Newton method with explicit group iteration for solving large scale unconstrained optimization problems

K Ghazali1,a, J Sulaiman1,b, Y Dasril2,c, and D Gabda1,d
1Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia
2Universiti Teknikal Malaysia Melaka, 76100 Melaka, Malaysia
E-mail: akhadizah@ums.edu.my, bjumat@ums.edu.my, cyosza@utem.edu.my, ddarmesah@ums.edu.my

Abstract. Numerous numerical methods have been used to solve unconstrained optimization problems. The Explicit Group iteration is one of the numerical methods that has an advantage of the efficient block iteration scheme for solving any linear system. In this paper we applied a combination of the Newton method with two-point Explicit Group (2-point EG) iterative scheme for solving large scale unconstrained optimization problems. In order to evaluate the performance of this method, combination between Newton method with classical Iterative methods namely Jacobi and Gauss-Seidel iterations were used as reference method. The numerical results show that our proposed method is more efficient than the reference method in terms of execution time, number of iteration and absolute error.

1. Introduction
Large scale unconstrained optimization problems are one of the most challenging problems in applied mathematics and physics which extensively and increasingly used in engineering, medical, computational economics, finance, marketing, telecommunications and manufacturing. Most of these problems arise from modeling systems with a demanding structure can be solved by different methods such as steepest descent method [1], Newton’s method [2], modified Newton method [3], Levenberg-Marquardt’s method [4], conjugate gradient method [5], Quasi-Newton method [6], Boryden-Fletcher-Goldfarb-Shanno method [7], Powell’s method [8] and the Nelder-Mead algorithm [9]. Choosing method for large scale unconstrained optimization may vary according to how much information on the characteristics of the objective function and its gradient for any point are accessible. A better method can be chosen if the second order derivatives are exists. Otherwise, finite difference or automatic differentiation can be used. Although all of these methods can solve large scale unconstrained optimization problems, only Newton method will be discussed in this paper. This is due to the fact that Newton method is the best known method for its good performance when the starting point is chosen appropriately and also will yield the best convergence properties which is quadratic convergence.

For that reasons, numerous researchers also deal with this method [10-25]. Moré and Sorensen [10] have been presented and explored on Newton’s method for unconstrained minimization which are appropriate for large scale problem. They pointed out that it is possible to reduce amount of work and storage for the Newton method by using the Cholesky decomposition of symmetric matrix. In that regard, Gundersen and Steihaug [11] also shown that the ratio of the number of arithmetic operation of Newton’s method is constant per iteration for a large class of sparse matrices. Other than that, the basic idea of Newton method for unconstrained optimization have been discussed in [12-16] and noted that by using the quadratic approximation to the objective function with the starting point is close to the
optimum the Newton method will converge quickly. However, when the starting point is far away from
the optimum, the Newton method is not guaranteed to converge. In particular, the method may not be a
descent method. Therefore, Sorensen [17], Goldstein and Price [18], Sisser [19] and Dasril et. al. [20]
have modified the Newton iteration so that the descent property holds. Furthermore, as a result of
modifying Newton’s method several methods are proposed such as Newton-type method [21], truncated
Newton method [22], nonmonotone Newton’s method [23] and inexact Newton’s method [24].

Apart from discussing family of modified Newton methods, Lin et. al. in [26] and Lin and Lin in
[27,28] have discussed a different approach than what has been discussed in [10-25]. They have
proposed a method by combining an approximate scale gradient method with block iterative method
to solve large scale unconstrained optimization problems. Lin et. al. in [26] presented an efficient descent
algorithm for a class of unconstrained optimization problems of nonlinear large mesh-interconnected
systems by combining an approximate scale gradient method with textured decomposition-based block
Gauss-Seidel (TDBGGS) method. Similarly in [27], they presented a new efficient method for solving
large scale unconstrained optimization problems by combining an approximate scaled gradient method
with a block Gauss-Seidel with line search (BGSLG) method. By using the idea of this approach as well,
in this paper, we have proposed a method by combining the Newton method with 2-point EG iterative
methods, namely Newton-EG for solving a large scale unconstrained optimization problems. This
combination is motivated by the advantage of the EG iterative method which is known as one of the
efficient block iterative methods which have been demonstrating by Evan [29], Yousef and Evans
[30,31], Abdullah [32] and Othman and Abdullah [33].

