symmetry

MDPI

Article

A Modified PRP-CG Type Derivative-Free Algorithm with Optimal Choices for Solving Large-Scale Nonlinear Symmetric Equations

Jamilu Sabi'u [1,2], Kanikar Muangchoo [3,*], Abdullah Shah [1], Auwal Bala Abubakar [4,5] and Lateef Olakunle Jolaoso [5]

[1] Department of Mathematics, COMSATS University Islamabad, Islamabad 12000, Pakistan; jsabiu@nwu.edu.ng (J.S.); abdullah_shah@comsats.edu.pk (A.S.)
[2] Department of Mathematics, Yusuf Maitama Sule University Kano, Kano 700241, Nigeria
[3] Department of Mathematics and Statistics, Faculty of Science and Technology, Rajamangala University of Technology Phra Nakhon (RMUTP), 1381, Pracharat 1 Road, Wongsawang, Bang Sue, Bangkok 10800, Thailand
[4] Department of Mathematical Sciences, Faculty of Physical Sciences, Bayero University, Kano 700241, Nigeria; ababubakar.mth@buk.edu.ng
[5] Department of Mathematics and Applied Mathematics, Sefako Makgatho Health Sciences University, Ga-Rankuwa, Pretoria, Medunsa 0204, South Africa; lateef.jolaoso@smu.ac.za
* Correspondence: kanikar.m@rmutp.ac.th

Abstract: Inspired by the large number of applications for symmetric nonlinear equations, this article will suggest two optimal choices for the modified Polak–Ribiére–Polyak (PRP) conjugate gradient (CG) method by minimizing the measure function of the search direction matrix and combining the proposed direction with the default Newton direction. In addition, the corresponding PRP parameters are incorporated with the Li and Fukushima approximate gradient to propose two robust CG-type algorithms for finding solutions for large-scale systems of symmetric nonlinear equations. We have also demonstrated the global convergence of the suggested algorithms using some classical assumptions. Finally, we demonstrated the numerical advantages of the proposed algorithms compared to some of the existing methods for nonlinear symmetric equations.

Keywords: measure function; Newton direction; approximate gradient; symmetric systems

MSC: 90C30; 90C26

Citation: Sabi'u, J.; Muangchoo, K.; Shah, A.; Abubakar, A.B.; Jolaoso, L.O. A Modified PRP-CG Type Derivative-Free Algorithm with Optimal Choices for Solving Large-Scale Nonlinear Symmetric Equations. *Symmetry* **2021**, *13*, 234. https://doi.org/10.3390/sym13020234

Academic Editor: Alina Alb Lupas
Received: 17 December 2020
Accepted: 29 December 2020
Published: 30 January 2021

1. Introduction

Consider the problem

$$G(x) = 0, \quad x \in R^n, \tag{1}$$

where $G : R^n \longrightarrow R^n$ is continuously differentiable mapping and its Jacobian is symmetric, i.e., $J(x) = J(x)^T$. Generally, the system of nonlinear equations has various applications in modern science and technology, some examples of nonlinear problems that are symmetrical in nature include "the discretized two-point boundary value problem, the gradient mapping of unconstrained optimization problem, the Karush-Kuhn-Tucker (KKT) of equality constrained optimization problems, the saddle point problem, and the discretized elliptic boundary value problem" see [1–3] for more details. There are different methods for solving symmetric nonlinear equations. Starting from the classic work of Li and Fukushima [1], where they proposed a Gauss-Newton-based Broyden-Fletcher-Goldfarb-Shanno (BFGS) method for the symmetric nonlinear equations. Subsequent improvements on the performance of [1] were discussed in [2–6] and the references therein. However, these methods are cost-effective in computing and storing the Jacobian approximate and therefore could not handle large-scale symmetric systems efficiently. These inspired researchers to develop

methods that could adequately deal with large-scale problems. The conjugate gradient (CG) method is an iterative scheme of the form:

$$x_0 \in R^n, \qquad x_{k+1} = x_k + t_k h_k, \quad k = 0, 1, \ldots, \tag{2}$$

where $t > 0$ is a stepsize to be determined by some line search techniques and h_k is the CG search direction defined by

$$h_0 = -G_0, \quad h_{k+1} = -G_{k+1} + \beta_k h_k, \quad k = 0, 1, \ldots, \tag{3}$$

$G_k = G(x_k)$, and β_k is the CG parameter [7]. Moreover, the efficiency of the CG method is based on the appropriate selection of the parameter β_k, see [7–9]. As dictated in the survey paper proposed by Hager and Zhang [7], one of the most effective CG parameters is the one suggested by Polak, Ribiére and Polyak (PRP) [8] with an outstanding restart feature given by

$$\beta_k^{PRP} = \frac{G_{k+1}^T y_k}{\|G_k\|}, \tag{4}$$

where $y_k = G_{k+1} - G_k$ and $\|.\|$ represents the Euclidean norm. However, despite the effectiveness and efficiency of the PRP method, the direction generated by the PRP method is not descent, i.e., it does not fulfill the condition

$$G_k^T h_k < 0, \quad \forall k \geq 0. \tag{5}$$

This prompted Zhang et al. [10] to propose the following modification of the PRP CG parameter given by

$$\beta_k^{DPRP} = \beta_k^{PRP} - \frac{G_{k+1}^T h_k}{\|G_k\|^2}. \tag{6}$$

The modified CG parameter (6) met the following sufficient descent condition

$$G_k^T h_k \leq -\gamma \|G_k\|^2, \quad \forall k \geq 0, \tag{7}$$

with γ as positive constant.

Furthermore, in the same vein as Zhang et al [10], Babaie-Kafaki and Ghanbari [11] have recently studied the following extension of the PRP method based on the Dai and Liao [12] strategies,

$$\beta_k^{EPRP} = \beta_k^{PRP} - \eta \frac{G_{k+1}^T h_k}{\|G_k\|^2}, \tag{8}$$

η is specified as a nonnegative constant. They considered the computation of η in (8) as an open problem in nonlinear CG methods. They also pointed out that their adaptive formula for selecting η for each iteration is better than a fixed selection [11].

Now, researchers have focused their attention on solving the symmetric systems of nonlinear equations using CG methods. Li and Wang [13] have suggested the modified Fletcher-Reeves-type (FR) CG method for solving symmetric equations. The method is a derivative and matrix-free method, therefore it could handle large dimensions of symmetric nonlinear equations efficiently. Besides, Zhou and Shen have suggested an inexact version of the PRP CG method for the systems of symmetric nonlinear equations [14]. Since then, different CG methods for the solution of symmetric nonlinear systems have been presented, see [15–23] for more details. Moreover, motivated by the efficiency of the extended PRP method [11] and the robustness of CG methods in solving large-scale symmetric nonlinear systems, we want to propose two optimal relations for the computation of η in (8) and use the corresponding modified PRP parameters to propose algorithms for solving large-scale symmetric nonlinear systems without using the exact gradient information.

The rest of this paper is organized as follows. In Section 2, we present the modified PRP CG-type algorithm for solving symmetric nonlinear equations. In Section 3, the proposed

algorithm is shown to converge globally. Section 4 provides computational experiments to demonstrate its practical performance. The paper is concluded in Section 5.

2. Modified PRP CG-Type with Optimal Choices

Li and Fukushima [1] observed that when the Jacobian $\nabla G(x_k)$ is symmetric, then the following relation holds:

$$G(x_k + t_k G_k) - G_k = t_k \int_0^1 \nabla G(x_k + r t_k G_k) G_k dr, \tag{9}$$

where t_k is an arbitrary scalar. They utilized (9) and approximated the gradient of the function G_k as

$$p_k = \frac{G(x_k + t_k G_k) - G_k}{t_k}, \tag{10}$$

and the step size t_k is to be determined as $t_k = \max\{1, a, a^2, \cdots\}$ such that

$$g(x_k + t_k h_k) - g(x_k) \leq -\zeta_1 ||t_k G(x_k)||^2 - \zeta_2 ||t_k h_k||^2 + \phi_k g(x_k), \tag{11}$$

where $\zeta_1 > 0, \zeta_2 > 0, a \in (0,1)$ are real constants and $\{\phi_k\}$ is a positive sequence such that

$$\sum_{k=0}^{\infty} \phi_k < \infty. \tag{12}$$

It is important to state that the function $g : R^n \longrightarrow R$ is a merit function defined by

$$g(x) = \frac{1}{2} \|G(x)\|^2. \tag{13}$$

Byrd and Nocedal [24] presented the following measure function given by

$$\varphi(Q_k) = Tr(Q_k) - \ln(\det(Q_k)) \tag{14}$$

where "Q_k" is a positive definite matrix, "$Tr(Q_k)$" is the trace of the matrix Q_k, "ln" denotes the natural logarithm, and "$\det(Q_k)$" is the determinant of Q_k. The function φ works with the trace and the determinant of the matrix Q_k and it involves all the eigenvalues of Q_k [24]. Now, using (8) and (11) we suggest the following modified PRP CG-type parameter

$$\beta_k^{MPRP} = \frac{p_{k+1}^T y_k}{\|p_k\|} - \eta_k^* \frac{p_{k+1}^T h_k}{\|p_k\|^2}, \tag{15}$$

where η_k^* is the optimal choice of η and $y_k = p_{k+1} - p_k$. In what follows, we are going to propose two optimal choices for the nonnegative constant η_k^* at every iteration.

