Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина: Методы численного анализа

ОТЧЁТ
к лабораторной работе №9
на тему

«МЕТОДЫ ЭЙЛЕРА И РУНГЕ-КУТТА»

Выполнил студент группы 253505
Снежко Максим Андреевич

____
(дата, подпись студента)

Проверил доцент кафедры
информатики
Анисимов Владимир Яковлевич

____
(дата, подпись преподавателя)

Минск 2023

Содержание

Цели работы

1. Изучить решение задачи Коши для обыкновенных дифференциальных уравнений методом Эйлера и методом Рунге-Кутта.
2. Составить алгоритм решения
3. Составить программную реализацию методов
4. Проверить правильность работы программы на тестовых примерах
5. Выполнить задание в соответствии с вариантом

Теоретические сведения

Рассмотрим дифференциальное уравнение $y' = f(x, y)$ с начальным условием $y(x_0) = y_0$. Будем предполагать, что *f(x,y)* непрерывная и непрерывно дифференцируемая по y функция в окрестности замкнутой области

$$D = \{(x, y) \,|\, a \le x \le b, c \le y \le d\},$$

содержащей внутри себя точку (x_0, y_0).

Требуется решить задачу Коши: найти непрерывно дифференцируемую функцию *y=y(x)*, такую что $y'(x) = f(x, y(x))$ при всех $x \in [a, b]$ и $y(x_0) = y_0$.

Разобьем отрезок *[a, b]* с помощью точек разбиения $a = x_0, x_1, ..., x_n = b$ с шагом $h = (b - a)/n$. Тогда узлы разбиения имеют вид $x_k = x_0 + kh,\ k = \overline{0, n}$.

Пусть $y(x_0), y(x_1), ..., y(x_n)$ - значения функции в точках разбиения.

1) Метод Эйлера

Построим рекуррентную последовательность:

$$y_{k+1} = y_k + hf(x_k, y_k), \qquad k = 0,1... \qquad (9.1)$$

$$y_0 = y(x_0),$$

которую называют последовательностью Эйлера. Соединяя ломаными все точки (x_k, y_k), полученные из рекуррентной последовательности Эйлера, получим ломаную линию, приближающую график решения $y = y(x)$. Функция, график которой совпадает с ломаной Эйлера, принимается за приближенное решение задачи Коши.

Точность метода Эйлера на всем отрезке $[a, b]$ будет $O(h)$.

Для повышения точности вычислений иногда используется модифицированный метод Эйлера, в котором рекуррентная последовательность Эйлера вычисляется по формулам

$$y_{k+1} = y_k + hf(x_k + \frac{h}{2}, y_k + \frac{h}{2} f(x_k, y_k)), \qquad k = 0,1...,n-1. \qquad (9.2)$$

Модифицированный метод Эйлера обычно дает более точное приближение решения.

Пример. Пусть требуется решить задачу Коши:

$$\begin{cases} y' = -y, \ \ x \in [0,1] \\ y(0) = 1. \end{cases}$$

Полагая $h = 0{,}2$ и используя метод Эйлера, получим, как легко убедиться, из формулы Эйлера (9.1)

$$y_{k+1} = y_k + 0.2 \cdot (-y_k) = 0.8 \cdot y_k .$$

С другой стороны, используя модифицированный метод Эйлера, получим в силу формулы (2) рекуррентную последовательность

$$y_{k+1} = y_k + 0.2 \cdot (-y_k) = 0.82 \cdot y_k \ .$$

Поскольку точным решением задачи Коши, как легко проверить, является функция $y = e^{-x}$, можно сравнить точность обоих методов.

	0	1	2	3	4	5
x_k	0	0.2	0.4	0.6	0.8	1
y_k	1	0.8	0.64	0.572	0.4086	0.3277
$y_k^{модиф}$	1	0.82	0.6724	0.5514	0.4521	0.3708

e^{-x}	1	0.8187	0.6703	0.5488	0.4493	0.3679

Общепризнанным недостатком метода Эйлера является его не достаточно высокая точность. Несомненным достоинством метода Эйлера является его простота.

2) Метод Рунге-Кутта четвертого порядка.

На каждом шаге производится вычисление коэффициентов K_1, K_2, K_3, K_4:

$$K_1 = hf(x_k, y_k);$$

$$K_2 = hf(x_k + \frac{h}{2}, y_k + \frac{K_1}{2});$$

$$K_3 = hf(x_k + \frac{h}{2}, y_k + \frac{K_2}{2});$$

$$K_4 = hf(x_k + h, y_k + K_3).$$

Затем вычисляем

$$y_{k+1} = y_k + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4).$$

Данный метод имеет точность $O(h^4)$ на $[a,b]$.