To investigate the capability of Newton-EG method, let us consider a large scale unconstrained
optimization problem be defined as

\[
\min_{x \in \mathbb{R}^n} f(x)
\]  

(1)

where the objective function \( f: \mathbb{R}^n \rightarrow \mathbb{R} \) is twice continuously differentiable. In order to solve problem
(1) by using the Newton method, an important search direction name as Newton direction, can be
obtained by solving the linear system involving the Hessian matrix and the negative gradient. Thus, in
this paper, we approximate Newton direction using an EG iterative method as an inner iteration and find
an approximate solution for problems (1) by using Newton method as an outer iteration. Then, for the
purpose of comparison, this paper also considered Jacobi and Gauss-Seidel iterative methods and
namely Newton-Jacobi and Newton-GS.

2. Formulation of Newton Scheme

In this section, we will describe Newton iterative method using the basic idea of Newton method for
solving large unconstrained optimization by iteratively by using the second-order quadratic
approximation to the objective function \( f(x) \) in problem (1) at the current iterate \( x^{(k)} \). In other words,
\( f(x) \) can be approximated directly from the Taylor series using the first three terms as

\[
f(x) \approx f(x^{(k)}) + \nabla f(x^{(k)})^T (x - x^{(k)}) + \frac{1}{2} (x - x^{(k)})^T H(x^{(k)})(x - x^{(k)}),
\]

(2)

where \( \nabla f(x^{(k)}) \) is the gradient of \( f(x) \) and \( H(x^{(k)}) = \nabla^2 f(x^{(k)}) \) is the Hessian matrix of \( f(x) \). Since
approximation (2) is a quadratic function, we can simply find the minimizer of approximation (2) by
differentiating it with respect to \( x \) and equating the resulting expression to zero. Note that \( H(x^{(k)}) \) is a
symmetric matrix, hence we can have;

\[
\nabla_x \left( f(x^{(k)}) + \nabla f(x^{(k)})^T (x - x^{(k)}) + \frac{1}{2} (x - x^{(k)})^T H(x^{(k)})(x - x^{(k)}) \right) = 0
\]

\[
\Rightarrow \nabla f(x^{(k)}) + H(x^{(k)})(x - x^{(k)}) = 0
\]
In equation (3), the new iteration is given by substituting $x = x^{(k)}$ as:

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

(4)

where $H(x^{(k)})^{-1}$ is the inverse of the Hessian matrix $H(x^{(k)})$. Notice that, by letting $x^{(k+1)} = x^{(k)} + d^{(k)}$ in equation (4) such as

$$d^{(k)} = -H(x^{(k)})^{-1} \nabla f(x^{(k)})$$

(5)

will give us the Newton direction. Obviously, this Newton direction is a descent direction because it satisfies

$$\nabla^T f(x^{(k)}) d^{(k)} = - \nabla^T f(x^{(k)}) H(x^{(k)})^{-1} \nabla f(x^{(k)}) < 0$$

(6)

if $H(x^{(k)})$ is positive definite. Equation (5) also can be written as

$$H(x^{(k)})d^{(k)} = -\nabla f(x^{(k)})$$

(7)

3. Derivation of proposed Iterative Methods

Based on linear system in equation (7), it can be seen that the characteristic of its coefficient matrix has large scale. Therefore, using direct method to solve it is not a smart choice to store and factor the dense Hessian matrix. Thus, we proposed method by using iterative method as in [34,35]. Since equation (7) is known as a linear system, let the linear system be rewritten as

$$Ad = b$$

(8)

where,

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{3,1} & \ldots & a_{1,n} \\ a_{1,2} & a_{2,2} & a_{3,2} & \ldots & a_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{1,n} & a_{2,n} & a_{3,n} & \ldots & a_{n,n} \end{bmatrix}, \quad d = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$
in which $D$ is the nonzero diagonal entries of $A$, $L$ is strictly lower matrix and $U$ is strictly upper matrix. Then, the general form of the Gauss-Seidel iterative method for solving the linear system (8) can be stated in vector form as

$$d^{(k+1)} = (D - L)^{-1} Ud^{(k)} + (D - L)^{-1} b$$

(10)

