Quest Journals
Journal of Research in Applied Mathematics
Volume 2~ Issue 5 (2015) pp: 01-06
ISSN(Online) : 2394-0743 ISSN (Print):2394-0735
www.questjournals.org

Research Paper

On Derivation of Numerical Implementaion of Penalty Function Method Imbedded In Conjugate Gradient Method Algorithm

T. E. OLAOSEBIKAN, K. J. ADEBAYO AND J.S. APANAPUDOR
Department of Mathematical Sciences, Ekiti State University, Ado Ekiti, Nigeria.
Department of mathematics and computer science delta state university Abraka

Received 31 January, 2015; **A**ccepted 25June, 2015 © The author(s) 2015. Published with open access at**www.questjournals.org**

ABSTRACT:-In this paper, we derive the mathematical implementation of the Penalty Function Method (PFM) imbedded in Conjugate Gradient Method (CGM), which enables Conjugate Gradient Method (CGM) to be employed for solving constrained optimization problems of either equality, inequality constraint or both. In the past, Penalty Function Method has been used extensively to solve constrained optimization problems. However, with some special features in CGM which makes it unique in solving unconstrained optimization problems, we intend to capitalize on the strength of the CGM to solve constrained optimization problems on adding or subtracting one or two things to the CGM. This, then call for the derivation of the new algorithm to show the numerical implementation of the new method that is aimed at taking care of any constrained optimization problems, either with equality or inequality constraint. The authors of this paper desire that, with the construction of the algorithm and the derivation of the numerical implementation of the algorithm, one will bypass the difficulties undergone using only PFM to solve constrained optimization problems and its application can easily be implemented following the procedures. It is observed that this invariably improves the result of the Conjugate Gradient Method in solving this class of optimization problems.

Keywords:-Penalty Function Method, Constrained Optimization problem, Conjugate Gradient Method, Numerical implementation of Penalty Function Conjugate Gradient Method.

I. INTRODUCTION

The general optimization problem to be considered is of the form described by [1] and [2] as:

Optimize: $f(x)$ 1.1

Subject to: $h_i(x) = 0 \ i = 1, 2, \dots, m_1$ 1.2

$g_j(x) \geq 0 \ j = 1, 2, \dots, m_2$ 1.3

where $x \in R^n, h_i(x)$, an equality vector equations of dimension m_1, and $g_j(x)$ is an inequality vector of dimension m_2, such that the sum of the constraints m $= (m_1 + m_2)$. The functions $f(x),\ h_i(x)\ and\ g_j(x)$are differentiable functions. Methods for solving this model have been developed, tested and successfully applied to many important problems of scientific and economic interest. However, in spite of the proliferation of the methods, there is no universal method for solving all optimization problems which calls for application of ILMCGA to solve constrained optimization problems.

II. CONJUGATE GRADIENT METHOD

In 1952, Hestenes and Stiefel developed a Conjugate Gradient Method (CGM) algorithm for solving algebraic equations which was successfully applied to nonlinear equations with results reported by Fletcher and Reevesin 1964.

The CGM algorithm for iteratively locating the minimum x^* of $f(x)$ in $\mathcal{H}$ is described as follows:

Step 1: Guess the first element x_0 □ $\mathcal{H}$ and compute the remaining members of the sequence with the aid of the formulae in the steps 2 through 6.

Step 2: Compute the descent direction $p_0 = -g_0$ 1.4

Step 3: Set $x_{i+1=} x_i + \alpha_i p_i$; where $\alpha_i = \frac{\langle g_i, g_i \rangle_{\mathcal{H}}}{\langle p_i, Gp_i \rangle_{\mathcal{H}}}$ 1.5

Step 4: Compute $g_{i+1} = g_i + \alpha_i G p_i$ 1.6

*Corresponding Author: T. E. OLAOSEBIKAN
Department of Mathematical Sciences, Ekiti State University, Ado Ekiti, Nigeria.

Step 5: Set $p_{i+1} = -g_{i+1} + \beta_i p_i$; $\beta_i = \frac{\langle g_{i+1}, g_{i+1} \rangle_{\mathcal{H}}}{\langle g_i, g_i \rangle_{\mathcal{H}}}$ 1.7

Step 6: If $g_i = 0$ for some i, then, terminate the sequence; else set $i = i + 1$ and go to step 3.

