1. ***Найти стационарную точку функции соответствующей Вашему варианту без учета ограничений. Определить чему соответствует найденная точка (максимуму, минимуму).***

Найдем стационарную точку функции:



Составим систему уравнений из частных производных функции f:





Решим эту систему:

Составим квадратичную форму в найденной стационарной точке, для этого найдем вторые частные производные:

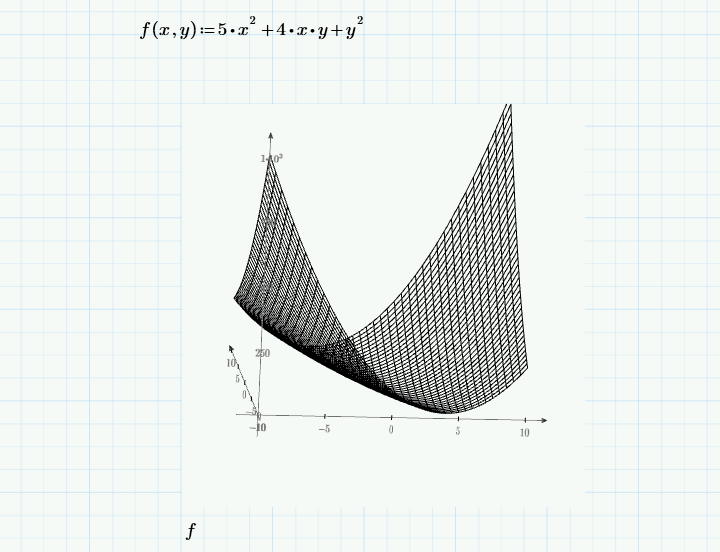
Составим матрицу Гессе и вычислим ее главные миноры:

Определитель матрицы Гессе больше нуля, а это значит, что точка (0;0) – точка минимума функции f.



График функции:



1. ***Найти точку экстремум функции с учетом ограничений.***







Составим функцию Лагранжа:



Запишем систему уравнений для определения  и стационарной точки:

Выпишем второй дифференциал функции L(x,y):



Получаем после подстановки:



Принимая во внимание начальное условие:





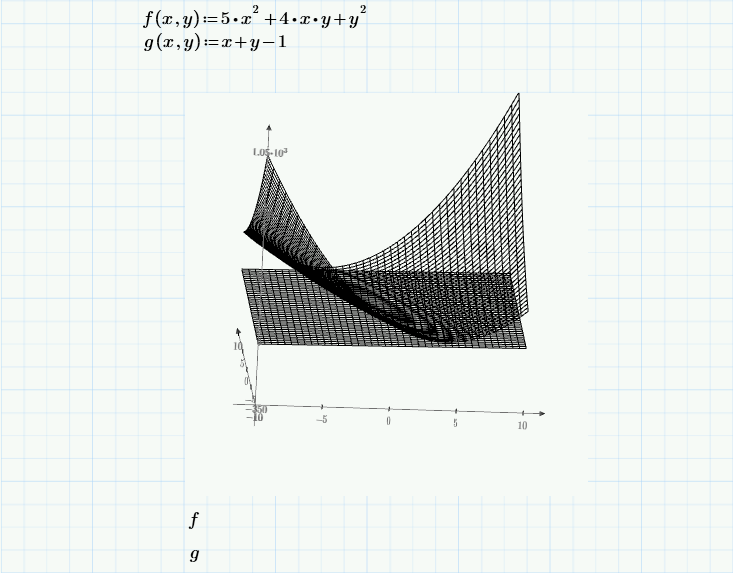


Исходя из этого, можно сказать, что функция f в точке экстремума имеет минимум.

т.М(-0,5; 1,5) – точка минимума функции f с учетом ограничения.



