Pertanika J. Sci. & Technol. 30 (3): 1841 - 1859 (2022)

PERTANIKA JOURNALS

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Spectral Gradient Method with Log-determinant Norm for Solving Non-Linear System of Equations

Yeong Lin Koay[1], Hong Seng Sim[1]*, Yong Kheng Goh[1] and Sing Yee Chua[2]

[1]*Department of Mathematical and Actuarial Sciences, Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman, 43000 Selangor, Malaysia*
[2]*Department of Electrical and Electronic Engineering, Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman, 43000 Selangor, Malaysia*

ABSTRACT

Solving a system of non-linear equations has always been a complex issue whereby various methods were carried out. However, most of the methods used are optimization-based methods. This paper has modified the spectral gradient method with the backtracking line search technique to solve the non-linear systems. The efficiency of the modified spectral gradient method is tested by comparing the number of iterations, the number of function calls, and computational time with some existing methods. As a result, the proposed method shows better performance and gives more stable results than some existing methods. Moreover, it can be useful in solving some non-linear application problems. Therefore, the proposed method can be considered an alternative for solving non-linear systems.

Keywords: Jacobian, log-determinant norm, nonlinear systems, optimization, spectral gradient method

INTRODUCTION

Solving a system of non-linear equations has always been a complex issue whereby various methods were carried out. The scenario becomes challenging if the system does not show good linear or polynomial characteristics. Non-linear equation systems exist in various fields, such as chemistry, engineering, and medicine.

Let $F : R^n \to R^n$ be a continuously differentiable function, and then the non-linear equation systems can be expressed as Equation 1:

$$F(x) = 0, \ x \in R^n \qquad (1)$$

ARTICLE INFO

Article history:
Received: 30 October 2021
Accepted: 27 January 2022
Published: 20 April 2022

DOI: https://doi.org/10.47836/pjst.30.3.05

E-mail addresses:
koayyeonglin@gmail.com (Yeong Lin Koay)
simhs@utar.edu.my (Hong Seng Sim)
gohyk@utar.edu.my (Yong Kheng Goh)
sychua@utar.edu.my (Sing Yee Chua)
* Corresponding author

ISSN: 0128-7680
e-ISSN: 2231-8526

In solving a system of non-linear Equation 1, some iterative methods exist. Most of the methods used are optimization-based. There is a close relationship between solving a series of non-linear equations and finding a local minimum. A local minimum of an objective function corresponds to the point where derivatives of the objective function are zero. If one considers a system of non-linear equations as the derivatives of a particular objective function, then seeking a solution to the non-linear system is equivalent to minimizing the objective function. Such equations satisfied at the current point are considered constraints at each stage, whereas others are considered objective functions.

This paper proposes modifying the spectral gradient method in solving the non-linear systems under a modified backtracking line search strategy. The paper is organized as follows: the section materials and methods introduce standard optimization methods that compare with the proposed method and the modified spectral gradient method to solve non-linear systems. Then, the section "Result and Discussion" shows the numerical results of the test problems and some real-life applications. Finally, conclusions will be presented in the last section of the paper.

MATERIALS AND METHODS

Standard Optimization Methods

The standard way of solving a non-linear system of Equation 1 is to assume an initial approximation x_0 and then perform an iterative formula in the form of Equation 2

$$x_{k+1} = x_k + \mu_k d_k, \text{ for } k \geq 0 \quad (2)$$

where x_k is the current solution approximation and x_{k+1} is the next approximation of the solution for the non-linear system. The vector d_k represents the search direction, and the scalar μ_k defines the step length. At each step, the results of the current iteration are used as the initial point for the next iteration. To generate x_{k+1} closer to the solution, we will need to choose an appropriate form of d_k and μ_k.

The steepest descent (SD) method was first raised by Cauchy (1847). It is one of the simplest and most well-known methods for minimizing non-linear functions. SD method updates the current point μ_k in the opposite direction of the gradient, g_k of the function (Equation 3).

$$d_k = -g_k \quad (3)$$

Apart from well-conditioned problems, the traditional SD approach performs poorly. Raydan and Svaiter (2002) noted that the bad behavior of the SD approach is not related to the choice of search direction. Instead, poor behavior is related to the optimal selection of step length by Cauchy. Despite the small storage capacity and very low computational

expense per execution, the SD approach has been known as extremely poor and inefficient due to the slow convergence speed and oscillatory behavior. Therefore, the SD method is not often used in practice.

The convergence of the Cauchy traditional SD method has been deeply studied. It has been found that it is related to the spectral properties of the Hessian matrix. De Asmundis et al. (2013) recommended a way to improve the SD method. The purpose of the modification is to force the gradients into a one-dimensional subspace as the iterations progress. It may avoid the key reason for the SD method's slow convergence, which is the classical zigzag pattern.

Hestenes and Stiefel (1952) published the first paper on the conjugate gradient (CG) method for solving linear systems. Currently, the CG method is a commonly used method to solve non-linear problems of large-scale systems. It performs the update by combining the previous and new directions to approximate the optimal solutions. The search direction d_k and the scalar β_{k-1} are defined as Equations 4 and 5

$$d_k = -g_k + \beta_{k-1} d_{k-1}, \tag{4}$$

where

$$\beta_{k-1} = \frac{g_k^{\,T} g_k}{g_{k-1}^{\,T} g_{k-1}}. \tag{5}$$

CG method has a small space requirement and good properties for global convergence. CG method is characterized to carry out a learning approach that falls between SD and Newton's method (Marini, 2009). The method aims to speed up the convergence rate of the classic SD method while minimizing the computational load associated with Newton's method's processes, such as storage requirement and computation of the inverse Hessian.

