Mathematics and Statistics 8(2A): 40-46, 2020
DOI: 10.13189/ms.2020.081307 http://www.hrpub.org

An Alternative Approach for Finding Newton's Direction in Solving Large-Scale Unconstrained Optimization for Problems with an Arrowhead Hessian Matrix

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

[1]Faculty of Science and Natural Resources, Universiti Malaysia Sabah, Malaysia
[2]Faculty of Electronic and Computer Engineering, Universiti Teknikal Malaysia Melaka, Malaysia

Received August 10, 2019; Revised February 1, 2020; Accepted February 20, 2020

Abstract In this paper, we proposed an alternative way to find the Newton direction in solving large-scale unconstrained optimization problems where the Hessian of the Newton direction is an arrowhead matrix. The alternative approach is a two-point Explicit Group Gauss-Seidel (2EGGS) block iterative method. To check the validity of our proposed Newton's direction, we combined the Newton method with 2EGGS iteration for solving unconstrained optimization problems and compared it with a combination of the Newton method with Gauss-Seidel (GS) point iteration and the Newton method with Jacobi point iteration. The numerical experiments are carried out using three different artificial test problems with its Hessian in the form of an arrowhead matrix. In conclusion, the numerical results showed that our proposed method is more superior than the reference method in term of the number of inner iterations and the execution time.

Keywords Newton Method, Explicit Group Iteration, Unconstrained Optimization Problems, Large-Scale Optimization, Arrowhead Matrix

1. Introduction

We consider large-scale unconstrained optimization problem:

$$\min_{\underline{\mathbf{x}}\in\mathbb{R}^n} f\left(\underline{\mathbf{x}}\right) \tag{1}$$

where $f:\mathbb{R}^n\to\mathbb{R}$ is twice continuously differentiable and $n\geq 1000$. Theoretically, classical Newton's method is one of the most powerful gradient descent techniques that possess the fast quadratic rate of convergence for solving problem (1). In practice, however, Newton's method may be not very favorable if it involves large-scale problems. This matter happens because Newton's method uses direct methods to find Newton's direction that contains calculating not only first derivatives but also second derivatives [1].

Therefore, in large-scale problems, many researchers avoid this method by using other gradient descent methods such as in [2-5] or by merely modifying this classical Newton's method [6-8]. As proposed by Esmaeili and Kimiaei in [2], they developed an improved adaptive trust-region method for solving the unconstrained optimization problem and tested it on 93 standard un-constrained optimization test problems with the dimension up to 20000. Also, a new generalized nonmonotone line search for solving large-scale unconstrained optimization is proposed by Shuai *et al.* [3] to integrate advantages of the existing line searches. Other than that Esmaeili *et al.* [4] presented a new spectral conjugate gradient method based on the Dai-Yuan strategy for solving large-scale unconstrained optimization, while Moyi and Leong [5] proposed a three-term conjugate gradient method via the symmetric rank-one update.

In addition to the methods described in [2-5], there are also researchers who still retain the use of the Newton method but without finding the second derivatives of the function problem. Dembo and Steihaug [6] proposed a combination of the Newton method with the conjugate gradient method for solving problem (1) and named it as truncated-Newton algorithms, while Grapsa [7] modified Newton direction thru a proper gradient vector modification and then introduced Componentwise Approximation Gradient method. Recently, Boggs and

Byrd [8] studied the L-BFGS method, then suggested two computationally efficient ways to measure the effectiveness of various memory sizes and showed that their approach improves the performance of the L-BFGS method for solving problem (1).

Just as what the researchers have discussed in [6-8], we have also modified this Newton direction, but we use a different approach compared to them. We noticed that the Newton direction is obtained from solving the large linear system. Thus we used a family of Explicit Group (EG) block iterative method to find the Newton direction. This EG block iterative method was introduced by Evans [9] by grouping the linear system to several points for generating a new system of linear equations and it is famous as one of the efficient block iterative methods which later have been further established by Yousif and Evans [10,11], Othman and Abdullah [13] and Sulaiman *et al.* [12,13].

