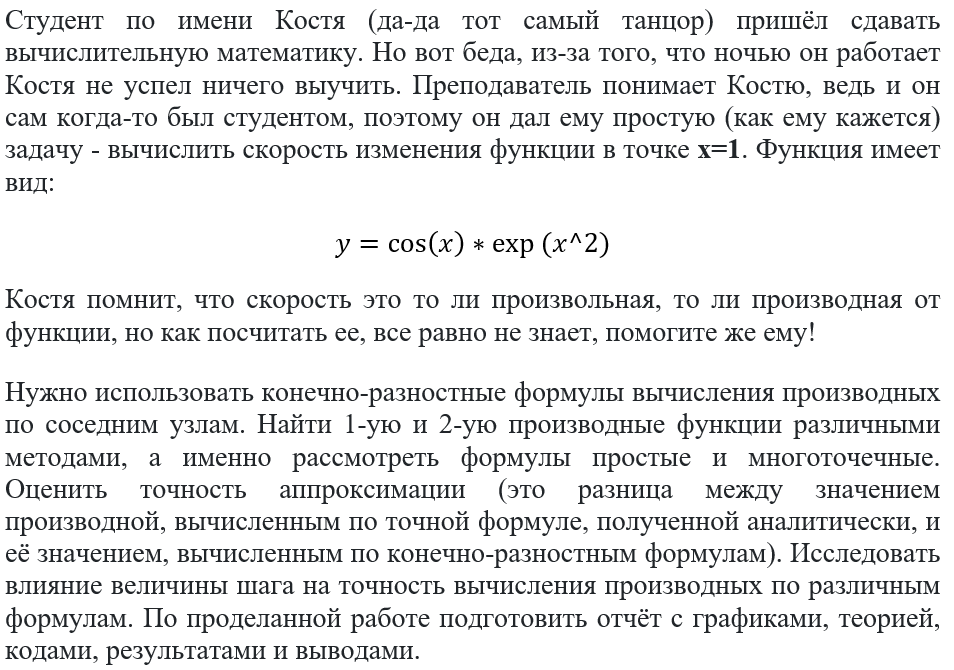
*Дубкова Валерия КС-26*

**9-ая лабораторная работа**

**Задание**

****

**Код lab9\_1pr.m**

clc;

%полином из лабы 7

x = [3.37; 8.08; 11.76; 12.15; 13.25; 14.53];

y = [1.4272; 17.6441; 63.6593; 71.6087; 98.4015; 139.1986];

p = [0.0001 0.0032 -0.0196 -0.0106 0.0100 7.7443];

kP1 = polyder(p);

kP2 = polyder(kP1);

zP1 = polyval(kP1, x);

zP2 = polyval(kP2, x);

hold on;

plot(x, y, '-+green');

grid on;

plot(x, zP1, '-or');

plot(x, zP2, '-oblack');

legend('Func', 'I произ', 'II произ');

figure;

%Поиск 1-ой производной

%определим шаг сетки

h = 0.2;

%определим сетку

x = 0:h:pi;

n=length(x);

syms f(m);

f(m) = cos(m)\*exp(m^2);

% y = cos(x)\*exp(x^2)

%точное значение производной

dy = 2.\*x.\*exp(x.^2).\*cos(x)-exp(x.^2).\*sin(x);

%определим производную по формуле (6) правой конечной разности и

%соответствующую абсолютную погрешность

for i=1:(n-1)

dy1(i) = (f(x(i+1))-f(x(i)))/h;

err1(i) = abs(dy(i)-dy1(i));

end

%определим производную по формуле (5)

%левой конечной разности и соответствующую

%абсолютную погрешность

for i=2:n

dy2(i)=(f(x(i))-f(x(i-1)))/h;

err2(i)=abs(dy(i)-dy2(i));

end

%определим производную по формуле (7)

%центральной разности и соответствующую

%абсолютную погрешность

for i=2:(n-1)

dy3(i) = (f(x(i+1))-f(x(i-1)))/(2\*h);

err3(i) = abs(dy(i)-dy3(i));

end

%определим производную по формуле (9) четвертого порядка точности и соответствующую

%абсолютную погрешность, которая вычисляется в точке x(i)-0.5\*h

for i=3:(n-1)

dy4(i)=(f(x(i+1))+27\*f(x(i))-27\*f(x(i-1))...

