






# Combination of Penalty Function, Lagrange Multiplier and Conjugate Gradient Methods for the Solution of Constrained Optimization Problems

R. B. Ogunrinde<sup>1</sup>   
T. E. Olaosebikan<sup>2</sup> 

<sup>1,2</sup>Department of Mathematical Sciences, Ekiti State University, Ado Ekiti, Nigeria.  
( Corresponding Author)

## Abstract


In this paper, we combined Lagrange Multiplier, Penalty Function and Conjugate Gradient Methods (CPLCGM), to enable Conjugate Gradient Method (CGM) to be employed for solving constrained optimization problems. In the year past, Lagrange Multiplier Method (LMM) has been used extensively to solve constrained optimization problems likewise Penalty Function Method (PFM). However, with some special features in CGM, which makes it unique in solving unconstrained optimization problems, we see that this features we be advantageous to solve constrained optimization problems if it can be properly amended. This, then call for the CPLCGM that is aimed at taking care of some constrained optimization problems, either with equality or inequality constraint but in this paper, we focus on equality constraints. The authors of this paper desired that, with the construction of the new Algorithm, it will circumvent the difficulties undergone using only LMM and as well as PFM to solve constrained optimization problems and its application will further improve the result of the Conjugate Gradient Method in solving this class of optimization problem. We applied the new algorithm to some constrained optimization problems and compared the results with the LMM and PFM.

**Keywords:** Lagrange multiplier method, Constrained optimization problem, Conjugate gradient algorithm, Penalty function method, Conjugate gradient method.


## Contents

1. Introduction.....	2
2. Lagrange Multiplier Method (LMM) .....	2
3. Penalty Function Method (PFM).....	2
4. Conjugate Gradient Method (CGM) .....	3
5. Combination of Penalty Function, Lagrange Multiplier and Conjugate Gradient Methods (CPLCGM) .....	3
6. Combination of Penalty Method, the Lagrange Multiplier Method and the Conjugate Gradient Method (CPLCGM) Algorithm.....	4
7. Mathematical Computation of CPLCGM for Equality Constraint .....	4
8. Data Analysis.....	7
9. Conclusion .....	8
References.....	8

**Citation** | R. B. Ogunrinde; T. E. Olaosebikan (2016). Combination of Penalty Function, Lagrange Multiplier and Conjugate Gradient Methods for the Solution of Constrained Optimization Problems. Asian Bulletin of Energy Economics and Technology, 3(1): 1-8.

**DOI:** 10.20448/journal.507/2016.3.1/507.1.1.8 

**ISSN(E) :** 2313-819X

**Licensed:** This work is licensed under a [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/) 

**Contribution/Acknowledgement:** All authors contributed to the conception and design of the study.

**Funding:** This study received no specific financial support.

**Competing Interests:** The authors declare that they have no conflict of interests.

**Transparency:** The authors confirm that the manuscript is an honest, accurate, and transparent account of the study was reported; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained.

**Ethical:** This study follows all ethical practices during writing.

**History:** **Received:** 21 September 2015/ **Revised:** 7 October 2015/ **Accepted:** 28 October 2015/ **Published:** 24 October 2015

**Publisher:** Asian Online Journal Publishing Group

## 1. Introduction

The basic problem to be considered in this paper is the minimization of a function subject to equality and inequality constraints of the form:

$$\text{Minimize } f(\mathbf{X}) \quad 1.1a$$

$$\text{Subject to: } g_i(\mathbf{X}) \leq 0, i = 1, 2, \dots, n \quad 1.1b$$

$$h_j(\mathbf{X}) = 0, j = 1, 2, \dots, m \quad 1.1c$$

Where  $\mathbf{X}$  is a vector of  $n$  variables  $f(\mathbf{X})$  is the objective function to be minimized,  $g_i(\mathbf{X})$ , a set of inequality constraints and  $h_j(\mathbf{X})$ , a set of equality constraints which are twice differentiable. [6]

## 2. Lagrange Multiplier Method (LMM)

In mathematical optimization, the method of Lagrange Multipliers (named after Joseph Louis Lagrange) provides a strategy for finding the maximum/minimum of a function subject to constraints. The Lagrange multiplier method was basically introduced to solve constrained optimization problems with equality constraints of the form (1.1a) and (1.1c) and the theorem associated with Lagrange Multiplier Method states that:

