Three terms of derivative free projection technique for solving nonlinear monotone equations

M M Mahdi¹ and M A K Shiker¹²
¹Department of Mathematics, College of Education for Pure Sciences, University of Babylon, Babil-Iraq.
E-mail: ¹mohmath44@gmail.com, ²mmttmhh@yahoo.com
²Corresponding Author.

Abstract. The derivative-free projection technique is one of the efficient methods for solving nonlinear monotone equations. In this study, three terms of the derivative-free projection method with a monotone line search technique is presented. This method based on extension of a conjugate gradient descent and a developed gradient projection method to solve the nonlinear system of monotone equations. The proposed method can be used for large scale equations due to limited memory requirement. We investigated the global convergence of the suggested approach without requiring differentiability and also the equation is Lipschitz continuous. The numerical results showed that the new algorithm is efficient and promised.

Keywords: Projection Algorithm, Monotone Equations, Nonlinear Systems and Line search method.

1. Introduction
We consider a derivative-free projection technique as the most effective line search methods to solve the following nonlinear system of equations:

\[ F(x) = 0, \quad x \in \mathbb{R}^n, \]  \tag{1.1}

s.t. \( F: \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}^n \) be continuous and nonlinear monotone function, \( \Omega \neq \emptyset \) closed convex, the monotonicity means

\[ \langle F(x) - F(y), x - y \rangle \geq 0, \text{ for all } x, y \in \mathbb{R}^n. \]

The gradient projection techniques are efficient to find the solution of large scale unconstraint optimization due to their simplicity and limited memory. A lot of computation methods have been proposed to solve unconstraint nonlinear problems. For example, Newton method, quasi newton method and Levenberg-Marquardt type method [1, 2]. A good property of the derivative-free for solving the monotone equation is that competitive with conjugate gradient descent [3, 4]. In this work, we developed a derivative-free projection to three terms of a derivative-free with a monotone line search technique. Also, motivated by the idea of Liu [5], we construct a new projection method of three terms derivative-free for solving large scale systems of equations. The proposed approach used to solve a large scale systems of equations because it inherit nice properties of conjugate gradient descent such as the limited memory require and high efficient. The organized of this paper as: in section one we showed the conjugate gradient projection algorithm, in section two, we presented our algorithm with a new line search, in section three some lemma and global convergence are established and in section four we introduced the numerical experiments.

The conjugate gradient descent (CGD) is one of the important methods for solving unconstraint optimization problems and nonlinear equations. It is search direction as follows:
\[ d_k = \begin{cases} -g_k & \text{if } k = 0, \\ -g_k - \beta_k d_{k-1} & \text{if } k \geq 1, \end{cases} \]

with \( g_k \) is a cost function at \( x \), \( \beta_k = \{ \beta_k^{H}, \eta_k \} \), here

\[ \beta_k^{H} = \frac{1}{\gamma k_{1}y_{k-1} - \lambda k_{1}2y_{k-1}} \norm{y_{k-1}}^2 g_k \]

\[ \eta_k = \frac{\norm{d_k} \min(\eta, \norm{g_k})}{\eta} \quad y_{k-1} = g_k - g_{k-1}, \quad t = 2, \lambda, \eta > 0. \]

Liu and Li [6] considered a conjugate gradient technique of Hager-Zhang [7] and suggested that the conjugate gradient descent method with \( t = 1 \) is high competitive than with \( t = 2 \). Also, Yan et al. [8] applied the spectral technique to analyze the conjugate gradient descent method and showed that the CGD method with \( t = 1 \) is best than with \( t = 2 \). In our method, we choose the parameter \( t = 1 \) in the proposed algorithm.

The projection operator is a mapping \( P_{\Omega}: \mathbb{R}^n \to \Omega \) for all \( x \), \( y \in \mathbb{R}^n \) holds that [9]

\[ \| P_{\Omega}(x) - P_{\Omega}(y) \| \leq \| x - y \|, \tag{1.2} \]

where \( P_{\Omega}(x) = \arg\min\{\| x - y \| | y \in \Omega \} \).

