Enhanced Derivative-Free Conjugate Gradient Method for Solving Symmetric Nonlinear Equations

Jamilu Sabi’u
Department of Mathematics, Faculty of Science, Northwest University Kano, Nigeria

ABSTRACT
In this article, an enhanced conjugate gradient approach for solving symmetric nonlinear equations is propose without computing the Jacobian matrix. This approach is completely derivative and matrix free. Using classical assumptions the proposed method has global convergence with nonmonotone line search. Some reported numerical results shows the approach is promising.

KEYWORDS:
Backtracking line search
Conjugate gradient method
Hyperplane
Symmetric nonlinear equations

1. INTRODUCTION
Let us consider the systems of nonlinear equations

\[ F(x) = 0, \] (1)

where \( F : \mathbb{R}^n \rightarrow \mathbb{R}^n \) is a nonlinear mapping. Often, the mapping, \( F \) is assumed to satisfying the following assumptions:

A1. There exists an \( x^* \in \mathbb{R}^n \) s.t \( F(x^*) = 0 \)
A2. \( F \) is a continuously differentiable mapping in a neighborhood of \( x^* \)
A3. \( F'(x^*) \) is invertible
A4. The Jacobian \( F(x) \) is symmetric.

Monotonicity means

\[ (F(x) - F(y))^T (x - y) \geq 0, \quad \forall x, y \in \mathbb{R}^n \] (2)

The prominent method for finding the solution of (1), is the classical Newton’s method which generates a sequence of iterates \( \{x_k\} \) from a given initial point \( x_0 \) via

\[ x_{k+1} = x_k - (F'(x_k))^{-1}F(x_k), \] (3)

where \( k = 0, 1, 2 \ldots \). The attractive features of this method are; rapid convergence and easy to implement. Nevertheless, Newton’s method requires the computation of the Jacobian matrix, which require the first-order derivative of the systems. In practice, computations of some functions derivatives are quite costly and
sometime they are not available or could not be done precisely. In this case Newton’s method cannot be applied directly. In this work, we are interested in handling large-scale problems for which the Jacobian is either not available or requires a low amount storage, the best method is CG approach. It is vital to mention that the conjugate gradient methods are among the popular used methods for unconstrained optimization problems. They are particularly efficient for handling large-scale problems due to their convergence properties, simply to implement and low storage [10]. Not withstanding, the study of conjugate gradient methods for large-scale symmetric nonlinear systems of equations is scanty, this is what motivated us to have this paper. In general, CG methods for solving nonlinear systems of equations generates an iterative points \( x_k \) from initial given point \( x_0 \) using

\[
x_{k+1} = x_k + \alpha_k d_k,
\]

where \( \alpha_k > 0 \) ia attained via line search, and direction \( d_k \) are obtained using

\[
d_k = \begin{cases} 
-F(x_k) & \text{if } k = 0 \\
-F(x_k) + \beta_k d_k & \text{if } k \geq 1
\end{cases}
\]

\( \beta_k \) is termed as conjugate gradient parameter.

This problems under study, may arise from an unconstrained optimization problem, a saddle point problem, Karush-Kuhn-Tucker (KKT) of equality constrained optimization problem, the discretized two-point boundary value problem, the discretized elliptic boundary value problem, and etc. Equation (1) is the first-order necessary condition for the unconstrained optimization problem when \( F \) is the gradient mapping of some function \( f : \mathbb{R}^n \rightarrow \mathbb{R} \)

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

For the equality constrained problem

\[
\min f(x), \quad s.t \quad h(x) = 0,
\]

where \( h \) is a vector-valued function, the KKT conditions can be represented as the system (1) with \( x = (z, v) \), and

\[
F(z, v) = (\nabla F(z) + \nabla h(z)v, h(z)),
\]

where \( v \) is the vector of Lagrange multipliers. Notice that the Jacobian \( \nabla F(z, v) \) is symmetric for all \((z, v)\) (see, e.g., [7]). Problem (1) can be converted to the following global optimization problem (5) with our function \( f \) defined by

\[
f(x) = \frac{1}{2} ||F(x)||^2
\]

A large number of efficient solvers for large-scale symmetric nonlinear equations have been proposed, analyzed, and tested in the last decade. Among them are [4, 2, 9]. Still the matrix storage and solving of n-linear system are required in the BFGS type methods presented in the literature. The recent designed nonmonotone spectral gradient algorithm [1] falls within the frame work of matrix-free.