Thus, in this paper, inspired by the advantages of the Newton method, we proposed 2-point explicit group Gauss-Seidel (2EGGS) block iterative method (as inner iteration) to find the Newton direction and then solve problem (1) using Newton's method (as outer iteration). We named this method as the Newton-2EGGS method. To evaluate the performance of the Newton-2EGGS method, we consider a combination of the Newton method with Gauss-Seidel iteration and a combination of the Newton method with Jacobi iteration as reference method and they are called as the Newton-GS method and the Newton-Jacobi method respectively.

This paper is organized as follows. In the next section, we described a Newton scheme with its Hessian type, while in Section 3, we formulated the 2EGGS iteration for computing Newton's direction. Numerical experiment and result are analyzed in Section 4. Finally, the conclusion is stated in Section 5.

2. Approaches to Newton Scheme with an Arrowhead Hessian Matrix

In this section, we obtained the Newton iteration base on a quadratic model of the $f(\underline{x})$, by using the first three terms of the Taylor series expansion about $\underline{x}^{(k)}$;

$$f(\underline{x}) \approx f\left(\underline{\mathbf{x}}^{(k)}\right) + \left[\nabla \underline{f}\left(\underline{\mathbf{x}}^{(k)}\right)\right]^T \Delta\underline{\mathbf{x}} + \frac{1}{2}\left[\Delta\underline{\mathbf{x}}\right]^T \mathrm{H}\left(\underline{\mathbf{x}}^{(k)}\right)\Delta\underline{\mathbf{x}} \qquad (2)$$

where $\Delta\underline{\mathbf{x}} = \underline{\mathbf{x}} - \underline{\mathbf{x}}^{(k)}$, $\nabla \underline{f}\left(\underline{\mathbf{x}}^{(k)}\right)$ is the gradient of $f(\underline{x})$ and $\mathrm{H}\left(\underline{\mathbf{x}}^{(k)}\right) = \nabla^2 f\left(\underline{\mathbf{x}}^{(k)}\right)$ is the Hessian of $f(\underline{x})$. Next, we set the partial derivatives of equation (2) to zero (as followed the optimality conditions [1]) to obtained the minimum value of $f(\underline{x})$;

$$\nabla f\left(\underline{\mathbf{x}}^{(k)}\right) = \mathbf{0}. \qquad (3)$$

From equations (2) and (3);

$$\nabla f\left(\underline{\mathbf{x}}^{(k)}\right) + \mathrm{H}\left(\underline{\mathbf{x}}^{(k)}\right)\Delta\underline{\mathbf{x}} = \mathbf{0}, \qquad (4)$$

so thus, we obtained the new iteration scheme as;

$$\underline{\mathrm{x}}^{(k+1)} = \underline{\mathrm{x}}^{(k)} - \left[\mathrm{H}\left(\underline{\mathbf{x}}^{(k)}\right)\right]^{-1} \nabla f\left(\underline{\mathbf{x}}^{(k)}\right) \qquad (5)$$

where $\left[\mathrm{H}\left(\underline{\mathbf{x}}^{(k)}\right)\right]^{-1}$ is the inverse of the Hessian. The iteration scheme defined by equation (5) is known as Newton's iteration. Noted that from equation (5), we properly defined the Newton direction as;

$$\underline{d}^{(k)} = -\left[\mathrm{H}\left(\underline{\mathbf{x}}^{(k)}\right)\right]^{-1} \nabla f\left(\underline{\mathbf{x}}^{(k)}\right). \qquad (6)$$

It seems that each iteration of this classical Newton's method requires computational of the inverse of the Hessian. In order to prevent this procedure from occurring (since our problem is a large-scale problem), we rewrite equation (6) as the Newton equation;

$$\mathrm{H}\left(\underline{\mathbf{x}}^{(k)}\right)\underline{d}^{(k)} = -\nabla f\left(\underline{\mathbf{x}}^{(k)}\right). \qquad (7)$$

We now focus our attention on seeking the Newton direction through equation (7) for the large-scale problem with *n* by *n* real arrowhead matrix. Therefore, we only considered problems with Hessian of an arrowhead matrix of order *n* with the general form given by [15];

$$\mathrm{H}\left(\underline{\mathbf{x}}^{(k)}\right) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_1 \partial x_3} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & 0 & \cdots & 0 \\ \frac{\partial^2 f}{\partial x_3 \partial x_1} & 0 & \frac{\partial^2 f}{\partial x_3^2} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & 0 & \cdots & 0 & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \qquad (8)$$

This Hessian is a square matrix of second-order partial derivatives of order *n* by *n*.

