Newton-MSOR Method for Solving Large-Scale Unconstrained Optimization Problems with an Arrowhead Hessian Matrices

Khadizah Ghazali[1#], Jumat Sulaiman[1], Yosza Dasril[2], Darmesah Gabda[1]

1 Faculty of Science and Natural Resources, Mathematics with Economics Programme, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia.
2 Faculty of Electronic and Computer Engineering, Universiti Teknikal Malaysia Melaka, 76100 Melaka, Malaysia.
#Corresponding author. E-Mail: khadizah@ums.edu.my; Tel: +6088-320000-5747; Fax: +6088-435324.

ABSTRACT Due to a large-scale problem, solving unconstrained optimization using classical Newton's method is typically expensive to store its Hessian matrix and solve its Newton direction. Therefore, in this paper, we proposed a Newton-MSOR method for solving large scale unconstrained optimization problems whose Hessian matrix is an arrowhead matrix to overcome these problems. This Newton-MSOR method is a combination of the Newton method and modified successive-over relaxation (MSOR) iterative method. Some test functions are provided to show the validity and applicability of the proposed method. In order to calculate the performance of the proposed method, combinations between the Newton method with Gauss-Seidel point iterative method and the Newton method with successive-over relaxation (SOR) point iterative method were used as reference methods. Finally, the numerical results show that our proposed method provides results that are more efficient compared to the reference methods in terms of execution time and a number of iterations.

KEYWORDS: Newton method; MSOR iteration; Unconstrained optimization problems; Large-scale optimization; Iterative solution of linear systems; Arrowhead matrix.
I Received 2 May 2019 **II Revised** 14 August 2019 **II Accepted** 16 August 2019 **II Online** 28 August 2019 **II © Transactions on Science and Technology I**
Full Paper

INTRODUCTION

In this paper we only interested in large-scale problems ($n \geq 1000$), therefore we study a large-scale unconstrained optimization problem which specified as

$$\min_{\underline{\mathbf{x}} \in \Re^n} f(\underline{\mathbf{x}}) \tag{1}$$

where $f : \Re^n \to \Re$ is twice continuously differentiable. Generally, problem (1) can be solved using various type of method such as stated in Sun & Yuan (2006). Through the list of techniques discussed by them, we are more attracted to Newton's method. Since Newton's method possesses a fast quadratic rate of convergence, and it is also known as the best-known method on its outstanding performance when the starting point is choosing appropriately (Nocedal & Wright, 2000). Despite these reasons, notice that Newton's method has disadvantages when the problem is in the large-scale which is lead to difficulties in finding its second derivatives. Therefore many researchers have modified Newton's method to overcome the disadvantages such as proposed by Kaniel & Dax (1979), Shi (2000), Grapsa (2014) and Dehghan Niri *et al.* (2018).

Kaniel & Dax (1979) proposed a modified Newton's method for unconstrained minimization through the use of the symmetric decomposition as an alternative method for searching the Newton direction in solving the classical Newton method, while Shi (2000) combining the Newton direction with the steepest descent direction to achieve global and high local convergence order. Also, Grapsa (2014) proposed a new class of modified Newton's direction methods using a proper gradient's vector modification to have an efficient quadratic model with a new direction for solving problems of unconstrained optimization. Dehghan Niri *et al.* (2018) proposed a modified regularized Newton method for solving unconstrained optimization problems whose Hessian matrix may be singular without line search and analyze its convergence.

Thus, in this paper, we proposed an alternative method for finding large-scale unconstrained optimization problems with an arrowhead Hessian matrix by combining the Newton method with MSOR point iterative method, namely as Newton-MSOR method. This combination uses the MSOR iterative method for finding the Newton direction, while Newton's method is used to estimate the solution of problem (1). Kincaid and Young (1972) who are responsible for introducing the MSOR iterative method using two different relaxation factors to produce the fastest convergence which categorized as one of the numerical techniques that have an advantage of the efficient point iteration for solving any linear systems including large-scale system. To analyze the performance of our proposed method, we consider a combination of Newton method with SOR iteration and Newton method with Gauss-Seidel iteration as reference methods and they are called as Newton-SOR method and Newton-GS method respectively.

