SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Newton-SOR Iterative Method with Lagrangian Function for Large-Scale Nonlinear Constrained Optimization Problems

Peng Cheng[1,2], Jumat Sulaiman[1,*], Khadizah Ghazali[1], Majid Khan Majahar Ali[3], Ming Ming Xu[4]

1 Faculty of Science and Natural Resources, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia
2 Chongqing College of Finance and Economics, 402160 Chongqing, China
3 School of Mathematical Sciences, Universiti Sains Malaysia, 11800 Gelugor, Pulau Pinang, Malaysia
4 School of Mathematics and Information Technology, Xingtai University, Xingtai, 054001 Hebei, China

ARTICLE INFO

Article history:
Received 29 March 2023
Received in revised form 13 July 2023
Accepted 29 April 2024
Available online 9 June 2024

Keywords:

Nonlinear constrained optimization problem; Newton iteration; SOR iteration; Computational efficiency

ABSTRACT

With the rapid development of computer technology and the wide application of nonlinear constrained optimization problems, many researchers are committed to solve large-scale constrained optimization problems. In this article, a new combinatorial iterative method is proposed on the basis of previous research, which can efficiently solve large-scale nonlinear constrained optimization problems. We first transform a large nonlinear constrained optimization problem into a corresponding unconstrained optimization problem by using the Lagrange multiplier method, and then the Newton iterative method is used to solve the transformed unconstrained optimization problem. To perform the iterative method, we need to compute its Newton direction, and the inverse matrix of Hessian matrix. To deal with the large-scale Hessian matrix, calculation of the inverse matrix for the Hessian matrix may not be easy to be determined. To overcome this issue, we propose the matrix iteration method to compute the Newton direction by solving the linear system as the internal iteration solution. Therefore, this paper investigates a Newton-SOR (NSOR) iterative method to solve this problem, in which the proposed NSOR iterative method combines the Newton method with Successive Over-Relaxation (SOR) iterative method. Based on the numerical experiments, the effectiveness of the proposed NSOR iterative method is more effective than the Newton-Gauss-Seidel (NGS) iterative method in terms of computing time and number of iterations.

1. Introduction

Nonlinear optimization problems widely exist in engineering design, economic management, military research, and other applications with the rapid development of computer information technology. The requirements for large-scale optimizations are also increasing. For example, in petroleum exploration, aerospace, data mining and many other optimization problems, there are a lot of unknown variables being used to the optimization problems. In other words, the objective function is becoming more complex, and the dimension to the problem is also large-scale. Therefore,

[*] *Corresponding author.*
E-mail address: jumat@ums.edu.my

https://doi.org/10.37934/araset.46.2.251262

the solution of large-scale constrained optimization problems has already become research hotspots for researchers. Due to these issues, this paper mainly considers the following large-scale constrained optimization problems.

$$\begin{aligned} &\min \quad f(x) \\ &s.t. \quad g_i(x) \leq 0, \quad i = 1, \dots, q, \\ &g_i(x) = 0, \quad i = q+1, \dots, m. \end{aligned} \tag{1}$$

where $x = (x_1, x_2, \dots, x_n)^T \in R^n, f(x): R^n \to R$ and $g_i: R^n \to R, i = 1,2, \dots, m+q$ are continuous functions on R^n.

In recent years, many researchers began to extend some mature and effective algorithms to solve large-scale nonlinear constrained optimization problems. There are many classical methods for handling constraints, including Lagrangian multiplier method [1], penalty function method [2-5], obstacle function method, adaptive dynamic penalty function method, etc. Among them, the classical Lagrangian multiplier method is one of important methods for solving constrained optimization problems. Apart from its effective calculation, this method is preferred by many researchers because it can combine multiple constraints to reduce the problem as an unconstrained optimization problem. In addition to that, the Lagrangian method is mainly used to solve extreme value problems under constraint conditions, and additional variables (Lagrangian multipliers) need to be introduced to solve constrained optimization problems directly and accurately. Although strict implementation of constraints has obvious advantages, this method has some difficulties due to the extra cost of solving the multiplier. It means that the Lagrangian multiplier method will result in zero elements on the diagonal of the equations and the direct solution method will still be used at this time, in which this matter will bring additional difficulties in calculation. However, the formula based on the Lagrange multiplier method is widely used. Until today, we can still find the application cases of this method [6-8]. Therefore, in this paper, we mainly use the Lagrangian multiplier method to transform constrained optimization problems. By introducing Lagrange multipliers, an optimization problem with n variables and m constraints can be transformed into a corresponding unconstrained optimization problem with $n + m$ variables. To solve problem Eq. (1), let us define the Lagrangian function as

$$L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) \, . \tag{2}$$

where, $\lambda = (\lambda_1, \lambda_2, \cdots, \lambda_m)^T$ is a Lagrangian multiplier.