2.1. The First Optimal Choice for η_k^*

This subsection will presents the first optimal choice by minimizing the major function of the proposed search direction matrix over η_k^*. Now, considering Perry's point of view [25], and Equations (3), (10) and (15), we write the search directions of the MPRP CG method as follows:

$$h_{k+1} = Q_{k+1} p_{k+1}, \quad k = 0, 1, \cdots, \tag{16}$$

with Q_{k+1} being the search direction matrix defined by

$$Q_{k+1} = I - \frac{h_k y_k^T}{\|p_k\|^2} + \eta_k^* \frac{h_k h_k^T}{\|p_k\|^2}. \tag{17}$$

The major function of the search directions matrix Q_{k+1} is defined by

$$\varphi(Q_{k+1}) = tr(Q_{k+1}) - \ln(\det(Q_{k+1})). \tag{18}$$

Since the matrix Q_{k+1} is a rank 2 update, its determinant is given by

$$\det(Q_{k+1}) = 1 - \frac{h_k^T y_k}{\|p_k\|^2} + \eta_k^* \frac{\|h_k\|^2}{\|p_k\|^2}, \tag{19}$$

and it trace as

$$tr(Q_{k+1}) = n - \frac{h_k^T y_k}{\|p_k\|^2} + \eta_k^* \frac{\|h_k\|^2}{\|p_k\|^2} \tag{20}$$

respectively. Now, using (19) and (20) the major fuction of Q_{k+1} is

$$\varphi(Q_{k+1}) = n - \frac{h_k^T y_k}{\|p_k\|^2} + \eta_k^* \frac{\|h_k\|^2}{\|p_k\|^2} - \ln\left(1 - \frac{h_k^T y_k}{\|p_k\|^2} + \eta_k^* \frac{\|h_k\|^2}{\|p_k\|^2}\right). \tag{21}$$

Differentiating (21) with respect to η_k^* to have

$$\frac{d\varphi(Q_{k+1})}{d\eta_k^*} = \frac{\|h_k\|^2}{\|p_k\|^2} - \frac{1}{1 - \frac{h_k^T y_k}{\|p_k\|^2} + \eta_k^* \frac{\|h_k\|^2}{\|p_k\|^2}} \frac{\|h_k\|^2}{\|p_k\|^2}. \tag{22}$$

Hence, by setting $\frac{d\varphi(Q_{k+1})}{d\eta_k^*} = 0$, and solving for η_k^* to get the first optimal choice for η_k^* as

$$\eta_k^{1*} = \frac{h_k^T y_k}{\|h_k\|^2}. \tag{23}$$

Therefore the modified PRP CG-type parameter (15) becomes

$$\beta_k^{MPRP} = \frac{p_{k+1}^T y_k}{\|p_k\|} - \eta_k^{1*} \frac{p_{k+1}^T h_k}{\|p_k\|^2}, \tag{24}$$

2.2. The Second Optimal Choice for η_k^*

Now, for the second optimal choice of the parameter η_k in (15). Recall that using the approximate gradient (10) the Newton direction is given by

$$h_{k+1} = -J_{k+1}^{-1} p_{k+1}. \tag{25}$$

Moreover, the search direction for the modified PRP CG-type method (15) can be written as

$$h_{k+1} = -p_{k+1} + \left(\frac{p_{k+1}^T y_k}{\|p_k\|^2} - \eta_k^* \frac{p_{k+1}^T h_k}{\|p_k\|^2}\right) h_k. \tag{26}$$

Newton direction is considered to be among the most robust schemes [7]. Therefore by Equating (25) with (26), we get

$$-J_{k+1}^{-1} p_{k+1} = -p_{k+1} + \left(\frac{p_{k+1}^T y_k}{\|p_k\|^2} - \eta_k^* \frac{p_{k+1}^T h_k}{\|p_k\|^2}\right) h_k. \tag{27}$$

Now, assuming that the Jacobian of G_k is symmetric and multiplying the both sides of (27) by $s_k^T J_{k+1}$ to obtain

$$-s_k^T p_{k+1} = -s_k^T J_{k+1} p_{k+1} + s_k^T J_{k+1} \left(\frac{p_{k+1}^T y_k}{\|p_k\|^2} - \eta_k^* \frac{p_{k+1}^T h_k}{\|p_k\|^2} \right) h_k. \tag{28}$$

Recall that, the secant equation is given by

$$y_k = J_{k+1} s_k. \tag{29}$$

From the fact that $J_{k+1} = J_{k+1}^T$ and the vitue of the secant equation, we rewrite (28) as

$$-s_k^T p_{k+1} = -y_k^T p_{k+1} + y_k^T \left(\frac{p_{k+1}^T y_k}{\|p_k\|^2} - \eta_k^* \frac{p_{k+1}^T h_k}{\|p_k\|^2} \right) h_k. \tag{30}$$

After simple algebraic simplifications we get the second optimal choice as

$$\eta_k^* = \frac{(s_k - y_k)^T p_{k+1} \|p_k\|^2}{p_{k+1}^T h_k y_k^T h_k} + \frac{p_{k+1}^T y_k}{p_{k+1}^T h_k}. \tag{31}$$

Now, for the prove of the global convergence of propose algorithm and promising numerical result, we selected our second optimal choice to be given as

$$\eta_k^{2*} = \min\left\{ 1, \frac{(s_k - y_k)^T p_{k+1} \|p_k\|^2}{p_{k+1}^T h_k y_k^T h_k} + \frac{p_{k+1}^T y_k}{p_{k+1}^T h_k} \right\}. \tag{32}$$

Therefore, the modified PRP CG-type parameter with second optimal choice becomes

$$\beta_k^{MPRP} = \frac{p_{k+1}^T y_k}{\|p_k\|^2} - \eta_k^{2*} \frac{p_{k+1}^T h_k}{\|p_k\|^2}. \tag{33}$$

Below is the modified PRP CG-type algorithm for solving large system of nonlinear symmetric equations.

3. Global Convergence

To start with the optimal choice η_k^{1*}, we define the level set by

$$\Psi = \{x | g(x) \leq \exp^{\phi} g(x_0)\}, \tag{34}$$

where ϕ satisfies (12).

Lemma 1. *Let* $\{x_k\}$ *be generated by the MPRP Algorithm 1. Then* $\{x_k\} \in \Psi$. *In addition,* $\{G_k\}$ *converges.*

Algorithm 1: MPRP Algorithm

step 0 Choose $x_0 \in R^n$, $t_0, \epsilon > 0$, $\zeta \in (0,1)$, $h_0 = -p_0$ and set $k = 0$.
step 1 Check if $\|G(x_k)\| \leq \epsilon$ is satisfied, else go to step 2.
step 2 Determine the stepsize t_k by using (11).
step 3 Compute $x_{k+1} = x_k + t_k h_k$
step 4 Determine the CG direction as

$$h_{k+1} = -p_{k+1} + \beta_k^{MPRP} h_k, \tag{35}$$

using any of the optimal choices in (23) or (32).
step 5 Set $k = k + 1$ and go back to step 1.

Proof. From (11), we have $\|G_{k+1}\| \leq (1+\phi)^{\frac{1}{2}}\|G_k\|$ for all k. As ϕ fulfills (12), we conclude that $\{\|G_k\|\}$ converges from Lemma 3.3 in [26]. In fact, we have for all k

$$\begin{aligned}
\|G_{k+1}\| &\leq (1+\phi_k)^{\frac{1}{2}}\|G_k\| \\
&\vdots \\
&\leq \prod_{i=0}^{k}(1+\phi_k)^{\frac{1}{2}}\|G_0\| \\
&\leq \|G_0\|\left[\frac{1}{k+1}\sum_{i=0}^{k}(1+\phi_i)\right]^{\frac{k+1}{2}} \\
&\leq \|G_0\|\left[1+\frac{1}{k+1}\sum_{i=0}^{k}\phi_i\right]^{\frac{k+1}{2}} \\
&\leq \|G_0\|\left(1+\frac{\phi}{k+1}\right)^{\frac{k+1}{2}} \\
&\leq \|G_0\|\left(1+\frac{\phi}{k+1}\right)^{k+1} \\
&\leq e^{\phi}\|G_0\|.
\end{aligned} \tag{36}$$

This means $\{x_k\} \in \Psi$. □

We make the following assumptions, which in the rest of this section will be frequently used.

Assumption 1.