+f(x(i-2)))/(24\*h);

err4(i)=abs(cot(x(i)-0.5\*h)-dy4(i));

end

%рисуем все три абсолютные ошибки на одном графике

plot(x([1:(n-1)]),err1([1:(n-1)]),'-or',...

x([2:n]),err2([2:n]),'-pb',...

x([2:(n-1)]),err3([2:(n-1)]),'-hy',...

x([3:(n-1)]),err4([3:(n-1)]),'-\*g');

title('Погрешности производной 1-й степени');

legend('Правая', ' Левая', 'Центральная', '4-ый порядок точности');

grid on;

**lab9\_2pr.m**

%Поиск 2-ой производной

% шаг сетки

h=0.2;

x=0:h:pi;

n=length(x);

syms f(m);

f(m) = cos(m)\*exp(m^2);

%анализируемая функция cos(x)\*exp(x^2)

%второй производной 2\*x\*exp(x^2)\*cos(x) - exp(x^2)\*sin(x)

d2y= 2.\*x.\*exp(x.^2).\*cos(x) - exp(x.^2).\*sin(x);

%определим вторую производную по формуле (8)

for i=2:(n-1)

d2y1(i)=(f(x(i+1))-f(x(i))+f(x(i-1)))/h^2;

err1(i)=abs(d2y(i)-d2y1(i));

end

%определим вторую производную по формуле (10)

for i=3:(n-2)

d2y2(i)=(f(x(i+2))+16\*f(x(i+1))-30\*f(x(i))+...

16\*f(x(i-1))-f(x(i-2)))/(12\*h^2);

err2(i)=abs(d2y(i)-d2y2(i));

end

%рисуем абсолютные ошибки на одном графике

plot(x([2:(n-1)]),err1([2:(n-1)]),'-\*r',...

x([3:(n-2)]),err2([3:(n-2)]),'-pb');

title('Погрешности производной 2-й степени');

legend('II порядок точности', 'IV порядок точности');

grid on;

**lab9\_h.m**

%Исследование влияния величины шага

%на точность вычисления производных 2 порядка по различным формулам

clc;

clear all;

% шаг сетки

h(1)=1.0;

% Количество "кругов"

n = 6;

% n = 10;

%формируем массив шагов

for i=2:n

h(i)=h(i-1)/2;

end

syms f(m);

%выбираем фиксированную точку для функции численной оценки второй

%производной функции y = cos(x)\*exp(x^2)

x = 0.96;

f(m) = cos(m)\*exp(m^2);

D2y = 2.\*x.\*exp(x.^2).\*cos(x) - exp(x.^2).\*sin(x);

% находим численное значение второй производной при различных значениях

% h и сравниваем их с эталоном

for i=1:n

D2y2 = (f(x+h(i))-2\*f(x)+f(x-h(i)))/h(i)^2;

err1(i) = abs(D2y2-D2y);

end

%строим график зависимости ошибки аппроксимации в зависимости

%от величины шага сетки h

for i=1:n

D2y3 = (f(x+2\*h(i))+16\*f(x+h(i))-30\*f(x)+...

16\*f(x-h(i))-f(x-2\*h(i)))/(12\*h(i)^2);

err2(i) = abs(D2y3-D2y);

end

%строим график зависимости ошибки аппроксимации

%в зависимости от величины шага сетки h

figure;

plot(h, err1, '-\*');

grid on;

title ('Формула II порядка точности');

xlabel('h'); ylabel('err');

figure;

hold off

plot(h, err2, '-\*')

grid on;

title ('Формула IV порядка точности');

xlabel('h'); ylabel('err');