Рассмотрим пример, который мы использовали для иллюстрации точности метода Эйлера.

Пример. Требуется решить задачу Коши:

$$\begin{cases} y' = -y \\ y(0) = 1 \end{cases} \text{ на отрезке } [0,\ 1].$$

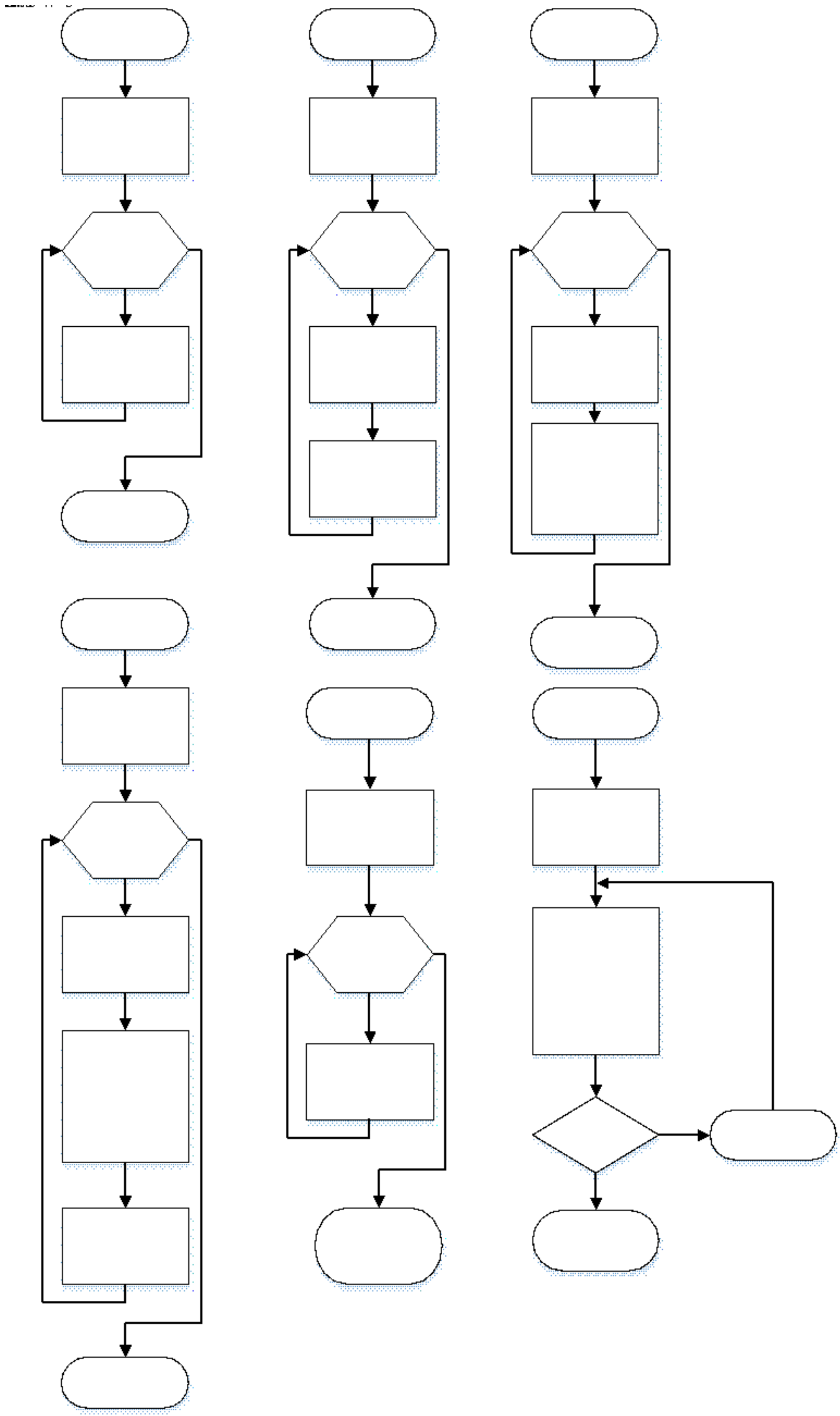
Выберем шаг $h = 0{,}2$. Результат вычислений поместим в таблицу.