For the implementation of point iterations, the formulation of Gauss-Seidel iterative method and equation (10) can be represented the $i$th component of the vector $d_i$ yields

$$d_i^{(k+1)} = \frac{1}{a_{i,i}} \left( b_i - \sum_{j=1}^{i-1} a_{i,j} d_j^{(k+1)} - \sum_{j=i+1}^{n} a_{i,j} d_j^{(k)} \right), i = 1,2,...,n$$

(11)

Clearly, the formulation in equation (11) can be categorized as one of the point iterative methods. Based on the concept of this point Gauss-Seidel iterative method, the next subsection will discuss the formulation of block Gauss-Seidel method which is known as the Explicit Group (EG) iterative method.

3.2. Formulation of 2-point Explicit Group (EG) Iterative Method

As we know that Evans [29] has also proposed four point block iterative methods via the Explicit Group iterative method for solving large linear systems. Due to the advantages of the implementation of block iterations, this paper is to examine the performance of 2-point Newton-EG iterative method for solving a large linear system generated by imposing the Newton method to large scale unconstrained optimization problem (1). Therefore, this section shows on how to derive the formulation of 2-point Newton-EG iterative method. To derive the formulation of the proposed iterative method, let us consider a group of two points from the linear system (8) as follows [36,37];

$$\begin{bmatrix} a_{i,i} & a_{i,i+1} \\ a_{i+1,i} & a_{i+1,i+1} \end{bmatrix} \begin{bmatrix} d_i \\ d_{i+1} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}$$

(12)

where,

$$S_t = b_{i+t-1} - \sum_{j=1}^{i-1} a_{i+t-1,j} d_j^{(k+1)} - \sum_{j=i+1}^{n} a_{i+t-2,j} d_j^{(k)}, t = 1,2.$$ 

Hence, solving and simplifying equation (12) gives that the general form of the 2-point EG iteration can be stated as;

$$d_i^{(k+1)} = \frac{a_{i+1,i+1} S_1 - a_{i,i+1} S_2}{\alpha},$$

$$d_{i+1}^{(k+1)} = \frac{a_{i,i} S_2 - a_{i+1,i} S_1}{\alpha},$$

(13)

where,

$$\alpha = (a_{i,i})(a_{i+1,i+1}) - (-a_{i,i+1})(-a_{i+1,i})$$

Therefore, by using equation (8) and (13), we proposed the algorithm of 2-point Newton-EG iterative method for solving problem (1), as stated in Algorithm 1.
Algorithm 1. Newton-EG Scheme

i. Initialize
   Set up the objective function: $f(x), f(x^k) \in \mathbb{R}, x^{(0)} \in \mathbb{R}^n, \varepsilon_1 \leftarrow 10^{-5}, \varepsilon_2 \leftarrow 10^{-10}$,
   $n \leftarrow \{1000, 5000, 10000, 20000, 30000\}$.

ii. For $j = 1, 2, \ldots, n$, implement
   a. Set $d^{(0)} \leftarrow 0$
   b. Calculate $f(x^{(k)})$
   c. For $i = 1, 2, \ldots, n$, calculate iteratively
      Solve equation (8) by using equation (13),
   d. Check the convergence test, $\|d^{(k+1)} - d^{(k)}\| < \varepsilon_2$. If yes, go to step (e). Otherwise go back to step (b)
   e. For $i = 1, 2, \ldots, n$, calculate
      $x^{(k+1)} \leftarrow x^{(k)} + d^{(k)}$
   f. Check the convergence test, $\|\nabla f(x^{(k)})\| \leq \varepsilon_1$. If yes, go to (iii). Otherwise go back to step (a)

iii. Display approximate solutions.

