The Davidon Fletcher and Powel Method Tested on Quadratic Functions

Awatif M. A. Elsiddieg

Abstract:- In this work we give a detailed look to the DFP method for solving unconstrained optimization problems. Section(0), shows the history and developments of the method . Section (1), general theory of the problem is described , constricting on the practical side in the description . Section (2), Newton's method is described . It constitute a solid base, both theoretical and practical , for the class of methods known as Quasi –Newton method. From this class comes the DFP method is described in section (3) . Detailed result on the DFP are shown in this work .Section (4) shows a practical implementation of the DFP method on quadratic function to test the theoretical results shown in the work.

Tables of Notations:

$\underline{x} = \begin{bmatrix} x^{(1)} \\ \vdots \\ x^{(n)} \end{bmatrix} \equiv$ variable in an optimization (column vector in R^n)

$\underline{x}^k,\ k = 1,2, \ldots \equiv$.iterates in an iteration method .

$\underline{x}^T = (\underline{x}^{(1)} \ldots \ldots \underline{x}^{(n)}$Arrow vector (transpose of a column vector) .

$\underline{x}^* \equiv$Local minimizer or local solution.

$\underline{P}, \underline{P}^{(k)} \equiv$Search direction (on iteration k) .

$f(\underline{x}) \equiv$ Objective function .

$\nabla \equiv$ First derivative operator .

$$\nabla f(\underline{x}) = g(\underline{x}) = \begin{bmatrix} \frac{\partial f}{\partial x^{(1)}} \\ \vdots \\ \frac{\partial f}{\partial x^{(n)}} \end{bmatrix}.$$

$\underline{\delta}, \underline{\delta}^{(k)} \equiv$Correction to $\underline{x}^{(k)}$

$\underline{p} \equiv$ Feasible direction.

$f^{(k)}, g^{(k)} \equiv f(\underline{x}^{(k)}), g(\underline{x}^{(k)})$ Respectively.

$A^T \equiv$ A transpose.

$C^{(k)} \equiv$Set of k times continuously differentiable functions .

$H^{(k)} \equiv$Hessian matrix .

$$\nabla^2 f(\underline{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^{(1)2}} & \frac{\partial^2 f}{\partial x^{(1)} \partial x^{(n)}} \\ \vdots & \vdots \\ \frac{\partial^2 f}{\partial x^{(n)} \partial x^{(1)}} & \frac{\partial^2 f}{\partial x^{(n)2}} \end{bmatrix} = G .$$

$\exists \equiv$ There exist .

$\forall \equiv$ for all .

$\in \equiv$ belongs to .

$\subset \equiv$ conclusion .

$> \equiv$ greater than .

$< \equiv$ less than .

$\geq \equiv$ greater than or equal .

$\leq \equiv$ less than or equal .

$\lambda^{*(i)} \equiv$ is the minimizing step length in iteration .

Revised Version Manuscript Received on October 06, 2015.

Awatif M. A. Elsiddieg, Department of Mathematical Sciences, Elneilain University Math, Khartoum Sudan. (Present address) Prince Sattam Bin Abdul-Aziz University Faculty of Science & Humanities Studies Math. Dept. Hotat Bani –Tamim P. O .Box 13 Kingdom of Saudi Arabia.

I. INTRODUCTION

Section (0):

Optimization might be defined as science of determining the (best) solution to certain mathematically defined problems , which are often models of physical reality . It involves the study of optimality criteria for problems. Many methods for solving minimization problems are variants of Newton's method , which requires the specification of the Hessian matrix of second derivatives .Quasi – Newton's methods are intended for the situation where the Hessian is expensive or difficult to calculate . Quasi – Newton's methods use only the first derivatives to build an approximate Hessian over a number of iterations. This approximation is updated each iteration by a matrix of low rank, in unconstrained minimization. his work is the variable metric method by Davidon , Fletcher and Powel DFP.[3] Steepest descent type methods had been applied in some physics problems. The Newton's method in many variables was known .

Cauchy made the first application of the steepest descent method to solve unconstrained minimization problems .

The development of the simplex method by [1]for linear problems. The work by Kuhn and Tucker [2]on the necessary and sufficiency conditions for optimal solution of programming problems laid the foundation for a great deal of later research in non-linear problems laid the foundation for a great deal of later research in non-linear programming . First application to Newton's method is by [5].

Section (1):

In this section, we give the theory of optimization problems , we give definitions and theorems and some examples .

Definitions and theorems:

Def.(1): (Unconstrained Optimization Problems)
The problem takes the form:

$$\min_{\underline{x} \in \Re^n} \quad f(\underline{x}) .$$

Subject to $\underline{x} \in \Re^n$

Is a compact set) Where f is a continuous real valued function ($\Re^n$

Def. (2): A point $\underline{x} \in \Re^n$is said to be a relative minimum point or a local $\Re^n$ if $\exists$ an $\mathcal{E} > 0$ such that f($\underline{x}$)$\geq$ f($\underline{x}^*$) $\forall \underline{x} \in \Re^n$a strict relative minimum point of over $\Re^n$.

Def.(3): (constrained optimization problem)
The general form of a constrained optimization problem the form.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

$$\min_{\underline{x}\in\Re^n} \; f(\underline{x}) .$$

$c_i(\underline{x}) = 0 \,,\; i = 1,2,\dots\dots,p$ Subject to

$\mathrm{c_i}(\underline{\mathrm{x}}) \geq 0, \quad \mathrm{i} = p+1,\dots\dots,n$ (1).

Where c_i is the ith constraint function . the constraints $c_i(\underline{x}) = 0$ are termed equality constraints and the set of such constraints is denoted by (E) and the constraints $c_i(\underline{x}) \geq 0$ are termed inequality constraints denoted by(I).

Def. (4):A point $\underline{x} \in \Re^n$ satisfying (1) is called a feasible point and the set of all such points is called the feasible region R .

R= $\{\, \underline{x} \in \Re^n , c_i(x) \geq 0, i \in E , i \in I\}$

Def. (5) : Given $\underline{x} \in \Re^n$we say that a vector $\underline{p}$ is a feasible direction at $\underline{x}$ if there is an $\bar{\alpha} > 0$ such that $\underline{x} + \alpha\underline{p} \in \mathcal{R}^n, 0 < \alpha < \bar{\alpha}$. points is

II. Some Theoretical Preliminaries

Def. (6): (lines on $\Re^n$)

Let $\underline{x}'$ be a fixed point of $\Re^n$, let $\underline{p}$ be a fixed vector having length unity($\|p\|_2^2$ =1) ($\underline{p}$ here is termed a direction) , the line that passes through $\underline{x}'$ and having direction $\underline{p}$ is define by the set $\{\underline{x}(\alpha): \underline{x}(\alpha) = \underline{x}' + \alpha\underline{p}\}$. Thus $\underline{x}(0) = \underline{x}'$, α is usually taken to be positive .

