

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Поволжский государственный университет телекоммуникаций и
информатики»

Факультет информатики и вычислительной техники

Кафедра При

ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №9
Дисциплина: Численные методы

Выполнил: студент
При-21 Морзюков М.А.
Проверил(а):
Осанов В.А.

Самара 2024

Цель работы: изучить методы решения задачи Коши методом Адамса и реализовать его программными средствами.

Вариант 11

№ варианта	$F_1(x, y_1, y_2)$	$F_2(x, y_1, y_2)$	$y_1(a)$	$y_2(a)$	a	b
11	$\arctg(\frac{1}{1+y_1^2+y_2^2})$	$\sin(y_1 y_2)$	1	1	1	4

Метод Адамса — это численный метод для решения систем обыкновенных дифференциальных уравнений (ОДУ), который применяется для нахождения приближённых решений задачи Коши. Он относится к многократным методам предсказания и использует значения производной на нескольких предыдущих шагах для вычисления следующего значения функции. В отличие от методов Рунге-Кутты, где для прогноза используется информация только о текущем шаге, метод Адамса учитывает данные о нескольких предыдущих шагах.

```
public static double[] adamsMethod(Function[] f, double[] y, double x0, double h, int n) {
    double[] results = new double[2 * (n + 1)];
    results[0] = y[0];
    results[1] = y[1];

    for (int i = 1; i <= 2; i++) {
        double x = x0 + (i - 1) * h;
        double k1Y1 = h * f[0].evaluate(results[2 * (i - 1)], results[2 * (i - 1) + 1]);
        double k1Y2 = h * f[1].evaluate(results[2 * (i - 1)], results[2 * (i - 1) + 1]);
        results[2 * i] = results[2 * (i - 1)] + k1Y1;
        results[2 * i + 1] = results[2 * (i - 1) + 1] + k1Y2;
    }

    for (int i = 2; i < n; i++) {
        double x = x0 + i * h;
        double f1Prev = f[0].evaluate(results[2 * (i - 1)], results[2 * (i - 1) + 1]);
        double f1Prev2 = f[0].evaluate(results[2 * (i - 2)], results[2 * (i - 2) + 1]);
        double f2Prev = f[1].evaluate(results[2 * (i - 1)], results[2 * (i - 1) + 1]);
        double f2Prev2 = f[1].evaluate(results[2 * (i - 2)], results[2 * (i - 2) + 1]);

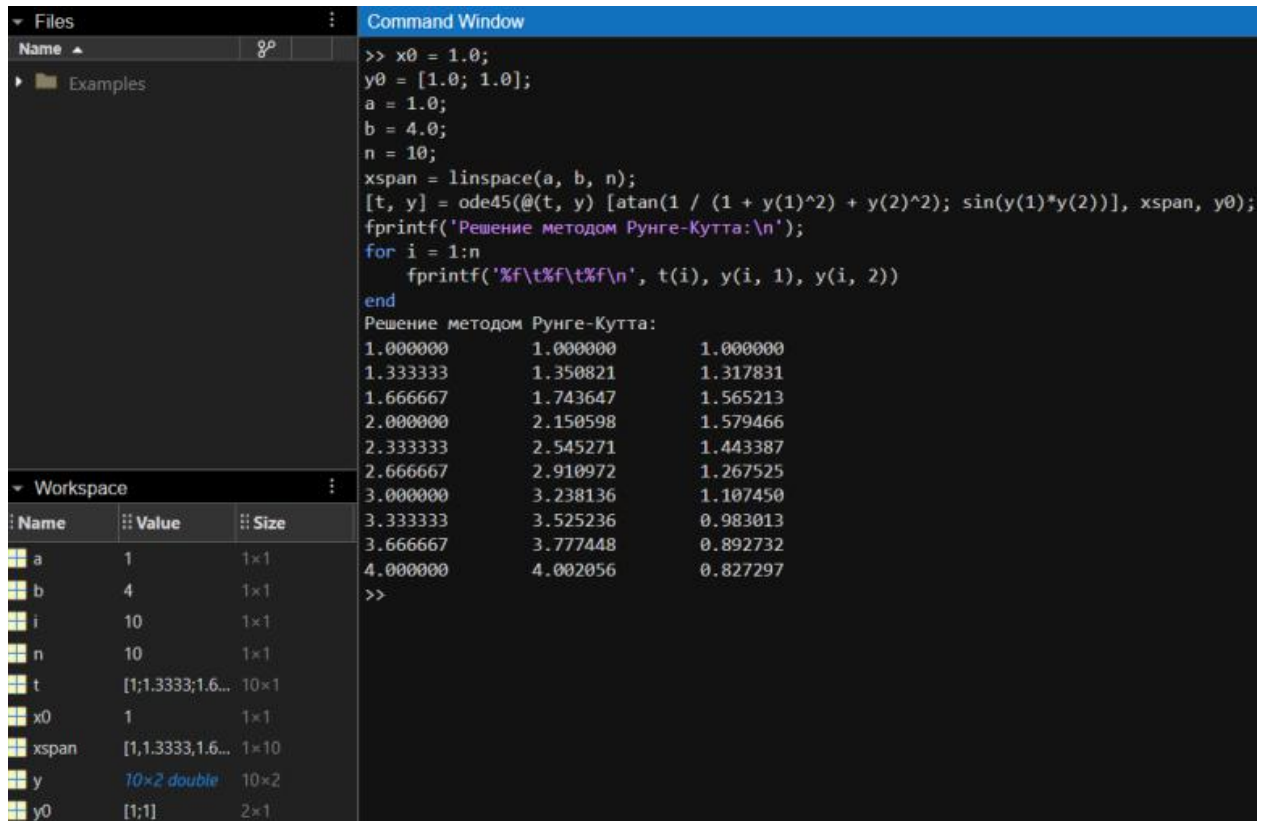
        results[2 * (i + 1)] = results[2 * i] + (h / 2) * (f1Prev + f1Prev2);
        results[2 * (i + 1) + 1] = results[2 * i + 1] + (h / 2) * (f2Prev + f2Prev2);
    }

    return results;
}
```

