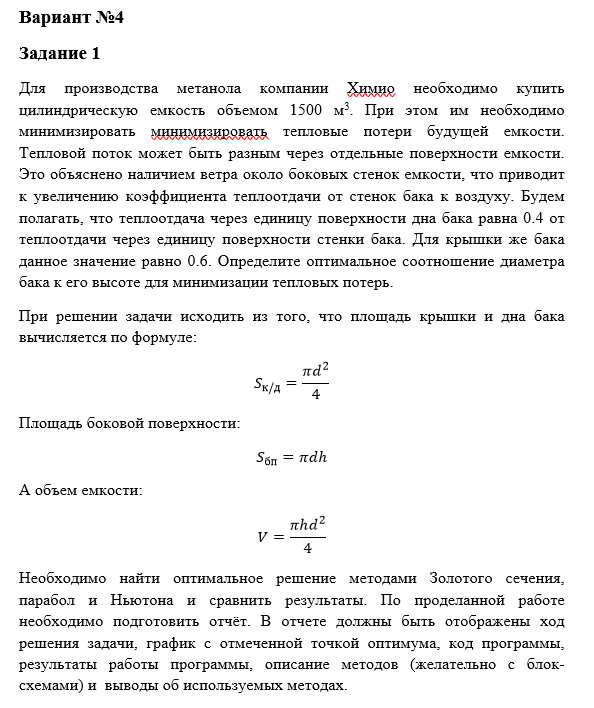
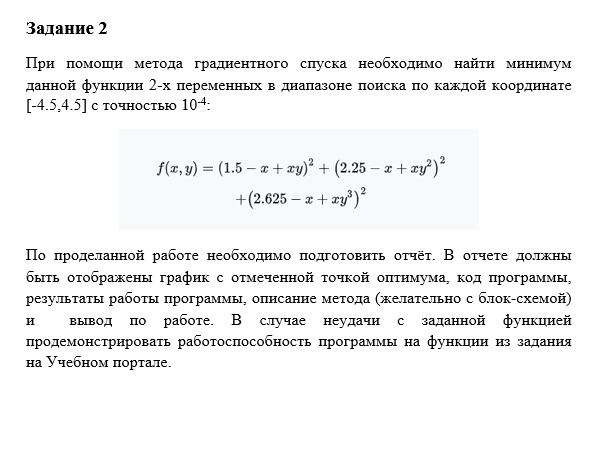
*Дубкова Валерия КС-26*

**11-ая лабораторная работа**

**Задание**





**Код Lab11z1.m**

clear all; clc

f=@(x)0.25\*pi\*x.^2+(6000/x);

df=@(x)-6000/x^2 + pi\*x/2;

d2f=@(x)pi/2 + 12000/x^3;

figure;

ezplot('0.25\*pi\*x^2+(6000/x)', [0, 30]);

xlabel('x'); ylabel('y');

grid on;

% исходя из графика делаем вывод, что точка минимума находится между 10 и 20

a = 10; b = 20;

% ЗОЛОТОЕ СЕЧЕНИЕ

%определяем точность нахождения точки минимума

eps=1e-3;

%определяем константу ksi и z

ksi=(sqrt(5)-1)/2;

z = 1 - ksi;

%определяем левую и правую границы отрезка

xl=a; xr=b;

%определяем номер итерации

k=1;

%организуем цикл метода поиска точки минимума методом ЗОЛОТОГО СЕЧЕНИЯ

x1 = xl + z\*(xr - xl);

x2 = xr - z\*(xr - xl);

while xr-xl > eps

F1 = f(x1); F2 = f(x2);

if (F1 <= F2)

xBst(k) = x1;

xr=x2; x2=x1;

x1 = xl + z\*(xr - xl);

else

xBst(k) = x2;

xl = x1;

x1 = x2;

x2 = xr - z\*(xr - xl);

end

hold on

k = k + 1;

end

% График номера итерации от оптимального значения

figure

plot(1:k-1, xBst, '-\*r');

grid on

xlabel('k'); ylabel('x');

title('Золотое сечение');

fprintf('Ответ с помощью метода ЗОЛОТОГО СЕЧЕНИЯ: %f\n\tкол-во итераций: %d',...

xBst(k-1), k-2);

% МЕТОД ПАРАБОЛ

%определяем параметр h метода парабол

h=1e-4;

%определяем число итераций в методе парабол

iterm=10;

%задаем набор начальных значений

x0=a:1:b;

%организуем цикл расчетов по методу парабол стартуя с каждого начального значения

figure;

for i=1:length(x0)

xs=x0(i);

iter=1; x(1)=xs;

%организуем цикл итераций метода парабол

while iter<iterm

xs=xs-0.5\*h\*((f(xs+h)-f(xs-h))/...