Taylor's Expansion of $f(\underline{x})$:

If f is smooth enough we can obtain an expression for Taylor's expansion of f similar to functions of one variable . Before we do that we consider the rate of change of f along given line . Let the line $\underline{x}(\alpha) = \underline{x}' + \alpha\underline{p}$ be given . Along this line , f changes with α , is considered as a function of α , so we write :

$f(\alpha) = f\big(\underline{x}(\alpha)\big) = f\left(\underline{x}' + \alpha\underline{p}\right) = f\big(x^{(1)\prime} + \alpha p^{(1)}, x^{(2)\prime} + \alpha p2,\dots,xn' + \alpha pn$

The rate of change of f with respect to α , $\frac{df}{d\alpha}$ is given by :

$$\frac{df}{d\alpha} = \frac{\partial}{\partial \mathrm{x}^{(\mathrm{i})}}\frac{\mathrm{dx}^{(\mathrm{i})}}{\mathrm{d}\alpha} + \dots\dots + \frac{\partial f}{\partial x^{(n)}}\frac{dx^{(n)}}{d\alpha} = \sum_{i=1}^{n}\frac{\partial f}{\partial x^{(i)}}\frac{dx^{(i)}}{d\alpha} .$$

$\underline{x}^{(i)} = \underline{x}^{(i)\prime} + \alpha\underline{p}^{(i)}$. Now $\frac{d\underline{x}^{(i)}}{d\alpha} = \underline{p}^{(i)}$.

$$\therefore \frac{df}{d\alpha} = \sum_{i=1}^{n}\frac{\partial f}{\partial \underline{x}^{(i)}}p^{(i)} = \underline{p}^T \nabla f\left(\underline{x}(\alpha)\right) = \underline{\mathrm{p}}^{\mathrm{T}} g\left(\underline{x}(\alpha)\right).$$

$\frac{df}{d\alpha} = \underline{p}^T g\left(\underline{x}(\alpha)\right)$ or

Also we obtain the second derivative of f with respect to α as follows :

$\frac{d^2f}{d\alpha^2} = \frac{d}{d\alpha}\left(\frac{df}{d\alpha}\right) = \frac{d}{d\alpha}\left(\sum_{i=1}^{n}\frac{\partial f}{\partial x^{(i)}}\underline{p}^{(i)}\right)$.

$$= \sum_{\mathrm{i}=1}^{\mathrm{n}}\frac{d}{d\alpha}\left(\frac{\partial f}{\partial \mathrm{x}^{(\mathrm{i})}}\right)\underline{\mathrm{p}}^{(\mathrm{i})}.$$

$\frac{d}{d\alpha}\left(\frac{\partial f}{\partial x^{(i)}}\right) = \frac{\partial^2 f}{\partial x^{(1)}\partial x^{(i)}}\frac{dx^{(i)}}{d\alpha} + \dots\dots + \frac{\partial^2 f}{\partial x^{(1)}\partial x^{(n)}}\frac{dx^{(i)}}{d\alpha}$ .Now

$$= \sum_{i=1}^{n}\frac{d}{d\alpha}\left(\frac{\partial f}{\partial x^{(i)}}\right)\underline{p}^{(i)}.$$

$= \sum_{i=1}^{n}\frac{\partial^2 f}{\partial x^{(i)}\partial x^{(i)}}\underline{p}^{(i)}. = \sum_{j=1}^{n}\frac{\partial^2 f}{\partial x^{(j)}\partial x^{(j)}}\underline{p}^{(j)}$.

$$= \left(G\left(\underline{x}(\alpha)\underline{p}\right)^{(i)}\right).$$

$\frac{d^2f}{d\alpha^2} = \sum_{i=1}^{n}\underline{p}^{(i)}\left(G\underline{p}\right)^{(i)}$. Thus $\therefore \frac{d^2f}{d\alpha^2} = \underline{p}^T G\underline{p}$. now we have

$f(\alpha) = f(0) + \alpha f'(0) + \frac{\alpha^2}{2}f''(0) + \dots\dots$.

$$f'(0) = \left.\frac{df}{d\alpha}\right|_{\alpha=0} = \underline{P}^T g(\underline{x}') .$$

$f''(0) = \left.\frac{d^2f}{d\alpha^2}\right|_{\alpha=0} = \underline{p}^T G(\underline{x}')\underline{p}$.

$$f(0) = f\left(\underline{x}(0)\right) = f(\underline{x}') .$$

$f(\alpha) = f\left(\underline{x}(\alpha)\right) = f\left(\underline{x}' + \alpha\underline{p}\right). = f(\underline{x}') + \alpha\underline{p}^T g(\underline{x}') + \frac{\alpha^2}{2}\underline{p}^T G(\underline{x}')\underline{p} + \dots\dots$

The Taylor expansion of $f(\underline{x})$along the line $\underline{x}(\alpha) = \underline{x}' + \alpha\underline{p}$ about

$\underline{x} = \underline{x}'$. putting $\underline{h} = \alpha\underline{p}$,we get

$$f(\underline{x}' + \underline{h}) = f(\underline{x}') + \underline{h}^T g(\underline{x}') + \frac{1}{2}\underline{h}^T G(\underline{x}')\underline{h} + \dots\dots$$

Proposition (1): (first order necessary condition)

Let S be a subset of $\Re^n$, and let $f\in \mathcal{C}^{(1)}$ be a function on S . If $\underline{x}^*$ is a relative any minimum point of f over $\underline{p}$, then for any $f \in \Re^n$, that is a feasible direction at $\underline{x}^*$, we have $\nabla f(\underline{x}^*)\underline{p}^T \geq 0$.

Corollary(1) :

Let S be a subset of $\Re^n$, let $f \in \mathcal{C}^{(1)}$, be a function on $\Re^n$. If $\underline{x}^*$ is a relative minimum point of f over $\Re^n$ and if $\underline{x}^*$ is an interior point of $\Re^n$, then

$\nabla f(\underline{x}^*) = \underline{0}$.

3-Descent directions at a point:

Consider the Taylor expansion of $f(\underline{x})$ about $\underline{x}'$ up to the first order term

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

$f\left(\underline{x}' + \alpha \underline{p}\right) = f(\underline{x}') + \alpha \underline{p}^T g\left(\underline{x}' + \alpha\theta \underline{p}\right).\alpha > 0$.