NEWTON SCHEME WITH AN ARROWHEAD HESSIAN MATRIX

In this paper, we start by using the quadratic Taylor approximation to $\underline{f}(\underline{\mathbf{x}})$ around the current point $\underline{\mathbf{x}}^{(k)}$, and then we minimize this approximation to have the next point $\underline{\mathbf{x}}^{(k+1)}$. Therefore we replace problem (1) as

$$\min_{\underline{\mathbf{x}}\in\Re^n} f\left(\underline{\mathbf{x}}^{(k)}\right)+\left[\nabla \underline{f}\left(\underline{\mathbf{x}}^{(k)}\right)\right]^T\left(\underline{\mathbf{x}}-\underline{\mathbf{x}}^{(k)}\right)+\frac{1}{2}\left(\underline{\mathbf{x}}-\underline{\mathbf{x}}^{(k)}\right)^T\nabla^2 \underline{f}\left(\underline{\mathbf{x}}^{(k)}\right)\left(\underline{\mathbf{x}}-\underline{\mathbf{x}}^{(k)}\right). \tag{2}$$

To solve problem (2), we set the gradient of this approximation to zero, so that we can have

$$\nabla \underline{f}\left(\underline{\mathbf{x}}^{(k)}\right)+\mathbf{H}\left(\underline{\mathbf{x}}^{(k)}\right)\left(\underline{\mathbf{x}}^{(k+1)}-\underline{\mathbf{x}}^{(k)}\right)=0, \tag{3}$$

where $\mathbf{H}\left(\underline{\mathbf{x}}^{(k)}\right)=\nabla^2 \underline{f}\left(\underline{\mathbf{x}}^{(k)}\right)$ as the Hessian matrix of second partial derivatives of $\underline{f}(\underline{\mathbf{x}})$. By simplifying equation (3), we can obtain the Newton iteration;

$$\underline{\mathbf{x}}^{(k+1)}=\underline{\mathbf{x}}^{(k)}-\left[\mathbf{H}\left(\underline{\mathbf{x}}^{(k)}\right)\right]^{-1}\nabla \underline{f}\left(\underline{\mathbf{x}}^{(k)}\right), \tag{4}$$

with its Newton direction;

$$\underline{d}^{(k)}=-\left[\mathbf{H}\left(\underline{\mathbf{x}}^{(k)}\right)\right]^{-1}\nabla \underline{f}\left(\underline{\mathbf{x}}^{(k)}\right). \tag{5}$$

As a particularly interesting case, we only considered Hessian of an arrowhead matrix of order n with the general form given by Stanimirovic *et al.* (2019);

$$\mathbf{H}\left(\underline{\mathbf{x}}^{(k)}\right)=\begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1\partial x_2} & \frac{\partial^2 f}{\partial x_1\partial x_3} & \cdots & \frac{\partial^2 f}{\partial x_1\partial x_{n-1}} \\ \frac{\partial^2 f}{\partial x_2\partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & 0 & \cdots & 0 \\ \frac{\partial^2 f}{\partial x_3\partial x_1} & 0 & \frac{\partial^2 f}{\partial x_3^2} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \frac{\partial^2 f}{\partial x_n\partial x_1} & 0 & \cdots & 0 & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

FORMULATION OF THE PROPOSED ITERATIVE METHOD

Since the Hessian of an arrowhead matrix, $\mathbf{H}\left(\underline{x}^{(k)}\right)$ is large and sparse, therefore finding the inverse of $\mathbf{H}\left(\underline{x}^{(k)}\right)$ by using direct method can cause a great computational cost that will lead to a solution involving very tedious work. As another solution, we used an iterative method as in Young (1971) and Youssef and Taha (2013) for solving a large linear system of equation (5). Let the linear system (5) **Error! Reference source not found.**be rewritten in general form as

$$\mathbf{A}\underline{d} = \underline{f} \tag{6}$$

where,

$$\mathbf{A} = \begin{bmatrix} b_1 & c_1 & c_2 & \cdots & c_{n-1} \\ a_2 & b_2 & 0 & \cdots & 0 \\ a_3 & 0 & b_3 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ a_n & 0 & \cdots & 0 & b_n \end{bmatrix}, \ \underline{d} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{bmatrix}, \ \underline{f} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_n \end{bmatrix}.$$

with $b_1, b_2 ..., b_n, a_2, ..., a_n, c_1, c_2 ..., c_{n-1} \in \Re$. To develop the formulation of our proposed iterative method, we decomposed the real coefficient matrix **A** of the linear system (6) as;