(i) *The set Ψ is bounded.*

(ii) *The Jacobian is Lipschitz in some neighbourhood $x \in \Psi_1$ of Ψ. The positive constant M exists, such that*

$$\|J(x) - J(y)\| \leq M\|x-y\|, \quad \forall x, y \in \Psi_1. \tag{37}$$

Assumptions (i) and (ii) mean that there are constants $M > m > 0$ such that for all $x \in \Psi_1$

$$\|G(x)\| \leq C_1, \quad \|J(x)\| \leq C_2 \quad \forall x \in \Psi_1. \tag{38}$$

$$\|\nabla g(x) - \nabla g(y)\| \leq M_1\|x-y\|, \quad \forall x, y \in \Psi_1. \tag{39}$$

Lemma 2. *Let Assumption K holds. Then we have*

$$\lim_{k\to\infty}\|t_k h_k\| = \lim_{k\to\infty}||s_k|| = 0, \tag{40}$$

and

$$\lim_{k\to\infty}\|t_k G_k\| = 0. \tag{41}$$

Proof. This comes directly from (11) and (12). Now,

$$\zeta_1\|t_k G(x_k)\|^2 + \zeta_2\|t_k h_k\|^2 \leq g(x_k) - g(x_{k+1}) + \phi_k g(x_k), \tag{42}$$

when we sum up the k inequality above, then we get:

$$\sum_{i=0}^{m}\zeta_1\|t_k G(x_k)\|^2 + \zeta_2\|t_k h_k\|^2 \leq g(x_1) - g(x_m) + \sum_{i=0}^{m}\phi_i g(x_k). \tag{43}$$

So, from hypothesis K and that $\{\phi_k\}$ satisfies (12) the results will follow. □

The theorem below shows that MPRP algorithm converge globally.

Theorem 1. *Let Assumption K holds. Then the MPRP Algorithm 1 generated sequence* $\{x_k\}$ *converges globally, that is,*

$$\liminf_{k\to\infty}\|\nabla g(x_k)\| = 0. \tag{44}$$

Proof. We prove this theorem by contradiction. Suppose that (44) is not true, then there exists a positive constant α such that

$$\|\nabla g(x_k)\| \geq \alpha, \quad \forall k \geq 0. \tag{45}$$

Since $\nabla g(x_k) = J_k^T$, then (45) implies that there exists a positive constant α_1 satisfying

$$\|G_k\| \geq \alpha_1, \quad \forall k \geq 0. \tag{46}$$

CASE I: $\limsup_{k\to\infty} t_k > 0$, then by (41), we have $\limsup_{k\to\infty}\|G_k\| = 0$. This and Lemma 2 show that $\lim_{k\to\infty}\|G_k\| = 0$, which contradicts with (46).

CASE II: $\limsup_{k\to\infty} t_k = 0$. Since $t_k \geq 0$, this case implies that

$$\lim_{k\to\infty} t_k = 0. \tag{47}$$

By definition of p_k in (10) and the symmetry of the Jacobian, we have

$$\begin{aligned}\|p_k - \nabla g(x_k)\| &= \left\|\frac{G(x_k + t_k G_k) - G_k}{t_k}\right\| - J_k G_k \\ &= \left\|\oint_0^1 (J(x_k + s t_k G_k) - J_k) ds G_k\right\| \\ &\leq t_k M C_1^2,\end{aligned} \tag{48}$$

where we used the Lipschitz assumption on the Jacobian and the boundedness on G_k as well in the last inequality. However, (45) and (47) show that there exist a constant $\alpha_2 > 0$ such that

$$\|p_k\| \geq \alpha_2, \forall k \geq 0. \tag{49}$$

By considering (10) and the boundness of J_k and G_k

$$\|p_k\| = \left\|\int_0^1 J(x_k + s t_k G_k) G_k ds\right\| \leq M C_1, \quad \forall k \geq 0. \tag{50}$$

Now, from Equaton (48) and the Lipschitz assuption on $\nabla g(x)$

$$\begin{aligned}\|y_k\| &= \|p_{k+1} - p_k\| \\ &\leq \|p_{k+1} - \nabla g(x_{k+1})\| + \|p_k - \nabla g(x_k)\| + \|\nabla g(x_{k+1}) - \nabla g(x_k)\| \\ &\leq M C_1 (t_{k+1} - t_k) + M_1 \|s_k\|.\end{aligned} \tag{51}$$

This together with (40) and (47) show that $\lim_{k\to\infty} y_k = 0$. Then from (49) and (50) we get

$$\begin{aligned}\left|\beta_k^{MPRP}\right| &\leq \frac{\|p_{k+1}\|\|y_k\|}{\alpha_2^2} + \frac{\|h_k\|\|y_k\|}{\|h_k\|^2}\frac{\|p_{k+1}\|\|h_k\|}{\alpha_2^2} \\ &\leq 2\frac{M C_1}{\alpha_2^2}\|y_k\| \longrightarrow 0,\end{aligned} \tag{52}$$

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

$$\left|\beta_k^{MPRP}\right| \leq \tau. \tag{53}$$

We assume that the inequality (52) is true for all $k > 0$ without any loss of generality. Then from (3) and (50), we get

$$\begin{aligned} \|h_{k+1}\| &\leq \|p_{k+1}\| + |\beta_{k+1}|\|h_k\| \leq MC_1 + \tau\|h_k\|, \\ &\leq MC_1\left(1 + \tau + \tau^2 + \cdots + \tau^k\right) + \tau^k\|h_0\|, \\ &\leq \frac{MC_1}{1-\tau} + MC_1 = \frac{MC_1(2-\tau)}{1-\tau}, \end{aligned} \tag{54}$$

this shows that the sequence $\{h_{k+1}\}$ is bounded. Because of $\lim_{k\to\infty} t_k = 0$, then $t'_k = \frac{t_k}{a}$ is not satisfied (11), that is to say,

$$g(x_k + t'_k h_k) - g(x_k) > -\zeta_1\left\|t'_k G(x_k)\right\|^2 - \zeta_2\left\|t'_k h_k\right\|^2 + \phi_k g(x_k), \tag{55}$$

which means that

$$\frac{g(x_k + t'_k h_k) - g(x_k)}{t'_k} > -\zeta_1 t'_k\|G(x_k)\|^2 - \zeta_2 t'_k\|h_k\|^2. \tag{56}$$

By means of the mean value theorem, $\delta \in (0,1)$ exists in such a way as

$$\frac{g(x_k + t'_k h_k) - g(x_k)}{t'_k} = \nabla g(x_k + \delta t'_k h_k)^T h_k. \tag{57}$$

Because $x \in \Psi$ is bounded, We assume that $x_k \to x^*$ and have the following result using (48) and the boundedness of h_{k+1}

$$\lim_{k\to\infty} h_{k+1} = -\lim_{k\to\infty} p_{k+1} + \lim_{k\to\infty} \beta_{k+1} h_k = -\nabla g(x^*). \tag{58}$$

We have, on the other hand,

$$\lim_{k\to\infty} \nabla g(x_k + \delta t'_k h_k) = \nabla g(x^*). \tag{59}$$

Hence, using (58) and (59) in (57) we get $-\nabla g(x^*)^T \nabla g(x^*) \geq 0$. Which implies $\|\nabla g(x^*)\| = 0$. This is in contradiction with (45). This complete the proof of the theorem. □

4. Numerical Experiments

In this section, we compare the numerical performances of the modified PRP CG-type algorithm using the proposed optimal choice with the norm descent derivative-free algorithm (NDDA) [21] and the ICGM algorithm [22] for solving the nonlinear symmetric Equation (1). For the MPRP algorithm, we set: $\zeta = 10^{-4}$, $a = 0.4$, $t_0 = 0.01$ and $\phi_k = \frac{1}{(10^4+k)^2}$. While for the remaining two methods we adopted the same parameter as in their respective papers. The codes were written in Matlab R2014a and run on a personal computer with a 1.6 GHz CPU and 8 GB RAM. If the total number of iterations exceeds 1000 or $\|G(x_k)\| \leq 10^{-5}$, then the iteration is stopped. On the following eight test problems, we tested all three methods with different initial points and n values:

Problem 1 ([1]).

$$\begin{aligned} G(x_i) &= 4x_i + (x_{i+1} - 2x_i) - \tfrac{x_{i+1}^2}{3}, \text{for } i = 1, 2, \ldots, n-1. \\ G(x_n) &= 4x_n + (x_{n-1} - 2x_n) - \tfrac{x_{n-1}^2}{3}, \end{aligned}$$

Problem 2 ([1]).

$$G_1(x) = x_1(x_1^2 + x_2^2) - 1$$
$$G_i(x) = x_i(x_{i-1}^2 + 2x_i^2 + x_{i+1}^2) - 1, \quad i = 2, 3, \ldots, n-1,$$
$$G_n(x) = x_n(x_{n-1}^2 + x_n^2)$$

Problem 3 ([14]).

$$G_i(x) = x_i - (1 - \tfrac{c}{2n}\sum_{j=1}^{n} \tfrac{\mu_i x_j}{\mu_i + \mu_j})^{-1}, \textit{for } i = 1, 2, \ldots, n, \mu = \tfrac{i-0.5}{n}, c = 0.9$$