	0	1	2	3	4	5
x_k	0	0.2	0.4	0.6	0.8	1
y_k	1	0.8187	0.6703	0.5487	0.4493	0.3678
e^{-x}	1	0.8187	0.6703	0.5488	0.4493	0.3679

Таким образом, метод Рунге-Кутта 4-го порядка отличается очень высокой точностью. К определенным его недостаткам относится большая сложность и трудоемкость (на каждом шаге необходимо четырежды вычислять значения функции f вместо одного раза в методе Эйлера).

Отметим, что на практике выбирают начальную длину шага h таким образом, чтобы $h^4 < \varepsilon$, где ε - заданная точность вычисления решения. Затем шаг выбирают вдвое меньшим и останавливают вычисления, если разность полученных значений y_k со значениями, полученными при начальном выборе шага меньше ε. В противном случае шаг еще раз уменьшают вдвое и т.д.

Алгоритм решения



Программная реализация

```
def euler(xdot, N, y0, y_diff):
    ydots = [y0]
    h = xdot / N
    for i in range(N):
        x = i * h
        y = ydots[-1]
        ydots += [y + h * y_diff(x, y)]
    return ydots


2 usages
def modifiedEuler(xdot, N, y0, y_diff):
    ydots = [y0]
    h = xdot / N
    for i in range(N):
        x = i * h
        y = ydots[-1]
        ydots += [y + h * y_diff(x + h / 2, y + h / 2 * y_diff(x, y))]
    return ydots
def rungeKutt(xdot, N, y0, y_diff):
    ydots = [y0]
    h = xdot / N
    for i in range(N):
        x = i * h
        y = ydots[-1]
        K1 = h * y_diff(x, y)
        K2 = h * y_diff(x + h / 2, y + K1 / 2)
        K3 = h * y_diff(x + h / 2, y + K2 / 2)
        K4 = h * y_diff(x + h, y + K3)
        ydots += [y + 1/6 * (K1 + 2*K2 + 2*K3 + K4)]
    return ydots

def getValueAtPoint(method, x, y0, y_diff, eps):
    n = 1
    old_dots, new_dots = method(x, n, y0, y_diff), method(x, 2 * n, y0, y_diff)
    while max(abs(new_dots[2*i] - old_dots[i]) for i in range(n + 1)) > eps:
        n *= 2
        old_dots, new_dots = method(x, n, y0, y_diff), method(x, 2 * n, y0, y_diff)
    return new_dots[-1], 2 * n
```

```python
def createYDots(method, xdots, y0, y_diff, eps):
    ydots = []
    maxn = 0
    midn = []
    for x in xdots:
        y, n = getValueAtPoint(method, x, y0, y_diff, eps)
        ydots.append(y)
        maxn = max(maxn, n)
        midn += [n]
    midn = sum(midn) / len(xdots)
    return ydots, midn, maxn
```

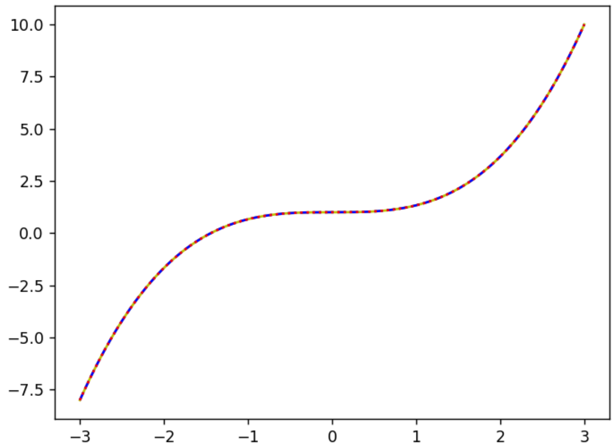
Тестовые примеры

Тестовый пример 1. С помощью метода Эйлера, модифицированного метода Эйлера, метода Рунге-Кутта найти с заданной точностью решение заданного уравнения на заданном отрезке. Сравнить результаты.