$$\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U} \tag{7}$$

where **D** is the nonzero diagonal part, **L** is strictly lower triangular part and **U** is strictly upper part, of **A**. By applying the decomposition in equation (7) into linear system (6) and considering the implementation of two different relaxation parameters, the iterative formulation of the MSOR method can be stated in vector form as (Kincaid & Young, 1972);

$$\begin{aligned} \underline{d}_{i+1}^{(k+1)} &= (\mathbf{D} - \omega_1 \mathbf{L})^{-1} (\omega_1 \mathbf{U} + (1-\omega_1)\mathbf{D}) \underline{d}_i^{(k)} + \omega_1 (\mathbf{D} - \omega_1 \mathbf{L})^{-1} \underline{f}_i, \ i = 1,3,...,n-1 \\ \underline{d}_{i+1}^{(k+1)} &= (\mathbf{D} - \omega_2 \mathbf{L})^{-1} (\omega_2 \mathbf{U} + (1-\omega_2)\mathbf{D}) \underline{d}_i^{(k)} + \omega_2 (\mathbf{D} - \omega_2 \mathbf{L})^{-1} \underline{f}_i, \ i = 2,4,...,n \end{aligned} \tag{8}$$

where ω_1 and ω_2 represent as a relaxation parameter with the optimal value in the range of $[1,2)$ and selected based on the smallest number of inner iterations. For the implementation of point iterations, each component $d_i^{(k+1)}$ can be computed as;

$$\begin{aligned} d_i^{(k+1)} &= (1-\omega_1) d_{i-1}^{(k)} + \frac{\omega_1}{b_i} \left(f_i - \sum_{1}^{n-1} c_j d_{i+1}^{(k)} \right), \ \text{for } i = 1 \\ d_i^{(k+1)} &= (1-\omega_1) d_{i-1}^{(k)} + \frac{\omega_1}{b_i} \left(f_i - a_i d_1^{(k)} \right), \ \text{for } i = 3,5,...,n-1 \\ d_i^{(k+1)} &= (1-\omega_2) d_{i-1}^{(k)} + \frac{\omega_2}{b_i} \left(f_i - a_i d_1^{(k)} \right), \ \text{for } i = 2,4,...,n \end{aligned} \tag{9}$$

By using the formulation of the MSOR iterative method to calculate the Newton direction (4) in Newton equation (5), we proposed the algorithm of Newton-MSOR scheme for solving problem (1). Note that for $\omega_1 = \omega_2 = 1$ equation (9) is reduced to the GS method and if $\omega_1 = \omega_2 \equiv \omega$, then equation (9) is reduced to the SOR method. Therefore, by using equation (6) and (9), we propose the reliable algorithm of Newton-MSOR scheme with an arrowhead Hessian matrix for solving problem (1) and stated it in Algorithm 1.

Algorithm 1: Newton-MSOR with an Arrowhead Hessian Matrix Scheme

i. Initialize

   Set up the objective function: $f(\underline{x})$, $\underline{x}^{(0)} \leftarrow \Re^n$, $\varepsilon_1 \leftarrow 10^{-6}$, $\varepsilon_2 \leftarrow 10^{-8}$ and $n \leftarrow \Re$

ii. Assign the optimal value of ω_1 and ω_2

iii. For $j = 1,2,...,n$ implement

   a. Set $\underline{d}^{(0)} \leftarrow 0$

   b. Calculate $f\left(\underline{x}^{(k)}\right)$

   c. Calculate the approximate value of $d_i^{(k+1)}$ by solve equation (6) using equation (9)

   d. Check the convergence test, $\left\| \underline{d}^{(k+1)} - \underline{d}^{(k)} \right\| \le \varepsilon_2$. If yes, go to step (e). Otherwise, go back to step (b)

   e. For $i = 1,2,...,n$ calculate; $\underline{x}^{(k+1)} \leftarrow \underline{x}^{(k)} + \underline{d}^{(k)}$

   f. Check the convergence test, $\left\| \nabla \underline{f}\left(\underline{x}^{(k)}\right) \right\| \le \varepsilon_1$. If yes, go to (iii). Otherwise, go back to step (a)