One of the methods that can be used to solve non-linear equations is Newton's method, also known as the Newton-Raphson method. The method was first published by Wallis (1095). Simpson (1740) defined Newton's method as an iterative approach used to solve general non-linear equations using calculus. In addition, Simpson claimed that Newton's method could also be used to solve optimization problems by setting the slope to zero.

However, the weakness of Newton's approach is that it often fails to converge and might be stuck in a repeating cycle. Even the convergence requirements of this approach are well understood, but still, this approach depends on the assumption that the initial solution is reasonably good. Thus, this method is not considered a successful practical procedure. Broyden (1965) also noted another disadvantage: the difficulty of measuring the Jacobian matrix. Even though the functions are extremely straightforward to obtain their partial derivatives, the efforts needed to determine the matrix may be excessive. The execution of Newton's method will become expensive.

Therefore, quasi-Newton approaches have developed. Martinez (2000) has mentioned that the quasi-Newton method is a static Newton method and a discrete Newton method. If the Jacobian matrix is large for the discrete Newton method, it is not comparable with the inexpensive linear algebra models. Nevertheless, discrete Newton algorithms are successful in many large sparse problems. Here a large sparse problem means a problem involving a high dimensional sparse matrix. In such situations, the limited difference method allows us to use a small number of functional calculations to measure the estimated Jacobian. The matrix form is not expensive to be factorized.

Quasi-Newton approaches are used for solving unconstrained optimization problems. Some quasi-Newton approaches are popular because many linear algebra iterations are avoided. Martinez (2000) stated that before 1990, there had been many published articles on numerical analysis research of the quasi-Newton method for solving non-linear systems. However, the study might be out of practice after the method involved in the usual practice of problem solvers in other fields, such as engineering and manufacturing. While the users know these benefits and weaknesses, quasi-Newton methods can be used to solve large-scale non-linear problems. There are some common quasi-Newton algorithms such as symmetric rank-one, Davidon-Fletcher-Powell (DFP), Broyden-Fletcher-Goldfarb-Shanno (BFGS), and Berndt-Hall-Hall-Hausman.

BFGS method is part of the quasi-Newton methods. BFGS method can be used to solve unconstrained non-linear optimization problems. Instead of directly computing the exact Hessian matrix, the BFGS method approximates the Hessian matrix by using a full rank matrix. It is an efficient method to deal with small or medium-scale problems. However, the BFGS method requires many iterations and function calls in solving large-scale problems. When the Hessian matrix is ill-conditioned, an inappropriate initial approximation of the Hessian matrix or a poorly defined search direction will result in the inefficiency of the method (Cheng & Li, 2010).

The approximation of Hessian B_{k+1} is defined by Equation 6

$$B_{k+1}=\begin{cases} B_k-\dfrac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}+\dfrac{y_k y_k^T}{y_k^T s_k}, \text{ if } s_k{}^T y_k>0 \\ B_k, \text{ otherwise} \end{cases} \tag{6}$$

where s_k and y_k are defined as Equations 7 and 8

$$s_k = x_{k+1} - x_k \tag{7}$$

$$y_k = F_{k+1} - F_k \tag{8}$$

and F is the non-linear equation system.

The presence of Jacobian matrix computation during the selection of the step length might increase the difficulty. Hence, the modified BFGS approach has been proposed with

the backtracking line search techniques that avoid computing the Jacobian matrix. Yuan and Lu (2008) introduced a new backtracking inexact BFGS method for solving symmetric non-linear equations. The modified BFGS method has a descent property norm, where under appropriate circumstances, the global and superlinear convergence will be guaranteed. The authors have shown that the modified BFGS method with the new backtracking line search is more efficient than the Jacobian matrix computation technique.

Spectral gradient (SG) methods for minimization originated from Barzilai-Borwein. Barzilai and Borwein (1988) proposed a two-point step size gradient method. Raydan (1993) has developed the convergence for quadratics of the spectral gradient method. One unique way of dealing with large-scale problems is the spectral gradient method. This method is a nonmonotone step length associated with the gradient approach to overcome the Cauchy method's weaknesses. Different techniques have been proposed since there are many variations when choosing the effective step length along the negative gradient direction (Biglari & Solimanpur, 2013). This method is obtained by approximation of the secant equation for the SD method. At each iteration, a descent in the objective function is not guaranteed in the SG method; however, it outperforms the traditional SD method in practice (Raydan, 1997). This method delivers better efficiency and low-cost computations than the traditional SD method since it needs a small number of storage locations. By combining the classical SG method with better nonmonotone line search strategies, the method's effectiveness can be greatly increased (Xiao et al., 2010).

If the Hessian matrix of the objective function is ill-conditioned, it will lead to the inefficiency of the gradient methods. The gradient methods have a fixed condition in selecting the step length to reduce the function value. Therefore, it will cause the slow convergence of a stable complex system. Dealing with the problem of inefficiency, Sim et al. (2019) have modified the SG method. This method is proposed to improve the slow convergence issues. It operates separately on the gradient vector norm and the objective function simultaneously. Furthermore, this method is combined with some line search strategies. The line search reduces the function evaluation, whereas an individual adaptive parameter damps the gradient vector. The proposed method is developed under the backtracking and nonmonotone line search. Finally, the comparison is made between the proposed method and some well-known CG-based methods since the CG methods have extremely good convergence properties. Sim et al. (2019) proved that the proposed spectral gradient method is a comparative alternative for solving large-scale problems.

The modification of SG methods and their applications have been studied in recent years (Abubakar et al., 2020; Antonelli et al., 2016; Ibrahim et al., 2020). For example, Raydan (1997) combined the Barzilai and Borwein technique with a nonmonotone line search strategy that ensures global convergence. The results show that using the global Barzilai and Borwein technique might reduce the gradient evaluations and the number of line searches. Besides, the SG method can be accelerated by using an alternating strategy

that cycles between the SD and SG steps (Xiao et al., 2010). The Barzilai–Borwein (CBB) approach (Dai et al., 2006) contributed significantly to this scheme.

