symmetry

MDPI

Article

A Two-Step Spectral Gradient Projection Method for System of Nonlinear Monotone Equations and Image Deblurring Problems

Aliyu Muhammed Awwal [1,2], Lin Wang [3,*], Poom Kumam [1,4,*] and Hassan Mohammad [5]

[1] KMUTTFixed Point Research Laboratory, Room SCL 802 Fixed Point Laboratory, Science Laboratory Building, Department of Mathematics, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), 126 Pracha-Uthit Road, Bang Mod, Thung Khru, Bangkok 10140, Thailand; maaliyu@gsu.edu.ng

[2] Department of Mathematics, Faculty of Science, Gombe State University, Gombe 760214, Nigeria

[3] Office of Science and Research, Yunnan University of Finance and Economics, Kunming 650221, China

[4] Department of Medical Research, China Medical University Hospital, China Medical University, Taichung 40402, Taiwan

[5] Department of Mathematical Sciences, Faculty of Physical Sciences, Bayero University, Kano 700241, Nigeria; hmuhd.mth@buk.edu.ng

* Correspondence: bgs@ynufe.edu.cn (L.W.); poom.kum@kmutt.ac.th (P.K.)

Received: 23 March 2020; Accepted: 16 April 2020; Published: 26 May 2020

Abstract: In this paper, we propose a two-step iterative algorithm based on projection technique for solving system of monotone nonlinear equations with convex constraints. The proposed two-step algorithm uses two search directions which are defined using the well-known Barzilai and Borwein (BB) spectral parameters.The BB spectral parameters can be viewed as the approximations of Jacobians with scalar multiple of identity matrices. If the Jacobians are close to symmetric matrices with clustered eigenvalues then the BB parameters are expected to behave nicely. We present a new line search technique for generating the separating hyperplane projection step of Solodov and Svaiter (1998) that generalizes the one used in most of the existing literature. We establish the convergence result of the algorithm under some suitable assumptions. Preliminary numerical experiments demonstrate the efficiency and computational advantage of the algorithm over some existing algorithms designed for solving similar problems. Finally, we apply the proposed algorithm to solve image deblurring problem.

Keywords: spectral gradient method; nonlinear monotone equations; projection method; line search; image deblurring

MSC: 65K05; 90C30; 90C06; 90C56; 92C55

1. Introduction

Many problems arising from various applications such as optimization, differential equations, variational inequalities problems and so on, can be converted into nonlinear system of equations. Hence the study of iterative algorithms for solving nonlinear equations is of paramount importance especially when analytical method is not feasible or difficult to implement.

Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a monotone mapping and Λ be a subset of $\mathbb{R}^n$. We wish to find a point $x^* \in \Lambda$, such that

$$F(x^*) = 0. \tag{1}$$

The feasible set Λ is assumed to nonempty closed and convex. We call problem (1) system of nonlinear monotone equations with convex constraints. This problem appears as a subproblem in generalized proximal algorithms with Bregman distance [1]. In addition, some monotone variational inequality problems of finding $y \in C$ for which $\langle x - y,\ F(y) \rangle \geq 0, \ \ \forall\ x \in C$ can be converted into systems of monotone equations [2]. Furthermore, l_1- norm regularized optimization problems can be reformulated as monotone nonlinear equations [3].

Consider the following unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{2}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is assumed to be continuous, bounded below and its gradient, denoted by F, exists. Fermat's extremum theorem suggests that if a point x^* is the local minimizer of the unconstrained optimization problem (2) then problem (1) holds. In addition, suppose x^* is the minimizer of problem (2), then problem (1) is the first order necessary condition for the unconstrained optimization problem (2). This also underlines the importance of problem (1).

Starting from a given initial point $x_0 \in \mathbb{R}^n$, popular iterative methods, such as Newton's method, quasi-Newton method, conjugate gradient method, for solving (2) use an updating rule defined as follows

$$x_{k+1} = x_k + \alpha_k d_k, \ \ k = 0, 1, 2, ..., \tag{3}$$

where α_k and d_k denote stepsize and search direction respectively.

The search direction in (3) is usually defined as $d_k = -B_k^{-1} F(x_k)$ where B_k is either the exact Hessian matrix $\nabla^2 f(x_k)$ in the case of Newton's method or the approximation of the Hessian matrix in the case of quasi-Newton method. The approximation of the Hessian matrix, B_k, is required to satisfy the following secant equation

$$B_k s_{k-1} = y_{k-1}, \tag{4}$$

$s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = F(x_k) - F(x_{k-1})$.

The Quasi-Newton method was developed to overcome one of the major shortcomings associated with the famous Newton's method which is the need to compute second derivative of the objective function in every iteration. However, it inherits the problem of storing $n \times n$ matrices throughout the iteration process which makes it unsuitable for large scale problems. One of the crucial approaches developed to overcome the storage problem of the quasi Newton method is the matrix-free method proposed by Barzilai and Borwein (BB) [4]. The BB method uses (3) to generate the next iterate with the search direction given by $d_k = -F(x_k)$ and the stepsize taken as diagonal matrix $D_k = \tau_k I$ which is supposed to satisfy the secant Equation (4). However, since $\tau_k I$ produces diagonal matrices with identical diagonal elements, it is usually very difficult to find τ_k for which $D_k^{-1} = \tau_k^{-1} I$ satisfies (4) when the dimension is greater than one. Consequently, Barzilai and Borwein required D_k^{-1} to approximately satisfies (4) by finding $\tau_k \in \mathbb{R}$ which minimizes the following least square problems

$$\min_{\tau} \|\tau s_{k-1} - y_{k-1}\|^2, \tag{5}$$

and

$$\min_{\tau} \|s_{k-1} - \tau y_{k-1}\|^2. \tag{6}$$

The solutions of the minimization problems (5) and (6) are respectively given as

$$\tau_k^{BB1} = \frac{\|s_{k-1}\|^2}{\langle y_{k-1},\ s_{k-1} \rangle} \quad \text{and} \quad \tau_k^{BB2} = \frac{\langle y_{k-1},\ s_{k-1} \rangle}{\|y_{k-1}\|^2}. \tag{7}$$

By Cauchy Schwarz inequality, we see that the stepsize produced by τ_k^{BB1} is always greater than or equal to the one produced by τ_k^{BB2} whenever $\langle y_{k-1},\ s_{k-1} \rangle > 0$. Barzilai and Borwein proved that the iterative scheme (3) with $\alpha_k = \tau_k^{BB1}$ and $d_k = -F(x_k)$ converges with R-superlinear rate for two-dimensional strictly convex quadratic problems.

One disadvantage of the BB method, however, is that the stepsizes τ_k^{BB1} and τ_k^{BB2} may become negative if the objective function is not convex. Thus, Dai et al. [5] proposed and analyzed the following positive stepsize

$$\widehat{\tau}_k = \frac{\|s_{k-1}\|}{\|y_{k-1}\|}. \tag{8}$$

The stepsize (8) is the geometric mean of τ_k^{BB1} and τ_k^{BB2}. They showed that the iterative scheme (3) with $\alpha_k = \widehat{\tau}_k$ has the same rate of convergence with the stepsize τ_k^{BB1} under certain conditions for two-dimensional strictly convex quadratic functions. Recently, Dai et al. [6] proposed a family of gradient methods whose stepsize is a convex combination of τ_k^{BB1} and τ_k^{BB2}. The stepsize is obtained by solving the following problem

$$\Psi_{\xi}(\tau) = \|\xi[(1/\tau)s_{k-1} - y_{k-1}] + (1-\xi)[s_{k-1} - \tau y_{k-1}]\|^2. \tag{9}$$

It was shown that if $0 \leq \xi \leq 1$ and $\langle y_{k-1}, s_{k-1}\rangle > 0$, then $\frac{d\Psi_{\xi}(\tau)}{d\tau} = 0$, has a unique solution in the closed interval $[\tau_k^{BB1}, \tau_k^{BB2}]$. They proved that their method is R-superlinearly and R-linearly convergent for two- dimensional strictly convex quadratics and any finite dimensional case respectively. Convergence analysis of the BB stepsizes has been explored and interested reader may refer to the following References [7–12].

