Solving a Large-Scale Nonlinear System of Monotone Equations by using a Projection Technique

Department of Mathematics, College for Pure Science, University of Babylon, Hillah, Iraq

Abstract: In this study, we suggest a new projection algorithm for solving nonlinear systems of monotone equations. The projection methods are an efficient family of derivative free methods for solving nonlinear systems of monotone equation that is in each iteration, the current iterate is strictly separated from the solution set of the problem by an appropriate hyperplane that constructs by the new projection algorithm. Then the current iterate is projected onto this hyperplane to determine the new approximation. Under standard assumptions, the global convergence of the proposed algorithm are proved. The numerical experiments indicate the efficiency of the proposed algorithm.

Key words: Nonlinear system of equations, projection method, monotone strategy, global convergence, experiments, assumptions

INTRODUCTION

The nonlinear systems and their solutions are of great importance in the various sciences which have included different fields and aspects. It is an important part of the sciences of mathematical and physics (since, most physical systems are nonlinear) as well as their importance in engineering, especially, mechanical engineering, electricity, management, economy, population growth, weather and other natural phenomena. The nonlinear systems are studied alongside the linear system because of the possibility of converting nonlinear problems into linear ones from many variables. Consider the following nonlinear system of equations:

\[ F(x) = 0 \]  

where, \( F \) is a continuous and monotone function from \( \mathbb{R}^n \) to \( \mathbb{R}^n \) condition of monotony mean:

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

The solution of nonlinear system equations is one of the problem and difficulties in the mathematical and engineering applications that analyzing it by analytical methods is difficult. Therefore, we can only rely on iterative methods that use the iterative procedure to obtain approximate solutions. The Newton method may be one of the best numerical methods that use the iterative method to solve these systems and it is consider a smooth method to find approximate values of equations.

In recent years, many modifications have been made to the Newton method these suggested methods may be equivalent to (or better than) the Newton method to solve the nonlinear system of equations. The line search method and the trust region method are the most important two methods to solve these systems.

The important idea of the line search method is finding the step length in the specified direction but the trust region method always cares to find a neighborhood of the current step \( x_k \), so that, the new iterate falls within the trust region determined by its radius (Amini et al., 2016) also this technique is used to solve unconstrained optimization (Shiker and Sahib, 2018).

Some methods proved ineffective for solving large-scale nonlinear system of equations as Newton method and quasi-Newton methods (Hager and Zhang, 2005; Li, 2017; Hassan and Shiker, 2018) because they need to solve the Jacobian matrix or an approximation of it in each iterative.

This research focuses on solving the system of large-scale non-linear equations by a new projection technique. The simple idea of the projection technique is always interested in separating the present approximation from the result set of the problem (Eq. 1) by appropriate hyperplane that is built in each iterate and then projecting this approximation on the same hyperplane to obtain the new approximation (Koorapetse et al., 2019). Several researchers use conjugate gradient approaches combining with projection techniques for solving (Eq. 1) as well as optimization issues (Dai, 2002).

Corresponding Author: Mushtak A.K. Shiker, Department of Mathematics, College for Pure Science, University of Babylon, Hillah, Iraq
The first projection approach was suggested by Solodov and Svaiter (1998) and it showed the totally convergent of solving nonlinear problems. In this study, the new algorithm is used to solve the nonlinear systems, we proved its global convergence. Then we compare with two famous methods, SBM method by Yan et al. (2010) and DFPB1 by Ahookhosh et al. (2013), the new algorithm will be more efficient.

**The framework:** The projection technique is one of the ways that proved to be active in solving nonlinear problems and it is a suitable and applicable way to solve large-scale difficulties these methods use a series of repetitions to arrive to the next iterate:

$$x_{k+1} = x_k + \alpha_k d_k$$

(3)

Where:
- $\alpha_k = \text{A step length}$
- $d_k = \text{The step direction}$

These processes are called an iterative procedures (Ortega and Rheinboldt, 1970), so, the projection techniques are called iterative methods. The projection approaches are family of derivative free. To define these effective methods, we use the projection operator $\Phi_{\Omega}[.]$.

$$\Phi_{\Omega}[x] = \arg\min\{\|x-z\|, z \in \Omega\}, \forall x \in \mathbb{R}^n$$

(4)

The projection operator has interesting features is non-expansive property:

$$\|\Phi_{\Omega}(x) - \Phi_{\Omega}(y)\| \leq \|x-y\|, \forall x, y \in \mathbb{R}^n$$

(5)