3. The Proposed Approach for Finding the Newton Direction

As noted in the previous section, we start searching the Newton direction from equation (7) and this equation is nothing but a linear system. For simplicity, we rewrite equation (7) as in the general form;

$$\mathbf{A}\underline{d} = \underline{f} \qquad (9)$$

where,

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

To formulate the Jacobi iteration, we decomposed matrix

$\mathbf{A}$ in equation (9) as a summation of three matrices as follows [16,17];

$$\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U} \tag{10}$$

where $\mathbf{D}$ is the nonzero diagonal matrix of $\mathbf{A}$, $\mathbf{L}$ and $\mathbf{U}$ are strictly lower and upper triangular matrices of $\mathbf{A}$, respectively. By applying the decomposition in equation (10) into a linear system (9), the iterative formulation of the Jacobi method stated in vector form as [16];

$$\underline{d}^{(k+1)} = \mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})\underline{d}^{(k)} + \mathbf{D}^{-1}\underline{f} \tag{11}$$

In the same way, the Gauss-Seidel iteration also can be stated in vector form as [16,18];

$$\underline{d}^{(k+1)} = (\mathbf{D}-\mathbf{L})^{-1}\mathbf{U}\underline{d}^{(k)} + (\mathbf{D}-\mathbf{L})^{-1}\underline{f} \tag{12}$$

The following subsections will discuss the formation of 2-point explicit group Gauss-Seidel iterative methods.

3.1. Formation of 2EGGS Iteration

The 2EGGS iteration involves a complete group of two points accordingly for generating 2 by 2 systems of a linear equation. Thus, the formulation of the proposed iterative method can be expressed as a group of two points from the linear system (9) as [9,12,13];

$$\begin{aligned} &\begin{bmatrix} b_i & c_i \\ a_{i+1} & b_{i+1} \end{bmatrix}\begin{bmatrix} d_i \\ d_{i+1} \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}, i = 1 \\ &\begin{bmatrix} b_i & 0 \\ 0 & b_{i+1} \end{bmatrix}\begin{bmatrix} d_i \\ d_{i+1} \end{bmatrix} = \begin{bmatrix} s_3 \\ s_4 \end{bmatrix}, i = 3,5,...,n-1 \end{aligned} \tag{13}$$

where,

$$s_1 = f_i - \sum_{j=2}^{n-1} c_j d_{j+1},\ s_2 = f_{i+1},\ s_3 = f_i - a_i d_1,\ s_4 = f_{i+1} - a_{i+1}d_1$$

Hence, solving and simplifying equation (13) gives the explicit form of the 2EGGS iteration as;

$$\begin{aligned} i = 1; \quad & d_i^{(k+1)} = \frac{1}{\mu}\left[b_{i+1}s_1 - c_i s_2\right], \\ & d_{i+1}^{(k+1)} = \frac{1}{\mu}\left[-a_{i+1}s_1 + b_i s_2\right]. \\ i = 3,5,...,n-1; \quad & d_i^{(k+1)} = \frac{1}{\mu'}\left[b_{i+1}s_3\right], \\ & d_{i+1}^{(k+1)} = \frac{1}{\mu'}\left[b_i s_4\right]. \end{aligned} \tag{14}$$

where,

$$\mu = b_{i+1}b_i - a_{i+1}c_i \quad \text{and} \quad \mu' = b_{i+1}b_i .$$

This equation (14) is the alternative approach for finding Newton direction. Thus, we proposed the following algorithm for solving problem (1).

