Computer Aided Design Model for Optimization Techniques (Newton's Method)

Ashish G. Shende, Sushil B. Wankhede and Rahul R. Burde
M.Tech Department of Chemical Engineering
Laxminarayan Institute of Techonology, Nagpur-440033
Maharastra, India

ABSTRACT
A Computer Aided Design (CAD) module is developed for solving Descent Methods. Here mainly we focus on Unconstrained optimization which specially includes Newton's Method using Visual Basic 6.0. The module is made using Scrip Control and finite differentiation method. A sample design problem was solved using the module. Results from the module and manual calculations were compared.

General Terms
Optimization Techniques, Finite Differentiation method, Scrip Control.

Keywords
Optimization Techniques, Newton's Method, Unconstrained optimization. Descent Methods, Visual basic 6.0.

1. INTRODUCTION
Optimization is a subject that deals with the problem of minimizing or maximizing a certain function in a finite dimensional Euclidean space over a subset of that space, which is usually determined by functional inequalities[1]. A constraint optimization problem can be defined as a regular constraint satisfaction problem in which constraints are weighted and the goal is to find a solution maximizing the weight of satisfied constraints. In optimization, the optimality conditions for interior points are usually much simpler than the optimality conditions for boundary points. Boundary points appear more prominently in constrained optimization, when one tries to optimize a function, subject to several functional constraints. For this reason, the optimality conditions for boundary points are generally discussed in constrained optimization, whereas the optimality conditions for interior points are discussed in unconstrained optimization, regardless of whether the optimization problem at hand has constraints.[1,2]

2. GENERAL ALGORITHM FOR OPTIMIZATION PROBLEMS
All the unconstrained minimization methods are iterative in nature and hence they start from an initial trial solution and proceed towards the minimum points in sequential manner. The general iterative scheme is shown in fig.1 as a flow diagram. It is important to note that all the unconstrained minimization method.[1][2]

i. Requires an initial point $\mathbf{x_1}$ to start the iterative procedure, and

ii. Differ from one another only in the method of generating the new point $\mathbf{X_{i+1}}$ (from $\mathbf{x_i}$) and in testing the point $\mathbf{X_{i+1}}$ for optimality.

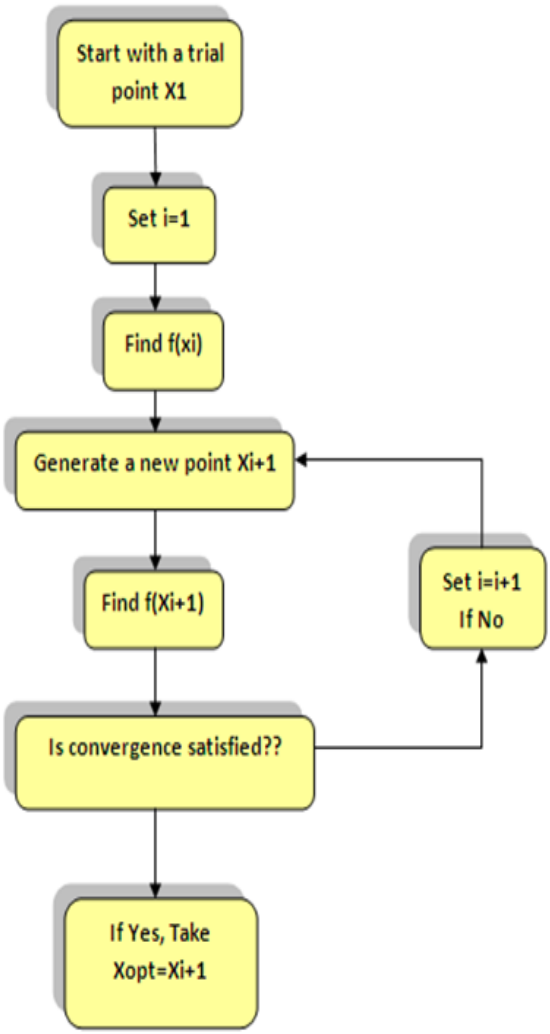

Fig1. Algorithm for Optimization [1]