(f(xs+h)-2\*f(xs)+f(xs-h)));

iter=iter+1; x(iter)=xs;

end

%наносим очередной график зависимости значений итераций от номера итераций

plot(1:iter, x);

hold on

end

xlabel('k'); ylabel('x');

title('Парабол');

grid on

fprintf('\nОтвет с помощью метода ПАРАБОЛ: %f\n\tкол-во итераций: %d',...

x(iter), iter);

% НЬЮТОНА

eps = 1e-4;

k = 1;

xN(k) = a;

while 1

newX = xN(k) - (df(xN(k))/d2f(xN(k)));

k = k + 1;

xN(k) = newX;

if abs(df(xN(k))) <= eps

break

end

end

figure

plot(1:k, xN, '-\*b');

grid on

xlabel('k'); ylabel('x');

title('Ньютона');

fprintf('\nОтвет с помощью метода НЬЮТОНА: %f\n\tкол-во итераций: %d',...

xN(k), k);

fprintf('\nОтвет с помощью функции fminbnd: %f', fminbnd(f, a, b));

**Lab11z2.m**

clear all; clc

f=@(x)(1.5-x(1)+x(1).\*x(2)).^2+(2.25-x(1)+x(1).\*x(2).^2).^2+(2.625-x(1)+x(1).\*x(2).^3).^2;

ax = -4.5; bx = 4.5;

ay = -4.5; by = 4.5;

eps1 = 1e-4;

%задаем максимально допустимое число итераций

iterm=5000;

ku = 2;

k = 1;

x(k) = bx;

y(k) = by;

R(k) = f([x(k), y(k)]);

while 1

h = 1;

dx(k) = dfx(x(k), y(k));

dy(k) = dfy(x(k), y(k));

deltx = dx(k)/(sqrt((dx(k)^2+dy(k)^2)));

delty = dy(k)/(sqrt((dx(k)^2+dy(k)^2)));

if and(abs(deltx), abs(delty)) < eps1

break

elseif k >= iterm

break

else

while 1

newx = x(k) - deltx\*h;

newy = y(k) - delty\*h;

newR = f([newx, newy]);

if newR < R(k)

break

else

h = h/ku;

end

end

if h <= eps1

break

end

k = k + 1;

x(k) = newx;

y(k) = newy;

R(k) = newR;

end

end

figure;

plot3(x, y, R, '-.black');

grid on

title('Изменение минимума при поиске с помощью м. град. сп.');

xlabel('k'); ylabel('f(x)');

fprintf('Ответ за %d итерации с помощью метода градиентного спуска: %f; в точке [%f; %f]',...

k, R(k), x(k), y(k));

xs = -4.5:0.1:4.5;

ys = -4.5:0.1:4.5;

for i=1:length(xs)

for j=1:length(ys)

z(i, j) = f([xs(i), ys(j)]);

end

end

figure;

surf(xs, ys, z);

hold on

plot3(x, y, R, '-.r');

title('Поверхность значений');

fms = fminsearch(f, [4.5, 4.5]);

fprintf('\nОтвет с помощью fminsearch: %f; в точке [%f; %f]',...

f(fms), fms);

dx(k) = dfx(x(k), y(k));

dy(k) = dfy(x(k), y(k));

**dfx.m**

function p=dfx(x, y)

p = 2\*(y^2 - 1)\*(x\*y^2 - x + 9/4) + 2\*(y^3 - 1)\*(x\*y^3 - x + 21/8)...

+ 2\*(y - 1)\*(x\*y - x + 3/2);

End

**dfy.m**

function p=dfy(x, y)

p = 2\*x\*(x\*y - x + 3/2) + 4\*x\*y\*(x\*y^2 - x + 9/4) + ...

6\*x\*y^2\*(x\*y^3 - x + 21/8);