Входные данные: $y' = x^2$, [-3,3], y(0) = 1

Вывод программы:

```
Число точек:  1000
Точность:   0.001
Метод Эйлера:
x[1]: -2.4
y[1]: -3.607156284332276

x[4]: -0.6000000000000001
y[4]: 0.928841552734375

x[7]: 1.2000000000000002
y[7]: 1.575156524658203

Максимальное количество отрезков (n): 16384
Среднее количество отрезков (n): 4889.364635364635

Модифицированный метод Эйлера:
x[1] = -2.4
y[1] = -3.60771875

x[4] = -0.6000000000000001
y[4] = 0.92828125

x[7] = 1.2000000000000002
y[7] = 1.575859375

Максимальное количество отрезков (n): 128
Среднее количество отрезков (n): 49.06893106893107
```



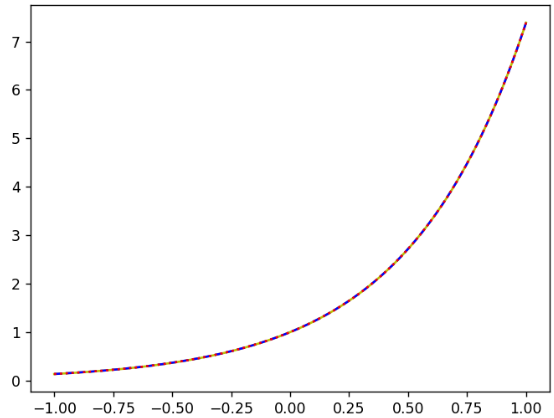

```
Метод Рунге-Кутта:
x[1] = -2.4
y[1] = -3.6079999999999983

x[4] = -0.6000000000000001
y[4] = 0.9279999999999999

x[7] = 1.2000000000000002
y[7] = 1.5760000000000003

Максимальное количество отрезков (n): 2
Среднее количество отрезков (n): 2.0
```

Тестовый пример 2. С помощью метода Эйлера, модифицированного метода Эйлера, метода Рунге-Кутта найти с заданной точностью решение заданного уравнения на заданном отрезке. Сравнить результаты.

Входные данные: $y' = 2y$, [-1,1], y(0) = 1

Вывод программы:

```
Число точек:  1000
Точность:   0.001
Метод Эйлера:
x[1]: -0.8
y[1]: 0.20139135572023567

x[4]: -0.19999999999999996
y[4]: 0.6694791661408149

x[7]: 0.40000000000000013
y[7]: 2.224845917514761

Максимальное количество отрезков (n): 16384
Среднее количество отрезков (n): 2400.945054945055

Модифицированный метод Эйлера:
x[1] = -0.8
y[1] = 0.2020363123152348

x[4] = -0.19999999999999996
y[4] = 0.6704360492918494

x[7] = 0.40000000000000013
y[7] = 2.225358916832791

Максимальное количество отрезков (n): 256
Среднее количество отрезков (n): 48.527472527472526
```



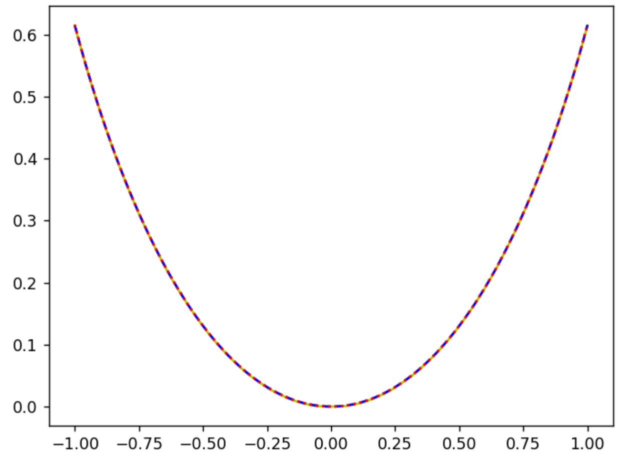

```
Метод Рунге-Кутта:
x[1] = -0.8
y[1] = 0.20190160831589016

x[4] = -0.19999999999999996
y[4] = 0.6703242711111113

x[7] = 0.40000000000000013
y[7] = 2.225520825778562

Максимальное количество отрезков (n): 16
Среднее количество отрезков (n): 5.8321678321678325
```

Тестовый пример 3. С помощью метода Эйлера, модифицированного метода Эйлера, метода Рунге-Кутта найти с заданной точностью решение заданного уравнения на заданном отрезке. Сравнить результаты.

