

Numerical Experiment with the Introduction of a Scaling Factor to the Multiplier Method to Solve Constrained Optimization Problems

Adebayo Kayode James^{1*}, Dele-Rotimi Adejoke Olumide², Adisa Isaac Olabisi³, Alabi Taiye John⁴, and Ademoroti Albert Olalekan⁵

¹Department of Mathematics, Ekiti State University, Ado Ekiti, Ekiti State, Nigeria

²Department of Mathematics, Bamidele Olumilua University of Education, Science, and Technology, Ikere Ekiti, Ekiti State, Nigeria

³Department of Mathematics, Adeyemi University of Education, Ondo, Ondo State, Nigeria

⁴Department of Statistics, Kogi State Polytechnic, Lokoja, Kogi State, Nigeria

⁵Department of Physics, Bamidele Olumilua University of Education, Science, and Technology, Ikere Ekiti, Ekiti State, Nigeria

*Corresponding Author

DOI: <https://doi.org/10.51584/IJRIAS.2025.100700128>

Received: 30 May 2025; Accepted: 07 June 2025; Published: 21 August 2025

ABSTRACT

This paper discusses the intertwining of the Conjugate Gradient Method (CGM) as a scaling factor for Newton's algorithm, which is employed to determine the solution of the constrained optimization problems. The constrained problem has to be converted to an unconstrained problem via the multiplier method, after which a scaling is introduced to Conjugate Gradient Method (CGM) of Hestenes and Stiefel that forces the problem to collapse toward the resulting equation quickly. Newton's algorithm, which was embedded in the algorithm of the multiplier method, was used to carry out the minimization process. At each descent direction search, a formulated dynamic is used to update the Lagrange multiplier, such that this is done at each one-dimensional search until the optimality is reached. The use of this method was not limited to optimization problems alone, it has been extended to optimal control problems of the Lagrange and the Meyer forms, with a huge success recorded when the results obtained were compared with results of other existing methods.

Keywords: Scaling Factor, Multiplier Method, Constrained Optimization, Penalty Parameter, Conjugate Gradient Method (CGM), Penalty Function.

INTRODUCTION

In the study of a collection of techniques that is employed for finding solution to quadratic augmented Lagrangian based on constrained optimization problems, for which the functions of the multipliers hinge on the penalty parameters, this leads to Lagrangian that the multipliers are nonlinear. We equally studied some methods for unconstrained multivariate problems, such as the Conjugate Gradient Algorithm (CGM) and Newton's method, that are known for their reliability and swift convergence. The conjugate gradient method is preferred to the Newton's method because it converges very fast in a specified number of iteration than the Newton's method as supported by [1], but when a problem is subjected to one or more constraints, experience shows that the iterate of the conjugate gradient method may move away from the feasible region in its quest to locate an optimal path in a short time, thereby wasting computer time that is not regarded as a characteristics of a good algorithm according to [2] and [3].

On the other hand, Newton's method moves slowly in search of the global minimum, thereby generating more iterations. To balance the two sides, a non-differentiable conjugate gradient method of Hestenes and Stiefel is proposed by [4] and [5] as

$$f(x) = f_0 + \langle a, x \rangle + \frac{1}{2} \langle x, Hx \rangle \quad (1)$$

was taken into consideration for the multiplier method that is of the form

$$f(x) = \psi(x_n) + \lambda g(x_n) + \mu (gx_n)^2 \quad (2)$$

and Newton's method, which had been embedded in the multiplier algorithm (2), was considered for the minimization process of (1). Since the conjugate gradient method was used as the scaling factor for the multiplier method, and the algorithm of Newton's method is used to carry out the minimization process, the three methods shall be briefly discussed in the next sections.