Problem 4 ([27]).

$$G(x_i) = 2x_i - \sin|x_i|, \textit{for } i = 1, 2, \ldots, n$$

Problem 5 ([27]).

$$G_1(x) = hx_1 + x_2 - 1,$$
$$G_i(x) = x_{i-1} + hx_i + x_{i-1} - 1, \textit{for } i = 2, 3, \ldots, n-1, h = 2.5$$
$$G_n(x) = x_{n-1} + hx_n - 1$$

Problem 6 ([27]).

$$G_1(x) = x_1 + \exp(\cos(hx_1 + x_2)),$$
$$G_i(x) = x_i + \exp(\cos(hx_{i-1} + x_i + x_{i+1})), \textit{for } i = 2, 3, \ldots, n-1, h = \tfrac{1}{n+1}$$
$$G_n(x) = x_n + \exp(\cos(hx_{n-1} + x_n))$$

Problem 7 ([15]).

$$G_1(x) = 2x_1 - x_2 + uh^2 \log(\cosh(x_1 - 1)) - 1,$$
$$G_i(x) = 2x_i - x_{i-1} + uh^2 \log(\cosh(x_i - 1)) - 1, \textit{for } i = 2, 3, \ldots, n-1, h = \tfrac{1}{n-1}$$
$$G_i(x) = 2x_n - x_{n-1} + uh \log(\cosh(x_n - 1)) - 1,$$

Problem 8 ([26]).

$$G(x) = \begin{pmatrix} \frac{5}{2} & 1 & & & \\ 1 & \frac{5}{2} & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & \frac{5}{2} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \\ x_n \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ \vdots \\ \\ 1 \end{pmatrix}$$

Tables 1–4 contained the numerical results of the three methods for the test problems with the eight different initial points namely; $x_1 = (1, 1, \ldots, 1)$, $x_2 = (1, \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{n})$, $x_3 = (0.1, 0.1, \ldots, 0.1)$, $x_4 = (\frac{1}{n}, \frac{2}{n}, \ldots, 1)$, $x_5 = (1 - \frac{1}{n}, 1 - \frac{2}{n}, \ldots, 0)$, $x_6 = (-1, -1, \ldots, -1)$, $x_7 = (n - \frac{1}{n}, n - \frac{2}{n}, \ldots, n - 1)$ and $x_8 = (\frac{1}{2}, 1, \frac{2}{3}, \ldots, \frac{2}{n})$.

In Tables 1–4, "ITER" indicates the number of iteration; "TIME" for the CPU time; "FEV" for the number of function evaluations, and "NORM" indicate the norm of the function at the stopping point. Table 1 contained Problems 1 and 2, although for the number of iterations MPRP with the optimal choice η_k^{2*} is the winner, followed by the NDDA methods and then the remaining two other algorithms. For the number of function evaluations and the CPU time, the proposed algorithms are also promising. The same observations can be made throughout the remaining tables concerning the number of iterations, CPU time, and the number of function evaluations.

Table 1. Numerical Comparisons of MPRP, NDDA [21] and ICGM [22].

PROBLEM 1																	
		MPRP(η_k^{1*})				**MPRP(η_k^{2*})**				**NDDA**				**ICGM**			
DIMENSION	**INITIAL POINT**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**
50,000	x_1	31	93	1.51757	7.72×10^{-6}	11	44	0.517181	5.84×10^{-6}	19	95	0.602169	8.39×10^{-6}	13	39	0.394323	6.33×10^{-6}
	x_2	25	75	1.182817	7.05×10^{-6}	22	88	0.825475	8.39×10^{-7}	21	105	0.641815	7.71×10^{-6}	18	54	0.531024	4.16×10^{-6}
	x_3	33	99	1.604626	8.78×10^{-6}	11	44	0.503097	4.82×10^{-6}	18	90	0.558197	7.96×10^{-6}	19	57	0.58265	4.37×10^{-6}
	x_4	39	117	1.894566	5.22×10^{-6}	30	120	1.348329	5.4×10^{-6}	20	100	0.608259	6.12×10^{-6}	27	81	0.830261	4.49×10^{-6}
	x_5	39	117	1.737297	5.31×10^{-6}	28	112	1.227482	5.55×10^{-6}	20	100	0.614197	8.51×10^{-6}	27	81	0.82587	6.28×10^{-6}
	x_6	43	129	1.95166	8.01×10^{-6}	19	76	0.938165	9.28×10^{-6}	21	105	0.643514	5.89×10^{-6}	21	63	0.663403	5.11×10^{-6}
	x_7	39	117	1.561943	5.31×10^{-6}	28	112	1.237306	5.55×10^{-6}	20	100	0.611944	8.51×10^{-6}	27	81	0.817997	6.28×10^{-6}
	x_8	24	72	0.866136	5.98×10^{-6}	17	68	0.578612	7.32×10^{-7}	25	125	0.744996	6.89×10^{-6}	26	78	0.801215	2.77×10^{-6}
100,000	x_1	33	99	2.503401	4.79×10^{-6}	11	44	1.060649	8.26×10^{-6}	20	100	1.261737	4.86×10^{-6}	13	39	0.824816	8.95×10^{-6}
	x_2	25	75	1.828535	7.05×10^{-6}	22	88	1.748833	8.39×10^{-7}	21	105	1.29851	7.71×10^{-6}	18	54	1.103288	5.3×10^{-6}
	x_3	35	105	2.51006	5.45×10^{-6}	11	44	1.062033	6.82×10^{-6}	19	95	1.200428	4.61×10^{-6}	19	57	1.206809	6.18×10^{-6}
	x_4	39	117	2.825586	7.37×10^{-6}	30	120	2.821417	7.3×10^{-6}	20	100	1.258583	8.6×10^{-6}	28	84	1.74041	6.85×10^{-6}
	x_5	39	117	2.623883	7.38×10^{-6}	28	112	2.570306	7.91×10^{-6}	20	100	1.236652	9.1×10^{-6}	29	87	1.816512	6.83×10^{-6}
	x_6	45	135	3.021735	4.97×10^{-6}	21	84	2.200632	1.13×10^{-6}	21	105	1.268213	8.33×10^{-6}	21	63	1.341347	7.22×10^{-6}
	x_7	39	117	2.603868	7.38×10^{-6}	28	112	2.621052	7.91×10^{-6}	20	100	1.17081	9.1×10^{-6}	29	87	1.804864	6.83×10^{-6}
	x_8	24	72	1.63477	5.98×10^{-6}	17	68	1.197102	7.63×10^{-7}	25	125	1.476767	6.89×10^{-6}	26	78	1.619349	2.77×10^{-6}
PROBLEM 2																	
		MPRP(η_k^{1*})				**MPRP(η_k^{2*})**				**NDDA**				**ICGM**			
DIMENSION	**INITIAL POINT**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**
50,000	x_1	170	510	7.931493	8.93×10^{-6}	150	600	4.737454	8.27×10^{-6}	382	1910	11.20393	9.75×10^{-6}	237	711	7.620697	9.98×10^{-6}
	x_2	141	423	5.058762	6.93×10^{-6}	269	1076	9.397728	9.33×10^{-6}	155	775	4.362288	8.76×10^{-6}	244	732	7.83738	9.88×10^{-6}
	x_3	102	306	2.842245	8.74×10^{-6}	137	548	4.299041	8.93×10^{-6}	384	1920	8.689407	9.91×10^{-6}	211	633	4.873246	9.22×10^{-6}
	x_4	324	972	9.531927	5.72×10^{-6}	309	1236	10.46505	9.73×10^{-6}	619	3095	12.28244	9.99×10^{-6}	1000	3000	22.33118	10.91535
	x_5	244	732	6.331992	5.68×10^{-6}	268	1072	7.62583	8.86×10^{-6}	455	2275	8.708046	9.02×10^{-6}	1000	3000	19.49895	10.68423
	x_6	67	201	1.95987	7.29×10^{-6}	122	488	3.443245	9.85×10^{-6}	352	1760	6.605092	9.02×10^{-6}	324	972	6.04279	9.73×10^{-6}
	x_7	276	828	9.354755	8.37×10^{-6}	278	1112	8.543894	8.97×10^{-6}	455	2275	9.163612	9.02×10^{-6}	1000	3000	21.8643	10.68423
	x_8	207	621	6.630383	7.9×10^{-6}	157	628	4.880471	6.37×10^{-6}	288	1440	6.438931	8.97×10^{-6}	166	498	3.764327	9.71×10^{-6}
100,000	x_1	152	456	9.113001	7.61×10^{-6}	148	592	8.525471	9.9×10^{-6}	383	1915	15.44497	9.74×10^{-6}	91	273	4.354032	9.74×10^{-6}
	x_2	163	489	10.66124	9.28×10^{-6}	223	892	15.55174	9.09×10^{-6}	155	775	6.307293	9.95×10^{-6}	399	1197	17.15205	9.74×10^{-6}
	x_3	77	231	3.928872	4.92×10^{-6}	127	508	6.798184	7.22×10^{-6}	388	1940	14.24754	9.96×10^{-6}	260	780	10.00556	7.72×10^{-6}
	x_4	317	951	16.27087	9.79×10^{-6}	292	1168	14.8396	6.16×10^{-6}	614	3070	22.54226	9.65×10^{-6}	1000	3000	38.68036	10.18736
	x_5	278	834	14.69331	9.92×10^{-6}	296	1184	15.86248	7.74×10^{-6}	547	2735	19.97045	8.39×10^{-6}	1000	3000	37.66586	13.37156
	x_6	114	342	5.378932	9.74×10^{-6}	110	440	5.0539	7.91×10^{-6}	354	1770	12.81065	8.79×10^{-6}	258	774	11.08156	7.29×10^{-6}
	x_7	274	822	13.97211	6.32×10^{-6}	275	1100	15.80533	8.28×10^{-6}	546	2730	19.73994	8.9×10^{-6}	1000	3000	37.19552	13.37156
	x_8	208	624	12.82466	8.67×10^{-6}	144	576	9.141726	6.49×10^{-6}	160	800	7.659241	9.25×10^{-6}	516	1548	21.02884	9.86×10^{-6}

Table 2. Numerical Comparisons of MPRP, NDDA [21] and ICGM [22].