The conjugate gradient methods for symmetric nonlinear equations has received a good attention and take an appropriate progress. However, Li and Wang [5] proposed a modified Flectcher-Reeves conjugate gradient method which is based on the work of Zhang et al. [3], and the results illustrate that their proposed conjugate gradient method is promising. In line with this development, further studies on conjugate gradient are [7, 10, 8, 13]. Extensive numerical experiments showed that each over mentioned method performs quite well.

We organized the paper as follows: In the next section, we present the details of the proposed method. Convergence results are presented in Section 3. Some numerical results are reported in Section 4. Finally, conclusions are made in Section 5.

Enhanced Derivative-Free Conjugate Gradient Method For Solving Symmetric ... (Jamila Sabi’u)
2. ENHANCED DERIVATIVE-FREE (EDF)

Given an initial point $x_0$, an iterative scheme for problem (1) generally generates a sequence of iterates

$$x_k = x_{k-1} + \alpha_k d_{k-1}, \ k = 1, 2 \ldots$$

which employs a line search procedure along the direction $dk$ to compute the stepsize $\alpha_k$. Typical line searches include Backtracking, Armijo or Wolfe line searches. Let $z_k = x_k + \alpha_k d_k$, the hyperplane

$$H_k = \{ x \in \mathbb{R}^n | (x - z_k)^T F(z_k) = 0 \}$$

strictly separates $x_k$ from the solution set of (1). Therefore, from this facts Solodov and Svaiter[6] advised to let the next iterate $x_{k+1}$ be the projection of $x_k$ onto this hyperplane $H_k$. That is, $x_{k+1}$ is defined by

$$x_{k+1} = x_k - \frac{F(z_k)^T (x_k - z_k)}{||F(z_k)||^2} F(z_k)$$

In this paper, the direction $d_k$ is base on [13], specifically

$$d_k = \begin{cases} -F_k & \text{if } k = 0 \\ -\beta_k F_k + \beta_k d_k & \text{if } k \geq 1 \end{cases}$$

where $F_k$ means $F(x_k)$ and $\beta_k$ defined as

$$\beta_k = \frac{(\theta_k y_k - s_k)^T}{y_k^T d_k} F_k$$

$$\theta_k = \frac{s_k^T s_k}{s_k^T y_k}$$

Throughout this paper, $||.||$ is the Euclidean norm, $s_k = x_k - x_{k-1}$ and $y_k = F_k - F_{k-1}$

However, to compute the stepsize $\alpha_k$, nonmonotone line search proposed by [4] is an interesting idea that avoids the necessity of descent directions to guarantee that each iteration is well defined. Let $\alpha_1 > 0$, $\alpha_2 > 0$, $r \in (0, 1)$ be constants and $\{\eta_k\}$ be a given positive sequence such that

$$\sum_{k=0}^{\infty} \eta_k < \infty$$

Let $\alpha_k = \max \{1, r^k\}$ that satisfy

$$||F(x_k + \alpha_k d_k)||^2 - ||F(x_k)||^2 \leq -\omega_1 ||\alpha_k F(x_k)||^2 - \omega_2 ||\alpha_k d_k||^2 + \eta_k ||F(x_k)||^2$$

EDF Algorithm

Step 1: Given $x_0$, $\alpha > 0$, $\omega \in (0, 1)$, $r \in (0, 1)$ and a positive sequence $\eta_k$ satisfying (16), and set $k = 0$.

Step 2: Test a stopping criterion. If yes, then stop; otherwise continue with Step 3.

Step 3: Compute $d_k$ by (13).

Step 4: Compute $\alpha_k$ by the line search (17).

Step 5: Compute $x_{k+1} = x_k - \frac{F(x_k)^T (x_k - z_k)}{||F(z_k)||^2} F(z_k)$

Step 6: Consider $k = k + 1$ and go to step 2.

3. GLOBAL CONVERGENCE

This section presents global convergence results of the Enhanced derivative-free conjugate gradient method. In order to analyze the convergence of our method, we will make the following assumptions on nonlinear systems $F$. 

IJAAS Vol. 5, No. 1, March 2016 : 50 – 57
Assumption
(i) The level set $\Omega = \{x | F(x) \leq e^0 F(x_0)\}$ is bounded
(ii) In some neighborhood $N$ of $\Omega$, $F(x)$ is Lipschitz continuous i.e there exists a positive constant $L > 0$ such that
\[
\|F(x) - F(y)\| \leq L\|x - y\| \quad \text{(18)}
\]
$\forall x, y \in N$. Properties (i) and (ii) implies that there exists positive constants $M_1, M_2$ such that
\[
\|F(x)\| \leq M_1, \quad \|F(z)\| \leq M_2, \quad \forall x \in N, \quad \text{(19)}
\]

Lemma 3.1 [4] Let the sequence $\{x_k\}$ be generated by the algorithms above. Then the sequence $\{\|F_k\|\}$ converges and $x_k \in N$ for all $k \geq 0$.