On the other hand, the BB method with the stepsize τ_k^{BB1} has been extended to solve unconstrained nonlinear equations by La Cruz and Raydan [13]. Their algorithm is built on the strategy of nonmonotone line search technique which guarantees the global convergence of the method. Numerical experiments presented reveal their method competes with some well-established existing methods. However, their algorithm requires descent directions with respect to the squared norm of the residual. This means computation of a directional derivative, or its good approximation is needed at every iteration. Consequently, La Cruz et al. [14] proposed another BB method with a different nonmonotone line-search technique for solving unconstrained nonlinear equations. Their approach has advantage because unlike the former, the computations of directional derivatives are completely avoided. Based on the projection technique of Solodov and Svaiter [15], Zhang and Zhou [16] proposed an interesting projection spectral method which can be viewed as a modification of the method given in References [13,14]. They proposed a new line search strategy which does not require any merit function and takes the monotonicity of F into account. They established the global convergence of the method under some suitable assumptions and present some numerical experiments to demonstrate its computational advantage. In Reference [17], Yu et al. extended the method given by Zhang and Zhou [16] to solve monotone system of nonlinear equations with convex constraints. Their method is globally convergent under some conditions and preliminary numerical results show that the method works well and is more suitable compared to the projection method in Reference [18]. Recently, Mohammad and Abubakar [19] proposed a positive spectral method for unconstrained monotone nonlinear equations based on the projection technique in Reference [15]. The spectral parameter proposed is a convex combination of modified τ_k^{BB1} and $\widehat{\tau}_k$. Their method works well and was extended to solve monotone nonlinear equations with convex constraints in Reference [20] as well as signal and image restoration in Reference [21].

Inspired by above contributions, we propose a two step iterative scheme based on the projection technique for solving system of monotone nonlinear equations with convex constraints. We define two search directions using Barzilai and Borwein (BB1 and BB2) spectral parameters with modifications. In addition, we investigate the efficiency of the propose algorithm in restoring blurred images. The symbols $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ denote inner product and Euclidean norm respectively. The remaining part of this paper is organized as follows. In Section 2, we describe the proposed method and its global convergence. We report numerical experiments to show the efficiency of the algorithm in Section 3.

We describe the application of the proposed algorithm in Section 4 and give some conclusions as well as possible future research perspective in Section 5.

2. Two Step Iterative Scheme and Its Convergence Analysis

We begin this section with the following definition.

Definition 1. *Let $x, y \in \mathbb{R}^n$, a mapping $F : \mathbb{R}^n \to \mathbb{R}^n$ is said to be*

(i) monotone if

$$\langle F(x) - F(y),\ x - y\rangle \geq 0. \tag{10}$$

(ii) Lipschitzian continuous if there exists $L > 0$ such that

$$\|F(x) - F(y)\| \leq L\|x - y\|. \tag{11}$$

From the discussions in the preceding section, we observe that all the methods use the one-step formula (3) to update their respective sequence of iterates. Let I be an identity map in $\mathbb{R}^n$, if we set $d := (F - I)$, then formula (3) is closely related to the well-known Mann iterative scheme [22]

$$u_{k+1} = u_k + \alpha_k(F(u_k) - u_k), \tag{12}$$

where $0 \leq \alpha_k < 1$. Mann iteration has been applied to solve different kind of nonlinear problems successfully. However, its convergence speed is relatively slow. Different studies have shown that the famous two-step Ishikawa iterative scheme [23]

$$\begin{aligned} v_k &= (1 - \alpha_k)u_k + \alpha_k F(u_k), \\ u_{k+1} &= (1 - \beta_k)u_k + \beta_k F(v_k), \end{aligned} \tag{13}$$

where $\alpha_k, \beta_k \in [0, 1)$, converges faster than the one-step Mann iteration.

Let $\bar{d}_k = (F - I)u_k$ and $\hat{d}_k = F(v_k) - u_k$, then Ishikawa iterative scheme can be rewritten as follows

$$\begin{aligned} v_k &= u_k + \alpha_k \bar{d}_k, \\ u_{k+1} &= u_k + \beta_k \hat{d}_k. \end{aligned} \tag{14}$$

Based on the fact that the two step Ishikawa iterative scheme has faster convergence speed than the one-step Mann iterative scheme, in this paper, we propose a new two-step iterative scheme incorporating nonnegative BB parameters with projection strategy to solve monotone nonlinear equation with convex constraints. Given a starting point $x_0 \in \Lambda$ and $\alpha_k, \beta_k \in (0, 1]$, we define the updating formula for the proposed two-step scheme as follows

$$\begin{aligned} w_k &= x_k + \alpha_k d_k^I, \\ x_{k+1} &= P_\Lambda\left[x_k - \frac{\langle F(z_k),\ x_k - z_k\rangle}{\|F(z_k)\|^2} F(z_k)\right], \end{aligned} \tag{15}$$

where $z_k = x_k + \beta_k d_k^{II}$, $P_\Lambda(\cdot)$ is a projection operator defined below and

$$\begin{cases} d_k^I = -F(x_k), & \text{if } k = 0, \\ d_k^I = -\lambda^I(x_k)F(x_k), & \text{if } k > 0, \\ d_k^{II} = -\lambda^{II}(w_k)F(x_k), & \text{for } \ k \geq 0. \end{cases} \tag{16}$$

For simplicity we denote $\lambda^I(x_k) := \lambda_k^I$ and $\lambda^{II}(w_k) := \lambda_k^{II}$. The parameters λ_k^I and λ_k^{II} are modifications of the BB parameters (7) given as follows

$$\begin{cases} \lambda_k^I = \frac{\|s_k^I\|^2}{\langle y_k^I,\, s_k^I\rangle}, \\ \lambda_k^{II} = \frac{\langle y_k^{II},\, s_k^{II}\rangle}{\|y_k^{II}\|^2}, \end{cases} \tag{17}$$

where

$$\begin{cases} y_k^I = F(x_{k+1}) - F(x_k) + rs_k^I, & r > 0, \\ y_k^{II} = F(w_k) - F(x_k) + ts_k^{II}, & t > 0, \\ s_k^I = x_{k+1} - x_k, \\ s_k^{II} = w_k - x_k. \end{cases} \tag{18}$$

Assumption 1. *Throughout this paper, we assume the following*

(i) *The solution set of problem (1) is nonempty.*
(ii) *The mapping $F : \mathbb{R}^n \to \mathbb{R}^n$ satisfies (10)–(11).*
(iii) *The sequence $\{\alpha_k\}$ is in $(0,1)$ such that $\lim\limits_{k\to\infty} \alpha_k = 0$.*

The following Lemma shows that the spectral parameters (17) are well-defined and bounded.

Lemma 1. *Suppose that Assumption 1 holds and $t > L > 0$, then we have*

$$\eta \leq \lambda_k^I \leq \mu,$$

and

$$\delta \leq \lambda_k^{II} \leq \gamma,$$

where $\eta = \frac{1}{L+r}$, $\mu = \frac{1}{r}$, $\delta = \frac{t}{(t+L)^2}$, and $\gamma = \frac{t+L}{(t-L)^2}$.