iv. Display approximate solutions

NUMERICAL EXPERIMENTS AND COMPUTATIONAL RESULTS

To check the performance of the proposed Algorithm, we have run the numerical experiments using three artificial problems obtained from the collection collected by Andrei (2004; 2008) where all of these problems have the same characteristic of Hessian that is an arrowhead Hessian matrix. The details for each of the three test functions are stated in Table 1. For each of these test functions, we use three different initial points. One of them is as suggested in Andrei (2004; 2008), and we labeled it as a standard initial point while the other two initial points are chosen randomly from a range surrounding the standard initial point or the optimal point. We specify it as a nonstandard initial point 1 and nonstandard initial point 2 in Table 1. Therefore in total, we have nine test cases which can be defined by test number. For example, we indicate the test case defined by test number 1 with a nonstandard initial point 2 as 1(b).

Table 1. Specifications of the test functions.

Test Number	Test Name, Algebraic Expression, local optimal value and optimal point	Initial point, $\mathbf{x}^{(0)}$	
		(a) Standard	(b) Nonstandard 1 (c) Nonstandard 2
1	LIARWHD $f(\underline{x})=\sum_{i=1}^{n}4\left(x_i^2-x_1\right)^2+\left(x_i-1\right)^2$ $f^*=0$ and $\mathbf{x}^*=(1.0,1.0,\ldots,1.0)$	$(4.0,4.0,\ldots,4.0)$	$(1.5,1.5,\ldots,1.5)$ $(3.3,3.5,\ldots,3.3,3.5)$
2	NONDIA $f(\underline{x})=(x_1-1)^2+\sum_{i=2}^{n}100\left(x_i-x_{i-1}^2\right)^2$ $f^*=0$ and $\mathbf{x}^*=(1.0,1.0,\ldots,1.0)$	$(-1.0,-1.0,\ldots,-1.0)$	$(2.0,2.0,\ldots,2.0)$ $(2.0,1.5,\ldots,2.0,1.5)$
3	DIAG-AUP1 $f(\underline{x})=\sum_{i=1}^{n}4\left(x_i^2-x_1\right)^2+\left(x_i^2-1\right)^2$ $f^*=0$ and $\mathbf{x}^*=(1.0,1.0,\ldots,1.0)$	$(4.0,4.0,\ldots,4.0)$	$(1.5,1.5,\ldots,1.5)$ $(3.3,3.5,\ldots,3.3,3.5)$

The computational result of the proposed and references method was compiled using C language with double precision arithmetic. For each test cases, we performed five numerical experiments with a different order of Hessian matrix as n = {1000, 5000, 10000, 20000, 30000}. We report the computational result of the Newton-GS (N_{GS}), Newton-SOR (N_{SOR}), and Newton-MSOR (N_{MSOR}) in Table 2. The detailed numerical results, including the number of inner iteration (NI_i), the number of outer iteration (NI_o), the execution time in seconds (t), function value at the iterate where execution terminated (FV_t) and maximum error (Max_ε). All values tabulated in Table 2 are rounded up to two decimal places. Therefore, all maximum error values are smaller than the convergence tolerance, ε_2. Finally, to have well understood for the efficiency of the comparison results in term of the execution time, we have computed the comparison of speedup ratio for Newton-MSOR method with both the reference methods in Table 3. In this table, we used the total execution time in seconds (Σ_t) for every test cases.

CONCLUSION

Base on all the results given in the previous section, we can conclude that in the process for solving large-scale unconstrained optimization problems with an arrowhead Hessian matrix, our proposed algorithm is efficient compared to the reference methods. This comparison has shown through the execution time and the number of iterations given in Table 2. For the accuracy, we can observe through all value indicates as the approximate value under column FV_t and it is shown that they are very approaching the optimal value except seven values obtained from the references method (in the test case 1(c)) and not from the proposed method. Therefore, the selection of an excellent initial point can influence the efficiency of the initial search process. As expected, with the use of the relaxation factor, the speedup ratio for Newton-MSOR and Newton-SOR were much faster than Newton-GS. From the speedup ratio in Table 3, we see that Newton-MSOR is up to 1.93 times faster than Newton-SOR and up to 358.18 times more rapid than Newton-GS. Thus, it can be concluded that our proposed iterative method (Newton-MSOR) can show substantial improvement in the number of iterations and execution time compared to the Newton-SOR and Newton-GS point iterative methods. In the future, we will investigate the efficiency of the combination of the Newton method with block iterative approach as in Ghazali *et al.* (2018, 2019) and Sulaiman *et al.* (2012).