Cruz and Raydan (2003) developed a method for solving non-linear systems of equations using the spectral method. The authors present an approach for ensuring global convergence based on nonmonotone line search techniques and details of the implementation for handling large-scale problems. Zhang and Zhou (2006) have proposed an approach for solving non-linear monotone equations. A modified spectral gradient approach and a projection method (Solodov & Svaiter, 1998) are combined in this method. If the non-linear equations to be solved are monotone and Lipschitz continuous, it has been proven to be globally convergent to a solution of the system. This method is also able to solve non-smooth equations.

In recent years, there has been a significant increase in the application of optimization techniques. Due to the advantages and disadvantages of different classical optimization methods, many modifications have been made. Modifying those methods aims to improve the overall performance, such as efficiency, computational time, and convergence rate. In a nutshell, various optimization methods are proposed to solve the non-linear system, and the SG method has been modified to solve non-linear systems in this paper.

Line Search Strategy

The backtracking Armijo algorithm (BTA) is a line search strategy to select the best step length. The BTA algorithm begins with a large approximation of the step length. Then, depending on the local gradient of the objective function, it will gradually reduce the step length, known as "backtracking," until a satisfactory reduction is detected in the objective function. The algorithm for modified backtracking line search strategy with Armijo condition is given as Equation 9:

Step 0: Given constants $\delta \in (0,1)$.

Step 1: Set $k = 0$ and $\mu = 1$.

Step 2: Test the relation

$$||F(x_k + \mu_k d_k)||_2^2 \leq ||F(x_k)||_2^2 + \delta \mu_k^2 F_k^T d_k \qquad (9)$$

where $d_k = -B_k^{-1} g_k$.

Step 3: If Equation 9 does not satisfy, choose a new $\mu \to \frac{1}{2}\mu$ and set $k = k + 1$, then go to Step 2. Otherwise, set $\mu_k = \mu$ and $x_{k+1} = x_k + \mu_k d_k$.

Spectral Gradient Method with Log-Determinant Norm

In order to derive an updated scheme for B_k, a restriction for components of B_k under some measures is imposed by minimizing the log-determinant norm and allowing them to satisfy

the secant equation. Hence, for any positive matrix B, the solution is given by the updated B_{k+1} (Equations 10 & 11):

$$\min tr\left(B_{k+1}\right)-\ln\left(\det\left(B_{k+1}\right)\right) \tag{10}$$

$$\text{s.t. } s_k^T B_{k+1} s_k = s_k^T y_k \tag{11}$$

where tr is the trace of a square matrix, defined as the sum of elements on the main diagonal of the matrix;det is the determinant of a matrix. Note that B_k is a symmetric matrix; for simplicity, this paper only consider the case B_k is diagonal.

Let $B_{k+1}=diag(B_{k+1}^{(1)},\ldots,B_{k+1}^{(n)})$ and $s_k^T=(s_k^{(1)},\ldots,s_k^{(n)})$, the minimization Equations 10 and 11 become Equations 12 and 13:

$$\min\left(\sum_{i=1}^{n}B_{k+1}^{(i)}\right)-\ln\left(\prod_{i=1}^{n}B_{k+1}^{(i)}\right) \tag{12}$$

$$\text{s.t. } (\sum_{i=1}^{n}(s_k^{(i)})^2 B_{k+1}^{(i)})-\sum_{i=1}^{n}s_k^{(i)}y_k^{(i)}=0 \tag{13}$$

By applying the Lagrange method to the minimization, Equations 12 and 13 become Equation 14:

$$L\left(\omega\right)=\left(\sum_{i=1}^{n}B_{k+1}^{(i)}\right)-\ln\left(\prod_{i=1}^{n}B_{k+1}^{(i)}\right)+\omega\left[\sum_{i=1}^{n}(s_k^{(i)})^2\left(B_{k+1}^{(i)}\right)-s_k^T y_k\right] \tag{14}$$

where ω is the Lagrange multiplier.

Differentiate Equation 14 with respect to $B_{k+1}^{(i)}$ and set the derivatives to zero (Equation 15):

$$\frac{\partial L}{\partial B_{k+1}^{(i)}}=1-\frac{1}{B_{k+1}^{(i)}}+\omega(s_k^{(i)})^2=0, i=1,2,...,n \tag{15}$$

then gives Equation 16

$$B_{k+1}{}^{(i)}=\frac{1}{1+\omega\left(s_k{}^{(i)}\right)^2}, i=1,2,\ldots,n \tag{16}$$

By substituting Equation 16 into constraint Equation 13 and rewriting the left-hand side of Equation 13 as a function of ω (Equation 17):

$$F(\omega)=\sum_{i=1}^{n}\frac{(s_k^{(i)})^2}{1+\omega(s_k^{(i)})^2}-s_k^T y_k \tag{17}$$

where the Lagrange multiplier ω can be obtained by solving the non-linear equation $F(\omega)=0$. It can be approximated by applying only one iteration of Newton-Raphson,

with $\omega = 0$. When $s_k^T s_k > s_k^T y_k$, Equation 17 has a unique positive solution and hence, the Lagrange multiplier ω_k can be approximated by using Equation 18:

$$\omega_k \approx \omega - \frac{F(\omega)}{F'(\omega)} = \frac{s_k{}^T s_k - s_k{}^T y_k}{\sum_{i=1}^{n} (s_k{}^{(i)})^4} \tag{18}$$

Lastly, the updating formula for B_{k+1} is given as Equation 19

$$B_{k+1} = \begin{cases} \text{diag}(B_{k+1}^{(1)},...,B_{k+1}^{(n)}), \text{ if } s_k^T s_k > s_k^T y_k \\ \dfrac{s_k{}^T y_k}{s_k{}^T s_k} I, \text{ otherwise} \end{cases} \tag{19}$$

Where $\frac{s_k^T y_k}{s_k^T s_k}$ exactly the Oren-Luenberger scaling (Luenberger and Ye, 1984), B_{k+1} as defined in Equation 19 and ω is defined in Equation 18.