Where,$0 < \theta < 1$,

Since $f(\underline{x})$ is smooth enough (i.e, all the pertial derivatives are continuous), then $\underline{p}^T g(\underline{x})$ is $\underline{p}^T$ $g(\underline{x}) < 0$ $\forall \underline{x}$,then sufficiently close to $\underline{x}'$ (by continuity). Thus if α is taken sufficiently small,

$\underline{p}^T g\left(\underline{x}' + \alpha\theta \underline{p}\right) < 0$.

more precisely $\exists \overline{\alpha} > 0$ such that

$$\underline{p}^T g\left(\underline{x}' + \alpha\theta \underline{p}\right) < 0 \quad \forall \alpha \in [0, \overline{\alpha}] \text{ .Thus}$$
$$f\left(\underline{x}' + \alpha \underline{p}\right) < f(\underline{x}') .$$

We notice that if $\underline{p}^T g' < 0$, then the value of f decreases (locally if we move in the direction $\underline{p}$) [7].

Such a direction $\underline{p}$ is called a descent direction at $\underline{x}'$, and it chamcterized by :

$\underline{p}^T g' < 0$.

An example of a descent direction at $\underline{x}'$ is $\underline{p} = -g'$since $-g'^T g' < 0$ provided $\underline{g}' \neq 0$.

Def .(7): (Definite and semi definite matrices)

Let C be symmetric matrix we say that C is positive definite if $\underline{x}^T C\underline{x} > 0$ $\forall \underline{x} \in \Re^n$,$\underline{x} \neq \underline{0}$, C is called positive semi definite if $\underline{x}^T C\underline{x} \geq 0$ for $\underline{x} \in \Re^n$.

Def. (8): (Coincide with def . (7))

The point $\underline{x}^* \in \Re^n$ is called an unconstrained local minimizer of $f(\underline{x})$,if $f(\underline{x}) \geq f(\underline{x}^*)$ in some neighbourhood of $\underline{x}^*$.

In other words there exists no descent direction at $\underline{x}^*$ or equivalently.

$f\left(\underline{x}^* + \alpha \underline{p}\right) \geq f(\underline{x}^*)$ $\forall \underline{p} \in \Re^n$.

and sufficiently small values of α .

Now let $\underline{x}^*$ be a minimum of $f(\underline{x})$, let x be the solution of :

$\min_{\underline{x} \in \Re^n} f(\underline{x})$.

Take $\underline{p}$ to be any direction . Expand f about $\underline{x}^*$ along the direction $\underline{p}$.

$f\left(\underline{x}^* + \alpha \underline{p}\right) = f^* + \alpha \underline{p}^T g\left(\underline{x}^* + \alpha\theta \underline{p}\right)$.

(here is not restricted in sign) .

We claim that $\underline{x}^*$ must be zero . For if it is not $\underline{0}$, then $\underline{p}^T g^*$ can be positive or negative if $\underline{p}$ being not orthogonal to g^* .

Without loss of generality assume that $\underline{p}^T g^* < 0$ I.et $\underline{p}$ is taken to be a descent direction . Taking $\alpha > 0$ and the fact that $\underline{p}^T g(\underline{x})$ is continuous we conclude that (for sufficiently smaller α)

$f^* > f\left(x^* + \alpha \underline{p}\right)$, contradicting the minimality of f at $\underline{x}^*$ .Thus

$\underline{g}^* = \underline{0}$ is a necessary condition for $\underline{x}^*$ to be a local minimizer .

This condition is termed the first order necessary condition .

Now carrying this first order necessary condition and expand f about x^* up to the second orber term , we have :

$f\left(\underline{x}^* + \alpha \underline{p}\right) = f(\underline{x}^*) + \frac{1}{2}\alpha^2 \underline{p}^T \left[G\left(\underline{x}^* + \alpha\theta \underline{p}\right)\right] \underline{p}$.

We claim that G^* must be positive semi-definite .

For if not , there must exist a direction $\underline{p}$ (i.e $\underline{p} \in \Re^n$) for which $\underline{p}^T G^* \underline{p} < 0$. The continuity of $G(\underline{x})$ ensures that there is $\overline{\alpha} > 0$ such that

$\underline{p}^T \left(G\left(\underline{x}^* + \alpha\theta \underline{p}\right)\right) \underline{p} < 0, \forall \alpha < \overline{\alpha}$.

Thus $f\left(\underline{x}^* + \alpha \underline{p}\right) < f^*$.

Contradicting the minimiality of f at $\underline{x}^*$. Thus G^* must be positive semi definite for $\underline{x}^*$ to be a minimizer . Thus we proved :

Theorem (1):

Let $\underline{x}^*$ be a a local minimizer for

$\min_{\underline{x} \in \Re^n} f(\underline{x})$.

Then the following two conditions are necessary :

$(A1) g^* = 0$.

$(A2) G^*$is positive semi definite .

The conditions $(A1)$ and $(A2)$ are only necessay but not sufficient for $\underline{x}^*$ to be local minimizer as the following example reflects .

Example (1):

$\min_{\underline{x} \in \Re^n} f\left(x^{(1)}, x^{(2)}\right) = x^{(1)^4} - x^{(2)^4}$.

$\underline{x}^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

$g^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

$G^* = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$.

but a long $\underline{p} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ the function decreases from $f^* = 0$.

Condition (A2) is termed the second order necessary condition for $\underline{x}^*$ to be a locel minimizer . It is possible to obtain a sufficient condition using second order partial derivatives as the following theorem shows :

Theorem (2):

Let x^* be any point in $\Re^n$ satisfying the conditions :

(B1)$\underline{g}^* = \underline{0}$.

is positive definite , then $\underline{x}^*$ is a local solution of B2)G^*(

$$\min_{\underline{x} \in R^n} f(\underline{x}) .$$

Proof :

Let $\underline{p}$ be any direction .

Expand f at $\underline{x}^*$ along $\underline{p}$.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

$f\left(\underline{x}^* + \alpha \underline{p}\right) = f^* + {\alpha^2}/{2}\ \underline{p}^T G\left(\underline{x}^* + \alpha\theta \underline{p}\right)\underline{p}$.
(UsingB1)
It is straight forward to prove that there exists $\overline{\alpha} > 0$ such that
$\underline{p}^T G\left(\underline{x}^* + \alpha\theta \underline{p}\right)\underline{p} > 0$.
(using continuity and positive definiteness of G).
Thus $\exists$ a neighbouhood of $\underline{x}^*$ for which $f^* < f(\underline{x}) \forall \underline{x}$ in that neighborhood .

