленное значение $\tilde{y}_{i+1}$ подставляем вместо y_{i+1} в правую часть соотношения (9) и находим окончательное значение:

$$y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+1}, \tilde{y}_{i+1})] \text{ или:}$$

$$\boldsymbol{y_{i+1} = y_i + \frac{h}{2}\left[f(x_i, y_i) + f\big(x_{i+1}, y_i + hf(x_i, y_i)\big)\right],\ i = 0, 1 \dots} \qquad (10)$$

Данные рекуррентные соотношения описывают новую разностную схему, являющуюся ***модифицированным методом Эйлера***, которая называется методом ***Эйлера с пересчетом***. Метод Эйлера с пересчетом имеет ***второй порядок точности*** $\delta_n = O(h^2)$.

Метод Милна

Метод Милна относится к многошаговым методам и представляет один из методов прогноза и коррекции.

Для получения формул Милна используется первая интерполяционная формула Ньютона с разностями до третьего порядка.

Решение в следующей точке находится в два этапа. На первом этапе осуществляется прогноз значения функции, а затем на втором этапе - коррекция полученного значения. Если полученное значение *y* после коррекции существенно отличается от спрогнозированного, то проводят еще один этап коррекции. Если опять имеет место существенное отличие от предыдущего значения (т.е. от предыдущей коррекции), то проводят еще одну коррекцию и т.д.

Вычислительные формулы:

а) этап прогноза:

$$y_i^{\text{прогн}} = y_{i-4} + \frac{4h}{3}(2f_{i-3} - f_{i-2} + 2f_{i-1})$$

б) этап коррекции:

$$y_i^{\text{корр}} = y_{i-2} + \frac{h}{3}(f_{i-2} + 4f_{i-1} + f_i^{\text{прогн}})$$

$$f_i^{\text{прогн}} = f(x_i, y_i^{\text{прогн}})$$

Для начала счета требуется задать решения в трех первых точках, которые можно получить одношаговыми методами (например, методом Рунге-Кутта).

Листинг программы

```python
def perform_euler(self) -> [float, PrettyTable, list[float], list[float]]:
    table = PrettyTable()
    table.title = self.name
    table.field_names = ["i", "xi", "yi", "f(xi, yi)", "Точное решение"]
    table.float_format = ".3"
    x = self.x0
    y = self.y0
    y_list = [y]
    x_list = [x]
    i = 0
    while x < self.xn:
        table.add_row([i, x, y, self.f(x, y), self.f_ac(x)])
        y += self.h * self.f(x, y)
        x += self.h
        x_list.append(x)
        y_list.append(y)
        i += 1
    # print(*[f"{i : .3f}" for i in x_list])
    # print(*[f"{i : .3f}" for i in y_list])
    return y, table, x_list, y_list
```

```python