Table 2. Computational result of the Newton-GS, Newton-SOR, and Newton-MSOR.

Test Cases	n	NI_i			NI_o			t			FV_t			Max_ε		
		N_{GS}	N_{SOR}	N_{2MSOR}	N_{GS}	N_{SOR}	N_{2MSOR}	N_{GS}	N_{SOR}	N_{2MSOR}	N_{GS}	N_{SOR}	N_{2MSOR}	N_{GS}	N_{SOR}	N_{2MSOR}
1(a)	1000	1567	376	366	38	14	17	0.04	0.01	0.00	2.40E-13	1.47E-14	4.61E-17	9.90E-07	3.16E-07	7.07E-07
	5000	2194	408	366	52	17	17	0.38	0.05	0.04	2.35E-13	6.00E-16	1.95E-16	9.72E-07	6.63E-07	5.87E-07
	10000	2509	443	405	58	17	17	0.63	0.09	0.08	2.23E-13	1.60E-15	2.44E-16	9.46E-07	4.64E-07	7.10E-08
	20000	2821	507	445	63	18	17	1.46	0.19	0.18	2.40E-13	1.00E-15	1.25E-15	9.80E-07	2.31E-07	2.49E-07
	30000	2899	549	507	65	21	17	1.95	0.33	0.31	2.45E-13	2.00E-16	6.31E-16	9.91E-07	2.89E-07	9.48E-07
1(b)	1000	1013	207	189	32	10	10	0.02	0.01	0.00	2.31E-13	2.00E-16	1.24E-16	9.70E-07	6.40E-07	2.54E-07
	5000	1039	232	214	46	8	8	0.10	0.03	0.02	2.21E-13	3.33E-14	2.87E-14	9.41E-07	3.66E-07	7.82E-07
	10000	1022	232	216	36	10	14	0.19	0.06	0.05	2.48E-13	2.51E-18	4.88E-16	9.97E-07	3.94E-07	4.64E-07
	20000	1038	232	213	52	8	8	0.40	0.09	0.08	2.48E-13	1.38E-17	3.66E-15	9.97E-07	6.88E-07	4.89E-07
	30000	1042	236	214	58	15	9	0.06	0.16	0.14	2.31E-13	2.78E-15	6.46E-15	9.61E-07	6.97E-07	5.18E-07
1(c)	1000	3169	761	349	43	34	18	0.05	0.01	0.01	3.73E+00	3.73E+00	1.53E-16	9.40E-07	7.42E-07	1.00E-02
	5000	2731	609	374	51	25	15	0.23	0.06	0.04	3.73E+00	3.73E+00	1.66E-15	9.58E-07	6.44E-07	4.00E-02
	10000	2394	463	412	57	21	25	0.40	0.10	0.09	2.24E-13	2.57E-18	9.03E-18	9.47E-07	4.61E-07	9.00E-02
	20000	2954	709	507	60	15	16	0.99	0.24	0.19	3.73E+00	3.73E+00	1.08E-15	9.68E-07	7.78E-07	9.42E-07
	30000	2703	669	530	65	11	22	1.39	0.34	0.31	3.73E+00	1.40E-15	3.67E-16	9.68E-07	4.09E-07	9.91E-07
2(a)	1000	2296	1079	1028	7528	124	112	1.34	0.03	0.02	2.47E-15	2.40E-18	5.94E-17	1.00E-06	9.72E-07	8.13E-07
	5000	52084	1883	1844	21374	225	55	18.95	0.26	0.16	2.47E-15	2.39E-15	1.76E-13	1.00E-06	9.55E-07	3.86E-07
	10000	128743	2633	2369	96881	320	179	123.63	0.66	0.53	2.47E-15	1.06E-15	5.29E-15	1.00E-06	6.73E-07	9.64E-07
	20000	243256	3308	3184	209658	319	266	518.18	1.58	1.47	2.47E-15	2.67E-15	1.47E-14	1.00E-06	3.12E-07	3.23E-07
	30000	351492	4129	3906	326806	669	444	1182.51	3.70	2.97	2.47E-15	7.32E-18	2.70E-15	9.99E-07	8.80E-07	9.47E-07
2(b)	1000	38789	2762	2711	7536	214	218	1.55	0.07	0.06	2.50E-15	1.12E-17	1.12E-17	9.99E-07	9.81E-07	9.69E-07
	5000	168711	7024	6378	45606	526	697	37.86	0.78	0.77	2.50E-15	1.99E-18	3.43E-18	1.00E-06	9.99E-07	9.91E-07