The algorithm for modified spectral gradient method is shown below:

Step 0: Set $k = 0$. Given initial guessing point x_0, tolerance $\in (0,1)$ and B_0.

Step 1: If $||F_k||_2 = 0$, then stop.

Step 2: Calculate $y_k = F_{k+1} - F_k$, $y_k = F_{k+1} - F_k$ and ω based on Equation 18.

Step 3: Calculate B_{k+1} from Equation 19.

Step 4: Obtain μ_k through the modified BTA algorithm.

Step 5: Compute $x_{k+1} = x_k + \mu_k d_k$, for $k \geq 0$, where $d_k = -B_k^{-1} F_k$.

Step 6: Set $k = k + 1$ and go to Step 1.

Convergence Analysis

This section briefly discusses the convergence of the modified spectral gradient method. The detailed proofs can be referred to Sim et al. (2019).

Assumption 1.

a. The objective function f is twice continuously differentiable.

b. The level set $D = \left\{ x \in R^n : f(x) \leq f(x_0) \right\}$ is convex.

c. There exist positive constants M_2 and M_2 such that

$$M_1 ||z||_2^2 \leq z^T G(x) z \leq M_2 ||z||_2^2,$$

for $\forall z \in D$ and $\forall z \in D$. It implies that the objective function f has a unique minimize x^* in x_0 .

Lemma 1. Let x_0 be a starting point and $B_0 = I$, where I is the $n \times n$ identity matrix.

Then for c_1 defined by Equation 19, the sequence $\{||B_k||_2\}$ is bounded by some positive constants c_1 and c_2, i.e. $0 < c_1 < B_0 < B_1 < ... < B_k < ... < c_2$.

The proof of Lemma 1 is based on the assumptions provided by Byrd and Nocedal (1989). The next lemma is a direct result of Lemma 1.

Lemma 2. Suppose that the assumptions in Lemma 1 hold, then there exist positive constants c_4 and c_4 such that

$$d_k^T g_k \leq -c_3||F_k||_2^2 \text{ and } ||d_k||_2^2 \leq -c_4||F_k||_2^2,$$

where $d_k = -B_k^{-1}F_k$ where c_1 is defined by Equation 19.

The convergence of the spectral gradient method using the BTA line search algorithm is presented in the next theorem.

Theorem 1. Under assumption by Bryd and Nocedal (1989), there exist positive constants c_1 and c_2 such that, for any x_k and any d_k with $F_k^T d_k < 0$, the step length μ_k produced by the BTA Algorithm will satisfy either

$$||F(x_k + \mu_k d_k)||_2^2 - ||F(x_k)||_2^2 \leq -c_1 \frac{\left(F_k^T d_k\right)^2}{||\, d_k||_2^2}$$

or

$$||F(x_k + \mu_k d_k)||_2^2 - ||F(x_k)||_2^2 \leq -c_2 F_k^T d_k$$

Furthermore, if d_k satisfies the following conditions:

$$F_k^T d_k \leq c_3||F(x_k)||_2^2,$$

$$||d_k||_2 \leq -c_4||F_k||_2$$

for some positive constants c_3 and c_4, then

$$\lim_{k\to\infty} ||F_k||_2 = 0.$$

In the next section, numerical experiments are conducted to compare the efficiency of the proposed method.

RESULTS AND DISCUSSIONS

Numerical Experiments and Discussion

This section applies the modified SG method to solve some non-linear systems. The comparison is made between the modified SG method and three other methods under the modified BTA line search strategy. The following methods are taken into consideration:

1. Modified Spectral Gradient Method (Modified SG)

2. Broyden-Fletcher-Goldfarb-Shanno Method (BFGS)
3. Steepest Descent Method (SD)
4. Conjugate Gradient Method (CG)

The step lengths μ_k are generated by the BTA algorithm with the parameter $\delta = 0.1$. The step length $\mu = 1$ is used as the initial step length and reduced if the Armijo condition does not satisfy. The minimum value for step length is set as 2^{-7}. Since the modified SG method and BFGS method require the computation of matrix B_k, the matrix B_0 is initialized to an identity matrix with dimension *n*.

There are two termination criteria for these methods: the norm of the non-linear functions and the number of iterations. The first termination criterion is $||F_k||_2 \leq 10^{-4}$ and the maximum number of iterations is set to be 10^4. If the number of iterations exceeds 10^4, the tested problem will be considered "fail to converge." The codes are written in Python 3.7.9.

A total of 31 problems given by Fang et al. (2018) and Andrei (2008) have been used to test the performance of these methods. The dimensions of the tested problems are set as *n* = 10, 100, 200 and 500, if the dimensions are not provided in the tested problems.

Using Dolan and Moré's (2002) performance profile, the performance of the modified SG, BFGS, SD, and CG methods can be evaluated clearly. The performance of problem *p* by solver *s* is defined as Equation 20:

$$P(t \leq \tau) = \frac{1}{|\mathrm{P}|}\mathrm{size}\{\mathrm{p} \in \mathrm{P}:\mathrm{t}_{p,s} \leq \tau\} \tag{20}$$

where the function $P(t \leq \tau)$ is the cumulative distribution function for the performance ratio, P is a set of test problems, $|\mathrm{P}|$ denotes the cardinality of P and $t_{p,s}$ represents the performance ratio within a factor τ which is a real number (Equation 21)

$$t_{p,s} = \frac{m_{p,s}}{\min\{m_{p,s} : s \in S\}} \tag{21}$$

where $m_{p,s}$ represents the performance measure of interest accordingly. It is obtained for each pair (p, s) of solver *s* in a set *S* of optimization solvers and problem *p* in a set *P* of test problems.