As a result produces:

$$\|\Phi_{\Omega}(x) - y\| \leq \|x-y\|, \forall x, y \in \Omega$$

(6)

After a series of iterations, in every iteration, the present approximation $x_k$ is isolated from the result set of the problem by the hyperplane $H_k$ that is construction by using a line search technique:

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

where:

$$z_k = x_k + \alpha_k d_k$$

(8)

By Solodov and Svaiter (1998) suggestion, the following iterate $x_{k+1}$ can be resolute by projection $z_k$ onto $H_k$ where:

$$C_k = \{x \in \mathbb{R}^r : F(z_k)^T(x-z_k) \leq 0\}$$

(9)

The approximation that is best among all result of system (Eq. 1) can be determined by projection $x_k$ onto $C_k$ but $x_k \not\in C_k$. Then the following approximation, $x_{k+1}$ can be determined by:

$$x_{k+1} = P_{C_k}(x_k) = \arg\min\{\|x-x_k\|, x \in C_k\}$$

So:

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

(10)

The suggested method, built on the projection free-derivatives method for the system of nonlinear equations, determine a direction $d_k$, a new direction has foreword as:

$$d_k = \begin{cases} -F(x_k) & \text{if } k = 0 \\ -\mu F(x_k) + \tau_k & \text{otherwise} \end{cases}$$

(11)

Where:

$$\mu_k = \frac{\sqrt{s_k}}{\gamma_k}, s_k = x_{k+1} - x_k, y_k = F(x_{k+1}) - F(x_k)$$

With:

$$\tau_k = \frac{F(x_{k+1})}{\|F(x_{k+1})\|}$$

(12)

Generally, used the direction $d_k$ which satisfies:

$$F^T d_k \leq -C \|F\|^2$$

(13)

where, $C$ is appositive constant (Ahookhosh et al., 2013). By Shiker and Amini (2018), introduced a new line search strategy for separating hyperplane in projection technique, encourage us to take advantages of this line search which needs $\alpha_k = \{0, 1, 2, \ldots\}$ satisfies the condition:

$$-F(x_k + \alpha_k d_k)^T d_k \geq \theta \alpha_k \|F(z_k)\|$$

(14)

where, $\lambda_k = \lambda / 1 + \|d_k\|^2$ and $\theta, \lambda$ are parameters. Our new algorithm will be state as below.
Algorithm 1 (NBM):
Input: An initial point \( x_0 \in \mathbb{R}^n \) and the parameters \( 0, \lambda, \varepsilon \in (0.2) \) and \( \beta \in (0.1) \).

Start
Set \( k = 0 \)
\( F_0 = F(x_0) \)
\( d_0 = -F_0 \)
While \( ||F_k|| > \varepsilon \)

Step 1: Compute \( ||F_k||. \) If \( ||F_k|| \leq \varepsilon \) stop
Set \( \alpha_k = \beta \)
Find the minimum index \( i_k \in \{1, 2, 3, \ldots\} \) such that
\[ -F(x_k + \alpha_k d_k)^T d_k \leq \lambda ||F(x_k)|| \]
where \( \lambda = \frac{\lambda}{1 + ||d_k||^2} \)
Set \( z_k = x_k + \alpha_k d_k \)
End while

Step 2: If \( ||F(z_k)|| \leq \varepsilon \), stop. Otherwise compute \( x_{k+1} \) by Eq. 10.

Step 3: Compute \( d_k \) by Eq. 11:
\( F_{k+1} = F(x_{k+1}) \)
If \( F_{k+1}^T d_k \geq -\varepsilon ||F_{k+1}|| \)
\( d_k = -F_{k+1} \)
End if
\( k = k + 1 \)
End while

End

Remark (R1): Shiker and Amini (2018) from stage 3 of algorithm 1, it is easy to note that the introduced direction satisfy the sufficient descent condition and for any \( k \), \( F_k^T d_k \leq -\varepsilon ||F_k||^2 \).

Convergence possessions: In this part, we need some interesting lemmas and assumptions in showing the global convergence of algorithm 1.

Assumption (B1): The result set of (Eq. 1) is nonempty.

