CHAPTER 6

NONLINEAR PROGRAMMING II: MULTI-VARIABLE UNCONSTRAINED MINIMIZATION METHODS

Introduction

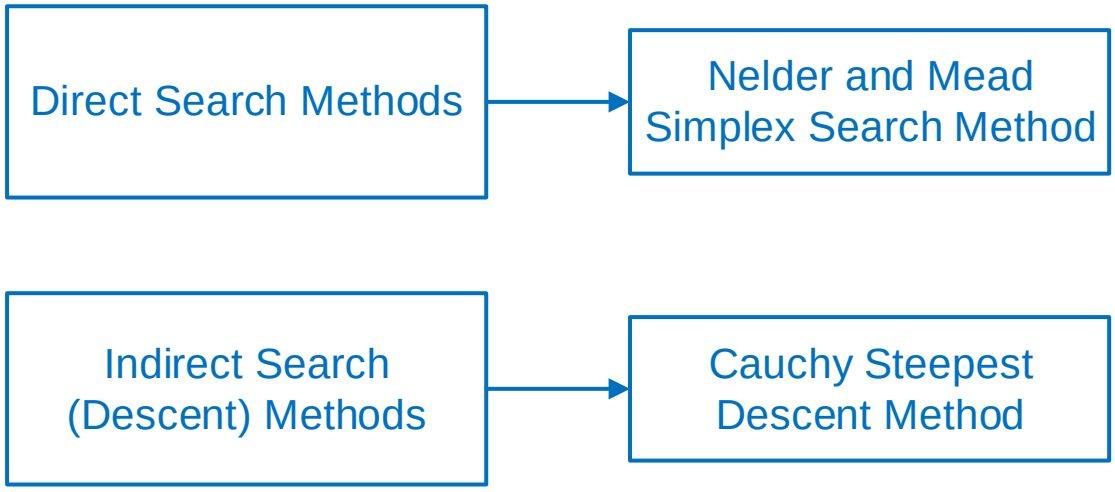
In this chapter, we extend the concepts of Chapter 5 to solve more realistic engineering problems.

These techniques can be used to solve engineering problems such as displacement response of a mechanical system under specific loads determining the eigenvalues and eigenvectors of discrete system.

Most of these algorithms were developed in the sixties and seventies.

Later, we will approach more realistic problems where variables are constrained.

The following is a layout of the algorithms that will be discussed in this chapter.



Finding a Minimum: Direct Search Methods

The direct search methods depend only on finding the objective functions. That is why they are also called *nongradient methods*. These methods are robust to deal with objective functions with discontinuities or when the calculating the gradient is computationally expensive.

In this chapter, we will focus on pattern directions. The basic philosophy is still the same as in the previous chapter.

1. Let i=1
2. Start with an **initial point** X_i
3. Find a suitable **direction** S_i that has the potential to lead toward the minimum
4. Determine a **step length** λ_i to move along S_i direction.
5. Let the new point be,

$$X_{i+1} = X_i + \lambda_i S_i$$

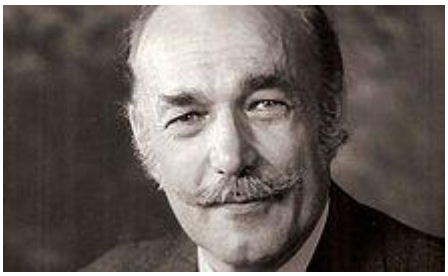
6. Test the optimality of X_{i+1}. This is typically done by checking the neighborhood of X_{i+1}.
7. If X_{i+1} meets the optimality test, STOP. Otherwise, let i=i+1, move to Step 3.

Simplex Search Method

The method is based on creating a polygon with *n+1* vertices in the case of *n* variables. The basic idea is to evaluate these points and to move away from the vertex with the highest function value.

The original algorithm was created by *Spendley* in 1962. However, it was a slow as it the simplex was not adaptive to the changes in the function topography. Several versions were created to address these limitations.

The version we will discuss was developed by *Nelder and Mead* in 1965.



