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

№ вари-анта	$F_1(x,y_1,y_2)$	$F_2(x,y_1,y_2)$	$y_1(a)$	$y_2(a)$	a	b
11	$arctg(\, {}^{1}\!/_{1+y_1^2} + y_2^2 \,)$	$sin(y_1y_2)$	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.

```java
double[][] ys = new double[n + 1][2];
ys[0][0] = y1;
ys[0][1] = y2;

for (int i = 1; i <= 3; i++) {
    double k11 = h * f1(y1, y2);
    double k21 = h * f2(y1, y2);

    double k12 = h * f1(y1 + 0.5 * k11, y2 + 0.5 * k21);
    double k22 = h * f2(y1 + 0.5 * k11, y2 + 0.5 * k21);

    double k13 = h * f1(y1 + 0.5 * k12, y2 + 0.5 * k22);
    double k23 = h * f2(y1 + 0.5 * k12, y2 + 0.5 * k22);

    double k14 = h * f1(y1 + k13, y2 + k23);
    double k24 = h * f2(y1 + k13, y2 + k23);

    y1 += (k11 + 2 * k12 + 2 * k13 + k14) / 6;
    y2 += (k21 + 2 * k22 + 2 * k23 + k24) / 6;

    ys[i][0] = y1;
    ys[i][1] = y2;
}

for (int i = 3; i < n; i++) {
    double f1n = f1(ys[i][0], ys[i][1]);
    double f1n1 = f1(ys[i - 1][0], ys[i - 1][1]);
    double f1n2 = f1(ys[i - 2][0], ys[i - 2][1]);

    double f2n = f2(ys[i][0], ys[i][1]);
    double f2n1 = f2(ys[i - 1][0], ys[i - 1][1]);
    double f2n2 = f2(ys[i - 2][0], ys[i - 2][1]);

    y1 = ys[i][0] + h * (23 * f1n - 16 * f1n1 + 5 * f1n2) / 12;
    y2 = ys[i][1] + h * (23 * f2n - 16 * f2n1 + 5 * f2n2) / 12;
```

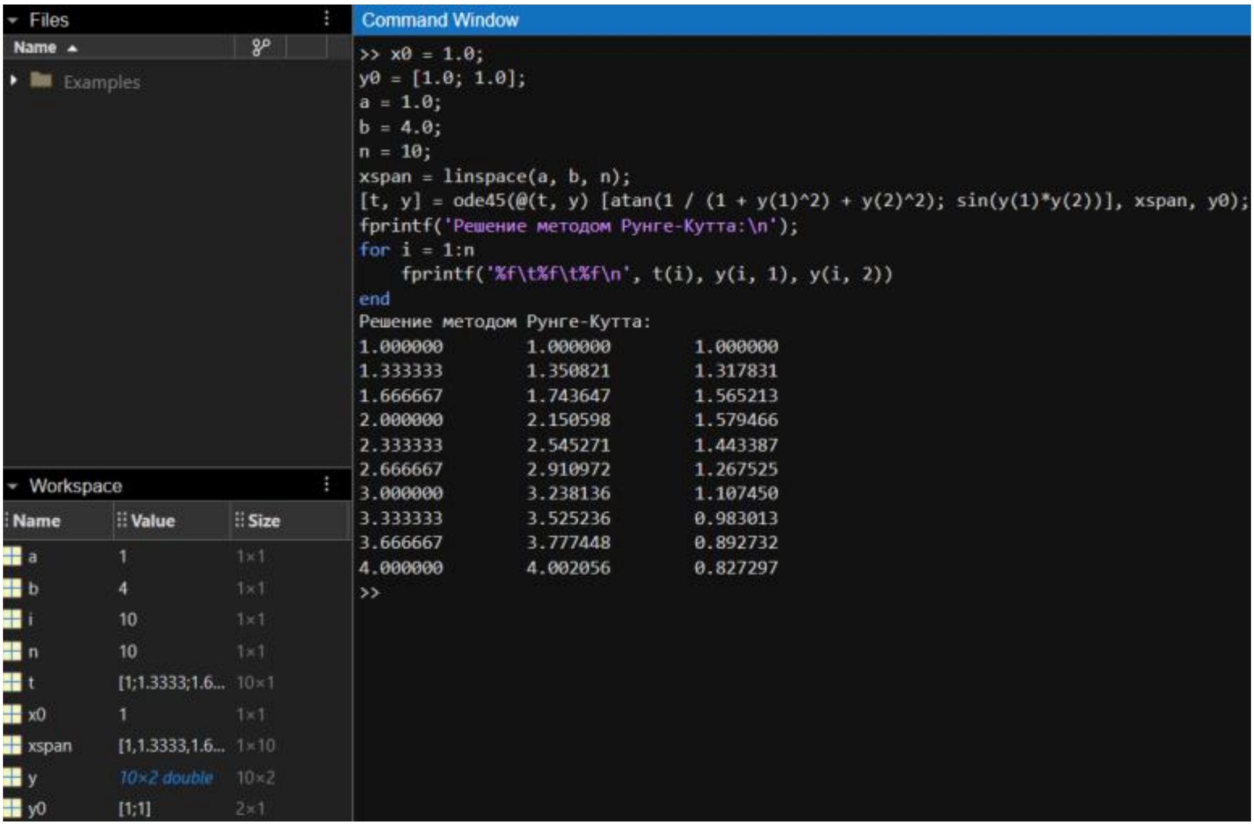
```java
        ys[i + 1][0] = y1;
        ys[i + 1][1] = y2;
    }

    for (int i = 0; i <= n; i++) {
        System.out.printf("Шаг %d: y1 = %.6f, y2 = %.6f%n", i, ys[i][0], ys[i][1]);
    }
}
```

Результат выполнения программы:

```
Шаг 0: y1 = 1,000000, y2 = 1,000000
Шаг 1: y1 = 1,313580, y2 = 1,284757
Шаг 2: y1 = 1,662890, y2 = 1,532998
Шаг 3: y1 = 2,028762, y2 = 1,596937
Шаг 4: y1 = 2,387711, y2 = 1,441367
Шаг 5: y1 = 2,710313, y2 = 1,380471
Шаг 6: y1 = 3,043037, y2 = 1,161783
Шаг 7: y1 = 3,297278, y2 = 1,130071
Шаг 8: y1 = 3,588750, y2 = 0,895657
Шаг 9: y1 = 3,747711, y2 = 1,026671
Шаг 10: y1 = 4,061751, y2 = 0,613654
```

Проверка в MATLAB online:



```
>> 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-Кутта:\n');
for i = 1:n
    fprintf('%f\t%f\t%f\n', t(i), y(i, 1), y(i, 2))
end
Решение методом Рунге-Кутта:
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
>>
```


Name	Value	Size
a	1	1×1
b	4	1×1
i	10	1×1
n	10	1×1
t	[1;1.3333;1.6...	10×1
x0	1	1×1
xspan	[1,1.3333,1.6...	1×10
y	10×2 double	10×2
y0	[1;1]	2×1