In the iterative steps 2 through 6 above, p_i denotes the descent direction at ith step of the algorithm, α_i, is the step length of the descent sequence $\{x_i\}$ and g_i denotes the gradient of f at x_i. Steps 3, 4 and 5 of the algorithm reveal the crucial role of the linear operator G in determining the step length of the descent sequence and also in generating a conjugate direction of search.

Doctoral Thesis of [3] threw light on the theoretical applicability of the CGM, which was extended to optimal control problems by [4], [5] and [6]. Applicability of the CGM algorithm thus depends solely on the explicit knowledge of the linear operator, G.Generally, for optimization problems, G is readily determined and such enjoys the beauty of the CGM as a computational scheme since the CGM exhibits quadratic convergence and requires only a little more computation per iteration.

The Concept of Penalty Function Method

Considering the equality constrained optimization problem such as (1.1) and (1.2), where $f(X)$ and $h(X)$ are the objective function and the equality constraint respectively. The unconstrained optimization problem of (1.1) and (1.2) is:

$Minimize\ f(X) + \mu h^2(X)$ 1.8

$Subject\ to\ X \in \mathbb{R}^n$

whereμ is the PFM parameter which must be greater than zero, (i.e. $\mu > 0$).

Intuitively we see that an optimal solution to (1.8) must have $h^2(X)$ close to zero, else, a large penalty term $\mu h^2(X)$ will be incurred and (1.8) approaches infinity which makes it difficult to minimize (1.8). Bazaraa et al, (2006).

Now, considering (1.1) and (1.3), is not appropriate to write the unconstrained form of (1.1) and (1.3) as:

$Minimize\ f(X) + \mu g^2(X)$ 1.9

Since a PFM parameter, μ, will be incurred where $g(X) < 0$ or $g(X) > 0$; which means that a PFM parameter is added to the objective function whether X is inside or outside the feasible region. Needless to say, a penalty is desired only if the point is not feasible, that is, if $g(X) > 0$. A suitable unconstrained problem for (1.1) and (1.3) is therefore given as:

$Minimize\ f(X) + \mu maximum\{0, g(X)\}$ 1.10

$Subject\ to\ X \in \mathbb{R}^n$

It must be noted that if $g(X) \le 0$, then $maximum\{o, g(X)\} = 0$, and no penalty is incurred on the other hand but if $g(X) > 0$, then the $maximum\{o, g(X)\} > 0$, and the penalty term $\mu g(X)$ is realized. Now, it is however observe that at points $X$where $g(X) = 0$, the forgoing objective function might not be differentiable, even if g is differentiable.

III. PENALTY FUNCTION METHOD ALGORITHM

The use of Penalty Function Method (PFM) to solving constrained optimization problems is generally attributed to Courant.In 1943, Courant introduced the earliest PFM with equality constraint and in 1969; Pietrgykowski discussed this approach to solve nonlinear problems.

The significant progress of PFM to solving practical problems follows the classic work of Fiacco and McCormick which is titled sequential unconstrained minimization technique (SUMT) and the algorithm is as follows:

Step 1: Select a growth parameter, $\eta > 1$, a stopping parameter, $\varepsilon > 0$, and an initial value of the penalty parameter, $C_{0.}$

Step2: Choose a starting point,X_0, that violates at least one of the constrained(incase of multiple constraints) and formulate the augmented objective function, $\theta(C_{0.}, X)$. Let $k = 1$

Step3: Starting fromX^{k-1}, use an unconstrained search techniques to find the point that $Minimizes\ \theta(C_{k-1.}, X)$, call it X^k and determine which constraints are violated at this point.

Step4: If the distance between X^{k-1} and X^k is smaller than ε(i.e. $\| X^{k-1} - X^k \| < \varepsilon$) or the difference between two successive objective functions values is smaller than ε (i.e. $|f(X^{k-1}) - f(X^k)| < \varepsilon$), stop with X^k an estimate of the optimal solution. Otherwise, put $C^k \leftarrow \eta C^{k-1}$, formulate the new $\theta(C_{k.}, X)$ based on which constraints are violated at X^k, put $k \leftarrow k + 1$ and return to the iteration step 3.

In changing the PFM parameters in numerical problems have been investigated by several authors such as: Fiacco and McCormick (1968) and Himmelblau (1972) which discusses the effectiveness of the unconstrained optimization algorithms since the constrained optimization problems must be converted to unconstrained with the help of PFM parameter, μ. Furthermore, Fletcher recorded that several extensions to the concepts of PFM have been made in which one of them how to avoid the difficulties is associated with the ill-conditioning as the PFM parameter tends to infinity.