Example (2):
Use the sufficient conditions to obtain the solution of:
$\boldsymbol{min}_{x^{(1)},x^{(2)}\in\Re^2} \boldsymbol{f}(\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)}) = \boldsymbol{3x^{(1)^2} - x^{(1)}x^{(2)} + 4x^{(2)^2} - x^{(1)} + x^{(2)} + 5}$

Solution:

$$\underline{g} = \begin{bmatrix} \boldsymbol{6x^{(1)} - x^{(2)} - 1} \\ \boldsymbol{-x^{(1)} + 8x^{(2)} + 1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}$$

$$\Rightarrow \quad x^{(1)} = 7 / 47$$

$$x^{(2)} = \frac{-5}{47}$$

$$x^* = \frac{1}{47}\begin{pmatrix} 7 \\ -5 \end{pmatrix}$$

$$G^* = \begin{bmatrix} 6 & -1 \\ -1 & 8 \end{bmatrix}$$

$$\left|G^* - \lambda I\right| = \begin{vmatrix} 6-\lambda & -1 \\ -1 & 8-\lambda \end{vmatrix}$$

$$= (6-\lambda)(8-\lambda) - 1$$
$$= 48 - 14\lambda + \lambda^2 - 1$$
$$= \lambda^2 - 14\lambda + 47$$
$$\therefore \lambda^2 - 14\lambda + 47 = 0$$

$$\lambda_{1,2} = \frac{14 \pm \sqrt{196-188}}{2} = \frac{14 \pm \sqrt{8}}{2} > 0$$

Thus G* is positive definite. Therefore

$\underline{x}^* = \frac{1}{47}\begin{pmatrix} 7 \\ -5 \end{pmatrix}$ is a local minimize (In fact it is the only one)

Section (2):

Introduction:

In this section we give a brief description to Newton' s method to minimize a general function $f(\underline{x})$. This method , although lacks practically , gives a solid ground to wide class of methods known as variable metric methods . Three detailed description is given to the

DFP method which belongs to the class of variable methods .This method constitutes the major part of this work . Section (4) gives a practical implementation to this method .

Newton' s Method :

All the local minimum $\underline{x}^*$ of a continuous differential function f satisfy the necessary condition

$g(\underline{x}^*) = \nabla f(\underline{x}^*) = \underline{0}$ (1)

Equation (1) represents a set of non-linear equations which must be solved to obtain $\underline{x}^*$. One approach to the minimization of $f(\underline{x}^*)$ is therefore to seek the solutions of equations(1) by including a provision to ensure that the solution found does indeed correspond to a local minimum.

The oldest method for solving a set of non-linear equations is the Newton's method[2] . We shall consider this method briefly and then turn to a class of methods known as quasi-Newton's methods. Since they can be regarded as approximation to Newton's method in some sense.
To solve the system of non-linear equations (1),
we first linearize the set of equations about some point $\underline{x}^{(i)}$ (which can be considered as the ith approximation to the minimum point $\underline{x}^*$). Thus if
$\underline{x}^*$ can be written as:

$\underline{x}^* = \underline{x}^{(i)} + \underline{p}$, the Taylor's expansion of $g(\underline{x}^*)$ gives :

2) $(g(\underline{x}^*) = g\left(\underline{x}^{(i)} + \underline{p}\right) = g(\underline{x}^{(i)}) + G^{(i)}\underline{p} + \ldots\ldots$

($G^{(i)}$ is the Hessian matrix) . By neglecting the high order terms in equation (2) and setting $g(\underline{x}^*) = \underline{0}$ (3) we obtain : $g^{(i)} + G^{(i)}\underline{p} = \underline{0}$

and $G^{(i)} = G(\underline{x}^{(i)})$ where: $g^{(i)} = G^{(i)}\underline{p}$

is the symmetric Hessian matrix of f evaluated at $\underline{x}^{(i)}$

If $G^{(i)}$is non-singular, the set of linear equation (3) can be easily solved for the vector $\underline{p}$, and the derived minimum can be obtained as
$\underline{x}^* = \underline{x}^{(i)} + \underline{p}$.

Thus the equation (3) gives :

$\because \underline{p} = G^{(i)^{-1}}\underline{g}^{(i)}$. However ,in general , the higher order terms in equation (2) are not neglible and hence an iterative procedure has to be used to find the improved approximations. The scheme is thus given by:

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

$$\underline{x}^{(i+1)} = \underline{x}^{(i)} + \underline{p} = \underline{x}^{(i)} + G^{(i)^{-1}}\underline{g}^{(i)} \qquad (4)$$

$$x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(i+1)}$$

The sequence of points can be shown to converge to the actual solution $\underline{x}^*$ from any initial point

$\underline{x}^{(1)}$ sufficiently close to the solution $\underline{x}^*$ (theorem(2))

Provided the $G^{(i)}$ for every i is non-singular .

Very restrictive and the method frequently fails to converge . If $f(\underline{x})$ is quadratic ,

$\underline{x}^* = A^{-1}b$. (5)

$\underline{x}^{(i+1)} = \underline{x}^{(i)} - A^{-1}(A\underline{x}^{(i)} - \underline{b})$ $.\underline{x}^{(i+1)} = \underline{x}^* = A^{-1}\underline{b}$. (6)

as :

$$.= \frac{2}{\left(x^{(1)^2}+x^{(2)^2}+2\right)^3}\begin{bmatrix}\left(-3x^{(1)^2}+x^{(1)^2}+2\right) & -4x^{(1)}x^{(2)} \\ -4x^{(1)}x^{(2)} & \left(-3x^{(2)^2}+x^{(1)^2}+2\right)\end{bmatrix} \cdot \text{At } \underline{x}^{(1)} = \begin{bmatrix}4\\0\end{bmatrix} \cdot \underline{g}^{(1)} = \begin{bmatrix}0.024\\0\end{bmatrix}$$

$$.G^{(1)} = \begin{bmatrix}-0.04580 & 0\\ 0 & 0.0617\end{bmatrix} \cdot G^{-1} = \frac{1}{(-0.0000975)}\begin{bmatrix}0.000617 & 0\\ 0 & -0.01580\end{bmatrix} .= \begin{bmatrix}-63.4 & 0\\ 0 & 16.0\end{bmatrix}.$$

Hence equation (2.4) gives

$$\underline{x}^{(2)} = \underline{x}^{(1)} - G^{-1}\underline{g}^{(!)} = \begin{bmatrix}5.57\\0\end{bmatrix} .$$

If we compare the values of f at $\underline{x}^{(1)}$ and $\underline{x}^{(2)}$, we find that ,

$$f^{(2)} = -0.0556, \qquad and f^{(2)} = -0.0303 .$$

Thus $f^{(2)}$ is greater than $f^{(1)}$ and hence the method fails to make progress . (The true minimum point is $\underline{x}^* = \begin{bmatrix}0\\0\end{bmatrix}$, with $f^* = -0.5$).

However if we use equation (7) instead of (5) the method can be made to converge to the minimize point.

$$\underline{x}^{(i+1)} = \underline{x}^{(i)} + \lambda^{*(i)}\underline{p}^{(i)} = \underline{x}^{(i)} - G^{(i)^{-1}}\underline{g}^{(i)} . \qquad (7)$$

Example (3):

$\min_{\underline{x}\in R^n} f(x^{(1)}, x^{(2)}) = \frac{-1}{x^{(1)^2}+x^{(2)^2}+2}$.