Newton's method

Suppose $f(x)$ is approximated by a quadratic function at x^k such that the equation below holds for the function:

$$f(x) = f(x^k) + \nabla^T f(x^k) \nabla x^k + \frac{1}{2} (\nabla x^k)^T H(x^k) \nabla x^k \quad (3)$$

where $H(x^k)$ is the Hessian matrix of $f(x)$. The method makes use of the second order (quadratic) approximation of $f(x)$ at x^k and thus employs second order information about $f(x)$. The method takes into account the curvature of $f(x^k)$ at x^k in identifying better search directions than can be obtained via the gradient method as opined by [6]. The minimum of the quadratic approximation of $f(x)$ in (3) to each of the components of ∇x^k and equating the resulting expression to zero gives:

$$\nabla f(x) = \nabla f(x^k) + H(x^k) \nabla x^k = 0 \quad (4)$$

therefore

$$x^{k+1} - x^k = \nabla x^k = -[H(x^k)]^{-1} \nabla f(x^k) \quad (5)$$

If $f(x)$ is quadratic, only one step is required to reach the minimum of $f(x)$. However, for a general nonlinear objective function, the minimum of $f(x)$ cannot be reached in one step, therefore the Newton's method has to be modified so that (5) can be adjusted to conform to the form

$$f(x) = f(x^k + \alpha s^k) = f(x^k) + \nabla^T f(x^k) \alpha s^k + \frac{1}{2} (\alpha s^k)^T H(x^k) \alpha s^k \quad (6)$$

where $\nabla x^k = \alpha s^k$

if $\frac{df}{d\alpha}(x^k + \alpha s^k) = 0 = \nabla^T f(x^k) s^k + (s^k)^T H(x^k) s^k$ then,

$$\alpha = \frac{-\nabla^T f(x^k) s^k}{(s^k)^T H(x^k) s^k} \quad (7)$$

By introducing the parameter for the step length into (5), then (5) becomes

$$x^{k+1} - x^k = -\alpha^k [H(x^k)]^{-1} \nabla f(x^k) \quad (8)$$

The authors in [7] observed that the direction s is now given by $s^k = [H(x^k)]^{-1} \nabla f(x^k)$ and the step length α^k (α is a scalar denoting the distance moved along the search direction) and (8) is applied iteratively until the terminating criteria is/are satisfied [8].

The Conjugate Gradient Method

The Conjugate Gradient Method (CGM) by [9] is one of the variations of the gradient method. In its simplest form, the gradient method uses the iterative scheme:

$$x_{i+1} = x_i - \alpha \nabla f(x) \quad (9)$$

to generate a vectors sequence, $\{x_i\}_{i=1}^n$, that converges to the minimum of $f(x)$ achieved on minimizing the function

$$f(x) = \frac{1}{2} x^T A x - b^T x \quad (10)$$

The parameter α that appears in (9) is referred to as the step length of the descent direction sequence. In particular, if F is a function on the Hilbert space \mathcal{H} such then in the Hilbert space F admits a Taylor series expansion given as:

$$f(x) = f_0 + \langle a, x \rangle_{\mathcal{H}} + \frac{1}{2} \langle x, Qx \rangle_{\mathcal{H}} \quad (11)$$

Where $a, x \in \mathcal{H}$ and \mathcal{H} is a positive definite matrix that is symmetric and called the linear operator according to [10]. It can be shown by [11] that it possesses a unique minimum x^* in \mathcal{H} , and that the gradient of $f(x)$ at the minimum, $\nabla f(x^*) = 0$. The CGM algorithm for iteratively locating the minimum x^* of in \mathcal{H} as described by [11] is as follows:

Step 1: The first element $x_0 \in \mathcal{H}$ is guessed while the remaining members of the sequence are computed with the aid of the formulae in steps 2 through 6.

Step 2: The descent direction is computed using $p_0 = -g_0$ (12)

Step 3: Set $x_{i+1} = x_i + \alpha_i p_i$; where the step length $\alpha_i = \frac{\langle g_i, g_i \rangle_{\mathcal{H}}}{\langle p_i, Qp_i \rangle_{\mathcal{H}}}$ (13)

Step 4: Update the gradient value using the dynamic $g_{i+1} = g + \alpha_i Qp_i$ (14)

Step 5: Set $p_{i+1} = -g_{i+1} + \beta_i p_i$; $\beta_i = \frac{\langle g_{i+1}, g_{i+1} \rangle_{\mathcal{H}}}{\langle g_i, g_i \rangle_{\mathcal{H}}}$ (15)

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

In the iterative steps 2 through 6 above, denotes the descent direction at the i -th step of the algorithm, is the step length of the descent sequence $\{x_i\}$, and denotes the gradient of F at steps 3, 4, and 5 of the algorithm reveal the crucial role of the linear operator Q in determining the step length of the descent sequence and also in generating a conjugate direction of search.