Результат выполнения программы:

```
Приближенное решение:
x = 1,00: y1 = 1,0000; y2 = 1,0000
x = 1,60: y1 = 1,2948; y2 = 1,2524
x = 2,20: y1 = 1,6234; y2 = 1,5521
x = 2,80: y1 = 1,9352; y2 = 1,8281
x = 3,40: y1 = 2,2816; y2 = 2,0653
x = 4,00: y1 = 2,6582; y2 = 2,0948
x = 4,60: y1 = 3,0549; y2 = 1,8869
x = 5,20: y1 = 3,4601; y2 = 1,6386
x = 5,80: y1 = 3,8586; y2 = 1,4659
x = 6,40: y1 = 4,2377; y2 = 1,3052
x = 7,00: y1 = 4,5931; y2 = 1,1309
```

Проверка в MATLAB online:



The screenshot displays the MATLAB online environment. On the left, the 'Files' pane shows a folder named 'Examples'. Below it, the 'Workspace' pane lists variables: 'a' (1, 1x1), 'b' (4, 1x1), 'i' (10, 1x1), 'n' (10, 1x1), 't' ([1;1.3333;1.6..., 10x1), 'x0' (1, 1x1), 'xspan' ([1,1.3333,1.6..., 1x10), 'y' (10x2 double, 10x2), and 'y0' ([1;1], 2x1). The main 'Command Window' on the right shows the execution of a script. The script defines parameters, uses 'ode45' to solve a differential equation, and prints the results using 'fprintf'. The output shows the solution values for t, y(1), and y(2) at various time steps.

```
>> x0 = 1.0;
y0 = [1.0; 1.0];
a = 1.0;
b = 4.0;
n = 10;
xspan = linspace(a, b, n);
[t, y] = ode45(@(t, y) [atan(1 / (1 + y(1)^2) + y(2)^2); sin(y(1)*y(2))], xspan, y0);
fprintf('Решение методом Рунге-Кутты:\n');
for i = 1:n
    fprintf('%f\t%f\t%f\n', t(i), y(i, 1), y(i, 2))
end
Решение методом Рунге-Кутты:
1.000000      1.000000      1.000000
1.333333      1.350821      1.317831
1.666667      1.743647      1.565213
2.000000      2.150598      1.579466
2.333333      2.545271      1.443387
2.666667      2.910972      1.267525
3.000000      3.238136      1.107450
3.333333      3.525236      0.983013
3.666667      3.777448      0.892732
4.000000      4.002056      0.827297
>>
```