Solving Systems of Nonlinear Monotone Equations by Using a New Projection Approach

Mohammed M. Mahdi¹, Hasan Hadi Dwail², H.A. Wasi³, Karrar Habeeb Hashim⁴, Nabiha kahtan Dreeb⁵, Hussein Ali Hussein⁶ and Mushtak A.K. Shiker⁷*

Department of Mathematics, Collage of Education for Pure Sciences, University of Babylon, Hilla, Iraq

mohmath44@gmail.com
mathhasan74@gmail.com
Mathman1972@gmail.com
teacher89karrar@gmail.com
nabiha250222@gmail.com
hussainalseaidi@gmail.com
mmttmhhhi@yahoo.com

Abstract. The projection technique is one of the famous method and highly useful to solve the optimization problems and nonlinear systems of equations. In this work, a new projection approach for solving systems of nonlinear monotone equation is proposed combining with the conjugate gradient direction because of their low storage. The new algorithm can be used to solve the large-scale nonlinear systems of equations and satisfy the sufficient descent condition. The new algorithm generates appropriate direction then employs a good line search along this direction to reach a new point. If this point solves the problem then the algorithm stops, otherwise, it constructs a suitable hyperplane that strictly separate the current point from the solution set. The next iteration is obtained by projection the new point onto the separating hyperplane. We proved that the line search of the new projection algorithm is well defined. Furthermore, we established the global convergence under some mild conditions. The numerical experiment indicates that the new method is effective and very well.

Keywords. Projection Method, Monotone Equation, Conjugate Gradient Method and Line Search Method.

1. Introduction

The projection method is a class of iterative methods for solving the systems of nonlinear equations. In general, this system is a family of problems close to optimization problems and often appear in the applied sciences, industry and technology.

Many examples from all of these branches have been considered in recent years. The mathematical formula of nonlinear system of equations is:

\[ F(x) = 0, \quad (1.1) \]
Where \( F : R^n \to R^m \) is a continuous and monotone function. If \( m > n \) this system is called an overdetermined system, while it is called an underdetermined when \( n > m \). Furthermore, this system is called a square nonlinear system of equations if \( m = n \). By a monotone function we mean:

\[
(F(x) - F(y))^T(x - y) \geq 0, \quad \forall x, y \in R^n.
\] (1.2)

The solution set of (1.1) is a convex set when it is not empty [1]. The system of nonlinear monotone equations arise in various applications such as subproblems in the extension proximal technique with Bregman distances and first order necessary condition (FONC) of the unconstraint convex optimization. They can be converted some monotone variational inequality into systems of nonlinear monotone equations by means of fixed point map or normal map if the implicit maps hold some coercive conditions [2]. Other methods have been developed by several applications of systems of nonlinear monotone equations. The iterative technique is a suitable method to solve these equations, they have received more attention in recent years ([3,4,5]). Solodov and Svaiter [6] proposed an inexact Newton method for solving (1.1) by combining the Newton method with a projection strategy. This method has an attractive property is that the whole sequence of iterates converges to a solution of the system without any regularity assumptions. In addition, the sequence of the distances is decreasing from the iterates to the solution set of the equations. Zhou and Toh [5] extended Solodov and Svaiter's result and obtained the superlinear convergence of a Newton-type methods even if the equation has singular solutions. Zhou and Li [7] combination Solodov and Svaiter's projection technique and limited memory BFGS methods for solving a large scale monotone equation by another algorithm.

Furthermore, conjugate gradient algorithm are another class of approaches extended by appropriate method to solve large-scale optimization problems and systems of nonlinear equations, there is many approaches to handle large scale problems similar in the references [1,8,9].

Shiker and Amini [5] presented a new projection based algorithm for solving large scale nonlinear system of monotone equations. Numerical results of the reported refer that these methods have been suitable.

In this work, a new projection approach for solving systems of nonlinear monotone equations is proposed, the suggested method can be applied to solve nonsmooth equations. Moreover, it is a suitable to solve large-scale equations due to the lower storage requirement of the conjugate gradient method. The rest of this paper, the second section presents a projection method based algorithms and subsequently their conjugate gradient procedure. In the third section, under some mild conditions we investigate the global convergence of the proposed technique. In the fourth section, we establish some preliminary numerical experiments and compared by some famous algorithm. The last section, conclusion of the new algorithm is presented.