Proof of Lemma 1. The monotonicity of F gives $\langle F(x_{k+1}) - F(x_k),\, x_{k+1} - x_k\rangle \geq 0$. Therefore, by the definition of y_k^I and y_k^{II} we have

$$\langle y_k^I,\, s_k^I\rangle = \langle F(x_{k+1}) - F(x_k),\, s_k^I\rangle + r\langle s_k^I,\, s_k^I\rangle \geq r\|s_k^I\|^2. \tag{19}$$

$$\langle y_k^{II},\, s_k^{II}\rangle = \langle F(w_k) - F(x_k),\, s_k^{II}\rangle + t\langle s_k^{II},\, s_k^{II}\rangle \geq t\|s_k^{II}\|^2. \tag{20}$$

On the other hand, by (11) and Cauchy Schwarz inequality, we have

$$\langle y_k^I,\, s_k^I\rangle = \langle F(x_{k+1}) - F(x_k),\, s_k^I\rangle + r\langle s_k^I,\, s_k^I\rangle \leq (L+r)\|s_k^I\|^2. \tag{21}$$

$$\langle y_k^{II},\, s_k^{II}\rangle = \langle F(w_k) - F(x_k),\, s_k^{II}\rangle + t\langle s_k^{II},\, s_k^{II}\rangle \leq (t+L)\|s_k^{II}\|^2. \tag{22}$$

$$\|y_k^{II}\| = \|F(w_k) - F(x_k) + t(w_k - x_k)\| \leq (t+L)\|s_k^{II}\|. \tag{23}$$

Also since $t > L$, from (23) we can have

$$(t-L)\|s_k^{II}\| \leq \|F(w_k) - F(x_k) + t(w_k - x_k)\| \leq (t+L)\|s_k^{II}\|. \tag{24}$$

Therefore, by (19) and (21) we have

$$\frac{1}{L+r} \leq \frac{\|s_k^I\|^2}{\langle y_k^I,\, s_k^I\rangle} \leq \frac{1}{r},$$

and from (20), (22) and (24) we have

$$\frac{t}{(t+L)^2} \leq \frac{\langle y_k^{II},\, s_k^{II}\rangle}{\|y_k^{II}\|^2} \leq \frac{t+L}{(t-L)^2}.$$

□

Remark 1. *We give the following remarks*

(i) *From Lemma 1, it is not difficult to see that the two search directions d_k^I and d_k^{II} satisfy the descent condition. That is,*

$$\begin{cases} \langle d_k^I,\, F(x_k)\rangle \leq -\eta\|F(x_k)\|^2 \\ \langle d_k^{II},\, F(x_k)\rangle \leq -\delta\|F(x_k)\|^2. \end{cases} \tag{25}$$

(ii) *The two search directions d_k^I and d_k^{II} satisfy the following inequalities*

$$\begin{cases} \eta\|F(x_k)\| \leq \|d_k^I\| \leq \mu\|F(x_k)\| \\ \delta\|F(x_k)\| \leq \|d_k^{II}\| \leq \gamma\|F(x_k)\|. \end{cases} \tag{26}$$

Next, we describe the projection operator in (15) which is usually used in iterative algorithms for solving problems such as fixed point problem, variational inequality problem, and so on. Let $x \in \mathbb{R}^n$ and define an operator $P_\Lambda : \mathbb{R}^n \to \Lambda$ by $P_\Lambda(x) = \operatorname{argmin}\{\|x-y\| : y \in \Lambda\}$. The operator P_Λ is called a projection onto the feasible set Λ and it enjoys the nonexpansive property, that is, $\|P_\Lambda(x) - P_\Lambda(y)\| \leq \|x-y\|, \ \forall \ x, y \in \mathbb{R}^n$. If $y \in \Lambda$, then $P_\Lambda(y) = y$ and therefore, we have

$$\|P_\Lambda(x) - y\| \leq \|x-y\|. \tag{27}$$

We now state the steps of the proposed algorithm which we call two-step spectral gradient method.

Remark 2. *We quickly note the following remarks*

(i) *We claim that there exists a step-size β_k satisfying the line search (1) for any $k \geq 0$. Suppose on the contrary that there exists some k_0 such that for any $i = 0, 1, 2, ...$, the line search (1) is not satisfied, that is*

$$-\langle F(x_{k_0} + \kappa\varrho^i d(w_{k_0})),\, d(w_{k_0})\rangle < \sigma\kappa\varrho^i\|d(w_{k_0})\|^2\|F(x_{k_0} + \kappa\varrho^i d(w_{k_0}))\|^{1/c}. \tag{28}$$

Since F is continuous and λ_k^{II} is bounded for all k, letting $i \to \infty$ yields

$$\|F(x_{k_0})\| \leq 0. \tag{29}$$

It is clear that the inequality (29) cannot hold. Hence the line search (1) is well-defined.

(ii) *The line search defined by (1) is more general than that of Reference [24].*

(iii) *It follows from (15) and Assumption 1 that $\lim\limits_{k\to\infty}\|w_k - x_k\| = 0$.*

The next Lemma is very crucial to the convergence of Algorithm 1.

Algorithm 1: Two-Step Spectral Gradient Projection Method (TSSP)

```
Input: Given $x_0 \in \Lambda, 0 < \kappa \leq 1, r, t > 0, \sigma, \varrho \in (0,1)$ and $\{\alpha_k\} \in (0,1)$. Set $k = 0$.
Step 1: Compute $d_k^I$ using (16).
if $d_k^I = 0$, then
    $x_k$ is a solution and the iteration process stops.
end
Step 2: Compute $w_k = x_k + \alpha_k d_k^I$.
Step 3: Compute $d_k^{II}$ using (16).
Step 4: Define a set $\{\kappa, \kappa\varrho, \kappa\varrho^2, \cdots, \kappa\varrho^n\}$,
while
    $$-\langle F(x_k + \kappa\varrho^i d_k^{II}),\ d_k^{II}\rangle \geq \sigma\kappa\varrho^i \|d_k^{II}\|^2 \|F(x_k + \kappa\varrho^i d_k^{II})\|^{1/c},\ \ c \geq 1,$$
  do
    set $\beta_k = \kappa\varrho^i$ where $i$ is the smallest nonegative integer.
end
Step 5: Set $z_k = x_k + \beta_k d_k^{II}$,
if $z_k \in \Lambda$ and $\|F(z_k)\| = 0$, then
    stop,
else
    $$x_{k+1} = P_\Lambda\left[x_k - \frac{\langle F(z_k),\ x_k - z_k\rangle}{\|F(z_k)\|^2} F(z_k)\right].$$
end
Step 6: Set $k := k + 1$ and go to Step 1.
```

Lemma 2. *Let the Assumption 1 holds, then the sequences $\{w_k\}$, $\{z_k\}$ and $\{x_k\}$ generated by Algorithm 1 are bounded. In addition, there exist some positive constants m_1, m_2 and m_3 such that*

$$\begin{cases} \|F(x_k)\| \leq m_1 \\ \|F(w_k)\| \leq m_2 \\ \|F(z_k)\| \leq m_3. \end{cases} \tag{30}$$

Furthermore,

$$\lim_{k\to\infty} \beta_k \|d_k^{II}\| = 0, \tag{31}$$

and

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0. \tag{32}$$

Proof of Lemma 2. Let x^* be a solution of problem (1), then by monotonicity of F, we have

$$\begin{aligned} \langle F(z_k),\ x_k - x^*\rangle &= \langle F(z_k),\ x_k - z_k + z_k - x^*\rangle \\ &= \langle F(z_k),\ x_k - z_k\rangle + \langle F(z_k) - F(x^*),\ z_k - x^*\rangle \\ &\geq \langle F(z_k),\ x_k - z_k\rangle. \end{aligned} \tag{33}$$