If  $\mathbf{X}^*$  affords a local minimum to  $f(\mathbf{X})$  subject to the constraints  $h_i(\mathbf{X}) = 0$ , then there exists a unique set of multipliers

$\lambda_i$ , ( $i = 1, 2, \dots, m$ ) such that:

$$L(\mathbf{X}, \boldsymbol{\lambda}) = f(\mathbf{X}) + \sum_{i=1}^m \lambda_i h_i(\mathbf{X}) \quad 2.1a$$

then,

$$\nabla L(\mathbf{X}^*, \boldsymbol{\lambda}^*) = \nabla f(\mathbf{X}^*) + \nabla(\sum_{i=1}^m \lambda_i h_i(\mathbf{X})) = 0 \quad 2.1b$$

and

$$L''(\mathbf{X}^*, \boldsymbol{\lambda}^*) = \frac{\partial^2 L(\mathbf{X}^*, \boldsymbol{\lambda}^*)}{\partial^2 X_i} > 0, i = 1, 2, \dots \quad 2.1c$$

Where  $\nabla$  denotes the gradient of the function and  $L''$  denotes the second derivatives of (2.1a) by Hestenes (1975).

Equation (2.1b) and (2.1c) are the necessary conditions for locally constrained minima. Equation (2.1b) and the feasibility condition (2.1b) constitute the Kuhn-Tucker necessary conditions for optimality. It is assumed that  $f(\mathbf{X})$  and  $h_i(\mathbf{X})$  are second order differentiable and that the gradients  $\nabla h_i(\mathbf{X})$  are not zero at  $\mathbf{X}^*$ .

The problem can now be stated in terms of the equivalent classical Lagrangian as:

$$\text{Minimize } L(\mathbf{X}, \boldsymbol{\lambda}) \quad 2.2a$$

$$\text{Subject to } h_i(\mathbf{X}) = 0 (i = 1, 2, \dots, m < n) \quad 2.2b$$

Assuming the existence of the saddle points of the Lagrangian (1.3), the following condition exists:

$$L(\mathbf{X}^*, \boldsymbol{\lambda}) \leq L(\mathbf{X}^*, \boldsymbol{\lambda}^*) \leq L(\mathbf{X}, \boldsymbol{\lambda}^*) \quad 2.3$$

The optimal pair  $(\mathbf{X}^*, \boldsymbol{\lambda}^*)$  can be obtained by first minimizing  $L(\mathbf{X}, \boldsymbol{\lambda})$  with respect to  $\mathbf{X}$ , then maximizing  $L(\mathbf{X}^k, \boldsymbol{\lambda})$  with respect to  $\boldsymbol{\lambda}$  by updating equation:

$$\lambda_i^{k+1} = \lambda_i^k + c[h_i(\mathbf{X}^k)] \quad 2.4$$

Where  $c$  is a scalar parameter (step size),  $k$  is the iteration number and  $\mathbf{X}^k$  is the local minimum of  $L(\mathbf{X}^k, \boldsymbol{\lambda}^k)$ . The procedure is repeated until convergence is attained. This is also called “primal-dual” method. Serious disadvantages are encountered in the primal-dual method.

First, the problem (2.3) must have a locally convex structure for the dual problem to be well defined and for (2.4) to be meaningful (Luenberger, 1973).

Second, a large number of iterations are usually required to minimize (2.1a). since the ascent iteration (2.4) converges only moderately fast.

Because of this, primal-dual methods have found application in a limited class of problems where minimization of the Lagrangian (2.1a) can be efficiently carried out due to special structure, as shown by Luenberger (1973) or where the design problem exhibits a unique form, as shown by Schmit and Fleury (1979).

## 3. Penalty Function Method (PFM)

Penalty Function Methods have been used extensively since the mid-1940, Pierre and Lowe (1975) considered PFM to be efficient for inequality constrained problems such as equation (1.1a) with respect to (1.1b).