Figures 1, 2, and 3 are the performance profiling graphs for these methods, based on the number of iterations, function calls, and computational time. Figures 1 to 3 show that the BFGS method performs the best among these methods in terms of the number of iterations, function calls, and computational time. The modified SG method indicates a better performance compared to SD and CG methods. Besides, SD and CG methods exhibit a similar pattern, which shows a poorer performance than the modified SG and BFGS methods.

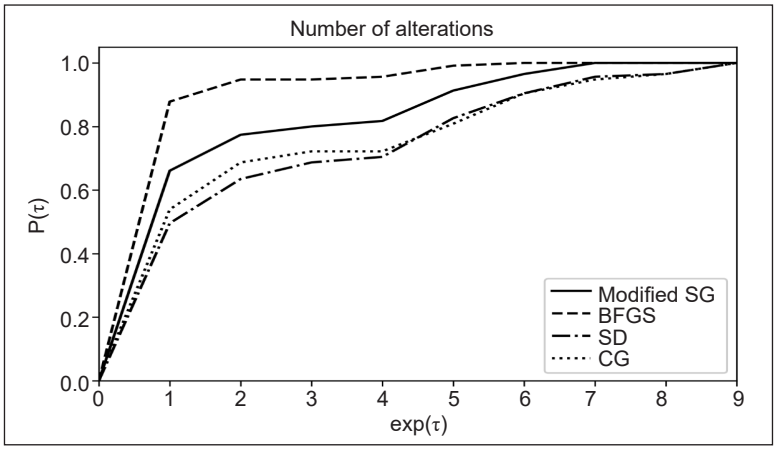

Figure 1. Comparison of methods in terms of the number of iterations

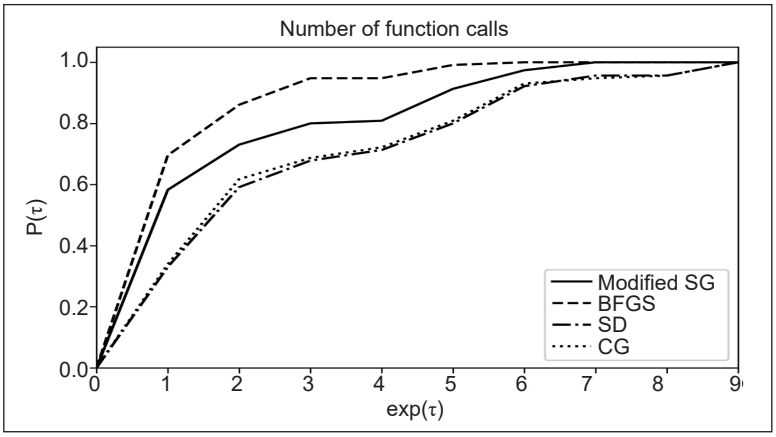

Figure 2. Comparison of methods in terms of the number of function calls

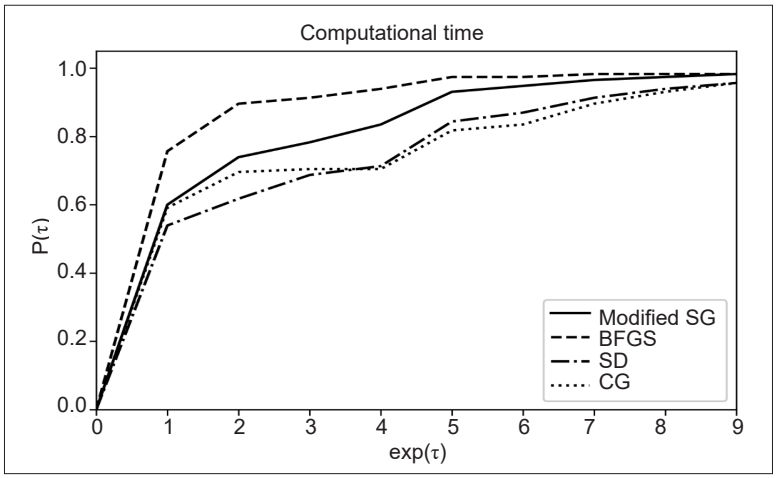

Figure 3. Comparison of methods in terms of CPU time in seconds

For the BFGS method, some of the tested problems require an extremely high number of iterations, number of function calls, and computational time compared to the modified SG method. Therefore, this study concludes that the modified SG method gives a more stable result than the BFGS method. One of the reasons the BFGS method gives better performance is that it uses the full rank matrix for B_k, while the modified SG method uses the diagonal matrix in the updating formula of B_k. Numerous researchers have suggested that the BFGS method is not appropriate for solving large-scale problems since high storage is required. Although the modified SG method is not showing the best performance among these methods, it is still considered an alternative to solve the non-linear tested problems.

Applications

Systems of non-linear equations occur in many areas of practical importance, such as engineering. In order to evaluate the performance of the modified SG method, six systems of nonlinear equations are considered. The application problems are provided by Chen et al. (2017), Grosan and Abraham (2008), Buzzi-Ferraris and Manenti (2013), and Turgut et al. (2014). These problems [(a) - (f)] are applied in both the engineering and science fields.

(a) Kinematic Application

$$\begin{cases} x_i^2 + x_{i+1}^2 - 1 = 0 \\ a_{1i}x_1x_3 + a_{2i}x_1x_4 + a_{3i}x_2x_3 + a_{4i}x_2x_4 + \\ a_{5i}x_2x_7 + a_{6i}x_5x_8 + a_{7i}x_6x_7 + a_{8i}x_6x_8 + \\ a_{9i}x_1 + a_{10i}x_2 + a_{11i}x_3 + a_{12i}x_4 + a_{13i}x_5 + \\ a_{14i}x_6 + a_{15i}x_7 + a_{16i}x_8 + a_{17i} = 0 \\ 1 \le i \le 4 \end{cases}$$

The initial guessing point used is

$$x^{(0)} = [-0.06, 0.78, -0.05, 0.38, -0.56, -0.70, 0.40, 0.09]^T$$

and the coefficients a_{ki}, $1 \le k \le 17$, $1 \le i \le 4$, are given in Table 1.