By the definition of x_{k+1}, and (33) we have

$$\begin{aligned}
\|x_{k+1}-x^*\|^2 &= \left\| P_\Lambda \left[x_k - \frac{\langle F(z_k),\, x_k - z_k\rangle}{\|F(z_k)\|^2} F(z_k) \right] - x^* \right\|^2 \\
&\le \left\| x_k - x^* - \frac{\langle F(z_k),\, x_k - z_k\rangle}{\|F(z_k)\|^2} F(z_k) \right\|^2 \\
&= \|x_k - x^*\|^2 - 2\frac{\langle F(z_k),\, x_k - z_k\rangle}{\|F(z_k)\|^2} \langle F(z_k),\, x_k - x^*\rangle + \frac{\langle F(z_k),\, x_k - z_k\rangle^2}{\|F(z_k)\|^2} \\
&\le \|x_k - x^*\|^2 - 2\frac{\langle F(z_k),\, x_k - z_k\rangle}{\|F(z_k)\|^2} \langle F(z_k),\, x_k - z_k\rangle + \frac{\langle F(z_k),\, x_k - z_k\rangle^2}{\|F(z_k)\|^2} \\
&= \|x_k - x^*\|^2 - \frac{\langle F(z_k),\, x_k - z_k\rangle^2}{\|F(z_k)\|^2} \\
&\le \|x_k - x^*\|^2.
\end{aligned} \tag{34}$$

This implies that $\|x_k - x^*\| \le \|x_0 - x^*\|$ for all k, and therefore the sequence $\{x_k\}$ is bounded and $\lim_{k\to\infty} \|x_k - x^*\|$ exists. Let m_1 be a positive constant such that $\|x_0 - x^*\| = m_1/L$, since F is Lipschitzian continuous, we have

$$\begin{aligned}
\|F(x_k)\| &= \|F(x_k) - F(x^*)\| \\
&\le L\|x_k - x^*\| \\
&\le L\|x_0 - x^*\| \\
&= m_1.
\end{aligned} \tag{35}$$

It follows from (26), that $\|d_k^I\| \le \mu m_1$ and $\|d_k^{II}\| \le \gamma m_1$. It further follows from (15) that $\{w_k\}$ is bounded. By Lipschitzian continuity of F, there exists $m_2 > 0$ such that $\|F(w_k)\| \le m_2$.

Since $\{d_k^{II}\}$ is bounded, it follows from the definition z_k that $\{z_k\}$ is also bounded. By Lipschitzian continuity of F, there exists some constant m_3 for which

$$\|F(z_k)\| \le m_3. \tag{36}$$

Since the stepsize β_k in Step 4 of Algorithm 1 satisfies $\beta_k \le 1, \forall\, k$, then from (1), we have

$$\sigma^2 \beta_k^4 \|d_k^{II}\|^4 \|F(z_k)\|^{2/c} \le \sigma^2 \beta_k^2 \|d_k^{II}\|^4 \|F(z_k)\|^{2/c} \le \langle F(z_k),\, \beta_k d_k^{II}\rangle^2.$$

Combining with (34) gives

$$\frac{\sigma^2 \beta_k^4 \|d_k^{II}\|^4 \|F(z_k)\|^{2/c}}{\|F(z_k)\|^2} \le \frac{\langle F(z_k),\, \beta_k d_k^{II}\rangle^2}{\|F(z_k)\|^2} \le \|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2. \tag{37}$$

By (36) and (37), we have

$$\sigma^2 \beta_k^4 \|d_k^{II}\|^4 \le \|F(z_k)\|^{2-\frac{2}{c}} \left(\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 \right) \le m_3^{2-\frac{2}{c}} \left(\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 \right).$$

Taking limit gives

$$\sigma^2 \lim_{k\to\infty} \beta_k^4 \|d_k^{II}\|^4 = 0. \tag{38}$$

Hence, it holds that

$$\lim_{k\to\infty} \beta_k \|d_k^{II}\| = 0. \tag{39}$$

This together with the definition of z_k in Step 5 of Algorithm 1 yields

$$\lim_{k\to\infty} \|z_k - x_k\| = 0. \tag{40}$$

By the property of projection (27), we have

$$\begin{aligned}
\lim_{k\to\infty} \|x_{k+1} - x_k\| &= \lim_{k\to\infty} \left\| P_\Lambda \left[x_k - \frac{\langle F(z_k),\, x_k - z_k\rangle}{\|F(z_k)\|^2} F(z_k) \right] - x_k \right\| \\
&\leq \lim_{k\to\infty} \left\| x_k - \frac{\langle F(z_k),\, x_k - z_k\rangle}{\|F(z_k)\|^2} F(z_k) - x_k \right\| \\
&\leq \lim_{k\to\infty} \|x_k - z_k\| \\
&= 0.
\end{aligned} \tag{41}$$

□

Theorem 1. *Let $\{x_k\}$ be the sequence generated by Algorithm 1. Suppose that Assumption 1 holds, then the sequence $\{x_k\}$ converges to a point x^* which satisfies $F(x^*) = 0$.*

Proof of Theorem 1. We begin by proving that

$$\liminf_{k\to\infty} \|F(x_k)\| = 0. \tag{42}$$

Suppose on the contrary that (42) does not hold, then there exists $q > 0$ for which

$$\|F(x_k)\| \geq q,\ \forall\, k \geq 0. \tag{43}$$

If $\beta_k \neq \kappa$, since Algorithm 1 uses a backtracking process to compute β_k starting from κ, then $\varrho^{-1}\beta_k$ does not satisfy (1), that is,

$$-\langle F(x_k + \varrho^{-1}\beta_k d_k^{II}),\, d_k^{II}\rangle < \sigma\varrho^{-1}\beta_k \|d_k^{II}\|^2 \|F(x_k + \varrho^{-1}\beta_k d_k^{II})\|^{1/c}. \tag{44}$$

Consequently, we have from Remark 1 (i),

$$\begin{aligned}
\delta\|F(x_k)\|^2 &\leq -\langle d_k^{II},\, F(x_k)\rangle \\
&= -\langle d_k^{II},\, F(x_k) - F(x_k + \varrho^{-1}\beta_k d_k^{II}) + F(x_k + \varrho^{-1}\beta_k d_k^{II})\rangle \\
&= -\langle d_k^{II},\, F(x_k) - F(x_k + \varrho^{-1}\beta_k d_k^{II})\rangle - \langle d_k^{II},\, F(x_k + \varrho^{-1}\beta_k d_k^{II})\rangle \\
&< -\langle d_k^{II},\, F(x_k) - F(x_k + \varrho^{-1}\beta_k d_k^{II})\rangle + \sigma\varrho^{-1}\beta_k\|d_k^{II}\|^2\|F(x_k + \varrho^{-1}\beta_k d_k^{II})\|^{1/c} \\
&\leq \|d_k^{II}\|\|F(x_k + \varrho^{-1}\beta_k d_k^{II}) - F(x_k)\| + \sigma\varrho^{-1}\beta_k\|d_k^{II}\|^2\|F(x_k + \varrho^{-1}\beta_k d_k^{II})\|^{1/c} \\
&\leq L\|d_k^{II}\|\|x_k + \varrho^{-1}\beta_k d_k^{II} - x_k\| + \sigma\varrho^{-1}\beta_k\|d_k^{II}\|^2\|F(x_k + \varrho^{-1}\beta_k d_k^{II})\|^{1/c} \\
&\leq L\varrho^{-1}\beta_k\|d_k^{II}\|^2 + \sigma\varrho^{-1}\beta_k\|d_k^{II}\|^2\|F(x_k + \varrho^{-1}\beta_k d_k^{II})\|^{1/c} \\
&\leq (L\varrho^{-1}m_4 + \sigma\varrho^{-1}m_4 m_5^{1/c})\beta_k\|d_k^{II}\|,
\end{aligned}$$

where $\|F(x_k + \varrho^{-1}\beta_k d_k^{II})\|$ is bounded above by a positive constant m_5. This means

$$\begin{aligned}
\beta_k\|d_k^{II}\| &\geq \frac{\varrho\delta\|F(x_k)\|^2}{m_4(L + \sigma m_5^{1/c})} \\
&\geq \frac{\varrho\delta q^2}{m_4(L + \sigma m_5^{1/c})}.
\end{aligned}$$