John Ashworth Nelder (1924-2010), Wikipedia

Generating the first simplex (Spendley, 1962)

The first step is to generate the *n* equally-spaced points based on the initial point X_0.

$$X_i = X_0 + pu_i + \sum_{j=1, j\neq i}^{n} q\, u_j \qquad i = 1, 2, \ldots, n$$

where,

X_0 is the initial guess point

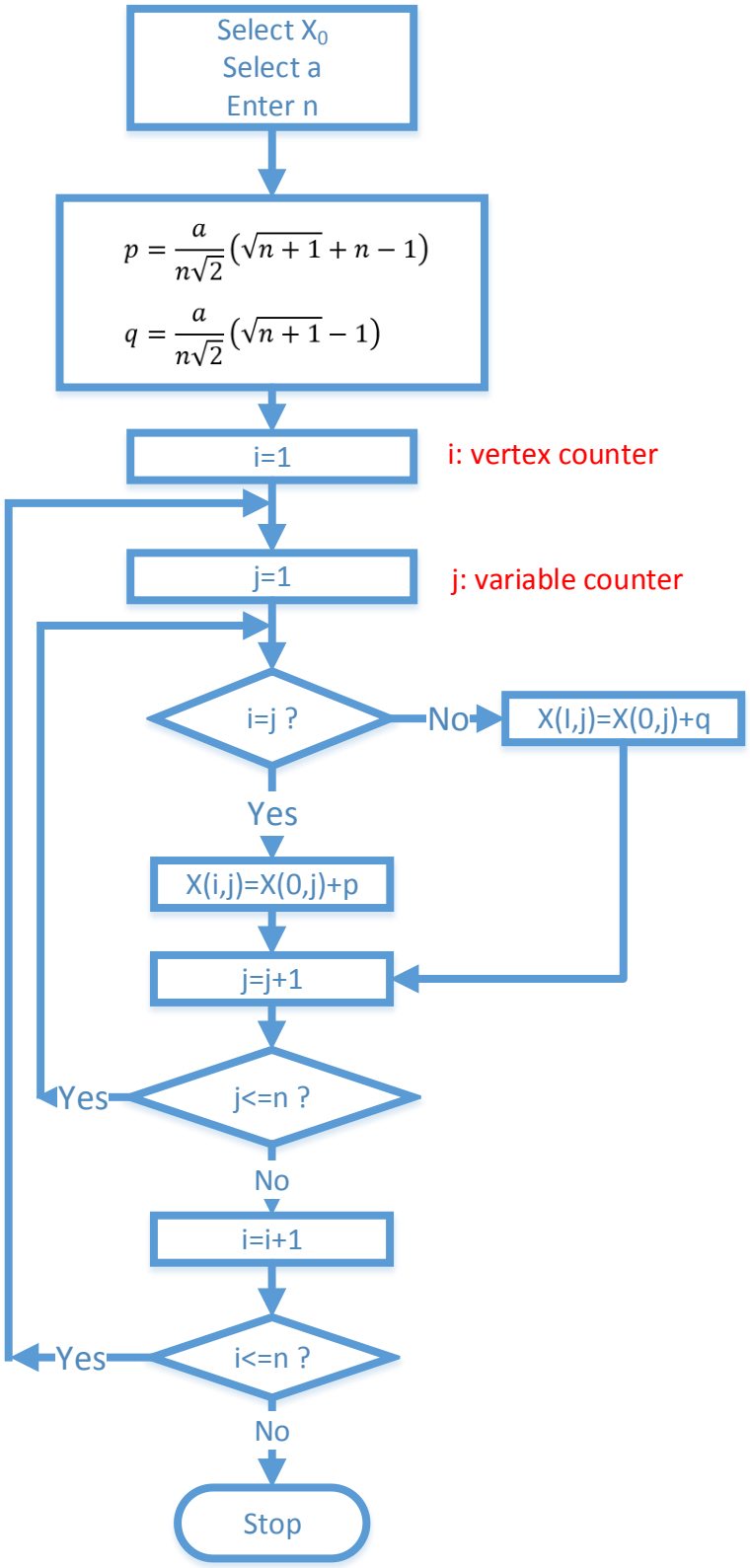
a is the size of the polygon

u_j is the unit vector along the j^{th} coordinate axis

$$p = \frac{a}{n\sqrt{2}}\left(\sqrt{n+1} + n - 1\right)$$

$$q = \frac{a}{n\sqrt{2}}\left(\sqrt{n+1} - 1\right)$$

The flowchart of generating the initial set of points is shown below.



