DELIGHT PHYSICS LAB

Algorithms for Numerical Computations

Bisection Method:

The method is also called the interval halving method, the binary search method or the dichotomy method. This method is used to find root of an equation in a given interval that is value of 'x' for which f(x) = 0

The method is based on **The Intermediate Value Theorem** which states that if f(x) is a continuous function and there are two real numbers a and b such that f(a)*f(b) 0 and f(b) < 0), then it is guaranteed that it has at least one root between them.

If f(x) is a continuous function within the inteval [a,b] and if the root of f(x) also lies in the given limit:

Algorithm:1**.** Find middle point **c**= (a + b)/2 .

2. If f(c) == 0, then c is the root of the solution.

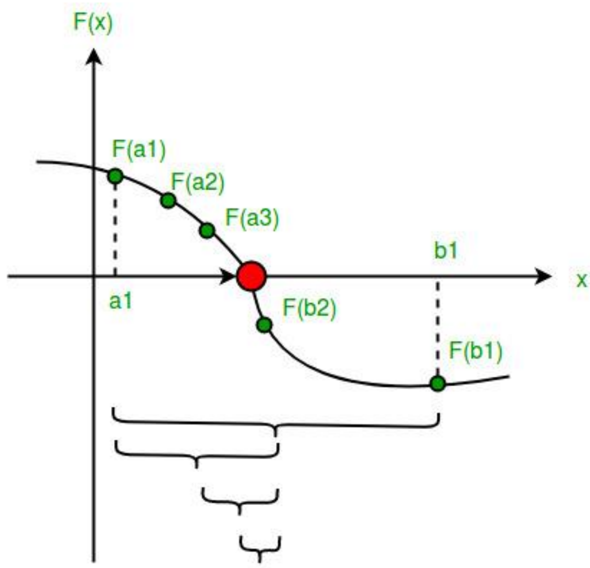
3. Else f(c) != 0

   A. If value f(a)*f(c) < 0 then root lies between a and c. So we recur for a and c

   B. Else If f(b)*f(c) < 0 then root lies between b and c. So we recur b and c.

   C. Else given function doesn't follow one of assumptions.

4. If A or B is satisfied then repeat from 2 till certain accuracy is achieved.



Newton Raphson Method:

Starting from initial guess x_1 the Newton Raphson method uses below formula to find next value of x, i.e., x_{n+1} from previous value x_n.

Algorithm:

1. Start the program.

2. Define the function f(x), f'(x)

3. Enter the initial guess of the root , say x_0

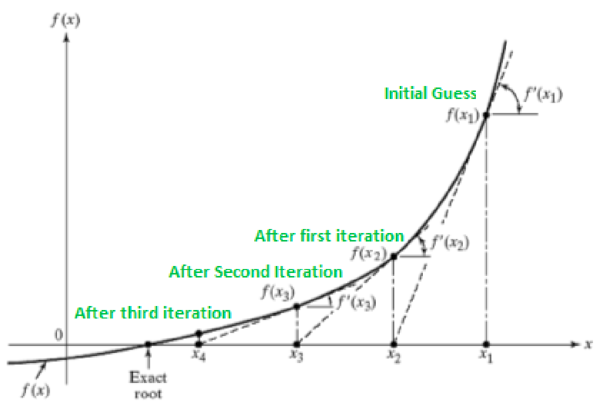
4. Calculate $x_{n+1} = x_n - [f(x_n)/ f'(x_n)]$,
   where n = 0, 1, 2, …

5. If $|x_{n+1} - x_n| < \epsilon$, ϵ being prescribed accuracy then go to step 7.

6. Set $x_n = x_{n+1}$ and go to step 4.

7. Print the value of x_n.

8. Stop the program.



$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \,. \quad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Euler Method for Solving ODE (1^{st} Order):

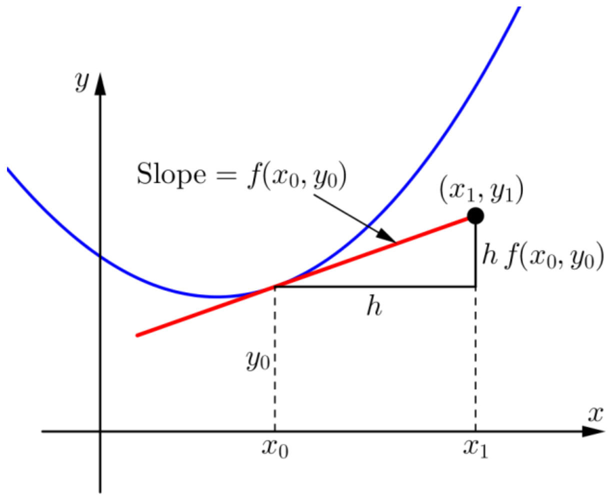
In mathematics and computational science, the Euler method is a first-order numerical procedurefor solving ordinary differential equations (ODEs) with a given initial value. Consider a differential equation

$dy/dx = f(x, y)$ with initial condition $y(x_0)= y_0$

then succesive approximation of this equation can be given by: $y_{n+1} = y_n + hf(t_n, y_n).$

Algorithm:

1. Define function f(x,y)
2. Get the values of x_0, y_0, h and n
   *Here x_0 and y_0 are the initial conditions
   h is the interval size
   n is the interval number
3. Start loop from i=1 to n
4. $y = y_0 + h*f(x_0,y_0)$
   $x = x + h$
5. Print values of y_0 and x_0
6. Check if i < n,
   If yes, assign $x_0 = x$ and $y_0 = y$
   If no, goto 7
7. End loop i
8. Stop the program



Gauss Seidel Method:

In numerical linear algebra, the Gauss–Seidel method, also known as the Liebmann method or the method of successive displacement, is an iterative method used to solve a linear system of equations. It is named after the German mathematicians Carl Friedrich Gauss and Philipp Ludwig von Seidel.

$$A\mathbf{x} = \mathbf{b}$$

Algorithm

1. Input A and b as array(Matrix).
2. Guess the solution & push them into 1d matrix X.
3. Create a lower triangular matrix(L) from A.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

4. Construct a upper triangulat matrix(U)
5. Iterate $L^{-1}(b – UX)$ with certain number of loops.
6. Stop the program and print the final solutions of the variables.

$$A = L_* + U \qquad \text{where} \qquad L_* = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad U = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}.$$

$$L_*\mathbf{x} = \mathbf{b} - U\mathbf{x} \qquad \mathbf{x}^{(k+1)} = L_*^{-1}(\mathbf{b} - U\mathbf{x}^{(k)}).$$