Taking limit on both sides as $k \to \infty$, we have

$$\lim_{k\to\infty} \beta_k \|d_k^{II}\| > 0. \tag{45}$$

This contradicts (39). Hence (42) must hold. Now, since F is continuous and the sequence $\{x_k\}$ is bounded, then there is some accumulation point of $\{x_k\}$ say x^* for which $\|F(x^*)\| = 0$. By boundedness of $\{x_k\}$, we can find subsequence $\{x_{k_j}\}$ of $\{x_k\}$ for which $\lim_{j\to\infty} \|x_{k_j} - x^*\| = 0$. From the proof of Lemma 2, we know that $\lim_{k\to\infty} \|x_k - x^*\|$ exists. Therefore, we can conclude that $\lim_{k\to\infty} \|x_k - x^*\| = 0$ and the proof is complete. □

3. Numerical Results and Comparison

Attention is now turn to numerical experiments. The experiment is divided into two parts. The first experiment aims to explore the role of the parameter c in the definition of the line search (1). On the other hand, the second experiment discusses the computational advantage of the proposed method in comparison with two existing methods. The two existing methods are:

(i) Spectral gradient projection method for monotone nonlinear equations with convex constraints proposed by Yu et al. [17].

(ii) Two spectral gradient projection methods for constrained equations and their linear convergence rate proposed by Liu and Duan [25]. This method has two algorithms i.e., Algorithm 2.1 and Algorithm 2.2. We only compare our proposed method with Algorithm 2.1 since Algorithm 2.2 is similar with that Yu et al. [17].

These two methods were chosen because their search directions are defined based on the BB parameters. For convenience, we respectively denote the two methods by SGPM and TSGP. Algorithm 1 TSSP is implemented using the following parameters $\kappa = 1$, $\sigma = 0.01$, $\varrho = 0.5$, $r = t = 0.01$, and $\alpha_k = \frac{1}{(k+1)^2}$. The parameters used for the SGPM and TSGP methods were taken respectively from References [17] and [25]. The metrics used for the comparison are: number of iterations (ITER), number of function evaluations (FVAL) and CPU time (TIME). In the course of the experiments, we solved six benchmark test problems using six (6) different starting points (see Table 1) by varying the number of dimension. The test problems are denoted by $Pi,\ i = 1,2,3,4,5,6$. Since the proposed algorithm is derivative-free, the test problems include two nonsmooth problems. The three solvers were coded in MATLAB R2017a and run on a PC with intel Core(TM) i5-8250u processor with 4 GB of RAM and CPU 1.60 GHZ. The MATLAB code for the TSSP algorithm is available in https://github.com/aliyumagsu/TSSP_Algorithm. The iteration process is terminated whenever the inequality $\|F(x_k)\| \leq 10^{-6}$ or $\|F(z_k)\| \leq 10^{-6}$ is satisfied and failure is declared whenever the number of iterations exceeds 1000 and the terminating criterion mentioned above has not been satisfied.

First experiment. This experiment discusses the role of the parameter c in the definition of the line search (1) with regards to the performance of the TSSP algorithm. We solved all the test Problems 1–6 with dimension $n = 1000$, using all the given initial guesses in Table 1 by varying the values of c. That is, $c = \{1,2,3,4,5\}$. The comparison is based on ITER, FVAL and norm of the objective function, (NORM), where the experimental results are presented in Table 2. CPU time results are omitted in Table 2 because virtually all are less than 1 s. The results obtained reveal that the parameter c slightly affected the performance of TSSP algorithm when solving Problems 2 and 6. For problem 2, Algorithm 1 TSSP recorded least ITER and FVAL when $c = 4$ and 5 while different ITER and FVAL values recorded for different values of c may be associated with the random starting points chosen independently by MATLAB. However, extensive numerical experiment is needed to investigate the role of the parameter c in the performance of the TSSP algorithm.

Second experiment. This experiment presents the computational advantage of the proposed method in comparison with the two existing methods mentioned above based on ITER, FVAL and TIME. All the test problems $1-6$ were solved using the starting points in Table 1 with three (3) different dimensions $n = 1000$, 50,000 and 100,000. In this experiment, we take $c = 2$. The results obtained by each solver are reported in Tables 3 and 4. The NORM results presented in Tables 3 and 4 show that each solver successfully obtained solutions of all the test Problems 1–6. However, it is clear that the TSSP algorithm obtained the solutions of virtually all the test problems with least ITER, FVAL and TIME. These information are summarized in Figures 1–3 based on the Dolan and Moré performance profile [26]. This performance profile tells the percentage win by each solver. In all the experiments, we see from Figures 1–3 that the proposed TSSP algorithm performs better with higher percentage win based on ITER, FVAL and TIME for solving all the test problems. In fact, the TSSP algorithm recorded 100 percent least FVAL for all the test problems.

Table 1. Starting points used for the test problems.

Starting Points (SP)	Values
x_1	$(0.1, 0.1, 0.1, \cdots, 0.1)^T$
x_2	$(\frac{1}{2}, \frac{1}{2^2}, \frac{1}{2^3}, \cdots, \frac{1}{2^n})^T$
x_3	$(2, 2, \ldots, 2)^T$
x_4	$(1, \frac{1}{2}, \frac{1}{3}, \cdots, \frac{1}{n})^T$
x_5	$(1 - \frac{1}{n}, 1 - \frac{2}{n}, 1 - \frac{3}{n}, \cdots, 0)^T$
x_6	$\text{rand}(0, 1)$

We use the following test problems where $F(x) = (f_1(x), f_2(x), \cdots, f_n(x))^T$, and $x = (x_1, x_2, \cdots, x_n)^T$.

Problem 1 ([27]).

$$f_1(x) = e^{x_1} - 1$$
$$f_i(x) = e^{x_i} + x_{i-1} - 1, \ \ i = 1, 2, \cdots, n - 1,$$

where $\Lambda = \mathbb{R}^n_+$.

Problem 2 ([28]).

$$f_i(x_i) = \log(x_i + 1) - \frac{x_i}{n}, \ \ i = 1, 2, \cdots, n,$$

where $\Lambda = \{x \in \mathbb{R}^n : \sum_{i=1}^{n} x_i \leq n, \ x_i > -1, \ i = 1, 2, \cdots, n\}$.

Problem 3 ([29]).

$$f_i(x) = 2x_i - \sin|x_i|, \ \ i = 1, 2, \cdots, n,$$

where $\Lambda = \mathbb{R}^n_+$.

Problem 4 ([30]).

$$f_i(x) = e^{x_i} - 1, \ \ i = 1, 2, \cdots, n,$$

where $\Lambda = \mathbb{R}^n_+$.

Problem 5 ([29]).

$$f_1(x) = x_1 - e^{\cos(h(x_1 + x_2))}$$
$$f_i(x) = x_i - e^{\cos(h(x_{i-1} + x_i + x_{i+1}))}, \ \ i = 1, 2, \cdots, n - 1,$$
$$f_n(x) = x_n - e^{\cos(h(x_{n-1} + x_n))},$$

where $h = \frac{1}{n+1}$ *and* $\Lambda = \mathbb{R}^n_+$.

Problem 6 ([31]).

$$f_i(x) = x_i - \sin(|x_i - 1|), \ \ i = 1, 2, \cdots, n - 1,$$

where $\Lambda = \{x \in \mathbb{R}^n : \sum_{i=1}^{n} x_i \leq n,\ x_i \geq -1,\ i = 1, 2, \cdots, n\}$.