By defining the Lagrangian function, we can see that we have transformed the optimization problem with inequality constraints in Eq. (1) into an unconstrained optimization problem of Eq. (2). For solving the unconstrained optimization problem of Eq. (2), we can use the direct search method [9-11] and the indirect search method [12-15] to get the optimal value of this function. However, all these methods have the limitation of their slow convergence, especially in dealing with large-scale optimization problems. For instance, direct search methods mainly include alternating direction method [9], simplex method [10], conjugate direction method [11], etc. These methods only need to calculate function values, so they have the advantages of easy use, simple structure, small memory, etc. The disadvantage is that most of these methods rely on intuitive skills. Indirect search methods mainly include gradient descent method [12], conjugate gradient method [13], Newton method [14] and quasi-Newton method [15]. Many researchers have improved these methods and obtained many effective methods, such as [16,17]. Based on these methods, the Newton method has become the focus of research because of its secondary locally convergence rate. Known as a locally convergence method, the advantage of the Newton method is that if the initial point is close to the minimum point, it will show its good convergence. With the characteristic of its locally convergence, we mainly consider the Newton method to solve the large-scale unconstrained optimization problem transformed by Lagrangian multiplier method in Eq. (2).

Although the Newton method has a fast convergence rate in theory [18], in practice, this method requires that its Hessian matrix must be positive definite. The selection of the initial point needs to select an initial point closer to the optimal solution to ensure the convergence of the method. At the same time, additional storage space is required to calculate the value of the Hessian matrix during the calculation, particularly when large-scale problems are involved. Therefore, many researchers have improved the original Newton method or combined it with other calculation methods to conquer the difficulty of this storage space [19-23]. For example, in [20], Grapsa proposed a component approximation gradient (CAG) method. When solving unconstrained optimization problems, this method uses appropriate gradient correction to make it have a descending characteristic, so it is not suitable for linear shrinkage technology. By using the improved Cholesky decomposition algorithm and replacing Hessian matrix in the objective function with positive definite matrix, an improved Newton method [21] is proposed to solve unconstrained optimization problems with local minimization. In [22], the author proposed a new method to solve unconstrained optimization problems by using the Newton method and steepest descent method and proved the global convergence of the algorithm. In reference [23], the author proposed a method combining Newton direction and inverse gradient direction, which improved the convergence speed of Newton method.

We observe that all improvements of these methods are related to solve the Newton direction of a linear system. If it refers to a small-scale problem, the original Newton method can be used to solve it well. However, to solve any large-scale problem, it is not useful to use the original Newton method because of high calculation cost. Therefore, the method needs to be combined with some iterative methods to improve its performance. Apart from this Newton method, Young [24] proposed an effective point iterative method to solve any large-scale linear system. This iterative method can be classified as a point iteration family and known as the Successive Over-Relaxation (SOR) iterative method. Further applications of the iterative method have been discussed by [25,26]. Due to the fact of its fast convergence rate, we combine the Newton method with the SOR iteration method, known as Newton-SOR (NSOR) iterative method. Hopefully, the proposed NSOR can solve large-scale unconstrained optimization problems transformed by the Lagrangian multiplier method, see in Section 4. Clearly, the main idea of the modified algorithm is the Newton method, in which the SOR iteration method is used to calculate Newton direction. During implementation of the iteration process, we perform the Newton's method as external iteration process and the SOR iterative method occurs for internal iteration process. Such research ideas mainly come from literature [27-32]. It should be noted that the NSOR method used in this paper is different from the improved methods of other Newton methods mentioned above. This paper mainly solves the constrained optimization problem after transformation via the Lagrangian multiplier method, and its Hessian matrix is not positive definite. To test the computational efficiency of NSOR iteration, we compare it with the Newton-GS (NGS) iterative method, which is known as a combination of the Newton method and Gauss-Seidel (GS) iteration.

Next, in Section 2, we briefly describe the formulation of the Hessian matrix, which is a non-positive definite Newton. Then in Section 3, we give the formulation and algorithm of the NSOR iteration. In Section 4, we will report the numerical experimental results of the NSOR iterative method and the results of the NGS method iteration act as a reference. At the same time, we give some main conclusions in Section 5.

2. Methodology

In this section, we briefly describe the derivation of the Hessian matrix, which is a non-positive definite Newton, and then construct the linear system, which is generated from calculating the Newton direction. Consequently, the following subsections will discuss three steps to implement the NSOR iterative method for solving the proposed problem in Eq. (1).

2.1 Derive Newton Iterative Steps with the Full Hessian Matrix

To make each iteration direction of the objective function $L(x)$ in the unconstrained optimization problem in Eq. (2) needs to follow the descending direction of the current point function value. Therefore, the Taylor series expansion is performed on the function $L(x)$ to the second order, and then the formulation of the Newton iteration for solving unconstrained optimization is derived as follows:

$$L(x) \approx L(x_k) + \nabla L(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 L(x_k)(x - x_k), \tag{3}$$

where $\nabla L(x_k)$ is the gradient vector formed by the first partial derivatives of $L(x)$, $\nabla^2 L(x_k)$=$H(x_k)$ is a square matrix formed by the second partial derivative of $L(x)$. Because the problem of finding the extreme value of a function can be transformed into that the derivative function is equal to 0, so take the derivative of Eq. (3) to be 0, that is, the following equation is equal to 0:

