*Дубкова Валерия КС-26*

**12-ая лабораторная работа**

**Код**

clear all; clc

% Определение символьной функции и её производной

syms f1(x)

f1(x) = 9 \* exp(x) + exp(2\*x);

df = diff(f1, x);

anAns = 'y''-y=exp(2\*x) => y''=2\*exp(2\*x) + 9\*exp(x)';

fprintf('%s\n', anAns);

fprintf('Т.к. предложенная функция и программный ответ совпали, то данная функция является решением диф уравнения\n');

% Определение функции f

f = @(x, y) x^3 + x + 3 \* (y / x);

% 2 ЗАДАНИЕ

% Метод Эйлера h=0.2

a = 1; b = 2;

x0 = 1; y0 = 3;

h = [0.2, 0.05];

% Метод Эйлера h=0.2

xE1 = x0;

yE1 = y0;

k = 1;

for i = a:h(1):b-h(1)

dyE1(k) = f(xE1(k), yE1(k));

yE1(k+1) = yE1(k) + h(1) \* dyE1(k);

xE1(k+1) = xE1(k) + h(1);

k = k + 1;

end

% Метод Эйлера h=0.05

xE2 = x0;

yE2 = y0;

k = 1;

for i = a:h(2):b-h(2)

dyE2(k) = f(xE2(k), yE2(k));

yE2(k+1) = yE2(k) + h(2) \* dyE2(k);

xE2(k+1) = xE2(k) + h(2);

k = k + 1;

end

% Метод Рунге-Кутта 4-го порядка с шагом 0.2

xRK1 = x0;

yRK1 = y0;

k = 1;

for i = a:h(1):b-h(1)

K1 = h(1) \* f(xRK1(k), yRK1(k));

K2 = h(1) \* f(xRK1(k) + h(1)/2, yRK1(k) + K1/2);

K3 = h(1) \* f(xRK1(k) + h(1)/2, yRK1(k) + K2/2);

K4 = h(1) \* f(xRK1(k) + h(1), yRK1(k) + K3);

yRK1(k+1) = yRK1(k) + (1/6) \* (K1 + 2\*K2 + 2\*K3 + K4);

xRK1(k+1) = xRK1(k) + h(1);

k = k + 1;

end

% Метод Рунге-Кутта 4-го порядка с шагом 0.05

xRK2 = x0;

yRK2 = y0;

k = 1;

for i = a:h(2):b-h(2)

K1 = h(2) \* f(xRK2(k), yRK2(k));

K2 = h(2) \* f(xRK2(k) + h(2)/2, yRK2(k) + K1/2);

K3 = h(2) \* f(xRK2(k) + h(2)/2, yRK2(k) + K2/2);

K4 = h(2) \* f(xRK2(k) + h(2), yRK2(k) + K3);

yRK2(k+1) = yRK2(k) + (1/6) \* (K1 + 2\*K2 + 2\*K3 + K4);

xRK2(k+1) = xRK2(k) + h(2);

k = k + 1;

end

% Решение с использованием ode113

[xp, yp] = ode113(f, [a, b], y0);

% Построение графиков

figure;

p1 = plot(xE1, yE1, '-\*r');

hold on;

p2 = plot(xE2, yE2, '-\*b');

p3 = plot(xRK1, yRK1, '-\*g');

p4 = plot(xRK2, yRK2, '-\*k');

p5 = plot(xp, yp, '-om');

grid on;

title('Графики');

xlabel('x');

ylabel('y');

legend([p1, p2, p3, p4, p5], {'Эйлера, h=0.2', 'Эйлера, h=0.05', 'Рунге-Кутта 4-го пор., h=0.2', 'Рунге-Кутта 4-го пор., h=0.05', 'ode113'});

legend('Location', 'northwest');

legend('boxoff');

% Вычисление погрешностей

fprintf('\nПогрешность в вычислениях метода Эйлера в сравнении с методом Рунге-Кутта 4-го порядка, h=0.2\nk:\tabs(yRK-yE):\n');

for i = 1:length(yRK1)

fprintf('%d\t%f\n', i, abs(yRK1(i) - yE1(i)));

end

fprintf('\nПогрешность в вычислениях метода Эйлера в сравнении с методом Рунге-Кутта 4-го порядка, h=0.05\nk:\tabs(yRK-yE):\n');

for i = 1:length(yRK2)

fprintf('%d\t%f\n', i, abs(yRK2(i) - yE2(i)));

end

**Результат выполнения программы**

*y'-y=exp(2\*x) => y'=2\*exp(2\*x) + 9\*exp(x)*

*Т.к. предложенная функция и программный ответ совпали, то данная функция является решением диф уравнения*

*Погрешность в вычислениях метода Эйлера в сравнении с методом Рунге-Кутта 4-го порядка, h=0.2*

*k: abs(yRK-yE):*

*1 0.000000*

*2 0.615572*

*3 1.723152*

*4 3.464788*

*5 5.989723*

*6 9.453597*

*Погрешность в вычислениях метода Эйлера в сравнении с методом Рунге-Кутта 4-го порядка, h=0.05*

*k: abs(yRK-yE):*

*1 0.000000*

*2 0.035879*

*3 0.079571*

*4 0.131750*

*5 0.193102*

*6 0.264318*

*7 0.346104*

*8 0.439168*

*9 0.544231*

*10 0.662022*

*11 0.793274*

*12 0.938731*

*13 1.099143*

*14 1.275267*

*15 1.467868*

*16 1.677716*

*17 1.905587*

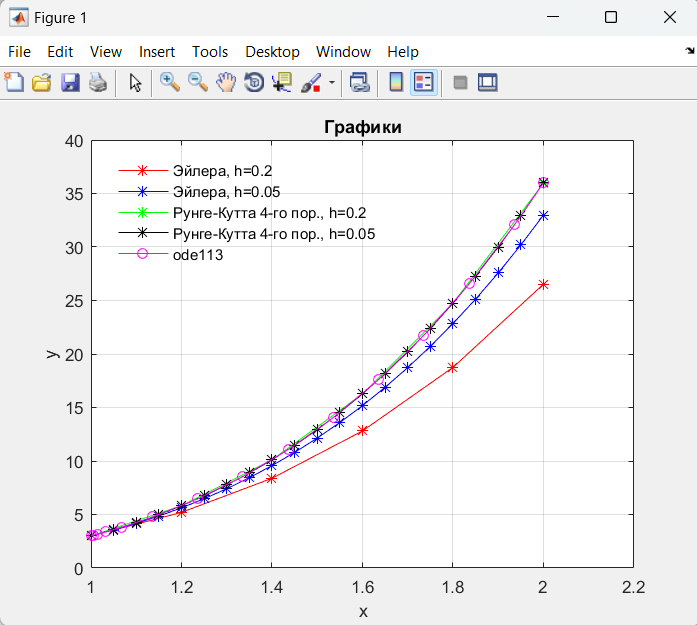
*18 2.152266*

*19 2.418542*

*20 2.705209*

*21 3.013070*

**Графики**



**Выводы**

Как можно увидеть по графику, и ответу – у метода Эйлера есть довольно большая погрешность, что не позволяет его использовать в тех случаях, когда вам нужны довольно точные значения, чего не скажешь о методе Рунге-Кутта. По графику видно, что с шагом h=0.05 метод Рунге-Кутта почти идентичен со встроенными функциями mallab для решения дифференциальных уравнений. Так же, стоит отметить, что чем меньше шаг, тем точнее получается ответ.