Lemma 3.2 Let the properties of (1) above hold. Then we have
\[
\lim_{k \to \infty} \|\alpha_k d_k\| = \lim_{k \to \infty} \|s_k\| = 0, \quad \text{(20)}
\]
\[
\lim_{k \to \infty} \|\alpha_k F_k\| = 0 \quad \text{(21)}
\]

Proof. by (16) and (17) we have for all $k > 0$,
\[
\omega_2 \|\alpha_k d_k\|^2 \leq \omega_1 \|\alpha_k F(x_k)\|^2 + \omega_2 \|\alpha_k d_k\|^2 \leq \|F_k\|^2 + \|F_{k+1}\|^2 + \eta_k \|F_k\|^2
\]
by summing the above $k$ inequality, then we obtain:
\[
\omega_2 \sum_{i=0}^{k} \|\alpha_i d_i\|^2 \leq \|F_k\|^2 \left(\sum_{i=0}^{k} (1 - \eta_i)\right) - \|F_{k+1}\|^2
\]
so, from (19) and the fact that $\eta_k$ satisfies (16) the series $\sum_{i=0}^{k} \|\alpha_i d_i\|^2$ is convergent. This implies (24). By a similar way, we can prove that (21) holds.

Lemma 3.1 Suppose that $\{x_k\}$ is generated by EDF algorithm. Let $s_k = x_k - x_{k-1}$. Then, we have
\[
\lim_{k \to \infty} \|\alpha_k d_k\| = \lim_{k \to \infty} \|s_k\| = 0, \quad \text{(24)}
\]

Proof.
\[
\|x_k - x_{k-1}\| = \frac{\|F(z_k)^T (x_k - z_k)\|}{\|F(z_k)\|} \leq \frac{\|F(z_k)\| \|x_k - z_k\|}{\|F(z_k)\|} = \|x_k - z_k\| = \alpha_k \|d_k\|. \quad \text{(25)}
\]

The following theorem establishes the global convergence of the EDF algorithm.

Theorem 3.3 Let $\{x_k\}$ be generated by EDF algorithm. Then, we have
\[
\lim_{k \to \infty} \inf \|F(x_k)\| = 0. \quad \text{(27)}
\]

Proof. We prove this theorem by contradiction. Suppose that (27) does not hold, then there exists a positive constant $\tau$ such that
\[
\|F(x_k)\| \geq \tau, \quad \forall k \geq 0. \quad \text{(28)}
\]
Clearly, $\|F_k\| \leq \|d_k\|$, which implies
\[
\|d_k\| \geq \tau, \quad \forall k \geq 0. \quad \text{(29)}
\]
Observe that,
\[
y_k = \|F(x_k) - F(x_{k-1})\| \leq L \|s_k\| \quad \text{(30)}
\]
meaning there exists a constant $\lambda \in (0,1)$ such that for sufficiently large $k$

$$|\theta_k| \leq \lambda. \quad (32)$$

Again from the definition of our $\beta_k^*$ we obtain

$$|\beta_k^*| \leq \frac{||\theta_k y_k - s_k||}{||\beta_k^*||s_k||} \leq M_1 \frac{||\theta_k y_k - s_k||}{||\beta_k^*||s_k||} \rightarrow 0 \quad (33)$$

which implies there exists a constant $\rho \in (0,1)$ such that for sufficiently large $k$

$$|\beta_k^*| \leq \rho. \quad (34)$$

Without lost of generality,

$$||d_k|| \leq ||\theta_k g_{k+1}|| + ||\beta_k|| ||d_k|| \leq \lambda M_1 + \rho ||d_{k-1}|| \quad (35)$$

which shows that the sequence $\{d_k\}$ is bounded. This together with (28) and (29) yields a contradiction. Hence the proof is complete.