$$\nabla L(x_k) + H(x_k)(x - x_k) = 0. \tag{4}$$

According to Eq. (4), the following equation is established:

$$x = x_k - [H(x_k)]^{-1} \nabla L(x_k), \tag{5}$$

where $[H(x_k)]^{-1}$ is the inverse matrix of the Hessian matrix $H(x_k)$. Then the calculation of the current value of the current point, x_{k+1} can be obtained from Eq. (5) as follows:

$$x_{k+1} = x_k - [H(x_k)]^{-1} \nabla L(x_k). \tag{6}$$

Therefore, Newton direction, d_k is given as

$$d_k = x_{k+1} - x_k = -[H(x_k)]^{-1} \nabla L(x_k). \tag{7}$$

According to Eq. (7), the following Newton equation can be obtained as

$$H(x_k) d_k = -\nabla L(x_k). \tag{8}$$

Because it is uncertain whether the Hessian matrix, $H(x_k)$ is positive definite or not. Subsequently, we need to multiply the transposed matrix $[H(x_k)]^{\mathrm{T}}$of the Hessian $H(x_k)$ matrix on both sides of Eq. (8) to ensure the positive definite of the coefficient matrix in the Newton equation as follows:

$$[H(x_k)]^{\mathrm{T}} H(x_k) d_k = -[H(x_k)]^{\mathrm{T}} \nabla L(x_k) \ . \tag{9}$$

Because $[H(x_k)]'H(x_k)$ is positive definite, the Newton direction is the descending direction,

$$[\nabla f(x_k)]^T[H(x_k)]^{\mathrm{T}}H(x_k)d_k = -[\nabla f(x_k)]^T[H(x_k)]^{\mathrm{T}}\nabla L(x_k) < 0. \quad (10)$$

But in this paper, we study the large-scale nonlinear inequality constrained optimization problem, and use the Lagrangian multiplier method to transform it into an unconstrained optimization problem in Eq. (2), in which $L\ (x,\lambda)$ is known as an objective function with $n+m$ variables. It is easy to see the variables from the function λ , in which the degree of is 1, so the diagonal elements of the Hessian matrix obtained by the Newton method must have 0 elements. Suppose the representation of the Hessian matrix, H_k is given as follows:

$$H_k = \begin{bmatrix} \frac{\partial^2 L}{\partial x_1^2} & \frac{\partial^2 L}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 L}{\partial x_1 \partial x_n} & \frac{\partial^2 L}{\partial x_1 \partial \lambda_1} & \cdots & \frac{\partial^2 L}{\partial x_1 \partial \lambda_m} \\ \frac{\partial^2 L}{\partial x_2 \partial x_1} & \frac{\partial^2 L}{\partial x_2^2} & \cdots & \frac{\partial^2 L}{\partial x_2 \partial x_n} & \frac{\partial^2 L}{\partial x_2 \partial \lambda_1} & \cdots & \frac{\partial^2 L}{\partial x_2 \partial \lambda_m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 L}{\partial x_n \partial x_1} & \frac{\partial^2 L}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 L}{\partial x_n^2} & \frac{\partial^2 L}{\partial x_n \partial \lambda_1} & \cdots & \frac{\partial^2 L}{\partial x_n \partial \lambda_m} \\ \frac{\partial^2 L}{\partial \lambda_1 \partial x_1} & \frac{\partial^2 L}{\partial \lambda_1 \partial x_2} & \cdots & \frac{\partial^2 L}{\partial \lambda_1 \partial x_n} & \frac{\partial^2 L}{\partial \lambda_1^2} & \vdots & \frac{\partial^2 L}{\partial \lambda_1 \partial \lambda_m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 L}{\partial \lambda_m \partial x_1} & \frac{\partial^2 L}{\partial \lambda_m \partial x_2} & \cdots & \frac{\partial^2 L}{\partial \lambda_m \partial x_n} & \frac{\partial^2 L}{\partial \lambda_m \partial \lambda_1} & \cdots & \frac{\partial^2 L}{\partial \lambda_m^2} \end{bmatrix}. \quad (11)$$

2.2 Derivation of Proposed Iterative Methods

By mean of the coefficient matrix of Eq. (11) to solve Eq. (9) directly, we can also consider using the Gaussian elimination method [33] or simultaneous method [34]. Because of the large-scale of the coefficient matrix in Eq. (11), more storage space and computing time are required. For solving this problem, both methods will have their difficulty to deal with large-scale problems. As we know, we need to pay a high calculation cost to compute the value of the Newton direction by solving the Newton Eq. (9), which arises from first and second partial derivatives of the Lagrangian function of Eq. (2). Therefore, we consider improving the Newton direction, that is, using an iterative method to solve the Newton direction. Eq. (9) is just a linear system that can be solved by iterative methods. In view of these iterative methods, we consider an iterative method to solve the linear system, see Sulaiman *et al.,* [27,28] and Ghazali *et al.,* [29-31]. To gain the internal iteration solution, let us rewrite the linear system of Eq. (9) as follows:

$$Ad = b, \quad (12)$$

where,

$$A=\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} & a_{1,n+1} & \cdots & a_{1,n+m} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} & a_{2,n+1} & \cdots & a_{2,n+m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} & a_{n,n+1} & \cdots & a_{n,n+m} \\ a_{n+1,1} & a_{n+1,2} & \cdots & a_{n+1,n} & a_{n+1,n+1} & \vdots & a_{n+1,n+m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{n+m,1} & a_{n+m,2} & \cdots & a_{n+m,n} & a_{n+m,n+1} & \cdots & a_{n+m,n+m} \end{bmatrix}, d=\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n+m-1} \\ d_{n+m} \end{bmatrix}, b=\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n+m-1} \\ b_{n+m} \end{bmatrix},$$

Clearly in Eq. (11), we need to solve the linear system for obtaining the approximate value of the Newton direction. Consequently, the following subsection will discuss the formulation of the SOR iterative method.

2.3 SOR Point Iteration

Based on the large-scale diagonal matrix H_k with 0 elements, we need to use such as a coefficient matrix H_k being considered to solve the linear system in Eq. (8). In line with testing the computational efficiency of NSOR iteration, we propose the SOR iterative method [30,31] to solve the linear system of Eq. (8). To ensure the proposed iterative method being used to solve the linear system of Eq. (8), in which each diagonal element of its Hessian matrix, $H(x_k)$ is not zero, we need to rewrite the linear system in Eq. (8) as:

$$H_k' \cdot H_k d_k = -H_k' \cdot g_k.$$

Again, let the above linear system be rewritten as

$$Hd = g, \tag{13}$$

where $H = H_k' \cdot H_k$, $g = -H_k' \cdot g_k$, $d^T = [d_1, d_2, d_3 \dots d_n]$, and $g^T = [g_1, g_2, g_3, \dots, g_n]$. Before starting to perform any point iterative method, we need to decompose the coefficient matrix, H into the following form:

$$H = D - E - U, \tag{14}$$

where D is the non-zero diagonal part of H, L is the strictly lower triangle part, and U is the strictly upper triangle part. The decomposition in Eq. (14) is applied to linear system of Eq. (13), and then the expression form of the SOR iterative method can be stated as [29-32]:

$$d_{k+1} = (D - \omega E)^{-1}(\omega U + (1-\omega)D)d_k + \omega(D - \omega E)^{-1}g, \tag{15}$$

where, ω represents the relaxation factor. Its optimal value is in the range of [0, 2), and its true value is selected according to the minimum number of internal iterations.

As taking $\omega = 1$, the SOR iterative method is naturally transformed into the GS iterative method, and its formulation can be given as

$$d_{k+1} = (D - E)^{-1}Ud_k + (D - E)^{-1}g. \tag{16}$$

Therefore, we use the formulation of SOR iterative method to calculate an approximate value of the Newton direction of Eq. (9) by solving the linear system in Eq. (13). Clearly to solve the proposed problem in Eq. (1) via the NSOR iterative method, the following is Algorithm 1 that may be used to describe all steps to get the approximate optimal value of the Lagrangian function of Eq. (2).

Algorithm 1: NSOR Scheme

Step 1. Assign the initial value x_0 , accuracy threshold $\varepsilon_1 = 10^{-5}$, $\varepsilon_2 = 10^{-10}$, $\alpha_k = 1$ and let k: =0.

Step 2. Calculate gradient g_k and matrix H_k.
If $\|g_k\| < \varepsilon$, that is, the value of the gradient at this point is close to 0, then the extreme point is reached, and go to Step 6,
otherwise, go to Step 3.1.

Step 3.
Step 3.1. Calculate the matrix $H = H_k' \cdot H_k$, matrix $D,\ E,\ U$.
Step 3.2. Calculate the search direction $d_{k+1} = (D - \omega E)^{-1}(\omega U + (1-\omega)D)d_k + \omega(D - \omega E)^{-1} g$.
Step 3.3. Calculate the convergence condition, if $\|d_{k+1} - d_k\| < \varepsilon_2$, then go to step 4, otherwise go to Step 3.2.

Step 4. Calculate the new iteration point as $x_{k+1} = x_k + \alpha_k d_k$.

Step 5. Let $k := k + 1$, go to step 2.

Step 6. Display the numerical results

3. Results
3.1 Symbol Description

We use the following symbol abbreviations as shown in Table 1.