PROBLEM 3																	
		MPRP(η_k^{1*})				**MPRP(η_k^{2*})**				**NDDA**				**ICGM**			
DIMENSION	**INITIAL POINT**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**
50,000	x_1	11	33	0.322182	5.17×10^{-6}	10	40	0.400361	5.96×10^{-6}	599	2995	12.23839	8.38×10^{-7}	13	39	0.241657	3.49×10^{-6}
	x_2	16	48	0.607104	7.34×10^{-6}	20	80	0.686698	8.82×10^{-6}	58	290	1.276429	8.28×10^{-6}	10	30	0.211414	6.37×10^{-6}
	x_3	11	33	0.320112	4.62×10^{-6}	15	60	0.505286	7.15×10^{-7}	495	2475	10.54336	9.96×10^{-6}	10	30	0.240468	9.4×10^{-6}
	x_4	9	27	0.292944	8.08×10^{-6}	6	24	0.372806		764	3820	13.48614	7.08×10^{-6}	14	42	0.410652	9.06×10^{-6}
	x_5	15	45	0.588793	3.6×10^{-6}	15	60	0.664088	3.97×10^{-6}	252	1260	4.319963	4.89×10^{-6}	12	36	0.377217	9.45×10^{-6}
	x_6	11	33	0.448889	5.17×10^{-6}	10	40	0.60674	5.96×10^{-6}	599	2995	11.48276	8.38×10^{-7}	13	39	0.437431	3.49×10^{-6}
	x_7	15	45	0.60361	3.6×10^{-6}	15	60	0.713857	3.97×10^{-6}	252	1260	3.986961	4.89×10^{-6}	12	36	0.399796	9.45×10^{-6}
	x_8	14	42	0.873261	9.37×10^{-6}	11	44	0.432592	8.11×10^{-6}	61	305	1.189075	5.05×10^{-6}	8	24	0.234699	3.02×10^{-6}
100,000	x_1	6	18	0.609567	6.69×10^{-6}	5	20	0.322736	1.14×10^{-6}	15	75	0.647448	5.11×10^{-6}	10	30	0.5706	1.66×10^{-6}
	x_2	12	36	1.599188	7.4×10^{-6}	11	44	0.893855	9.9×10^{-6}	76	380	3.81177	5.42×10^{-6}	8	24	0.481089	4.31×10^{-6}
	x_3	10	30	0.793137	8.9×10^{-6}	13	52	1.784466	9.06×10^{-6}	909	4545	28.60394	8.29×10^{-6}	11	33	0.773752	2.2×10^{-6}
	x_4	12	36	0.977956	6.37×10^{-6}	13	52	1.161575	4.14×10^{-6}	404	2020	14.25168	2.02×10^{-6}	12	36	0.857321	4.46×10^{-6}
	x_5	14	42	1.261941	9.15×10^{-6}	11	44	0.940573	7.49×10^{-6}	516	2580	15.59973	7.95×10^{-6}	12	36	0.842963	6.33×10^{-6}
	x_6	6	18	0.614141	6.69×10^{-6}	5	20	0.310198	1.14×10^{-6}	15	75	0.482139	2.5×10^{-6}	10	30	0.58923	1.66×10^{-6}
	x_7	14	42	1.242289	9.15×10^{-6}	11	44	0.93746	7.49×10^{-6}	516	2580	17.44822	7.95×10^{-6}	12	36	0.8428	6.33×10^{-6}
	x_8	13	39	1.664243	3.68×10^{-6}	13	52	1.209169	1.86×10^{-6}	75	375	2.555772	8.69×10^{-6}	8	24	0.49346	7.81×10^{-6}

Table 2. *Cont.*

PROBLEM 4																	
		MPRP(η_k^{1*})				MPRP(η_k^{2*})				NDDA				ICGM			
DIMENSION	INITIAL POINT	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
50,000	x_1	9	27	0.212201	9.28×10^{-6}	11	44	0.247058	4.22×10^{-6}	38	190	0.483188	8.05×10^{-6}	12	36	0.22417	3.58×10^{-6}
	x_2	14	42	0.321678	5.07×10^{-6}	11	44	0.263394	5.47×10^{-6}	27	135	0.398743	8.23×10^{-6}	22	66	0.394745	6.26×10^{-6}
	x_3	9	27	0.202813	1.19×10^{-6}	9	36	0.215252	1.46×10^{-6}	26	130	0.436246	8.17×10^{-6}	8	24	0.147041	6.88×10^{-6}
	x_4	23	69	0.593432	8.53×10^{-6}	21	84	0.50064	5.56×10^{-6}	39	195	0.690833	8.04×10^{-6}	36	108	0.63236	9.86×10^{-6}
	x_5	23	69	0.578756	4.6×10^{-6}	20	80	0.491674	3.58×10^{-7}	39	195	0.714213	8.04×10^{-6}	39	117	0.705334	9.49×10^{-6}
	x_6	17	51	0.45665	9.68×10^{-8}	19	76	0.595362	3.48×10^{-8}	37	185	0.682459	8.44×10^{-6}	12	36	0.219326	2.67×10^{-6}
	x_7	23	69	0.586717	4.6×10^{-6}	20	80	0.488278	3.58×10^{-7}	39	195	0.732293	8.04×10^{-6}	39	117	0.698136	9.52×10^{-6}
	x_8	18	54	0.47571	4.21×10^{-6}	32	128	0.906585	7.41×10^{-7}	28	140	0.51624	8.54×10^{-6}	27	81	0.479781	4.96×10^{-6}
100,000	x_1	11	33	0.476225	1.02×10^{-7}	11	44	0.484772	5.96×10^{-6}	39	195	1.445101	7.29×10^{-6}	12	36	0.438499	5.06×10^{-6}
	x_2	14	42	0.633407	5.06×10^{-6}	11	44	0.527568	5.47×10^{-6}	27	135	1.000728	8.23×10^{-6}	19	57	0.675121	9.94×10^{-6}
	x_3	9	27	0.406597	1.69×10^{-6}	9	36	0.42648	2.06×10^{-6}	27	135	0.974066	7.39×10^{-6}	8	24	0.291315	9.73×10^{-6}
	x_4	24	72	1.218071	8.4×10^{-6}	21	84	1.017136	1.52×10^{-6}	40	200	1.431425	7.28×10^{-6}	30	90	1.037708	2.42×10^{-6}
	x_5	23	69	1.171541	7.52×10^{-6}	20	80	0.960639	4.96×10^{-6}	40	200	1.427526	7.28×10^{-6}	43	129	1.466193	7.58×10^{-6}
	x_6	17	51	0.901294	1.37×10^{-7}	19	76	1.156228	4.92×10^{-8}	38	190	1.339707	7.63×10^{-6}	12	36	0.446339	3.78×10^{-6}
	x_7	23	69	1.171954	7.52×10^{-6}	20	80	0.942363	4.96×10^{-6}	40	200	1.419573	7.28×10^{-6}	43	129	1.472821	7.58×10^{-6}
	x_8	18	54	0.958594	4.21×10^{-6}	25	100	1.289564	8.32×10^{-6}	28	140	1.009042	8.54×10^{-6}	26	78	0.900203	8.57×10^{-6}

Table 3. Numerical Comparisons of MPRP, NDDA [21] and ICGM [22].

