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

Мегафакультет компьютерных технологий и управления
Факультет программной инженерии и компьютерной техники

Отчёт по лабораторной работе №6 по дисциплине «Вычислительная математика»

Вариант №4

Группа: P3218

Студент: Горло Евгений Николаевич

Преподаватель: Бострикова Дарья Константиновна

Санкт-Петербург
2024

Содержание

1 Цель

Решить задачу Коши для обыкновенных дифференциальных уравнений численными методами.

2 Задание

1. В программе численные методы решения обыкновенных дифференциальных уравнений (ОДУ) должен быть реализован в виде отдельного класса / метода / функции;

2. Пользователь выбирает ОДУ вида $y' = f(x, y)$ (не менее трех уравнений), из тех, которые предлагает программа;

3. Предусмотреть ввод исходных данных с клавиатуры: начальные условия, интервал дифференцирования $[x_0, x_n]$, шаг h, точность ϵ;

4. Для исследования использовать одношаговые методы и многошаговые методы (см. табл.1);

5. Составить таблицу приближенных значений интеграла дифференциального уравнения, удовлетворяющего начальным условиям, для всех методов, реализуемых в программе;

6. Для оценки точности одношаговых методов использовать правило Рунге;

7. Для оценки точности многошаговых методов использовать точное решение задачи: $\epsilon = max_{0<=i<=n}|y_{iexact} - y_i|$;

8. Построить графики точного решения и полученного приближенного решения (разными цветами);

9. Программа должна быть протестирована при различных наборах данных, в том числе и некорректных.

10. Проанализировать результаты работы программы.

3 Рабочие формулы

Формула Эйлера:

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

Модифицированння формула Эйлера:

$y_{i+1} = y_i + \frac{h}{2} \cdot (f(x_i, y_i) + f(x_{i+1}, y_i + h \cdot f(x_i, y_i)))$

Формула Рунге-Кутта 4-го порядка:

$y_{i+1} = y_i + \frac{h}{6} \cdot (k_1 + 2k_2 + 2k_3 + k_4)$

$k_1 = h \cdot f(x_i, y_i)$

$k_2 = h \cdot f(x_i + \frac{h}{2}, y_i + \frac{k_1}{2})$

$k_3 = h \cdot f(x_i + \frac{h}{2}, y_i + \frac{k_2}{2})$

$k_4 = h \cdot f(x_i = h, y_i + k_3)$

Формула Адамса:

$\Delta f_1 = f_i - f_{i-1}$

$\Delta f_2 = f_i - 2f_{i-1} + f_{i-2}$

$\Delta f_3 = f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3}$

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

4 Листинг программы

Ссылка на Github репозиторий

Метод Эйлера:

```
def euler(f, y0, x0, xn, h):
    n = int((xn - x0) / h) + 1
    x = np.linspace(x0, xn, n)
    y = np.zeros(n)
    y[0] = y0

    for i in range(n - 1):
        y[i + 1] = y[i] + h * f(x[i], y[i])

    #

    y = sanitize_small_numbers(y)
    return x, y
```

Метод Рунге-Кутта:

```
def runge_kutta_4th(f, y0, x0, xn, h):
    n = int((xn - x0) / h) + 1
    x = np.linspace(x0, xn, n)
    y = np.zeros(n)

    y[0] = y0

    for i in range(n - 1):
        k1 = f(x[i], y[i])
        k2 = f(x[i] + h / 2, y[i] + h * k1 / 2)
        k3 = f(x[i] + h / 2, y[i] + h * k2 / 2)
        k4 = f(x[i] + h, y[i] + h * k3)
        y[i + 1] = y[i] + (h / 6) * (k1 + 2 * k2 + 2 * k3 + k4)

    #

    y = sanitize_small_numbers(y)
    return x, y
```