2. **Algorithm**

Given an initial point \( x_0 \) in iterative scheme for (1.1) generates a sequence \( \{ x_k \} \) by

\[ x_{k+1} = x_k + \alpha_k d_k, \quad k \in \mathbb{N} \]

which a line search procedure employs the direction \( d_k \) to calculate step size \( \alpha_k \). Let \( z_k = x_k + \alpha_k d_k \) by monotonicity of \( F \), the hyperplane

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

strictly separates \( x_k \) from the solution of the problem (1.1). Based on Solodov and Svaiter [10] advised that the other iteration point \( x_{k+1} \) is constructed by projecting \( x_k \) onto \( H_k \) that is \( x_{k+1} \) is determined by:

\[ x_{k+1} = x_k - \frac{F(z_k)^T(x_k - z_k)}{\| F(z_k) \|^2} F(z_k). \tag{2.1} \]

We assume that \( F \) holds some assumptions as follows:

- **B1** The solution set of (1.1) is nonempty.
- **B2** The mapping \( F(x) \) is monotone and Lipschitz continuous i.e., \( \exists L > 0 \), such that

\[ \| F(x) - F(y) \| \leq L \| x - y \|, \tag{2.2} \]

We propose the following new direction formula for nonlinear monotone equations (1.1)

\[ d_{k+1} = \begin{cases} -\eta F_k + \beta_k^{NM} \lambda_k + \delta_k^m \beta_k & \text{if } k \geq 1, \\ -\beta_k & \text{if } k = 0, \end{cases} \tag{2.3} \]

where \( \tau = \rho \tau \), \( m \lambda, \lambda_k = x_k - x_{k-1}, \tau, m > 0, \quad y_k = F_k - F_{k-1}, \delta_k = \eta \frac{\lambda_k^m}{\lambda + \lambda_k^m} \).

\[ \eta = \frac{\| F(z_k) \|^2}{\| F(z_k) \|^2}, \quad \beta = \frac{\| F(x_k) \|}{\| F(z_k) \|^2}, \quad \beta_k^{NM} = \frac{\rho k \tau}{\lambda^2 \tau}, \quad \beta_k = \frac{\lambda \tau m \tau}{\lambda^2 \tau} - \frac{\lambda \tau^2}{\lambda^2 \tau^2}, \tag{2.4} \]

\[ \| x_{k+1} - x_k \| \leq \epsilon, \| y_{k} \| \leq \epsilon, \quad \text{break}. \tag{2.5} \]

where \( \delta_k = F(z_k)^T(x_k - z_k) \). Put \( k = k + 1 \), and return to (2).
3. Global Convergence of the New Method

3.1. Remark (i): We conclude that by definitions of $\tau$ and $\lambda$ that
\[ \lambda^T \tau = \lambda^T \gamma_{yk} + ma^T \lambda \geq m||\lambda||^2 \geq 0. \]
This inequality is based on the monotonicity of a mapping $F$, and always the divisors of $\beta_k^{MO}$ and $\varrho_k$ are greater than zero before the algorithm breaks.

The sufficiently descent property of Algorithm (2.1) is shown in the next lemma.

3.2. Lemma: Suppose $\{d_k\}$ is the sequence of the search direction, $\{f_k\}$ be generated by Algorithm (2.1). Then $\forall k \geq 0, \exists \epsilon > 0$ such that
\[ F_k^T d_k \leq -\epsilon ||F_k||^2 \quad (3.1) \]
Proof: By the definition of $\tau$ and (2.2) we have
\[ \lambda^T \tau \leq ||\lambda||^2 \]
\[ \leq \lambda^T (r ||F_k - \bar{r}_{k-1}|| + m||\lambda||) \]
\[ \leq (rL + m)||\lambda||^2, \]
so, we have
\[ \epsilon_k \geq \frac{1}{rL + m}, \quad k \geq 1 \quad (3.2) \]

By taking inner product (3.1) with $F_k$, and from (3.2) we get
\[ F_k^T d_k = -\epsilon_k ||F_k||^2 + \beta_k^{MO} \lambda_k F_k^T + \bar{\sigma} F_k^T \]
\[ = \left( \frac{-1}{rL + m} \right) ||F_k||^2 + \frac{F_k^T \tau F_k^T + \lambda F_k^T}{(\lambda^T \tau)^2} \]
\[ F_k^T d_k \leq \left( \frac{-1}{rL + m} + \frac{1}{l + \lambda y} \right) ||F_k||^2 + \frac{F_k^T \tau F_k^T + \lambda F_k^T}{(\lambda^T \tau)^2} \]
Let $= (\lambda^T \tau)/\sqrt{2} F_k^T$, $b = \sqrt{2} (F_k^T \lambda) \tau$.