Flowchart for generating the original simplex

Example 6.1

Create the initial vertices for a 2-variable problem. The initial point is $(3, 1)^T$.

$u=[1, 1]^T$

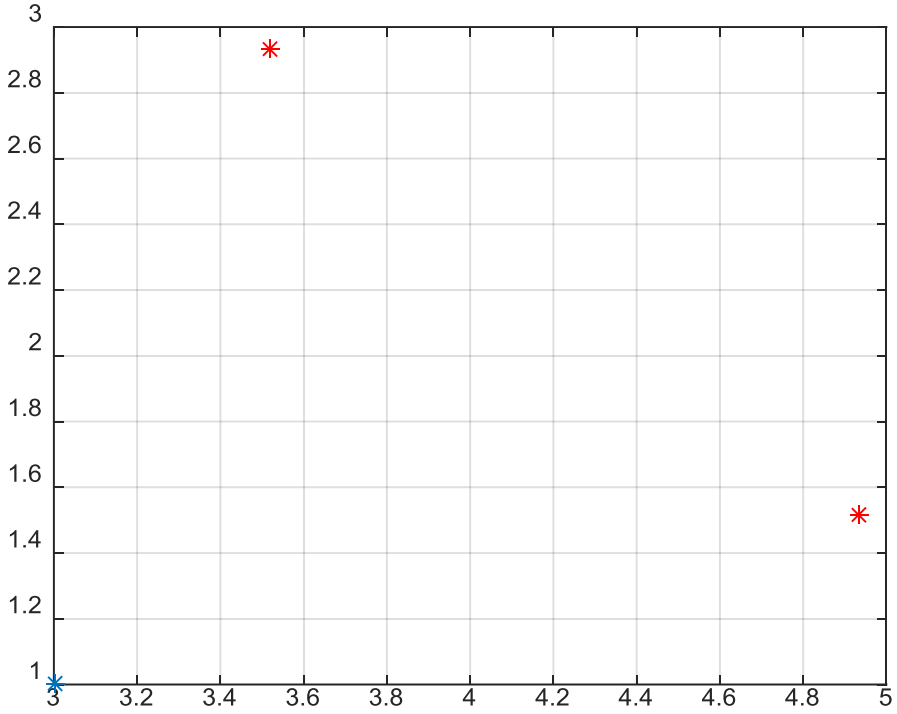
$a=2$

$$p = \frac{a}{n\sqrt{2}}\left(\sqrt{n+1}+n-1\right) = \frac{2}{2\sqrt{2}}\left(\sqrt{2+1}+2-1\right) = \frac{1}{\sqrt{2}}\left(\sqrt{3}+1\right) = 1.932$$

$$q = \frac{a}{n\sqrt{2}}\left(\sqrt{n+1}-1\right) = \frac{2}{2\sqrt{2}}\left(\sqrt{2+1}-1\right) = \frac{1}{\sqrt{2}}\left(\sqrt{3}-1\right) = 0.518$$

$$X_1 = \begin{Bmatrix}3\\1\end{Bmatrix} + 1.932\begin{Bmatrix}1\\0\end{Bmatrix} + 0.518\begin{Bmatrix}0\\1\end{Bmatrix} = \begin{Bmatrix}4.932\\1.518\end{Bmatrix}$$

$$X_2 = \begin{Bmatrix}3\\1\end{Bmatrix} + 1.932\begin{Bmatrix}0\\1\end{Bmatrix} + 0.518\begin{Bmatrix}1\\0\end{Bmatrix} = \begin{Bmatrix}3.518\\2.932\end{Bmatrix}$$