4. Numerical Experiment

In this section, we compare the performance of our method for solving non-linear equation (1) with A derivative-free conjugate gradient method and its global convergence for solving symmetric nonlinear equations [13]. The two algorithms were implemented with the following parameters: $\omega_1 = \omega_2 = 10^{-4}, \alpha_k = 0.01, r = 0.2$ and $\eta_k = \frac{1}{(k+1)^2}$. The codes for both methods were written in Matlab 7.4 R2010a and run on a personal computer 1.8 GHz CPU processor and 4 GB RAM memory. We stopped the iteration if the total number of iterations exceeds 2000 or $||F_k|| \leq 10^{-4}$. "-" represent failure due one of the following:

(i) insufficient memory

(ii) Number of iteration exceed 2000

(iii) If $||F_k||$ is not a number (NaN)

Problems 1-5 are from [12] and the rest are artificial problems.

Problem 1. The strictly convex function:

$$F_i(x) = e^{x_i} - 1 ; 1, 2, \ldots, n$$

Problem 2. (n is multiple of 3) for $i = 1, 2, \ldots, n/3$,

$$F_{3(i-2)}(x) = x_{3i-2}x_{3i-1} - x_{3i-2}^2 - 1,$$

$$F_{3i-1}(x) = x_{3i-2}x_{3i-1}x_{3i} - x_{3i-2}^2 + x_{3i-1}^2 - 2,$$

$$F_{3i}(x) = e^{x_{3i-2}^2} - e^{-x_{3i-1}^2}.$$  

Problem 3. The Exponential function:

$$F_i(x) = \frac{1}{10}(1 - x_i^2 - e^{-x_i^2}) ; 1, 2, \ldots, n - 1$$

$$F_n(x) = \frac{1}{10}(1 - e^{-x_n^2}).$$

Problem 4. Trigonometric Function:

$$F_i(x) = 2(n+i(1-\cos x_i)) - \sum_{j=1}^{n} \cos x_j - \sum_{j=1}^{n} \sin x_j) (1\sin x_i - \cos x_i) \quad \text{for} \ i = 1, 2, \ldots, n$$

Problem 5. The discretized Chandrasekar’s H-equation:

$$F_i(x) = x_i - (1 - \frac{c}{2n} \sum_{j=1}^{n} \frac{\mu_s x_j}{\mu + \mu_s})^{-1} \quad \text{for} \ i = 1, 2, \ldots, n,$$

with $c \in [0,1]$ and $\mu = \frac{n-0.5}{n}, \quad \text{for} \ 1 \leq i \leq n$. (In our experiment we take $c = 0.9$).

Problem 6. Artificial Problem:

$$F_{2i-1}(x) = x_{2i-1} + x_{2i}(x_{2i} - 2) - 13; 1, 2, \ldots, n$$

$$F_{2i}(x) = x_{2i-1} + x_{2i}(x_{2i} + x_{2i+1} - 14) - 29.$$  

Problem 7. Artificial Problem:

$$F_i(x) = 3n - \sum_{j=1}^{n} \cos(x_i - 2) - \sum_{j=1}^{n} x_i - \sin(x_i - 2), \quad \text{for} \ i = 1, 2, \ldots, n.$$  

Problem 8. Artificial Problem:
\[ F(x) = x_i - 0.1x_{i+1}^2; \quad 1, 2, \ldots, n = 1 \]
\[ F(x) = x_i - 0.1x_{i+1}^2 \]

Problem 9 Artificial Problem:

\[ F(x) = \sum_{i=1}^{n} x_i^2 \sin x_i - x_i^4 + \sin x_i^4, \quad \text{for } i = 1, 2, \ldots, n, \]

