Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

Вычислительная математика

Лабораторная работа №5.

Усовершенствованный метод Эйлера

Выполнил:

Маликов Глеб Игоревич

Группа № P3224

Преподаватели:

Перл Ольга Вячеславовна

Хохлов Александр Алексеевич

г. Санкт-Петербург

2024

Оглавление

Задание

Усовершенствованный метод Эйлера

Реализуйте усовершенствованный метод Эйлера для решения обыкновенных дифференциальных уравнений по начальному значению (задача Коши) в интервале от a до b [a,b].

```
f
epsilon
a
y(a)
b
```

f - номер уравнения, где уравнение в виде y'=f(x,y). Вы должны получить функцию по номеру из входных данных в методе get_function.

Вы должны определить и пересчитать шаг h самостоятельно.

Вы должны вычислить и вернуть y(b) с разницей, не превышающей epsilon.

Описание численного метода

Усовершенствованный метод Эйлера, это численный метод решения обыкновенных дифференциальных уравнений (ОДУ). Метод Эйлера основан на аппроксимации интегральной кривой кусочно-линейной функцией, известной как ломаная Эйлера.

Суть метода заключается в пошаговом вычислении значений решения дифференциального уравнения вида $y' = f(x, y)$ с начальным условием $(x_0; y_0)$. Функция $f(x, y)$ определена на некоторой области. Решение ищется на полуинтервале, в котором вводятся узлы. Они определяются по формуле

$$y_{i+1} = y_i + h \cdot f(x_i, y_i)$$

При этом, в усовершенствованном методе Эйлера выполняется коррекция

$$y_{i+1} = y_i + h \cdot f(x_{i+1}, \tilde{y}_{i+1})$$

, где $\tilde{y}_{i+1}$ определяется предыдущей формулой.