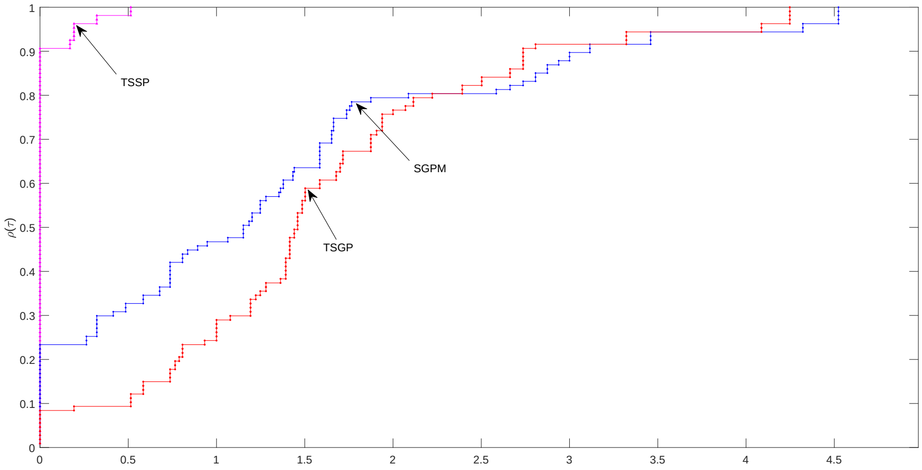

Figure 1. Dolan and Moré performance profile with respect to number of iterations.

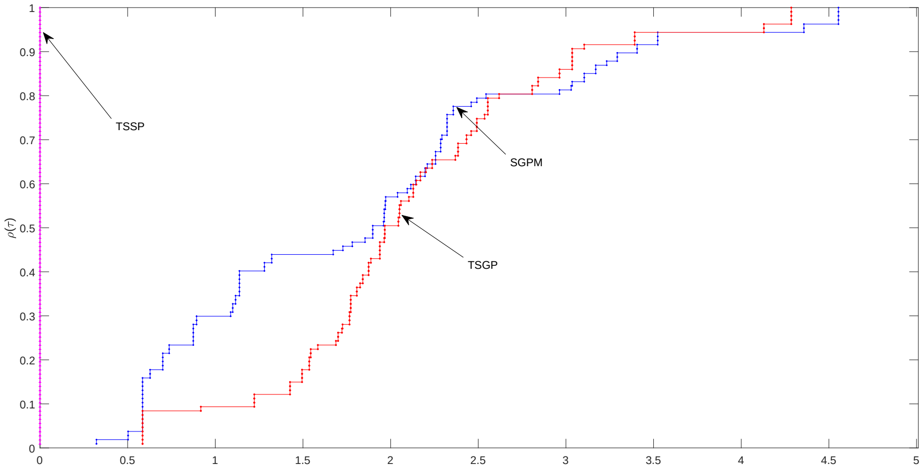

Figure 2. Dolan and Moré performance profile with respect to number of function evaluation.

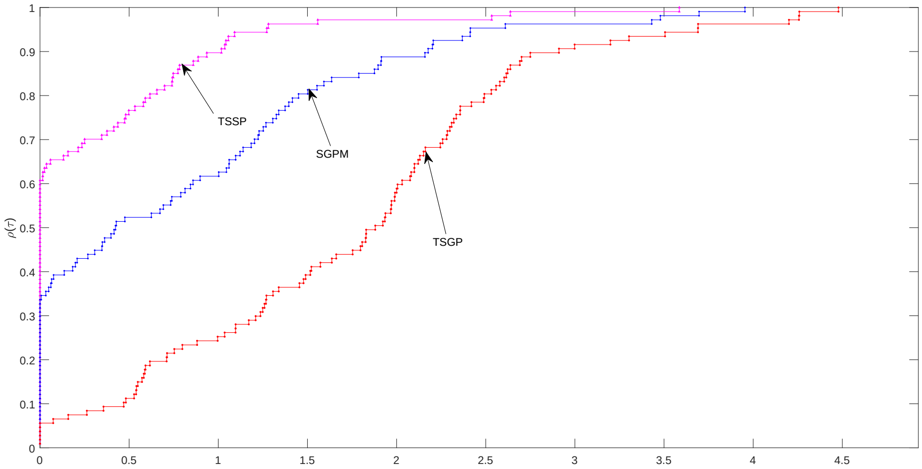

Figure 3. Dolan and Moré performance profile with respect to CPU time.

Table 2. Numerical results showing the effect of the parameter c in the line search.

	c	**1**			**2**			**3**			**4**			**5**		
	SP	**ITER**	**FVAL**	**NORM**	**ITER**	**FVAL**	**NORM**	**ITER**	**FVAL**	**NORM**	**ITER**	**FVAL**	**NORM**	**ITER**	**FVAL**	**NORM**
P1	x_1	4	6	2.18×10^{-7}	4	6	2.18×10^{-7}	4	6	2.18×10^{-7}	4	6	2.18×10^{-7}	4	6	2.18×10^{-7}
	x_2	7	9	2.82×10^{-7}	7	9	2.82×10^{-7}	7	9	2.82×10^{-7}	7	9	2.82×10^{-7}	7	9	2.82×10^{-7}
	x_3	5	7	1.05×10^{-7}	5	7	1.05×10^{-7}	5	7	1.05×10^{-7}	5	7	1.05×10^{-7}	5	7	1.05×10^{-7}
	x_4	7	9	6.34×10^{-8}	7	9	6.34×10^{-8}	7	9	6.34×10^{-8}	7	9	6.34×10^{-8}	7	9	6.34×10^{-8}
	x_5	7	9	6.13×10^{-8}	7	9	6.13×10^{-8}	7	9	6.13×10^{-8}	7	9	6.13×10^{-8}	7	9	6.13×10^{-8}
	x_6	7	9	2.77×10^{-8}	7	9	1.94×10^{-8}	7	9	2.3×10^{-8}	7	9	3.63×10^{-8}	7	9	1.94×10^{-8}
P2	x_1	3	5	2.41×10^{-8}	3	5	2.41×10^{-8}	3	5	2.41×10^{-8}	3	5	2.41×10^{-8}	3	5	2.41×10^{-8}
	x_2	8	10	5.09×10^{-8}	8	10	5.09×10^{-8}	8	10	5.09×10^{-8}	8	10	5.09×10^{-8}	8	10	5.09×10^{-8}
	x_3	8	10	1.23×10^{-7}	8	10	1.23×10^{-7}	8	10	1.23×10^{-7}	8	10	1.23×10^{-7}	9	11	1.49×10^{-7}
	x_4	9	11	1.53×10^{-7}	9	11	1.53×10^{-7}	9	11	1.53×10^{-7}	9	11	1.53×10^{-7}	9	11	1.53×10^{-7}
	x_5	10	12	5.81×10^{-8}	10	12	5.81×10^{-8}	10	12	5.81×10^{-8}	9	11	3.45×10^{-7}	9	11	3.45×10^{-7}
	x_6	10	12	4.97×10^{-8}	10	12	6.04×10^{-8}	10	12	5.27×10^{-8}	9	11	3.51×10^{-7}	9	11	3.49×10^{-7}
P3	x_1	3	5	4.03×10^{-8}	3	5	4.03×10^{-8}	3	5	4.03×10^{-8}	3	5	4.03×10^{-8}	3	5	4.03×10^{-8}
	x_2	3	5	1.19×10^{-7}	3	5	1.19×10^{-7}	3	5	1.19×10^{-7}	3	5	1.19×10^{-7}	3	5	1.19×10^{-7}
	x_3	4	6	8.27×10^{-7}	4	6	8.27×10^{-7}	4	6	8.27×10^{-7}	4	6	8.27×10^{-7}	4	6	8.27×10^{-7}
	x_4	4	6	2.12×10^{-8}	4	6	2.12×10^{-8}	4	6	2.12×10^{-8}	4	6	2.12×10^{-8}	4	6	2.12×10^{-8}
	x_5	5	7	1.46×10^{-7}	5	7	1.46×10^{-7}	5	7	1.46×10^{-7}	5	7	1.46×10^{-7}	5	7	1.46×10^{-7}
	x_6	5	7	1.46×10^{-7}	5	7	1.44×10^{-7}	5	7	9.49×10^{-8}	5	7	1.37×10^{-7}	5	7	1.88×10^{-7}
P4	x_1	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0
	x_2	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0
	x_3	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0
	x_4	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0
	x_5	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0
	x_6	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0
P5	x_1	3	5	7.96×10^{-7}	3	5	7.96×10^{-7}	3	5	7.96×10^{-7}	3	5	7.96×10^{-7}	3	5	7.96×10^{-7}
	x_2	3	5	8.26×10^{-7}	3	5	8.26×10^{-7}	3	5	8.26×10^{-7}	3	5	8.26×10^{-7}	3	5	8.26×10^{-7}
	x_3	3	5	2.18×10^{-7}	3	5	2.18×10^{-7}	3	5	2.18×10^{-7}	3	5	2.18×10^{-7}	3	5	2.18×10^{-7}
	x_4	3	5	8.24×10^{-7}	3	5	8.24×10^{-7}	3	5	8.24×10^{-7}	3	5	8.24×10^{-7}	3	5	8.24×10^{-7}
	x_5	3	5	6.8×10^{-7}	3	5	6.8×10^{-7}	3	5	6.8×10^{-7}	3	5	6.8×10^{-7}	3	5	6.8×10^{-7}
	x_6	3	5	6.77×10^{-7}	3	5	6.81×10^{-7}	3	5	6.8×10^{-7}	3	5	6.79×10^{-7}	3	5	6.87×10^{-7}
P6	x_1	3	5	1.07×10^{-7}	3	5	1.07×10^{-7}	3	5	1.07×10^{-7}	3	5	1.07×10^{-7}	3	5	1.07×10^{-7}
	x_2	9	11	8.56×10^{-9}	9	11	8.56×10^{-9}	8	10	7.89×10^{-7}	8	10	7.89×10^{-7}	8	10	7.89×10^{-7}
	x_3	4	6	1.15×10^{-8}	4	6	1.15×10^{-8}	4	6	1.15×10^{-8}	4	6	1.15×10^{-8}	4	6	1.15×10^{-8}
	x_4	11	13	1.26×10^{-7}	11	13	1.26×10^{-7}	11	13	1.26×10^{-7}	10	12	8.08×10^{-8}	10	12	8.08×10^{-8}
	x_5	10	12	9.99×10^{-9}	10	12	9.99×10^{-9}	10	12	9.99×10^{-9}	10	12	9.99×10^{-9}	10	12	9.99×10^{-9}
	x_6	10	12	2.72×10^{-8}	10	12	1.67×10^{-8}	9	11	1.46×10^{-7}	10	12	2.84×10^{-8}	10	12	3.82×10^{-8}