Algorithm: Newton-2EGGS with an Arrowhead Hessian Matrix Scheme

i. Initialize – Set up the objective function $f(\underline{\mathbf{x}})$, $\underline{x}^{(0)} \leftarrow \mathbb{R}^n$, $\varepsilon_1 \leftarrow 10^{-6}$, $\varepsilon_2 \leftarrow 10^{-8}$ and $n \leftarrow \mathbb{R}$

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

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

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

c. Calculate the approximate value of $d_i^{(k+1)}$ using equation (14).

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 the estimation value; $\underline{x}^{(k+1)} \leftarrow \underline{x}^{(k)} + \underline{d}^{(k)}$

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)

iii. Display approximate solutions

4. Numerical Experiments and Computational Results Conclusion

Table 1. Description of the Test Problem

Test Number	Name, Algebraic Expression, local optimal value, optimal point and initial point (a-standard; b-nonstandard1; c-nonstandard2)
1	LIARWHD $f(\underline{\mathbf{x}}) = \sum_{i=1}^{n} 4\left(-x_1 + x_i^2\right)^2 + \sum_{i=1}^{n}\left(x_i - 1\right)^2$ $f^* = 0$, $\mathbf{x}^* = (1.0,1.0,...,1.0)$ and (a) $(4.0,4.0,...,4.0)$ (b) $(1.5,1.5,...,1.5)$ (c) $(3.3,3.5,...,3.3,3.5)$
2	NONDIA $f(\underline{\mathbf{x}}) = (x_1 - 1)^2 + \sum_{i=2}^{n} 100\left(x_1 - x_{i-1}^2\right)^2$ $f^* = 0$, $\mathbf{x}^* = (1.0,1.0,...,1.0)$ and (a) $(-1.0,-1.0,...,-1.0)$ (b) $(2.0,2.0,...,2.0)$ (c) $(2.0,1.5,...,2.0,1.5)$
3	DIAG-AUP1 $f(\underline{\mathbf{x}}) = \sum_{i=1}^{n} 4\left(x_i^2 - x_1\right)^2 + \sum_{i=1}^{n}\left(x_i^2 - 1\right)^2$ $f^* = 0$, $\mathbf{x}^* = (1.0,1.0,...,1.0)$ and (a) $(4.0,4.0,...,4.0)$ (b) $(1.5,1.5,...,1.5)$ (c) $(3.3,3.5,...,3.3,3.5)$

In this section, we applied three different artificial unconstrained optimization test problems selected from the collection collected by Andrei in [19,20] to see the behavior and evaluate the performance of our proposed algorithm. The selection of all these test problems are based on its Hessian type, which is an arrowhead Hessian matrix, and the description of all these test problems are stated in Table 1. For each test problem, we used three initial points, and one of these initial points (denoted as (a)) is from the standard point suggested by Andrei in [19,20]. While the other two nonstandard initial points (denoted as (b) and (c)) are chosen randomly from a range surrounding the standard initial point. Moreover, all these test problems were run five times using five different order of Hessian matrix as

$$n = \{1000, 5000, 10000, 20000, 30000\}.$$

Thus, this gives the total number of test problems are 75 problems.

The proposed algorithm was coded in C language with double precision arithmetic. From the algorithm, we can see that this algorithm involves two loops with two difference convergence tolerance. The first loop is for finding the Newton direction using our proposed iterative method as inner iteration with $\varepsilon_2 = 10^{-8}$ as the convergence tolerance and the second loop is to estimate the solution for problem (1) using the Newton iteration (5) as an outer iteration with $\varepsilon_1 = 10^{-6}$ as the convergence tolerance.

In order to see the behavior of our proposed algorithm, five parameters are observed, and there are the number of inner iterations, number of outer iterations, execution time (measured in seconds), function value at the iterate where execution terminated, and maximum absolute error. Computational results corresponding to the algorithm are compared with the reference methods and tabulated in Table 2. We used the symbol N_J, N_{GS}, and N_{2EGGS} to represent the N ewton-Jacobi method, the Newton-GS method, and the Newton-2EGGS method, respectively.

All values shown in Table 2 are rounded up to two decimal places. Therefore, all maximum absolute error values are smaller than the convergence tolerance. We noted that our proposed method achieved about the same final objective value as both our reference method (see columns under function value of Table 2) and all of these value are closer to the optimal value.