Assumption (B2): The mapping \( F(x) \) is Lipschitz continuous on \( \mathbb{R}^n \) such that there exists a positive constant \( M \), i.e:
\[ ||F(x) - F(y)|| \leq M||x - y||, \forall x, y \in \mathbb{R}^n \]

Assumption (B3): The mapping \( F(x) \) is monotone on \( \mathbb{R}^n \) such that:
\[ (F(x) - F(y))^T (x - y) \geq 0, \forall x, y \in \mathbb{R}^n \]

Lemma (L1): Zarantonello (1971) let the set \( \Omega \subset \mathbb{R}^n \) be nonempty closed convex set and the projection operator \( \Phi_{\Omega}(x) \) be the projection of \( x \) onto closed convex set \( \Omega \). For any \( x, y \in \mathbb{R}^n \), the next statements hold:
- \( \forall x \in \Omega \), \( \langle \Phi_{\Omega}(x) - x, z - \Phi_{\Omega}(x) \rangle \geq 0 \)
- \( \langle \Phi_{\Omega}(x) - \Phi_{\Omega}(y), x - y \rangle \geq 0 \) and the inequality is strict when \( \Phi_{\Omega}(x) \neq \Phi_{\Omega}(y) \)
- \( ||\Phi_{\Omega}(x) - \Phi_{\Omega}(y)|| \leq ||x - y|| \)

Lemma (L2): Solodov and Svaiter (1998) assume the assumption B1, B2 and B3 hold and the sequence \( \{x_k\} \) is generated via. algorithm1. For any \( x^* \) such that \( F(x^*) = 0 \) then:
\[ ||x_{k+1} - x^*|| \leq ||x_k - x^*|| + ||x_k - x_{k+1}||^2 \] (15)

And the sequence \( \{x_k\} \) is bounded. Moreover, either the sequence \( \{x_k\} \) is finite although, the last iterate is a solution of (Eq. 1) or the sequence \( \{x_k\} \) is infinite and:
\[ \lim_{k \to \infty} ||x_{k+1} - x_k|| = 0 \] (16)

Proof: Let \( x^* \in R^n \) be any point such that \( F(x^*) = 0 \) by monotonicity of \( F(F(y), x^* - y) \leq 0 \). The hyperplane \( H = \{s \in R^n / (F(y), s - y) = 0\} \), separates \( x_k \) from \( x^* \), it is easy to satisfy that \( x_{k+1} \) is the projection of \( x_k \) onto the hyperplane \( H \). Since \( x^* \) belongs to this hyperplane from properties of the projection operator (Zarantonello, 1971) we get:
\[ ||x_k - x^*||^2 = ||x_k - z_k||^2 + ||z_k - x^*||^2 + 2\langle x_k - z_k, z_k - x^* \rangle \]
\[ \leq \left( \frac{F(y, x^*)}{||F(y)||} \right)^2 + ||x_{k+1} - x^*||^2 \]

Lemma (L3): Solodov and Svaiter (1998) assume that the assumption B1, B2 and B3 holds and the sequences \( \{x_k\} \) and \( \{z_k\} \) are generated by algorithm 1 then:
\[ \alpha_k \geq \min \left\{ \beta, \frac{\theta ||F(x)||}{M||d|| + ||F(z)||} \right\} \] (17)

Proof: By the line search rule (Eq. 14), if \( \alpha_k \cdot \beta \) then \( \alpha_k = \theta \alpha_k \) does not satisfy (Eq. 14) this mean that:
\[ -F(x_k + \theta \alpha_k d_k)^T d_k < 0. \theta \alpha_k \gamma_k ||F(z_k)|| \leq \lambda \alpha_k ||F(z_k)|| \]

where, \( \gamma_k = 1/1 + ||d_k||^2 \). By the Lipchitz continuity of \( F \) and (Eq. 12) we get:
\[ C||F||^2 \leq -F^T d_k = (F(z_k) - F(x_k))^T d_k - F(z_k)^T d_k \leq ||F(z_k) - F(x_k)|| ||d_k|| + \lambda \alpha_k ||F(z_k)|| = \alpha_k (M||d||^2 + \lambda ||F(z)||) \]
So:
\[ \alpha_k \geq \frac{\theta \beta ||F(x)||}{M||d||^2 + \lambda ||F(z)||} \]

The proof is complete and (Eq. 17) is correct. The results of Lemma L3 found that the line search of algorithm 1 is well defined.
**Theorem (T1):** Assume that $B_2$ and $B_3$ hold and the sequence $\{x_k\}$ is generated by algorithm 1 then:

$$\lim_{k \to \infty} \|F\| = 0$$

**(Proof):** From Eq. 10 and Eq. 14 we get:

$$\|z_{k+1} - z_k\|^2 \geq \frac{\|F(z_k)\|^2}{\|F(z_k)\|^2} = -\alpha_k \frac{\|d_k\|^2}{\|F(z_k)\|^2} \geq \frac{\lambda \alpha_k^2 \|F(z_k)\|^2}{(1 + \|d_k\|^2) \|F(z_k)\|^2} = \frac{2 \alpha_k^2}{1 + \|d_k\|^2}$$