And from $\epsilon, b \leq \frac{1}{2} (\epsilon^2 + b^2)$, we get
\[ F_k^T d_k \leq \left( \frac{-1}{rL + m} + \frac{1}{l + \lambda y} \right) ||F_k||^2 + \frac{1}{2} ||F_k||^2. \]
For $k=0$, $F_0^T d_0 = -||F_0||^2$
Thus (3.1) holds.

Now, the next lemma shows that the line search of the proposed algorithms is well-defined.

3.3. Lemma: Let assumptions (B1,B2) satisfied, then there exists a step size $\alpha_k$ holds the line search (2.4) $\forall k \geq 0$.
Proof: Suppose that $\exists k_0 > 0$, $k_0$ scalar for which (2.4) is not true for all positive integer $i$ such that:
\[-F(x_{k0} + \psi d_{k0}, d_{k0}) < \eta \gamma \psi ||d_{k0}||^2, \]
by the Lipschitz continuity of $F$, set $i \rightarrow \infty$, then
\[-F(x_{k0}, d_{k0}) < 0. \quad (3.3) \]
And from (3.1) we have
\[-F(x_{k0})^T d_{k0} \geq 0. \quad (3.4) \]
This means a contradiction between (3.3) and (3.4), this implies that (2.4) is well-defined.

3.4. Lemma: Let assumptions (B1, B2) holds. The sequence $\{x_k, z_k\}$ be generated by algorithm (2.1), then for any $\bar{x}$ is a solution of (1.1) the following relation is satisfied
\[ ||x_{k+1} - \bar{x}||^2 \leq ||x_k - \bar{x}||^2 - \gamma^2 \eta^2 ||x_k - z_k||^4. \quad (3.5) \]
Proof: By the monotonicity of $F$, we get
\[ (F(z_k) - F(\bar{x}), z_k - \bar{x}) \geq 0. \]
Then
\begin{equation}
(F(\bar{z}_k) - F(\bar{x}), \bar{x} - \bar{z}_k) \geq -i(F(\bar{z}_k) - F(\bar{x}), \bar{z}_k - \bar{x}_k),
\end{equation}
from the definition of $z_k$ and (2.4)
\begin{equation}
(F(\bar{z}_k), \bar{x} - \bar{z}_k) \geq -i(F(\bar{z}_k), \bar{z}_k - \bar{x}_k)
= \bar{z}_k^\top \gamma |d_k|^2 \geq 0.
\end{equation}
From (2.1) we have
\begin{align*}
\|x_{k+1} - \bar{x}\|^2 &= \|P[\bar{x} - \delta F(z_k)] - P[\bar{x}]\|^2 \\
&\leq \|x_k - \delta F(z_k) - \bar{x}\|^2 \\
&\leq \|x_k - \bar{x}\|^2 - 2\delta \|F(z_k)^T(x_k - \bar{x})\| + \delta^2 \|F(z_k)\|^2 \\
&\leq \|x_k - x\|^2 - F(z_k)^T(z_k - \bar{x}_k) \|F(z_k)\|^2 \\
&\leq \|x_k - \bar{x}\|^2 - \eta_2 \gamma^2 \|x_k - \bar{x}\|^4 \\
&\leq \|x_k - x\|^2 - \eta_2 \gamma^2 \|x_k - z_k\|^4,
\end{align*}
where the last three inequalities are followed from (1.2), (3.6) and (2.4) respectively.