Table 1
Coefficients a_{ki} for the Kinematic Application

a_{ki}	a_{k1}	a_{k2}	a_{k3}	a_{k4}
a_{1i}	- 0.249150680	+ 0.125016350	-0.635550070	+ 1.48947730
a_{2i}	+ 1.609135400	- 0.686607360	- 0.115719920	+ 0.23062341
a_{3i}	+ 0.279423430	- 0.119228120	- 0.666404480	+ 1.32810730
a_{4i}	+ 1.434801600	- 0.719940470	+ 0.110362110	- 0.25864503

Table 1 *(continue)*

a_{ki}	a_{k1}	a_{k2}	a_{k3}	a_{k4}
a_{5i}	+ 0.000000000	- 0.432419270	+ 0.290702030	+ 1.16517200
a_{6i}	+ 0.400263840	+ 0.000000000	+ 1.258776700	- 0.26908494
a_{7i}	- 0.800527680	+ 0.000000000	- 0.629388360	+ 0.53816987
a_{8i}	+ 0.000000000	- 0.864838550	+ 0.581404060	+ 0.58258598
a_{9i}	+ 0.074052388	- 0.037157270	+ 0.195946620	- 0.20816985
a_{10i}	- 0.083050031	+ 0.035436896	- 1.228034200	+ 2.68683200
a_{11i}	- 0.386159610	+ 0.085383482	+ 0.000000000	- 0.69910317
a_{12i}	- 0.755266030	+ 0.000000000	- 0.079034221	+ 0.35744413
a_{13i}	+ 0.504201680	- 0.039251967	+ 0.026387877	+ 1.24991170
a_{14i}	- 1.091628700	+ 0.000000000	- 0.057131430	+ 1.46773600
a_{15i}	+ 0.000000000	- 0.432419270	- 1.162808100	+ 1.16517200
a_{16i}	+ 0.049207290	+ 0.000000000	+ 1.258776700	+ 1.07633970
a_{17i}	+ 0.049207290	+ 0.013873010	+ 2.162575000	- 0.69686809

(b) Interval Arithmetic Benchmark Application

$$\begin{cases} x_1 - 0.25428722 - 0.18324757x_4x_3x_9 = 0 \\ x_2 - 0.37842197 - 0.16275449x_1x_{10}x_6 = 0 \\ x_3 - 0.27162577 - 0.16955071x_1x_2x_{10} = 0 \\ x_4 - 0.19807914 - 0.15585316x_7x_1x_6 = 0 \\ x_5 - 0.44166728 - 0.19950920x_7x_6x_3 = 0 \\ x_6 - 0.14654113 - 0.14654113x_8x_5x_{10} = 0 \\ x_7 - 0.42937161 - 0.21180486x_2x_5x_8 = 0 \\ x_8 - 0.07056438 - 0.17081208x_1x_7x_6 = 0 \\ x_9 - 0.34504906 - 0.19612740x_{10}x_6x_8 = 0 \\ x_{10} - 0.42651102 - 0.21466544x_4x_8x_1 = 0 \end{cases}$$

The initial guessing point used is $x^{(0)} = [1,1,1,1,1,1,1,1,1,1]^T$.

(c) Chemical Equilibrium

$$\begin{cases} x_1 + x_4 - 3 = 0 \\ 2x_1 + x_2 + x_4 + x_7 + x_8 + x_9 + 2x_{10} - R = 0 \\ 2x_2 + 2x_5 + x_6 + x_7 - 8 = 0 \\ 2x_3 + x_9 - 4R = 0 \\ x_1 x_5 - 0.193 x_2 x_4 = 0 \\ x_6 \sqrt{x_1} - 0.002597 \sqrt{x_2 x_4 \cdot \sum_{i=1}^{10} x_i} = 0 \\ x_7 \sqrt{x_4} - 0.003448 \sqrt{x_1 x_2 \cdot \sum_{i=1}^{10} x_i} = 0 \\ x_4 x_8 - 1.799 \times 10^{-5} x_1 \cdot \sum_{i=1}^{10} x_i = 0 \\ x_4 x_9 - 2.155 \times 10^{-4} x_1 \sqrt{x_3 \cdot \sum_{i=1}^{10} x_i} = 0 \\ x_{10} {x_4}^2 - 3.846 \times 10^{-5} {x_4}^2 \cdot \sum_{i=1}^{10} x_i = 0 \end{cases}$$

where R is 4.056734.
The initial guessing point used is

$$x^{(0)} = [0.15884, 0.89358, 8.11340, 2.84116, 3.08473,$$
$$0.04039, 0.00300, 0.00002, 0.00013, 0.00058]^T$$

(d) Neurophysiology application

$$\begin{cases} f_1 = x_1^2 + x_3^2 - 1 = 0 \\ f_2 = x_2^2 + x_4^2 - 1 = 0 \\ f_3 = x_5 x_3^3 + x_6 x_4^3 = 0 \\ f_4 = x_5 x_1^3 + x_6 x_2^3 = 0 \\ f_5 = x_5 x_1 x_3^2 + x_6 x_4^2 x_2 = 0 \\ f_6 = x_5 {x_1}^2 x_3 + x_6 x_2^2 x_4 = 0 \end{cases}$$

The initial guessing point used is $x^{(0)} = [0.446, -0.446, 0.895, 0.367, 0.367]^T$.