	10000	302950	10346	9963	98059	747	787	152.13	2.25	2.23	2.50E-15	1.02E-18	1.09E-18	1.00E-06	9.99E-07	9.97E-07
	20000	557276	13809	13774	209806	1444	1582	621.94	6.96	6.84	2.50E-15	5.73E-19	7.14E-19	1.00E-06	9.99E-07	9.94E-07
	30000	799141	17081	16872	326742	2134	1936	1194.97	13.28	12.67	2.50E-15	5.49E-19	4.67E-19	9.99E-07	9.97E-07	9.97E-07
2(c)	1000	38695	2810	2697	7535	110	203	1.18	0.06	0.06	2.50E-15	2.35E-18	1.01E-17	1.00E-06	9.07E-07	9.76E-07
	5000	173806	6905	6799	45608	652	779	33.15	0.84	0.85	2.50E-15	2.77E-18	3.77E-18	1.00E-06	9.95E-07	9.93E-07
	10000	302014	10175	10167	98058	1231	1203	129.24	2.64	2.61	2.50E-15	2.04E-18	1.97E-18	1.00E-06	9.93E-07	9.97E-07
	20000	554369	14569	14416	209808	1769	201	522.01	7.72	6.09	2.50E-15	9.80E-19	3.27E-14	1.00E-06	9.95E-07	4.07E-07
	30000	796433	17818	16942	326742	1446	1199	1193.60	11.90	10.79	2.50E-15	3.09E-19	4.65E-19	9.99E-07	9.94E-07	9.96E-07
3(a)	1000	412	185	167	17	15	14	0.03	0.01	0.00	4.69E-14	4.01E-17	5.27E-17	8.70E-07	6.74E-07	4.94E-07
	5000	452	184	168	20	17	14	0.05	0.03	0.02	4.95E-14	4.03E-18	1.40E-17	8.90E-07	4.85E-07	7.00E-07
	10000	463	187	168	21	17	14	0.10	0.08	0.04	5.79E-14	1.94E-18	4.98E-18	9.63E-07	4.84E-07	6.87E-07
	20000	470	188	168	23	17	14	0.20	0.14	0.08	5.32E-14	1.73E-18	1.63E-18	9.23E-07	6.50E-07	1.17E-07
	30000	473	188	169	24	18	15	0.27	0.23	0.13	4.54E-14	4.58E-19	1.68E-19	8.53E-07	4.09E-07	1.94E-07
3(b)	1000	279	105	92	13	10	9	0.01	0.00	0.00	4.66E-14	1.81E-17	1.75E-16	8.67E-07	4.55E-07	8.64E-07
	5000	284	106	92	17	10	10	0.07	0.04	0.02	5.70E-14	6.55E-19	5.74E-18	9.56E-07	6.81E-08	5.70E-07
	10000	286	108	93	19	10	11	0.07	0.06	0.04	4.63E-14	1.23E-18	7.71E-18	8.61E-07	1.85E-07	7.98E-08
	20000	287	107	93	20	10	11	0.14	0.11	0.05	5.90E-14	2.40E-18	1.46E-17	9.72E-07	4.20E-07	1.28E-07
	30000	289	107	93	21	10	11	0.19	0.15	0.08	5.66E-14	3.58E-18	2.15E-17	9.52E-07	6.54E-07	1.76E-07
3(c)	1000	404	176	162	16	12	12	0.01	0.01	0.00	5.94E-14	4.51E-17	4.18E-17	9.79E-07	7.57E-07	5.00E-07
	5000	434	170	158	20	14	14	0.05	0.03	0.02	4.65E-14	5.95E-18	4.66E-18	8.64E-07	6.11E-07	3.74E-07
	10000	441	172	156	22	15	14	0.09	0.08	0.04	4.09E-14	9.36E-19	8.32E-18	8.09E-07	3.46E-07	6.82E-07
	20000	444	178	157	23	16	16	0.18	0.16	0.08	4.11E-14	9.74E-19	6.38E-18	8.11E-07	4.74E-07	7.53E-07
	30000	446	176	157	24	16	16	0.28	0.24	0.13	5.64E-14	3.83E-19	3.30E-18	9.50E-07	3.80E-07	7.27E-07

Table 3. Comparison of speedup ratio for Newton-MSOR method with Newton-GS method and Newton-SOR method.