3.5. Remark (ii): By (3.5) we have
\begin{equation}
\eta_2 \gamma^2 \|x_k - z_k\|^4 \leq \|x_k - \bar{x}\|^2 - \|x_{k+1} - \bar{x}\|^2.
\end{equation}
It's not difficult to show that
\begin{equation}
\sum_{k=0}^{\infty} \eta_2 \gamma^2 \|x_k - z_k\|^4 \leq \|x_0 - \bar{x}\|^2 < +\infty.
\end{equation}
Which means that
\begin{equation}
\lim_{k \to \infty} \|x_k - z_k\| = 0.
\end{equation}

3.6. Theorem: Let assumptions (B1,B2) satisfied and the sequence $\{F_k\}$ be determined by algorithm (2.1), then
\begin{equation}
\lim inf_{k \to \infty} \|F_k\| = 0
\end{equation}
\textbf{Proof:} Assume that (3.8) is not hold. Let a constant $M > 0$ satisfies
\begin{equation}
\|F_k\| > M, \quad \forall k \geq 0.
\end{equation}
It follows from the definition of $\tau$ and (2.2) that
\begin{equation}
\|\tau_{k-1}\| \leq r \|F_k - F_{k-1}\| + m \lambda \leq (rL + m)\|\lambda\|.
\end{equation}
From remark (i) and the definition of $\beta_k^{MO}$, we get
\begin{align*}
\|\beta_k^{MO}\| &\leq \frac{r \|\lambda\| \|F_k\|}{\lambda \|F_k\|} + \frac{\|\tau\|^2 \|F_k\| \|\lambda\|}{\lambda^2 \|F_k\|^2} \\
&\leq \frac{m \|\lambda\|}{(rL + m)\|\lambda\|} \|F_k\| + \frac{(rL + m)^2 \|\lambda\|^2 \|F_k\|}{m\|\lambda\|} \\
&\leq \frac{m \|\lambda\|}{(rL + m)\|\lambda\|} \|F_k\| + \frac{(rL + m)^2 \|\lambda\|^2 \|F_k\|}{m\|\lambda\|}.
\end{align*}
By remark (i), we have
\begin{equation}
\|d_k\| \leq \frac{1}{m} \|F_k\| + \|\beta_k^{MO}\| \|\lambda\| + \|\delta\| \|\beta\|
\end{equation}
\begin{align*}
&\leq \left(1 + rL + r \frac{L^2}{m} \right) \|F_k\| + \frac{2rL}{m} \|\beta_k^{MO}\| \|\lambda\| + \frac{\|F(z_k)\|}{L + \lambda \gamma (1 + \|F(z_k)\|^2)} \|F_k\|.
\end{align*}
\[
\mathbf{V} = V \| F_k \| + \left( V + \frac{1}{L + \lambda \gamma (1 + \| F(z_k) \|^2)} \right) \| F_k \|
\]

where \( V = 1 + \frac{r^2 L + r m}{m^2} + \frac{r^2 L + r m}{m^2} \).

It follows from (2.4) that
\[
-\mathbf{r}(\mathbf{x}_k + \psi^{-1} \mathbf{a}_k d_k)^T d_k < \nu \gamma \psi^{-1} \mathbf{a}_k \| d_k \|^2,
\]

where \( \mathbf{z}_k = \mathbf{x}_k + \psi^{-1} \mathbf{a}_k d_k \).

From (3.1) and (3.2) we have
\[
\mathbf{z}_k = \mathbf{x}_k + \psi^{-1} \mathbf{a}_k d_k.
\]

From (3.11) and (3.10) we have
\[
\mathbf{a}_k \| d_k \| \geq \frac{c \| \mathbf{r}_k \|^2}{(L + \eta \gamma) \| d_k \|}.
\]

So, by (3.5) and the definition of \( \mathbf{z}_k \), we get
\[
\lim_{k \to \infty} \mathbf{a}_k \| d_k \| = 0.
\]

This implies a contradiction with (3.11), so, the assumption does not satisfied, and (3.8) holds.

4. Numerical Experiments

Numerical results are used to assess the efficiency of the new approach (MOH3). We compare it with three famous algorithms:

(GC) which is introduced by Yan et al. [8].
(HS) which is introduced by Liu and Li [6].
(SP) which is introduced by Awwal et al. [11].

