ИТМО

Отчет
По лабораторной работе №5
Усовершенствованный метод Эйлера

Выполнил:
Рахматов Нематджон
P3233
Преподаватель:
Перл Ольга
Вячеславовна

Санкт-Петербург
2024

Оглавление

Задача

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 Метода

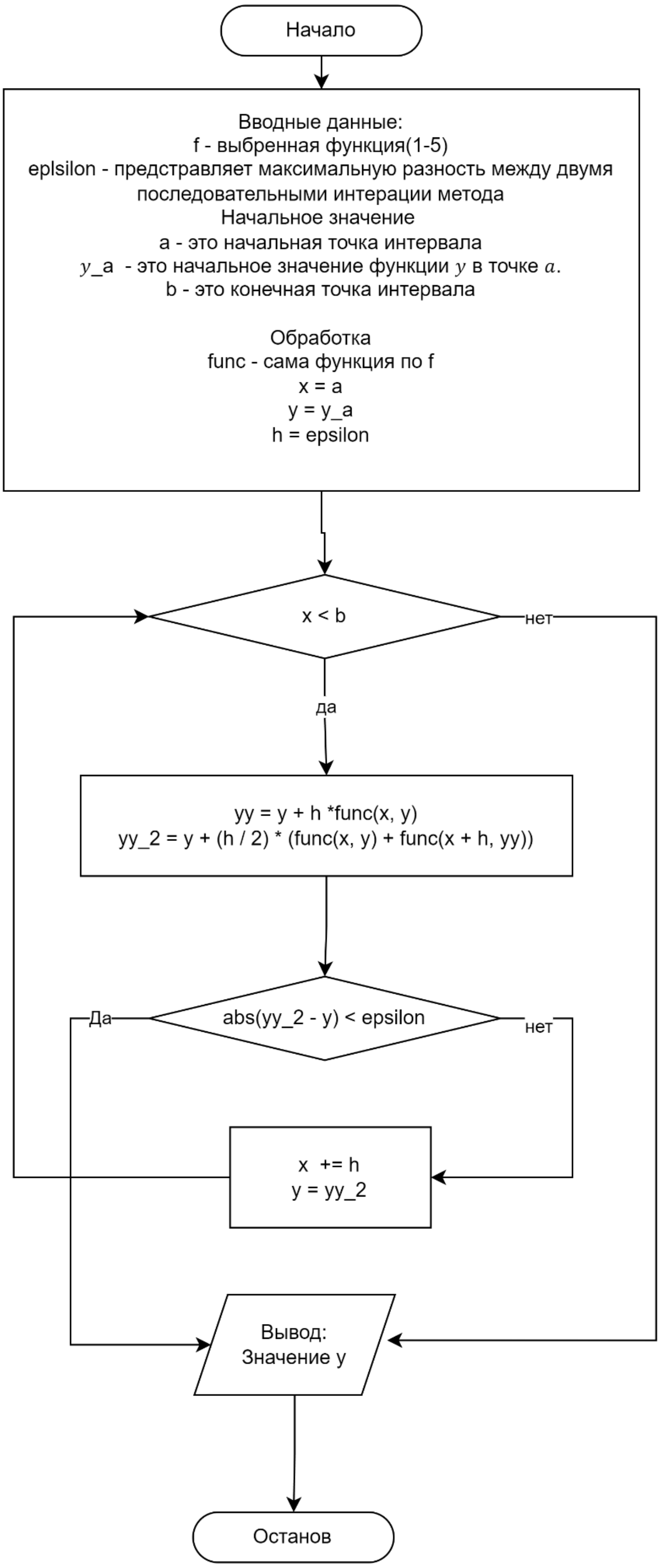
Усовершенствованный метод Эйлера (или метод Эйлера с пересчетом) представляет собой модификацию базового метода Эйлера, которая позволяет улучшить точность результата.

Основное усовершенствование заключается в том, что шаг $\hbar$ пересчитывается на каждой итерации, чтобы адаптировать его к изменяющимся условиям и обеспечить достижение требуемой точности. Это достигается путем оценки погрешности на каждом шаге и корректировки шага так, чтобы погрешность была в пределах заданной точности ϵ.