Considering (1.1a) and (1.1b) the general Exterior Penalty Function for this class of problem is defined as:

$$L(\mathbf{X}, c) = f(\mathbf{X}) + c \sum_{i=1}^m \phi_i(t) \quad 3.1$$

Where  $\phi(t)$  in (3.1) is some scalar of the Penalty Function Method of the constraints and  $c$  is the penalty parameter.

Now, the most common PFM is the quadratic type where  $\phi_i(t)$  in (3.1) is defined as:  $\phi(t) = \frac{t^2}{2}$ . However, it may be desirable at times to use other PFM.

The Quadratic Penalty Function of (1.1a) with respect to (1.1b) is given as:

$$L(\mathbf{X}, c) = f(\mathbf{X}) + \frac{c}{2} \sum_{i=1}^m p_i^2(\mathbf{X}) \quad 3.2$$

where

$$p_i(\mathbf{X}) = \begin{cases} g_i(\mathbf{X}), & \text{if } g_i(\mathbf{X}) \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

(3.1) and (3.2) are now an unconstrained minimization.

We need to note that as  $c$  in (3.1) and (3.2) gets larger, the function values changes more rapidly and the optimum becomes more difficult to find regardless of the minimization techniques. This causes the numerical ill-conditioning inherent with Penalty Methods.

Interior Penalty Function Method has the advantage of approaching the optimum from the feasible region thus, yielding a feasible solution. However, the penalty function is discontinuous at the constraint boundaries.

The Interior Penalty Function form of (1.1a) with respect to (1.1b) is defined as:

$$L(\mathbf{X}) = f(\mathbf{X}) - \frac{1}{c} \sum_{i=1}^m \frac{1}{g_i(\mathbf{X})} \quad 3.3$$

We discovered that both Exterior and Interior Penalty Function exhibit the same problem of ill-conditioning and slow convergence, the exact solution is not possible but the solution achieved is feasible.

#### 4. Conjugate Gradient Method (CGM)

The development of the Conjugate Gradient Method (CGM) algorithm for solving algebraic equations can be traced to Hestenes and Stiefel (1952) which was successfully applied to nonlinear equations with results reported by Fletcher and Reeves in 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  $x_0 \in \mathcal{H}$  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  $p_0 = -g_0$

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

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

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

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  $i$ th step of the algorithm,  $\alpha_i$ , 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. Applicability of the CGM algorithm thus depends on the 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 computation per iteration.

Since so many researchers have worked on CGM, for the effort expended by these researchers in constructing the control operator and even the method in question (CGM) not to be limited to solving this class of problems alone, the desire to combination of the Lagrange multiplier method (LMM), the Penalty Function Method and the Conjugate Gradient Method (CGM) for solving Constrained Optimization Problems was borne out and the resulting algorithm is as follows

#### 5. Combination of Penalty Function, Lagrange Multiplier and Conjugate Gradient Methods (CPLCGM)

The Lagrange multiplier method can be perceived to be a combined primal-dual and Penalty Function Methods. Though they are theoretically similar but their behavior is quite different.

It has been shown that the original equality constrained problem (1.1a) with (1.1c) is equivalent to the classical Lagrangian (1.3). Since (1.3) is still an equality constrained problem, it can be solved by the usual Exterior Penalty Function method. The Quadratic Penalty Function is used so that first derivatives are continuous. Substituting (1.3) into (3.2) we have:

$$\begin{aligned} L(\mathbf{X}, \boldsymbol{\lambda}, \mathbf{c}) &= L(\mathbf{X}, \boldsymbol{\lambda}) + \frac{c}{2} \sum_{i=1}^m p_i^2(\mathbf{X}) \\ &= f(\mathbf{X}) + \sum_{i=1}^m \lambda_i h_i(\mathbf{X}) + \frac{c}{2} \sum_{i=1}^m p_i^2(\mathbf{X}) \end{aligned} \quad 5.1$$

where  $p_i(\mathbf{X}) = h_i(\mathbf{X})$

Equation (5.1) is referred to as Augmented Lagrange Function for the equality constrained problem.

Now, we want to extend this discussion to include inequality constraints.

