To understand the basics, I’ve referred to chapter 7 and 8 of the book “Linear and Nonlinear Programming” by David Luenberger and Yinyu Ye

# Mathematical Optimization

* Maximizing or minimizing a real function by systematically choosing input values within an allowed set and computing the value of the function.
* Problems involve maximizing or minimizing a function f(x) subject to x є Ω and Ω is a subset of En.

**Basics**- (which I had to brush up on)

The Del Function

\nabla = \mathbf{\hat{x}} {\partial \over \partial x}  + \mathbf{\hat{y}} {\partial \over \partial y} + \mathbf{\hat{z}} {\partial \over \partial z} , where  are the unit vectors in their respective directions.

࣑shows how vectors and scalars vary with position

1. Gradient: how quickly the function varies  
   \nabla f = {\partial f \over \partial x} \mathbf{\hat{x}} + {\partial f \over \partial y} \mathbf{\hat{y}} + {\partial f \over \partial z} \mathbf{\hat{z}}
2. Curl: It is represented as-\mbox{curl}\;\vec v = \left( {\partial v_z \over \partial y} - {\partial v_y \over \partial z} \right) \mathbf{\hat{x}} + \left( {\partial v_x \over \partial z} - {\partial v_z \over \partial x} \right) \mathbf{\hat{y}} + \left( {\partial v_y \over \partial x} - {\partial v_x \over \partial y} \right) \mathbf{\hat{z}} = \nabla \times \vec v
3. Divergence: Amount of field entering or leaving a point in a region. It is the measure of compression or decompression of a field.

# Convex and Concave Functions

A function f within a set Ω is convex if for all x1, x2 є Ω and every α, 0≤ α ≤1

f(αx­1+(1- α)x2) ≤ αf(x1)+(1- α)f(x2)

**Properties**

1. Let f1 and f2 be convex functions on the convex set Ω. Then the function f1+f2 is convex.
2. If f is convex, af is also convex for all a≥0
3. If f is convex on Ω then  
   Γc={x: xєΩ, f(x)≤c} is convex for all real ‘c’
4. Let f є C1. F is convex over a convex set Ω if and only if,  
   f(y)≥f(x)+࣑f(x)(y-x)

# Few other topics:

1. Maximization and minimization of convex functions
2. Zero Order conditions
3. Global convergence of descent algorithms: In an iterative algorithm, from a particular starting point, if the sequences of points generated are converging to a solution, then the algo is said to be globally converging
4. Speed of convergence

# Line Search Algorithms

Line search techniques are optimization algorithms for one-dimensional minimization problems. They are often used for nonlinear optimization algorithms.

## http://upload.wikimedia.org/wikipedia/commons/thumb/8/8c/Bisection_method.svg/250px-Bisection_method.svg.pngBisection Search

The **bisection method** is a root-finding method which repeatedly bisects an interval and then selects a subinterval in which a root must lie for further processing. The method is also called the binary search method or the dichotomy method.

The method is used to solve *f*(*x*) = 0 within an interval [a,b] such that f(a)f(b)<0

1. m =

2. If f(m) = 0 Stop and return.

3. f(m)f(a) < 0; b = m

4. f(m)f(b) < 0; a = m Image copied from wikipedia

5. < є stop and return, where є is the accepted tolerance.

## http://upload.wikimedia.org/wikipedia/commons/thumb/5/52/GoldenSectionSearch.png/325px-GoldenSectionSearch.pngGolden Section Search

Two quantities a and b are said to be in the *golden ratio* *φ* if:

 \frac{a+b}{a} = \frac{a}{b} = \varphi.

On solving this equation, we arrive at

\varphi = \frac{1 + \sqrt{5}}{2} = 1.61803\,39887\dots

And

\varphi = \frac{1 - \sqrt{5}}{2} = -0.6180\,339887\dots

The golden section search is a technique for finding the minima/maxima of a unimodal function. The function values for three of points whose distances form a golden ratio.

To find the maximum of a function within [*x1,x3*], whose length is *b* + *a*.

We pick two points in the interval [*x1,x3*] and evaluate the function at these points.

The two points x2, x4 are chosen such that each point sub-divides the interval of uncertainty into two parts like  
x2=x3- φ(x3-x1) or x3- φ(a+b)  
x4=x1+ φ(x3-x1) or x1+ φ(a+b)

1. If f (x2 ) < f (x4 ) , then we know that in the range [x2,x4] that the function is increasing. Therefore, at worst, we know that the function value must be greater than *f* (*x*2). Since the function is unimodal, then we know that the maximum cannot be less than *x*2. Thus, we may conclude that the maximum is in the range of (*x*2, *x3*].
2. If *f* (*x*2 ) > *f* (*x*4 ) , then we know the lower bound on the function is *f*(*x4*). Since the function is unimodal, the function’s maximum must be greater than *x4*. Therefore the maximum must lie in the range [*x1,x4*).
3. If *f* (*x*2 ) = *f* (*x*4 ) , then we know the maximum must lie in the range (*x2,x4*) since the points *x2* and *x4* have to be on either side of the maximum.

Given this information, all we have to do is update our interval of uncertainty and to restart the process. For example, if *f* (*x2*) < *f* (*x4*) , and the interval of uncertainty is [*x1,x3*], the new interval becomes (*x2*, *x3*]. If *f* (*x2*) > *f* (*x4*) , and the interval of uncertainty is [*x1,x3*], the new interval becomes [*x1*, *x*2). If *f* (*x2*) = *f* (*x4*) , and the interval of uncertainty is [*x1,x3*] the new interval becomes [*x2*, *x4*).

On successive iterations, we tend to reduce the interval each time, and eventually find out the maximum.

## Fibonacci Search Method

The method determines the minimum value of a function f over a closed interval [c1, c2].

Its operations are similar to the Golden Section method.

Let c1 ≤ x1 < x2 … < xn−1 < xn ≤ c2

Where c1 and c2 are the extreme points and the minimum must lie within this interval

Let

d1= c2 −c1, the initial width of uncertainty

dk = width of uncertainty after k measurements.

Then, if a total of N measurements are to be made, we have

,where the integers Fk are members of the Fibonacci sequence generated by the recurrence relation: n>=2,

The resulting sequence is 1, 1, 2, 3, 5, 8, 13...

To reduce uncertainty to dN, separate values if k are used. Each value of k gives a new interval of uncertainty. Having determined the intervals, the golden section method is used to solve the problem.