---

**Table 1. Test Results for the Two Methods, where e = ones(n,1)**

| Problem (P) | x₀ | n  | Iter | Time | \|F₀\| | Iter | Time | \|Fₙ\| |
|-------------|----|----|------|------|-------|------|------|-------|
| P1          | e  | 50 | 7    | 0.013401 | 1.01E-07 | -    | -    | -     |
|             |    | 100| 7   | 0.007188 | 1.43E-07 | -    | -    | -     |
|             |    | 500| 7   | 0.010701 | 3.19E-07 | -    | -    | -     |
|             |    | 1000| 7   | 0.014594 | 4.51E-07 | -    | -    | -     |
|             |    | 10000| 7   | 0.15437 | 1.43E-06 | -    | -    | -     |
|             |    | 100000| 7   | 0.505017 | 4.51E-06 | -    | -    | -     |
|             |    | 500000| 7   | 2.964011 | 1.01E-05 | -    | -    | -     |
|             |    | 1000000| 7  | 6.490167 | 1.43E-05 | -    | -    | -     |
| P2          | e  | 50  | 3  | 0.003589 | 4.66E-06 | 4    | 0.0056 | 2.71E-06 |
|             |    | 3   | 0.00514 | 1.47E-05 | 4    | 0.010405 | 0.010405 |
|             |    | 5000| 3   | 0.022715 | 4.66E-05 | 4    | 0.072571 | 2.71E-05 |
|             |    | 50000| 4   | 1.59338 | 1.87E-08 | 4    | 0.61336 | 8.58E-05 |
|             |    | 500000| 4   | 1.769383 | 5.92E-08 | 7    | 3.878576 | 4.32E-08 |
| P3          | e  | 50  | 3  | 0.007857 | 1.47E-05 | 4    | 0.012256 | 8.58E-06 |
|             |    | 3   | 0.02127 | 4.66E-05 | 4    | 0.069252 | 2.71E-05 |
|             |    | 5000| 4   | 1.61407 | 1.87E-08 | 4    | 0.600705 | 8.58E-05 |
|             |    | 50000| 4   | 1.790072 | 5.92E-08 | 5    | 7.61203 | 4.32E-08 |
| P4          | e  | 1000| 26  | 0.149828 | 6.48E-05 | -    | -    | -     |
|             |    | 5000| 31  | 5.597554 | 4.53E-05 | -    | -    | -     |
|             |    | 50000| 37  | 9.69744 | 9.73E-07 | -    | -    | -     |
|             |    | 10000| 23  | 0.105184 | 3.13E-05 | -    | -    | -     |
|             |    | 100000| 34  | 5.610234 | 5.63E-06 | -    | -    | -     |
| P5          | e  | 1000| 24  | 0.067117 | 3.47E-05 | 51   | 0.223826 | 5.77E-05 |
|             |    | 5000| 28  | 0.40068 | 5.09E-05 | 60   | 1.123916 | 8.65E-05 |
|             |    | 10000| 29  | 0.644661 | 8.39E-05 | 61   | 1.961932 | 8.35E-05 |
|             |    | 50000| 29  | 2.38082 | 9.88E-05 | -    | -    | -     |
|             |    | 500000| 19  | 0.864227 | 5.53E-05 | 215  | 1.107148 | 5.42E-05 |
| P6          | e  | 500  | 546 | 0.639782 | 9.63E-05 | -    | -    | -     |
|             |    | 1000| 558  | 2.025724 | 9.86E-05 | -    | -    | -     |
|             |    | 10000| 605  | 9.510888 | 9.87E-05 | -    | -    | -     |
|             |    | 50000| 637  | 4.246983 | 9.69E-05 | -    | -    | -     |
| P7          | e  | 500  | 8  | 0.02121 | 5.14E-09 | 9    | 0.034247 | 1.34E-05 |
|             |    | 5000| 8  | 0.128485 | 2.69E-07 | 10   | 0.226067 | 2.24E-07 |
|             |    | 50000| 8  | 0.658527 | 7.36E-05 | 9    | 1.520098 | 5.07E-05 |
|             |    | 100000| 8  | 1.252776 | 3.68E-05 | 10   | 3.480638 | 1.50E-05 |
|             |    | 500  | 7  | 0.018371 | 1.61E-07 | 10   | 0.035181 | 1.08E-08 |
|             |    | 5000| 7  | 0.154703 | 2.97E-07 | 10   | 0.226576 | 9.02E-07 |
|             |    | 50000| 8  | 0.734833 | 1.07E-05 | 13   | 2.043117 | 7.45E-05 |
|             |    | 100000| 10  | 1.8931  | 6.70E-05 | 10   | 3.425698 | 7.75E-05 |

---

Enhanced Derivative-Free Conjugate Gradient Method For Solving Symmetric ... (Jamila Sabi'uw)
Table 3. Test Results for the Two Methods, where e = ones(n,1)