Входные данные: $y' = \tan x$, [-1,1], y(0) = 0

Вывод программы:

```
Число точек:  1000
Точность:   0.001
Метод Эйлера:
x[1]: -0.8
y[1]: 0.36058655737772266

x[4]: -0.19999999999999996
y[4]: 0.019501437951658214

x[7]: 0.40000000000000013
y[7]: 0.08156855014102604

Максимальное количество отрезков (n): 1024
Среднее количество отрезков (n): 313.6283716283716

Модифицированный метод Эйлера:
x[1] = -0.8
y[1] = 0.36128042930315074

x[4] = -0.19999999999999996
y[4] = 0.020117692643383377

x[7] = 0.40000000000000013
y[7] = 0.08193418433901481

Максимальное количество отрезков (n): 32
Среднее количество отрезков (n): 7.242757242757243
```



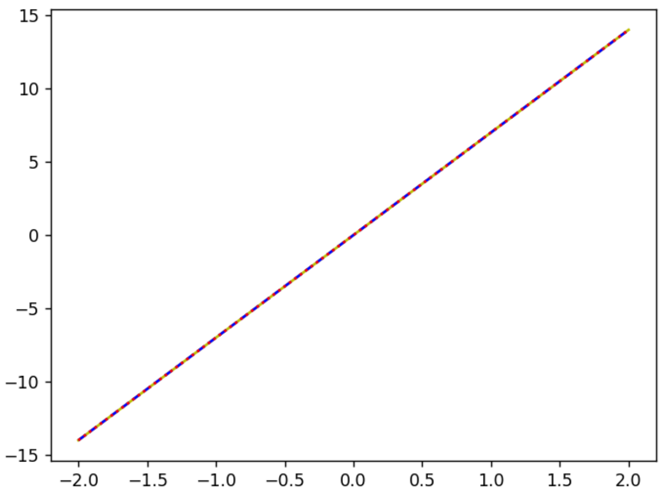

```
Метод Рунге-Кутта:
x[1] = -0.8
y[1] = 0.36139886167450647

x[4] = -0.19999999999999996
y[4] = 0.020134784756915138

x[7] = 0.40000000000000013
y[7] = 0.08222989921786011

Максимальное количество отрезков (n): 4
Среднее количество отрезков (n): 2.4515484515484514
```

Тестовый пример 4. С помощью метода Эйлера, модифицированного метода Эйлера, метода Рунге-Кутта найти с заданной точностью решение заданного уравнения на заданном отрезке. Сравнить результаты.

Входные данные: $y' = 7$, [-2,2], y(0) = 0

Вывод программы:

```
Число точек:  1000
Точность:   0.001
Метод Эйлера:
x[1]: -1.6
y[1]: -11.200000000000001

x[4]: -0.3999999999999999
y[4]: -2.7999999999999994

x[7]: 0.8000000000000003
y[7]: 5.600000000000001

Максимальное количество отрезков (n): 2
Среднее количество отрезков (n): 2.0

Модифицированный метод Эйлера:
x[1] = -1.6
y[1] = -11.200000000000001

x[4] = -0.3999999999999999
y[4] = -2.7999999999999994

x[7] = 0.8000000000000003
y[7] = 5.600000000000001

Максимальное количество отрезков (n): 2
Среднее количество отрезков (n): 2.0
```




```
Метод Рунге-Кутта:
x[1] = -1.6
y[1] = -11.2

x[4] = -0.3999999999999999
y[4] = -2.7999999999999994

x[7] = 0.8000000000000003
y[7] = 5.600000000000001

Максимальное количество отрезков (n): 2
Среднее количество отрезков (n): 2.0
```

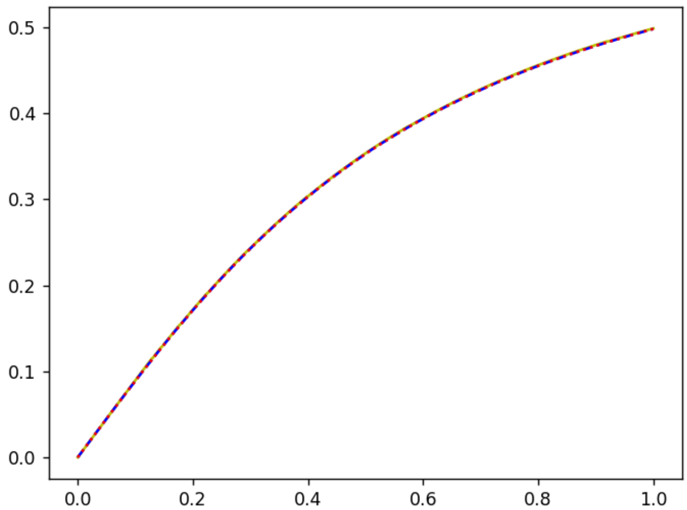
Решение задания
Вариант 8

ЗАДАНИЕ. С помощью метода Эйлера, а затем метода Рунге-Кутта найти с точностью до 0.001 решения следующих уравнений на отрезке [0; 1].