3. CLASSIFICATION OF UNCONSTRAINED MINIMIZATION METHODS
Several methods are available for solving an unconstrained minimization problem. These methods can be classified into two broad categories as direct search methods and descent method as indicated in the table. the direct search method required only the objective function in finding the minimum and hence are often called the nongradient methods. The descent technique require, in addition to the function values, the first and in some cases the second derivative of the

objective function. Since the more information about the function being minimized is used through the use of derivatives, descent methods are generally more efficient than the direct search techniques. The descent methods are known as gradient methods.[1]-[3]

Direct Search Methods	Descent methods
Random search method	Steepest descent (Cauchy) method
Grid search method	Fletcher-Reeves method
Univariate method	Newton's method
Pattern search methods	Marquardt methods
Powell's method	Quasi-Newton method
Hooke-Jeeves methods	Davidon-Fletcher-powell method
Rosenbrock's methods	Broyden-Fletcher-Goldfarb-Shanno
Simplex method	method

Table1.classification of methods

3.1 Gradient of a Function

The gradient of a function is an n component vector given by

$$\nabla f = \begin{Bmatrix} \frac{\partial f}{\partial x1} \\ \frac{\partial f}{\partial x2} \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial f}{\partial xn} \end{Bmatrix}$$

The gradient has very important property. If we move along the gradient direction from any point in n-dimensional space, the function value increases at the fastest rate. Hence gradient direction is called the direction of steepest ascent. In this project we select two methods one is the Newton's method. The description, theory and the examples on the both method we will see in the next section.

4. NEWTON METHOD [2]

In mathematics, Newton's method is an iterative method for finding roots of equations. In optimization, Newton's method is specialized to find stationary points of differentiable functions, which are the zeros of the derivative function.

Newton's Method attempts to construct a sequence x_n from an initial guess x_0 that converges towards x^* such that $f'(x^*)=0$ This x^* is called a stationary point of f(.).

The second order Taylor expansion $f_T(x)$ of function f(.) around x_n (where $\Delta x=x-x_n$) is:

$$f_T(x_n+\Delta x)=f_T(x_n)+ f'_T(x_n)\Delta x+\frac{1}{2} f''_T(x_n)\Delta x^2$$

Attains its extremum when its derivative with respect to Δx is equal to zero, i.e. when Δx solves the linear equation:

$$f'(x_n)\Delta x+ f''(x_n)\Delta x = 0$$

(Considering the right-hand side of the above equation as a quadratic in Δx , with constant coefficients.)

Thus, provided that f(x) is a twice-differentiable function well approximated by its second order Taylor expansion and the initial guess x_0 is chosen close enough to x*, the

$$\Delta x = x-x_n = -\frac{f'(x_n)}{f''(x_n)}$$

sequence (x_n) defined by:

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}, \qquad n = 0 , 1, 2, \ldots$$

Will converge towards a root of f', i.e. x_* for which

$$f'(x_*) = 0.$$

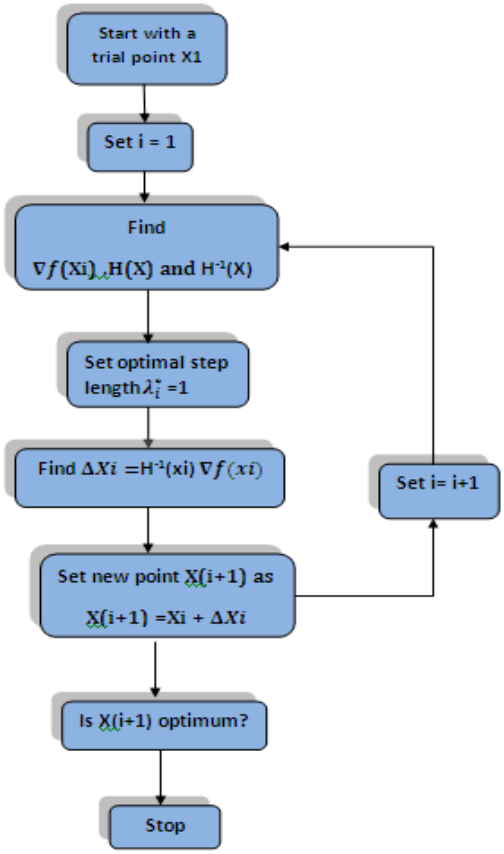

Fig2. Flowchart and Algorithm for Newton's Method[1]

4.1 Geometric interpretation

The geometric interpretation of Newton's method is that at each iteration one approximates f(x) by a quadratic function around x_n, and then takes a step towards the maximum/minimum of that quadratic function (in higher dimensions, this may also be a saddle point). Note that if f(x) happens to *be* a quadratic function, then the exact extreme is found in one-step[2].