2. Projection Algorithms: Motivation and Theory

There are numerous methods for solving (1.1) generally, based on the following unconstrained optimization problem

\[
\text{minimize } f(x) = \frac{1}{2} \|F(x)\|^2. \quad \text{s. t. } x \in R^n
\]

Newton and quasi-Newton strategies [2, 3, 4, 13] are examples for such methods. Iterative technique customarily solving the system of nonlinear equations and generate a sequence of iterates \( \{x_k\} \) by \( x_{k+1} = x_k + \alpha_k d_k \), with \( \alpha_k > 0 \) is a step length limited by a line search procedures and a search direction \( d_k \) which (for all \( k \in N \)) satisfies

\[
F_k^T d_k \leq -\theta \|F_k\|^2,
\] (2.1)

where \( \theta \) is a positive constant [10, 6]. Note that this condition is exactly the sufficient descent condition when \( F(x) \) is a real value function has a gradient vector.

Generally, these methods firstly compute a step direction \( d_k \) by some specific approaches for example [11,4,8] and using a line search condition to find a step length \( \alpha_k > 0 \) such that

\[
F(z_k)^T(x_k - z_k) > 0,
\] (2.2)
where
\[ z_k = x_k + \alpha_k d_k. \tag{2.3} \]

After that, the hyperplane
\[ H_k = \{ x \in R^n | F(z_k)^T (x - z_k) = 0 \}, \tag{2.4} \]

strictly separates the current iterate \( x_k \) from zeros of the equation system \((1.1)\).

And we obtained the next iteration point, \( x_{k+1} \), by projecting \( x_k \) onto hyperplane \( H_k \). We can obtain a best approximation for a solution set of \((1.1)\) by determined \([6]\)
\[ C_k = \{ x \in R^n | F(z_k)^T (x - z_k) \leq 0 \} \]

This contains the solution of problem, but \( x_k \notin C_k \) by projecting \( x_k \) on \( C_k \). can be determined the next approximation,
\[ x_{k+1} = P_C(x_k) = \text{argmin} \{ \| x - x_k \| | x \in C_k \} \]
\[ = x_k - \frac{F(z_k)^T (x_k - z_k)}{\| F(z_k) \|^2} F(z_k). \tag{2.5} \]

Indicate theoretical properties and the numerical results to the efficiency and the robustness of the projection-procedure for monotone equations introduced to locate cheap and effective directions in the score. It is famous that the conjugate gradient technique are a class of iterative methods that developed extensively to solve large-scale systems of nonlinear equations as well as optimization problems. As a consequence of low desired memory, these are proper to solve large-scale problems. Inspired by this valuable property, it is prepared to using the conjugate gradient directions for the projection algorithm satisfying \((2.2)\).

Constructing and employ conjugate gradient procedures in our algorithm grant us some advantages: firstly, these techniques are globally convergent without the differentiability assumptions, and the next advantages decrease the computational cost of the algorithm and the number of iterations and function evaluations and, at last, they can improve the efficiency of the new algorithm for solving large scale nonlinear system of monotone equation.

The convergence property of this method was investigated using a new line search condition
\[ -F(x_k + \delta \alpha_k d_k)^T d_k \geq (1 - \delta) \alpha_k \| d_k \|^2 \tag{2.6} \]
Where \( \delta > 0 \) is a constant.

Based on the idea described by Koorapetse et al. \([11]\), to determine a conjugate gradient direction, a new direction can be defined as follows:
\[ d_k = \begin{cases} -F_k & \text{if } k = 0, \\ -\zeta F_k + (1 - \zeta) \beta_k^m w_{k-1} & \text{if } k \geq 1, \end{cases} \tag{2.7} \]
where
\[ \zeta = \frac{1}{2} \frac{\| F(x_k) \|}{\| F(z_k) \|}. \]

To use in a projection procedure where
\[ \beta_k^m = \delta \xi + (1 - \delta) \psi, \tag{2.8} \]
\[ \xi = \frac{\| y_k \|}{\| F_{k-1} \|^2}, \quad \psi = \frac{F(x_k)^T y_k}{\| F_{k-1} \|^2}, \]

where \( y_k = F_k - F_{k-1} \) and \( w_k = z_k - x_k \).