Considering (1.1a) with respect to (1.1b) introducing slack variables, (1.1b) becomes:

$$g_i(\mathbf{X}) + z_i^2 = 0 \quad 5.2$$

Where  $z_i^2$  in (5.2) is the slack variable for the  $i^{th}$  constraints. The problem is now an equality constrained problem of the form (1.1a) with respect to (1.1c); however, the number of design variables will increase to  $n + m$  and (5.2) becomes:

$$L(\mathbf{X}, \boldsymbol{\lambda}, \mathbf{c}, \mathbf{z}) = f(\mathbf{X}) + \sum_{i=1}^m \lambda_i (g_i(\mathbf{X}) + z_i^2) + \frac{c}{2} \sum_{i=1}^m (g_i(\mathbf{X}) + z_i^2)^2 \quad 5.3$$

If the number of constraints,  $n$ , in (5.3) is much greater than the number of design variables as is often the case in engineering design problems, the unconstrained minimization problem is sizable. The scope of the problem can, however, be reduced by eliminating the slack variables,  $z_i^2$ , by first minimizing (5.3) with respect to  $\mathbf{z}$ . For a local minimum to exist, the stationary conditions:

$$\frac{\partial L}{\partial z_i} = 0 \quad (i = 1, 2, \dots, n) \quad 5.4$$

must hold. Differentiating (5.3) we have:

$$\frac{\partial L}{\partial z_i} = 2\lambda_i z_i + c(g_i(X) + z_i^2)(2z_i)$$

From (5.4) we have that

$$[2z_i(\lambda_i + c(g_i(X) + z_i^2))] = 0 \tag{5.5}$$

The solution to (5.5) is:

$$2z_i = 0 \Rightarrow z_i = 0$$

and also,

$$\lambda_i + c(g_i(X) + z_i^2) = 0$$

$$cz_i^2 = -\lambda_i - cg_i(X)$$

Therefore,

$$z_i^2 = \frac{-\lambda_i - cg_i(X)}{c} \tag{5.6}$$

Since  $z_i^2 < 0$  is meaningless, the solution becomes:

$$z_i^2 = \max(0, -g_i(X) - \frac{\lambda_i}{c}) \tag{5.7}$$

(5.7) shows that  $z_i$  is no longer an independent variable. From this (5.7) it is observed that if  $g_i(X)$  is a critical constraint,  $z_i = 0$  and if otherwise,  $z_i > 0$ . Therefore:

$$g_i(X) + z_i^2 = \max(g_i(X), -\frac{\lambda_i}{c}) \tag{5.8}$$

with the slack variables eliminated, the augmented Lagrangian becomes:

$$L(\mathbf{X}, \boldsymbol{\lambda}, \mathbf{c}) = f(\mathbf{X}) + \sum_{i=1}^m \left( \lambda_i p_i + \frac{c}{2} p_i^2(\mathbf{X}) \right) \tag{5.9}$$

where

$$p_i = \max(g_i(X), -\frac{\lambda_i}{c})$$

Equation (5.9) is referred to as Rockafellar's augmented Lagrange Function.

It must be noted that since we have succeeded in converting inequality constrained problem to an equivalent equality constrained problem, it invariably mean that the convergence properties are identical.

At this junction, we now apply one of the methods used for solving unconstrained optimization problem which is Conjugate Gradient Method with some modification to suit a constrained optimization problem and this resulted to the following steps:

## 6. Combination of Penalty Method, the Lagrange Multiplier Method and the Conjugate Gradient Method (CPLCGM) Algorithm

Haven investigated these 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: Choose an Lagrange Multiplier  $\lambda > 0$ , Penalty parameter  $c > 0$  and guess the initial elements  $X_0$ .