It is noticeable that the number of inner iteration and the execution time (in seconds) for our proposed method indicate that there is a reduction (or at least the same) compared to the referenced method. Therefore, to have well understood for the efficiency of these reductions and as well as to evaluate the performance of our proposed algorithm, we presented Table 3 and Table 4. Table 3 specified the decrement percentage of the number of inner iteration for the Newton-2EGGS method and the Newton-GS method compared to the Newton-Jacobi method, while Table 4 stated the comparison of speed ratio for Newton-2EGGS method with both the reference methods. Furthermore, in Table 4, we used the total execution time in seconds for every test problems.

Table 2. Computational Result of the Newton-Jacobi method, the Newton-GS method, and the Newton-2EGGS method

Test Problem (initial point)	n	Number of inner iteration			Number of outer iteration			Execution time (seconds)			Function value at the iterate where execution terminated			Maximum absolute error		
		N_J	N_{GS}	N_{2EGGS}	N_J	N_{GS}	N_{2EGGS}	N_J	N_{GS}	N_{2EGGS}	N_J	N_{GS}	N_{2EGGS}	N_J	N_{GS}	N_{2EGGS}
1(a)	1000	3343	1567	1566	129	38	38	0.07	0.04	0.03	2.60E-15	2.40E-13	2.37E-13	9.58E-07	9.90E-07	9.84E-07
	5000	4771	2194	2194	203	52	52	0.44	0.38	0.33	4.51E-17	2.35E-13	2.35E-13	9.67E-07	9.72E-07	9.71E-07
	10000	5479	2509	2509	227	58	58	0.99	0.63	0.62	2.23E-17	2.23E-13	2.23E-13	9.64E-07	9.46E-07	9.46E-07
	20000	6149	2821	2821	240	63	63	2.23	1.46	1.30	1.00E-16	2.40E-13	2.40E-13	9.98E-07	9.80E-07	9.80E-07
	30000	6456	2899	2899	264	65	65	3.58	2.07	1.96	2.35E-18	2.45E-13	2.45E-13	9.82E-07	9.91E-07	9.91E-07
1(b)	1000	2289	1013	1012	130	32	32	0.06	0.03	0.02	2.10E-15	2.31E-13	2.27E-13	9.59E-07	9.70E-07	9.64E-07
	5000	2274	1039	1039	178	46	46	0.25	0.21	0.17	8.00E-16	2.21E-13	2.21E-13	9.75E-07	9.41E-07	9.41E-07
	10000	2281	1022	1022	219	36	36	0.53	0.35	0.30	3.66E-17	2.48E-13	2.47E-13	9.48E-07	9.97E-07	9.96E-07
	20000	2287	1038	1038	240	52	52	1.11	0.65	0.63	5.94E-18	2.48E-13	2.48E-13	9.48E-07	9.97E-07	9.96E-07
	30000	2303	1042	1042	260	58	58	1.68	0.97	0.88	2.53E-18	2.31E-13	2.31E-13	9.85E-07	9.61E-07	9.61E-07
1(c)	1000	3378	1556	1556	148	37	37	0.07	0.04	0.03	2.00E-16	2.42E-13	2.42E-13	9.62E-07	9.93E-07	9.93E-07
	5000	4670	2131	2131	202	50	50	0.43	0.27	0.26	4.14E-17	2.47E-13	2.47E-13	9.98E-07	9.95E-07	9.95E-07
	10000	5242	2384	2384	227	57	57	1.07	0.56	0.55	1.27E-17	2.25E-13	2.25E-13	9.73E-07	9.50E-07	9.50E-07
	20000	5487	2553	2553	214	61	61	2.02	1.19	1.18	6.00E-16	2.49E-13	2.49E-13	9.78E-07	9.99E-07	9.99E-07
	30000	5513	2593	2593	254	33	33	3.12	1.64	1.63	3.61E-17	2.39E-13	2.39E-13	9.90E-07	9.78E-07	9.78E-07
2(a)	1000	56822	22986	20391	14805	7528	4790	2.09	1.34	1.11	1.00E-16	2.47E-15	6.18E-13	1.00E-06	1.00E-06	1.00E-06
	5000	196378	52084	41128	91368	28374	16286	53.87	18.95	13.79	8.24E-19	2.47E-15	3.09E-12	1.00E-06	1.00E-06	1.00E-06
	10000	359514	128743	92343	198451	96881	58220	227.37	123.63	105.12	9.28E-19	2.47E-15	6.19E-12	1.00E-06	1.00E-06	1.00E-06