Now, we state the steps our algorithm as follows:
Algorithm (2.1) New Projection Algorithm (MOH):
**Step 0:** An initial point \( x_0 \in R^n \), \( n, k, \gamma > 0 \) and \( \delta, \varepsilon \in (0, 1) \).
**Step 1:**
Set \( k = 0; \)
\( F_0 = F(x_0) \)
\( d_0 = -F_0. \)
While \( \| F_k \| > \varepsilon \).
Step 2:
Determine an initial step length $\beta$
Set $\alpha_3 = \beta$;
Compute $-F(x_k + \delta\alpha_3 d_k)^T d_k \geq (1 - \delta)\alpha_3 \|d_k\|^2$
While $\alpha_k = \delta \alpha_3$;
Set $z_k = x_k + \alpha_k d_k$.
End While
Step 3:
If $\|F(z_k)\| \leq \varepsilon$ break. Otherwise compute $x_{k+1}$ by (2.5).
Step 4: choose the search direction $d_k$ by (2.7).
$F_{k+1} = F(x_{k+1})$;
If $F_k^T d_k > -\varepsilon \|F_k\|^2$
$d_k = -F_k$;
End If
$k = k + 1$.
End While
Remark (2.2) : from Step 4 of Algorithm (2.1) it is easy to see that, for any $k$,
$F_k^T d_k \leq -\theta \|F_k\|^2$
Then $d_k$ is always hold the sufficient descent conditions.

3. Convergence Properties
Now, we devoted to the convergence property of Algorithm (2.1), in this paper we need some assumptions:

(i) The solution set is nonempty of the problem (1.1).
(ii) $F(x)$ is monotone function on $R^n$.
(iii) The mapping $F(x)$ is Lipschitz continuous function on $R^n$, i.e, there exists a constant $L \geq 0$, such that
$\|F(x) - F(y)\| \leq L \|x - y\|, \forall x, y \in R^n$.

Now, we give some lemmas and remarks indicate that important properties of a projection on a closed convex and we establish these algorithm is globally convergence.

Lemma(3.1):[9] let $\Omega \subseteq R^n$ be a nonempty closed convex set and $P_\Omega(x)$ be the projection of $x$ on $\Omega$, for all $x, y \in R^n$, the following hold:
$\forall i \in \Omega, \langle P_\Omega(x) - x, z - P_\Omega(x) \rangle \geq 0$ and the inequality is a strict when $P_\Omega(x) \neq P_\Omega(y)$.
$\|P_\Omega(x) - P_\Omega(y)\| \leq \|x - y\|$.

Lemma(3.2): suppose assumptions (i) and (ii) hold and the sequence $\{x_k\}$ is generated by algorithm (2.1), for any $x^*$ s. t. $F(x^*) = 0$, then
$\|x_{k+1} - x^*\| \leq \|x_k - x^*\|^2 - \|x_{k+1} - x_k\|^2$. (3.1)
And, the sequence $\{x_k\}$ is bounded, either the sequence $\{x_k\}$ is finite while the last iterate is a solution for (2.1) or the sequence is infinite and
$\lim_{k \to \infty} \|x_{k+1} - x_k\| = 0$. (3.2)

{$x_k$} converse to some solution of (1.1).
Proof: theorem (2.1) in Solodov and Svaiter [6] give the conclusion immediately
Now, we derive some properties of algorithm(2.1) and show that the line search is well defined.
Lemma(3.3): suppose that the assumption (i),(ii)and (iii) holds, and the sequence $\{x_k\}$ and $\{z_k\}$ are generated by algorithm (2.1) then we have
\[ \alpha_k \geq \min \left\{ \beta, \frac{\rho \theta \|F_k\|^2}{(\delta L + 1 - \delta)\|d_k\|^2} \right\}, \quad \alpha = \beta \rho^i , i = 1, 2, \ldots, k \] (3.3)