| Problem (P) | x₀ | n | Iter | Time | ||Fk|| | Iter | Time | ||Fk|| |
|-------------|----|---|------|------|-------|------|------|-------|
| P8          | e  | 500| 4    | 0.019037 | 2.85E-06 | 5 | 0.013187 | 5.32E-05 |
|             |    | 5000| 4    | 0.031794 | 9.00E-06 | 6 | 0.139212 | 3.28E-08 |
|             |    | 500000| 4  | 0.206865 | 2.85E-05 | 6 | 0.803616 | 1.04E-07 |
|             |    | 5000000| 4  | 2.693915 | 9.00E-05 | 6 | 7.289005 | 3.28E-07 |
|             | 0.1e| 10000| 5  | 5.726167 | 1.44E-09 | 6 | 14.47488 | 4.64E-07 |
|             | 0.1e| 100000| 3 | 0.423481 | 3.19E-07 | 3 | 1.642134 | 3.45E-05 |
|             | 0.1e| 1000000| 3 | 3.699026 | 1.01E-06 | 4 | 12.16758 | 3.99E-09 |
| P9          | e  | 1000| 23  | 0.137972 | 9.38E-05 | - | - | - |
|             |    | 10000| 29 | 1.073695 | 9.28E-05 | - | - | - |
|             |    | 50000| 32 | 3.617281 | 8.27E-05 | - | - | - |
|             |    | 100000| 28 | 6.610935 | 8.65E-05 | - | - | - |
|             |    | 500000| 30 | 24.53221 | 9.35E-05 | - | - | - |

Figure 1. Performance profile of EDF and DFCG conjugate gradient methods as the dimension increases (in term of CPU time)

Figure 2. Performance profile of EDF and DFCG conjugate gradient methods as the dimension increases (in term of number of iterations)

5. CONCLUSION

In this paper, we developed an enhanced derivative-free conjugate gradient method with less number of iterations and CPU time compared to the existing methods. It is a fully derivative-free iterative method which possesses global convergence under some reasonable conditions. Numerical comparisons using a set of large-scale test problems show that the proposed method is promising. However, to extend the method to general smooth and nonsmooth nonlinear equations will be our further research.

5.1. Conflict of Interests

There is no conflict of interest regarding the publication of this paper.
REFERENCES

[1] Cheng, W., Chen, Z., Nonmonotone Spectral method for large-Scale sym-metric nonlinear equations. Numer. Algorithms, 62(2013) 149-162.

[2] Gu, G. Z., Li, D.-H., Qi, L., Zhou, S.-Z., Descent direction of quasi-Newton methods for symmetric nonlinear equations. SIAM J. Numer. Anal. 40(2002), 1763-1774.

[3] Zhang, L., Zhou, W., Li, D.-H., Global convergence of a modi_ed Fletcher-Reeves conjugate gradient method with Armijo-type line search. Numer. Math. 104(2006), 561-572.

[4] Li, D.H., Fukushima, A globally and superlinearly convergent Gauss-Newton-based BFGS methods for symmetric nonlinear equations. SIAM J. Numer. Anal. 37(1999), 152-172.

[5] Li, D.-H., Wang, X., A modi_ed Fletcher-Reeves-type derivative-free method for symmetric nonlinear equations . Numer. Algebra Control Optim. 1(2011), 71-82.

[6] M.V. Solodov, B.F. Svaiter, A globally convergent inexact Newton method for systems of monotone equations, in: M. Fukushima, L. Qi (Eds.), Re-formulation: Nonsmooth, Piecewise Smooth, Semismooth and SmoothingMethods, Kluwer Academic Publishers,1998, pp.355369.

[7] Zhou, W., Shen, D., Convergence properties of an iterative method for solving symmetric nonlinear equations, J. Optim. Theory Appl. (2014) doi: 10.1007/s10957-014-0547-1.

[8] Yunhai, X., Chunjie, W., Soon, Y.W., Norm descent conjugate gradient method for solving symmetric nonlinear equations, J. Glo. Optim.(2014) DOI 10.1007/s10898-014-0218-7.

[9] Yuan, G., Lu, X., Wei, Z., BFGS trust-region method for symmetric non-linear equations, J. Comput. Appl. Math. 230(2009), 44-58.

[10] Zhou, W., Shen, D., An inexact PRP conjugate gradient method for sym-metric nonlinear equations Numer. Functional Anal. Opt. 35(2014), 370-388.

[11] Dolan, E.D., More, J.J., Benchmarking Optimization software with Performance pro_les. Maths. program. 91(2002), 201-213.

[12] La Cruz, W., Martinez, J.M., Raydan, M., spectral residual method without gradient information for solving large-scale nonlinear systems of equations: Theory and experiments, P. optimization 76(2004),79,1008-00.

[13] M.Y. Waziri, J. Sabi’u, A derivative-free conjugate gradient method and its global convergence for solving symmetric nonlinear equations. International J. of mathematics and mathematical science vol.(2015), doi:10.1155/2015/961487.

[14] W.W. Hager, H.Zhang, A New conjugate gradient Method with Guaranteed Descent and an efficient line search. SIAM J. Optim. 16(2005), 170-192.