The parameter of suggested algorithm set as follows: \( \gamma = 3.8, \psi = 0.7, m = 0.5, r = 0.001 \) and \( L = 0.5 \). The parameter of the other methods comes from [0, 6, 11]. All Algorithms are terminated whenever \( \| F_k \| \leq 1 \times 10^{-6} \). The total number of iteration exceeds 500000. Our computations were carried using MATLAB R2014a and run PC with 4GH, CPU2.30- Windows8 operation system. We test the performance of the algorithm (2.1) with different initial starting points [12] and various dimensions. Similar is [13, 14], we check the test problem when the variables number \( n=5000, 10000, \ldots \) with the following starting points

\[
\begin{align*}
\mathbf{x}_0 &= (10 \cdot 10, \ldots, 10)^T, \quad \mathbf{x}_1 = (-10, -10, \ldots, -10)^T, \quad \mathbf{x}_2 = (1, 1, \ldots, 1)^T, \quad \mathbf{x}_3 = (-1, -1, \ldots, -1)^T, \\
\mathbf{x}_4 &= (1, 2, 1, \ldots, 1)^T, \quad \mathbf{x}_5 = (0.1, 0.1, \ldots, 0.1)^T, \quad \mathbf{x}_6 = (1, 2, 1, \ldots, 1)^T, \quad \mathbf{x}_7 = (1, 1, 1, \ldots, 1)^T.
\end{align*}
\]

We compare the suggested method with the other method for number of iteration (NI), number of function evaluations (NF) and CPU time (CPU). From the tables (4.1, 4.2) it's clear to show that the MOH3 is better than the other methods.
Table 4.1: Numerical results

| P. | Dim. | S.P. | MOH3 | GC | HS | SP |
|----|------|------|------|----|----|----|
|    |      |      | N_v | N_f | N_v | N_f | N_v | N_f | N_v | N_f | N_v | N_f |
|    |      |      | N_v | N_f | N_v | N_f | N_v | N_f | N_v | N_f | N_v | N_f |
| 100 | 20000 | x_0 | 27  | 93  | 419 | 2546 | 121 | 244 | 56  | 114 |
| 100 | 20000 | x_1 | 27  | 93  | 419 | 2546 | 121 | 244 | 56  | 114 |
| 100 | 20000 | x_2 | 25  | 84  | 63  | 193  | 112 | 226 | 52  | 106 |
| 100 | 20000 | x_3 | 26  | 87  | 63  | 193  | 112 | 226 | 52  | 106 |
| 100 | 20000 | x_4 | 14  | 44  | 21  | 44   | 118 | 303 | 33  | 68  |
| 100 | 20000 | x_5 | 19  | 59  | 28  | 58   | 95  | 192 | 44  | 90  |
| 100 | 20000 | x_6 | 25  | 83  | 47  | 126  | 86  | 200 | 50  | 102 |
| 100 | 20000 | x_7 | 25  | 83  | 47  | 126  | 78  | 189 | 50  | 102 |
|    | 50000 | x_0 | 36  | 150 | 419 | 2546 | 121 | 244 | 56  | 114 |
|    | 50000 | x_1 | 31  | 121 | 392 | 2430 | 41  | 84  | 21  | 44  |
|    | 50000 | x_2 | 34  | 133 | 63  | 193  | 112 | 226 | 52  | 106 |
|    | 50000 | x_3 | 28  | 112 | 61  | 242  | 33  | 68  | 18  | 38  |
|    | 50000 | x_4 | 22  | 90  | 21  | 44   | 120 | 321 | 33  | 68  |
|    | 50000 | x_5 | 32  | 122 | 28  | 58   | 95  | 192 | 44  | 90  |
|    | 50000 | x_6 | 35  | 132 | 47  | 126  | 91  | 209 | 50  | 102 |
|    | 50000 | x_7 | 35  | 132 | 47  | 126  | 91  | 209 | 50  | 102 |
|    | 10000 | x_0 | 50  | 244 | 108 | 418  | 62  | 127 | 99  | 273 |
|    | 10000 | x_1 | 81  | 388 | 261 | 1340 | 90  | 183 | 124 | 333 |
|    | 10000 | x_2 | 51  | 248 | 65  | 194  | 55  | 113 | 90  | 250 |
|    | 10000 | x_3 | 63  | 303 | 90  | 273  | 76  | 155 | 104 | 282 |
|    | 10000 | x_4 | 59  | 284 | 66  | 175  | 68  | 139 | 91  | 247 |
|    | 10000 | x_5 | 65  | 312 | 70  | 183  | 72  | 147 | 92  | 248 |
|    | 10000 | x_6 | 58  | 280 | 67  | 182  | 66  | 157 | 95  | 259 |
|    | 10000 | x_7 | 58  | 280 | 67  | 182  | 65  | 156 | 95  | 259 |
Table 4.1: Numerical results - continued