Reflecting the Simplex (Spendley)

After the algorithm generates the first simplex, it tries to move it toward the minimum through these steps:

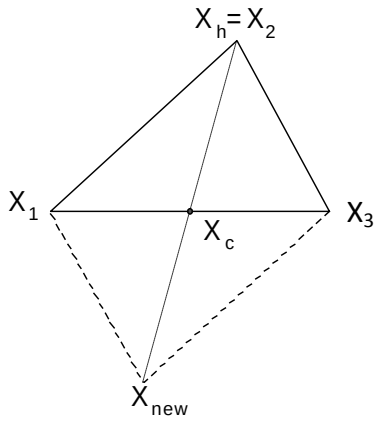
1. Identify the simplex point with the highest function value, X_h
2. Calculate the coordinates of point X_c, which is the centroid of all the simplex points except X_h.

$$X_c = \frac{1}{n} \sum_{j=0, j \neq h}^{n+1} X_j$$

3. A new point, X_{new} is created as a reflection of X_h along the X_h- X_c vector such that,

$$X_{new} = = X_c + \left(X_c - X\right) = 2X_c - X_h$$

4. Replace X_h by X_{new}



Reflection of the Highest Point in the Simplex Algorithm

Adaptive Movement of the Simplex (Nelder and Mead, 1965)

Limiting the movement to reflection only results in a slow progression toward the minimum, especially if the initial guess is far away from minimum.

Nelder and Mead proposed modification that would make the search adaptive by changing the size and direction of the simplex. The following is a summary of the algorithm.

After the algorithm generates the first simplex, it tries to move the simplex toward the minimum. The first step in this process is to identify the following simplex vertices:

- The vertex with the highest function value, X_h,
- The vertex with next highest function value, X_g,
- The vertex with the lowest function value, X_l.

The simplex moves toward the minimum using three operations: *reflection*, *expansion,* and *contraction*.

Decisions for the Simplex Movement

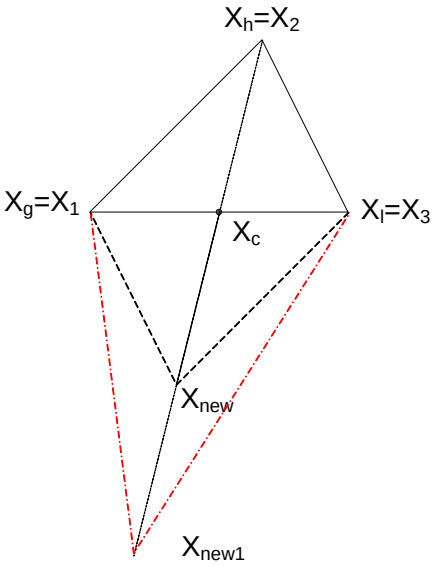
Based on comparing the function values at the simplex points with the function value at the new point, X_{new}, one of these decisions is taken,

$$\begin{aligned}
&if \quad f(X_{new}) \le f(X_l), && \text{explore expansion by generating } X_{new1}\\
&if \quad f(X_l) < f(X_{new}) \le f(X_g), && \text{replace } X_h \text{ by } X_{new}\\
&if \quad f(X_g) < f(X_{new}) \le f(X_h), && \text{replace } X_h \text{ by } X_{new}\\
&if \quad f(X_{new}) > f(X_h), && \text{explore contraction by generating } X_{new2}
\end{aligned}$$