def perform_euler(self) -> [float, PrettyTable, list[float], list[float]]:
    table = PrettyTable()
    table.title = self.name
    table.field_names = ["i", "xi", "yi", "f(xi, yi)", "Точное решение"]
    table.float_format = ".3"
    x = self.x0
    y = self.y0
    y_list = [y]
    x_list = [x]
    i = 0
    while x < self.xn:
        table.add_row([i, x, y, self.f(x, y), self.f_ac(x)])
        y += self.h / 2 * (self.f(x, y) + self.f(x + self.h, y + self.h * self.f(x, y)))
        x += self.h
        x_list.append(x)
        y_list.append(y)
        i += 1
    # print(*[f"{i : .3f}" for i in x_list])
    # print(*[f"{i : .3f}" for i in y_list])
    return y, table, x_list, y_list
```

```python
def perform_miln(self) -> [float, PrettyTable, list[float], list[float]]:
    table = PrettyTable()
    table.title = self.name
    table.field_names = ["i", "xi", "yi", "f(xi, yi)", "Точное решение", "e"]
    table.float_format = ".5"
    eps_max = 0.0
    x = self.x0
    f = self.f
    h = self.h
    y = self.y0
    x_list = [x]
    y_list = [y]
    f_list = [f(x, y)]
    for i in range(3):
        y += h * self.f(x, y)
        x += h
        x_list.append(x)
        y_list.append(y)
        f_list.append(f(x, y))
    # print(x_list)
    # print(y_list)
    # print(f_list)
    for i, x in enumerate(x_list):
        table.add_row([i, x, y_list[i], self.f(x, y), self.f_ac(x), abs(y - self.f_ac(x))])
    i = 0
    while x < self.xn:
        x += self.h
        y = y_list[-4] + 4 * h / 3 * (2 * f_list[-3] - f_list[-2] + 2 * f_list[-1])
        new_y_corrected = y_list[-2] + h / 3 * (f_list[-2] + 4 * f_list[-1] + self.f(x, y))
        table.add_row([i + 3, x, new_y_corrected, self.f(x, y), self.f_ac(x), abs(new_y_corrected - self.f_ac(x))])
        while abs(y - new_y_corrected) > self.e:
            y = new_y_corrected
            new_y_corrected = y_list[-2] + h / 3 * (f_list[-2] + 4 * f_list[-1] + self.f(x, y))
            table.add_row(["", "", new_y_corrected, self.f(x, y), self.f_ac(x), abs(new_y_corrected - self.f_ac(x))])
        print(abs(y - new_y_corrected))
        y = new_y_corrected
        x_list.append(x)
        y_list.append(y)
        f_list.append(self.f(x, y))
        eps_max = max(eps_max, abs(y - self.f_ac(x)))
        i += 1
    # print(*[f"{i : .3f}" for i in x_list])
    # print(*[f"{i : .3f}" for i in y_list])
    return eps_max, table, x_list, y_list
```