Table1
Description of symbols used in the depicted results

Notation	Description
n	Number of variables
M	Method
ω	Optimal value of ω in SOR iterative method
NOI	Number of internal iterations
NGS	Newton-GS method
NSOR	Newton-SOR method
TM	computational time (Unit: Second)
LOP	Local optimal point $(x_1, \dots, x_n)$
L2-g	L2 Norm of Function Gradient at Termination of Calculation
FOV	Local optimal Value $f(x)$

3.2 The Test Functions

In this section, we have carried out the numerical experiments using Algorithm 1 and highlighted the observed comparison results. In Table 2, there are four test functions considered to execute the numerical experiment part. All four test functions are classified as nonlinear inequality constrained optimization problems. Basically, each test function is selected based on the type of its Hessian matrix, which is known as a full Hessian matrix and not necessarily as a positive definite matrix. At the beginning of calculation over each test function, we use the randomly selected initial point $x_0 = (1, \dots, 1)$. For the implementation of numerical experiments, we use the MATLAB software to test the effectiveness of the proposed NSOR and NGS iteration algorithms by considering five different order Hessian matrices such as *n* = 100, 200, 300, 400, 500. Therefore, these five different values of n are equivalent to providing a total of 20 test cases. Clearly, each test function has a different Hessian matrix. For the sake of implementing the iteration process, Algorithm 1 considered at two convergence tests such as the external iteration condition as $||\nabla f(x)|| < 10^{-5}$, and the internal iteration condition as $||d_k - d_0|| < 10^{-10}$.

Table 2
Test function of numerical experiments

Test No	Test function	Lagrangian function	FOV	LOP
1	$min f(x) = \sum_{i=1}^{n} x_i^2$ $s.t.\ \sum_{i=1}^{n} -x_i - n \leq 0$	$min P_\lambda(x) = \sum_{i=1}^{n} x_i^2 + \lambda(\sum_{i=1}^{n} -x_i - n)$	$f^* = n$	$x^* = (-1, \dots, -1)$
2	$min f(x) = \sum_{i=1}^{n}(x_i - x_1)^2$ $s.t.\ \sum_{i=1}^{n} -x_i - 1 \leq 0$	$min P_\lambda(x) = \sum_{i=1}^{n}(x_i - x_1)^2 + \lambda(\sum_{i=1}^{n} -x_i - 1)$	$f^* = 0$	Different dimensions n have different advantages
3	$min\ f(x) = \sum_{i=1}^{n}(x_i^2 - x_i)$ $s.t. \sum_{i=1}^{n}(x_i^2 - 1) \leq 0$	$min P_\lambda(x) = \sum_{i=1}^{n}(x_i^2 - x_i) + \lambda(\sum_{i=1}^{n}(x_i^2 - 1))$	$f^* = 0$	$x^* = (1, \dots, 1)$
4	$min f(x) = \sum_{i=1}^{n}(x_i - x_1)^2$ $s.t. \sum_{i=1}^{n}(x_i^2 - 1) \leq 0$	$min P_\lambda(x) = \sum_{i=1}^{n}(x_i - x_1)^2 + \lambda(\sum_{i=1}^{n}(x_i^2 - 1))$	$f^* = 0$	$x^* = (1, \dots, 1)$

To make a comparative analysis, we performed both NGS and NSOR iterative methods and tabulated all numerical results in Table 3. Since the external iterations of all four test functions are one, we mainly considered the total computing time and internal iterations.

Table 3
Calculation Results of Newton-GS and Newton-SOR

Test No	n	ω	NOI		TM		L2-g		FOV	
			NGS	NSOR	NGS	NSOR	NGS	NSOR	NGS	NSOR
1	100	0.250	736	199	48.25	26.29	8.20E-10	1.02E-9	100	100
	200	0.140	2674	386	440.36	255.44	1.62E-9	2.19E-9	200	200
	300	0.100	5819	573	1803.96	825.51	2.44E-9	3.19E-9	300	300
	400	0.070	10168	756	5868.83	3038.18	3.21E-9	6.51E-9	400	400
	500	0.050	15708	948	11482.07	4732.38	4.01E-9	5.13E-9	500	500
2	100	0.120	8225	883	387.30	56.24	4.40E-8	3.51E-8	9.02E-18	8.05E-18
	200	0.063	31900	1735	4000.81	398.59	1.44E-8	1.39E-7	4.18E-17	4.32E-17
	300	0.042	70742	2579	17488.25	1559.04	1.15E-7	1.44E-7	4.79E-17	5.29E-17
	400	0.032	124698	3414	57163.24	4033.78	1.02E-7	2.49E-7	1.89E-16	1.07E-16
	500	0.020	193499	4245	121971.22	9510.46	2.51E-7	4.39E-7	1.22E-16	2.16E-16
3	100	0.260	579	166	44.20	31.29	8.15E-10	1.10E-9	1.52E-19	3.04E-19
	200	0.150	1946	310	362.81	222.63	1.63E-9	2.17E-9	6.55E-19	1.18E-18
	300	0.110	4032	451	1405.12	1065.80	2.43E-9	3.09E-9	1.46E-18	2.37E-18
	400	0.070	6784	581	6254.08	2941.89	3.24E-9	4.62E-9	2.61E-18	5.33E-18
	500	0.058	10170	711	9262.85	8793.55	4.02E-9	5.67E-9	4.02E-18	8.03E-18