| Dim. | S.P | MOH3 | GC | HS | SP |
|------|-----|------|----|----|----|
| 10000 | $x_0$ | 61 | 247 | 21094 | 209916 | 2344 | 11705 | 133 | 268 |
| 10000 | $x_1$ | 68 | 276 | 20987 | 208704 | 2386 | 11804 | 133 | 268 |
| 10000 | $x_2$ | 66 | 268 | 21084 | 209805 | 2498 | 12442 | 133 | 268 |
| 10000 | $x_3$ | 67 | 272 | 21067 | 209609 | 2377 | 11912 | 133 | 268 |
| 10000 | $x_4$ | 67 | 272 | 21075 | 209709 | 2534 | 12640 | 133 | 268 |
| 10000 | $x_5$ | 67 | 272 | 21076 | 209719 | 2529 | 12634 | 133 | 268 |
| 10000 | $x_6$ | 67 | 272 | 21082 | 209786 | 2509 | 12550 | 133 | 268 |
| 10000 | $x_7$ | 66 | 268 | 21077 | 209728 | 2444 | 12208 | 133 | 268 |
| 5000  | $x_0$ | 25 | 97 | 301 | 1550 | 128 | 258 | 59 | 120 |
| 5000  | $x_1$ | 26 | 99 | 550 | 3263 | 133 | 268 | 61 | 124 |
| 5000  | $x_2$ | 24 | 87 | 95 | 350 | 117 | 236 | 54 | 110 |
| 5000  | $x_3$ | 24 | 79 | 174 | 788 | 123 | 248 | 57 | 116 |
| 5000  | $x_4$ | 25 | 96 | 125 | 500 | 121 | 244 | 56 | 114 |
| 5000  | $x_5$ | 25 | 89 | 122 | 486 | 120 | 242 | 55 | 112 |
| 5000  | $x_6$ | 27 | 102 | 112 | 435 | 119 | 240 | 55 | 112 |
| 5000  | $x_7$ | 24 | 93 | 112 | 435 | 119 | 240 | 55 | 112 |
| 50000 | $x_0$ | 15550 | 31145 | 44711 | 90690 | 129477 | 258956 | 118747 | 237496 |
| 50000 | $x_1$ | 789 | 1960 | 471 | 2844 | 71 | 144 | 36 | 74 |
| 50000 | $x_2$ | 15528 | 31058 | 44380 | 88766 | 129445 | 258892 | 118732 | 237466 |
| 50000 | $x_3$ | 647 | 1553 | 48 | 192 | 54 | 110 | 49 | 60 |
| 50000 | $x_4$ | 896 | 1794 | 2565 | 5132 | 7490 | 14982 | 3190 | 6382 |
| 50000 | $x_5$ | 15234 | 30470 | 43530 | 87062 | 126968 | 253938 | 117676 | 235354 |
| 50000 | $x_6$ | 15330 | 30662 | 43775 | 87552 | 128794 | 257609 | 117633 | 235268 |
| 50000 | $x_7$ | 15330 | 30662 | 43775 | 87552 | 128634 | 257289 | 117632 | 235266 |
Table 4.2: Numerical results (CPU time)

