МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра вычислительные системы и технологии

Приближенное решение обыкновенных дифференциальных
уравнений методом Эйлера с пересчетом

Отчет

по лабораторной работе №6

по дисциплине

Вычислительная математика

РУКОВОДИТЕЛЬ:

________________ Суркова А. С.

СТУДЕНТ:

________________ Соляник Д. Р.

19-ИВТ-2

Работа защищена «___» _____________

С оценкой _________________________

Нижний Новгород 2021

1. Тема лабораторной работы:

Приближенное решение обыкновенных дифференциальных уравнений методом Эйлера с пересчетом

Цель лабораторной работы

Закрепление знаний и умений по численному решению обыкновенных дифференциальных уравнений методом Эйлера и методом Адамса.

2. Вариант задания на лабораторную работу

Вариант № 18

Задание 1

Используя метод Эйлера и метод Эйлера с пересчетом, составить таблицу приближенных значений интеграла дифференциального уравнения y′=f(x,y), удовлетворяющего начальным условиям y(x0) = y0 на отрезке [a,b]; шаг h=0.1. Все вычисления вести с четырехзначными знаками. Проверить полученные значения, используя метод Рунге-Кутты 4 порядка

$$18.\ y' = x + \sin\frac{y}{\sqrt{2}},\ y_0(0.8) = 1.3\ ,\ x \in [0.8;1.8]$$

Задание 2

Используя метод Адамса с третьими разностями составить таблицу приближенных значений интеграла дифференциального уравнения y′=f(x,y), удовлетворяющего начальным условиям y(x0) = y0 на отрезке [0,1]; шаг h=0.1. Все вычисления вести с четырехзначными знаками. Начальный отрезок определить методом Рунге-Кутты. Проверить полученные значения, используя метод Эйлера с пересчетом.

$$18.\ y' = 1 + (1-x)\sin y - (2+x)y\ ,\ y(0) = 0$$

3. Теоретические сведения и описание лабораторной работы

- *Метод Эйлера*

Метод Эйлера — простейший численный метод решения систем обыкновенных дифференциальных уравнений. Метод Эйлера является явным, одношаговым методом первого порядка точности. Он основан на аппроксимации интегральной кривой кусочно-линейной функцией, так называемой ломаной Эйлера.

Пусть дана задача Коши для уравнения первого порядка:

$$\frac{dy}{dx} = f(x, y),$$
$$y_{|_{x=x_0}} = y_0,$$

где функция f определена на некоторой области $D \subset \mathbb{R}^2$. Решение ищется на интервале (x0, b]. На этом интервале введем узлы: $x_0 < x_1 < \cdots < x_n \leq b.$

Приближенное решение в узлах xi, которое обозначим через yi, определяется по формуле:

$$y_i = y_{i-1} + (x_i - x_{i-1}) f(x_{i-1}, y_{i-1}), \quad i = 1, 2, 3, \ldots, n.$$

Эти формулы непосредственно обобщаются на случай систем обыкновенных дифференциальных уравнений.

- *Метод Эйлера с пересчетом*

Повысить точность и устойчивость вычисления решения можно с помощью неявного метода Эйлера следующего вида.

Прогноз:

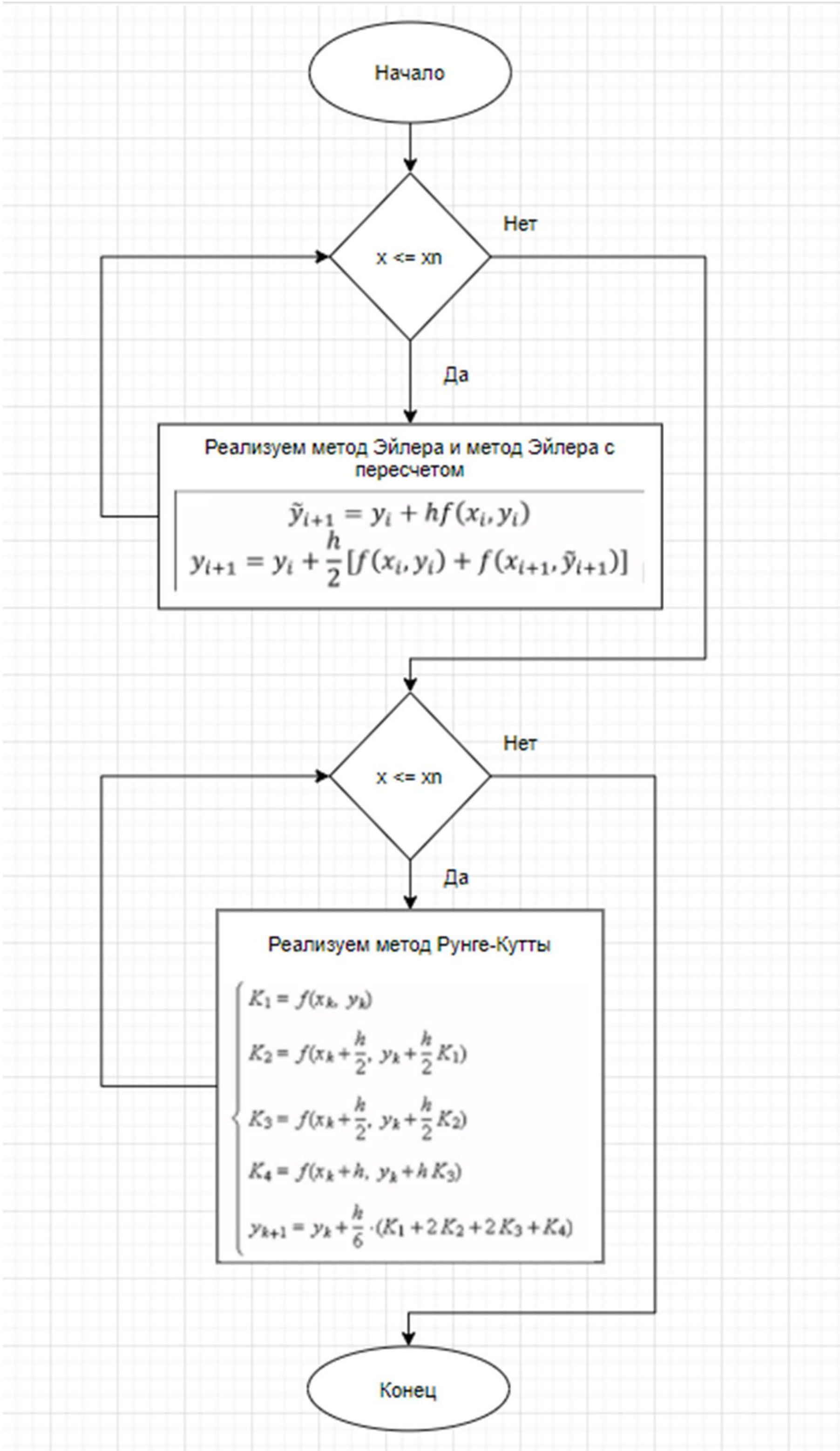
$$\tilde{y}_i = y_{i-1} + (x_i - x_{i-1}) f(x_{i-1}, y_{i-1}).$$

Коррекция:

$$y_i = y_{i-1} + (x_i - x_{i-1}) \frac{f(x_{i-1}, y_{i-1}) + f(x_i, \tilde{y}_i)}{2}.$$

Для повышения точности корректирующую итерацию можно повторить, подставляя $\tilde{y}_i = y_i$.

Модифицированный метод Эйлера с пересчетом имеет второй порядок точности, однако для его реализации необходимо как минимум дважды вычислять f(x,y). Метод Эйлера с пересчетом представляет собой разновидность методов Рунге-Кутты (предиктор-корректор).


Начало
x <= xn
Нет
Да
Реализуем метод Эйлера и метод Эйлера с пересчетом
$\tilde{y}_{i+1} = y_i + hf(x_i, y_i)$
$y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+1}, \tilde{y}_{i+1})]$
x <= xn
Нет
Да
Реализуем метод Рунге-Кутты
$K_1 = f(x_k,\ y_k)$
$K_2 = f(x_k + \frac{h}{2},\ y_k + \frac{h}{2}K_1)$
$K_3 = f(x_k + \frac{h}{2},\ y_k + \frac{h}{2}K_2)$
$K_4 = f(x_k + h,\ y_k + hK_3)$
$y_{k+1} = y_k + \frac{h}{6}\cdot(K_1 + 2K_2 + 2K_3 + K_4)$
Конец