From the starting point

$\underline{x}^{(1)} = \begin{bmatrix}4\\0\end{bmatrix}$.

Solution :

The gradient g and the Hessian matrix G of f are given by :

$$\underline{g} = \left(\frac{\partial f}{\partial \underline{x}}\right) = \frac{2}{\left(x^{(1)^2}+x^{(2)^2}+2\right)^2}\begin{bmatrix}x^{(1)}\\x^{(2)}\end{bmatrix}.$$

$$G = \begin{bmatrix}\frac{\partial^2 f}{\partial x^{(1)^2}} & \frac{\partial^2 f}{\partial x^{(1)}\partial x^{(2)}}\\ \frac{\partial^2 f}{\partial x^{(1)}\partial x^{(2)}} & \frac{\partial^2 f}{\partial x^{(2)^2}}\end{bmatrix}$$

III. QUASI-NEWTON'S METHOD:

In optimization, quasi-Newton methods (also known as variable metric methods) are algorithms for finding local maxima and minima of functions. Quasi-Newton methods are based on Newton's method to find the stationary point of a function, where the gradient is 0. Newton's method assumes that the function can be locally approximated as a quadratic in the region around the optimum, and use the first and second derivatives (gradient and Hessian) to find the stationary point [8].

In Quasi-Newton methods the Hessian matrix of second derivatives of the function to be minimized does not need to be computed. The Hessian is updated by analyzing successive gradient vectors instead. Quasi-Newton methods are a generalization of the secant method to find the root of the first derivative for multidimensional problems. In multi-dimensions the secant equation is under-determined, and quasi-Newton methods differ in how they constrain the solution, typically by adding a simple low-rank update to the current estimate of the Hessian.

IV. DESCRIPTION OF THE METHOD

As in Newton's method, one uses a second order approximation to find the minimum of a function *f*(*x*). The Taylor series of *f*(*x*) around an iterate is:

$$f(\underline{x}_k + \Delta\underline{x}) \approx f(\underline{x}_k) + \nabla f(\underline{x}_k)^T\Delta\underline{x} + \frac{1}{2}\Delta\underline{x}^T\boldsymbol{B}\Delta\underline{x}$$

where (∇f) is the gradient and ***B*** an approximation to the Hessian matrix. The gradient of this approximation (with respect to $\Delta\underline{x}$) is:

$$\nabla f(\underline{x}_k + \Delta\underline{x}) \approx \nabla f(\underline{x}_k) + \boldsymbol{B}\Delta\underline{x}_k$$

and setting this gradient to zero provides the Newton step:

$$\Delta\underline{x} = -\boldsymbol{B}^{-1}\nabla f(\underline{x}_k)$$

The Hessian approximation ***H*** is chosen to satisfy:

$$\nabla f(\underline{x}_k + \Delta\underline{x}) = \nabla f(\underline{x}_k) + \boldsymbol{B}\Delta\underline{x}$$

which is called the secant equation(the Taylor series of the gradient itself). In more than one dimension ***B*** is under determined. In one dimension, solving for ***B***and applying the Newton's step with the updated value is equivalent to the secant method. Various methods are used to find the

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

solution to the secant equation that is symmetric ($\boldsymbol{B^T = B}$) and closest to the current approximate value $\boldsymbol{B_k}$ according to some metric $\mathbf{min_B}\|\boldsymbol{B - B_k}\|$ An approximate initial value of $\boldsymbol{B_0 = I}$ is often sufficient to achieve rapid convergence. The unknown $\underline{x}_k$ is updated applying the Newton's step calculated using the current approximate Hessian matrix $\boldsymbol{B_k}$

$\Delta \underline{x}_k = -\alpha_k \boldsymbol{B_k}^{-1} \nabla f(\underline{x}_k)$with α chosen to satisfy the Wolfe conditions:

$\underline{x}_{k+1} = \underline{x}_k + \Delta \underline{x}_k$

The gradient computed at the new point$\nabla f(\underline{x}_{k+1})$, and

$\boldsymbol{y_k} = \nabla f(\underline{x}_{k+1}) - \nabla f(\underline{x}_k)$is used to update the approximate Hessian$\boldsymbol{B_{k+1}}$, or directly its inverse $\boldsymbol{H_{k+1}} = \boldsymbol{B^{-1}}_{\boldsymbol{k+1}}$using the Sherman-Morrison formula.

Gradient descent:

The analytical method called "Steepest descent".

Gradient descent is a first-order optimization algorithm. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point. If instead one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient descent.

Gradient descent is also known as steepest descent, or the method of steepest descent. When known as the latter, gradient descent should not be confused with the method of steepest descent for approximating integrals.

Gradient descent is based on the observation that if the multivariable function $f(\underline{x})$is defined and differentiable in a neighborhood of a point $\boldsymbol{\underline{a}}$, then $f(\underline{x})$decreases fastest if one goes from $\boldsymbol{\underline{a}}$ in the direction of the negative gradient of f at $\boldsymbol{a}$,$f(\boldsymbol{\underline{a}})$. It follows that, if

$$\underline{b} = \underline{a} - \gamma \nabla f(\underline{a})$$

for γ > 0 a small enough number, then$f(\underline{a}) > f(\underline{b})$. With this observation in mind, one starts with a guess $\underline{x}_0$for a local minimum of f, and considers the sequence $\underline{x}_0, \underline{x}_1, \underline{x}_2, \ldots$such that

$$\underline{x}_{n+1} = \underline{x}_n - \gamma \nabla f(\underline{x}_n), n \geq 0$$

We have:

$$f(\underline{x}_0) \geq f(\underline{x}_1) \geq f(\underline{x}_2), \ldots$$

so hopefully the sequence $(\underline{x}_n)$converges to the desired local minimum. Note that the value of the step size γ is allowed to change at every iteration.

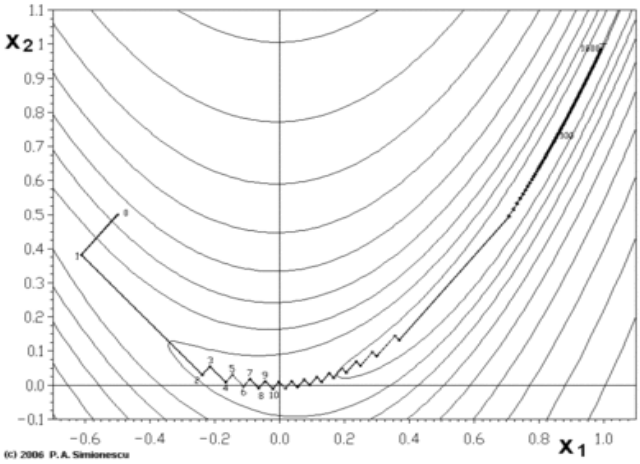
This process is illustrated in the picture to the right. Here f is assumed to be defined on the plane, and that its graph has a bowl shape. The blue curves are the contour lines, that is, the regions on which the value of F is constant. A red arrow originating at a point shows the direction of the negative gradient at that point. Note that the (negative) gradient at a point is orthogonal to the contour line going through that point. We see that gradient descent leads us to the bottom of the bowl, that is, to the point where the value of the function f is minimal.