$$y' = \frac{a(1-y^2)}{(1+m)x^2 + y^2 + 1}, \quad y(0) = 0,$$

m=1.5, a=0.9

```
Число точек:  1000
Точность:   0.001
Метод Эйлера:
x[1]: 0.1
y[1]: 0.08954030398452803

x[4]: 0.4
y[4]: 0.30355913429975584

x[7]: 0.7000000000000001
y[7]: 0.4280534885376263

Максимальное количество отрезков (n): 512
Среднее количество отрезков (n): 161.93806193806194

Модифицированный метод Эйлера:
x[1] = 0.1
y[1] = 0.08886460970606444

x[4] = 0.4
y[4] = 0.303099321153505

x[7] = 0.7000000000000001
y[7] = 0.42722004200039176

Максимальное количество отрезков (n): 16
Среднее количество отрезков (n): 8.43956043956044
```



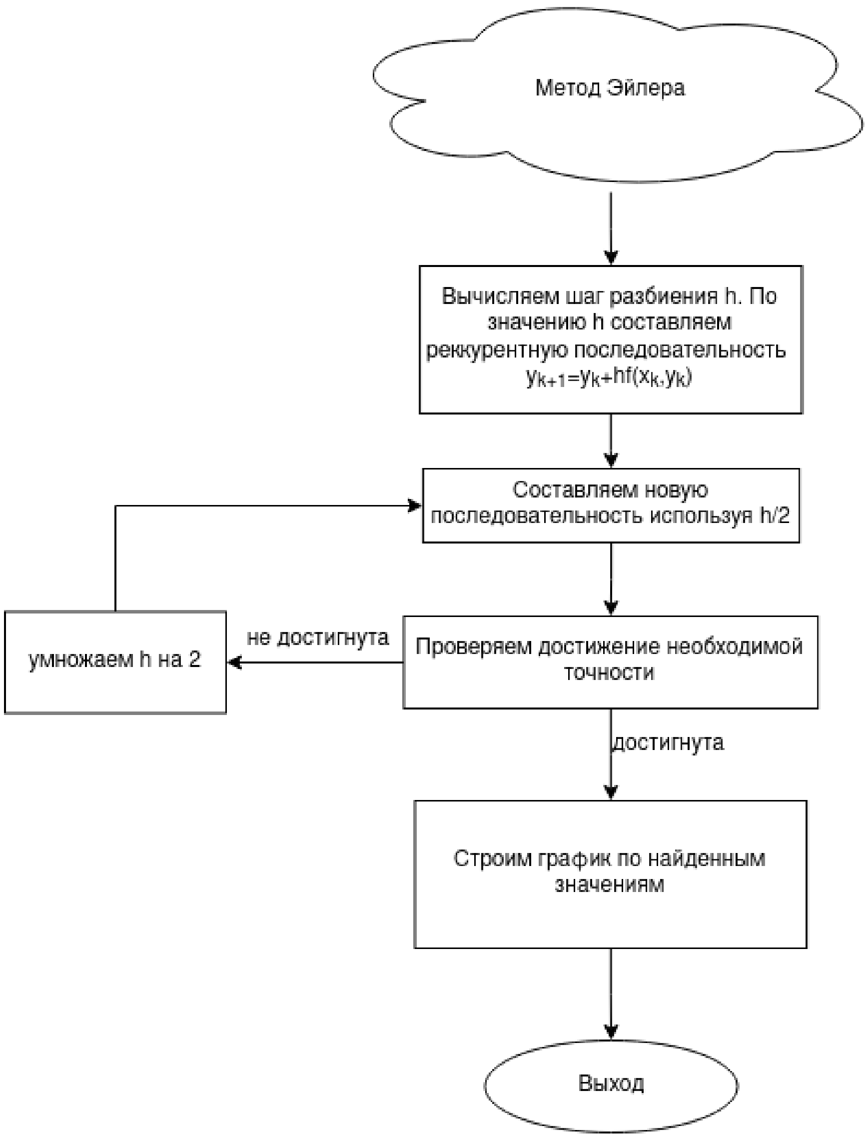

```
Метод Рунге-Кутта:
x[1] = 0.1
y[1] = 0.08879554193787481

x[4] = 0.4
y[4] = 0.30304830127705923

x[7] = 0.7000000000000001
y[7] = 0.42736178013296894

Максимальное количество отрезков (n): 4
Среднее количество отрезков (n): 2.907092907092907
```