Алгоритм усовершенствованного метода Эйлера:

1) Задается начальное значение x0 и начальное значение функции y0.
2) Выбирается начальное значение шага $\hbar$.
3) Используется формула Эйлера для вычисления предварительного значения следующей точки: y_*Euler* = $yi + \hbar \cdot f(xi, yi)$.
4) Вычисляется значение функции на той же точке с использованием более точного метода, например, метода Рунге-Кутты, и получается предполагаемое значение y_RK.
5) Оценивается погрешность как разница между значениями y_*Euler и* y_RK.
6) Если погрешность меньше заданной точности ϵ, алгоритм завершается. В противном случае, шаг $\hbar$ пересчитывается, и алгоритм повторяется с новым значением шага.
Усовершенствованный метод Эйлера обеспечивает более точное приближенное решение дифференциального уравнения по сравнению с базовым методом Эйлера, особенно при больших значениях $\hbar$ или быстро изменяющихся функциях $f(x,y)$.

Блок-схема алгоритма



Код программы

```python
def calc(x, y, b, func, e):
    h = e

    while x < b:
        yy = y + h * func(x, y)
        yy_2 = y + (h / 2) * (func(x, y) + func(x + h, yy))

        if abs(yy_2 - y) < 10e-9:
            break

        x += h
        y = yy_2

    return y

class Result:
    def first_function(x: float, y: float):
        return math.sin(x)

    def second_function(x: float, y: float):
        return (x * y) / 2

    def third_function(x: float, y: float):
        return y - (2 * x) / y

    def fourth_function(x: float, y: float):
        return x + y

    def default_function(x: float, y: float):
        return 0.0

    @staticmethod
    def get_function(n: int):
        if n == 1:
            return Result.first_function
        elif n == 2:
            return Result.second_function
        elif n == 3:
            return Result.third_function
        elif n == 4:
            return Result.fourth_function
        else:
```

```
            return Result.default_function

    def solveByEulerImproved(f, epsilon, a, y_a, b):
        func = Result.get_function(f)

        x = a
        y = y_a

        h = epsilon

        return calc(x, y, h, b, func, epsilon)

if __name__ == '__main__':
    f = int(input().strip())
    epsilon = float(input().strip())
    a = float(input().strip())
    y_a = float(input().strip())
    b = float(input().strip())
    result = Result.solveByEulerImproved(f, epsilon, a, y_a, b)
    print(result)
```

Примеры Работы программы

1) Пример 1:

```
if __name__ == '__main__':
    f = 1
    e = 0.001
    a = 0
    y_a = 0
    b = 1
    result = Result.solveByEulerImproved(f, e, a, y_a, b)
    print(result)
```

result: `0.4596976558237189б`

2) Пример 2:

```
if __name__ == '__main__':
    f = 2
    e = 0.001
    a = 0
```

```
    y_a = 1
    b = 1
    result = Result.solveByEulerImproved(f, e, a, y_a, b)
    print(result)
```

result: `1.2840254099620019`

3) Пример 3:

```
if __name__ == '__main__':
    f = 3
    e = 0.001
    a = 1
    y_a = 2
    b = 2
    result = Result.solveByEulerImproved(f, e, a, y_a, b)
    print(result
```

result: `3.5221932136912915`

4) Пример 4:

```
if __name__ == '__main__':
    f = 4
    e = 0.001
    a = 0
    y_a = 0
    b = 1
    result = Result.solveByEulerImproved(f, e, a, y_a, b)
    print(result)
```

result: `0.718281375751761`

5) Пример 5:

```
if __name__ == '__main__':
    f = 5
    e = 0.001
    a = 0
    y_a = 0
    b = 1
    result = Result.solveByEulerImproved(f, e, a, y_a, b)
    print(result)
```

result: `0`

6) Пример 6 (Пример работы программы с нулевой точностью):

```
if __name__ == '__main__':
    f = 1
    e = 0
    a = 0
    y_a = 0
    b = 1
    result = Result.solveByEulerImproved(f, e, a, y_a, b)
    print(result)
```

result: `0`

7) Пример 7 (Пример работы программы на отрицательной точности):

```
if __name__ == '__main__':
    f = 1
    e = -0.001
    a = 0
    y_a = 0
    b = 1
    result = Result.solveByEulerImproved(f, e, a, y_a, b)
    print(result)
```

result: `1.9999997129664937`

Вывод

Усовершенствованный метод Эйлера представляет собой численный метод для решения обыкновенных дифференциальных уравнений с заданным начальным значением. Этот метод обеспечивает адаптивный выбор шага, что позволяет повысить точность результата. Результаты его работы зависят от выбора начального значения шага и точности. По сравнению с классическим методом Эйлера, усовершенствованный метод обладает большей точностью, хотя и может быть менее эффективным по сравнению с более точными методами, такими как метод Рунге-Кутты. Однако он остается простым в реализации и может быть полезным инструментом для решения широкого класса дифференциальных уравнений при относительно простых условиях.