Test Cases	Σ_t			Speedup ratio		
	N_{GS} (I)	N_{SOR} (II)	N_{MSOR} (III)	$\frac{I}{II}$	$\frac{I}{III}$	$\frac{II}{III}$
1(a)	4.46	0.67	0.61	6.66	7.31	1.10
1(b)	0.77	0.35	0.29	2.20	2.66	1.21
1(c)	3.06	0.75	0.64	4.08	4.78	1.17
2(a)	1844.61	6.23	5.15	296.09	**358.18**	1.21
2(b)	2008.45	23.34	22.57	86.05	88.99	1.03
2(c)	1879.18	23.16	20.4	81.14	92.12	1.14
3(a)	0.65	0.49	0.27	1.33	2.41	1.81
3(b)	0.48	0.36	0.19	1.33	2.53	1.89
3(c)	0.61	0.52	0.27	1.17	2.26	**1.93**

ACKNOWLEDGMENTS

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

REFERENCES

[1] Andrei, N. (2004). Test functions for unconstrained optimization. *Research Institute for informatics. Center for Advanced Modeling and Optimization*, 8-10, Averescu Avenue, Sector 1, Bucharest, Romania, 1-15.

[2] Andrei, N. (2008). An unconstrained optimization test function collection. *Advanced Modeling and Optimization. An Electronic International Journal,* **10**(1), 147-161.

[3] Dehghan Niri, T., Hosseini, M. M. & Heydari, M. (2018). A new modified trust region algorithm for solving unconstrained optimization problems. *Journal of Mathematics Extension,* **12**(4), 115-135.

[4] Grapsa, T. N. (2014). A modified Newton direction for unconstrained optimization. *Optimization*, **63**(7), 983-1004.

[5] Ghazali, K., Sulaiman, J., Dasril, Y. & Gabda, D. (2018). Newton method with explicit group iteration for solving large scale unconstrained optimization problems. *Journal of Physics Conference Series,* **1132**(2018), 012056.

[6] Ghazali, K., Sulaiman, J., Dasril, Y. & Gabda, D. (2019). Application of Newton-4EGSOR Iteration for Solving Large Scale Unconstrained Optimization Problems with a Tridiagonal Hessian Matrix. *In*: Alfred, R., Lim, Y., Ibrahim, A. & Anthony, P. (eds.) *Computational Science and Technology. Lecture Notes in Electrical Engineering 481*. Singapore: Springer.

[7] Kincaid, D. R. & Young, D. M. (1972). The modified successive overrelaxation method with fixed parameters. *Mathematics of Computation*, **119**(26), 705-717.

[8] Kaniel, S. & Dax, A. (1979). A modified Newton's method for unconstrained minimization. *Society for Industrial and Applied Mathematics,* **16**(2), 324-331.

[9] Nocedal, J. & Wright, S. J. (2000). *Numerical Optimization 2nd end*. Berlin: Springer-Verlag.

[10] Shi, Y. (2000). Globally Convergent Algorithms for Unconstrained Optimization. *Computational Optimization and Application,* **16,** 295-308.

[11] Stanimirovic, P. S., Katsikis, V. N. & Kolundzija, D. (2019). Inversion and pseudoinversion of block arrowhead matrices. *Applied Mathematics and Computation,* **341,** 379-401.

[12] Sulaiman, J., Hasan, M. K. Othman, M. & Abdul Karim, S. A. (2012). Newton-EGMSOR Methods for Solution of Second Order Two-Point Nonlinear Boundary Value Problems. *Journal of Mathematics and System Science,* **2**(3), 185-190.

[13] Sun, W. & Yuan, Y. (2006). *Optimization Theory and Methods-Nonlinear Programming*. United States: Springer.

[14] Young, D. M. (1971). *Iterative solution of large linear systems*. London: Academic Press.

[15] Youssef, I. K. & Taha, A. A. (2013). On the modified successive overrelaxation method. *Applied Mathematics and Computational,* **219**(2013), 4601-4613.