The Philosophy of Penalty Function methods is violating the constraints and obtain approximate solutions to the original problem by balancing the objective function and a penalty term involving the constraints.

Increasing the penalty parameter, μ, the approximate solution is forced to approach the feasible domain and hopefully, the solution of the original constrained problem is attained.

IV. IMBEDDED PENALTY FUNCTION CONJUGATE GRADIENT METHOD (IPFCGM) ALGORITHM

Haven investigated the two methods; we now draw out the following steps which will be used to solve some constrained optimization problems. The steps are as follows:

Step 1: Equate the constraint to zero and square (in case of equation is of the form: $AX = b$ .i.e. $(AX - b)^2 = 0$)

Step 2: Append the new equation in step1 (i. e. $(AX - b)^2$) into the performance index using Lagrange Multiplier λ to form Lagrangian or Augmented Lagrangian function [i.e. $L(x, \lambda) = f(x) + \lambda(AX - b)^2 = 0$]

Step 3: Guess the initial elements $x_0, \lambda > 0$

Step 4: Compute the initial gradient, g_0, as well as the initial descent direction, $p_0 = -g_0$

Step 5: Compute the Hessian Matrix, H, in step 2

Step 6: Set $x_{i+1} = x_i + \alpha_i p_i,\ where\ \alpha_i = \frac{g^T{}_i g_i}{p^T{}_i H p_i},\ i = 1,\ 2,\ \dots,\ n$

Step 7: Update the gradient using: $g_{i+1} = g_i + \alpha_i H p_i,\ \ i = 1,\ 2,\ \dots,\ n$

Step 8: Update the descent direction using: $p_{i+1} = -g_i + \beta_i p_i,\ where\ \beta_i = \frac{g^T{}_{i+1} g_{i+1}}{g^T{}_i g_i}, i = 1,\ 2,\ \dots,\ n$

Step 9: If $g_i = 0$ stop, else, set $i = i + 1$ and return to step 6.

V. NUMERICAL IMPLEMENTATION OF THE PFCGM

Considering (1.1) and (1.2), there exists a Penalty Function parameter, μ, which imbed (1.2) into (1.1) to give a Modified Penalty Function such as:

$$L(X,\ \mu) = f(X) + \sum_{i=1}^{n} \mu_i h^2{}_i(X) \qquad 3.1$$

Let the initial guess be:

$$x_0 = \begin{pmatrix} x_{1(0)} \\ x_{2(0)} \\ . \\ . \\ . \\ x_{n(0)} \end{pmatrix} \qquad 3.2$$

$$\mu_0 = \begin{pmatrix} \mu_{1(0)} \\ \mu_{2(0)} \\ . \\ . \\ . \\ \mu_{n(0)} \end{pmatrix} \qquad 3.3$$

Putting (3.2) and (3.3) in (1.1) and (3.1) respectively gives the initial functions values i.e. $f(x_0)$ and $L(x_0, \lambda_0)$. Computing the gradient of (3.1) with respect to $(x_1,\ x_2,\ \dots,\ x_n)^T$ we have:

$$\begin{pmatrix} \frac{\partial}{\partial x_1} L(X, \mu) = \frac{\partial}{\partial x_1} f(X) + \mu_i \frac{\partial}{\partial x_1} \sum_{i=1}^{n} h^2{}_i(X) \\ \frac{\partial}{\partial x_2} L(X, \mu) = \frac{\partial}{\partial x_2} f(X) + \lambda_i \frac{\partial}{\partial x_2} \sum_{i=1}^{n} h^2{}_i(X) \\ . \\ . \\ . \\ \frac{\partial}{\partial x_n} L(X, \mu) = \frac{\partial}{\partial x_n} f(X) + \lambda_i \frac{\partial}{\partial x_n} \sum_{i=1}^{n} h^2{}_i(X) \end{pmatrix} \qquad 3.4$$

Putting (3.2) and (3.3) for X and λ respectively in (3.4) gives us the initial gradient as:

$$g_0 = \begin{pmatrix} \frac{\partial}{\partial x_1} L(x_0, \mu_0) \\ \frac{\partial}{\partial x_2} L(x_0, \mu_0) \\ . \\ . \\ . \\ \frac{\partial}{\partial x_n} L(x_0, \mu_0) \end{pmatrix} \quad 3.5$$