By lemma 3 from Ahookhosh et al. (2013), the sequence of direction $\{d_k\}$ that generated by algorithm 1 are bounded there is a constant $N > 0$ such that:

$$\|F(x_k)\| \leq N$$

And result that for all $k$ there exists a constant $L > 0$ such that:

$$\|d_k\| \leq L$$

By the Lipschitz continuity of $F$, it can be concluded that:

$$\|F(z_k)\| \leq \|F(z_k) - F(x_k)\| + 1 \|F(x_k)\| \leq M \|z_k - x_k\| + N = M \|d_k\| + N$$

From (Eq. 19) together with (Eq. 21) gives:

$$\|x_{k+1} - x_k\|^2 \geq \frac{\lambda \alpha_k^2}{1 + L^2}$$

So:

$$\lim_{k \to \infty} \alpha_k ||d_k|| = 0$$

Now by using Cauchy Schwartz inequality along with (Eq. 12), we get:

$$C \|F_k\| \leq -k^2 \|d_k\| \leq \|F_k\| \|d_k\|$$

So:

$$\|d_k\| \geq C \|F_k\|$$

For all $k$. Giving to this condition and (Eq. 23), it follows that:

$$\lim_{k \to \infty} \alpha_k = 0$$

On the other hand, multiplying (Eq. 17) by $||d_k||^2$ result that:

$$\alpha_k ||d_k||^2 \geq \min \left\{ \beta ||d_k||^2, \frac{\theta C \lambda^2 \|F_k\|^2}{(M \|d_k\| + N)} \right\}$$

From (24) and (26) we have:

$$\lambda_0 ||F_k||^2 \leq \alpha_k ||d_k||^2$$

Where:

$$\lambda_0 = \min \left\{ \beta c^2, \frac{\theta C \lambda^2}{(M \|d_k\| + N)} \right\}$$

The relation (23) and (27) conclude that $\lim_{k \to \infty} ||F_k|| = 0$.

**Numerical experiment:** In this study, we compare the performance of the new algorithm (NBM) with two famous algorithms:

**SBM:** This technique is taken from Yan et al. (2010) and it uses two modified HS approaches with the projection technique by Solodov and Svaiter (1998).

**DFPB1:** This technique is taken from Ahookhosh et al. (2013), it uses a three-term PRP-based conjugate gradient direction.

The performances of these approaches are compared with reference to the number of iterations $N_i$, the number of function evaluations $N_f$ and CPU time. In order to compare these algorithms, some well-known test problems by Ahookhosh et al. (2013) and Yan et al. (2010) are used where the dimensions are confined between 5000-50000 for the taken primary points.

The tests were run on a PC with CPU 2.70 GHz and 4 GB RAM. All of the codes were written in MATLAB R2014 a programming environment. The running of the codes checks if the provided data for problems in all algorithms converges to the equal points. All of the algorithms terminate whenever $\|F_k\| < 10^{-8}$ or $\|F(z_k)\| < 10^{-4}$ or the whole number of iterates surpasses 500000. In all of the algorithms, the parameters are stated as follows $\theta = 0.4, \beta = 0.9, \lambda = 0.1, \epsilon = 10^{-4}$. The numerical results of consecutively the algorithms are registered in Table 1 and 2. Table 1 contains $N_i$ and $N_f$ while Table 2 contains the numerical results of CPU time.

From Table 1, we can see that the new approach NBM is better than the other two methods SBM and DFPB1 that it has a number of iterations and number of evaluation functions less than in the other methods in most of the problems with most of initial points. As well as the results in Table 2, we can see that the CPU time spent by the new technique NBM is lower than in the other two methods in most of problem that indicated the efficiency and quality of our new method.
Table 1: Numerical results (Ni and Nf)