Step 2: Formulate the unconstrained minimization problem:

$$\text{Min } L(X, \lambda, c) = f(X) + \sum_{i=1}^m \left( \lambda_i p_i + \frac{c}{2} p_i^2(X) \right) + \sum_{j=1}^{m+n} \left( \lambda_j h_j(X) + \frac{c}{2} h_j^2(X) \right)$$

$$\text{where } p_i = \max(g_i(X), -\frac{\lambda_i}{c})$$

Step 3: Compute the initial gradient,  $g_0$ , as well as the initial descent direction,  $p_0 = -g_0$

Step 4: Compute the Hessian Matrix,  $H$ , in step 2

Step 6: Set  $x_{i+1} = x_i + \alpha_i p_i$ ,

$$\text{where } \alpha_i = \frac{g_i^T g_i}{p_i^T H p_i}, i = 1, 2, \dots, n$$

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:

$$p_{i+1} = -g_{i+1} + \beta_i p_i,$$

$$\text{where } \beta_i = \frac{g_{i+1}^T g_{i+1}}{g_i^T g_i}, i = 1, 2, \dots, n$$

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

## 7. Mathematical Computation of CPLCGM for Equality Constraint

Considering (1.1) and (1.3), there exists a Lagrange Multiplier  $\lambda$  which imbed (1.3) into (1.1) to give a Lagrangian function such as:

$$L(X, \lambda) = f(X) + \sum_{i=1}^n \lambda_i h_i(X) + \sum_{i=1}^n \frac{c}{2} p_i^2(X) \tag{7.1}$$

Where  $h_i = p_i$

Let the initial guess be:

$$x_0 = \begin{pmatrix} x_{1(0)} \\ x_{2(0)} \\ \vdots \\ \vdots \\ x_{n(0)} \end{pmatrix} \tag{7.2}$$

$$\lambda = \begin{pmatrix} \lambda_{1(0)} \\ \lambda_{2(0)} \\ \vdots \\ \vdots \\ \lambda_{n(0)} \end{pmatrix} \tag{7.3}$$

$$C = \begin{pmatrix} C_{1(0)} \\ C_{2(0)} \\ \vdots \\ \vdots \\ C_{n(0)} \end{pmatrix} \tag{7.4}$$

Putting (7.2), (7.3) and (7.4) in (7.1) and (5.6) respectively gives the initial functions values i.e.  $f(X_0)$  and  $L(X_0, \lambda, c)$ .

Computing the gradient of (7.1) with respect to  $(x_1, x_2, \dots, x_n)^T$  we have:

$$\begin{pmatrix} \frac{\partial}{\partial x_1} L(X, \lambda, C) = \frac{\partial}{\partial x_1} f(X) + \frac{\partial}{\partial x_1} \sum_{i=1}^n \lambda_i h_i(X) + \frac{\partial}{\partial x_1} \sum_{i=1}^n \left( \frac{c}{2} p_i^2(X) \right) \\ \frac{\partial}{\partial x_2} L(X, \lambda, C) = \frac{\partial}{\partial x_2} f(X) + \frac{\partial}{\partial x_2} \sum_{i=1}^n \lambda_i h_i(X) + \frac{\partial}{\partial x_2} \sum_{i=1}^n \left( \frac{c}{2} p_i^2(X) \right) \\ \vdots \\ \frac{\partial}{\partial x_n} L(X, \lambda, C) = \frac{\partial}{\partial x_n} f(X) + \frac{\partial}{\partial x_n} \sum_{i=1}^n \lambda_i h_i(X) + \frac{\partial}{\partial x_n} \sum_{i=1}^n \left( \frac{c}{2} p_i^2(X) \right) \end{pmatrix} \tag{7.5}$$

Putting (7.2) through (7.4) in (7.5) gives us the initial gradient as:

$$g_0 = \begin{pmatrix} \frac{\partial}{\partial x_1} L(x_0, \lambda, c) \\ \frac{\partial}{\partial x_2} L(x_0, \lambda, c) \\ \vdots \\ \vdots \\ \frac{\partial}{\partial x_n} L(x_0, \lambda, c) \end{pmatrix} \tag{7.6}$$

Multiplying (7.6) by negative gives the decent direction as:

$$p_0 = -g_0 = \begin{pmatrix} -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) \\ -\frac{\partial}{\partial x_2} L(x_0, \lambda, c) \\ \vdots \\ \vdots \\ -\frac{\partial}{\partial x_n} L(x_0, \lambda, c) \end{pmatrix} \tag{7.7}$$