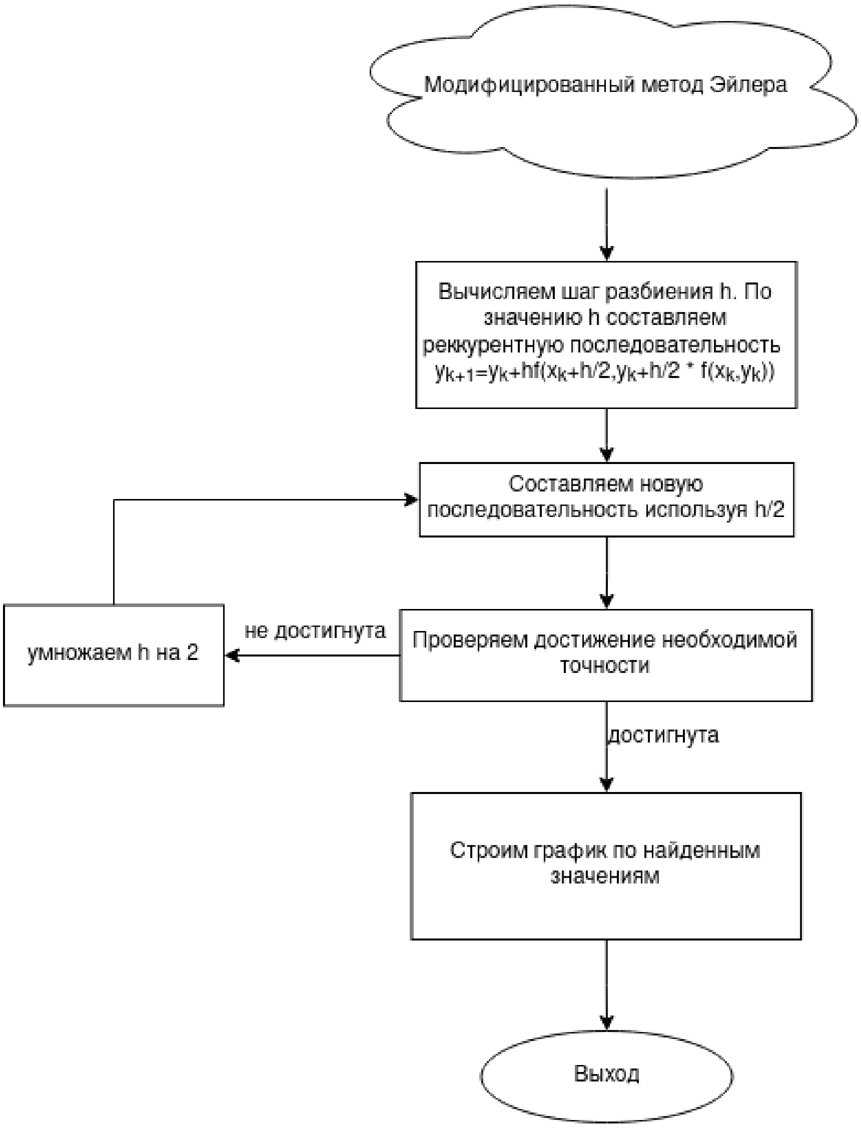
Выводы

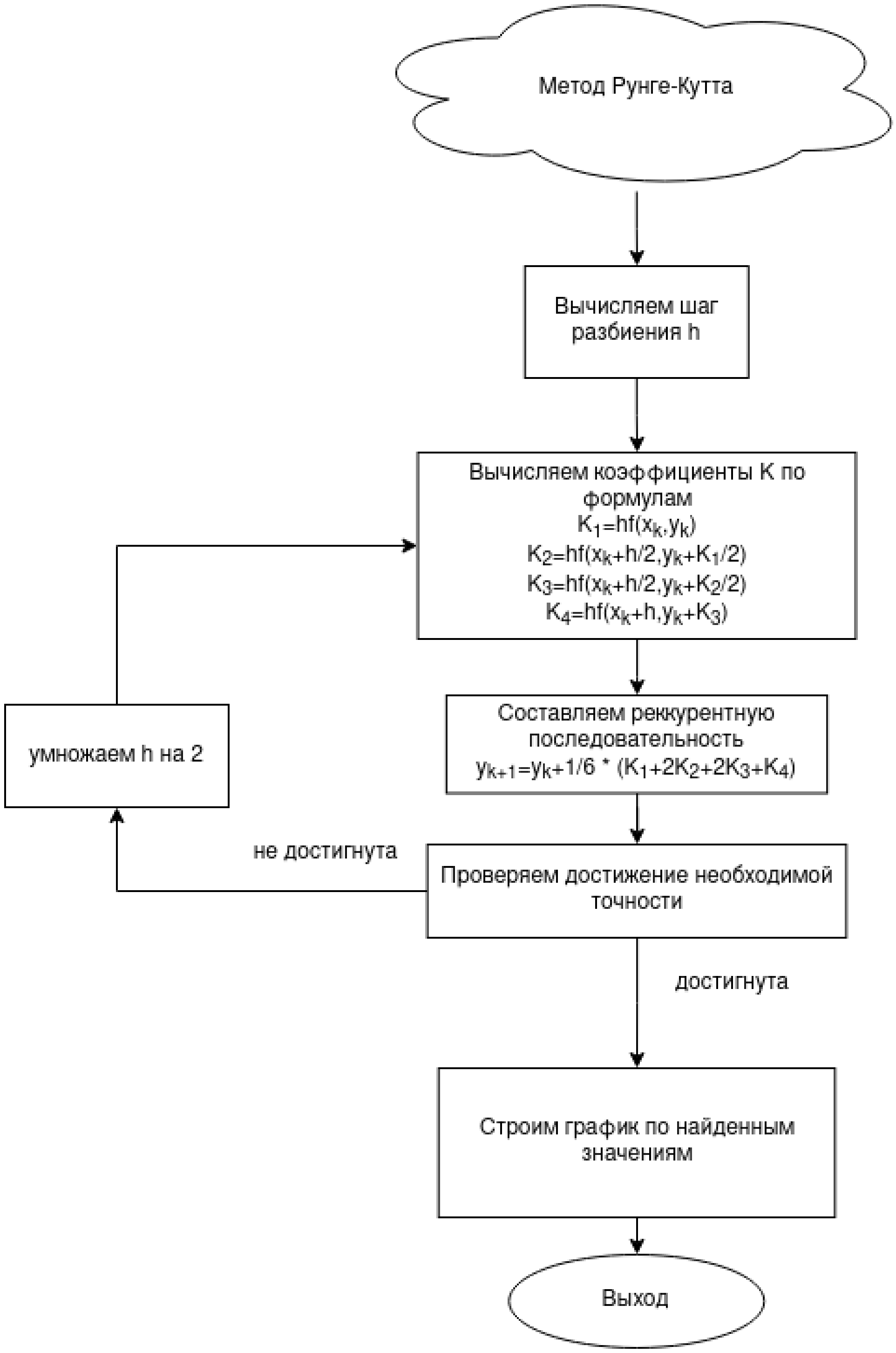
Таким образом, в ходе выполнения лабораторной работы были продемонстрированы метод Эйлера, модифицированный метод Эйлера, метод Рунге-Кутта четвёртого порядка для решения задачи Коши для обыкновенных дифференциальных уравнений. Составлена компьютерная программа, на тестовых примерах проверена правильность её работы, с заданной точностью построены графики решения дифференциального уравнения заданного варианта, по количеству необходимых для этого отрезков сравнена трудоёмкость методов.

Из результата работы программы можем сделать вывод, что метод Рунге-Кутта даёт более точные результаты, чем метод Эйлера. Как и ожидалось, увидели, что модифицированный метод Эйлера точнее, чем обычный метод Эйлера.

Алгоритм решения




Модифицированный метод Эйлера
Вычисляем шаг разбиения h. По значению h составляем реккурентную последовательность $y_{k+1}=y_k+hf(x_k+h/2,y_k+h/2 * f(x_k,y_k))$
Составляем новую последовательность используя h/2
Проверяем достижение необходимой точности
не достигнута
умножаем h на 2
достигнута
Строим график по найденным значениям
Выход


Метод Рунге-Кутта
Вычисляем шаг разбиения h
Вычисляем коэффициенты K по формулам
$K_1=hf(x_k,y_k)$
$K_2=hf(x_k+h/2,y_k+K_1/2)$
$K_3=hf(x_k+h/2,y_k+K_2/2)$
$K_4=hf(x_k+h,y_k+K_3)$
Составляем реккурентную последовательность
$y_{k+1}=y_k+1/6 * (K_1+2K_2+2K_3+K_4)$
умножаем h на 2
не достигнута
Проверяем достижение необходимой точности
достигнута
Строим график по найденным значениям
Выход