<table>
<thead>
<tr>
<th>p-value/Dim</th>
<th>SP</th>
<th>Ni</th>
<th>Nf</th>
<th>Ni</th>
<th>Nf</th>
<th>Ni</th>
<th>Nf</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NBM</td>
<td>MHS</td>
<td>DFPB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50000</td>
<td>x1</td>
<td>11</td>
<td>91</td>
<td>1421</td>
<td>13288</td>
<td>1421</td>
<td>13288</td>
</tr>
<tr>
<td></td>
<td>x2</td>
<td>19</td>
<td>200</td>
<td>1421</td>
<td>13288</td>
<td>1421</td>
<td>13288</td>
</tr>
<tr>
<td></td>
<td>x3</td>
<td>13</td>
<td>110</td>
<td>142</td>
<td>880</td>
<td>142</td>
<td>880</td>
</tr>
<tr>
<td></td>
<td>x4</td>
<td>16</td>
<td>162</td>
<td>142</td>
<td>880</td>
<td>142</td>
<td>880</td>
</tr>
<tr>
<td></td>
<td>x5</td>
<td>18</td>
<td>281</td>
<td>5404</td>
<td>53012</td>
<td>9</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>x6</td>
<td>9</td>
<td>96</td>
<td>17</td>
<td>65</td>
<td>17</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>x7</td>
<td>25</td>
<td>188</td>
<td>4439</td>
<td>40365</td>
<td>88</td>
<td>504</td>
</tr>
<tr>
<td></td>
<td>x8</td>
<td>25</td>
<td>198</td>
<td>2252</td>
<td>19238</td>
<td>88</td>
<td>504</td>
</tr>
<tr>
<td>P2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50000</td>
<td>x1</td>
<td>11</td>
<td>91</td>
<td>1421</td>
<td>13288</td>
<td>1421</td>
<td>13288</td>
</tr>
<tr>
<td></td>
<td>x2</td>
<td>18</td>
<td>228</td>
<td>1371</td>
<td>12933</td>
<td>1421</td>
<td>13288</td>
</tr>
<tr>
<td></td>
<td>x3</td>
<td>13</td>
<td>110</td>
<td>142</td>
<td>880</td>
<td>142</td>
<td>880</td>
</tr>
<tr>
<td></td>
<td>x4</td>
<td>19</td>
<td>256</td>
<td>155</td>
<td>1086</td>
<td>142</td>
<td>880</td>
</tr>
<tr>
<td></td>
<td>x5</td>
<td>80</td>
<td>1120</td>
<td>3859</td>
<td>36651</td>
<td>9</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>x6</td>
<td>9</td>
<td>96</td>
<td>17</td>
<td>65</td>
<td>17</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>x7</td>
<td>56</td>
<td>666</td>
<td>7274</td>
<td>72430</td>
<td>88</td>
<td>504</td>
</tr>
<tr>
<td></td>
<td>x8</td>
<td>25</td>
<td>188</td>
<td>4439</td>
<td>40365</td>
<td>88</td>
<td>504</td>
</tr>
<tr>
<td>P3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10000</td>
<td>x1</td>
<td>16975</td>
<td>213353</td>
<td>418081</td>
<td>3099340</td>
<td>178397</td>
<td>721899</td>
</tr>
<tr>
<td></td>
<td>x2</td>
<td>67640</td>
<td>981540</td>
<td>415836</td>
<td>3005275</td>
<td>187559</td>
<td>759768</td>
</tr>
<tr>
<td></td>
<td>x3</td>
<td>14647</td>
<td>189321</td>
<td>390297</td>
<td>2977500</td>
<td>162500</td>
<td>656715</td>
</tr>
<tr>
<td></td>
<td>x4</td>
<td>55909</td>
<td>836601</td>
<td>402816</td>
<td>2960816</td>
<td>178197</td>
<td>720729</td>
</tr>
<tr>
<td></td>
<td>x5</td>
<td>47471</td>
<td>729546</td>
<td>371496</td>
<td>2767595</td>
<td>165523</td>
<td>668916</td>
</tr>
<tr>
<td></td>
<td>x6</td>
<td>54499</td>
<td>855259</td>
<td>368053</td>
<td>2766114</td>
<td>162292</td>
<td>655726</td>
</tr>
<tr>
<td></td>
<td>x7</td>
<td>22901</td>
<td>321922</td>
<td>163175</td>
<td>1263922</td>
<td>68435</td>
<td>276645</td>
</tr>
<tr>
<td></td>
<td>x8</td>
<td>22313</td>
<td>312113</td>
<td>153214</td>
<td>1138115</td>
<td>68436</td>
<td>276649</td>
</tr>
<tr>
<td>P4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10000</td>
<td>x1</td>
<td>24</td>
<td>255</td>
<td>20751</td>
<td>230201</td>
<td>469</td>
<td>4001</td>
</tr>
<tr>
<td></td>
<td>x2</td>
<td>370</td>
<td>5289</td>
<td>7709</td>
<td>73104</td>
<td>1494</td>
<td>14896</td>
</tr>
<tr>
<td></td>
<td>x3</td>
<td>18</td>
<td>183</td>
<td>2231</td>
<td>20120</td>
<td>155</td>
<td>1029</td>