4	100	0.160	1549	292	91.83	33.48	2.35E-8	3.25E-8	1.97E-18	3.76E-18
	200	0.080	5326	544	854.66	279.60	5.73E-9	2.52E-8	6.63E-18	3.09E-19
	300	0.070	11040	791	3028.51	1036.17	1.17E-7	6.97E-8	1.70E-17	1.34E-17
	400	0.048	18510	1012	9717.70	3240.24	1.75E7	2.41E-7	2.92E-17	5.52E-17
	500	0.040	27621	1240	26796.84	6148.74	2.51E-7	3.42E-7	4.73E-17	8.68E-17

3.3 Comparison Results

As we observe in Subsection 3.1, two proposed Newton iterative methods have successfully solved the proposed problems in Eq. (2). It means that we calculated the numerical results using the NGS and NSOR iteration via Algorithm 1 and all observed numerical results obtained from both iteration methods have been presented in Table 3. Since the number of external iterations of Algorithm 1 is one, only the values of internal iterations and the computing time in seconds for both proposed iteration methods are compared as measurement parameters. Then the function value and gradient norm at the end of the calculation are also shown in Table 3. For the sake of comparison, we reserve two decimal places for all values listed in Table 3. Therefore, the value 1 of the function gradient norm at the iteration where the termination is performed, is lesser than the convergence condition. Note that in Table 3, 20 test cases indicate that its approximate value is the optimal value of the problem. These values were obtained using both proposed methods. For n = 100, 200, 300, 400, 500, we have five different optimal values of ω for the NSOR iterative method in Table 3. Obviously, all these optimal values are still the range of the interval, $[0, 2)$. By taking a consideration for all numerical results in Table 3, we performed a comparative analysis between number of iterations and computing time of the NSOR iterative method compared to the NGS iterative method as depicted in Table 4.

Table 4
Decreasing Percentage of Iterations of Newton-SOR Relative to Newton-GS

Test case	Range of decreasing percentage of iterations	Computing time		
		NGS (I)	NSOR (II)	$\frac{I}{II}$
1	72.96 ~93.91	19643.47	10468.08	1.88
2	89.26 ~97.81	201010.8	15558.11	12.92
3	71.33 ~93.01	17329.06	13055.17	1.32
4	81.15 ~95.51	40489.55	10738.23	3.77

4. Conclusions

In this article, the combination of Newton method and Successive Over-Relaxation (SOR) iterative method is more effective than the reference NGS iterative method in solving large-scale unconstrained optimization problems transformed by the Lagrange multiplier method. Based on the findings in Table 3, the proposed NSOR iterative method has fewer iterations and shorter calculation time compared with the reference method. It means that the conclusion indicates the high efficiency of the proposed iterative method. As we can see in Table 4, the decreasing percentages of iteration times and computing time of the NSOR iterative method have reduced the calculation storage when the iteration process used the optimal value of ω. As a result, the number of internal iterations using the NSOR method is lesser than that using the NGS iterative method with a minimum reduction of 71.33% and a maximum reduction of 97.35%. It means that the calculation of the NSOR iterative method is faster than the reference method when the over-relaxation factor ω is used. At the same time, in Table 4, the calculation time of the proposed iterative method is 12.92 times faster than the

NGS iterative method. Therefore, it can be concluded that our proposed iterative method can substantially improve the number of iterations and computing time compared with the reference method. To expand this research, the combination concept of Newton method should be imposed with the existing block iterative methods (see in Ghazali *et al.,* [30,31]).

Acknowledgement
This research was funded by Universiti Malaysia Sabah for funding this research under UMSGreat research grant for postgraduate student (GUG0568-1/2022).