The conjugate gradient method is an iterative method that enjoys the quadratic convergence, ensuring that it converges in at most n iterations if there is no rounding off errors are encountered. Starting with an initial estimates x_0 of the solution h , one determines successfully new estimate value for x_0, x_1, x_2, \dots of k .

The estimate x_i is a value closer to h than x_{i+1} ; hence, at each step of the iteration, the residual, $r_i = k - Ax_i$, is computed. Normally, this vector can be used to determine the usability of the estimate x_i . However, this measure is not a reliable one because it is possible to construct cases in which the squared residual $|r_i|^2$ increases at each step while the length of the error vector, $h - x_i$, will decrease, forcing the round-off error to always occur except under very unusual circumstances.

The Multiplier Method

The method of multipliers or the augmented reduces the possibilities of ill conditioning by introducing explicit Lagrange multiplier estimates into the function to be optimized. In contrast to the penalty function, the

multiplier method largely preserves smoothness and guaranteed optimality within a short time without necessarily drag the penalty parameter to infinity [12].

For clarity, let us consider the equality constraint problem below:

Minimize $\emptyset(x)$,

Subject to $h(x)$ (16)

The quadratic penalty function $Q(x, \mu)$ is set to penalizes the constraint violations by squaring the infeasibilities and scaling them by $\frac{\mu}{2}$, however, the approximate minimizers x_k of $Q(x, \mu_k)$ do not quite satisfy the feasible conditions $c_i(x) = 0$, for $i = 1, 2, \dots, m$ instead, they are perturbed so that

$$c_i(x_k) = -\frac{\lambda^i}{\mu_k} \text{ for } i = 1, 2, \dots, m \quad (17)$$

just to be sure that $C_i(x) \rightarrow 0$ as $\mu_k \rightarrow \infty$. To avoid this systematic perturbation, [13] observed that the multiplier function $\ell(x, \lambda, \mu)$ achieves this goal by including an explicit estimate of the Lagrange multiplier λ , based on the modified objective function

$$\ell(x, \lambda, \mu) = f(x_k) - \sum_{i=1}^m \lambda_i^k + \sum_{i=1}^m c_i^2 \quad (18)$$

$$0 = \nabla_x \ell(x_k, \lambda^k, \mu_k) = \nabla f(x^*) - [\sum_{i=1}^m \lambda_i^k + \sum_{i=1}^m c_i^2] \nabla c_i(x_k) \quad (19)$$

By comparing (17) with

$$\nabla f(x_k) - A(x^*)^T \lambda^* = 0 \quad (20)$$

We have the updated formula

$$\lambda_i^* \approx \lambda_i^k - \mu_k c_i(x_k) \text{ for } i = 1, 2, \dots, m \quad (21)$$

By rearranging this expression, we have $c_i(x_k) \approx -\frac{1}{2}(\lambda_i^* - \lambda_i^k)$, so we conclude that if λ^k is close to the optimal multiplier vector λ^* [14]. The infeasibility in x_k will be much smaller than $\frac{1}{\mu_k}$, rather than being proportional to $\frac{1}{\mu_k}$ the relation (21) immediately suggests a formula for improving our current estimate λ^k of the multiplier vector using the approximate minimum x_k [15], one then can set $\lambda_i^{k+1} \approx \lambda_i^k - \mu_k c_i(x_k)$ for all $i \geq 1$.

Theorem 1:

Let x^* be the local solution of (16) at which the LICQ is satisfied (i.e., the gradients $\nabla c_i(x^*)$ for $i = 1, 2, \dots, m$ are linearly independent vectors) and the second order conditions are satisfied when $\lambda = \lambda^*$. Then, there is a threshold value $\bar{\mu}$ such that for all $\mu \geq \bar{\mu}$, λ^* is a restricted local minimum of $\ell(x, \lambda^*, \mu)$ as opined by [16].