</tr>
<tr>
<td></td>
<td>x4</td>
<td>365</td>
<td>5294</td>
<td>12747</td>
<td>135225</td>
<td>244</td>
<td>1713</td>
</tr>
<tr>
<td></td>
<td>x5</td>
<td>198</td>
<td>2896</td>
<td>27023</td>
<td>302729</td>
<td>85</td>
<td>395</td>
</tr>
<tr>
<td></td>
<td>x6</td>
<td>144</td>
<td>2092</td>
<td>11484</td>
<td>121948</td>
<td>76</td>
<td>314</td>
</tr>
<tr>
<td></td>
<td>x7</td>
<td>45</td>
<td>568</td>
<td>17308</td>
<td>191471</td>
<td>113</td>
<td>620</td>
</tr>
<tr>
<td></td>
<td>x8</td>
<td>27</td>
<td>291</td>
<td>2215</td>
<td>18879</td>
<td>113</td>
<td>620</td>
</tr>
<tr>
<td>P5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>x1</td>
<td>494</td>
<td>5834</td>
<td>154735</td>
<td>2156312</td>
<td>75926</td>
<td>1027957</td>
</tr>
<tr>
<td></td>
<td>x2</td>
<td>379</td>
<td>4285</td>
<td>147988</td>
<td>2085617</td>
<td>75401</td>
<td>1019862</td>
</tr>
<tr>
<td></td>
<td>x3</td>
<td>278</td>
<td>3260</td>
<td>153015</td>
<td>2148172</td>
<td>75883</td>
<td>1027901</td>
</tr>
<tr>
<td></td>
<td>x4</td>
<td>341</td>
<td>3944</td>
<td>149814</td>
<td>2109510</td>
<td>341</td>
<td>3944</td>
</tr>
<tr>
<td></td>
<td>x5</td>
<td>346</td>
<td>3797</td>
<td>158114</td>
<td>2198306</td>
<td>75839</td>
<td>1026614</td>
</tr>
<tr>
<td></td>
<td>x6</td>
<td>333</td>
<td>3718</td>
<td>156045</td>
<td>2174637</td>
<td>75843</td>
<td>1026679</td>
</tr>
<tr>
<td></td>
<td>x7</td>
<td>384</td>
<td>4216</td>
<td>146366</td>
<td>2075659</td>
<td>75872</td>
<td>1027109</td>
</tr>
<tr>
<td></td>
<td>x8</td>
<td>301</td>
<td>3459</td>
<td>152515</td>
<td>2142193</td>
<td>75850</td>
<td>1026796</td>
</tr>
<tr>
<td>P6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50000</td>
<td>x1</td>
<td>11</td>
<td>82</td>
<td>3353</td>
<td>28949</td>
<td>1009</td>
<td>8383</td>
</tr>
<tr>
<td></td>
<td>x2</td>
<td>18</td>
<td>210</td>
<td>8993</td>
<td>88244</td>
<td>1914</td>
<td>17404</td>
</tr>
<tr>
<td></td>
<td>x3</td>
<td>19</td>
<td>245</td>
<td>3932</td>
<td>35528</td>
<td>262</td>
<td>1813</td>
</tr>
<tr>
<td></td>
<td>x4</td>
<td>23</td>
<td>300</td>
<td>142</td>
<td>880</td>
<td>554</td>
<td>4288</td>
</tr>
<tr>
<td></td>
<td>x5</td>
<td>17</td>
<td>210</td>
<td>4450</td>
<td>40138</td>
<td>386</td>
<td>2805</td>
</tr>
<tr>
<td></td>
<td>x6</td>
<td>15</td>
<td>179</td>
<td>17</td>
<td>65</td>
<td>376</td>
<td>2725</td>
</tr>
<tr>
<td></td>
<td>x7</td>
<td>19</td>
<td>250</td>
<td>1615</td>
<td>12611</td>
<td>333</td>
<td>2381</td>
</tr>
<tr>
<td></td>
<td>x8</td>
<td>21</td>
<td>284</td>
<td>2057</td>
<td>16704</td>
<td>333</td>
<td>2381</td>
</tr>
<tr>
<td>P7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50000</td>
<td>x1</td>
<td>5</td>
<td>12</td>
<td>18560</td>
<td>45738</td>
<td>1421</td>
<td>13288</td>
</tr>
<tr>
<td></td>
<td>x2</td>
<td>22</td>
<td>280</td>
<td>1675</td>
<td>15191</td>
<td>1421</td>
<td>13288</td>
</tr>
<tr>
<td></td>
<td>x3</td>
<td>13</td>
<td>110</td>
<td>17325</td>
<td>34836</td>
<td>142</td>
<td>880</td>
</tr>
<tr>
<td></td>
<td>x4</td>
<td>19</td>
<td>256</td>
<td>158</td>
<td>1099</td>
<td>142</td>
<td>880</td>
</tr>
<tr>
<td></td>
<td>x5</td>
<td>12</td>
<td>138</td>
<td>993</td>
<td>1988</td>
<td>9</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>x6</td>
<td>4</td>
<td>16</td>
<td>16926</td>
<td>33854</td>
<td>17</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>x7</td>
<td>151</td>
<td>1728</td>
<td>17089</td>
<td>34332</td>
<td>88</td>
<td>504</td>
</tr>
<tr>
<td></td>
<td>x8</td>
<td>45</td>
<td>430</td>
<td>17089</td>
<td>34332</td>
<td>88</td>
<td>504</td>
</tr>
</tbody>
</table>
### Table 2: Numerical results (CPU time)