PROBLEM 5																	
		MPRP(η_k^{1*})				MPRP(η_k^{2*})				NDDA				ICGM			
DIMENSION	INITIAL POINT	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
50,000	x_1	69	207	2.267497	9.33×10^{-6}	105	420	3.891633	7.84×10^{-6}	191	955	4.947418	9.93×10^{-6}	135	405	3.611205	9.12×10^{-6}
	x_2	87	261	2.940586	8.23×10^{-6}	109	436	3.97732	8.31×10^{-6}	207	1035	5.307014	9.87×10^{-6}	220	660	5.912201	9.97×10^{-6}
	x_3	86	258	2.994068	9.81×10^{-6}	79	316	2.592411	9.4×10^{-6}	214	1070	5.430976	9.9×10^{-6}	998	2994	21.5748	9.96×10^{-6}
	x_4	79	237	2.599857	7.72×10^{-6}	101	404	3.660022	7.37×10^{-6}	205	1025	4.680842	9.23×10^{-6}	152	456	2.266204	9.89×10^{-6}
	x_5	80	240	2.659913	9.52×10^{-6}	111	444	3.90133	9.93×10^{-6}	205	1025	4.295369	9.23×10^{-6}	704	2112	12.14497	9.93×10^{-6}
	x_6	86	258	2.632166	9.8×10^{-6}	114	456	4.071767	9.07×10^{-6}	224	1120	4.554054	9.19×10^{-6}	1000	3000	15.05272	2.1×10^{-5}
	x_7	79	237	2.337165	8.46×10^{-6}	103	412	3.302599	9.43×10^{-6}	205	1025	3.89815	9.23×10^{-6}	719	2157	12.18638	9.86×10^{-6}
	x_8	86	258	2.501353	6.97×10^{-6}	122	488	3.74481	4.86×10^{-6}	216	1080	4.080285	9.5×10^{-6}	172	516	2.941251	7.99×10^{-6}
100,000	x_1	80	240	4.991337	6.4×10^{-6}	93	372	5.303749	8.04×10^{-6}	192	960	7.479403	9.02×10^{-6}	656	1968	21.91715	9.86×10^{-6}
	x_2	86	258	5.350999	9.84×10^{-6}	104	416	5.60915	9.53×10^{-6}	207	1035	7.35847	9.01×10^{-6}	713	2139	22.61809	9.95×10^{-6}
	x_3	80	240	4.52125	7.67×10^{-6}	94	376	4.548221	7.97×10^{-6}	212	1060	7.260601	9.45×10^{-6}	758	2274	25.39184	9.98×10^{-6}
	x_4	82	246	4.482321	8.53×10^{-6}	83	332	3.911853	9.59×10^{-6}	205	1025	8.032846	9.62×10^{-6}	197	591	6.705366	9.9×10^{-6}
	x_5	81	243	4.063491	8.48×10^{-6}	79	316	4.167463	8.7×10^{-6}	205	1025	7.737062	9.62×10^{-6}	173	519	6.832254	9.52×10^{-6}
	x_6	83	249	4.060558	6.42×10^{-6}	104	416	6.16264	8.65×10^{-6}	214	1070	7.483374	9.19×10^{-6}	790	2370	27.09478	9.86×10^{-6}
	x_7	82	246	4.173369	7.61×10^{-6}	85	340	4.378839	8.54×10^{-6}	205	1025	7.096066	9.62×10^{-6}	431	1293	14.45969	9.87×10^{-6}
	x_8	87	261	4.104345	9.47×10^{-6}	114	456	6.210763	9.51×10^{-6}	215	1075	7.667577	9.88×10^{-6}	150	450	4.905177	5.81×10^{-6}
PROBLEM 6																	
		MPRP(η_k^{1*})				MPRP(η_k^{2*})				NDDA				ICGM			
DIMENSION	INITIAL POINT	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
50,000	x_1	3	9	0.081907	1.46×10^{-6}	3	12	0.074954	1.46×10^{-6}	32	160	0.510629	6.43×10^{-6}	2	6	0.04543	2.36×10^{-6}
	x_2	1	3	0.033824	8×10^{-6}	1	4	0.032205	8.08×10^{-6}	1	5	0.032867	8×10^{-6}	1	3	0.031592	8×10^{-6}
	x_3	1	3	0.033946	8.57×10^{-6}	1	4	0.032505	8.64×10^{-6}	1	5	0.032653	8.57×10^{-6}	1	3	0.03293	8.57×10^{-6}
	x_4	1	3	0.034092	9.94×10^{-6}	1	4	0.035	9.99×10^{-6}	1	5	0.033783	9.94×10^{-6}	1	3	0.036741	9.94×10^{-6}
	x_5	1	3	0.037075	9.94×10^{-6}	1	4	0.037857	9.99×10^{-6}	1	5	0.037706	9.94×10^{-6}	1	3	0.037229	9.94×10^{-6}
	x_6	1	3	0.037861	1.3×10^{-6}	1	4	0.036916	1.16×10^{-6}	1	5	0.038647	1.3×10^{-6}	1	3	0.03646	1.3×10^{-6}
	x_7	1	3	0.037923	9.94×10^{-6}	1	4	0.039097	9.99×10^{-6}	1	5	0.038561	9.94×10^{-6}	1	3	0.038467	9.94×10^{-6}
	x_8	1	3	0.037427	8.01×10^{-6}	1	4	0.03926	8.08×10^{-6}	1	5	0.040607	8.01×10^{-6}	1	3	0.037036	8.01×10^{-6}
100,000	x_1	1	3	0.077721	3.79×10^{-6}	1	4	0.075644	3.8×10^{-6}	1	5	0.075443	3.79×10^{-6}	1	3	0.073561	3.79×10^{-6}
	x_2	1	3	0.079765	2.83×10^{-6}	1	4	0.076144	2.86×10^{-6}	1	5	0.077985	2.83×10^{-6}	1	3	0.076044	2.83×10^{-6}
	x_3	1	3	0.077061	3.03×10^{-6}	1	4	0.076729	3.05×10^{-6}	1	5	0.076138	3.03×10^{-6}	1	3	0.074692	3.03×10^{-6}
	x_4	1	3	0.081273	3.52×10^{-6}	1	4	0.075837	3.53×10^{-6}	1	5	0.075578	3.52×10^{-6}	1	3	0.078154	3.52×10^{-6}
	x_5	1	3	0.081851	3.52×10^{-6}	1	4	0.076603	3.53×10^{-6}	1	5	0.075728	3.52×10^{-6}	1	3	0.076166	3.52×10^{-6}
	x_6	1	3	0.077072	4.59×10^{-7}	1	4	0.076258	4.1×10^{-7}	1	5	0.075658	4.59×10^{-7}	1	3	0.075048	4.59×10^{-7}
	x_7	1	3	0.07455	3.52×10^{-6}	1	4	0.075514	3.53×10^{-6}	1	5	0.076933	3.52×10^{-6}	1	3	0.074581	3.52×10^{-6}
	x_8	1	3	0.087563	2.83×10^{-6}	1	4	0.077531	2.86×10^{-6}	1	5	0.074854	2.83×10^{-6}	1	3	0.074646	2.83×10^{-6}

Table 4. Numerical Comparisons of MPRP, NDDA [21] and ICGM [22].