The above iterative scheme can be generalized to several dimensions by replacing the derivative with the gradient, $\nabla f(x)$, and the reciprocal of the second derivative with the inverse of the Hessian matrix, Hf(x).Newton's method converges much faster towards a local maximum or minimum than gradient descent. In fact, every local minimum has a neighbourhood N such that, if we start with $x_0 \in N$ Newton's method with step size λ=1 converges in quadratic manner. Finding the inverse of the Hessian in high dimensions can be an expensive operation. In such cases,

instead of directly inverting the Hessian it's better to calculate the vector $P_n = [Hf(x_n)]^{-1}\nabla f(x_n)$ as the solution to the system of linear equations.

$$P_n[Hf(x_n)]^{-1} = \nabla f(x_n)$$

Which may be solved by various factorizations or approximately (but to great accuracy) using iterative methods. Some functions are poorly approximated by quadratics, particularly when far from a maximum or minimum. In these cases, approximations other than quadratic may be more appropriate[4].

The minimum of F(x) in the direction of S^n is obtained by differentiating the quadratic approximation of F(x) with respect to each of the component of X and equating the resulting the expression to zero.

$$x_{n+1} = x_n - [Hf(x_n)]^{-1}\nabla f(x_n), \qquad n \geq o\ .$$

4.2 Example[2]

Minimize $\boldsymbol{f(x_1,x_2) = x_1 . x_2 + 2x_1^2 + 2x_2^2 + 2x_1 * x_2}$ Staring from the point X1= $\binom{0}{0}$. Using Newton's Method.

Solution

To find X_2 we require $[J_1]^{-1}$ where

$$[J_1] = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1\, \partial x_2} \\ \frac{\partial^2 f}{\partial x_1\, \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}$$

Therefore,

$$[J_1]^{-1} = \frac{1}{4}\begin{bmatrix} 2 & -2 \\ -2 & 4 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{-1}{2} \\ \frac{-1}{2} & 1 \end{bmatrix}$$

$$g_1 = \begin{Bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{Bmatrix} \text{ at } x_1 = \begin{Bmatrix} 1+4x1+2x2 \\ -1+2x1+2x2 \end{Bmatrix} \text{ at } (0,0)$$

$$g_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$\mathbf{X_2 = X_1 - [J_1]^{-1} g_1}$$

$$X_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{bmatrix} \frac{1}{2} & \frac{-1}{2} \\ \frac{-1}{2} & 1 \end{bmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} -1 \\ \frac{3}{2} \end{pmatrix}$$

To see whether or not X_2 is optimum point, we evaluate

$$g_2 = \begin{Bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{Bmatrix}_{\text{at } x2} = \begin{Bmatrix} 1+4x1+2x2 \\ -1+2x1+2x2 \end{Bmatrix} \text{ at } (-1, 3/2)$$

$$g_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

As $g_2 = 0$, X_2 is the optimum point . thus the method has converged in one iteration for this quadratic function.

If *f* (X) is a non quadratic function. Newton's method may sometimes diverge, and it may converge to saddle points and relative maxima. This problem can be avoided by modifying the equation.

4.3 Merits and Demerits

The modification indicated by the equation has a number of advantages. First, it will find the minimum in the lesser number of steps compare to the original method. Second, it finds the minimum point in all cases whereas the original method may not converge in some cases. Third, it usually avoids convergence to a saddle point or maximum. With all these advantages, this method appears to be the most powerful minimisation method. Despite these advantages, the method is not very useful in practise, due to the following feature of this method.[2,5]

1. It requires the storing of n x n matrix [J_i].
2. It becomes very difficult and sometimes, impossible to compute the element of the matrix [J_i].
3. It requires the inversion of the matrix [J_i] at each step.
4. It requires the evolution of the quantity [J_i]$^{-1}$ ∇fi at each step. These features make the method impractical for problems involving a complicating objective with a large number of variables.