References
[1] Rockafellar, R. Tyrell. "The multiplier method of Hestenes and Powell applied to convex programming." *Journal of Optimization Theory and applications* 12, no. 6 (1973): 555-562. https://doi.org/10.1007/BF00934777
[2] Laptin, Yu P., and T. O. Bardadym. "Problems related to estimating the coefficients of exact penalty functions." *Cybernetics and Systems Analysis* 55 (2019): 400-412. https://doi.org/10.1007/s10559-019-00147-2
[3] Nguyen, V. H., and J. J. Strodiot. "On the convergence rate for a penalty function method of exponential type." *Journal of Optimization Theory and Applications* 27 (1979): 495-508. https://doi.org/10.1007/BF00933436
[4] Cominetti, Roberto, and Jean-Pierre Dussault. "Stable exponential-penalty algorithm with superlinear convergence." *Journal of Optimization Theory and Applications* 83, no. 2 (1994): 285-309. https://doi.org/10.1007/BF02190058
[5] Hassan, M., and A. Baharum. "A new logarithmic penalty function approach for nonlinear constrained optimization problem." *Decision Science Letters* 8, no. 3 (2019): 353-362. https://doi.org/10.5267/j.dsl.2018.8.004
[6] Fischer, K. A., and P. Wriggers. "Frictionless 2D contact formulations for finite deformations based on the mortar method." *Computational Mechanics* 36 (2005): 226-244. https://doi.org/10.1007/s00466-005-0660-y
[7] Hartmann, S., and E. Ramm. "A mortar based contact formulation for non-linear dynamics using dual Lagrange multipliers." *Finite Elements in Analysis and Design* 44, no. 5 (2008): 245-258. https://doi.org/10.1016/j.finel.2007.11.018
[8] Tur, M., F. J. Fuenmayor, and P. Wriggers. "A mortar-based frictional contact formulation for large deformations using Lagrange multipliers." *Computer Methods in Applied Mechanics and Engineering* 198, no. 37-40 (2009): 2860-2873. https://doi.org/10.1016/j.cma.2009.04.007
[9] Gabay, Daniel, and Bertrand Mercier. "A dual algorithm for the solution of nonlinear variational problems via finite element approximation." *Computers & mathematics with applications* 2, no. 1 (1976): 17-40. https://doi.org/10.1016/0898-1221(76)90003-1
[10] Dantzig, George B., Alex Orden, and Philip Wolfe. "The generalized simplex method for minimizing a linear form under linear inequality restraints." *Pacific Journal of Mathematics* 5, no. 2 (1955): 183-195. https://doi.org/10.2140/pjm.1955.5.183
[11] Powell, Michael JD. "An efficient method for finding the minimum of a function of several variables without calculating derivatives." *The computer journal* 7, no. 2 (1964): 155-162. https://doi.org/10.1093/comjnl/7.2.155
[12] Napitupulu, H., I. Bin Mohd, Y. Hidayat, and S. Supian. "Steepest descent method implementation on unconstrained optimization problem using C++ program." In *IOP Conference Series: Materials Science and Engineering*, vol. 332, no. 1, p. 012024. IOP Publishing, 2018. https://doi.org/10.1088/1757-899X/332/1/012024
[13] Moyi, Aliyu Usman, and Wah June Leong. "A sufficient descent three-term conjugate gradient method via symmetric rank-one update for large-scale optimization." *Optimization* 65, no. 1 (2016): 121-143. https://doi.org/10.1080/02331934.2014.994625
[14] Babaie-Kafaki, Saman. "Computational approaches in large-scale unconstrained optimization." *Big Data Optimization: Recent Developments and Challenges* (2016): 391-417. https://doi.org/10.1007/978-3-319-30265-2_17
[15] Aderibigbe, Felix Makanjuola, Kayode James Adebayo, and Adejoke O. Dele-Rotimi. "On quasi-newton method for solving unconstrained optimization problems." *America Journal of Applied Mathematics* 3, no. 2 (2015): 47-50. https://doi.org/10.11648/j.ajam.20150302.13
[16] Fasano, Giovanni, and Stefano Lucidi. "A nonmonotone truncated Newton–Krylov method exploiting negative curvature directions, for large scale unconstrained optimization." *Optimization Letters* 3 (2009): 521-535. https://doi.org/10.1007/s11590-009-0132-y
[17] Kou, Cai-Xia, and Yu-Hong Dai. "A modified self-scaling memoryless Broyden–Fletcher–Goldfarb–Shanno method for unconstrained optimization." *Journal of Optimization Theory and Applications* 165 (2015): 209-224. https://doi.org/10.1007/s10957-014-0528-4

[18] Nocedal, Jorge, and Stephen J. Wright, eds. *Numerical optimization*. New York, NY: Springer New York, 1999. https://doi.org/10.1007/b98874

[19] Polyak, Boris T. "Newton's method and its use in optimization." *European Journal of Operational Research* 181, no. 3 (2007): 1086-1096. https://doi.org/10.1016/j.ejor.2005.06.076

[20] Grapsa, Theodoula N. "A modified Newton direction for unconstrained optimization." *Optimization* 63, no. 7 (2014): 983-1004. https://doi.org/10.1080/02331934.2012.696115

[21] Shen, Chungen, Xiongda Chen, and Yumei Liang. "A regularized Newton method for degenerate unconstrained optimization problems." *Optimization Letters* 6 (2012): 1913-1933. https://doi.org/10.1007/s11590-011-0386-z

[22] Shi, Yixun. "Globally convergent algorithms for unconstrained optimization." *Computational Optimization and Applications* 16 (2000): 295-308. https://doi.org/10.1023/A:1008772414083

[23] Taheri, Sona, Musa Mammadov, and Sattar Seifollahi. "Globally convergent algorithms for solving unconstrained optimization problems." *Optimization* 64, no. 2 (2015): 249-263. https://doi.org/10.1080/02331934.2012.745529

[24] Young, David M. *Iterative solution of large linear systems*. Elsevier, 2014.

[25] Akhir, Mohd Kamalrulzaman Md, Mohamed Othman, Jumat Sulaiman, Zanariah Abdul Majid, and Mohamed Suleiman. "The four point-EDGMSOR iterative method for solution of 2D Helmholtz equations." In *Informatics Engineering and Information Science: International Conference, ICIEIS 2011, Kuala Lumpur, Malaysia, November 14-16, 2011, Proceedings, Part III*, pp. 218-227. Springer Berlin Heidelberg, 2011. https://doi.org/10.1007/978-3-642-25462-8_19

[26] Akhir, M. K. M., M. Othman, J. Sulaiman, Z. A. Majid, and M. Suleiman. "Half-sweep modified successive overrelaxation for solving two-dimensional helmholtz equations." *Australian Journal of Basic and Applied Sciences* 5, no. 12 (2011): 3033-3039.