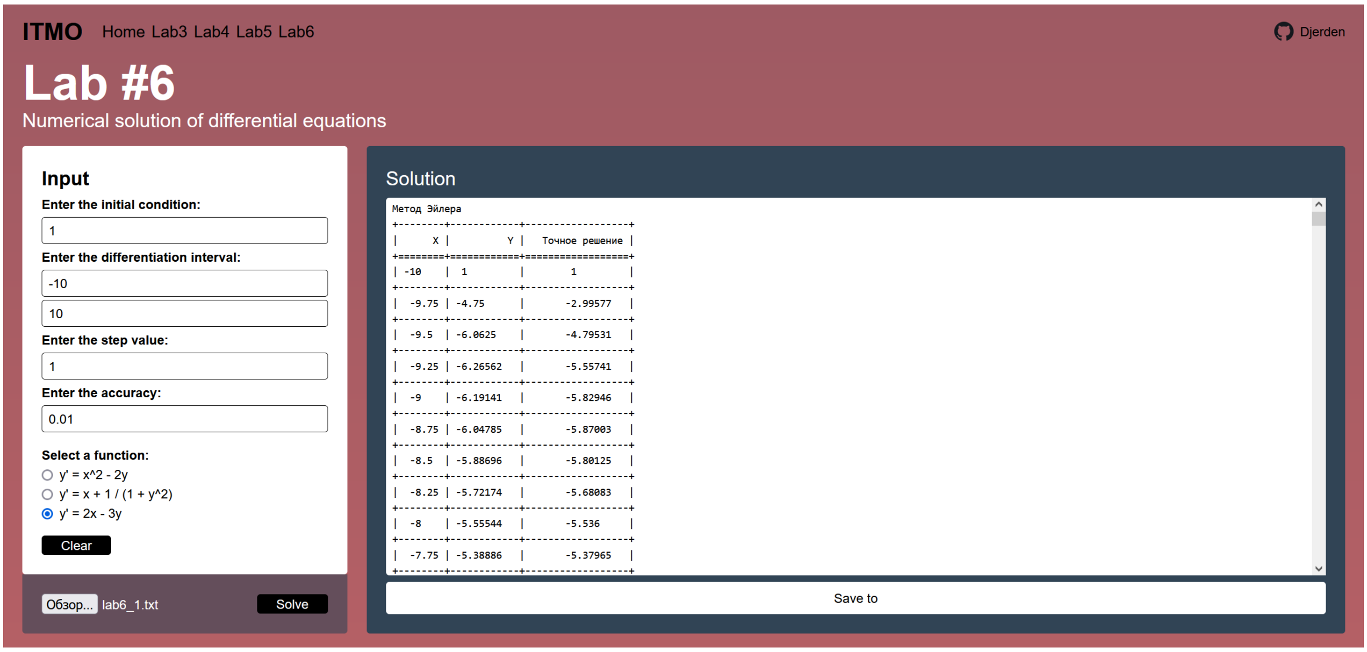
Метод Адамса:

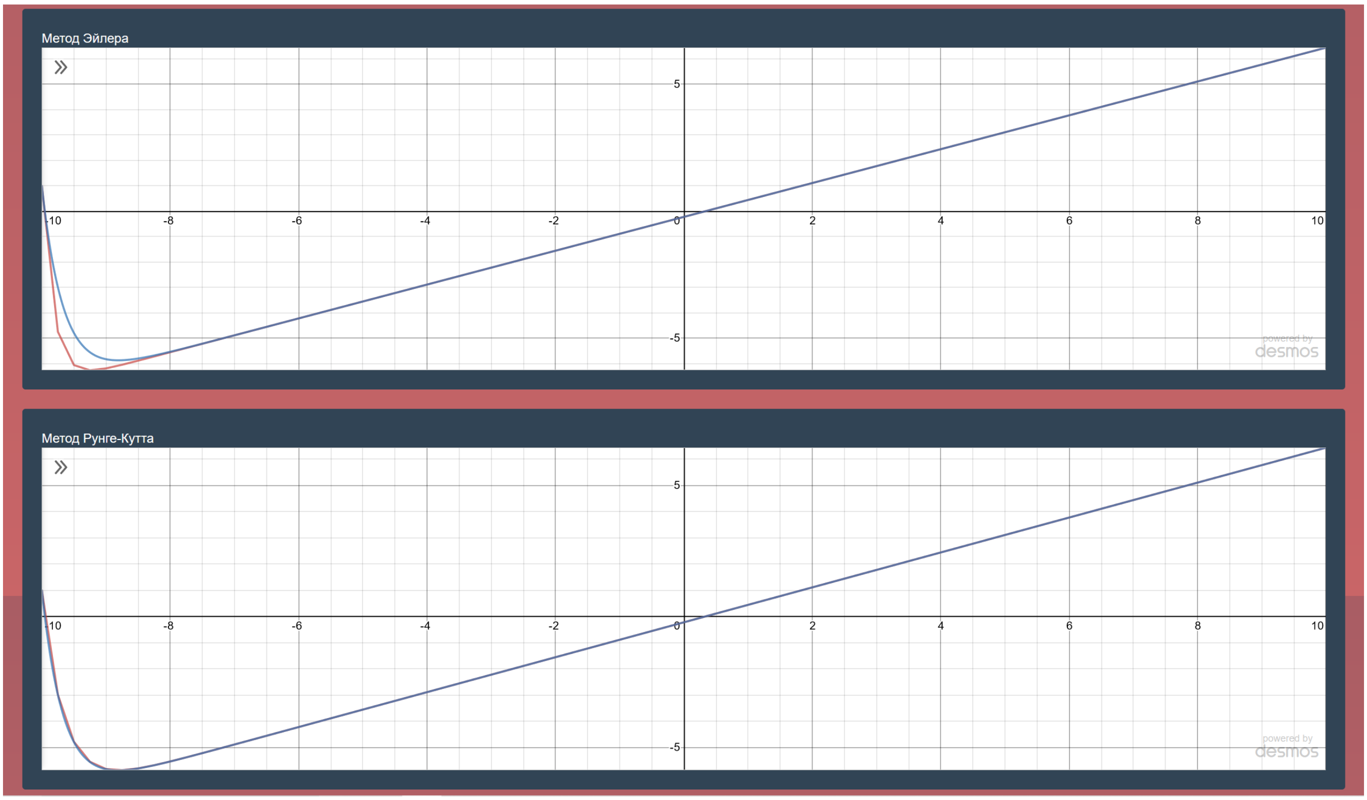
```
def adams(f, y0, x0, xn, h, eps, k=4):
    x_rk, y_rk, h_2, _ = compute_with_runge_rule(f, y0, x0, x0 + (k - 1) * h, h,
        eps, "          ␣          -          ␣4     ␣                ")
    n = int((xn - x0) / h_2) + 1
    x = np.linspace(x0, xn, n)
    y = np.zeros(n)
    for i in range(k):
        y[i] = y_rk[i]

    for i in range(k - 1, n - 1):
        y_pred = y[i] + h_2 / 24 * (
                55 * f(x[i], y[i]) - 59 * f(x[i - 1], y[i - 1]) + 37 * f(x[i -
                    2], y[i - 2]) - 9 * f(x[i - 3],