Computing the Hessian Matrix of (7.1) using (7.5) gives:

$$H = \begin{pmatrix} \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_1^2} & \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{12}} & \dots & \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{1n}} \\ \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{21}} & \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_2^2} & \dots & \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{m1}} & \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{m2}} & \dots & \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{mn}} \end{pmatrix} \tag{7.8}$$

On transposing (7.6) and (7.7) respectively, we have:

$$g_0^T = \left( \frac{\partial}{\partial x_1} L(x_0, \lambda, c) \quad \frac{\partial}{\partial x_2} L(x_0, \lambda, c) \dots \frac{\partial}{\partial x_n} L(x_0, \lambda, c) \right)^T \tag{7.9}$$

and

$$p_0^T = \left( -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) - \frac{\partial}{\partial x_2} L(x_0, \lambda, c) \dots - \frac{\partial}{\partial x_n} L(x_0, \lambda, c) \right)^T \quad 7.10$$

Multiplying (7.6) and (7.9) gives us a scalar,  $k$  i.e.

$$k = g_0^T g_0$$

$$k = \left( \frac{\partial}{\partial x_1} L(x_0, \lambda, c) \quad \frac{\partial}{\partial x_2} L(x_0, \lambda, c) \quad \dots \quad \frac{\partial}{\partial x_n} L(x_0, \lambda, c) \right)^T \begin{pmatrix} \frac{\partial}{\partial x_1} L(x_0, \lambda, c) \\ \frac{\partial}{\partial x_2} L(x_0, \lambda, c) \\ \vdots \\ \frac{\partial}{\partial x_n} L(x_0, \lambda, c) \end{pmatrix} \quad 7.11$$

$$k = \left( \left( \frac{\partial}{\partial x_1} L(x_0, \lambda, c) \right)^2 + \left( \frac{\partial}{\partial x_2} L(x_0, \lambda, c) \right)^2 + \dots + \left( \frac{\partial}{\partial x_n} L(x_0, \lambda, c) \right)^2 \right)$$

Similarly, multiplying (7.10), (7.8) and (7.7) gives a scalar,  $z$  i.e.

$$z = p_0^T H p_0$$

$$z = \left( -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) \quad -\frac{\partial}{\partial x_2} L(x_0, \lambda, c) \quad \dots \quad -\frac{\partial}{\partial x_n} L(x_0, \lambda, c) \right)^T \begin{pmatrix} \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_1^2} & \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{12}} & \dots & \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{1n}} \\ \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{21}} & \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_2^2} & \dots & \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{m1}} & \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{m2}} & \dots & \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{mn}} \end{pmatrix} \begin{pmatrix} -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) \\ -\frac{\partial}{\partial x_2} L(x_0, \lambda, c) \\ \vdots \\ -\frac{\partial}{\partial x_n} L(x_0, \lambda, c) \end{pmatrix} \quad 7.12$$

$$H p_0 = \begin{pmatrix} \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_1^2} & \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{12}} & \dots & \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{1n}} \\ \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{21}} & \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_2^2} & \dots & \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{m1}} & \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{m2}} & \dots & \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{mn}} \end{pmatrix} \begin{pmatrix} -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) \\ -\frac{\partial}{\partial x_2} L(x_0, \lambda, c) \\ \vdots \\ -\frac{\partial}{\partial x_n} L(x_0, \lambda, c) \end{pmatrix}$$

$$\begin{pmatrix} \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_1^2} \left( -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) \right) + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{12}} \left( -\frac{\partial}{\partial x_2} L(x_0, \lambda, c) \right) + \dots + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{1n}} \left( -\frac{\partial}{\partial x_n} L(x_0, \lambda, c) \right) \\ \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{21}} \left( -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) \right) + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_2^2} \left( -\frac{\partial}{\partial x_2} L(x_0, \lambda, c) \right) + \dots + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{2n}} \left( -\frac{\partial}{\partial x_n} L(x_0, \lambda, c) \right) \\ \vdots \\ \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{m1}} \left( -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) \right) + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{m2}} \left( -\frac{\partial}{\partial x_2} L(x_0, \lambda, c) \right) + \dots + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{mn}} \left( -\frac{\partial}{\partial x_n} L(x_0, \lambda, c) \right) \end{pmatrix} \quad 7.13$$