4. Numerical Experiments

Based on algorithm 1, three well-known test functions stated in [38-40] have been tested to compare the efficiency of the combination algorithm between Newton and three different iterative methods such as Newton-Jacobi, Newton-GS and Newton-EG. The efficiency of these iterative methods are evaluated based on the comparison of number of iterations (inner and outer), the execution time to reach the minimum points and absolute error for all the methods. The three test functions are labelled Example 1, Example 2 and Example 3 as follows;

**Example 1**: Dixon and Price Function [38]

$$f(x) = (x_1 - 1)^2 + \sum_{i=2}^{n} ((2x_i^2 - x_{i-1}^2)^2$$

This function has a global minimum, $f^* = 0$ at $x_i^* = 2^{-\frac{1}{2^{i-2}}}$, for $i = 1, 2, \ldots, n$ with Figure 1 shows the graph of this function when $n = 2$.

**Example 2**: Strait Function [39]

$$f(x) = \sum_{i=1}^{n-1} (x_{i+1} - x_i^2)^2 + 100(1 - x_i)^2$$

This function has a global minimum, $f^* = 0$ at $x_i^* = 1$, for $i = 1, 2, \ldots, n$ with Figure 2 shows the graph of this function when $n = 2$.

**Example 3**: Rosenbrock’s Function [40]

$$f(x) = \sum_{i=1}^{n-1} (x_i - 1)^2 + 100(x_{i+1} - x_i^2)^2$$

This function has a global minimum, $f^* = 0$ at $x_i^* = 1$, for $i = 1, 2, \ldots, n$ with Figure 3 shows the graph of this function when $n = 2$. 

The numerical calculations are compiled by using C language (Borland C++). Each of the test functions is tested with three different initial points, $\chi^{(0)}$ (stated in Table 1) which is closer to the solution point, $\chi^*$ and run for number of different order of Hessian matrix as $n = 1000, 5000, 10000, 20000$ and $30000$. All outer iterations considered the same stopping criterion for the outer iterative $\|\nabla f(\chi)\| < \varepsilon_1$, where $\varepsilon_1 = 10^{-5}$. While the stopping criterion of inner iteration was set to be $\varepsilon_2 = 10^{-10}$. The efficiency comparison results for the execution time (seconds) are presented in Table 1 and the numerical results are tabulated in Table 2.

### Table 1. Comparison in term of execution time (seconds).

| Example | $\chi^{(0)}$ | method | $n$ | 1000 | 2000 | 10000 | 20000 | 30000 |
|---------|--------------|---------|-----|------|------|-------|-------|-------|
| 1       | (0.5,0.5,...,0.5,0.5) | Newton-J | 0.01 | 0.04 | 0.08 | 0.16 | 0.23  |
|         |              | Newton-GS | 0.01 | 0.04 | 0.07 | 0.13 | 0.19  |
|         |              | Newton-EG | 0.01 | 0.02 | 0.04 | 0.07 | 0.09  |
|         | (1.0,1.0,...,1.0,1.0) | Newton-J | 0.01 | 0.05 | 0.10 | 0.20 | 0.30  |
|         |              | Newton-GS | 0.01 | 0.05 | 0.09 | 0.18 | 0.25  |
|         |              | Newton-EG | 0.00 | 0.02 | 0.04 | 0.09 | 0.12  |
|         | (0.8,1.1,...,0.8,1.1) | Newton-J | 0.02 | 0.08 | 0.14 | 0.28 | 0.41  |
|         |              | Newton-GS | 0.02 | 0.07 | 0.13 | 0.25 | 0.34  |
|         |              | Newton-EG | 0.00 | 0.03 | 0.06 | 0.11 | 0.16  |
| 2       | (0.5,0.5,...,0.5,0.5) | Newton-J | 0.01 | 0.01 | 0.02 | 0.03 | 0.06  |
|         |              | Newton-GS | 0.00 | 0.01 | 0.01 | 0.02 | 0.05  |
|         |              | Newton-EG | 0.00 | 0.00 | 0.01 | 0.01 | 0.03  |
|         | (1.5,1.5,...,1.5,1.5) | Newton-J | 0.01 | 0.02 | 0.03 | 0.04 | 0.07  |
|         |              | Newton-GS | 0.00 | 0.01 | 0.02 | 0.03 | 0.06  |
|         |              | Newton-EG | 0.00 | 0.00 | 0.01 | 0.02 | 0.02  |
|         | (0.5,1.5,...,5.1,5.1) | Newton-J | 0.01 | 0.02 | 0.02 | 0.04 | 0.05  |
|         |              | Newton-GS | 0.00 | 0.01 | 0.02 | 0.03 | 0.04  |
|         |              | Newton-EG | 0.00 | 0.00 | 0.01 | 0.02 | 0.03  |
| 3       | (1.2,1.2,...,1.2,1.2) | Newton-J | 0.74 | 3.68 | 6.99 | 13.74 | 20.45 |
|         |              | Newton-GS | 0.50 | 2.68 | 4.86 | 10.02 | 15.10 |
|         |              | Newton-EG | 0.10 | 0.46 | 0.90 | 1.78  | 2.54  |
|         | (1.5,1.5,...,1.5,1.5) | Newton-J | 1.05 | 5.78 | 10.35 | 20.54 | 29.01 |
|         |              | Newton-GS | 0.81 | 3.94 | 8.14  | 14.44 | 21.82 |
|         |              | Newton-EG | 0.13 | 0.56 | 1.03  | 1.98  | 3.37  |
|         | (1.3,1.6,...,1.3,1.6) | Newton-J | 1.08 | 5.46 | 10.26 | 20.07 | 29.91 |
|         |              | Newton-GS | 0.83 | 4.47 | 8.84  | 16.90 | 25.26 |
|         |              | Newton-EG | 0.11 | 0.48 | 0.94  | 1.83  | 2.75  |
Table 2. Comparison of number of inner iteration, number of outer iteration and absolute error for Newton-J, Newton-GS and Newton-EG method with \( n = 1000, 5000, 10000, 20000 \) and 30000.