<table>
<thead>
<tr>
<th>p-value/Dim</th>
<th>SP</th>
<th>NBM</th>
<th>MHS</th>
<th>FPB</th>
</tr>
</thead>
<tbody>
<tr>
<td>P_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50000 x 1</td>
<td>0.15625</td>
<td>21.21875</td>
<td>20.62500</td>
<td></td>
</tr>
<tr>
<td>50000 x 2</td>
<td>0.26562</td>
<td>22.26562</td>
<td>20.90625</td>
<td></td>
</tr>
<tr>
<td>50000 x 3</td>
<td>0.140625</td>
<td>1.15625</td>
<td>1.06250</td>
<td></td>
</tr>
<tr>
<td>50000 x 4</td>
<td>0.15625</td>
<td>1.140625</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>50000 x 5</td>
<td>0.328125</td>
<td>55.01562</td>
<td>0.01562</td>
<td></td>
</tr>
<tr>
<td>50000 x 6</td>
<td>0.140625</td>
<td>0.18750</td>
<td>0.10937</td>
<td></td>
</tr>
<tr>
<td>50000 x 7</td>
<td>0.26562</td>
<td>43.78125</td>
<td>0.60937</td>
<td></td>
</tr>
<tr>
<td>P_2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50000 x 1</td>
<td>0.12500</td>
<td>21.50000</td>
<td>20.79687</td>
<td></td>
</tr>
<tr>
<td>50000 x 2</td>
<td>0.28125</td>
<td>21.09375</td>
<td>20.93750</td>
<td></td>
</tr>
<tr>
<td>50000 x 3</td>
<td>0.109375</td>
<td>1.12500</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>50000 x 4</td>
<td>0.31250</td>
<td>1.328125</td>
<td>1.04687</td>
<td></td>
</tr>
<tr>
<td>50000 x 5</td>
<td>1.109375</td>
<td>39.98437</td>
<td>0.10937</td>
<td></td>
</tr>
<tr>
<td>50000 x 6</td>
<td>0.140625</td>
<td>0.10937</td>
<td>0.12500</td>
<td></td>
</tr>
<tr>
<td>50000 x 7</td>
<td>0.62500</td>
<td>80.98437</td>
<td>0.53125</td>
<td></td>
</tr>
<tr>
<td>P_3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10000 x 1</td>
<td>0.42221</td>
<td>6.36439</td>
<td>1.47290</td>
<td></td>
</tr>
<tr>
<td>10000 x 2</td>
<td>1.95898</td>
<td>6.28664</td>
<td>1.57473</td>
<td></td>
</tr>
<tr>
<td>10000 x 3</td>
<td>0.37418</td>
<td>6.14095</td>
<td>1.33835</td>
<td></td>
</tr>
<tr>
<td>10000 x 4</td>
<td>1.66826</td>
<td>6.05081</td>
<td>1.48701</td>
<td></td>
</tr>
<tr>
<td>10000 x 5</td>
<td>1.47228</td>
<td>5.68717</td>
<td>1.36782</td>
<td></td>
</tr>
<tr>
<td>P_4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10000 x 1</td>
<td>0.09375</td>
<td>70.75000</td>
<td>1.20312</td>
<td></td>
</tr>
<tr>
<td>10000 x 2</td>
<td>1.34375</td>
<td>22.57812</td>
<td>4.18750</td>
<td></td>
</tr>
<tr>
<td>10000 x 3</td>
<td>0.03125</td>
<td>6.34375</td>
<td>0.31250</td>
<td></td>
</tr>
<tr>
<td>10000 x 4</td>
<td>1.37500</td>
<td>41.62500</td>
<td>0.50000</td>
<td></td>
</tr>
<tr>
<td>10000 x 5</td>
<td>0.81250</td>
<td>92.67187</td>
<td>0.12500</td>
<td></td>
</tr>
<tr>
<td>10000 x 6</td>
<td>0.31250</td>
<td>37.10937</td>
<td>0.09375</td>
<td></td>