end

**Результат выполнения программы**

**Lab11z1.m**

*Ответ с помощью метода ЗОЛОТОГО СЕЧЕНИЯ: 15.631869*

*кол-во итераций: 19*

*Ответ с помощью метода ПАРАБОЛ: 15.631853*

*кол-во итераций: 10*

*Ответ с помощью метода НЬЮТОНА: 15.631846*

*кол-во итераций: 5*

*Ответ с помощью функции fminbnd: 15.631853*

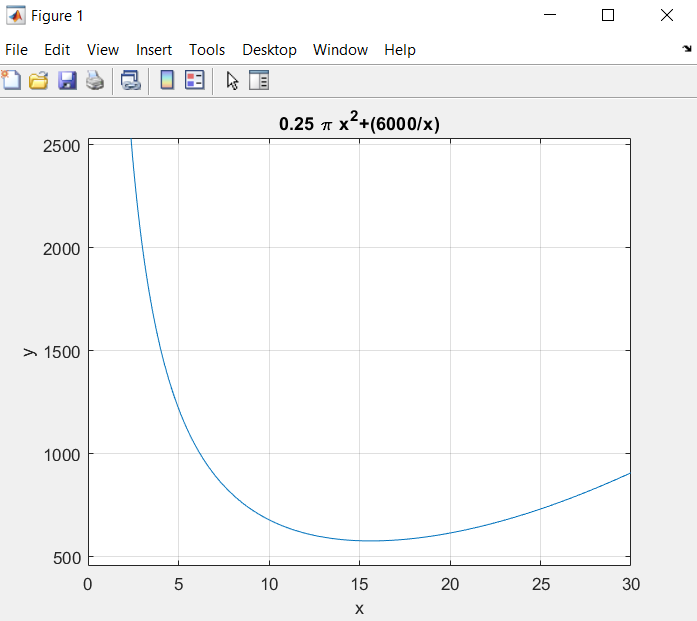
**Lab11z2.m**

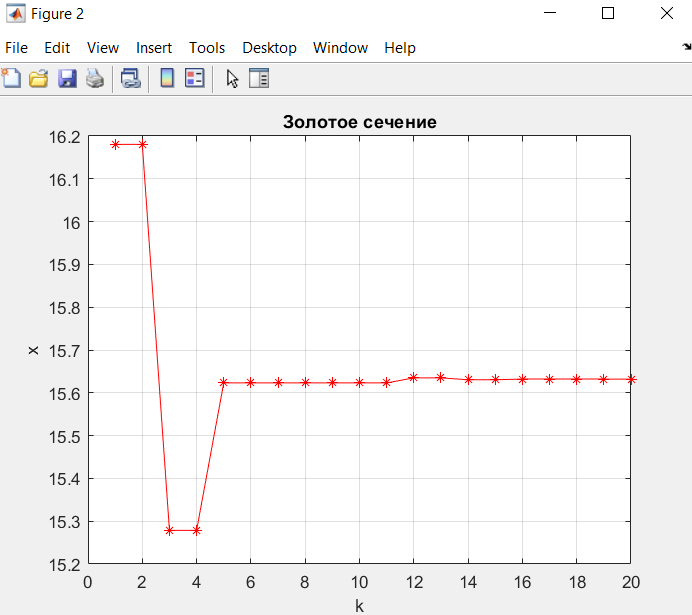
*Ответ за 769 итерации с помощью метода градиентного спуска: 0.000000; в точке [3.000006; 0.500006]*

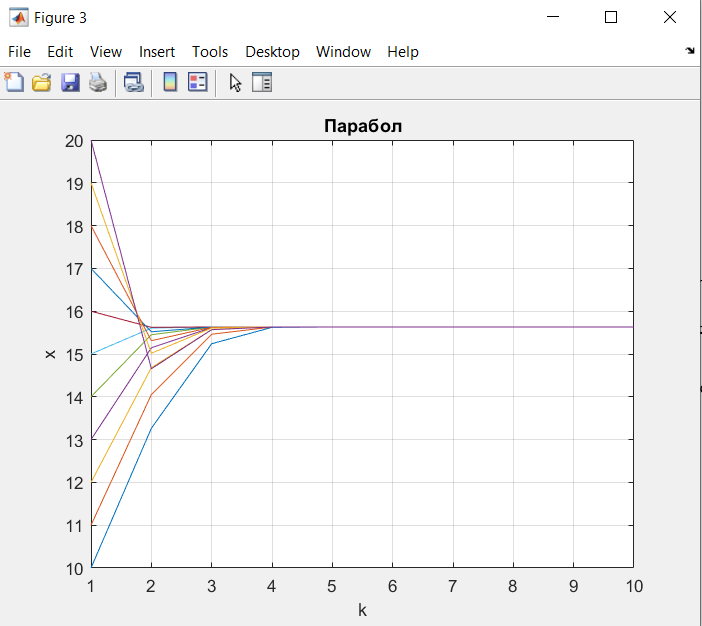
*Ответ с помощью fminsearch: 0.000000; в точке [3.000007; 0.500007]*