| No. of Value of | \( x_0 \) | \( n \) | Newton-J | Newton-GS | Newton-EG |
|----------------|---------|--------|----------|-----------|-----------|
|                | 1000    | 5000   | 10000   | 20000    | 30000    |
| Inner Iteration | (0.5, 0.5, ... , 0.5, 0.5) | 32 | 32 | 32 | 32 | 32 |
|                | (1.0, 1.0, ... , 1.0, 1.0) | 44 | 44 | 44 | 44 | 44 |
|                | (0.6, 1.1, ... , 0.6, 1.1) | 51 | 51 | 51 | 51 | 51 |
|                | (1.2, 1.2, ... , 1.2, 1.2) | 55715 | 55715 | 55715 | 55715 | 23879 |
|                | (1.5, 1.5, ... , 1.5, 1.5) | 77509 | 77509 | 77509 | 77509 | 35793 |
|                | (1.3, 1.6, ... , 1.3, 1.6) | 81578 | 81578 | 81578 | 81578 | 39702 |
| Outer Iteration | (0.5, 0.5, ... , 0.5, 0.5) | 5 | 5 | 5 | 5 | 5 |
|                | (1.0, 1.0, ... , 1.0, 1.0) | 8 | 8 | 8 | 8 | 8 |
|                | (0.6, 1.1, ... , 0.6, 1.1) | 9 | 9 | 9 | 9 | 9 |
|                | (1.2, 1.2, ... , 1.2, 1.2) | 32 | 32 | 32 | 32 | 32 |
|                | (1.5, 1.5, ... , 1.5, 1.5) | 77509 | 77509 | 77509 | 77509 | 35793 |
|                | (1.3, 1.6, ... , 1.3, 1.6) | 81578 | 81578 | 81578 | 81578 | 39702 |
| Absolute Error | (0.5, 0.5, ... , 0.5, 0.5) | 3.53e-14 | 3.53e-14 | 3.53e-14 | 3.53e-14 | 3.53e-14 |
|                | (1.0, 1.0, ... , 1.0, 1.0) | 3.53e-14 | 3.53e-14 | 3.53e-14 | 3.53e-14 | 3.53e-14 |
|                | (0.6, 1.1, ... , 0.6, 1.1) | 3.53e-14 | 3.53e-14 | 3.53e-14 | 3.53e-14 | 3.53e-14 |
|                | (1.2, 1.2, ... , 1.2, 1.2) | 3.53e-14 | 3.53e-14 | 3.53e-14 | 3.53e-14 | 3.53e-14 |
|                | (1.5, 1.5, ... , 1.5, 1.5) | 3.53e-14 | 3.53e-14 | 3.53e-14 | 3.53e-14 | 3.53e-14 |
|                | (1.3, 1.6, ... , 1.3, 1.6) | 3.53e-14 | 3.53e-14 | 3.53e-14 | 3.53e-14 | 3.53e-14 |
5. Conclusion
In this paper, a combined algorithm of the Newton method and Iterative methods has been presented for solving large scale unconstrained optimization problems. We have considered three different combinations with two of them were used as a reference. According to Table 1, we have shown that the execution time for our proposed combination algorithm (Newton-EG) is more efficient in comparison with the combination algorithm between Newton and other classical iterative method (Newton-Jacobi and Newton-GS). This result is consistent with what we can observe in Table 2, that our proposed method give a minimum number of inner iterations. In term of accuracy, almost all absolute errors for our proposed method (with the different is very small) are less than others. The efficiency of our proposed combination algorithm is obvious.