5. CODING LANGUAGE

The design program was developed using Visual basic 6.0 because of its advanced features that are well suited to modular programming. Two Special tools were used while making the software; one is Script Control for reading the equation and Flex grid for iterative purpose. Visual basic 6.0 is like the other programming language it creates visualized interface between the user.[7,8]

6. CONCLUSION

Manual calculations has been done as shown in Section 4.2 having optimum value as [-1, 3/2] and that were compared with the optimum solutions obtained with the help of software. It was found that they are nearly equal (see fig.3.a,b,c) but hand calculations were time consuming. On the other hand software solved the problems in the fraction of seconds. It is user-friendly and requires a less computer knowledge or skill. The software is also well equipped with the theory behind the method. The user does not require searching for theory and this is one of the good things about the software which is developed for Newton's method.

7. SOFTWARE SCREENS

Here user can insert equation and the initial guesses or say starting point for the problem, on clicking on the solve button this software can self calculate the derivative of these, and also calculate the hessian matrix, and give the solution to the problem. This screen shows the iterative solution for the given problem by taking the calculated point as next as initial ans so on

Fig 3.a:- Home Screen

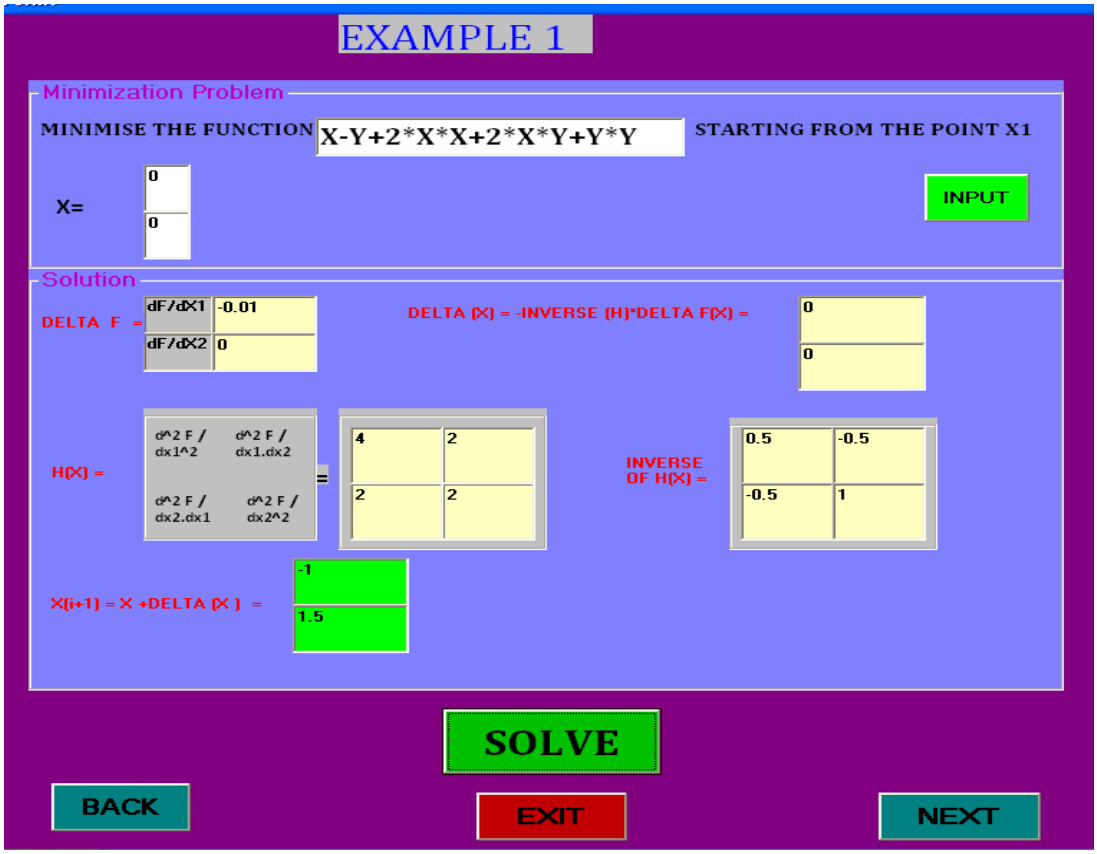

Fig 3.b:- Input & Solution Screen

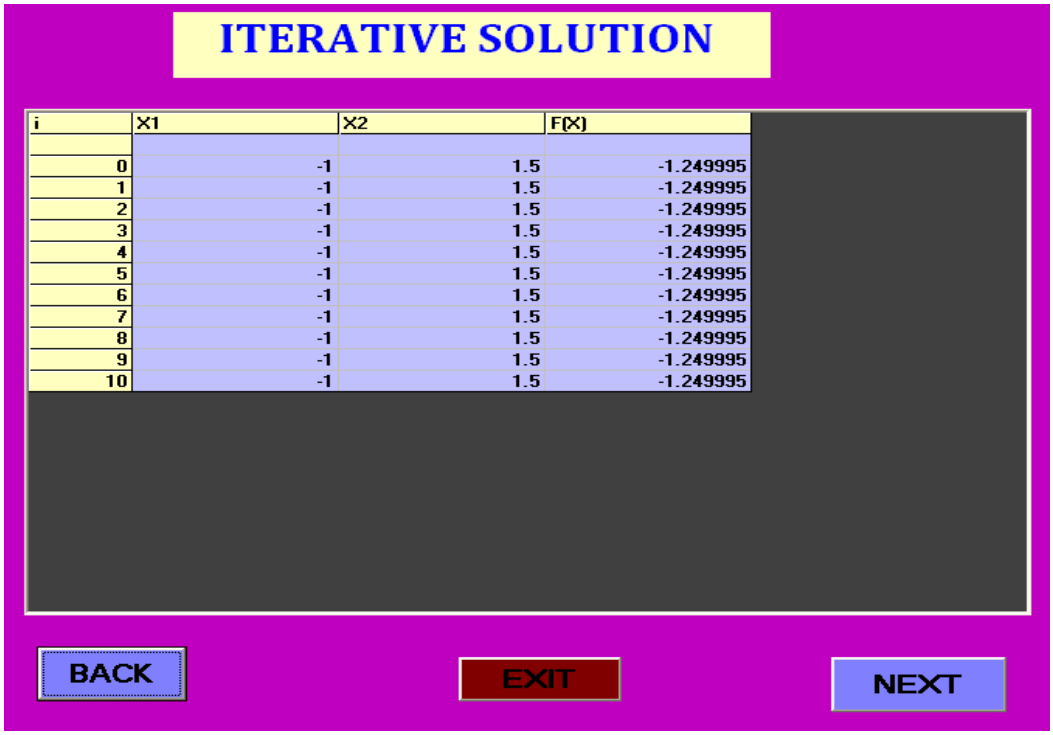

Fig 3.c:- Iterative Solution Screen

8. REFERENCES

[1] Optimization of Chemical Process, T.F. Edgar, D.M. Himmalblau, McGraw-Hill International Edition, Chemical series.

[2] Engineering Optimisation Theory And Practice, Third Edition, S.S. Rao

[3] Mathematics In Chemical Engineering, Bruce A. Finlayson, Lorentz T. Biegler, Ignacio E. Grossmann.

[4] Theory And Problems of Operation Research, Richard Bronson, Schaum's Outline Series, Asian Students edition

[5] Operations Research An Introduction ,Fourth Edition, Hamdy A. Taha, Maxwell Macmillan Publishing Company, New York

[6] Higher Engineering Mathematics, 38th Edition, Dr. B. S. Grewal, Khanna Publication

[7] Visual Basic-6 Programming Black Book, Steven Holzner, Dreamtech Press

[8] The Complete Reference Visual Basic 6, Noel Jerk, Tata McGraw Hill

[9] Mastering Visual Basic-6 Book

[10]Newton's Method for Unconstrained Optimization ,Robert M. Freund February, 2004 , Massachusetts Institute of Technology.

[11] J. E. Dennis and R. B. Schnabel. Numerical methods for Unconstrained Nonlinear Equations.SIAM, Philadelphia, 1996.

[12]Iterative Methods for Optimization, C.T. Kelley, North Carolina State University, Raleigh, North Carolina

[13]Newton-type Methods, Walter Murray, Department of Management Science and Engineering, Stanford University, Stanford, CA, July 5, 2010