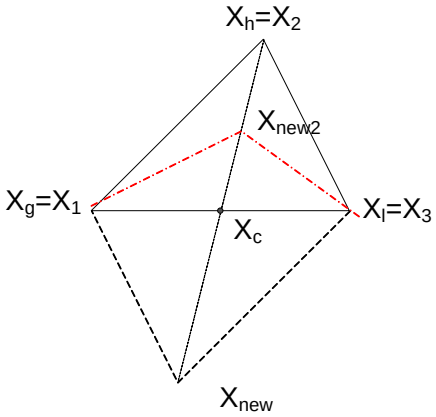
The two new points described in the above equation are generated as follows,

$$X_{new1} = X_c + 2(X_c - X_h)$$

$$X_{new2} = X_c - (0.5)(X_c - X_h)$$



Expansion of the Nelder and Mead Simplex



Contraction of the Nelder and Mead Simplex

The function values at these two new points are evaluated and one of these decisions is taken:

i. If $f(X_{new1})$ is less than $f(X_{new})$, replace X_h by X_{new1}. Otherwise, replace X_h by X_{new}.
ii. If $f(X_{new2})$ is less than $f(X_h)$, replace X_h by X_{new2}. Otherwise, generate a smaller simplex around X_l as follows,

$$X_i = \frac{X_i + X_l}{2} \qquad i = 1..n+1$$

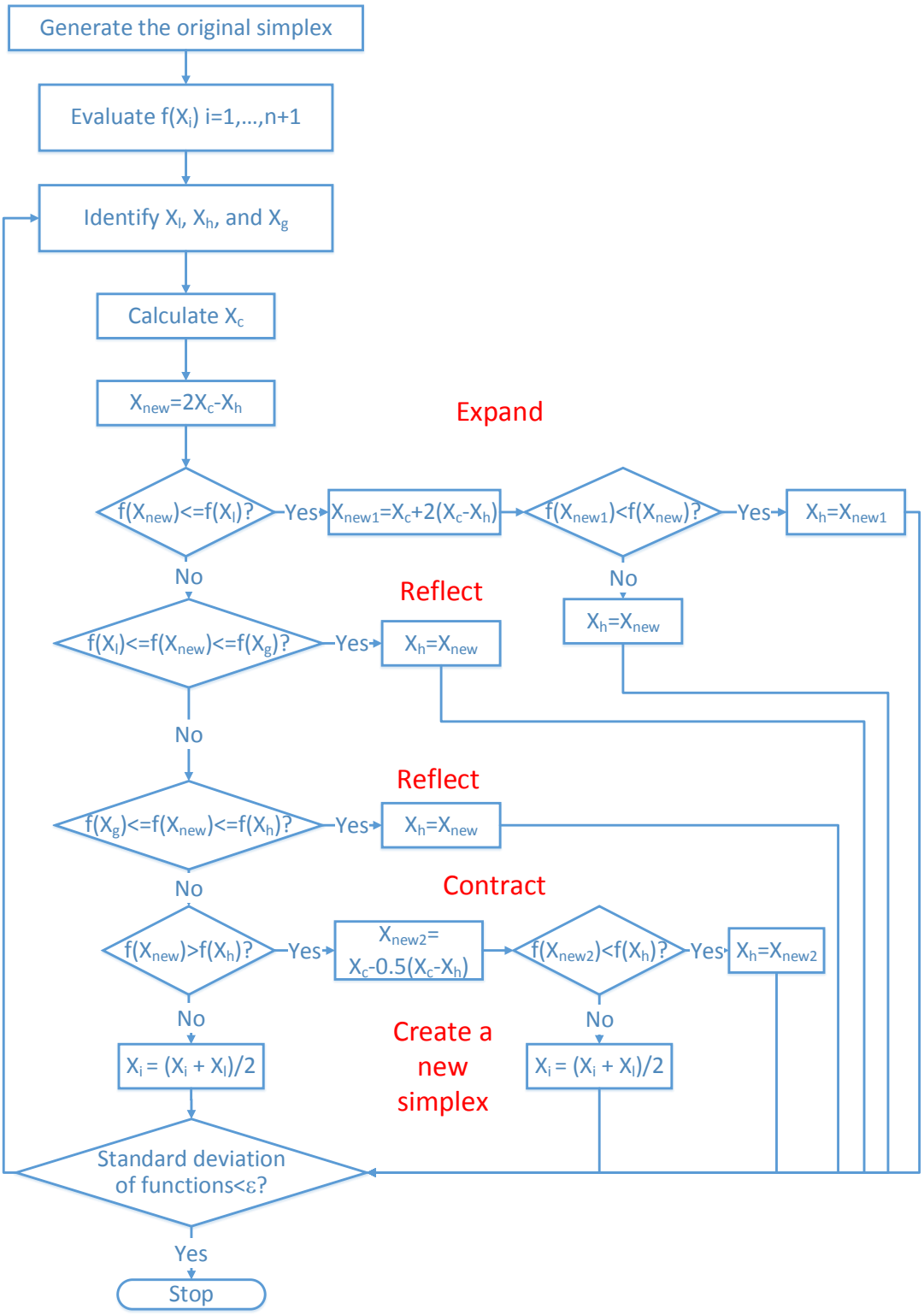
Termination Criterion:

A termination criterion based on the smallness of the standard deviation is evaluated at the end of each iteration.

$$if \quad \sqrt{\frac{1}{n}\sum_{i=0}^{n}\left(f(x_i) - \frac{\sum_{i=0}^{n} f(x_i)}{n+1}\right)^2} < \epsilon \quad stop$$

where ε is the termination accuracy.

The flowchart for the Nelder and Mead Simplex algorithm is presented below.



Flowchart for the Nelder and Mead Simplex Algorithm

Example 6.2

Minimize $f = {x_1}^2 + {x_1}^2$ The initial point is $(3, 1)^T$.

Let, ε=1e-3

<u>Iteration #1</u>

Based on Example, 6.1,

$X_1 = \begin{Bmatrix} 4.932 \\ 1.518 \end{Bmatrix}$

$X_2 = \begin{Bmatrix} 3.518 \\ 2.932 \end{Bmatrix}$

$f(X_0)=10$

$f(X_1)=26.629$

$f(X_2)=26.629$

$X_h = X_2 = \begin{Bmatrix} 3.518 \\ 2.932 \end{Bmatrix}$

$X_g = X_1 = \begin{Bmatrix} 4.932 \\ 1.518 \end{Bmatrix}$

$X_l = X_0 = \begin{Bmatrix} 3 \\ 1 \end{Bmatrix}$

Note: Since we have 2 points with the same function vale, this step assumes that X_2 is taken