	20000	593232	243256	163452	416445	209648	125329	880.42	518.18	318.13	1.00E-16	2.47E-15	1.24E-11	1.00E-06	1.00E-06	1.00E-06
	30000	708901	351492	225732	645745	326806	194282	2011.75	1182.51	730.05	1.23E-14	2.47E-15	1.86E-11	1.00E-06	9.99E-07	1.00E-06
2(b)	1000	101544	38789	35915	14976	7536	4799	2.71	1.55	1.03	1.07E-17	2.50E-15	6.17E-13	9.99E-07	9.99E-07	9.99E-07
	5000	461271	168711	150420	89585	45606	27952	72.64	37.86	23.21	4.70E-17	2.50E-15	3.09E-12	1.00E-06	1.00E-06	1.00E-06
	10000	763029	302950	262245	172737	98059	59324	256.17	152.13	90.57	4.00E-16	2.50E-15	6.19E-12	9.99E-07	1.00E-06	1.00E-06
	20000	1369207	557276	468968	357916	209806	125492	1004.41	621.94	354.63	2.19E-17	2.50E-15	1.24E-11	1.00E-06	1.00E-06	1.00E-06
	30000	1899369	799141	660672	517316	326742	194237	2181.71	1194.97	802.18	2.61E-17	2.50E-15	1.86E-11	9.99E-07	9.99E-07	1.00E-06
2(c)	1000	101288	38695	35821	14985	7535	4798	3.68	1.18	0.96	1.24E-17	2.50E-15	6.17E-13	1.00E-06	1.00E-06	9.99E-07
	5000	495794	173806	155654	89583	45608	27953	89.09	33.15	24.14	1.17E-18	2.50E-15	3.09E-12	1.00E-06	1.00E-06	1.00E-06
	10000	792833	302014	261383	192949	98058	59325	333.82	129.24	90.22	1.00E-16	2.50E-15	6.19E-12	9.99E-07	1.00E-06	1.00E-06
	20000	1328778	554369	465902	351904	209808	125492	1186.07	522.01	355.52	1.40E-15	2.50E-15	1.24E-11	1.00E-06	1.00E-06	1.00E-06
	30000	1938690	796433	659580	587757	326742	194237	2465.87	1193.60	805.08	1.00E-16	2.50E-15	1.86E-11	9.99E-07	9.99E-07	1.00E-06
3(a)	1000	827	412	408	44	17	17	0.06	0.03	0.01	1.30E-15	4.69E-14	4.55E-14	9.09E-07	8.70E-07	8.57E-07
	5000	927	452	450	56	20	20	0.22	0.05	0.05	2.00E-16	4.95E-14	5.23E-14	9.61E-07	8.90E-07	9.16E-07
	10000	950	463	462	64	21	22	0.30	0.11	0.10	1.00E-16	5.79E-14	4.71E-14	8.96E-07	9.63E-07	8.68E-07
	20000	971	470	470	71	23	23	0.56	0.20	0.19	1.79E-17	5.32E-14	4.84E-14	8.48E-07	9.23E-07	8.80E-07
	30000	977	473	473	74	24	24	0.87	0.27	0.27	1.58E-17	4.54E-14	4.26E-14	9.71E-07	8.53E-07	8.25E-07
3(b)	1000	595	279	279	42	13	14	0.02	0.01	0.01	4.00E-16	4.66E-14	4.30E-14	8.21E-07	8.67E-07	8.33E-07
	5000	615	284	284	56	17	17	0.08	0.07	0.03	1.00E-16	5.70E-14	5.61E-14	8.35E-07	9.56E-07	9.48E-07
	10000	623	286	286	62	19	19	0.15	0.07	0.06	3.91E-17	4.63E-14	4.59E-14	8.53E-07	8.61E-07	8.57E-07
	20000	629	287	287	68	20	20	0.33	0.14	0.13	2.05E-17	5.90E-14	5.88E-14	8.72E-07	9.72E-07	9.70E-07
	30000	633	289	289	72	21	21	0.47	0.19	0.19	1.26E-17	5.66E-14	5.64E-14	8.36E-07	9.52E-07	9.50E-07
3(c)	1000	820	404	401	45	16	16	0.02	0.01	0.01	3.00E-16	5.94E-14	5.00E-14	8.32E-07	9.79E-07	8.98E-07
	5000	892	434	432	58	20	20	0.10	0.05	0.04	1.00E-16	4.65E-14	5.66E-14	9.83E-07	8.64E-07	9.52E-07
	10000	913	441	439	66	22	21	0.21	0.09	0.09	4.77E-17	4.09E-14	5.60E-14	8.10E-07	8.09E-07	9.47E-07
	20000	922	444	443	71	23	23	0.40	0.18	0.17	2.11E-17	4.11E-14	5.99E-14	8.79E-07	8.11E-07	9.79E-07
	30000	928	446	445	75	24	24	0.64	0.28	0.27	1.89E-17	5.64E-14	5.38E-14	8.02E-07	9.50E-07	9.28E-07