График функции:



1. ***Разработать в среде MatLab программу нахождения точки экстремума функции с учетом ограничений.***

*Код*

function extremum

%

err = 10^(-6);

Npr = 1000;

ng = 1;

x = 0;

y = 0;

j = 0;

while ng>err

g = [10\*x + 4\*y - 1; 4\*x + 2\*y - 1];

lamd = (10\*x\*g(1) - g(2) + 4\*g(1)\*y + 4\*g(2)\*x + 2\*y\*g(2) - g(1))/(2\*(4\*g(1)\*g(2) + 5\*g(1)^2 + g(2)^2));

x = x - lamd\*g(1);

y = y - lamd\*g(2);

ng = norm(g);

j = j + 1;

if j > Npr

break;

end

end

j

x

y

*Результат*

>> extremum

j = 122

x = -0.5000

y = 1.5000

**Вспомогательные расчеты, выполненные в среде Maple**

> f:=5\*x^2+4\*x\*y+y^2-x-y+1;

f := 5 x^2 + 4 x y + y^2 - x - y + 1

> diff(f,x);

10 x + 4 y - 1

> diff(f,y);

4 x + 2 y - 1

> F:=5\*(x-lamd\*g1)^2+4\*(x-lamd\*g1)\*(y-lamd\*g2)+(y-lamd\*g2)^2-(x-lamd\*g1)-(y-lamd\*g2)+1;

> Fλ := diff(F,lamd);

Fλ := -10\*(x - lamd\*g1)\*g1 - 4\*g1\*(y - lamd\*g2) -

- 4\*(x - lamd\*g1)\*g2 - 2\*(y - lamd\*g2)\*g2 + g1 + g2

> Lamd := solve(Fλ,lamd);

10 x\*g1 - g2 + 4 g1\*y + 4 x\*g2 + 2 y\*g2 - g1

-------------------------------------------------------

2 \*(4\*g1\*g2 + 5\*g1^2 + g2^2 )

***Метод спуска. Описание метода.***

Рассмотрим нелинейное уравнение:



Запишем выражение для градиента этой функции в точке (точки начального приближения решения уравнения):

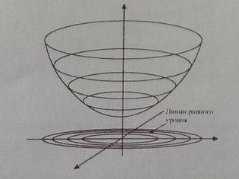
Принимая во внимание свойство градиента можно записать:

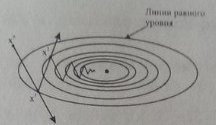


Точка находится ближе к минимуму, чем точка . Откуда следует выражение для итерационного процесса:



Если задается априори (например, остается постоянной или изменяется по заданному закону), то метод называется методом спуска.





*Последовательность направлений перемещения для метода типа спуска.*

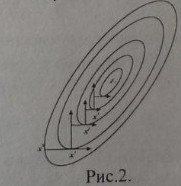
Если  определяется из условия минимума по  функции:



то метод называется методом наискорейшего спуска. Поскольку  является функцией одной переменной, то ее минимум можно искать, используя любой из методов одномерной оптимизации (например, метод золотого сечения). Из выражения для функции  следует:

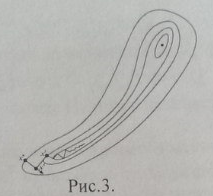


Т.е. последовательные направления перемещения ортогональны.



*Последовательность направлений перемещения для метода типа наискорейшего спуска.*

Преимущество метода спуска состоит в том, что он сходится к точному решению, начиная с любого начального приближения. Недостатком метода является малая скорость сходимости в особенности для так называемых овражных функций.



*Последовательность направлений перемещения для метода типа наискорейшего спуска в случае овражной функции.*

***Алгоритм итерационного процесса метода спуска.***

Задается вектор начального приближения и коэффициент ,

Малое положительное число eps<<1 и предельное число итераций N.

Вычисляется значение градиента в точке и норма градиента в этой точке. Если норма градиента меньше eps, или число итераций больше N, то решение найдено и работа программы закончена. Если норма градиента больше eps, то вычисляется следующее приближение по формуле: 

Далее полагаем и возвращаемся в начало алгоритма.