Proof:-

Let \( \alpha_k \neq \beta \), from the line search (2.6) we know that \( \alpha_k^\text{hat} = \rho^{-1} \alpha_k \) doesn't hold (2.6), this mean that:

\[ -F(x + \delta \alpha_k \rho^{-1} d_k)^T d_k < (1 - \delta) \alpha_k \rho^{-1} \|d_k\|^2 \]

\[ \leq (1 - \delta) \alpha_k^\text{hat} \|d_k\|^2 \]

where

\[ z_k^\text{hat} = x_k + \delta \alpha_k^\text{hat} d_k. \]

The Lipschitz continuity of \( F \) and from (2.1) conclude that:

\[ \frac{\|F(z_k^\text{hat}) - F(x_k)\|}{\|d_k\|} \leq \frac{\|(F(z_k^\text{hat}) - F(x_k))\|}{\|d_k\|} + \frac{(1 - \delta) \alpha_k^\text{hat} \|d_k\|^2}{\|d_k\|} \]

\[ \leq \frac{L \delta \alpha_k^\text{hat} \|d_k\|^2 + \alpha_k^\text{hat} \|d_k\|^2 - \delta \alpha_k^\text{hat} \|d_k\|^2}{\|d_k\|^2} \]

\[ \|F(z_k^\text{hat})\|^2 \leq \frac{\delta}{(\delta L + 1 - \delta)} \|d_k\|^2, \quad \beta = \alpha_k \]

This implies that (3.3) is correct.

In the present section, we introduce the global convergence of the new method

**Theorem (3.4):**

Let the assumptions (i) and (ii) hold and the sequence \( \{x\} \) is generated by algorithm (2.1), then we have

\[ \lim_{k \to \infty} \|F_k\| = 0 \] (3.4)

Proof:-from the line search procedure and the relation (2.5) and (2.6) result that:

\[ \|x_{k+1} - x_k\|^2 \leq \frac{|F(z_k^\text{hat})^T(x_k - z_k)|}{\|F(z_k^\text{hat})\|} \]

\[ \geq \frac{|(1 - \delta) \alpha_k \|d_k\|^2 \delta \alpha_k^\text{hat}}{\|F(z_k^\text{hat})\|} \]

\[ = \frac{(1 - \delta) \alpha_k^\text{hat} \|d_k\|^2}{\|F(z_k^\text{hat})\|} \] (3.5)

Since the sequence \( \{x_k\} \) has the boundedness along with the continuity of the function \( F \) implies that \( \|F(x_k)\| \) is bounded and there is a constant \( M > 0 \), s.t.

\[ \|F(x_k)\| \leq M. \]

By the Lipschitz continuity of \( F \), it can be result that

\[ \|F(z_k)\| \leq \|F(x_k)\| + \|F(x_k)\| \]

\[ \leq L(z_k - x_k) + M \]

\[ = L \delta \alpha_k \|d_k\| + M \] (3.6)

By (3.5) to gather with (3.6) implies

\[ \|x_{k+1} - x_k\|^2 \geq \frac{\delta(1 - \delta) \alpha_k^2 \|d_k\|^2}{L \delta \alpha_k \|d_k\| + M} \]
So,
\[ \lim_{k \to \infty} \|x_{k+1} - x_k\|^2 \geq \lim_{k \to \infty} \left( \frac{\delta(1 - \delta)\alpha_k^2\|d_k\|^2}{L\delta\alpha_k\|d_k\| + M} \right) \]

It's easy to result that
\[ \lim_{k \to \infty} \alpha_k\|d_k\| = 0. \quad (3.7) \]

Moreover, by using Cauchy Schwartiz inequality along with (2.1) we obtain
\[ \theta\|F_k\|^2 \leq - F_k^T d_k \leq \|F_k\|\|d_k\| \quad \Rightarrow \|F_k\|^2 \leq \|F_k\|\|d_k\| \to \|F_k\| \leq \|d_k\| \quad (3.8) \]

On the other hand multiplying (3.1) by \( \|d_k\|^2 \) result that
\[ \alpha_k\|d_k\|^2 \geq \text{Min} \left\{ \beta\|d_k\|^2, \frac{\rho \theta\|F_k\|^2}{(\delta L + 1 - \delta)} \right\} \quad (3.9) \]

by (3.8) and (3.9), it can be conclude that
\[ A\|F_k\|^2 \leq \alpha_k\|d_k\|^2 \quad (3.10) \]