[27] Sulaiman, Jumat, M. K. Hasan, M. Othman, And S. A. Abdul Karim. "Numerical Solutions Of Nonlinear Second-Order Two-Point Boundary Value Problems Using Half-Sweep Sor With Newton Method." *Journal of Concrete & Applicable Mathematics* 11, no. 1 (2013).

[28] Sulaiman, Jumat, Mohd Khatim Hasan, Mohamed Othman, and Samsul Ariffin Abdul Karim. "Fourth-order solutions of nonlinear two-point boundary value problems by Newton-HSSOR iteration." In *AIP Conference Proceedings*, vol. 1602, no. 1, pp. 69-75. American Institute of Physics, 2014. https://doi.org/10.1063/1.4882468

[29] Ghazali, K., J. Sulaiman, Y. Dasril, and D. Gabda. "Newton-SOR Iteration for Solving Large-Scale Unconstrained Optimization Problems with an Arrowhead Hessian Matrices." In *Journal of Physics: Conference Series*, vol. 1358, no. 1, p. 012054. IOP Publishing, 2019. https://doi.org/10.1088/1742-6596/1358/1/012054

[30] Ghazali, Khadizah, Jumat Sulaiman, Yosza Dasril, and Darmesah Gabda. "Application of newton-4EGSOR iteration for solving large scale unconstrained optimization problems with a tridiagonal hessian matrix." In *Computational Science and Technology: 5th ICCST 2018, Kota Kinabalu, Malaysia, 29-30 August 2018*, pp. 401-411. Springer Singapore, 2019. https://doi.org/10.1007/978-981-13-2622-6_39

[31] Ghazali, Khadizah, Jumat Sulaiman, Yosza Dasril, and Darmesah Gabda. "Newton-2EGSOR Method for Unconstrained Optimization Problems with a Block Diagonal Hessian." *SN Computer Science* 1 (2020): 1-11. https://doi.org/10.1007/s42979-019-0021-0

[32] Killingbeck, John P., and Alain Grosjean. "A Gauss elimination method for resonances." *Journal of mathematical chemistry* 47 (2010): 1027-1037. https://doi.org/10.1007/s10910-009-9622-5

[33] Liu, Yunfei, Jun Lv, and Xiaowei Gao. "The application of simultaneous elimination and back-substitution method (SEBSM) in finite element method." *Engineering Computations* 33, no. 8 (2016): 2339-2355. https://doi.org/10.1108/EC-10-2015-0287

[34] Kavitha, Krishnan, and Velusamy Vijayakumar. "Optimal control for Hilfer fractional neutral integrodifferential evolution equations with infinite delay." *Optimal Control Applications and Methods* 44, no. 1 (2023): 130-147. https://doi.org/10.1002/oca.2938

[35] Patel, Rohit, V. Vijayakumar, Juan J. Nieto, Shimpi Singh Jadon, and Anurag Shukla. "A note on the existence and optimal control for mixed Volterra–Fredholm-type integrodifferential dispersion system of third order." *Asian Journal of Control* 25, no. 3 (2023): 2113-2121. https://doi.org/10.1002/asjc.2860

[36] Patel, Rohit, Anurag Shukla, Juan J. Nieto, Velusamy Vijayakumar, and Shimpi Singh Jadon. "New discussion concerning to optimal control for semilinear population dynamics system in Hilbert spaces." *Nonlinear Analysis: Modelling and Control* 27, no. 3 (2022): 496-512. https://doi.org/10.15388/namc.2022.27.26407

[37] Mohan Raja, Marimuthu, and Velusamy Vijayakumar. "Optimal control results for Sobolev-type fractional mixed Volterra–Fredholm type integrodifferential equations of order 1< r< 2 with sectorial operators." *Optimal Control Applications and Methods* 43, no. 5 (2022): 1314-1327. https://doi.org/10.1002/oca.2892

[38] Hamrelaine, Salim, Fateh Mebarek-Oudina, and Mohamed Rafik Sari. "Analysis of MHD Jeffery Hamel flow with suction/injection by homotopy analysis method." *Journal of Advanced Research in Fluid Mechanics and Thermal Sciences* 58, no. 2 (2019): 173-186.

[39] Yagoub, Sami Abdelrahman Musa, Gregorius Eldwin Pradipta, and Ebrahim Mohammed Yahya. "Prediction of bubble point pressure for Sudan crude oil using Artificial Neural Network (ANN) technique." *Progress in Energy and Environment* (2021): 31-39.

[40] Ching, Ng Khai. "A 3D Mesh-Less Algorithm for Simulating Complex Fluid Structure Interaction (FSI) Problem involving Free Surface." *Journal of Advanced Research in Numerical Heat Transfer* 11: 23-28.