PROBLEM 7																	
		MPRP(η_k^{1*})				**MPRP(η_k^{2*})**				**NDDA**				**ICGM**			
DIMENSION	**INITIAL POINT**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**
50,000	x_1	0	0	0.011313	0	0	0	0.01209	0	0	0	0.016223	0	0	0	0.012003	0
	x_2	87	261	2.348944	9.16×10^{-6}	79	316	2.273501	8.88×10^{-6}	154	770	4.291607	9.45×10^{-6}	1000	3000	26.32372	0.016563
	x_3	14	42	0.516938	7.44×10^{-6}	6	24	0.220502	4.02×10^{-6}	3	15	0.080141	4.41×10^{-6}	5	15	0.084522	6.44×10^{-6}
	x_4	34	102	0.926885	8.95×10^{-6}	33	132	0.980913	8.11×10^{-6}	41	205	1.188357	9.99×10^{-6}	1000	3000	27.79381	3.13×10^{-5}
	x_5	36	108	1.009782	8.7×10^{-6}	36	144	1.084175	7.95×10^{-6}	49	245	1.316967	9.56×10^{-6}	1000	3000	45.34329	3.2×10^{-5}
	x_6	18	54	0.660205	8.81×10^{-6}	8	32	0.29928	9.78×10^{-6}	4	20	0.114832	1.58×10^{-6}	7	21	0.221179	8.72×10^{-6}
	x_7	36	108	1.03182	8.7×10^{-6}	36	144	1.087501	7.95×10^{-6}	49	245	1.318538	9.56×10^{-6}	936	2808	42.23109	NaN
	x_8	90	270	2.548954	9.86×10^{-6}	88	352	2.584854	7.43×10^{-6}	163	815	4.589919	9.82×10^{-6}	1000	3000	29.78786	0.037358
100,000	x_1	0	0	0.016449	0	0	0	0.017713	0	0	0	0.019086	0	0	0	0.014114	0
	x_2	87	261	4.427758	8.58×10^{-6}	80	320	4.971012	9.72×10^{-6}	154	770	7.208364	9.4×10^{-6}	1000	3000	49.27	0.017351
	x_3	12	36	0.706042	6.35×10^{-6}	6	24	0.4356	1.65×10^{-6}	3	15	0.121382	2.2×10^{-6}	4	12	0.164811	7.8×10^{-6}
	x_4	26	78	1.188328	9.34×10^{-6}	30	120	1.787787	9.37×10^{-6}	34	170	1.501828	9.85×10^{-6}	925	2775	48.35269	NaN
	x_5	29	87	1.375369	9.64×10^{-6}	31	124	1.774428	9.55×10^{-6}	45	225	1.998629	9.4×10^{-6}	1000	3000	53.17642	2.14×10^{-5}
	x_6	16	48	1.08696	7.41×10^{-6}	6	24	0.420024	5.49×10^{-6}	3	15	0.144028	7.36×10^{-6}	6	18	0.221337	4.8×10^{-6}
	x_7	29	87	1.496431	9.64×10^{-6}	31	124	1.761076	9.55×10^{-6}	45	225	2.231545	9.4×10^{-6}	1000	3000	54.97467	1.99×10^{-5}
	x_8	92	276	4.563438	9.67×10^{-6}	85	340	4.913522	8.77×10^{-6}	163	815	7.319916	9.37×10^{-6}	1000	3000	48.95704	0.032224
PROBLEM 8																	
		MPRP(η_k^{1*})				**MPRP(η_k^{2*})**				**NDDA**				**ICGM**			
DIMENSION	**INITIAL POINT**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**
1000	x_1	82	246	0.455454	9.96×10^{-6}	105	420	0.597806	8.96×10^{-6}	224	1120	0.94912	9.83×10^{-6}	135	405	0.58535	9.26×10^{-6}
	x_2	88	264	0.59479	9.39×10^{-6}	98	392	0.576119	8.76×10^{-6}	228	1140	1.088754	9.84×10^{-6}	148	444	0.678534	9.94×10^{-6}
	x_3	76	228	0.447432	8.79×10^{-6}	107	428	0.625566	8.22×10^{-6}	215	1075	1.036161	9.92×10^{-6}	953	2859	4.435764	9.97×10^{-6}
	x_4	83	249	0.571694	8.7×10^{-6}	84	336	0.544969	7.1×10^{-6}	218	1090	1.264795	9.9×10^{-6}	952	2856	5.6264	9.97×10^{-6}
	x_5	74	222	0.714045	7.74×10^{-6}	100	400	0.991748	9.97×10^{-6}	218	1090	1.652656	9.9×10^{-6}	722	2166	5.116361	1×10^{-5}
	x_6	78	234	0.864799	9.45×10^{-6}	104	416	1.442508	6.02×10^{-6}	188	940	1.717357	1×10^{-5}	114	342	0.906444	8.28×10^{-6}
	x_7	74	222	0.872029	7.75×10^{-6}	93	372	1.332749	8.21×10^{-6}	218	1090	2.532429	9.9×10^{-6}	151	453	1.475109	7.62×10^{-6}
	x_8	90	270	1.006991	7.67×10^{-6}	117	468	1.869405	8.9×10^{-6}	236	1180	2.831446	9.84×10^{-6}	162	486	1.559799	8.39×10^{-6}
2000	x_1	82	246	1.001497	9.96×10^{-6}	91	364	1.851274	8.63×10^{-6}	223	1115	3.376192	9.27×10^{-6}	787	2361	10.66835	9.86×10^{-6}
	x_2	88	264	1.205998	9.39×10^{-6}	90	360	1.751596	9.71×10^{-6}	228	1140	3.306594	9.9×10^{-6}	750	2250	6.84631	1×10^{-5}
	x_3	76	228	1.125966	8.79×10^{-6}	92	368	1.842185	9.36×10^{-6}	214	1070	3.016198	9.96×10^{-6}	159	477	1.140194	9.44×10^{-6}
	x_4	83	249	1.167799	8.7×10^{-6}	144	576	3.194961	8.2×10^{-6}	219	1095	2.994653	9.73×10^{-6}	885	2655	8.149794	9.97×10^{-6}
	x_5	74	222	1.045681	7.74×10^{-6}	109	436	2.276981	9.94×10^{-6}	219	1095	2.892295	9.75×10^{-6}	396	1188	3.821894	9.98×10^{-6}
	x_6	78	234	1.174389	9.45×10^{-6}	82	328	1.611686	9.42×10^{-6}	190	950	2.546115	9.85×10^{-6}	108	324	1.188268	8.95×10^{-6}
	x_7	74	222	1.069987	7.75×10^{-6}	114	456	2.431602	9.96×10^{-6}	219	1095	2.956538	9.75×10^{-6}	682	2046	6.50311	9.86×10^{-6}
	x_8	90	270	1.224429	7.67×10^{-6}	117	468	2.359865	7.39×10^{-6}	235	1175	3.148771	9.56×10^{-6}	272	816	3.325016	1×10^{-5}

Moreover, to clearly show the performance of these algorithms, Figures 1–3 were plotted according to the data in Tables 1–4 using the Dolan and Moré performance profiles [28]. According to the Dolan and Moré performance profiles [28], the most efficient method is the one whose curve is at the top left of all curves. Therefore, We concluded from Figures 1–3, that the MPRP with the optimal choice η_k^{2*} is the most effective for the number of iterations, followed by the MPRP with optimal choice η_k^{1*}, NDDA algorithm, and lastly the ICGM algorithm. Similarly, the proposed algorithm remained the most stable algorithms concerning the number of CPU time and the number of function evaluations as their curves correspond to the top left curves.

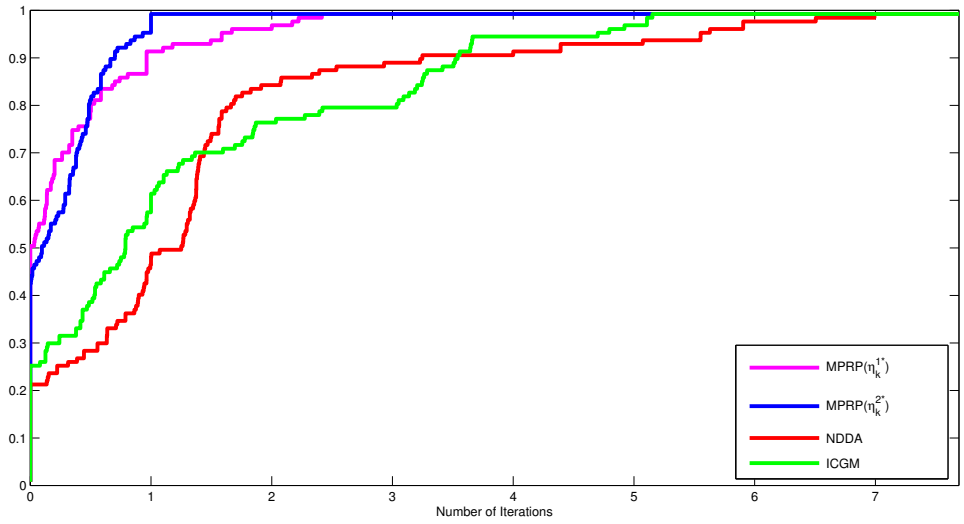

Figure 1. Performance profiles of MPRP, NDDA [21] and ICGM [22] for number of iterations.

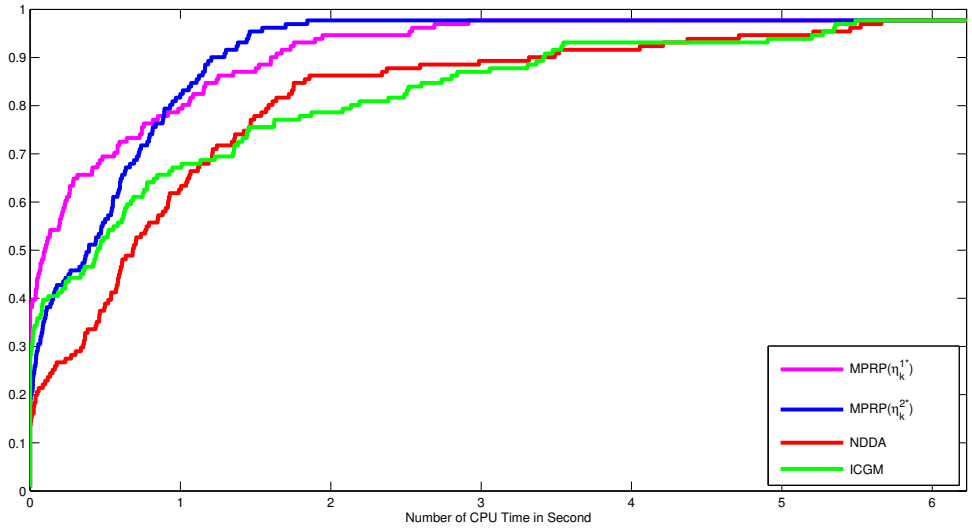

Figure 2. Performance profiles of MPRP, NDDA [21] and ICGM [22] for the CPU time in second.