Table 3. Numerical results obtained by each solver.

			TSSP				SGPM				TSGP			
Problem	**DIM**	**SP**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**
P1	1000	x_1	4	6	0.23423	2.18×10^{-7}	5	11	0.071452	1.98×10^{-7}	10	21	0.027483	4.81×10^{-7}
		x_2	7	9	0.024129	2.82×10^{-7}	21	43	0.020705	8.07×10^{-7}	16	33	0.00993	7.7×10^{-7}
		x_3	5	7	0.00479	1.05×10^{-7}	8	17	0.005439	9.35×10^{-9}	14	29	0.006688	7.24×10^{-7}
		x_4	7	9	0.006348	6.34×10^{-8}	22	45	0.007646	6.66×10^{-7}	10	21	0.006682	8.79×10^{-7}
		x_5	7	9	0.007179	6.13×10^{-8}	7	15	0.005533	2.11×10^{-8}	8	17	0.007977	9.36×10^{-7}
		x_6	7	9	0.01028	3.35×10^{-8}	7	15	0.008546	2.31×10^{-8}	14	29	0.006145	6.95×10^{-7}
	50,000	x_1	4	6	0.065863	6.63×10^{-8}	5	11	0.06618	3.18×10^{-7}	11	23	0.26178	8.02×10^{-7}
		x_2	7	9	0.06825	2.82×10^{-7}	21	43	0.16873	8.07×10^{-7}	16	33	0.23724	7.7×10^{-7}
		x_3	5	7	0.1091	7.38×10^{-7}	8	17	0.092683	6.61×10^{-8}	16	33	0.19828	7.44×10^{-7}
		x_4	7	9	0.07815	6.35×10^{-8}	22	45	0.19587	6.67×10^{-7}	10	21	0.11777	8.72×10^{-7}
		x_5	8	10	0.090836	8×10^{-9}	7	15	0.052803	1.34×10^{-7}	17	35	0.212	7.38×10^{-7}
		x_6	8	10	0.081964	6.53×10^{-9}	7	15	0.049061	1.32×10^{-7}	16	33	0.17428	8.9×10^{-7}
	100,000	x_1	4	6	0.10587	7.46×10^{-8}	5	11	0.12233	4.34×10^{-7}	12	25	0.35698	4.49×10^{-7}
		x_2	7	9	0.13938	2.82×10^{-7}	21	43	0.36669	8.07×10^{-7}	16	33	0.53382	7.7×10^{-7}
		x_3	6	8	0.17202	5.19×10^{-9}	8	17	0.15426	9.34×10^{-8}	17	35	0.58949	4.19×10^{-7}
		x_4	7	9	0.15223	6.35×10^{-8}	22	45	0.473	6.67×10^{-7}	10	21	0.24947	8.72×10^{-7}
		x_5	8	10	0.22073	1.13×10^{-8}	7	15	0.1036	1.9×10^{-7}	18	37	0.46019	4.18×10^{-7}
		x_6	8	10	0.15584	1.18×10^{-8}	7	15	0.1035	1.94×10^{-7}	17	35	0.51277	7.07×10^{-7}
P2	1000	x_1	3	5	0.053692	2.41×10^{-8}	19	39	0.022091	6.53×10^{-7}	14	29	0.009273	9.17×10^{-7}
		x_2	8	10	0.006247	5.09×10^{-8}	19	39	0.008377	5.84×10^{-7}	14	29	0.01022	7.19×10^{-7}
		x_3	8	10	0.007091	1.23×10^{-7}	24	49	0.009118	7.53×10^{-7}	19	39	0.0171	5.49×10^{-7}
		x_4	9	11	0.006962	1.53×10^{-7}	20	41	0.00765	7.01×10^{-7}	15	31	0.011734	7.44×10^{-7}
		x_5	10	12	0.010053	5.81×10^{-8}	23	47	0.010398	9.51×10^{-7}	17	35	0.012062	9.65×10^{-7}
		x_6	10	12	0.00733	7.06×10^{-8}	23	47	0.009659	9.64×10^{-7}	17	35	0.015027	9.61×10^{-7}
	50,000	x_1	3	5	0.058732	1.71×10^{-7}	22	45	0.27095	1.14×10^{-8}	17	35	0.26277	4.05×10^{-7}
		x_2	8	10	0.1112	5.09×10^{-8}	19	39	0.17143	5.86×10^{-7}	14	29	0.2504	7.22×10^{-7}
		x_3	8	10	0.11157	8.68×10^{-7}	27	55	0.33645	1.35×10^{-8}	21	43	0.44285	6.23×10^{-7}
		x_4	9	11	0.11286	1.52×10^{-7}	20	41	0.18758	7.05×10^{-7}	15	31	0.26827	7.49×10^{-7}
		x_5	10	12	0.13738	4.2×10^{-7}	26	53	0.24905	8.68×10^{-7}	20	41	0.32368	4.36×10^{-7}
		x_6	10	12	0.15618	4.3×10^{-7}	26	53	0.36605	8.69×10^{-7}	20	41	0.31156	4.35×10^{-7}
	100,000	x_1	3	5	0.091277	2.42×10^{-7}	21	43	0.41274	3.2×10^{-8}	17	35	0.55481	5.73×10^{-7}
		x_2	8	10	0.19525	5.09×10^{-8}	19	39	0.35045	5.86×10^{-7}	14	29	0.48308	7.22×10^{-7}
		x_3	9	11	0.27631	1.21×10^{-8}	27	55	0.628	1.91×10^{-8}	21	43	0.66514	8.81×10^{-7}
		x_4	9	11	0.32755	1.52×10^{-7}	20	41	0.44064	7.05×10^{-7}	15	31	0.48698	7.49×10^{-7}
		x_5	10	12	0.25909	5.94×10^{-7}	27	55	0.54056	6.2×10^{-7}	20	41	0.74411	6.16×10^{-7}
		x_6	10	12	0.28941	5.64×10^{-7}	27	55	0.57984	6.19×10^{-7}	20	41	0.66926	6.16×10^{-7}
P3	1000	x_1	3	5	0.027505	4.03×10^{-8}	5	11	0.004418	1.97×10^{-8}	11	23	0.004932	4.32×10^{-7}
		x_2	3	5	0.002243	1.19×10^{-7}	5	11	0.004881	3.84×10^{-8}	8	17	0.006148	5.84×10^{-7}
		x_3	4	6	0.003901	8.27×10^{-7}	6	13	0.00328	4.62×10^{-7}	13	27	0.010201	6.63×10^{-7}
		x_4	4	6	0.002999	2.12×10^{-8}	5	11	0.002216	3.93×10^{-7}	11	23	0.007859	9.11×10^{-7}
		x_5	5	7	0.004183	1.46×10^{-7}	6	13	0.002305	4.67×10^{-8}	14	29	0.009221	8.68×10^{-7}
		x_6	5	7	0.002517	1.08×10^{-7}	6	13	0.002634	4.62×10^{-8}	14	29	0.007964	8.55×10^{-7}
	50,000	x_1	3	5	0.037839	2.85×10^{-7}	5	11	0.046813	1.39×10^{-7}	13	27	0.3024	4.88×10^{-7}
		x_2	3	5	0.043702	1.19×10^{-7}	5	11	0.075547	3.84×10^{-8}	8	17	0.13	5.84×10^{-7}
		x_3	5	7	0.067654	5.79×10^{-8}	7	15	0.053226	3.24×10^{-8}	15	31	0.2087	7.48×10^{-7}
		x_4	4	6	0.043413	2.13×10^{-8}	5	11	0.041688	3.93×10^{-7}	11	23	0.20256	9.12×10^{-7}
		x_5	6	8	0.05719	1.03×10^{-8}	6	13	0.052196	3.31×10^{-7}	16	33	0.31984	9.81×10^{-7}
		x_6	6	8	0.080035	1.01×10^{-8}	6	13	0.047676	3.31×10^{-7}	16	33	0.31002	9.75×10^{-7}
	100,000	x_1	3	5	0.074948	4.03×10^{-7}	5	11	0.095503	1.97×10^{-7}	13	27	0.38401	6.89×10^{-7}
		x_2	3	5	0.068017	1.19×10^{-7}	5	11	0.070957	3.84×10^{-8}	8	17	0.18947	5.84×10^{-7}
		x_3	5	7	0.26113	8.19×10^{-8}	7	15	0.13652	4.58×10^{-8}	16	33	0.51768	4.22×10^{-7}
		x_4	4	6	0.13668	2.13×10^{-8}	5	11	0.073903	3.93×10^{-7}	11	23	0.31674	9.12×10^{-7}
		x_5	6	8	0.16527	1.46×10^{-8}	6	13	0.080572	4.68×10^{-7}	17	35	0.46576	5.54×10^{-7}
		x_6	6	8	0.15469	1.5×10^{-8}	6	13	0.1035	4.68×10^{-7}	17	35	0.49075	5.52×10^{-7}