$X_c = (1/2)(X_g + X_l) = \begin{Bmatrix} 3.966 \\ 1.259 \end{Bmatrix}$

$X_{new} = 2X_c - X_h = \begin{Bmatrix} 4.414 \\ -0.414 \end{Bmatrix}$

$f(X_{new})=19.528$

$f(X_{new}) < f(X_l)$? No

$f(X_l) <= f(X_{new}) <= f(X_g)$? Yes

$X_h = X_{new} = X_2 = \begin{Bmatrix} 4.414 \\ -0.414 \end{Bmatrix}$

Standard deviation<ε? No

Iteration #2

$f(X_0)=10$

$f(X_1)=26.629$

$f(X_2)=19.528$

$X_h= X_1= \begin{Bmatrix} 4.932 \\ 1.518 \end{Bmatrix}$

$X_{g=} X_1= \begin{Bmatrix} 4.414 \\ -0.414 \end{Bmatrix}$

$X_{l=} X_0= \begin{Bmatrix} 3 \\ 1 \end{Bmatrix}$

$X_{c=} (1/2)(X_g+ X_l)= \begin{Bmatrix} 3.707 \\ 0.293 \end{Bmatrix}$

$X_{new}=2 X_c - X_h= \begin{Bmatrix} 2.482 \\ -0.932 \end{Bmatrix}$

$f(X_{new})=7.029$

$f(X_{new})<f(X_l)$? Yes

$X_{new1}= X_c +2(X_c- X_h)= \begin{Bmatrix} 1.257 \\ -2.157 \end{Bmatrix}$

$f(X_{new1})=6.237$

$f(X_{new1})< f(X_{new})$? Yes

$X_h= X_{new1}= X_1= \begin{Bmatrix} 1.257 \\ -2.157 \end{Bmatrix}$

Standard deviation$<\varepsilon$? No

Iteration #3

f(X0)=10

f(X1)= 6.237

$f(X_2)$=19.528

$X_h = X_2 = \begin{Bmatrix} 4.414 \\ -0.414 \end{Bmatrix}$

$X_g = X_1 = \begin{Bmatrix} 3 \\ 1 \end{Bmatrix}$

$X_l = X_0 = \begin{Bmatrix} 1.257 \\ -2.157 \end{Bmatrix}$

$X_c = (1/2)(X_g + X_l) = \begin{Bmatrix} 2.129 \\ -.579 \end{Bmatrix}$

$X_{new} = 2X_c - X_h = \begin{Bmatrix} -0.157 \\ -0.743 \end{Bmatrix}$

$f(X_{new})$=0.577

$f(X_{new}) < f(X_l)$? Yes

$X_{new1} = X_c + 2(X_c - X_h) = \begin{Bmatrix} -2.443 \\ -0.908 \end{Bmatrix}$

$f(X_{new1})$=6.789

$f(X_{new1}) < f(X_{new})$? No

$X_h = X_{new} = X_2 = \begin{Bmatrix} -0.157 \\ -0.743 \end{Bmatrix}$

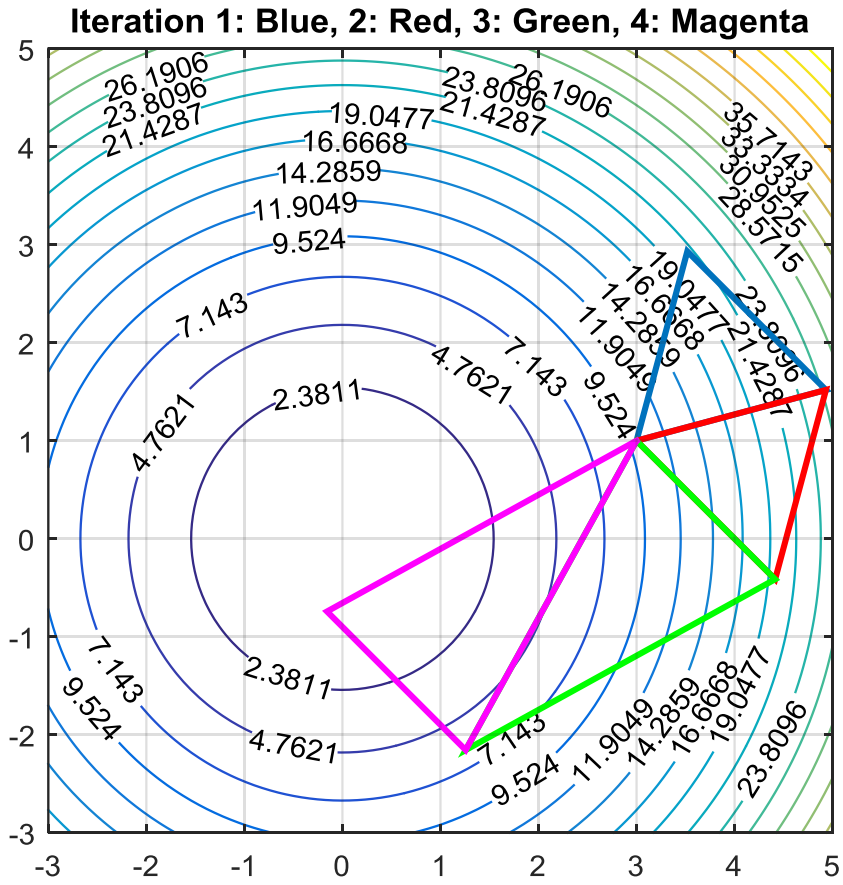
Standard deviation<ε? No

Summary

1:2 Reflection

2:3 Expansion

3:4 Reflection



Homework

Write a computer code based on Nelder and Mead Simplex algorithm. Use this code to solve Problems 6.2 and 6.4.