#### ***Тема 5.8. Технология решения задач многомерной оптимизации средствами MatLab***

Вычисление безусловного экстремума многих переменных y=f(x1,x2,…,xn) в **MatLab**осуществляется командой

[x, y]=fminsearch(name, x0 [, option]),

где name - имя функции, вычисляющей значение y=f(x1,x2,…,xn), зависящей от переменных;

X0- вектор из n элементов, содержащий координаты точки начального приближения;option- параметры, управляющие ходом решения;x- вектор из n элементов, содержащий координаты точки, в которой достигается минимум функции;y- значение функции в точке х.

Вычисление условного экстремума многих переменных y=f(x1,x2,…,xn) осуществляется командой

[x, y]=fmincon(name, x0, A,b[, option]),

где A– матрица, а b – вектор для задания условий ограничения.

Остальные параметры имеют тот же смысл, что и в предыдущей функции

**Пример 4.8-1.Найти и вывести координаты и значение минимума функции двух переменных f(x, y) = (x2 + y2 – 3)2 + (x2 + y2 – 2x – 3)2 + 1, если начальная точка поиска имеет координаты М0(1,1).**

Анализ функции показывает, что minf = 1,x = 0, .



Строим трехмерный график этой функции, чтобы убедиться в наличии минимума. Возьмем интервал х є [-1;1]; y є [1;3].

|  |
| --- |
| **Пример 4.8-1** |
| **>>[X,Y] = meshgrid ( [-1 : 1, 1 : 3] ) ;**  **>>Z=(X.^2+Y.^2-3).^2+(X.^2+Y.^2-2\*X-3).^2+1;**  **>>meshc(X,Y,Z);**  **>>%После построения трехмерного графика выполняется поиск минимума.**  **>> [xmin, minf] = fminsearch ( @ Z, [1; 1] )**  **xmin = - 0.0000 1.7320**  **minf =1.0000**  **>>** |

**Пример 4.8-2.Рассмотрим кубическую параболуy(x)=(x-2)(x-4)(x-6). Найти минимальное значение функции на отрезке [4;6].**

Определим параболу как m-файл cub246.m и вызовем процедуру fmincon( ) со следующими параметрами:

|  |
| --- |
| **Пример 4.8-2** |
| **>>[x,fx]=fmincon(@cub246,4.1,[],[],[],[],4,6)**  **x =5.1547**  **fx =-3.0792**  **>>** |

**Пример 4.8-3.Необходимо найти минимум функцииесли область определения есть прямоугольник ([-4;4],[-4;4]).**



Фактически, область определения можно записать в виде системы неравенств:

.



|  |
| --- |
| **Пример 1.8.6-8** |
| **>>a=[-4;-4];**  **>>b=[4;4];**  **>>% вызвать процедуру fmincon со следующим синтаксисом:**  **>>[x,fmin]=fmincon(@parab,[-2;-2],[],[],[],[],a,b)**  **x =**  **1.0e-007 \***  **-0.1215**  **-0.1215**  **fmin =**  **3.0000**  **>>** |

При таком вызове процедуры fmincon( ) мы нашли точку локального экстремума в области, заданной прямоугольником ([-4;4],[-4;4]).

**Пример 1.8.6-9.Найдем точку условного экстремума, если координатыx1 и x2,связаны отношением .**



В процедуре fmincon( ) это понимается как произведение матриц.



|  |
| --- |
| **Пример 1.8.6-9** |
| **>>[x,fx]=fmincon(@parab,[2;2],[],[],[1 1],[4],[-4;-4],[4;4])**  **x =**  **2.0000**  **2.0000**  **fx =**  **11** |

**Пример 1.8.6-10.Решить похожую по смыслу задачу, но отличающуюся по способу решения. Рассмотрим эту же функцию .**



Для нашей задачи более подходящим является следующий вид: . Данный параболоид пересекает плоскость . На линии пересечения плоскости и параболоида найти точку, наиболее близкую к плоскости .

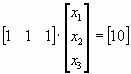


Составим файл-функцию func1.m:

|  |
| --- |
| **Пример 1.8.6-10** |
| **function f=func1(x)**  **f=x(3);**  **Написать файл-функцию условий uslfunc1.m, которые сформулированы выше:**  **function [c,ceq,gc,gceq]=uslfunc1(x)**  **c=[];**  **ceq=[3+x(1)^2+x(2)^2-x(3);x(1)+x(2)+x(3)-10];**  **if nargout>2**  **gc=[];**  **gceq=[2\*x(1)+2\*x(2)-1;3];**  **end;** |

Матрица C- вектор неравенств c(x)≤0. Неравенств у нас не имеется, поэтому матрицаСявляется пустой. Матрица сеq- вектор равенств, который, по сути, описывает наш параболоид и секущую плоскость. Уравнение секущей плоскости в матричном виде:

.



Итерационный процесс начнем с точки (2;2;2).

|  |
| --- |
| **Пример 1.8.6-10** |
| **>>[x,fx]=fmincon(@func1,[2;2;2],[],[],[],[],[],[],...**  **@uslfunc1,optimset('LargeScale','off','Display','off'))**  **x =**  **1.4365**  **1.4365**  **7.1270**  **>>** |