putting (7.13) into (7.12) we have:  
 $z =$  With matrix multiplication, (7.14) becomes:

$$z = \left( -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) \quad -\frac{\partial}{\partial x_2} L(x_0, \lambda, c) \quad \dots \quad -\frac{\partial}{\partial x_n} L(x_0, \lambda, c) \right)^T \begin{pmatrix} \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_1^2} \left( -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) \right) + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{12}} \left( -\frac{\partial}{\partial x_2} L(x_0, \lambda, c) \right) + \dots + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{1n}} \left( -\frac{\partial}{\partial x_n} L(x_0, \lambda, c) \right) \\ \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{21}} \left( -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) \right) + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_2^2} \left( -\frac{\partial}{\partial x_2} L(x_0, \lambda, c) \right) + \dots + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{2n}} \left( -\frac{\partial}{\partial x_n} L(x_0, \lambda, c) \right) \\ \vdots \\ \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{m1}} \left( -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) \right) + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{m2}} \left( -\frac{\partial}{\partial x_2} L(x_0, \lambda, c) \right) + \dots + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{mn}} \left( -\frac{\partial}{\partial x_n} L(x_0, \lambda, c) \right) \end{pmatrix} \quad 7.14$$

$$z = \begin{pmatrix} -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) \left( \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_1^2} \left( -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) \right) + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{12}} \left( -\frac{\partial}{\partial x_2} L(x_0, \lambda, c) \right) + \dots + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{1n}} \left( -\frac{\partial}{\partial x_n} L(x_0, \lambda, c) \right) \right) \\ -\frac{\partial}{\partial x_2} L(x_0, \lambda, c) \left( \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{21}} \left( -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) \right) + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_2^2} \left( -\frac{\partial}{\partial x_2} L(x_0, \lambda, c) \right) + \dots + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{2n}} \left( -\frac{\partial}{\partial x_n} L(x_0, \lambda, c) \right) \right) \\ \vdots \\ -\frac{\partial}{\partial x_n} L(x_0, \lambda, c) \left( \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{n1}} \left( -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) \right) + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{n2}} \left( -\frac{\partial}{\partial x_2} L(x_0, \lambda, c) \right) + \dots + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{nn}} \left( -\frac{\partial}{\partial x_n} L(x_0, \lambda, c) \right) \right) \end{pmatrix} \quad 7.15$$

dividing (7.11) and (7.15) i.e.:

$$\alpha_0 = \frac{\left( \left( \frac{\partial}{\partial x_1} L(x_0, \lambda, c) \right)^2 + \left( \frac{\partial}{\partial x_2} L(x_0, \lambda, c) \right)^2 + \dots + \left( \frac{\partial}{\partial x_n} L(x_0, \lambda, c) \right)^2 \right)}{\begin{pmatrix} -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) \left( \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_1^2} \left( -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) \right) + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{12}} \left( -\frac{\partial}{\partial x_2} L(x_0, \lambda, c) \right) + \dots + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{1n}} \left( -\frac{\partial}{\partial x_n} L(x_0, \lambda, c) \right) \right) \\ -\frac{\partial}{\partial x_2} L(x_0, \lambda, c) \left( \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{21}} \left( -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) \right) + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_2^2} \left( -\frac{\partial}{\partial x_2} L(x_0, \lambda, c) \right) + \dots + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{2n}} \left( -\frac{\partial}{\partial x_n} L(x_0, \lambda, c) \right) \right) \\ \vdots \\ -\frac{\partial}{\partial x_n} L(x_0, \lambda, c) \left( \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{n1}} \left( -\frac{\partial}{\partial x_1} L(x_0, \lambda, c) \right) + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{n2}} \left( -\frac{\partial}{\partial x_2} L(x_0, \lambda, c) \right) + \dots + \frac{\partial^2 L(x_0, \lambda, c)}{\partial x_{nn}} \left( -\frac{\partial}{\partial x_n} L(x_0, \lambda, c) \right) \right) \end{pmatrix}} \quad 7.16$$