Where
\[ A = \text{Min} \left\{ \beta\rho^2, \frac{\rho \theta}{(\delta L + 1 - \delta)} \right\} \]

The relation (3.7) together with (3.10) conclude that
\[ \lim_{k \to \infty} \|F_k\| = 0 \]

4. Numerical Experiments

We introduce these experimentations on some widely used problems with the different range of initial points and the large number of variables, to evaluation effectiveness and robustness of the suggested algorithm for large scale nonlinear monotone equations. We proposed some comparisons between our proposed approaches (MOH) with a class of derivative free methods (MPRP), and a PRP type method for systems of monotone equations (PRP) and spectral gradient projection method (SGPM) proposed by Li and Li in [3], Cheng [12] and Zhang and Zhou [9], respectively.

We give some comparisons among our proposed approaches, MOH, MPRP, PRP and SGPM respectively. All of these algorithms are terminated whenever \( \|F_k\| \leq 10^{-4} \) or \( \|F(z_k)\| \leq 10^{-4} \), or the total number of iterates exceeds 50000. The parameters in all of the technique, the variable are determined as follows \( \delta = 0.9 \), \( \gamma = 0.6 \), \( tol = 10^{-4} \) and the starting adaptive step length is evaluated by
\[ \beta = \frac{\gamma \rho \theta d_k}{\|F(x_k + \epsilon d_k - F_k)^T d_k\|} \]

where \( \epsilon = 10^{-8} \).

The experiments are run on a PC with CPU 2.20 GHz and 4 GB RAM. Every codes were written in MATLAB R2014a programming environment. The test problem are similar to the problems in [12,8], and the same starting points:

- \( x_0 = (10,10,...,10)^T \), \( x_1 = (-10,-10,...,-10)^T \)
- \( x_2 = (1,1,1)^T \), \( x_3 = (-1,-1,...,-1)^T \)
- \( x_4 = (1,\frac{1}{2},\frac{1}{3},\frac{1}{4},\ldots)^T \), \( x_5 = (0.1,0.1,...,0.1)^T \)
- \( x_6 = (\frac{1}{n},\frac{1}{n},\ldots)^T \), \( x_7 = (1 - \frac{1}{n},1 - \frac{2}{n},...,0)^T \)

In running the codes, it is checked whether various codes converge to the same point and if the provided data for problems in all algorithms converge to the identical points. The numerical results on the eight problems are presented in Table (4.1) and (4.2), where ‘Prob’, ‘Dim’, ‘Iter’, ‘Feval’ and ‘CPU’ stand for problem number, dimension of the problem, number of iterations, number of function evaluations and CPU time taken for each method to reach the optimal value or termination, respectively.

Thanks to our preliminary numerical experiments, it has been found that employing the parameters \( \delta = 0.9 \), \( \gamma = 0.6 \) for all considered algorithms have the best results.

| P. | Dim. | S.P | New MPRP | PRP | SGPM |
|---|---|---|---|---|---|
|  |  |  |  |  |  |
### Table 4.1. Numerical results – continued