(e) Combustion application

$$\begin{cases} f_1 = x_2 + 2x_6 + x_9 + 2x_{10} - 10^{-5} = 0 \\ f_2 = x_3 + x_8 - 3 \cdot 10^{-5} = 0 \\ f_3 = x_1 + x_3 + 2x_5 + 2x_8 + x_9 + x_{10} - 5 \cdot 10^{-5} = 0 \\ f_4 = x_4 + 2x_7 - 10^{-5} = 0 \\ f_5 = 0.5140437 \cdot 10^{-7} x_5 - x_1^2 = 0 \\ f_6 = 0.100632 \cdot 10^{-6} x_6 - 2x_2^2 = 0 \\ f_7 = 0.7816278 \cdot 10^{-15} x_7 - x_4^2 = 0 \\ f_8 = 0.1496236 \cdot 10^{-6} x_8 - x_1 x_3 = 0 \\ f_9 = 0.6194411 \cdot 10^{-7} x_9 - x_1 x_2 = 0 \\ f_{10} = 0.2089296 \cdot 10^{-14} x_{10} - x_1 {x_2}^2 = 0 \end{cases}$$

The initial guessing point used is

$$x^{(0)} = [-5.9286 \cdot 10^{-8}, -6.9428 \cdot 10^{-5}, -0.2980, -8.8526 \cdot 10^{-5}, -0.4127,$$

$$-0.0547, 4.9253 \cdot 10^{-5}, 0.2981, 0.9453, -0.4179]^T.$$

(f) Experimental Test

$$\begin{cases} f_1(x_1, x_2) = \cos(2x_1) - \cos(2x_2) - 0.4 = 0 \\ f_2(x_1, x_2) = 2(x_2 - x_1) + \sin(2x_2) - \sin(2x_1) - 1.2 = 0 \end{cases}$$

The initial guessing point used is $x^{(0)} = [0.15, 0.49]^T$.

The results of the application problems are listed in Table 2. According to Table 2, it can be concluded that the modified SG method serves as another option to solve different systems of non-linear equations in real-life applications. Although the modified SG method gives a higher number of iterations, function calls, and computation time than the SD method in problems a and f, it performs better than CG and BFGS methods since they fail to converge in these two problems. Generally, the modified SG method indicates comparable results with the other three existing methods. In certain situations, the modified SG method exhibits more robust convergence properties than CG and BFGS.

CONCLUSION

This paper proposes a modified spectral gradient method for solving non-linear systems. The modification is performed since the actual Hessian is unavailable or requires large

Table 2
Numerical results for application problems

Number of Iteration					
Problem	Dim	Modified SG	BFGS	SD	CG
a	8	169	-	152	-
b	10	4	4	4	3
c	10	2	2	2	2
d	6	2	2	2	2
e	10	2	2	2	2
f	2	8171	-	881	-
Number of Function Call					
Problem	Dim	Modified SG	BFGS	SD	CG
a	8	3450	-	2644	-
b	10	18	26	18	14
c	10	19	41	25	37
d	6	52	56	52	52
e	10	10	14	10	13
f	2	193489	-	21241	-
Computational Time					
Problem	Dim	Modified SG	BFGS	SD	CG
a	8	1.8066	-	1.2886	-
b	10	0.0100	0.0139	0.0096	0.0070
c	10	0.0091	0.0107	0.0087	0.0069
d	6	0.0070	0.0091	0.0082	0.0072
e	10	0.0060	0.0110	0.0060	0.0060
f	2	16.3163	-	1.5309	-

Note: **The symbol '-' represents that the method failed to converge.**

storage when handling large-scale problems. The proposed method uses a diagonal matrix to approximate the actual Hessian matrix instead of the full rank matrix.

It uses the eigenvalues of the actual Hessian matrix as the diagonal entries to approximate the inverse of Hessian. The proposed method is derived based on the log-determinant norm, where the Lagrange multiplier is approximated by using only one step of the Newton-Raphson method. The standard line search strategy with the Armijo condition is modified to solve the non-linear systems. The proposed method is compared with the common existing methods regarding the number of iterations, function calls, and computational time. The numerical results show that the proposed method can be an alternative in solving systems of non-linear equations in research-tested problems and real-life applications, therefore, justifying the contribution of the modified spectral gradient method.

ACKNOWLEDGEMENTS

The Ministry of Higher Education (MoHE) Malaysia supported this research through the Fundamental Research Grant Scheme (FRGS/1/2019/STG06/UTAR/02/4). In addition, the research was also supported by Universiti Tunku Abdul Rahman (UTAR) through Universiti Tunku Abdul Rahman Research Fund (UTARRF) project IPSR/RMC/UTARRF/2020-C2/S05.

REFERENCES

Abubakar, A. B., Kumam, P., & Mohammad, H. (2020). A note on the spectral gradient projection method for nonlinear monotone equations with applications. *Computational and Applied Mathematics*, *39*(2), 1-35. https://doi.org/10.1007/s40314-020-01151-5

Andrei, N. (2008). An unconstrained optimization test functions collection. *Advanced Modeling and Optimization, 10*(1), 147-161.

Antonelli, L., De Simone, V., & Di Serafino, D. (2016). On the application of the spectral projected gradient method in image segmentation. *Journal of Mathematical Imaging and Vision*, *54*(1), 106-116. https://doi.org/10.1007/s10851-015-0591-y

Barzilai, J., & Borwein, J. M. (1988). Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, *8*(1), 141-148. https://doi.org/10.1093/imanum/8.1.141

Biglari, F., & Solimanpur, M. (2013). Scaling on the spectral gradient method. *Journal of Optimization Theory and Applications*, *158*(2), 626-635. https://doi.org/10.1007/s10957-012-0265-5

Broyden, C. G. (1965). A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, *19*(92), 577-593.