EXAMPLES:

Gradient descent has problems with pathological functions such as the Rosen rock function shown here:

$$f(x_1, x_2) = (1 - {x_1}^2) + 100(x_2 - {x^2}_1)^2$$

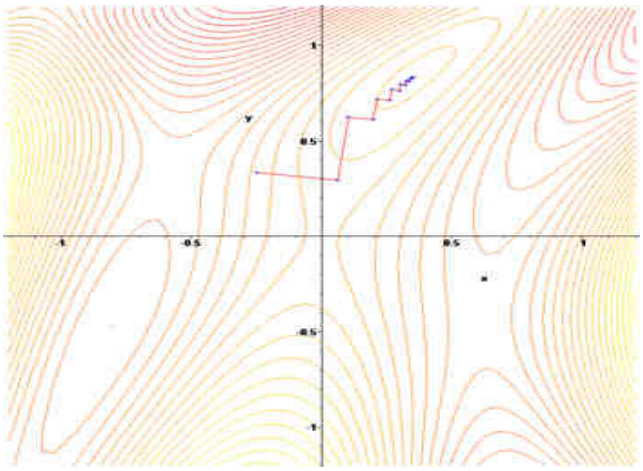
The Rosen rock function has a narrow curved valley which contains the minimum. The bottom of the valley is very flat. Because of the curved flat valley the optimization is zigzagging slowly with small step sizes towards the minimum.

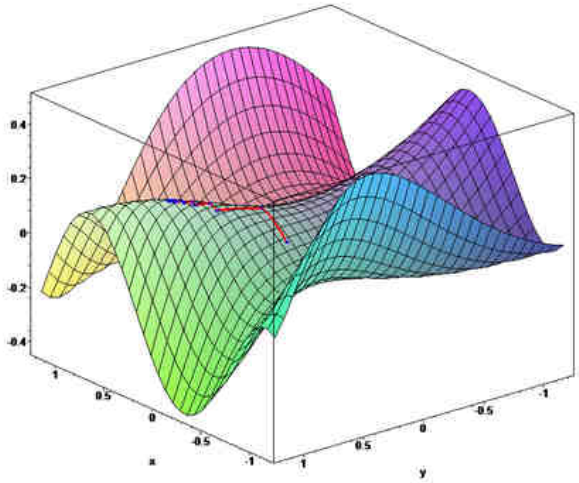


The "Zigzagging" nature of the method is also evident below, where the gradient ascent method is applied to.

$$F(x, y) = \sin\left(\frac{1}{2}x^2 - \frac{1}{4}y^2 + 3\right)\cos(2x + 1 - e^y)$$

F





LIMITATIONS:

For some of the above examples, gradient descent is

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

relatively slow close to the minimum: Technically, its asymptotic rate of convergence is inferior to other methods. For poorly conditioned convex problems, gradient descent increasingly 'zigzags' as the gradients point nearly orthogonally to the shortest direction to a minimum point. For more details, see Comments below:

For non-differentiable functions, gradient methods are ill-defined. For locally Lipchitz problems and especially for convex minimization problems, bundle methods of descent are well-defined. Non-descent methods, like sub gradient projection methods, may also be used.

V. SOLUTION OF A LINEAR SYSTEM

Gradient descent can be used to solve a system of linear equations, reformulated as a quadratic minimization problem, e.g., using linear least squares. Solution of

$$A\underline{x} - \underline{b} = \underline{0}$$

in the sense of linear least squares is defined as minimizing the function

$$F(\underline{x}) = \| A\underline{x} - \underline{b} \|^2$$

In traditional linear least squares for real A and $\underline{b}$ the Euclidean norm is used, in which case:

$$\nabla F(x) = 2A^T(A\underline{x} - \underline{b})$$

In the case that A is real, square, symmetric and positive definite, a different popular choice of the norm is

$$\| a \|^2 = a^T A^{-1} a$$

which produces a different equation with a better condition number:

$$\nabla F(\underline{x}) = 2(A\underline{x} - \underline{b})$$

In either case, the line search minimization, finding the locally optimal step size γ on every iteration, can be performed analytically, and explicit formulas for the locally optimal γ are known.

SOLUTION OF A NON-LINEAR SYSTEM:

Gradient descent can also be used to solve a system of nonlinear equations. Below is an example that shows how to use the gradient descent to solve for three unknown variables, x_1, x_2, and x_3. This example shows one iteration of the gradient descent.[7]

Consider a nonlinear system of equations:

$$3x_1 - \cos(x_2 x_3) - \frac{3}{2} = 0$$
$$4x^2{}_1 - 625x^2{}_2 + x_2 - 1 = 0$$
$$\exp(-x_1 x_2) + 20x_3 + \frac{10\pi - 3}{3} = 0$$

suppose we have the function:

$$3x_1 - \cos(x_2 x_3) - \frac{3}{2}$$
$$4x^2{}_1 - 625x^2{}_2 + x_2 - 1$$
$$\exp(-x_1 x_2) + 20x_3 + \frac{10\pi - 3}{3}$$

Where $\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

and the objective function : $F(\underline{x}) = \frac{1}{2} G^T(\underline{x}) G(\underline{x})$

$$= \frac{1}{2}\left(\left(3x_1 - \cos(x_2 x_3) - \frac{3}{2}\right)^2 + (4x_1^2 - 625x_2^2 + 2x_2 - 1)^2\right)$$

$$+\left(\left(exp(-x_1 x_2) + 20x_3 + \frac{10\pi - 3}{3}\right)^2\right) .$$

With initial guess:

$$\underline{x}^{(0)} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

We know that:

$$x^{(1)} = x^{(0)} - \gamma_0 \nabla F(x^{(0)})$$

Where:

$$\nabla F(x^{(0)}) = J_G(x^{(0)})^T * G(x^{(0)})$$

The Jacobian matrix $J_G(x^{(0)})$

$$J_G = \begin{bmatrix} 3 & \sin(x_2 x_3)\, x_3 & \sin(x_2 x_3)\, x_2 \\ 8x_1 & -1250x_2 + 2 & 0 \\ -x_2 \exp(-x_1 x_2) & -x_1 \exp(-x_1 x_2) & 20 \end{bmatrix}$$