- *Метод Рунге — Кутты*

$$\begin{cases} K_1 = f(x_k,\ y_k) \\ K_2 = f(x_k + \frac{h}{2},\ y_k + \frac{h}{2} K_1) \\ K_3 = f(x_k + \frac{h}{2},\ y_k + \frac{h}{2} K_2) \\ K_4 = f(x_k + h,\ y_k + h K_3) \\ y_{k+1} = y_k + \frac{h}{6} \cdot (K_1 + 2K_2 + 2K_3 + K_4) \end{cases}$$

- *Метод Адамса*

Метод Адамса - конечноразностный многошаговый метод численного интегрирования обыкновенных дифференциальных уравнений первого порядка. В отличие от метода Рунге-Кутты использует для вычисления очередного значения искомого решения не одно, а несколько значений, которые уже вычислены в предыдущих точках.

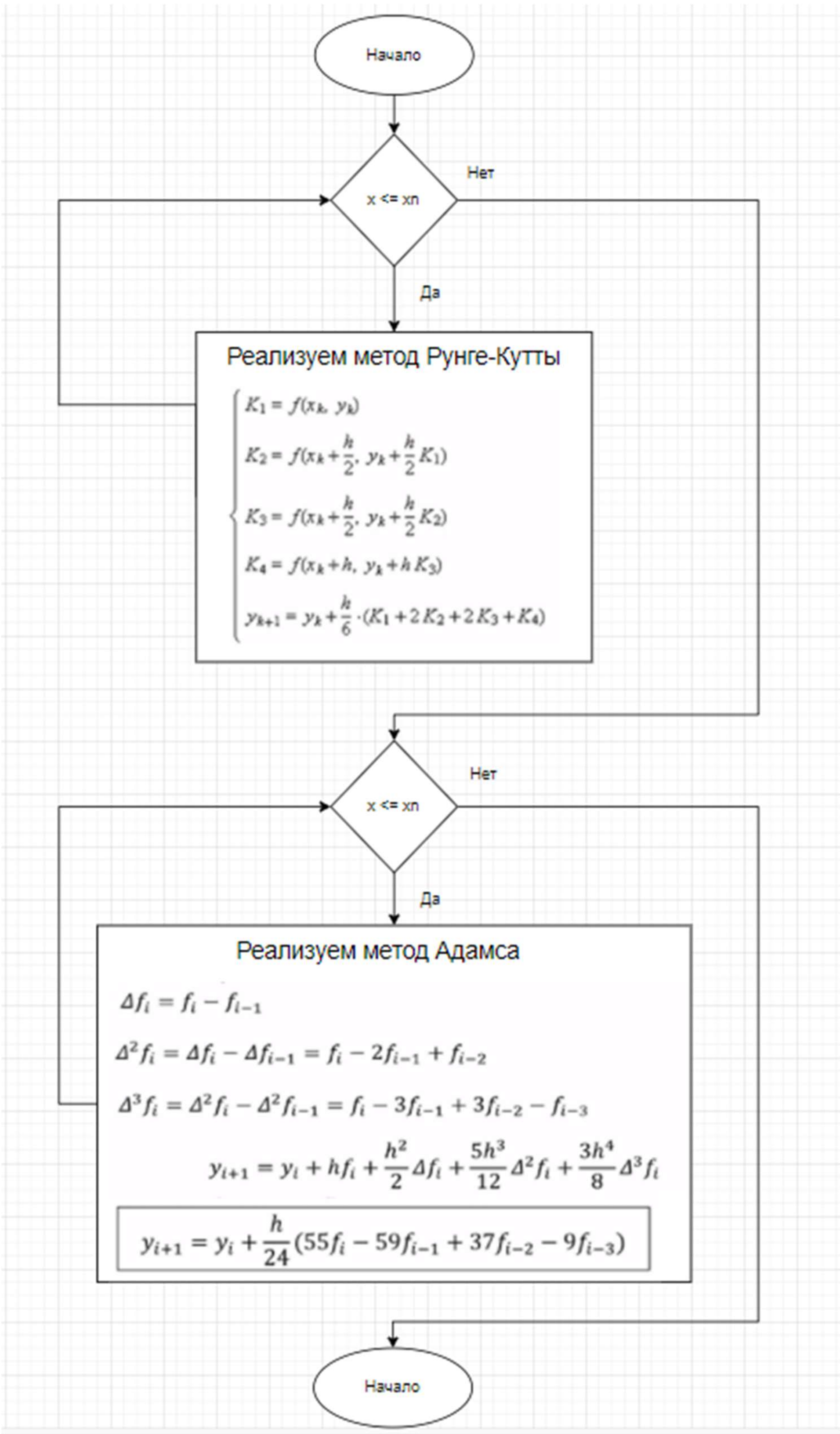
$$\Delta f_i = f_i - f_{i-1}$$

$$\Delta^2 f_i = \Delta f_i - \Delta f_{i-1} = f_i - 2f_{i-1} + f_{i-2}$$

$$\Delta^3 f_i = \Delta^2 f_i - \Delta^2 f_{i-1} = f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3}$$

$$y_{i+1} = y_i + hf_i + \frac{h^2}{2}\Delta f_i + \frac{5h^3}{12}\Delta^2 f_i + \frac{3h^4}{8}\Delta^3 f_i$$

$$\boxed{y_{i+1} = y_i + \frac{h}{24}(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3})}$$


Начало
x <= xn
Нет
Да
Реализуем метод Рунге-Кутты
$K_1 = f(x_k,\ y_k)$
$K_2 = f(x_k+\frac{h}{2},\ y_k+\frac{h}{2}K_1)$
$K_3 = f(x_k+\frac{h}{2},\ y_k+\frac{h}{2}K_2)$
$K_4 = f(x_k+h,\ y_k+hK_3)$
$y_{k+1} = y_k+\frac{h}{6}\cdot(K_1+2K_2+2K_3+K_4)$
x <= xn
Нет
Да
Реализуем метод Адамса
$\Delta f_i = f_i - f_{i-1}$
$\Delta^2 f_i = \Delta f_i - \Delta f_{i-1} = f_i - 2f_{i-1} + f_{i-2}$
$\Delta^3 f_i = \Delta^2 f_i - \Delta^2 f_{i-1} = f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3}$