Table 3. Decrement Percentage of the Number of Inner Iterations for the Newton-GS and the Newton-2EGGS compared to the Newton-Jacobi

Test Problem (initial point)	Number of inner iteration (%)	
	N_{GS}	N_{2EGGS}
1(a)	53.13-55.10	53.16-55.10
1(b)	54.31-55.74	54.31-55.79
1(c)	52.97-54.52	52.29-54.52
2(a)	50.42-**73.48**	64.11-**79.06**
2(b)	57.93-63.42	64.63-67.39
2(c)	58.28-64.94	64.63-68.61
3(a)	**50.18**-51.60	**50.67**-51.60
3(b)	53.11-54.37	53.11-54.37
3(c)	50.73-51.94	51.10-52.05

Table 4. Comparison of Speed Ratio for the Newton-2EGSOR with the Newton-GS and the Newton-SOR

Test Problem (initial point)	Total execution time (seconds)			Speed ratio		
	N_J (I)	N_{GS} (II)	N_{2EGGS} (III)	$\frac{I}{II}$	$\frac{I}{III}$	$\frac{II}{III}$
1(a)	7.31	4.58	4.24	1.60	1.72	1.08
1(b)	3.63	2.21	2.00	1.64	1.82	1.11
1(c)	6.71	3.70	3.65	1.81	1.84	1.01
2(a)	3175.50	1844.61	1168.20	1.72	2.72	**1.58**
2(b)	3517.64	2008.45	1271.62	1.75	2.77	**1.58**
2(c)	4078.53	1879.18	1275.92	2.17	**3.20**	1.47
3(a)	2.01	0.66	0.62	3.05	3.24	1.06
3(b)	1.05	0.48	0.42	2.19	2.50	1.14
3(c)	1.37	0.61	0.58	2.25	2.36	1.05

5. Conclusions

In this paper, we have proposed an alternative algorithm for solving large-scale unconstrained optimization problems with an arrowhead Hessian matrix. Through the implementation of block iterative method for finding the Newton direction, we can point out that the approach we proposed is superior compared to the reference methods. This point can be clarified in Table 3 and Table 4. As described in Table 3, the decrement percentage of the number of inner iteration is in the range 50.18%-73.48%, and 50.67%-79.06% correspond to the Newton-2EGGS and the Newton-GS methods compared to the Newton-Jacobi method. Furthermore, from the speed ratio shown in Table 4, we observe that our method is up to 1.58 times faster than the Newton-GS method and up to 3.20 times more rapid than the Newton-Jacobi method. Thus, it can be concluded that our proposed alternative method can show significant improvement in the number of inner iterations and execution time compared to the Newton-GS and the Newton-Jacobi iterative methods. In addition, we expect that the proposed Newton's direction presented in this paper can be extended to 4-point block iterative method, as in Ghazali *et al.* [21].