**Результат выполнения программы**

**lab9\_1pr.m**

kP1 =

0.0005 0.0128 -0.0588 -0.0212 0.0100

kP2 =

0.0020 0.0384 -0.1176 -0.0212

zP1 =

-0.1748

4.8832

22.0096

24.9267

34.5925

48.8392

zP2 =

0.0951

2.5906

7.1592

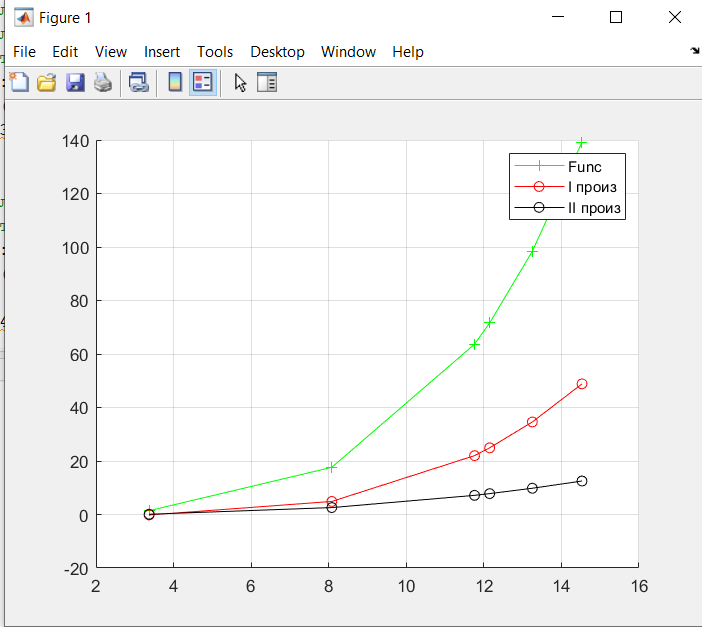
7.8059

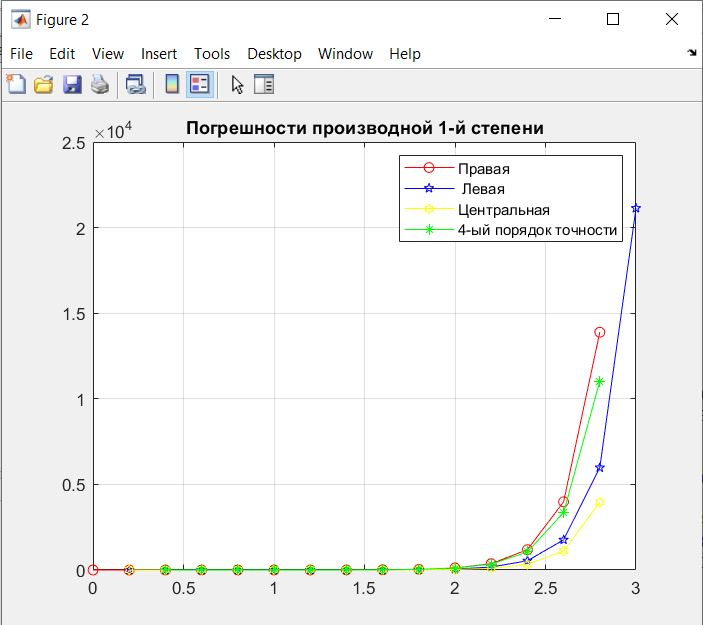
9.8146

12.5123

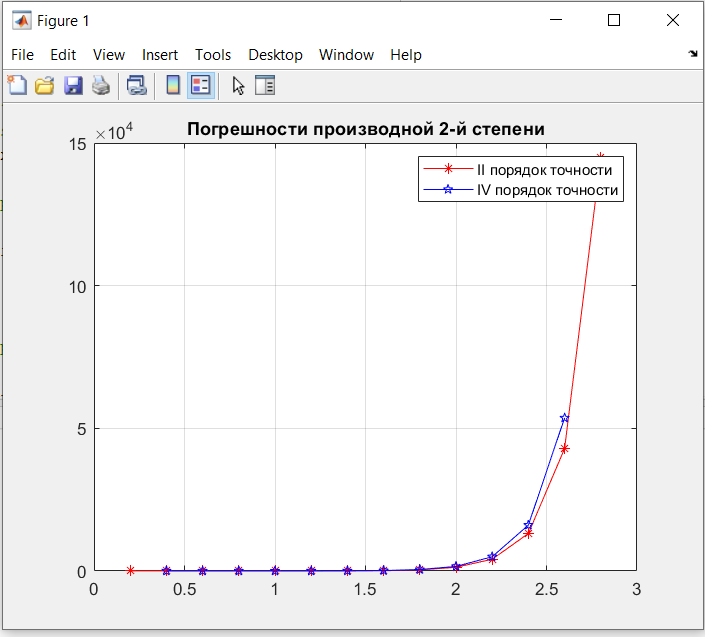
**Графики**

**lab9\_1pr.m**

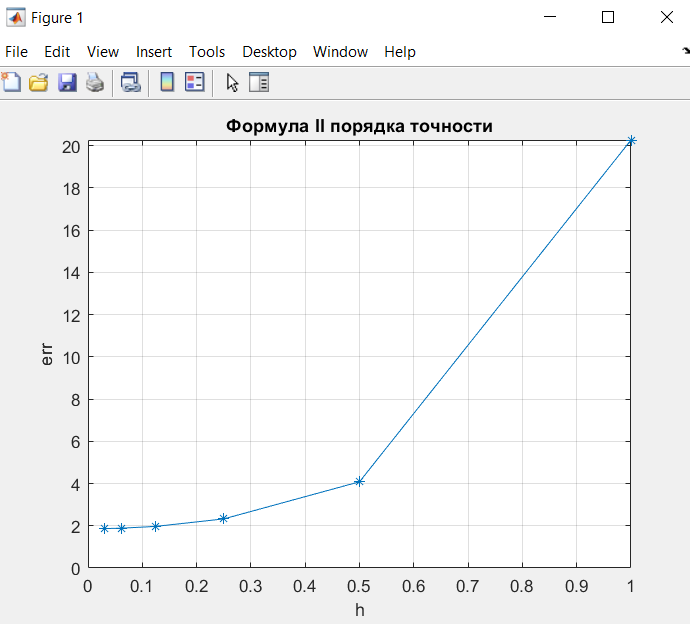


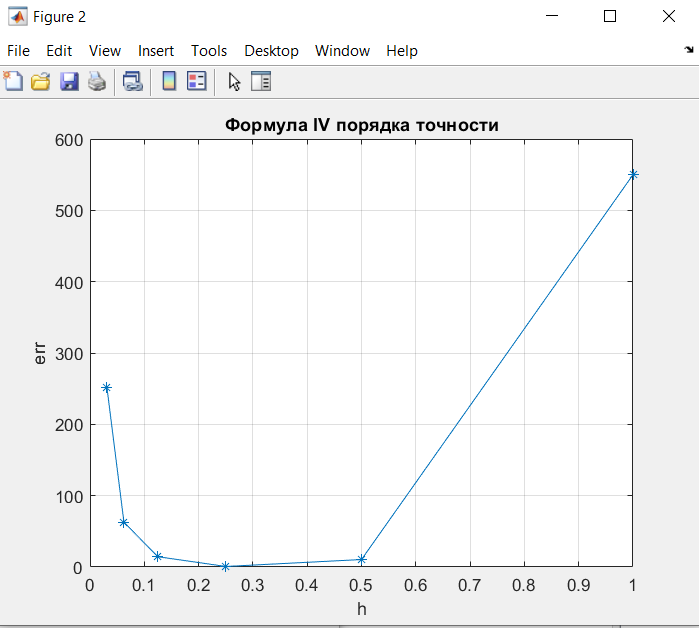


**lab9\_2pr.m**



**lab9\_h.m**





**Выводы**

Как можно заметить по графикам, первая производная сначала вычислялась довольно точно, но потом погрешность начала резко возрастать. Но при этом возрастать начала для всех методов, так, самым точным из представленных методов оказался метод средних прямоугольников.

Аналогичная ситуация со 2 и 4 производными (возможно, это связано с особенностями конкретной функции, то есть представленные методы численного дифференцирования для данной функции не подходят при x>2).

В третьем файле вычислялось влияние величины шага на погрешность при меньшем и большем значениях, нежели представленных на графике, погрешность увеличивается очень сильно, но, если посмотреть на данные точки, то можно найти удачную величину шага, при которой погрешность минимальна.