$y_{i+1} = y_i + hf_i + \frac{h^2}{2}\Delta f_i + \frac{5h^3}{12}\Delta^2 f_i + \frac{3h^4}{8}\Delta^3 f_i$
$\boxed{y_{i+1} = y_i + \frac{h}{24}(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3})}$
Начало

4. Листинг разработанной программы

- *Метод Эйлера, метод Эйлера с пересчетом и метод Рунге-Кутты*

```
import math
from sympy import *

x = Symbol("x")
y = Symbol("y")

def func(x, y):
    return x + math.sin(y / sqrt(2))

def printData(x, data):
    for i in range(len(data)):
        print("The value at", "%.1f" % x[i], "is", "%.4f" % data[i])

# Задаем начальные данные
x = 0.8
xn = 1.8
h = 0.1
Y = 1.3
y = Y

euler = []
euler.append(Y)
euler_recal = []
euler_recal.append(y)
x_value = []
x_value.append(x)

# Цикл реализует метож Эйлера и метод Эйлера с пересчетом
while x <= xn:
```

```
    # Считаем значение функции методом Эйлера
    Y = y + h * func(x, y)
    euler.append(Y)
    y += 0.5 * h * (func(x, y) + func(x + h, Y))
    # Пересчитываем полученное значение
    euler_recal.append(y)
    x += h
    x_value.append(x)

# Задаем данные для проверки
x = 0.8
y = 1.3
runge_kutte = []
runge_kutte.append(y)

# Цикл реализует метод Рунге-Кутты
while x <= xn:
    k1 = func(x, y)
    k2 = func(x + h / 2, y + (h * k1) / 2)
    k3 = func(x + h / 2, y + (h * k2) / 2)
    k4 = func(x + h, y + (h * k3))
    y += h * (k1 + 2 * k2 + 2 * k3 + k4) / 6
    runge_kutte.append(y)
    x += h

print("Euler:")
printData(x_value, euler)
print("Euler recalculation:")
printData(x_value, euler_recal)
print("Runge-Kutte:")
printData(x_value, runge_kutte)    print("", x[i], "|", "%.4f" % y[i][0], "|", "%.4f" %
proiz_1(x[i]),"|", "%.4f" % proiz_2(x[i]))
```