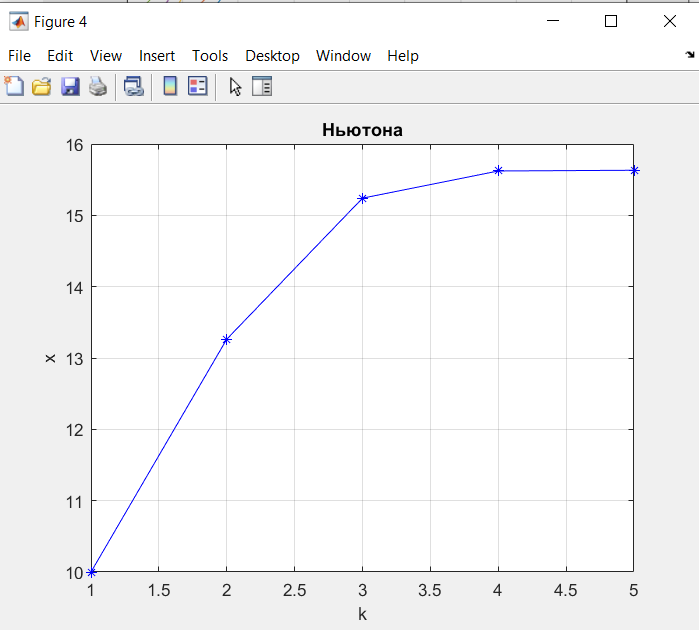
**Графики**

**Lab11z1.m**

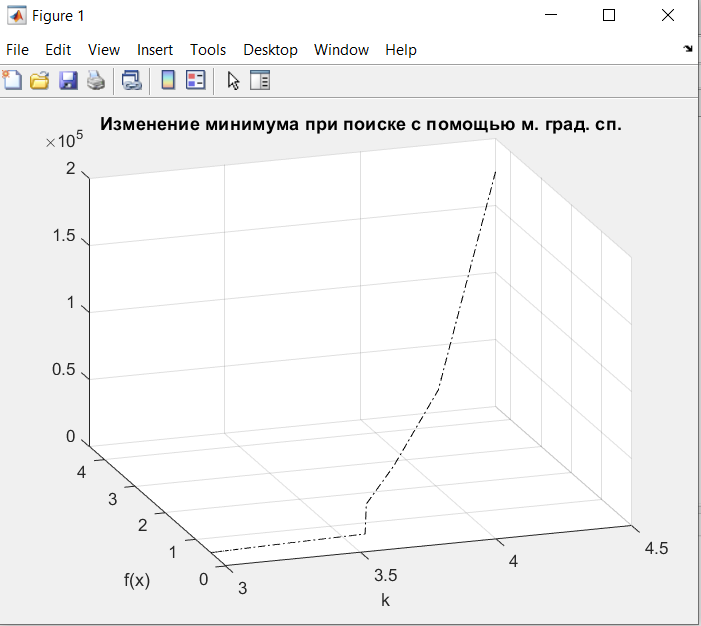


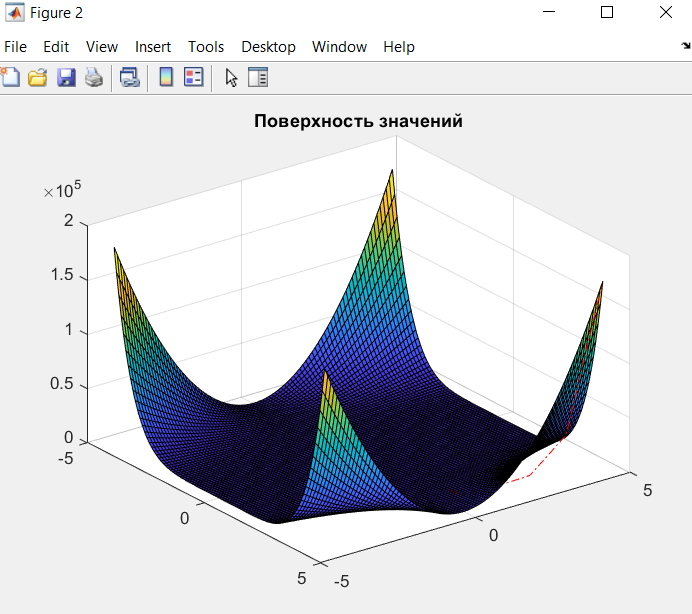






**Lab11z2.m**





**Выводы**

Каждый из представленных методов прекрасно справился со своей задачей. Можно отметить, что методу Ньютона понадобилось всего за 5 итераций, чтобы добиться заданной точности, методу золотых понадобилось 19 (в методе парабол мы сами задавали количество итераций). Поэтому, в сравнении с метода ЗС, метод Ньютона скорее всего быстрее.

Во втором задании я воспользовалась методом градиентного спуска, который так же почти идеально нашёл минимум. Погрешность, по сравнению со встроенным методом fminsearch оказалась минимальной. Стоит отметить, что немаловажную роль сыграло то, что заданная начальная точка оказалась удобной для избежания локальных минимумов.