Блок-схемы

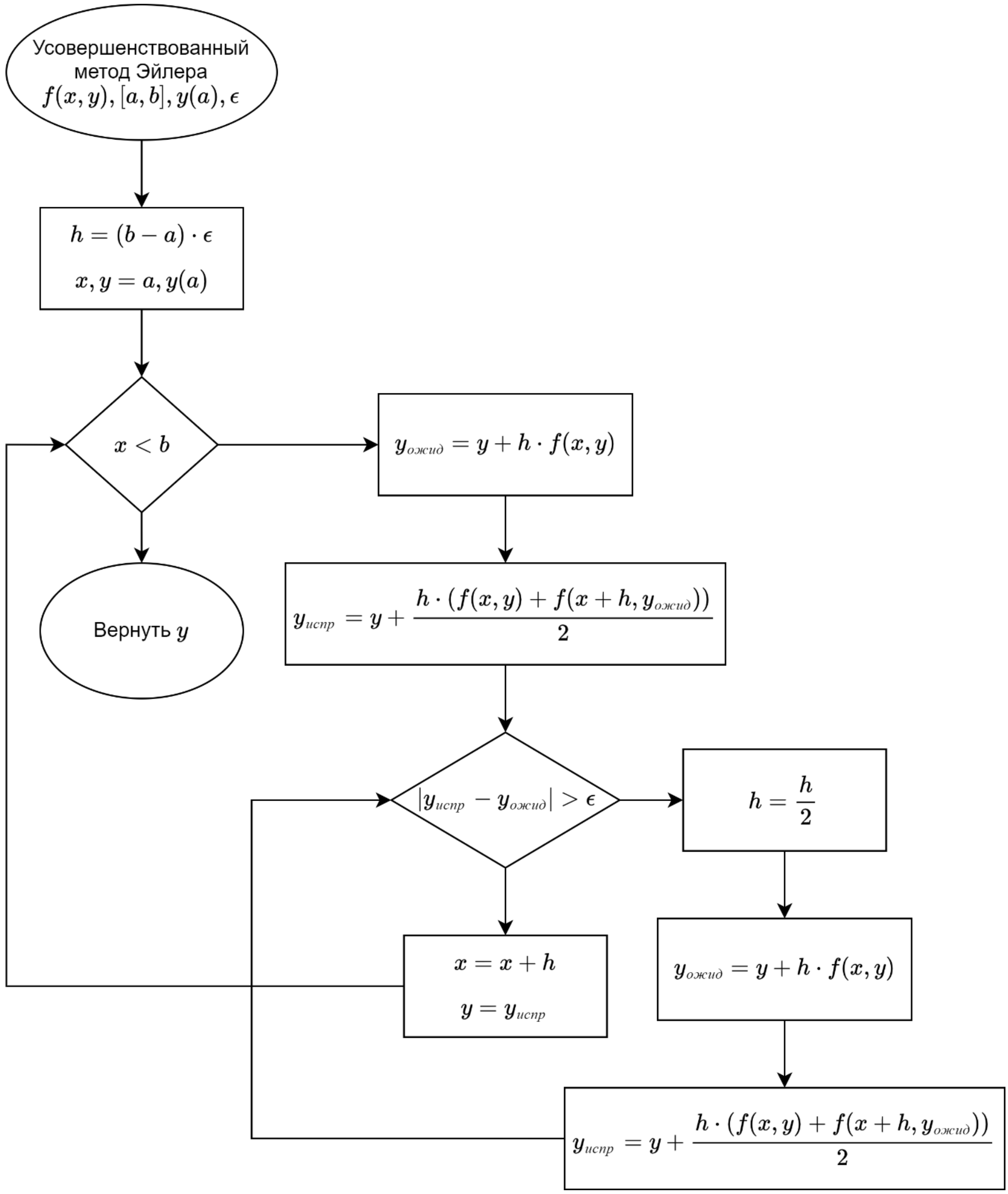

Схема 1 - Усовершенствованный метод Эйлера

Код

```python
def solveByEulerImproved(f, epsilon, a, y_a, b):
    """
    Решает обыкновенное дифференциальное уравнение методом усовершенствованного Эйлера.

    :param f: Номер выбранной функции
    :param epsilon: Точность
    :param a: Нижняя граница интервала.
    :param y_a: Значение функции в точке a.
    :param b: Верхняя граница интервала.
    :return: Вычисленное значение функции в точке b.
    """
    func = Result.get_function(f)

    # Начальные условия
    x, y = a, y_a
    h = (b-a) * epsilon  # начальный шаг

    while x < b:
        # Вычисляем прогноз
        y_predict = y + h * func(x, y)

        # Вычисляем коррекцию
        y_correct = y + h * (func(x, y) + func(x + h, y_predict)) / 2

        # Проверяем, достаточно ли маленькая разница между прогнозом и коррекцией
        while abs(y_correct - y_predict) > epsilon:
            # Если разница слишком большая, уменьшаем шаг и повторяем процесс
            h /= 2
            y_predict = y + h * func(x, y)
            y_correct = y + h * (func(x, y) + func(x + h, y_predict)) / 2

        # Обновляем x и y
        x += h
        y = y_correct

    return y
```

Примеры работы программы

Пример 1

Ввод:

```
1
1e-6
0
0
3.141592653589793
```

Вывод:

```
2.000000000001502
```

Информация:

```
Step: 3.141592653589793e-06
```

Пример 2

Ввод:

```
2
1e-6
0
2
2
```

Вывод:

5.436563656912098

Информация:

```
Step: 2e-06
```

Пример 3

Ввод:

```
3
1e-6
0
1
4
```

Вывод:

```
3.000000003315696
```

Информация:

```
Step: 4e-06
```

Пример 4

Ввод:

```
4
1e-6
5
-5
6
```

Вывод:

```
-4.281718171440303
```

Информация:

```
Step: 1e-06
```

Пример 5

Ввод:

```
4
1e-6
0
10
5
```

Вывод:

```
1626.5529078382444
```

Информация:

```
Step: 4.9999999999999996e-06
```

Вывод

Усовершенствованный метод Эйлера успешно находит значения обыкновенных дифференциальных уравнений без аналитического решения, что помогает решить многие уравнения, в которых невозможно разделить переменные или вычислить интегралы для их решения. По сравнению с другими методами, такими как метод Эйлера, усовершенствованный метод Эйлера показывает лучшую точность, однако его использование требует вдвое больше вычислений. При этом, усовершенствованный метод Эйлера, не требуется сложность и вычислительная затратность методов более высокого порядка, таких как методы Рунге-Кутты

Алгоритмическая сложность усовершенствованного метода Эйлера зависит от требуемой точности и интервала, на котором решается дифференциальное уравнение. Однако, в общем случае, эта сложность может быть оценена как $O(1/\varepsilon)$, где ε — это требуемая точность. Это связано с тем, что количество шагов обратно пропорционально размеру шага, который, в свою очередь, прямо пропорционален ε.

Код успешно решает ОДУ, но стоит отметить, что очень высокая точность ε может значительно уменьшить скорость работы программы, а низкая точность может в некоторых случаях показывать некорректные результаты.

Численная ошибка данного метода зависит от локальной ошибки и от глобальной ошибки. Локальная ошибка — это разность между численным решением после одного шага и точным решением в точке, и она зависит от второй производной. Глобальная ошибка — это погрешность в последней точке отрезка интегрирования уравнения. Эта ошибка накапливается с каждым шагом вычисления и отражает общую точность численного метода на всем отрезке интегрирования.