- *Метод Рунге-Кутты, метод Адамса и метод Эйлера с пересчетом*

```
import math
from sympy import *


x = Symbol("x")
y = Symbol("y")


def func(x, y):
	"Задаем данную функцию"
	return round(1 + (1 - x) * math.sin(y) - (2 + x) * y, 4)


# Задаем начальные данные
h = 0.1
x = []
y = [0]
yi = [0]



# Считаем первые 3 значения функции метод Рунге-Кутты
def rungeKutte(x, y):
	k1 = func(x, y)
	k2 = func(x + h / 2, y + (h * k1) / 2)
	k3 = func(x + h / 2, y + (h * k2) / 2)
	k4 = func(x + h, y + (h * k3))
	return y + h * (k1 + 2 * k2 + 2 * k3 + k4) / 6


# Считаем остольные значения функции метод Адамса
def adams(x, y, i):
	return y[i-1] + h / 24 * (55 * func(x[i-1],y[i-1]) - 59 * func(x[i-2],y[i-2]) + 37 * func(x[i-3],y[i-3]) - 9 * func(x[i-4],y[i-4]))
```

```
# Считаем значения функции метод Эйлера с пересчетом
def euler(x, y, X):
    Y = y + h * func(x, y)
    return y + 0.5 * h * (func(x, y) + func(X, Y))

# Считаем значения x с шагом h
for i in range(11):
    x.append(i * h)

print("Adams:")
for i in range(1,4):
    y.append(rungeKutte(x[i-1], y[i-1]))
    print("The value at", "%.1f" % x[i], "is", "%.4f" % y[i])

for i in range(4,11):
    y.append(adams(x, y, i))
    print("The value at", "%.1f" % x[i], "is", "%.4f" % y[i])

print("Euler recalculation:")
for i in range(1,11):
    yi.append(euler(x[i-1], yi[i-1], x[i]))
    print("The value at", "%.1f" % x[i], "is", "%.4f" % yi[i])
```

5. Результаты работы программы

- *Метод Эйлера, метод Эйлера с пересчетом и метод Рунге-Кутты*

```
Euler:
The value at 0.8 is 1.3000
The value at 0.9 is 1.4595
The value at 1.0 is 1.6438
The value at 1.1 is 1.8436
The value at 1.2 is 2.0574
The value at 1.3 is 2.2832
The value at 1.4 is 2.5183
The value at 1.5 is 2.7600
The value at 1.6 is 3.0054
The value at 1.7 is 3.2515
The value at 1.8 is 3.4959
Euler recalculation:
The value at 0.8 is 1.3000
The value at 0.9 is 1.4677
The value at 1.0 is 1.6516
The value at 1.1 is 1.8508
The value at 1.2 is 2.0638
The value at 1.3 is 2.2884
The value at 1.4 is 2.5223
The value at 1.5 is 2.7626
The value at 1.6 is 3.0065
The value at 1.7 is 3.2513
The value at 1.8 is 3.4946
Runge-Kutte:
The value at 0.8 is 1.3000
The value at 0.9 is 1.4679
The value at 1.0 is 1.6520
The value at 1.1 is 1.8514
The value at 1.2 is 2.0646
The value at 1.3 is 2.2895
The value at 1.4 is 2.5235
The value at 1.5 is 2.7640
The value at 1.6 is 3.0080
The value at 1.7 is 3.2530
The value at 1.8 is 3.4964
```

- *Метод Рунге-Кутты, метод Адамса и метод Эйлера с пересчетом*

```
Adams:
The value at 0.1 is 0.0945
The value at 0.2 is 0.1766
The value at 0.3 is 0.2446
The value at 0.4 is 0.2980
The value at 0.5 is 0.3371
The value at 0.6 is 0.3631
The value at 0.7 is 0.3775
The value at 0.8 is 0.3822
The value at 0.9 is 0.3792
The value at 1.0 is 0.3704
Euler recalculation:
The value at 0.1 is 0.0940
The value at 0.2 is 0.1755
The value at 0.3 is 0.2430
The value at 0.4 is 0.2960
The value at 0.5 is 0.3348
The value at 0.6 is 0.3606
The value at 0.7 is 0.3749
The value at 0.8 is 0.3797
The value at 0.9 is 0.3769
The value at 1.0 is 0.3684
```

6. Вывод

В ходе данной лабораторной работы были закреплены знания и умения по численному решению обыкновенных дифференциальных уравнений методом Эйлера и методом Адамса.