| Ni  | Nf  | Ni  | Nf  | Ni  | Nf  | Ni  | Nf  |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 110971 | 332923 | 127274 | 381981 | 169107 | 1689280 | 438303 | 6855466 |
| 26936 | 80812 | 128016 | 413610 | 164596 | 1645102 | 119586 | 1994620 |
| 6421 | 19266 | 6430 | 19325 | 167997 | 1677675 | 11887 | 1056987 |
| 6427 | 19285 | 136015 | 423023 | 165017 | 1648775 | 213193 | 1924591 |
| 6429 | 19290 | 136126 | 423426 | 166911 | 1668576 | 11896 | 107014 |
| 6427 | 19284 | 6433 | 19335 | 166179 | 1660544 | 11891 | 106970 |
| 6425 | 19278 | 6433 | 19326 | 166553 | 1665400 | 11890 | 106965 |
| 6425 | 19278 | 127940 | 413890 | 165194 | 1651450 | 212953 | 1921952 |
| 18 | 59 | 29 | 492 | 32 | 85 | 26 | 55 |
| 18 | 59 | 29 | 492 | 32 | 85 | 26 | 55 |
| 16 | 51 | 9 | 64 | 30 | 61 | 21 | 44 |
| 16 | 51 | 9 | 64 | 30 | 61 | 21 | 44 |
| 10 | 33 | 47 | 144 | 19 | 39 | 13 | 28 |
| 10 | 33 | 47 | 144 | 19 | 39 | 13 | 28 |
| 13 | 42 | 5 | 18 | 25 | 51 | 17 | 36 |
| 13 | 42 | 5 | 18 | 25 | 51 | 17 | 36 |
| 15 | 48 | 71 | 226 | 29 | 59 | 20 | 42 |
| 15 | 48 | 71 | 226 | 29 | 59 | 20 | 42 |
| 15 | 48 | 71 | 226 | 29 | 59 | 20 | 42 |
| 18 | 59 | 29 | 492 | 22 | 47 | 26 | 55 |
| 17 | 55 | 25 | 430 | 10 | 48 | 20 | 42 |
| 16 | 51 | 9 | 64 | 29 | 60 | 21 | 44 |
| 16 | 51 | 9 | 64 | 29 | 60 | 21 | 44 |
| 12 | 39 | 32 | 102 | 18 | 38 | 13 | 28 |
| 12 | 39 | 32 | 102 | 18 | 38 | 13 | 28 |
| 13 | 42 | 5 | 18 | 24 | 50 | 17 | 36 |
| 13 | 42 | 5 | 18 | 24 | 50 | 17 | 36 |
| 23 | 72 | 44 | 146 | 28 | 58 | 20 | 42 |
| 23 | 72 | 44 | 146 | 28 | 58 | 20 | 42 |
| 32173 | 96522 | 32219 | 96704 | 42222 | 661312 | 61439 | 532451 |
| 35733 | 107202 | 35786 | 107454 | 46862 | 734044 | 68174 | 590837 |
| 26224 | 78675 | 26261 | 78806 | 34365 | 538242 | 50181 | 434879 |
| 31017 | 93054 | 31060 | 93203 | 40881 | 640182 | 59260 | 513585 |
| 26575 | 79728 | 26590 | 79793 | 35596 | 557286 | 50857 | 440722 |
| 25648 | 76947 | 25681 | 77066 | 34222 | 535611 | 49105 | 425575 |
| 11258 | 33777 | 11289 | 33890 | 14757 | 231065 | 21519 | 186486 |
| 11256 | 33771 | 11288 | 33887 | 14763 | 231154 | 21516 | 186460 |

| Ni  | Nf  | Ni  | Nf  | Ni  | Nf  | Ni  | Nf  |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 34 | 105 | 74 | 281 | 32 | 317 | 58 | 367 |
| 50 | 153 | 78 | 501 | 34 | 333 | 79 | 495 |
| 31 | 96 | 66 | 205 | 28 | 279 | 57 | 379 |
| 40 | 123 | 58 | 188 | 36 | 352 | 66 | 437 |
| 35 | 108 | 70 | 213 | 34 | 336 | 63 | 452 |
| 38 | 117 | 62 | 189 | 26 | 259 | 66 | 455 |
| 36 | 111 | 70 | 213 | 33 | 325 | 66 | 476 |
| 36 | 111 | 70 | 213 | 31 | 307 | 66 | 476 |
### Table 4.2. Numerical results (CPU time)

| Dim. | S. P | P_1 | P_2 | P_3 | CPU time |
|------|------|-----|-----|-----|----------|
| 20000 | x_0 | 0.17850 | 2.46025 | 2.40625 | 3259.90625 |
| 20000 | x_1 | 0.15625 | 2.32812 | 2.32812 | 3249.78125 |
| 20000 | x_2 | 0.12500 | 0.20312 | 0.20312 | 3222.76562 |
| 20000 | x_3 | 0.12500 | 0.43750 | 0.43750 | 3222.76562 |
| 20000 | x_4 | 0.10937 | 0.06250 | 0.06250 | 3222.76562 |
| 20000 | x_5 | 0.140625 | 0.68750 | 0.68750 | 3222.76562 |
| 20000 | x_6 | 0.17187 | 0.71875 | 0.71875 | 3222.76562 |
| 20000 | x_7 | 0.23437 | 2.43750 | 2.43750 | 3222.76562 |
| 50000 | x_0 | 0.18750 | 2.46025 | 2.40625 | 4317.79687 |
| 50000 | x_1 | 0.15625 | 2.32812 | 2.32812 | 4317.79687 |
| 50000 | x_2 | 0.12500 | 0.20312 | 0.20312 | 4317.79687 |
| 50000 | x_3 | 0.12500 | 0.43750 | 0.43750 | 4317.79687 |
| 50000 | x_4 | 0.10937 | 0.06250 | 0.06250 | 4317.79687 |
| 50000 | x_5 | 0.140625 | 0.68750 | 0.68750 | 4317.79687 |
| 50000 | x_6 | 0.17187 | 0.71875 | 0.71875 | 4317.79687 |
| 50000 | x_7 | 0.23437 | 2.43750 | 2.43750 | 4317.79687 |