Proof

We prove the result by showing that x^* satisfies the second-order sufficient conditions to be a strict local minimum of $\ell(x, \lambda^*, \mu)$ for all μ that are sufficiently large, i.e.,

$$(i) \quad \nabla_x \ell(x^*, \lambda^*, \mu) = 0 \quad (22)$$

$$(ii) \quad \nabla_{xx}^2 \ell(x^*, \lambda^*, \mu) = 0 \quad (23)$$

Since x^* is a local solution of (8) at which the LICQ is satisfied, the sufficient condition can be applied to deduce that $\nabla_x \ell(x^*, \lambda^*, \mu) = 0$ and $c_i(x^*) = 0, i = 1, 2, \dots, m$, so that

$$\nabla_x \ell(x^*, \lambda^*, \mu) = \nabla f(x^*) - \sum_{i=1}^m [\lambda_i^* - \mu c(x^*)] \nabla c_i(x^*) = 0 \quad (24)$$

For the second part of (23), we write

$$\nabla_{xx}^2 \ell(x^*, \lambda^*, \mu) = \nabla_{xx}^2 \ell(x^*, \lambda^*, \mu) + \mu A^T A \quad (25)$$

where A is the constraint gradient matrix evaluated at x^* . If the claim in (25) is not true, then for each integer $k \geq 1$, one could choose a vector w_k with $\|w_k\| = 1$ such that

$$0 \geq w_k^T \nabla_{xx}^2 \ell(x^*, \lambda^*, \mu) w_k = w_k^T \nabla_{xx}^2 \ell(x^*, \lambda^*, \mu) w_k + k \|Aw_k\|_2^2 \leq 0 \quad (26)$$

Therefore, as $k \rightarrow \infty$ (2.12) gives rise to

$$\|Aw_k\|_2^2 \leq -\left(\frac{1}{k}\right) w_k^T \nabla_{xx}^2 \ell(x^*, \lambda^*, \mu) w_k \rightarrow 0 \quad (27)$$

Since the vectors $\{w_k\}$ lie within the compact set, they could have an accumulation point w , then the limit of (27) implies that $Aw = 0$. More so, by rearranging (26), one arrives at

$$w_k^T \nabla_{xx}^2 \ell(x^*, \lambda^*, \mu) w_k = -k \|Aw_k\|_2^2 \leq 0 \quad (28)$$

So, taking the limits of both sides of (28) gives rise to $w_k^T \nabla_{xx}^2 \ell(x^*, \lambda^*, \mu) w_k \leq 0$. However, this inequality contradicts the second-order conditions, such that when it is applied to (8), it states that one must have $w_k^T \nabla_{xx}^2 \ell(x^*, \lambda^*, \mu) w_k \leq 0$ for all non-zero vector w with $Aw = 0$. Hence, the second part of (23) holds for all μ sufficiently large.

SCALED MULTIPLIER METHOD

The scaled multiplier method preserved the smoothness of the Newton's method [17], fast convergence of the Conjugate gradient method [18] and robustness of the Multiplier method [5] to form an algorithm that is free of ill conditioning and does not necessarily drag the penalty parameter to infinity before optimality is guaranteed. Unlike the scaling of the entire process in [19], the SMM is all about introducing the scaling factor into the classical multiplier method.

The method transformed a constrained problem of the type (16) to an unconstrained one of the form (2), the Function (1) of Hestenes and Stiefel [20] as a scaling factor for the function (2). Newton's method for solving systems of necessary optimality conditions, i.e., Newton's Method for solving unconstrained optimization problems, was considered for the minimization, and the updating of λ was embedded into each iteration of Newton's Method until some indicated conditions for optimality are satisfied.

The techniques of changing λ at each one-dimensional search were tried for many problems with great success. The complete iterative procedure of the method after transformation of the constrained problem to an unconstrained one can be summarized as follows:

- (i). Scale the multiplier method to conform with the function $f_0 + \langle a, x \rangle + \frac{1}{2} \langle x, Hx \rangle$, where f_0 is the function value of at x_0 , H is a positive definite symmetric linear operator, while a is in Hilbert space.
- (ii). Knowing the initial guess, choose $\langle a, Hx \rangle$ as the gradient, where H is a function of λ and μ
- (iii) Compute $-[H^{-1}(x)][\langle a, Hx \rangle]$
- (iv) Update $x_{i+1} = x_i - [H^{-1}(x)][\langle a, Hx \rangle]$

(v) Use the formula $\lambda_i^{k+1} \approx \lambda_i^k - \mu_i c_i(x_k)$, to update the value of λ where constraint equation(s) is given as c.