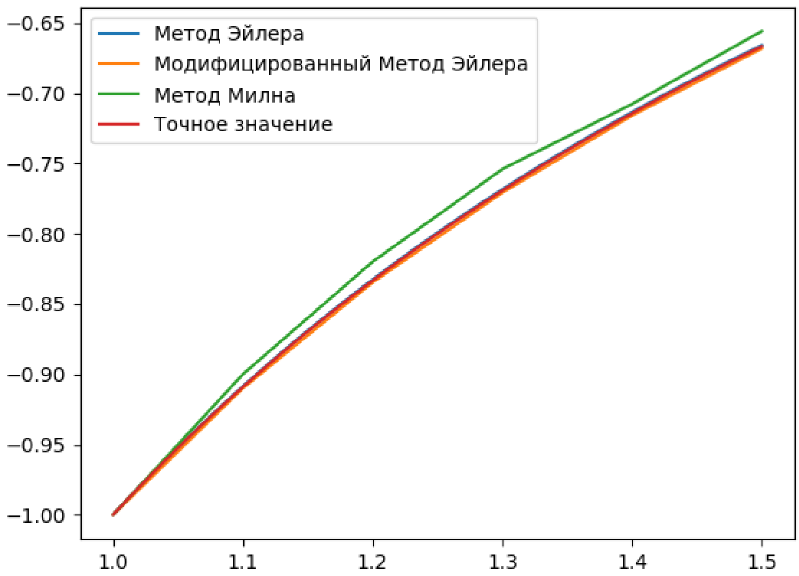
Результат работы программы

```
Выберите уравнение:
0) y` = y + (1 + x) * y ** 2
1) y` = 2 * x - y + x ** 2
2) y` = (x - y) ** 2 + 1
0
Введите x0 1
Введите y0 -1
Введите xn 1.5
Введите шаг h 0.1
Введите точность e 0.001

+-----------------------------------------------+
|                  Метод Эйлера                 |
+----+-------+--------+-----------+----------------+
| i  |  xi   |   yi   | f(xi, yi) | Точное решение |
+----+-------+--------+-----------+----------------+
| 0  | 1.000 | -1.000 |   1.000   |     -1.000     |
| 1  | 1.006 | -0.994 |   0.988   |     -0.994     |
| 2  | 1.013 | -0.988 |   0.975   |     -0.988     |
| 3  | 1.019 | -0.981 |   0.963   |     -0.982     |
| 4  | 1.025 | -0.975 |   0.951   |     -0.976     |
| 5  | 1.031 | -0.970 |   0.940   |     -0.970     |
| 6  | 1.038 | -0.964 |   0.928   |     -0.964     |
| 7  | 1.044 | -0.958 |   0.917   |     -0.958     |
| 8  | 1.050 | -0.952 |   0.906   |     -0.952     |
| 9  | 1.056 | -0.946 |   0.895   |     -0.947     |
```

```
| 74 | 1.463 | -0.683 |   0.465   |     -0.684     |
| 75 | 1.469 | -0.680 |   0.461   |     -0.681     |
| 76 | 1.475 | -0.677 |   0.458   |     -0.678     |
| 77 | 1.481 | -0.674 |   0.454   |     -0.675     |
| 78 | 1.488 | -0.671 |   0.450   |     -0.672     |
| 79 | 1.494 | -0.669 |   0.446   |     -0.669     |
+----+-------+--------+-----------+----------------+
Погрешность по правилу Рунге:  0.00044
+-----------------------------------------------+
|          Модифицированный Метод Эйлера         |
+---+-------+--------+-----------+----------------+
| i |  xi   |   yi   | f(xi, yi) | Точное решение |
+---+-------+--------+-----------+----------------+
| 0 | 1.000 | -1.000 |   1.000   |     -1.000     |
| 1 | 1.100 | -0.910 |   0.829   |     -0.909     |
| 2 | 1.200 | -0.835 |   0.698   |     -0.833     |
| 3 | 1.300 | -0.771 |   0.595   |     -0.769     |
| 4 | 1.400 | -0.716 |   0.514   |     -0.714     |
+---+-------+--------+-----------+----------------+
Погрешность по правилу Рунге:  0.00038
0.00013323171203505346
0.0008769475833476648
```

```
+---+-------+--------+----------+---------------+
Погрешность по правилу Рунге:  0.00038
0.0001332317120350534б
0.0008769475833476648
+---------------------------------------------------------------+
|                         Метод Милна                           |
+---+---------+----------+-----------+---------------+---------+
| i |    xi   |    yi    | f(xi, yi) | Точное решение |    e    |
+---+---------+----------+-----------+---------------+---------+
| 0 | 1.00000 | -1.00000 |  0.38303  |    -1.00000    | 0.24600 |
| 1 | 1.10000 | -0.90000 |  0.43988  |    -0.90909    | 0.15509 |
| 2 | 1.20000 | -0.81990 |  0.49673  |    -0.83333    | 0.07934 |
| 3 | 1.30000 | -0.75400 |  0.55358  |    -0.76923    | 0.01523 |
| 3 | 1.40000 | -0.70610 |  0.54059  |    -0.71429    | 0.00818 |
|   |         | -0.70777 |  0.49049  |    -0.71429    | 0.00651 |
|   |         | -0.70764 |  0.49449  |    -0.71429    | 0.00665 |
| 4 | 1.50000 | -0.65487 |  0.44358  |    -0.66667    | 0.01180 |
|   |         | -0.65575 |  0.41727  |    -0.66667    | 0.01092 |
+---+---------+----------+-----------+---------------+---------+
Оценка точности:  0.01092
```