Then evaluating these terms at

$$J_G(x^{(0)}) = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 20 \end{bmatrix}$$

and

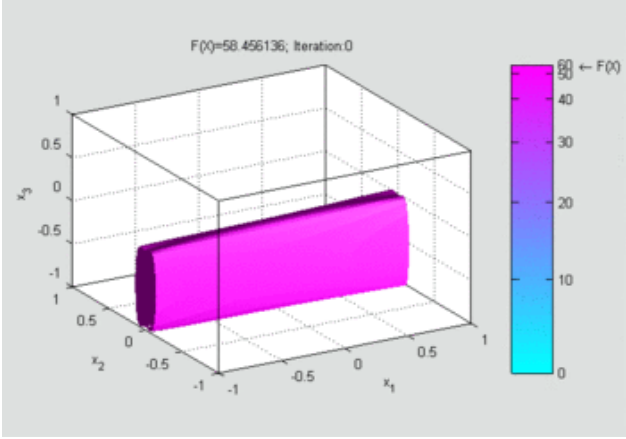
$$G(x^{(0)}) = \begin{bmatrix} -2.5 \\ -1 \\ 10.472 \end{bmatrix}$$

So that

$$x^{(1)} = x^{(0)} - \gamma_0 \nabla F(x^{(0)})$$

and $x^{(1)} = x^{(0)} - \gamma_0 \begin{bmatrix} -7.5 \\ -2 \\ 209.44 \end{bmatrix}$

$$F(x^{(0)}) = 0.5(-2.5)^2 + (-1)^2 + (10.472)^2 = 58.456$$



An animation showing the first 83 iterations of gradient decent applied to this example. Surfaces are isosurfaces of $F(x^n)$at current guessx^n, and arrows show the direction of descent. Due to a small and constant step size, the convergence is slow.

Now a suitable γ_0 must be found such that$F(x^{(1)}) \leq F(x^{(0)})$. This can be done with any of a variety of line search algorithms. One might also simply guess γ_0 = 0.001 which gives

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

$$x^{(1)} = \begin{bmatrix} 0.0075 \\ 0.002 \\ -0.20944 \end{bmatrix}$$

evaluating at this value,

$$F(x^{(1)}) = 0.5((-2.48)^2 + (-1.00)^2 + (6.28)^2) = 23.306$$

The decrease from $F(x^{(0)}) = 58.456$to the next step's value of $F(x^{(1)}) = 23.306$ is a sizable decrease in the objective function. Further steps would reduce its value until a solution to the system was found.

VI. COMMENTS

Gradient descent works in spaces of any number of dimensions, even in infinite-dimensional ones. In the latter case the search space is typically a function space, and one calculates the Gateaux derivative of the functional to be minimized to determine the descent direction.

The gradient descent can take many iterations to compute a local minimum with a required accuracy, if the curvature in different directions is very different for the given function. For such functions, preconditioning, which changes the geometry of the space to shape the function level sets like concentric circles, cures the slow convergence. Constructing and applying preconditioning can be computationally expensive, however.

The gradient descent can be combined with a line search, finding the locally optimal step size γ on every iteration. Performing the line search can be time-consuming. Conversely, using a fixed small γ can yield poor convergence.

Methods based on Newton's method and inversion of the Hessian using conjugate gradient techniques can be better alternatives.[3] Generally, such methods converge in fewer iterations, but the cost of each iteration is higher. An example is the BFGS method which consists in calculating on every step a matrix by which the gradient vector is multiplied to go into a "better" direction, combined with a more sophisticated line search algorithm, to find the "best" value of γ. For extremely large problems, where the computer memory issues dominate, a limited-memory method such as L-BFGS should be used instead of BFGS or the steepest descent.

Gradient descent can be viewed as Euler's method for solving ordinary differential equations $x'(t) = \nabla f(x(t))$of a gradient flowThe gradient descent algorithm is applied to find a local minimum of the function $f(x)=x^4-3x^3+2$, with derivative $f'(x)=4x^3-9x^2$. Here is an implementation in the Python scripting language.

Davidon- Fletcher and Powel method Tested On Quadratic Functions:

Introduction:

These methods are the best general purpose unconstrained optimization techniques making use of the derivatives , that is carefully available ,at the iterative procedure of this method can be shown in the following algorithm:

1) Start with an initial point $\underline{x}^{(1)}$ and an $(n \times n)$ positive definite symmetric matrix $H^{(1)}$ is taken as the identity matrix I . Set iteration number as i=1
2) Compute the gradient of the function $\nabla f^{(i)}$ at the point $\underline{x}^{(i)}$ and set $\underline{p}^{(i)} = -H^{(i)}\underline{g}^{(i)}$ (1)

(for the first iteration the search direction $\underline{p}^{(1)} = -\underline{g}^{(1)}$

3) Find the optimal step length $\lambda^{*(i)}$ =1 in the direction $\underline{p}^{(i)}$ and set

$$\underline{x}^{(i+1)} = \underline{x}^{(i)} + \lambda^{*(i)}\underline{p}^{(i)} \quad (2)$$

$$\lambda^{*(i)}\underline{p}^{(i)} = \underline{x}^{(i+1)} - \underline{x}^{(i)} \quad (3)$$

$$\underline{q}^{(i)} = \underline{g}^{(i+1)} - \underline{g}^{(i)} \quad (4)$$

4) Test the new point $\underline{x}^{(i+1)}$ for optimality . If $\underline{x}^{(i+1)}$ is optimal terminate the iterative process .Otherwise go to step (5).
5) Update the Hessian matrix as :

$$H^{(i+1)} = H^{(i)} + M^{(i)} + N^{(i)} \quad (5)$$

where $$M^{(i)} = \frac{\lambda^{*(i)}\underline{p}^{(i)}\underline{p}^{(i)^T}}{\underline{p}^{(i)^T}\underline{q}^{(i)}} \quad (6)$$

$$N^{(i)} = -\frac{(H^{(i)}\underline{q}^{(i)})(H^{(i)}\underline{q}^{(i)})^T}{\underline{q}^{(i)^T}H^{(i)}\underline{q}^{(i)}} \quad (7)$$

The following example describes the steps of the procedure

Example (1):

$$\min_{\underline{x}\in\Re^2} f(x^{(1)}, x^{(2)}) = x^{(1)} - x^{(2)} + 2x^{(1)^2} + 2x^{(1)}x^{(2)} + x^{(2)^2}$$

With initial point $\underline{x}^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

Solution :

are Known as inverse update formulas slice the inverse of the of *f*.