Multiplying (3.5) by negative gives the decent direction as:

$$p_0 = -g_0 = \begin{pmatrix} -\frac{\partial}{\partial x_1} L(x_0, \mu_0) \\ -\frac{\partial}{\partial x_2} L(x_0, \mu_0) \\ . \\ . \\ . \\ -\frac{\partial}{\partial x_n} L(x_0, \mu_0) \end{pmatrix} \quad 3.6$$

Computing the Hessian Matrix of (3.1) using (3.4) gives:

$$\mathrm{H} = \begin{pmatrix} \frac{\partial^2 L(x_0,\mu_0)}{\partial x^2{}_1} & \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{12}} & \cdots & \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{1n}} \\ \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{21}} & \frac{\partial^2 L(x_0,\mu_0)}{\partial x^2{}_2} & \cdots & \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{2n}} \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{m1}} & \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{m2}} & \cdots & \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{mn}} \end{pmatrix} \quad 3.7$$

On transposing (3.5) and (3.6) respectively, we have:

$$g_0{}^T = \left(\frac{\partial}{\partial x_1} L(x_0, \mu_0) \ \frac{\partial}{\partial x_2} L(x_0, \mu_0) \ \ldots \ \frac{\partial}{\partial x_n} L(x_0, \mu_0)\right)^T \quad 3.8$$

and

$$p_0{}^T = \left(-\frac{\partial}{\partial x_1} L(x_0, \mu_0) \ -\frac{\partial}{\partial x_2} L(x_0, \mu_0) \ \ldots \ -\frac{\partial}{\partial x_n} L(x_0, \mu_0)\right)^T \quad 3.9$$

Multiplying (3.5) and (3.8) gives us a scalar, k .i.e.

$$k = g_0{}^T g_0 \ \mathrm{k} = \left(\frac{\partial}{\partial x_1} L(x_0, \mu_0) \quad \frac{\partial}{\partial x_2} L(x_0, \mu_0) \quad \ldots \quad \frac{\partial}{\partial x_n} L(x_0, \mu_0)\right)^T \begin{pmatrix} \frac{\partial}{\partial x_1} L(x_0, \mu_0) \\ \frac{\partial}{\partial x_2} L(x_0, \mu_0) \\ . \\ . \\ . \\ \frac{\partial}{\partial x_n} L(x_0, \mu_0) \end{pmatrix}$$

$$\mathrm{k} = \left(\left(\frac{\partial}{\partial x_1} L(x_0, \mu_0)\right)^2 + \left(\frac{\partial}{\partial x_2} L(x_0, \mu_0)\right)^2 + \ \ldots + \left(\frac{\partial}{\partial x_n} L(x_0, \mu_0)\right)^2\right) \quad 3.10$$

Similarly, multiplying (3.9), (3.7) and (3.6) gives a scalar, z .i.e.

$$z = p_0{}^T H p_0$$

$$z = \left(-\frac{\partial}{\partial x_1} L(x_0, \mu_0) \quad -\frac{\partial}{\partial x_2} L(x_0, \mu_0) \quad \ldots \quad -\frac{\partial}{\partial x_n} L(x_0, \mu_0)\right)^T \begin{pmatrix} \frac{\partial^2 L(x_0,\mu_0)}{\partial x^2{}_1} & \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{12}} & \cdots & \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{1n}} \\ \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{21}} & \frac{\partial^2 L(x_0,\mu_0)}{\partial x^2{}_2} & \cdots & \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{2n}} \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{m1}} & \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{m2}} & \cdots & \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{mn}} \end{pmatrix} \begin{pmatrix} -\frac{\partial}{\partial x_1} L(x_0, \mu_0) \\ -\frac{\partial}{\partial x_2} L(x_0, \mu_0) \\ . \\ . \\ . \\ -\frac{\partial}{\partial x_n} L(x_0, \mu_0) \end{pmatrix} \quad 3.11$$