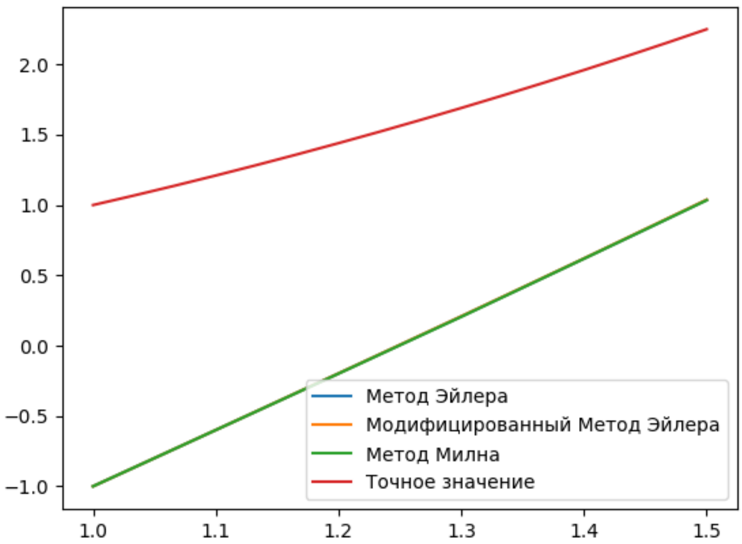


```
C:\Users\azat2\AppData\Local\pypoetry\Cache\virtualenvs\Лабораторная_работа_6-0cu5f8_
Выберите уравнение:
0) y` = y + (1 + x) * y ** 2
1) y` = 2 * x - y + x ** 2
2) y` = (x - y) ** 2 + 1
1
Введите x0 1
Введите y0 -1
Введите xn 1.5
Введите шаг h 0.1
Введите точность e 0.001

+-------------------------------------------------+
|                   Метод Эйлера                  |
+----+-------+--------+-----------+---------------+
| i  |   xi  |   yi   | f(xi, yi) | Точное решение |
+----+-------+--------+-----------+---------------+
| 0  | 1.000 | -1.000 |   4.000   |     1.000      |
| 1  | 1.006 | -0.975 |   4.000   |     1.013      |
| 2  | 1.013 | -0.950 |   4.000   |     1.025      |
| 3  | 1.019 | -0.925 |   4.000   |     1.038      |
| 4  | 1.025 | -0.900 |   4.001   |     1.051      |
| 5  | 1.031 | -0.875 |   4.001   |     1.063      |
| 6  | 1.038 | -0.850 |   4.001   |     1.076      |
| 7  | 1.044 | -0.825 |   4.002   |     1.089      |
| 8  | 1.050 | -0.800 |   4.002   |     1.103      |
| 9  | 1.056 | -0.775 |   4.003   |     1.116      |
| 10 | 1.063 | -0.750 |   4.004   |     1.129      |
| 11 | 1.069 | -0.725 |   4.005   |     1.142      |
```

```
| 78 | 1.488 | 0.984  |   4.204   |     2.213      |
| 79 | 1.494 | 1.010  |   4.209   |     2.231      |
+----+-------+--------+-----------+---------------+
Погрешность по правилу Рунге:  0.00028
+-------------------------------------------------+
|          Модифицированный Метод Эйлера          |
+---+-------+--------+-----------+---------------+
| i |   xi  |   yi   | f(xi, yi) | Точное решение |
+---+-------+--------+-----------+---------------+
| 0 | 1.000 | -1.000 |   4.000   |     1.000      |
| 1 | 1.100 | -0.599 |   4.010   |     1.210      |
| 2 | 1.200 | -0.197 |   4.037   |     1.440      |
| 3 | 1.300 | 0.209  |   4.081   |     1.690      |
| 4 | 1.400 | 0.620  |   4.140   |     1.960      |
+---+-------+--------+-----------+---------------+
Погрешность по правилу Рунге:  0.00025
6.173333333325814e-05
0.0001031583012345294
```

```
+---+-------+--------+----------+---------------+
Погрешность по правилу Рунге:  0.00025
6.173333333325814e-05
0.0001031583012345294
+-----------------------------------------------------------------+
|                          Метод Милна                            |
+---+---------+----------+-----------+---------------+---------+
| i |    xi   |    yi    | f(xi, yi) | Точное решение |    e    |
+---+---------+----------+-----------+---------------+---------+
| 0 | 1.00000 | -1.00000 |  2.79510  |    1.00000    | 0.79510 |
| 1 | 1.10000 | -0.60000 |  3.20510  |    1.21000    | 1.00510 |
| 2 | 1.20000 | -0.19900 |  3.63510  |    1.44000    | 1.23510 |
| 3 | 1.30000 | 0.20490  |  4.08510  |    1.69000    | 1.48510 |
| 3 | 1.40000 | 0.61831  |  4.13984  |    1.96000    | 1.34169 |
|   |         | 0.61837  |  4.14169  |    1.96000    | 1.34163 |
| 4 | 1.50000 | 1.03373  |  4.21318  |    2.25000    | 1.21627 |
|   |         | 1.03383  |  4.21627  |    2.25000    | 1.21617 |
+---+---------+----------+-----------+---------------+---------+
Оценка точности:  1.34163
```