</tr>
<tr>
<td>P_5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50000 x 1</td>
<td>0.98437</td>
<td>3.34671</td>
<td>1.55750</td>
<td></td>
</tr>
<tr>
<td>50000 x 2</td>
<td>0.68755</td>
<td>3.17734</td>
<td>1.54468</td>
<td></td>
</tr>
<tr>
<td>50000 x 3</td>
<td>0.51562</td>
<td>3.26009</td>
<td>1.56765</td>
<td></td>
</tr>
<tr>
<td>50000 x 4</td>
<td>0.53125</td>
<td>3.26873</td>
<td>0.00593</td>
<td></td>
</tr>
<tr>
<td>50000 x 5</td>
<td>0.59375</td>
<td>3.40265</td>
<td>1.56156</td>
<td></td>
</tr>
<tr>
<td>50000 x 6</td>
<td>0.60937</td>
<td>3.36593</td>
<td>1.56953</td>
<td></td>
</tr>
<tr>
<td>P_6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50000 x 1</td>
<td>0.21875</td>
<td>0.58468</td>
<td>16.76562</td>
<td></td>
</tr>
<tr>
<td>50000 x 2</td>
<td>0.43750</td>
<td>1.73859</td>
<td>32.28125</td>
<td></td>
</tr>
<tr>
<td>50000 x 3</td>
<td>0.53125</td>
<td>0.71421</td>
<td>3.25000</td>
<td></td>
</tr>
<tr>
<td>50000 x 4</td>
<td>0.60937</td>
<td>0.01062</td>
<td>8.23437</td>
<td></td>
</tr>
<tr>
<td>50000 x 5</td>
<td>0.59375</td>
<td>0.79390</td>
<td>5.43750</td>
<td></td>
</tr>
<tr>
<td>P_7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50000 x 1</td>
<td>0.01562</td>
<td>92.42187</td>
<td>20.92187</td>
<td></td>
</tr>
<tr>
<td>50000 x 2</td>
<td>0.34375</td>
<td>22.23437</td>
<td>21.37500</td>
<td></td>
</tr>
<tr>
<td>50000 x 3</td>
<td>0.09375</td>
<td>74.35937</td>
<td>0.92187</td>
<td></td>
</tr>
<tr>
<td>50000 x 4</td>
<td>0.20312</td>
<td>1.15625</td>
<td>1.09375</td>
<td></td>
</tr>
<tr>
<td>50000 x 5</td>
<td>0.93750</td>
<td>4.14062</td>
<td>0.03125</td>
<td></td>
</tr>
<tr>
<td>50000 x 6</td>
<td>0.03125</td>
<td>73.79687</td>
<td>0.12500</td>
<td></td>
</tr>
<tr>
<td>50000 x 7</td>
<td>1.65625</td>
<td>74.93750</td>
<td>0.53125</td>
<td></td>
</tr>
<tr>
<td>50000 x 8</td>
<td>0.40625</td>
<td>73.34375</td>
<td>0.60937</td>
<td></td>
</tr>
</tbody>
</table>

### CONCLUSION

The current research suggests a new projection technique for solving a system of large-scale nonlinear monotone equations. The projection-based algorithms belongs to the class of derivative-free function-value based approaches and it does not use any feature function and derivatives. Likewise, this method allows a simple globalization. The global convergence of the suggested algorithm is proved under standard assumptions. The numerical experiments indicated that the suggested algorithm is very efficient.

### REFERENCES


Amini, K., M.A. Shiker and M. Kimiaei, 2016. A line search trust-region algorithm with nonmonotone adaptive radius for a system of nonlinear equations. 4OR, 14: 133-152.