Acknowledgements

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

REFERENCES

[1] Sun W & Yuan Y, Optimization Theory and Methods-Nonlinear Programming. United States: Springer, (2006), pp. 119-302

[2] Esmaeili H & Kimiaei M, "An Improved Adaptive Trust-Region Method for Unconstrained Optimization", Mathematical Modeling and Analysis, Vol. 19, No. 4, (2014), pp. 469-490

[3] Shuai Z, Zong W & Jing Z, "An Extended Nonmonotone Line Search Technique for Large-scale Unconstrained Optimization", Journal of Computational and Applied Mathematics, Vol. 330, (2018), pp. 586-604

[4] Esmaeili H, Rostami M & Kimiaei M, "Extended Dai-Yuan Conjugate Gradient Strategy for Large-scale Unconstrained Optimization with Applications to Compressive Sensing", Filomat, Vol. 32, No. 6, (2018), pp. 2173-2191, https://doi.org/10.2298/FIL1806173E

[5] Moyi AU & Leong WJ, "A Sufficient Descent Three-term Conjugate Gradient Method via Symmetric Rank-one Update for Large-scale Optimization", Optimization, Vol.64, No.1, (2016), pp.121-143

[6] Dembo RS & Steihaug T, "Truncated-Newton Algorithms for Large-scale Unconstrained Optimization", Mathematical Programming, Vol. 26, (1983), pp. 190-212

[7] Grapsa TN, "A Modified Newton Direction for Unconstrained Optimization", Optimization, Vol. 63, No. 7, (2014), pp. 983-1004

[8] Boggs PT & Byrd RH, "Adaptive, Limited-Memory BFGS Algorithms for Unconstrained Optimization", SIAM Journal on Optimization, Vol. 29, No. 2, (2019), pp. 1282-1299

[9] Evans DJ. "Group explicit iterative methods for solving large linear systems". International Journal of Computer Mathematics. Vol. 11, (1985), pp. 81-108

[10] Yousif WS & Evans DJ. "Explicit group over-relaxation methods for solving elliptic partial differential equations". Mathematics and Computers in Simulation. Vol. 28, (1986), pp. 453–66

[11] Yousif WS & Evans DJ. "Explicit de-coupled group iterative methods and their implementations". Parallel Algorithms and Application. Vol. 7, (1995), pp. 53-71

[12] Sulaiman J, Hasan MK, Othman M & Karim SAA, "Newton-EGMSOR Methods for Solution of Second Order Two-Point", Journal of Mathematics and System Science. Vol. 2, (2012), pp. 185–190

[13] Sulaiman J, Hasan MK, Othman M & Karim SAA, "Application of Block Iterative Methods with Newton Scheme for Fisher's Equation by Using Implicit Finite Difference", Jurnal Kalam. Vol. 8, No. 1, (2015), pp. 39-46

[14] Othman M & Abdullah AR, "An efficient four points modified explicit group poison solver". International Journal of Computer Mathematics. Vol. 76, (2000), pp. 203–217

[15] Stanimirovic PS, Katsikis VN & Kolundzija D, "Inversion and pseudoinversion of block arrowhead matrices", Applied Mathematics and Computation, Vol. 341, (2019), pp. 379-340

[16] Varga RS, Matrix iterative analysis. Berlin: Springer, (2000), pp.63-101

[17] Saad Y, Iterative Methods for Sparse Linear Systems. United States: PWS Publishing Company, (1996), pp. 103-128

[18] Ghazali K, Sulaiman J, Dasri, Y, Gabda D, "Newton-MSOR Method for Solving Large-Scale Unconstrained Optimization Problems with an Arrowhead Hessian Matrices". Transactions on Science and Technology Vol. 6, No. 2-2, (2019), pp. 228-234

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

[20] Andrei N, "An unconstrained optimization test function collection. Advanced Modeling and Optimization". An Electronic International Journal, Vol. 10, No. 1, (2008), pp. 147-161

[21] Ghazali K, Sulaiman J, Dasri, Y, Gabda D, "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 Springer, Singapore (2019), pp. 401-411