$$Hp_0 = \begin{pmatrix} \frac{\partial^2 L(x_0,\mu_0)}{\partial x^2{}_1} & \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{12}} & \cdots & \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{1n}} \\ \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{21}} & \frac{\partial^2 L(x_0,\mu_0)}{\partial x^2{}_2} & \cdots & \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{2n}} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{m1}} & \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{m2}} & \cdots & \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{mn}} \end{pmatrix} \begin{pmatrix} -\frac{\partial}{\partial x_1} L(x_0,\mu_0) \\ -\frac{\partial}{\partial x_1} L(x_0,\mu_0) \\ \vdots \\ -\frac{\partial}{\partial x_n} L(x_0,\mu_0) \end{pmatrix}$$

$$= \begin{pmatrix} \frac{\partial^2 L(x_0,\mu_0)}{\partial x^2{}_1}\left(-\frac{\partial}{\partial x_1} L(x_0,\mu_0)\right) + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{12}}\left(-\frac{\partial}{\partial x_2} L(x_0,\mu_0)\right) + \cdots + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{1n}}\left(-\frac{\partial}{\partial x_n} L(x_0,\mu_0)\right) \\ \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{21}}\left(-\frac{\partial}{\partial x_1} L(x_0,\mu_0)\right) + \frac{\partial^2 L(x_0,\mu_0)}{\partial x^2{}_2}\left(-\frac{\partial}{\partial x_2} L(x_0,\mu_0)\right) + \cdots + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{2n}}\left(-\frac{\partial}{\partial x_n} L(x_0,\mu_0)\right) \\ \vdots \\ \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{m1}}\left(-\frac{\partial}{\partial x_1} L(x_0,\mu_0)\right) + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{m2}}\left(-\frac{\partial}{\partial x_2} L(x_0,\mu_0)\right) + \cdots + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{mn}}\left(-\frac{\partial}{\partial x_n} L(x_0,\mu_0)\right) \end{pmatrix} \quad 3.12$$

Putting (3.12) into (3.11), we have:

$$z = \left(-\frac{\partial}{\partial x_1} L(x_0,\mu_0) \quad -\frac{\partial}{\partial x_2} L(x_0,\mu_0) \quad \ldots \quad -\frac{\partial}{\partial x_n} L(x_0,\mu_0)\right)^T \begin{pmatrix} \frac{\partial^2 L(x_0,\mu_0)}{\partial x^2{}_1}\left(-\frac{\partial}{\partial x_1} L(x_0,\mu_0)\right) + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{12}}\left(-\frac{\partial}{\partial x_2} L(x_0,\mu_0)\right) + \cdots + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{1n}}\left(-\frac{\partial}{\partial x_n} L(x_0,\mu_0)\right) \\ \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{21}}\left(-\frac{\partial}{\partial x_1} L(x_0,\mu_0)\right) + \frac{\partial^2 L(x_0,\mu_0)}{\partial x^2{}_2}\left(-\frac{\partial}{\partial x_2} L(x_0,\mu_0)\right) + \cdots + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{2n}}\left(-\frac{\partial}{\partial x_n} L(x_0,\mu_0)\right) \\ \vdots \\ \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{m1}}\left(-\frac{\partial}{\partial x_1} L(x_0,\mu_0)\right) + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{m2}}\left(-\frac{\partial}{\partial x_2} L(x_0,\mu_0)\right) + \cdots + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{mn}}\left(-\frac{\partial}{\partial x_n} L(x_0,\mu_0)\right) \end{pmatrix} \quad 3.13$$

With matrix multiplication, (3.13) becomes:

$$z = \begin{pmatrix} -\frac{\partial}{\partial x_1} L(x_0,\mu_0)\left(\frac{\partial^2 L(x_0,\mu_0)}{\partial x^2{}_1}\left(-\frac{\partial}{\partial x_1} L(x_0,\mu_0)\right) + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{12}}\left(-\frac{\partial}{\partial x_2} L(x_0,\mu_0)\right) + \cdots + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{1n}}\left(-\frac{\partial}{\partial x_n} L(x_0,\mu_0)\right)\right) \\ -\frac{\partial}{\partial x_2} L(x_0,\mu_0)\left(\frac{\partial^2 L(x_0,\mu_0)}{\partial x_{21}}\left(-\frac{\partial}{\partial x_1} L(x_0,\mu_0)\right) + \frac{\partial^2 L(x_0,v_0)}{\partial x^2{}_2}\left(-\frac{\partial}{\partial x_2} L(x_0,v_0)\right) + \cdots + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{2n}}\left(-\frac{\partial}{\partial x_n} L(x_0,\mu_0)\right)\right) \\ \vdots \\ -\frac{\partial}{\partial x_n} L(x_0,\mu_0)\left(\frac{\partial^2 L(x_0,\mu_0)}{\partial x_{m1}}\left(-\frac{\partial}{\partial x_1} L(x_0,\mu_0)\right) + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{m2}}\left(-\frac{\partial}{\partial x_2} L(x_0,\mu_0)\right) + \cdots + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{mn}}\left(-\frac{\partial}{\partial x_n} L(x_0,\mu_0)\right)\right) \end{pmatrix} \quad 3.14$$