**New MPRP PRP SGPM**

**doi:10.1088/1742-6596/1804/1/012107**
To determine the performance of the new approach compared with the other three methods. Figures 1, 2 and 3 give the performance plots of these methods based on iteration numbers, function evaluations and CPU time respectively. We observe from fig 1, 2 and 3 that the proposed MOH method is a better competitive from the other methods.
Figure 1. Performance of iteration number

Figure 2. Performance of the function evaluations

Figure 3. Performance of the CPU time
5. Conclusion
In this work, a new projection approach for solving systems of nonlinear monotone equation is presented. This method is important and appropriate for a large scale systems of equations as well as optimization problem due to a consequence of low required memory of the conjugate gradient methods. The numerical results indicated that the new algorithm (MOH) is very efficient and effective to solve the nonlinear systems of monotone equations, and performance is better than some famous methods. We compare the results of the new method with three famous algorithm, and its result better than the others, that it needs less number of iterations, less number of function and less CPU time.

References
[1] M. A. K. Shiker and Z. Sahib, "A modified trust-region method for solving unconstrained optimization", Journal of Engineering and Applied Sciences, January, 13(22), p.9667-9671, (2018).
[2] A. M. Awwal, P. Kumam, A. B. Abubakar and A. Wakili, "A projection Hestenes-Stiefel like method for monotone nonlinear equations with convex constraints", Thai Journal of Mathematics, 16, p.181-199, (2018).
[3] Q. Li and D.H. Li, "A class of derivative-free methods for large-scale nonlinear monotone equations", IMA J. Numer. Anal 31, p.1625–1635, (2011).
[4] G. Yuan and M. Zhang, "A modified Hestenes-Stiefel conjugate gradient algorithm for large-scale optimization", Numerical Functional Analysis and Optimization, 34(8), p. 914–937, (2013).
[5] G. Zhou and K.C. Toh, "Superlinear convergence of a Newton-type algorithm for monotone equations", J. Optim. Theory Appl., 125, 205–221, (2005).
[6] M. V. Solodov and B. F. Svaiter, "A globally convergent inexact Newton method for systems of monotone equations", In: Fukushima, M., Qi, L. (eds.) Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, Kluwer Academic, 9, p. 355–369, (1998).
[7] W. J. Zhou and D. H. Li, "Limited memory BFGS method for nonlinear monotone equations", J. Comput. Math., 25, p. 89–96, (2007).
[8] Q. R. Yan, X. Z. Peng and D. H. Li, "A globally convergent derivative-free method for solving large-scale nonlinear monotone equations", J. Comput. Appl. Math., 234, p. 649–657, (2010).
[9] L. Zhang and W. J. Zhou, "Spectral gradient projection method for solving nonlinear monotone equations", J. Comput. Appl. Math., 196, p. 478–484, (2006).
[10] M. A. K. Shiker and K. Amini, "A new projection-based algorithm for solving a large scale nonlinear system of monotone equations", Croatian Operational Research Review, errr, 9, p. 63–73, (2018).
[11] M. Koorapetse, P. Kaelo and E. R. Offen, "A scaled derivative-free projection method for solving nonlinear monotone equations", Bulletin of The Iranian Mathematical Society, springer, 45, p.755-770, (2018).
[12] W. Cheng, "A PRP type method for systems of monotone equations", Math. Comput. Modell., 50, p. 15–20, (2009).