«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет программной инженерии и компьютерной техники  
Направление подготовки 09.03.04 Программная инженерия  
Дисциплина «Вычислительная математика»

Отчет

По лабораторной работе №6

Вариант 4

Студент:

*Ильин Н. С.*

*Р3210*

Преподаватель:

*Наумова Н. А.*

Санкт-Петербург, 2025 г.

**Оглавление**

[**Цель работы**: 3](#_Toc198542672)

[**Программная реализация** 3](#_Toc198542673)

[**Блок схемы** 3](#_Toc198542674)

[**Листинг программы** 7](#_Toc198542675)

[**Примеры и результаты работы программы** 8](#_Toc198542676)

[**Выводы**: 9](#_Toc198542677)

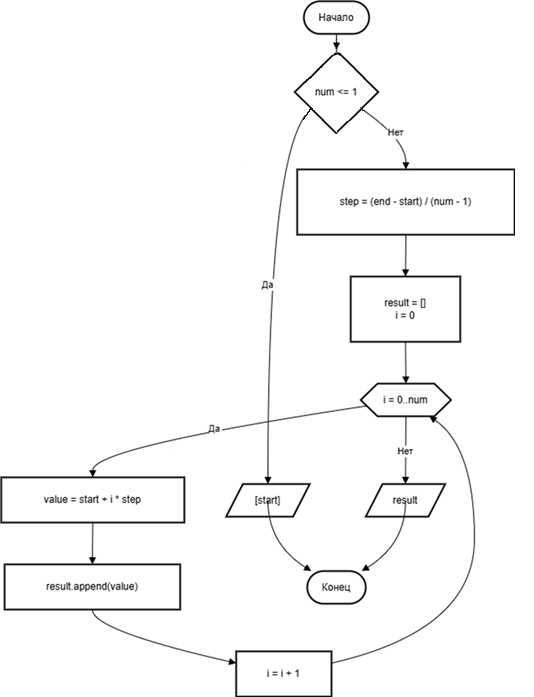
# **Цель работы**:

Решить задачу Коши для обыкновенных дифференциальных уравнений численными методами.