Dividing (3.10) and (3.14) .i.e.:

$$\alpha_0 = \frac{\left(\left(\frac{\partial}{\partial x_1} L(x_0,\mu_0)\right)^2 + \left(\frac{\partial}{\partial x_2} L(x_0,\mu_0)\right)^2 + \ldots + \left(\frac{\partial}{\partial x_n} L(x_0,\mu_0)\right)^2\right)}{\begin{pmatrix} -\frac{\partial}{\partial x_1} L(x_0,\mu_0)\left(\frac{\partial^2 L(x_0,\mu_0)}{\partial x^2{}_1}\left(-\frac{\partial}{\partial x_1} L(x_0,\mu_0)\right) + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{12}}\left(-\frac{\partial}{\partial x_2} L(x_0,\mu_0)\right) + \cdots + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{1n}}\left(-\frac{\partial}{\partial x_n} L(x_0,\mu_0)\right)\right) \\ -\frac{\partial}{\partial x_2} L(x_0,\mu_0)\left(\frac{\partial^2 L(x_0,\mu_0)}{\partial x_{21}}\left(-\frac{\partial}{\partial x_1} L(x_0,\mu_0)\right) + \frac{\partial^2 L(x_0,\mu_0)}{\partial x^2{}_2}\left(-\frac{\partial}{\partial x_2} L(x_0,\mu_0)\right) + \cdots + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{2n}}\left(-\frac{\partial}{\partial x_n} L(x_0,\mu_0)\right)\right) \\ \vdots \\ -\frac{\partial}{\partial x_n} L(x_0,\mu_0)\left(\frac{\partial^2 L(x_0,\mu_0)}{\partial x_{m1}}\left(-\frac{\partial}{\partial x_1} L(x_0,v_0)\right) + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{m2}}\left(-\frac{\partial}{\partial x_2} L(x_0,\mu_0)\right) + \cdots + \frac{\partial^2 L(x_0,\mu_0)}{\partial x_{mn}}\left(-\frac{\partial}{\partial x_n} L(x_0,\mu_0)\right)\right) \end{pmatrix}} \quad 3.15$$

(3.15) is the step length. Now set $x_{i+1} = x_i + \alpha_i p_i,\ i = 0,\ 1,\ 2,\ \ldots,\ n$.

VI. CONCLUSION

This paper has been able to show clearly the derivation of the numerical implementation of the Penalty Function Method imbedded in the Conjugate Gradient Method which is the interpretation of the new algorithm for easy applicability of this method to solving either equality or inequality constrained optimization problems.

In future, we hope to devote more attention on the application of this method to solving constrained optimization of both equality and inequality such as the form of(1.1).

REFERENCES

[1] RAO, S. S., (1978), Optimization Theory and Applications, Willy and Sons.
[2] THOMAS, F.E., and DAVID, M.H., (2001), Optimization of Chemical Processes, McGraw Hill
[3] IGOR, G., STEPHEN, G. N. and ARIELA, S., (2009), Linear and Nonlinear Optimization, George Mason University, Fairfax, Virginia, SIAM, Philadelphia.
[4] David, G. Hull, (2003), Optimal Control Theory for Applications, MechanicalEngineeringSeries, Springer-Verlag, New York, Inc., 175 Fifth Avenue, New York, NY 10010.
[5] Bersekas, D. P, (1982), Constrained Optimization and Lagrange Multipliers Method, Academic Press, Inc.
[6] Rockafellar, R. T., (2005), Multiplier Method of Hestenes and Powell applied to convex Programming, Journal of Optimization Theory and Applications, Vol. 4, No. 4.
[7] S. A. OLORUNSOLA**,** T .E. OLAOSEBIKAN and K. J. ADEBAYO (2014),On Application of Modified Lagrange Multipliers for Solving Constrained Optimization Problems, International Organization of Science Research Journal of Mathematics India. Vol.10, Issue 3, pp6-9.