| P.  | Dim. | S. P | MOH3 | CPU time | GC  | HS  | SP  |
|-----|------|------|------|----------|-----|-----|-----|
| 20000 | 2000 | 0.5642 | 16.59375 | 2.89062 | 0.50000 |
| 20000 | 2000 | 1.1094 | 16.46875 | 2.35937 | 0.39062 |
| 20000 | 2000 | 1.0937 | 0.84375 | 1.60937 | 0.32812 |
| 20000 | 2000 | 0.9687 | 0.92187 | 1.43750 | 0.35937 |
| 20000 | 2000 | 0.5468 | 0.21873 | 1.81250 | 0.23437 |
| 20000 | 2000 | 0.5937 | 0.25000 | 1.42187 | 0.29687 |
| 20000 | 2000 | 1.0000 | 0.65625 | 1.14062 | 0.37500 |
| 20000 | 2000 | 1.1094 | 0.59375 | 0.98437 | 0.28125 |
| 50000 | 5000 | 0.5781 | 16.37500 | 2.79687 | 0.48437 |
| 50000 | 5000 | 0.4062 | 15.32812 | 0.82812 | 0.14062 |
| 50000 | 5000 | 0.4843 | 0.95312 | 2.39062 | 0.35937 |
| 50000 | 5000 | 0.3125 | 1.07812 | 0.64062 | 0.12500 |
| 50000 | 5000 | 0.3437 | 0.23437 | 2.35937 | 0.23437 |
| 50000 | 5000 | 0.4843 | 0.26562 | 2.06250 | 0.29687 |
| 50000 | 5000 | 0.5156 | 0.60937 | 2.20312 | 0.20312 |
| 50000 | 5000 | 0.4843 | 0.60937 | 2.15625 | 0.32812 |
| 50000 | 5000 | 2.1189 | 494.59375 | 286.39062 | 394.71875 |
| 50000 | 5000 | 2.3528 | 452.89062 | 345.68750 | 439.18750 |
| 50000 | 5000 | 1.8631 | 336.79687 | 242.79687 | 319.79687 |
| 50000 | 5000 | 2.0645 | 396.03125 | 297.95312 | 376.59375 |
| 50000 | 5000 | 1.8710 | 341.31250 | 255.04687 | 327.46875 |
| 50000 | 5000 | 1.8355 | 331.34375 | 246.32812 | 313.59375 |
| 50000 | 5000 | 783.84375 | 144.21875 | 103.96875 | 137.81250 |
| 50000 | 5000 | 781.85937 | 144.04687 | 109.07812 | 181.32812 |
| 10000 | 1000 | 0.1406 | 0.34375 | 0.65620 | 0.20312 |
| 10000 | 1000 | 0.2500 | 1.43750 | 0.93750 | 0.25000 |
| 10000 | 1000 | 0.1562 | 0.20312 | 0.57812 | 0.15625 |
| 10000 | 1000 | 0.1875 | 0.32812 | 0.20312 | 0.17187 |
| 10000 | 1000 | 0.1562 | 0.17187 | 0.34375 | 0.15625 |
| 10000 | 1000 | 0.1875 | 0.23437 | 0.26562 | 0.17187 |
| 10000 | 1000 | 0.1718 | 0.18750 | 0.20312 | 0.18750 |
| 10000 | 1000 | 0.1093 | 0.25000 | 0.31250 | 0.20312 |
Table 4.2: Numerical results (CPU time) - continued

| P. | Dim. | S.P | CPU time |
|----|------|-----|----------|
|    |      |     | MOH3     | GC       | HS       | SP       |
|    |      |     | 0.17187  | 80.87500 | 12.76562 | 0.62500  |
|    |      |     | 0.10937  | 79.59375 | 12.84375 | 0.35937  |
|    |      |     | 0.14062  | 80.92187 | 20.25000 | 0.37500  |
|    |      |     | 0.10937  | 79.78125 | 23.26562 | 0.43750  |
|    |      |     | 0.09375  | 80.65625 | 24.42187 | 0.35937  |
|    |      |     | 0.10937  | 79.14062 | 12.0312  | 0.35937  |
|    |      |     | 0.10937  | 81.15625 | 12.23437 | 0.35937  |
|    |      |     | 0.07812  | 80.25000 | 11.59375 | 0.32812  |
|    |      |     | 0.76562  | 12.10937 | 3.53125  | 2.85937  |
|    |      |     | 0.62500  | 24.17187 | 3.53125  | 2.03125  |
|    |      |     | 0.57812  | 2.54687  | 3.04687  | 1.57812  |
|    |      |     | 0.53125  | 6.29687  | 3.28125  | 1.53125  |
|    |      |     | 0.67187  | 3.96875  | 3.01562  | 1.23437  |
|    |      |     | 0.56250  | 3.62500  | 3.43750  | 1.04687  |
|    |      |     | 0.73437  | 3.39062  | 3.62500  | 1.01562  |
|    |      |     | 0.68750  | 3.57812  | 3.07812  | 0.90625  |
|    |      |     | 151.26562| 392.12500| 1651.3750| 15330.6875|
|    |      |     | 8.32812  | 16.64062 | 1.07812  | 4.82812  |
|    |      |     | 148.37500| 367.04687| 1682.4375| 15549.17187|
|    |      |     | 6.62500  | 0.70312  | 0.70312  | 3.82812  |
|    |      |     | 8.39062  | 20.78125 | 97.00001 | 418.82812|
|    |      |     | 147.18750| 362.89062| 1624.6094 |15450.40625|
|    |      |     | 148.45312| 357.81250| 1672.00001|15450.40625|
|    |      |     | 150.73437| 359.92187| 1676.07812|15431.67187|