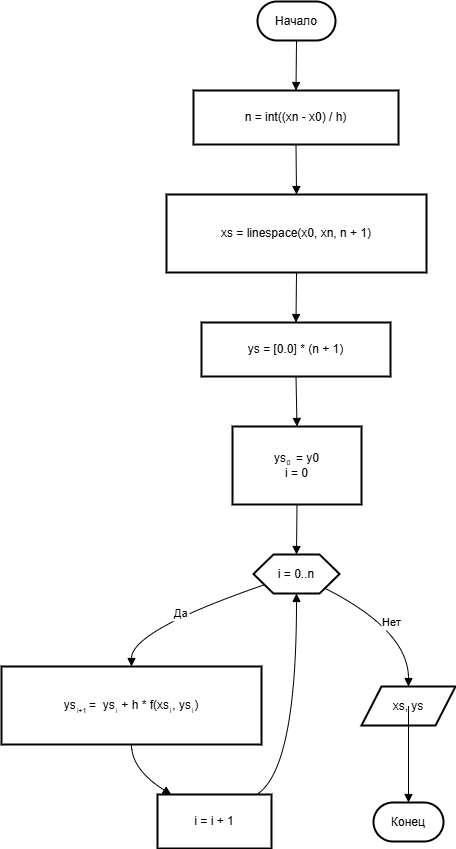
# **Программная реализация**

# **Блок схемы**

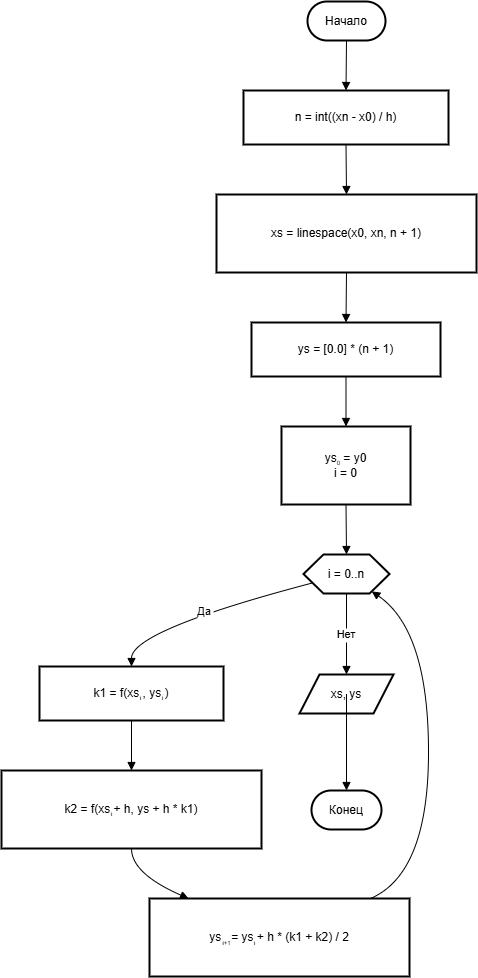
Linespace



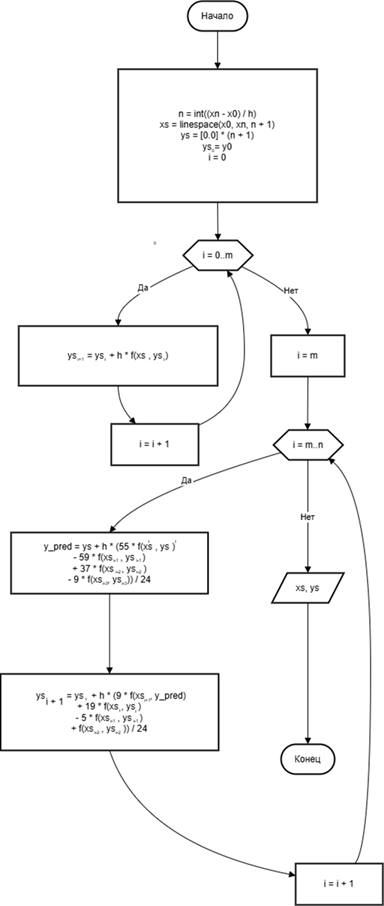
Euler



Improved Euler



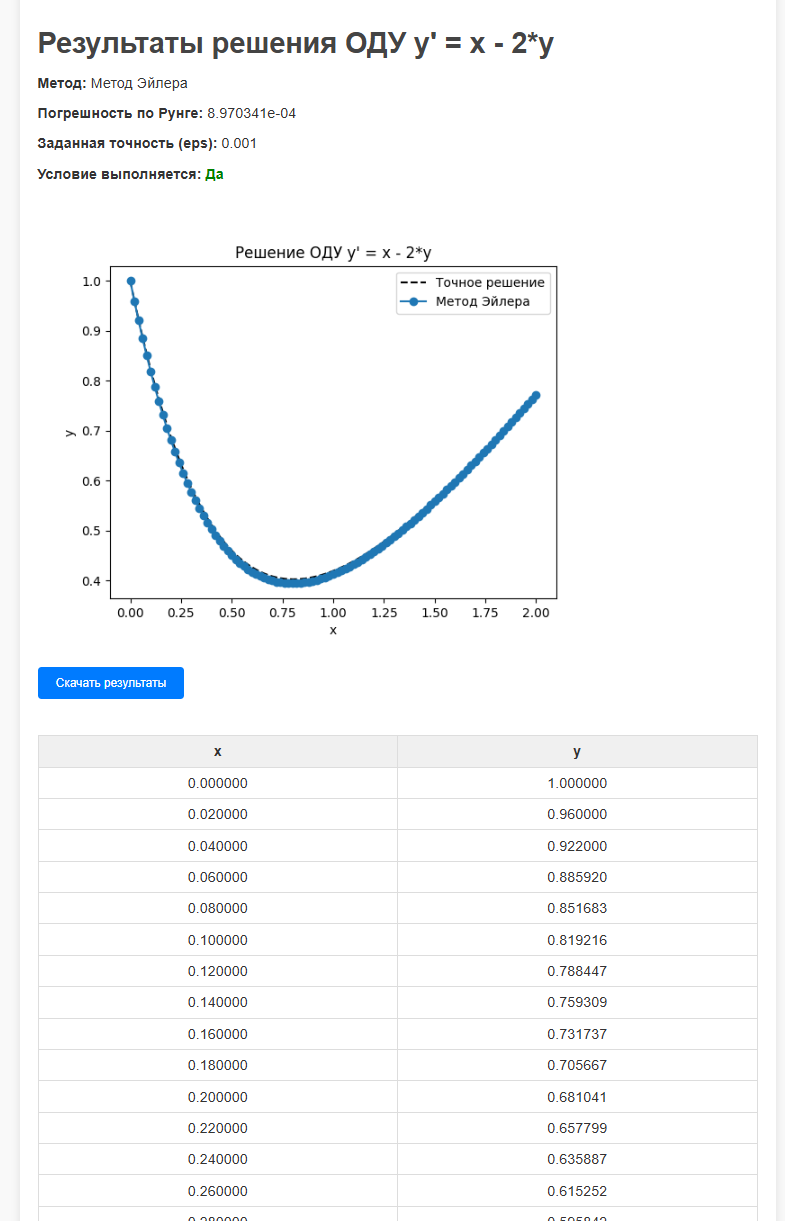
Adams



# **Листинг программы**

|  |
| --- |
| def linespace(start, end, num):  if num <= 1:  return [start]  step = (end - start) / (num - 1)  return [start + i \* step for i in range(num)]  def euler(f, y0, x0, xn, h):  n = int((xn - x0) / h)  xs = linespace(x0, xn, n + 1)  ys = [0.0] \* (n + 1)  ys[0] = y0  for i in range(n):  ys[i + 1] = ys[i] + h \* f(xs[i], ys[i])  return xs, ys  def improved\_euler(f, y0, x0, xn, h):  n = int((xn - x0) / h)  xs = linespace(x0, xn, n + 1)  ys = [0.0] \* (n + 1)  ys[0] = y0  for i in range(n):  k1 = f(xs[i], ys[i])  k2 = f(xs[i] + h, ys[i] + h \* k1)  ys[i + 1] = ys[i] + h \* (k1 + k2) / 2  return xs, ys  def adams(f, y0, x0, xn, h, m=3):  n = int((xn - x0) / h)  xs = linespace(x0, xn, n + 1)  ys = [0.0] \* (n + 1)  ys[0] = y0  for i in range(m):  ys[i + 1] = ys[i] + h \* f(xs[i], ys[i])  for i in range(m, n):  y\_pred = ys[i] + h \* (  55 \* f(xs[i], ys[i])  - 59 \* f(xs[i - 1], ys[i - 1])  + 37 \* f(xs[i - 2], ys[i - 2])  - 9 \* f(xs[i - 3], ys[i - 3])  ) / 24  ys[i + 1] = ys[i] + h \* (  9 \* f(xs[i + 1], y\_pred)  + 19 \* f(xs[i], ys[i])  - 5 \* f(xs[i - 1], ys[i - 1])  + f(xs[i - 2], ys[i - 2])  ) / 24  return xs, ys  def runge\_error(method, f, y0, x0, xn, h, p):  xs1, ys1 = method(f, y0, x0, xn, h)  xs2, ys2 = method(f, y0, x0, xn, h / 2)  return abs(ys1[-1] - ys2[-1]) / (2\*\*p - 1) |

## **Примеры и результаты работы программы**



Github: [https://github.com/MrTheFall/computational\_math/tree/main/lab](https://github.com/MrTheFall/computational_math/tree/main/lab6)6

# **Выводы**:

В рамках лабораторного исследования мной были изучены и практически применены следующие численные методы решения обыкновенных дифференциальных уравнений: метод Эйлера, модифицированный метод Эйлера, метод Адамса.

Для реализации алгоритмов был использован язык Python. Были не только разработаны вычислительные процедуры, но и применено правило Рунге для контроля погрешности в одношаговых методах. Сравнительный анализ точности и эффективности методов выполнен с использованием графиков, которые наглядно продемонстрировали особенности их работы.

Практическая часть работы углубила понимание применения численных подходов для решения ОДУ, а также продемонстрировала важность выбора оптимального метода для конкретных вычислительных задач.