Steps (i) through (v) have to be repeated until either of the specified condition are met, $c_i \rightarrow 0, \langle a, xH \rangle \approx 0$.

Data and Analysis

Having itemized the algorithm of the Scaled Multiplier method, we now apply the method to the following constrained optimization problems:

Problem 1:

Minimized $\phi(x) = 3x^2 + y^2 + 2xy + 6x + 2y$

subject to $h(x) = 2x - y - 4$

Problem 2:

Minimized $\phi(x) = (x - 3)^2 + (y - 4)^2$

subject to $h(x) = 2x + y - 3$

Problem 3:

Minimized $\phi(x) = 3x^2 + y^2 + 2xy + 6x + 2y$

subject to $h(x) = x^2 - y - 1$

Comparison of the Numerical Results

To ensure a wide range of comparison, the Scaled Multiplier Method (SMM) will be compared with other methods like Penalty Function Method (PFM), Lagrange Multiplier (LM), and ERKM in [21] with ERKM is close to SMM because of its modification but SMM is doing credibly well due to the scaling factor. The comparison will only reflect the iteration at which the result of each of the methods coincide and the values of λ^* , $\phi(x^*)$, and $\psi(x^*)$ at that iteration.

Table 1: Table of results for Problem 1

Methods	Iterations	μ	x^*	y^*	λ^*	$\phi(x^*)$	$\psi(x^*)$
<i>SMM</i>	7	1	0.63625	-2.72711	-2.180923	3.525362	3.54545
<i>ERKM</i>	9	1	0.63635	-2.72723	-2.181803	3.545368	3.54568
<i>PFM</i>	11	1000	0.62835	-2.73723	-2.183885	3.442365	3.70268
<i>LM</i>	14	—	0.63625	-2.72700	-2.179156	3.524764	3.53467

$x^* = 0.65$ and $y^* = -0.27$

Table 2:

Methods	Iterations	μ	x^*	y^*	λ^*	$\phi(x^*)$	$\psi(x^*)$
<i>SMM</i>	10	1	0.18998	2.68437	2.81027	9.78768	9.85657
<i>ERKM</i>	12	1	0.19998	2.59998	2.80002	9.79921	9.80001
<i>PFM</i>	25	1000	0.18989	2.58999	2.93541	9.72311	9.81011
<i>LM</i>	22	—	0.63625	2.72700	-2.17915	3.524764	3.53467

$x^* = 0.2$ and $y^* = 2.3$

Table 3:

Methods	Iterations	μ	x^*	y^*	λ^*	$\phi(x^*)$	$\psi(x^*)$
<i>SMM</i>	8	1	-0.734257	-0.43852	-0.22531	-1.10463	-1.22453
<i>ERKM</i>	21	1	-0.749985	-0.43748	-0.24987	-1.21095	-1.21093
<i>PFM</i>	27	1000	-0.737500	-0.41785	-1.92587	-1.20031	-1.21098
<i>LM</i>	25	—	-0.636257	-1.72700	-2.17915	3.524764	3.53467

$x^* = 0.75$ and $y^* = -0.45$

CONCLUSION

Unlike using the ERKM, PFM, and LM, the SMM behaves relatively well in terms of fewer iterations to reach the convergence point, relatively very low penalty parameter, μ gives one the desired result against situations where large penalty function values have to be used in PFM; however, the PFM has to go through many iterations before its result coincides or comes close to the existing results.

This paper clearly shows the derivation and numerical implementation of the scaling factor in the multiplier method, which interprets the new technique for its easy applicability to solve constrained optimization problems. In the near future, one hopes to devote more attention to the method's application to solve optimization problems with equality and inequality constraints.