The DFP and the BFGS[1] formulas belong to a family of rank 2 updates known as Huang's family of updates which can be expressed for updating the inverse of the' DAVIDON- FLETCHER and POWELL METHOD .

$$[\boldsymbol{B}_{i+1}] = \boldsymbol{P}_i([\boldsymbol{B}_i] - \frac{[\boldsymbol{B}_i]\boldsymbol{g}_i\boldsymbol{g}^{\boldsymbol{T}}{}_i[\boldsymbol{B}_i]}{\boldsymbol{g}^{\boldsymbol{T}}{}_i[\boldsymbol{B}_i]\boldsymbol{g}_i} + \boldsymbol{\theta}_i\boldsymbol{\gamma}_i\boldsymbol{\gamma}^{\boldsymbol{T}}{}_i + \frac{\boldsymbol{d}_i\boldsymbol{d}^{\boldsymbol{T}}{}_i}{\boldsymbol{d}^{\boldsymbol{T}}{}_i\boldsymbol{g}_i})$$

,where

$$\boldsymbol{\gamma}_i = (\boldsymbol{g}^{\boldsymbol{T}}{}_i[\boldsymbol{B}_i]\boldsymbol{g}_i)^{\frac{1}{2}}(\frac{\boldsymbol{d}_i}{\boldsymbol{d}^{\boldsymbol{T}}{}_i\boldsymbol{g}_i} - \frac{[\boldsymbol{B}_i]\boldsymbol{g}_i}{\boldsymbol{g}^{\boldsymbol{T}}{}_i[\boldsymbol{B}_i]\boldsymbol{g}_i}$$

P_i and $\theta_i$are constant parameters,

It has beer, shown that Eq. maintains the symmetry and

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

positive definiteness of $[B_{i+1}]if\ [B_i]$ is symmetric and positive definite. Different choices of P_i and θ_i in Eq. lead to different Algren algorithms. For example, when $P_i = 1\ and\ \theta_i = 1$ Eq. yields the BFGS formula, [9].

Example (2):

Show that the DFP method is a conjugate method.

Solution: consider the quadratic function

$$f(x) = \frac{1}{2}x^T Ax + B^T x + C \qquad (1)$$

For which the gradient is given by

$$\nabla f = Ax + b \qquad (2)$$

$$g_i = \nabla f_{i+1} - \nabla f_i = A(x_{i+1} - x_i) \qquad (3)$$

Since:

$$x_{i+1} = x_i - \lambda^* S_i \qquad (4)$$

Equation (3) becomes:

$$g_i = \lambda^* i\ AS_i \qquad (5)$$

Pre -multiplication of Eq. (5) by $[B_{i+1}]$ leads to :

$$[B_{i+1}][A]S_i = \frac{1}{\lambda^*}[B_i] + [M_i] + [N_i]g_i \qquad (6)$$

Eq. (4) and (5) yield:

$$[M_i]g_i = \frac{S_i S_i^T g_i}{S_i^T g_i} = \lambda_i^* S_i \qquad (7)$$

Eq .(6) can be used to obtain :

$$[N_i]g_i = \frac{([B_i]g_i)(g_i^T [B_i]^T g_i)}{g_i^T [B_i] g_i} = -[B_i]g_1 \ldots\ldots\ldots\ldots\ldots\ldots (8)$$

Since [Bi] is symmetric . By subsisting Eqs (7) and (8) into. (6), we obtain:

$$[B_{i+1}][A]S_i = \frac{1}{\lambda_i^*}([B_i]g_1 + \lambda_i^* S_i - [B_i]g_i) = S_i \ldots\ldots\ldots\ldots (9)$$

The quantity $S_i^T +_1 [A]S_i$ can be written as:

$$S_i^T + [A]S_i = -([B_{i+1}]\nabla f_{i+1})^T [A]S_i$$
$$= -\nabla f_{i+1}^T [B_{i+1}][A]S_i = -\nabla f_{i+1}^T S_i = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (10)$$

Since λ_i^* is the minimizing step in the direction S_i. Equation (10) proves that the successive directions generated in the DFP method are conjugate and hence the method is conjugate gradient method.

Section(4):

Implementation of the Davidon –Fletcher –Powel method:

In this section we write a Matlab program and test it on some quadratic problems.

Applications and Results:

1) $\min_{\underline{x}\in\Re^2} f(\underline{x}) = x^{(1)} - x^{(2)} + x^{(1)^2} + 2x^{(1)}x^{(2)} + x^{(2)^2}$ $\underline{x}^{(0)} = \begin{bmatrix}0\\0\end{bmatrix}$

Iteration no.0

alpha = 1.0000

f = -1.00000

$\underline{x}^{(1)} = \begin{bmatrix}-1.000\\1.000\end{bmatrix}$

$g(\underline{x}^{(1)}) = \begin{bmatrix}-1.000\\1.000\end{bmatrix}$

Iteration no.1

Alpha = 0.5000 f = -1.25000 $\underline{x}^* = \underline{x}^{(1)} = \begin{bmatrix}-1.000\\1.5000\end{bmatrix}$

$g(\underline{x}^*) = \begin{bmatrix}0.000\\0.000\end{bmatrix}$

2) $\min_{\underline{x}\in\Re^2} f(\underline{x}) = 2x^{(1)^2} + x^{(2)^2}$ $\underline{x}^{(0)} = \begin{bmatrix}1\\0.5\end{bmatrix}$

Iteration no.0

Alpha = 0.2576 f = 0.060606

$\underline{x}^{(1)} = \begin{bmatrix}-0.0303\\1.2424\end{bmatrix}$ $g(\underline{x}^{(1)}) = \begin{bmatrix}-0.1212\\0.4848\end{bmatrix}$

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

alpha = 0.4924 f = 0.0000

$$\underline{x}^* = \underline{x}^{(2)} = \begin{bmatrix} 0.0000 \\ 0.0000 \end{bmatrix} \; g(\underline{x}^{(2)}) = \begin{bmatrix} 0.0000 \\ 0.0000 \end{bmatrix}$$

REFERENCES

1. Brian D. Bundy 1985 Basic Optimization Methods.
2. Boyd and L.Vandenberghe 2004, Convex Optimization Cambridge University.
3. David G. Luenberger Linear and nonlinear Programming 2nd Edition.
4. Masanao Aoki 1971 Introduction to Optimization Techniques Fundamental and Applications of non-linear Programming .
5. Mohsen Hassan Abdulla 1997 An extension to the Dantzig-wolfe Method .
6. Phillip E. Gill Waller Murray and Margret H.Wright 1981 Practical Optimization .
7. S.S. Rao 1977 Optimization Theory and Applications 2nd Edition. .
8. Stephen G.Nash and Ariela Sofer (1996).Linear and Nonlinear Programming.
9. Singiresu S. Rao 1996Engineering optimization: theory and practice
10. Fletcher R. 1987 Practical Methods of Optimization Second Edition.
11. Forsgen J. Osbome 2014 Kuhn Tucker.
12. Forsgren , A; Gill,P.E; and Wright ,M.H." Interior Methods for Non linear optimization ". SIAm Rev 525-597n(2002).
13. Tokhomirov, V.M." The Evaluation of Methods of Convex Optimization '. Amer .Math . Monthly 103 65-71 (1996).

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.