Acknowledgement
The authors are grateful for the fund received from Universiti Malaysia Sabah upon publication of this paper (GUG0160-2/2017).

References
[1] Shi Z-J and Shen J 2005 Step-size Estimation for Unconstrained Optimization Methods Comput. Appl. Math. 24(3), 399-416.
[2] Babaie-Kafaki S 2016 Computational Approaches in Large-Scale Unconstrained Optimization Studies in Big Data, vol 18. Springer DOI 10.1007/978-3-319-30265-2_17.
[3] Kaniel S and Dax A 1979 A modified Newton's method for unconstrained minimization SIAM J. Numerical Analysis 16(2) 324-31.
[4] Higham D J 1999 Trust Region Algorithms And Timestep Selection SIAM J. Numer. Anal. 37(1) 194-210.
[5] Andrei N 2016 An adaptive conjugate gradient algorithm for large-scale unconstrained optimization Journal of Computational and Applied Mathematics 292 83-91.
[6] Adesibigbe F M., Adebayo K J and Dele-Rotimi A. O 2015 On Quasi-Newton Method for Solving Unconstrained Optimization Problems America Journal of Applied Mathematics 3(2) 47-50.
[7] Liu D C and Nocedal J 1989 On The Limited Memory Bfgs Method For Large Scale Optimization Mathematical Programming 48 503-28.
[8] Powell M J D 1984 Numerical Analysis (Dundee, 1983), Lecture Notes in Mathematics, vol. 1066 122–41 (Springer: Berlin).
[9] Dennis Jr J E and Woods D J 1985 In: Wouk, A New computing environments: Microcomputers in large-scale scientific computing. United States.
[10] Moré J and Sorensen D 1982 Newton’s method, in: Studies in Numerical Analysis, Golub, G. (ed) (Washington DC: The Math. Association of America) 29-82.
[11] Gundersen G and Stiehaug T 2010 On large-scale unconstrained optimization problems an higher order methods Optimization Methods and Software 25(3), pp. 337-58.
[12] Sun W and Yuan Y 2006 Optimization Theory and Methods-Nonlinear Prog. Springer, United States.
[13] Nocedal J and Wright S J 2000 Numerical Optimization 2nd end. Springer-Verlag, Berlin.
[14] Roma M 2001 Large Scale Unconstrained Opti. Encyclopedia of Opti. Springer, United States.
[15] Sun W 1995 Generalized Newton method for LC1 unconstrained optimization, Journal of Computational Mathematical 15 (1995) 502-08.
[16] Pasano G 2011 Methods For Large-Scale Unconstrained Optimization. Wiley Encyclopedia of Operations Research and Management Science John Wiley & Sons.
[17] Sorensen D 1982 Newton's Method with a Model Trust Region Modification SIAM Journal on Numerical Analysis 19(2) 409-26.
[18] Goldstein A A and Price J F 1967 An Effective Algorithm for Min. Numerical Math. 10 184-89.
[19] Sisser F S 1982 A modified Newton's method for minimization. J of Opt. and Application 38(4) 461-82.
[20] Dasril Y, Mohd I and Mamat M 2004 Proc. Int. Conf. on Integrating Technology In The Mathematical Sciences(2004), USM, Malaysia 426-32.
[21] Gill P E and Murray W 1974 Newton-Type Methods For Unconstrained And Linearly Constrained Optimization Mathematical Programming 7 (1) 311-50.
[22] Dixon L C and Price R C 1989 Truncated Newton method for sparse unconstrained optimization using automatic differentiation *Journal of Optimization and Application* 60(2) 216-75.