REFERENCES

1. Nash, S. G. and Sofer, A., (1995), Linear and Nonlinear programming, McGraw-Hill, New York, 1995.
2. Boggs, P. T., Kearsley, A. J., and Tolle, J. W., (1999), Practical algorithm for general large- scale Nonlinear optimization problems, SIAM J. Opt. 9(3), 755-778.
3. Nocedal, J. and Wright S. J., (1999), Numerical Optimization, Dover Publishing Company, New York, (U.S.A).
4. Brent R. P., (1973), Algorithm for minimization without derivatives Prentice-Hall, Englewood cliffs, New Jersey.
5. Hestenes, R. T., (1969), Multiplier and gradient method, Journal of optimization theory and applications, Vol.4, No5, 346-463.
6. Rockafella, R. T., (2005), Multiplier method of Hestenes and Powell applied to convex programming, Journal of Optimization theory and applications, Vol.4, No. 4.
7. Aderibigbe, F. M., Dele-Rotimi, A. O., and Kayode James Adebayo, (2015), On Application of a New Quasi-Newton Algorithm for Solving Optimal Control Problems. Pure and Applied Mathematics Journal. Vol. 4, No. 2, pp. 52-56. doi: 10.11648/j.pamj.20150402.14.
8. Schuidy, S. B., (1992), A method of multiplier for Mathematical programming problems with equality and Inequality constraints, Journal of optimization theory and applications, Vol. 4, No 5.
9. Oke, M. O., Oyelade, T. A., Adebayo, K. J., and Adenipekun, E. A., (2021), On a Modified Higher Order Form of Conjugate Gradient Method for Solving Some Optimal Control Problem, Global Science Journal, Volume 9, Issue 6, pp 244-251.
10. Adebayo, K. J., Aderibigbe, F. M., Ayinde, S. O., Olaosebikan, T. E., Adisa, I. O., Akinmuyise, F. M., Gbenro, S. O., Obayomi, A. A., and Dele-Rotimi, A. O., (2024), On the Introduction of a Constructed Operator to an Extended Conjugate Gradient Method (ECGM) Algorithm, Journal of Xi'an Shiyou University, Natural Science Edition, Vol., 20 (04), pp. 562-570.
11. Hasdorff, L., (1976), Gradient Optimization and Nonlinear Control, J. Wiley and Sons, New York.
12. Tripathi, S. S. and Narendra, K. S., (1972), Constrained optimization using multiplier method, Journal of optimization theory and applications, vol. 9, No1, 59-63.
13. Olorunsola, S. A., Olaosebikan, T. E., and Adebayo, K. J., (2014), Numerical Experiments with the Lagrange Multiplier and Conjugate Gradient Methods (ILMCGM). American Journal of Applied Mathematics. Vol. 2, No. 6, pp. 221-226. doi: 10.11648/j.ajam.20140206.15

14. Conn, A. R., Gould, N. I. M., and Toint, ph. L., (1991), A global convergent Augmented Lagrangian Algorithm for Optimization with general constraints and simple bounds, SIAM J. Num. Anal. Vol. 28, No.2, 545-572.
15. Snyman, J. A., (2005), Practical Mathematical Optimization, Prentice-Hall, Englewood cliffs, New Jersey.
16. Yamamoto, T., (2000), Optimization and Nonlinear equations, Journal of computational and applied Mathematics 124, 9-19.
17. Thomas, E. E., Himmelblau, D. M., and Lasdon, L. S., (2001), Optimization of chemical process, McGraw-Hill, New York.
18. Powell, M. J. D., (1969), A method for nonlinear constraint in minimization problems (R. Fletcher, ed), Academic press, New York.
19. Adebayo, K. J., Ademoroti, A. O., Ayinde, S. O., Akinmuyise, M. F., and Olaosebikan, T. E., (2025), Solution to Bolza Form Continuous-Time Linear Regulator Problems (CLRP) Using Scaled Extended Conjugate Gradient Method (SECGM) Algorithm, Journal of Emerging Technologies and Innovative Research (JETIR), Volume 12, Issue 5, 493-501.
20. Hestenes, M. R., and Stiefel, (1952), Method of Conjugate gradient method for solving Linear system, J. Res, Natl Bur. Stand. 49, 409-436.
21. Akinmuyise, Mathew Folorunsho, Adebayo, Kayode James, Adisa, Olabisi, Olaosebikan, Emmanuel Temitayo, Gbenro, Sunday Oluwaseun, Dele-Rotimi, Adejoke Olumide, Obayomi, Abraham Adesoji, Ayinde, Samuel Olukayode, (2024), Extended Runge-Kutta Method (ERKM) Algorithm for the Solution of Optimal Control Problems (OCP), Vol. 19, Issue 2, pp. 471-478, <https://doi.org/10.5281/zenodo.10049652#180>.