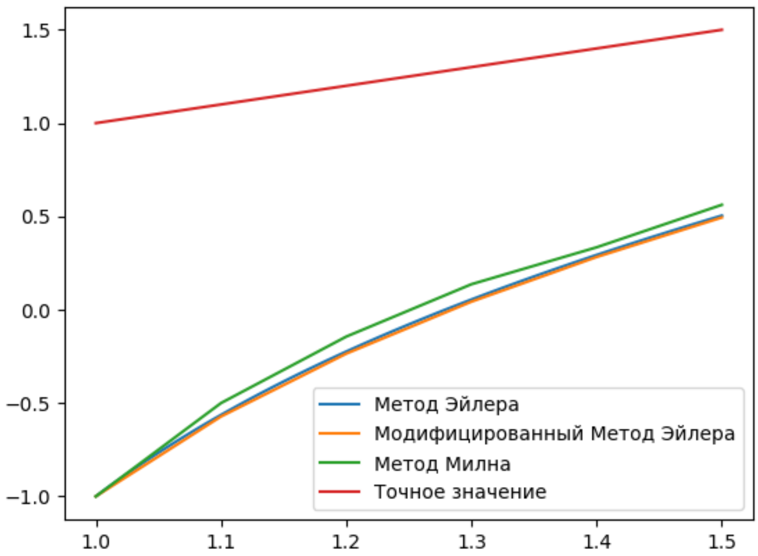


```
C:\Users\azat2\AppData\Local\pypoetry\Cache\virtualenvs\лабор
Выберите уравнение:
0) y` = y + (1 + x) * y ** 2
1) y` = 2 * x - y + x ** 2
2) y` = (x - y) ** 2 + 1
2
Введите x0 1
Введите y0 -1
Введите xn 1.5
Введите шаг h 0.1
Введите точность e 0.01

+-----------------------------------------------------+
|                     Метод Эйлера                    |
+----+-------+--------+-----------+----------------+
| i  |   xi  |   yi   | f(xi, yi) | Точное решение |
+----+-------+--------+-----------+----------------+
| 0  | 1.000 | -1.000 |   5.000   |     1.000      |
| 1  | 1.006 | -0.969 |   4.901   |     1.006      |
| 2  | 1.013 | -0.938 |   4.805   |     1.013      |
| 3  | 1.019 | -0.908 |   4.713   |     1.019      |
| 4  | 1.025 | -0.879 |   4.624   |     1.025      |
| 5  | 1.031 | -0.850 |   4.538   |     1.031      |
| 6  | 1.038 | -0.821 |   4.455   |     1.038      |
| 7  | 1.044 | -0.794 |   4.376   |     1.044      |
| 8  | 1.050 | -0.766 |   4.299   |     1.050      |
| 9  | 1.056 | -0.739 |   4.224   |     1.056      |
| 10 | 1.063 | -0.713 |   4.152   |     1.063      |
```

```
| 78 | 1.488 | 0.479  |   2.017   |     1.488      |
| 79 | 1.494 | 0.492  |   2.004   |     1.494      |
+----+-------+--------+-----------+----------------+
Погрешность по правилу Рунге:  0.00219
+---------------------------------------------------+
|            Модифицированный Метод Эйлера          |
+---+-------+--------+-----------+----------------+
| i |   xi  |   yi   | f(xi, yi) | Точное решение |
+---+-------+--------+-----------+----------------+
| 0 | 1.000 | -1.000 |   5.000   |     1.000      |
| 1 | 1.100 | -0.572 |   3.796   |     1.100      |
| 2 | 1.200 | -0.235 |   3.060   |     1.200      |
| 3 | 1.300 | 0.043  |   2.579   |     1.300      |
| 4 | 1.400 | 0.283  |   2.249   |     1.400      |
+---+-------+--------+-----------+----------------+
Погрешность по правилу Рунге:  0.00145
0.0007526467057408781
0.004003225603507077
```

```
+---+-------+--------+----------+---------------+
Погрешность по правилу Рунге:  0.00145
0.0007526467057408781
0.004003225603507077
+---------------------------------------------------------------+
|                          Метод Милна                          |
+---+---------+----------+-----------+---------------+---------+
| i |    xi   |    yi    | f(xi, yi) | Точное решение |    e    |
+---+---------+----------+-----------+---------------+---------+
| 0 | 1.00000 | -1.00000 |  1.74540  |    1.00000     | 0.86337 |
| 1 | 1.10000 | -0.50000 |  1.92807  |    1.10000     | 0.96337 |
| 2 | 1.20000 | -0.14400 |  2.13075  |    1.20000     | 1.06337 |
| 3 | 1.30000 | 0.13663  |  2.35342  |    1.30000     | 1.16337 |
| 3 | 1.40000 | 0.34445  |  2.43345  |    1.40000     | 1.05555 |
|   |         | 0.33381  |  2.11419  |    1.40000     | 1.06619 |
|   |         | 0.33456  |  2.13677  |    1.40000     | 1.06544 |
| 4 | 1.50000 | 0.56617  |  1.99213  |    1.50000     | 0.93383 |
|   |         | 0.56217  |  1.87203  |    1.50000     | 0.93783 |
+---+---------+----------+-----------+---------------+---------+
Оценка точности:  1.06544
```



Вывод

В ходе выполнения лабораторной работы я рассмотрел и реализовал численные методы решения обыкновенных дифференциальных уравнений методами Эйлера, усовершенствованным методом Эйлера и методом Милна