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

### 8. Data Analysis

Haven developed the algorithm for Combination of Penalty Function, the Lagrange Multiplier and Conjugate Gradient Methods (CPLCGM), we now apply the method to some constrained optimization problems which are pertain to quadratic functions. Some of these functions are subject to linear, nonlinear and inequality constraints. The table of results for these problems are shown below:

#### Problem

$$\text{Min } f = (x_1 - x_2)^2 + (x_2 + x_3 - 2)^2 + (x_4 - 1)^2 + (x_5 - 1)^2$$

$$\text{Subject to: } x_1 + 3x_2 = 0, x_2 - x_4 - 2x_5 = 0, x_2 - x_5 = 0$$

#### Problem2

$$\text{Mini } f = x_1^2 + 2x_2^2 + x_1 x_2 + 2x_1^2 + 3x_2^2$$

$$\text{Su bject to: } x_1^2 - x_2^2 - 1$$

Table-1. Table of Result for problem 1

It	X(1)	X(2)	X(3)	X(4)	X(5)	FV	Alfa	G*GT	Beta
0	2	2	2	2	70	0	0	0	
1	1.78571607	0.92858033	2.05357098	1.75893057	2.53570983	12.4379783	0.13392746e-1	8596	0
2	0.21699105	-0.46953752	4.20298643	0.2823569	-0.72421736	-8.42598613	0.51860834	80.4613535	0.9360325e-2
3	-0.3647372	-1.19577461	4.99007296	-0.24771488	-1.54482321	-2.83819616	0.10832661e-1	1248.13076	15.512177
4	-0.54175069	-1.39354356	5.32378142	-0.36434898	-1.5671256	1.73663368	0.19691487e-1	144.268189	0.1155874
5	-1.43238023	-1.69691014	7.68095155	-0.73402633	-1.38843562	4.66611907	0.10509827	115.292829	0.79915628

Table 1, shows the numerical solution of problem 2. Using LPCGM, The problem converged at the 4<sup>h</sup> iteration with:

Calculated value = 4.66611907

Analytical solution = 4.09300567

The calculated value compares favourably with the analytical solution. This shows that the scheme developed in this paper is quite effective as well.

Table-2. Table of Result for problem 2

It	X(1)	X(2)	FV	Alfa	G*GT	Beta
0	2	3	51	0	0	0
1	0.61690962	-0.97638484	-0.26078717	0.1728863	593	0
2	0.2220446e-15	-0.83333333	-1.08333333	0.24100618	6.82593409	0.1151085e-1

Table 2, shows the numerical solution of problem 2. Using LPCGM, The problem converged at the 4<sup>h</sup> iteration with: Calculated value = -1.08333333

Analytical solution = -1.210927261

The calculated value compares favourably with the analytical solution. This shows that the scheme developed in this paper is quite effective as well.

## **9. Conclusion**

Computationally, the resulting algorithm from the Combination of Penalty Function, the Lagrange Multiplier and Conjugate Gradient Methods was tested on some constrained optimization problems but in this paper, we recorded two as a sample of the problems tested. These problems are pertained to quadratic functions. Problem 1 and 2 are subject to linear and nonlinear constraints respectively.

On using the function value as the terminating criterion, Problem 1 with the numerical value 4.16661167 which almost coincide with the analytical result 4.0930 and problem 2 with the numerical result  $-1.08333333$  when compare with the analytical result  $-1.210927261$ , it invariably establishes the relevance of the new Algorithm for solving constrained optimization problems. All these points to the fact that, the constructed CPLCGM algorithm efficiently solve the problems and converges as supposed.

## **References**

- Hestenes, M.R., 1975. Optimization theory. New York: Wiley.
- Hestenes, M.R. and E. Stiefel, 1952. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standard*, 49(6): 409-436.
- Luenberger, D.G., 1973. Introduction to linear and nonlinear programming. Reading, MA: Addison-Wesley, 28.
- Pierre, D.A. and M.J. Lowe, 1975. Mathematical programming via augmented Lagrangian. Reading, MA: Addison-Wesley.
- Schmit, L.A. and C. Fleury, 1979. An improved analysis/synthesis capability based on dual methods-access 3. AIAA 20th Structures, Structural Dynamics, and Materials Conference. pp: 23-50.