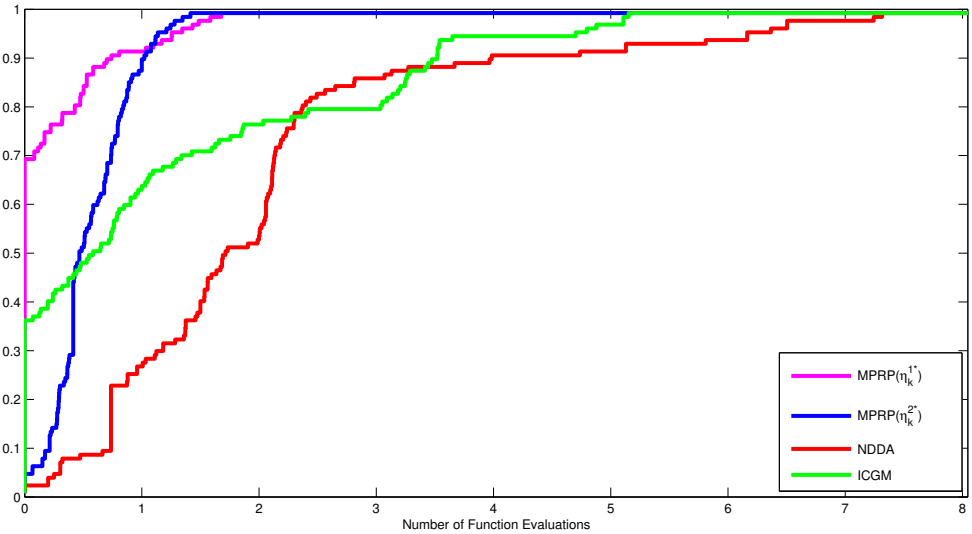

Figure 3. Performance profiles of MPRP, NDDA [21] and ICGM [22] for number of function evaluations.

5. Conclusions

In this paper, we provided a derivative-free PRP CG-type algorithm for solving the symmetric nonlinear equations and proved its global convergence by using the backtracking type line search. No information on the Jacobian matrix of G is used in the entire process of the proposed algorithm. The proposed algorithm is therefore appropriate for solving large-scale symmetric nonlinear systems. Computational outcomes also show that the proposed algorithm is robust and performs better than the NDDA [21] and ICGM [22] schemes for the symmetric nonlinear equations in number of iterations and the CPU time in seconds. This is because our algorithm makes full use of the optimal choice η_k^* at every iteration.

Author Contributions: Conceptualization, J.S.; methodology, J.S.; software, A.B.A.; validation, K.M. and A.S.; formal analysis, K.M. and L.O.J.; investigation, K.M. and A.B.A.; resources, K.M.; data curation, A.B.A. and A.S.; writing—original draft preparation, J.S.; writing—review and editing, A.B.A.; visualization, L.O.J.; supervision, K.M.; project administration, A.S.; funding acquisition, L.O.J. All authors have read and agreed to the published version of the manuscript.

Funding: The first author is grateful to TWAS/CUI for the Award of FR number: 3240299486.

Acknowledgments: The second author was financially supported by Rajamangala University of Technology Phra Nakhon (RMUTP) Research Scholarship.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, D.; Fukushima, M. A Globally and Superlinearly Convergent Gauss-Newton-Based BFGS Method for Symmetric Nonlinear Equations. *SIAM J. Numer. Anal.* **1999**, *37*, 152–172. [CrossRef]
2. Yuan, G.; Lu, X. A new backtracking inexact BFGS method for symmetric nonlinear equations. *Comput. Math. Appl.* **2008**, *55*, 116–129. [CrossRef]
3. Yuan, G.; Lu, X.; Wei, Z. BFGS trust-region method for symmetric nonlinear equations. *J. Comput. Appl. Math.* **2009**, *230*, 44–58. [CrossRef]
4. Gu, G.Z.; Li, D.H.; Qi, L.; Zhou, S.Z. Descent directions of quasi-Newton methods for symmetric nonlinear equations. *SIAM J. Numer. Anal.* **2002**, *40*, 1763–1774. [CrossRef]
5. Dauda, M.K.; Mamat, M.; Mohamad, F.S.; Magaji, A.S.; Waziri, M.Y. Derivative Free Conjugate Gradient Method via Broyden's Update for solving symmetric systems of nonlinear equations. *J. Phy. Conf. Ser.* **2019**, *1366*, 012099. [CrossRef]
6. Zhou, W. A modified BFGS type quasi-Newton method with line search for symmetric nonlinear equations problems. *J. Comput. Appl. Math.***2020**, *367*, 112454. [CrossRef]
7. Hager, W.W.; Zhang, H. A survey of nonlinear conjugate gradient methods. *Pac. J. Optim.* **2006**, *2*, 35–58.
8. Polak, E.; Ribiére, G. Note on the convergence of methods of conjugate directions (in French: Note sur la convergence de méthodes de directions conjuguées). *RIRO* **1969**, *3*, 35–43.
9. Andrei, N. A double parameter scaled BFGS method for unconstrained optimization. *J. Comput. Appl. Math.* **2018**, *332*, 26–44. [CrossRef]
10. Zhang, L.; Zhou, W.; Li, D.H. A descent modified Polak–Ribiére–Polyak conjugate gradient method and its global convergence. *IMA J. Numer. Anal.* **2006**, *26*, 629–640. [CrossRef]
11. Babaie-Kafaki, S.; Ghanbari, R. A descent extension of the Polak–Ribiére–Polyak conjugate gradient method. *Comput. Math. Appl.* **2014**, *68*, 2005–2011. [CrossRef]
12. Dai, Y.H.; Liao, L.Z. New conjugacy conditions and related nonlinear conjugate gradient methods. *Appl. Math. Optim.* **2001**, *43*, 87–101. [CrossRef]
13. Li, D.H.; Wang, X.L. A modified Fletcher-Reeves-type derivative-free method for symmetric nonlinear equations. *Numer. Algebra Control Optim.* **2011**, *1*, 71–82. [CrossRef]
14. Zhou, W.; Shen, D. An inexact PRP conjugate gradient method for symmetric nonlinear equations. *Numer. Func. Anal. Opt.* **2014**, *35*, 370–388. [CrossRef]
15. Zhou, W.; Shen, D. Convergence properties of an iterative method for solving symmetric non-linear equations. *J. Optim. Theory Appl.* **2015**, *164*, 277–289. [CrossRef]
16. Waziri, M.Y.; Sabi'u, J. A derivative-free conjugate gradient method and its global convergence for solving symmetric nonlinear equations. *Int. J. Math. Math. Sci.* **2015**, *2015*, 1–8. [CrossRef]
17. Zou, B.; Zhang, L. A nonmonotone inexact MFR method for solving symmetric nonlinear equations. *AMO* **2016**, *18*, 87–95.
18. Sabi'u, J. Enhanced derivative-free conjugate gradient method for solving symmetric nonlinear equations. *Int. J. Adv. Appl. Sci.* **2016**, *5*, 50–57. [CrossRef]

19. Waziri, M.Y.; Sabi'u, J. An alternative conjugate gradient approach for large-scale symmetric nonlinear equations. *J. Math. Comput. Sci.* **2016**, *6*, 855–874.
20. Sabi'u, J. Effective algorithm for solving symmetric nonlinear equations. *J. Contemp. Appl. Math.* **2017**, *7*, 157–164.
21. Liu, J.K.; Feng, Y.M. A norm descent derivative-free algorithm for solving large-scale nonlinear symmetric equations. *J. Comput. Appl. Math.* **2018**, *344*, 89–99. [CrossRef]
22. Abubakar, A.B.; Kumam, P.; Awwal, A.M. An inexact conjugate gradient method for symmetric nonlinear equations. *Comput. Math. Methods* **2019**, *1*, e1065. [CrossRef]
23. Dauda, M.K.; Mustafa, M.; Mohamad, A.; Nor, S.A. Hybrid conjugate gradient parameter for solving symmetric systems of nonlinear equations. *Indones. J. Electr. Eng. Comput. Sci. (IJEECS)* **2019**, *16*, 539–543. [CrossRef]
24. Byrd, R.H.; Nocedal, J. A tool for the analysis of quasi-Newton methods with application to unconstrained minimization. *SIAM J. Numer. Anal.* **1989**, *26*, 727–739. [CrossRef]
25. Perry, A. A modified conjugate gradient algorithm. *Oper. Res.* **1976**, *26*, 1073–1078. [CrossRef]
26. Dennis, E.J.; Moré, J.J. A characterization of superlinear convergence and its application to quasi-Newton methods. *Math. Comput.* **1974**, *28*, 549–560. [CrossRef]
27. La Cruz, W.; Martínez, J.; Raydan, M.W. Spectral residual method without gradient information for solving large-scale nonlinear systems of equations. *Math. Comput.* **2006**, *75*, 1429–1448. [CrossRef]
28. Dennis, E.J.; Moré, J.J. Benchmarking optimization software with performance profiles. *Math. Program.* **2002**, *9*, 201–213.