Buzzi-Ferraris, G., & Manenti, F. (2013). *Nonlinear systems and optimization for the chemical engineer: Solving numerical problems*. John Wiley & Sons.

Byrd, R. H., & Nocedal, J. (1989). A tool for the analysis of quasi-Newton methods with application to unconstrained minimization. *SIAM Journal of Numerical Analysis, 26*, 727-739. https://doi.org/10.1137/0726042

Cauchy, A. (1847). Méthode générale pour la résolution des systemes d'équations simultanées [General method for solving systems of simultaneous equations]. *Comptes rendus de l'Académie des Sciences, 25*(1847), 536-538.

Chen, X., Liu, Y., Zhou, W., & Peng, X. (2017). Simplex-fruit fly optimization algorithm for solving systems of non-linear equations. In *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)* (pp. 615-620). IEEE Publishing. https://doi.org/10.1109/FSKD.2017.8393341

Cheng, W. Y., & Li, D. H. (2010). Spectral scaling BFGS method. *Journal of Optimization Theory and Applications*, *146*(2), 305-319. https://doi.org/10.1007/s10957-010-9652-y

Cruz, W. L., & Raydan, M. (2003). Nonmonotone spectral methods for large-scale nonlinear systems. *Optimization Methods and Software*, *18*(5), 583-599. https://doi.org/10.1080/10556780310001610493

Dai, Y. H., Hager, W. W., Schittkowski, K., & Zhang, H. (2006). The cyclic Barzilai-Borwein method for unconstrained optimization. *IMA Journal of Numerical Analysis*, *26*(3), 604-627. https://doi.org/10.1093/imanum/drl006

De Asmundis, R., di Serafino, D., Riccio, F., & Toraldo, G. (2013). On spectral properties of steepest descent methods. *IMA Journal of Numerical Analysis*, *33*(4), 1416-1435. https://doi.org/10.1093/imanum/drs056

Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming, 91*(2), 201-213. https://doi.org/10.1007/s101070100263

Fang, X., Ni, Q., & Zeng, M. (2018). A modified quasi-Newton method for nonlinear equations. *Journal of Computational and Applied Mathematics*, *328*, 44-58. https://doi.org/10.1016/j.cam.2017.06.024

Grosan, C., & Abraham, A. (2008). A new approach for solving nonlinear equations systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *38*(3), 698-714. https://doi.org/10.1109/TSMCA.2008.918599

Hestenes, M. R., & Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards, 49*(6), 409-436.

Ibrahim, S. M., Yakubu, U. A., & Mamat, M. (2020). Application of spectral conjugate gradient methods for solving unconstrained optimization problems. *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, *10*(2), 198-205. https://doi.org/10.11121/ijocta.01.2020.00859

Luenberger, D. G., & Ye, Y. (1984). *Linear and nonlinear programming* (Vol. 2). Addison-Wesley.

Marini, F. (2009). Neural networks. In R. Tauler & B. Walczak (Eds.), *Comprehensive Chemometrics: Chemical and Biochemical Data Analysis* (pp. 477-505). Elsevier. https://doi.org/10.1016/B978-044452701-1.00128-9

Martinez, J. M. (2000). Practical quasi-Newton methods for solving nonlinear systems. *Journal of Computational and Applied Mathematics*, *124*(1-2), 97-121. https://doi.org/10.1016/S0377-0427(00)00434-9

Raydan, M. (1993). On the Barzilai and Borwein choice of steplength for the gradient method. *IMA Journal of Numerical Analysis*, *13*(3), 321-326. https://doi.org/10.1093/imanum/13.3.321

Raydan, M. (1997). The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM Journal on Optimization*, *7*(1), 26-33. https://doi.org/10.1137/S1052623494266365

Raydan, M., & Svaiter, B. F. (2002). Relaxed steepest descent and Cauchy-Barzilai-Borwein method. *Computational Optimization and Applications*, *21*(2), 155-167. https://doi.org/10.1023/A:1013708715892

Sim, H. S., Leong, W. J., & Chen, C. Y. (2019). Gradient method with multiple damping for large-scale unconstrained optimization. *Optimization Letters*, *13*(3), 617-632. https://doi.org/10.1007/s11590-018-1247-9

Simpson, T. (1740). Essays on several curious and useful subjects, in speculative and mix'd mathematics. *London, 1740*, Article 81.

Solodov, M. V., & Svaiter, B. F. (1998). A globally convergent inexact Newton method for systems of monotone equations. In *Reformulation: Nonsmooth, piecewise smooth, semismooth and smoothing methods* (pp. 355-369). Springer. https://doi.org/10.1007/978-1-4757-6388-1_18

Turgut, O. E., Turgut, M. S., & Coban, M. T. (2014). Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations. *Computers & Mathematics with Applications*, *68*(4), 508-530. https://doi.org/10.1016/j.camwa.2014.06.013

Wallis, J. (1095). A treatise of algebra, both historical and practical. *Philosophical Transactions of the Royal Society of London*, *15*(173), 1095-1106. https://doi.org/10.3931/e-rara-8842

Xiao, Y., Wang, Q., & Wang, D. (2010). Notes on the Dai–Yuan–Yuan modified spectral gradient method. *Journal of Computational and Applied Mathematics*, *234*(10), 2986-2992. https://doi.org/10.1016/j.cam.2010.04.012

Yuan, G., & Lu, X. (2008). A new backtracking inexact BFGS method for symmetric nonlinear equations. *Computers & Mathematics with Applications*, *55*(1), 116-129. https://doi.org/10.1016/j.camwa.2006.12.081

Zhang, L., & Zhou, W. (2006). Spectral gradient projection method for solving nonlinear monotone equations. *Journal of Computational and Applied Mathematics*, *196*(2), 478-484. https://doi.org/10.1016/j.cam.2005.10.002