5. Conclusions

We have suggested a new class of three terms of derivative-free projection technique for solving unconstrained optimization. The suggested approach is appropriate for large scale equations, because it has a nice property which is the low memory requirement. The global convergence of our method is established. The numerical results showed that our method is efficient and working better than the three other algorithms that the new algorithm is compared with.

6. References

[1] Cheng W Y and Li D H 2010 Spectral scaling BFGS method, J. Optim. Theor. Appl. 146 305–319.
[2] Li Q N and Li D H 2011 A class of derivative-free methods for large-scale nonlinear monotone equations, *IMA J. Numer. Anal.* 31 1625–1635.

[3] Amini K, Shiker M A K and Kimiae M 2016 A line search trust-region algorithm with nonmonotone adaptive radius for a system of nonlinear equations, *4 OR- Journal of operation research* 14 (2) 133-152.

[4] Hashim K H, Dreeb, N K, Dwail, H H, Mahdi M M, Wasi H A, Shiker M A K and Hussein H A 2019 A new line search method to solve the nonlinear systems of monotone equations, *Journal of Engineering and Applied Sciences*, 14 10080-10086.

[5] Liu J K 2016 Derivative-free spectral PRP projection method for solving nonlinear monotone equations with convex constraints, *Math Numer Sin.*, 38 113–24[in Chinese].

[6] Liu J K and Li S J 2015 Spectral DY-type projection method for nonlinear monotone systems of equations, *JCM.*, 4 341–354.

[7] Hager W W and Zhang H 2005 A new conjugate gradient method with guaranteed descent and an efficient line search, *SIAM J. Optim.* 16 170–92.

[8] Yan Q R, Peng X Z and Li D H 2010 A globally convergent derivative-free method for solving large-scale nonlinear monotone equations, *J. Comput. Appl. Math.*, 234 649-657.

[9] Shiker M A K and Amini K 2018 A new projection-based algorithm for solving a large scale nonlinear system of monotone equations, *Croatian operational research review*, 9 63-73.

[10] Solodov M V and Svaiter B F 1999 A globally convergent inexact Newton method for systems of monotone equations, in: M. Fukushima, L. Qi (Eds.), Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, *Kluwer Academic Publishers*, 355-369.

[11] Awwal A M, Kumam P, Abubakar A B and Wakili A 2018 A projection Hestenes-Stiefel like method for monotone nonlinear equations with convex constraints, *Thai Journal of Mathematics*, 16 181-199.

[12] Shiker M A K and Sahib Z 2018 A modified technique for solving unconstrained optimization, *J. Eng. Applied Sci.*, 13 9667-9671.

[13] Dreeb N K, Hashim K H, Mahdi M M, Wasi H A, Dwail H H, Shiker M A K and Hussein H A 2019 Solving a large-scale nonlinear system of monotone equations by using a projection technique, *Journal of Engineering and Applied Sciences*, 14 10102-10108.

[14] Hassan Z A H and Shiker M A K 2018 Using of generalized baye’s theorem to evaluate the reliability of aircraft systems. Journal of Engineering and Applied Sciences, Special Issue 13, 10797–10801.