```

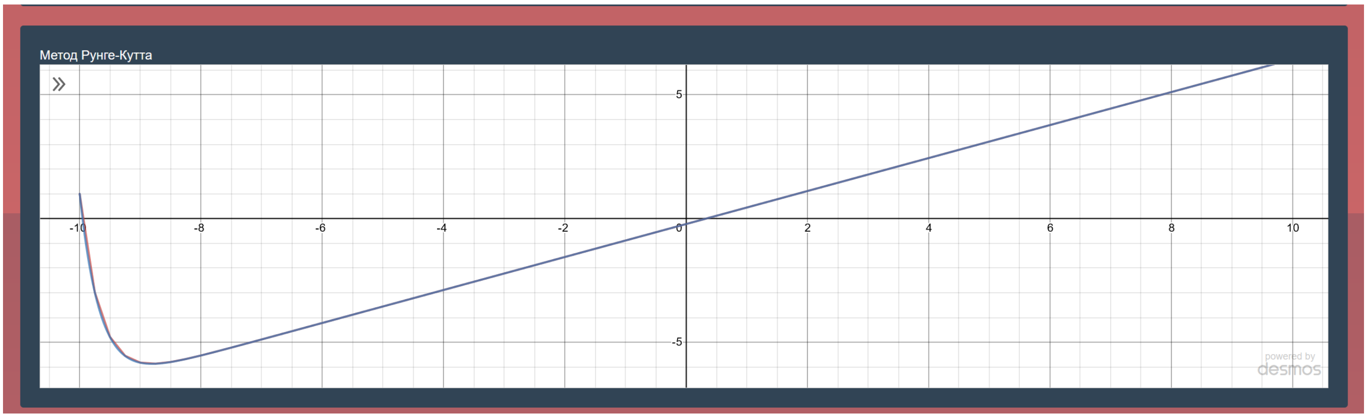
```
        f_pred = f(x[i + 1], y_pred)
        y_corr = y[i] + h_2 / 24 * (
                9 * f_pred + 19 * f(x[i], y[i]) - 5 * f(x[i - 1], y[i - 1]) + f(x[i - 2], y[i - 2]))

        while abs(y_pred - y_corr) > eps:
            y_pred = y_corr
            y_corr = y[i] + h_2 / 24 * (
                    9 * f_pred + 19 * f(x[i], y[i]) - 5 * f(x[i - 1], y[i - 1]) + f(x[i - 2], y[i - 2]))
        y[i + 1] = y_corr

    #

    y = sanitize_small_numbers(y)
    y_real_solution = exact_solution(f, x0, y0)(x)

    return x, y, y_real_solution
```

5 Результаты работы программы

5.1 Пример 1







6 Вывод

В ходе данной лабораторной работы, были изучены несколько численных методов, которые используются для решения задачи Коши.

1. Метод Эйлера прост в реализации, однако довольно груб, удовлетворительная точность достигается только при малом шаге h.

2. Метод Рунге-Кутта сложнее в реализации, требует большего объема вычислений по сравнению с методом Эйлера и его модификации, однако это окупается повышенной точностью, что дает возможность проводить счет с большим шагом.

3. Метод Адамса - многошаговый метод и по сравнению с Рунге-Кутта той же точности, метод Адамса экономичный, поскольку он требует вычисления лишь одного значения правой части на каждом шаге (в методе Рунге-Кутта - четырех). Однако метод Адамса расчет может быть начат только с узла x3, а не x0, иными словами предварительно нужно воспользоваться другим методом.