[23] Kostopaulos A E, Androulakis G S and Grapsa T N 2009 A New Nonmonotone Newton’s Modification for Unconstrained Optimization. Technical Report, No. 09-05 (2009). Department of Mathematics, University of Patras.

[24] Dembo R S, Eisenstat S C and Steihaug T 1982 Inexact Newton Methods *SIAM Journal on Numerical Analysis* 19(2) 400-08.

[25] Sanjeev K, Vinary K, Sushil K T and Sukhjit S 2011 Geometrically constructed families of newton’s method for unconstrained optimization and nonlinear equations *International Journal of Mathematics and Mathematics Science* 1-9.

[26] Lin S Y, Lin C H and Yu S L 1993 An Efficient Descent Algorithm For A Class Of Unconstrained Optimization Problems Of Nonlinear Large Mesh-Interconnected Systems *Proceedings of the 32nd IEEE Conference on Decision and Control* 3584-89.

[27] Lin S Y and Lin C H 1995 An efficient method for unconstrained optimization problems of nonlinear large mesh-interconnected systems *IEEE Transaction on automatic control* 40(3) 490-95.

[28] Lin S Y and Lin C H 1996 A Modified Distributed Block Jacobi Method for Block Additive Unconstrained Optimization Problems of Large Systems *UKACC International Conference on Control ’96* 400-05.

[29] Evans D J 1985 Group explicit iterative methods *Int. J. Computer Maths.* 17 81-108.

[30] Yousif W S and Evans D J 1986 Explicit group over-relaxation methods for solving elliptic partial differential equations *Mathematics and Computer in Simulations* 28 453-66.

[31] Yousif W S and Evans D J 1995 Explicit de-coupled group iterative methods and their implementations, *Parallel Algorithms and Applications* 7 53-71.

[32] Abdullah A R 1991 The four point explicit decoupled group (EDG) method: A fast Poisson solver. *Int. J. Computer Maths.* 38 61-70.

[33] Othman M and Abdullah A R 2000 An efficient four points modified explicit group poisson solver *Intern. J. of Computer Maths.* 76 203-17.

[34] Sulaiman J, Hasan M K, Othman M and Karim S A A 2014 Fourth-order solutions of nonlinear two-point boundary value problems by Newton-HSSOR iteration *AIP Conference Proceedings* 1602, 69-75.

[35] Sulaiman J, Hasan M K, Othman M and Karim S A A 2015 Application Of Block Iterative Methods With Newton Scheme For Fisher’s Equation By Using Implicit Finite Difference *Jurnal Kalam.* 8(1) 039-46.

[36] Sulaiman J, Hasan M K, Othman M and Karim S A A 2012 Newton-EGMSOR Methods for Solution of Second Order Two-Point *Journal of Mathematics and System Science* 2 185-90.

[37] Sulaiman J, Hasan M K, Othman M and Karim S A A 2013 Numerical solutions of nonlinear second-order two-point boundary value problems using half-sweep SOR with Newton method. *Journal of Concrete & Applicable Mathematics.* 11(1), 112-20.

[38] Laguna M and Marti R 2005 Experimental Testing of Advanced Scatter Search Designs for Global Optimization of Multimodal Functions *Journal of Global Optimization* 33(2), 235-55.

[39] Witte B F and Holst W R 1964 Two New Direct Minimum Search *Proc. OF APRIL 21-23, 1964, Spring Joint Computer Conference*Afips, 64 (Spring) ACM, New York, NY, 195-209.

[40] Yang X 2010 *Engineering Optimization - An introduction with metaheuristic application.* John Wiley & Sons, Inc., New Jersey.