Table 4. Numerical results obtained by each solver.

			TSSP				SGPM				TSGP			
Problem	**DIM**	**SP**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**
P4	1000	x_1	1	2	0.024999	0	1	3	0.002082	0	1	3	0.002665	0
		x_2	1	2	0.00153	0	1	3	0.001492	0	10	21	0.013708	6.89×10^{-7}
		x_3	1	2	0.002033	0	1	3	0.001752	0	1	3	0.003047	0
		x_4	1	2	0.001605	0	11	23	0.00416	8.04×10^{-7}	1	3	0.001204	0
		x_5	1	2	0.001898	0	20	41	0.006966	6.33×10^{-7}	17	35	0.024513	5.22×10^{-7}
		x_6	1	2	0.002199	0	20	41	0.005064	6.89×10^{-7}	17	35	0.009568	5.82×10^{-7}
	50,000	x_1	1	2	0.009816	0	1	3	0.02162	0	1	3	0.027567	0
		x_2	1	2	0.015792	0	1	3	0.010298	0	10	21	0.11697	6.89×10^{-7}
		x_3	1	2	0.019539	0	1	3	0.009652	0	1	3	0.04943	0
		x_4	1	2	0.022933	0	11	23	0.087566	7.32×10^{-7}	1	3	0.016433	0
		x_5	1	2	0.014115	0	23	47	0.15224	5.8×10^{-7}	19	39	0.25956	5.87×10^{-7}
		x_6	1	2	0.015431	0	23	47	0.17212	5.81×10^{-7}	19	39	0.34396	5.8×10^{-7}
	100,000	x_1	1	2	0.029333	0	1	3	0.018075	0	1	3	0.027167	0
		x_2	1	2	0.029502	0	1	3	0.021229	0	10	21	0.20962	6.89×10^{-7}
		x_3	1	2	0.029413	0	1	3	0.029106	0	1	3	0.042574	0
		x_4	1	2	0.029978	0	11	23	0.15785	7.32×10^{-7}	1	3	0.029647	0
		x_5	1	2	0.024201	0	23	47	0.37503	8.2×10^{-7}	19	39	0.46268	8.31×10^{-7}
		x_6	1	2	0.023344	0	23	47	0.30268	8.28×10^{-7}	19	39	0.44689	8.24×10^{-7}

Table 4. *Cont.*

			TSSP				SGPM				TSGP			
Problem	**DIM**	**SP**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**	**ITER**	**FVAL**	**TIME**	**NORM**
P5	1000	x_1	3	5	0.026584	7.96×10^{-7}	22	45	0.012781	4.76×10^{-7}	20	41	0.02354	8.85×10^{-7}
		x_2	3	5	0.003513	8.26×10^{-7}	21	43	0.021446	9.78×10^{-7}	20	41	0.026418	9.18×10^{-7}
		x_3	3	5	0.003116	2.18×10^{-7}	20	41	0.011601	5.12×10^{-7}	19	39	0.02013	6.08×10^{-7}
		x_4	3	5	0.002986	8.24×10^{-7}	24	49	0.008455	1.26×10^{-7}	20	41	0.038547	9.16×10^{-7}
		x_5	3	5	0.003386	6.8×10^{-7}	26	53	0.015125	2.64×10^{-8}	20	41	0.021084	7.56×10^{-7}
		x_6	3	5	0.003238	6.8×10^{-7}	26	53	0.014889	2.67×10^{-8}	20	41	0.019351	7.56×10^{-7}
	50,000	x_1	6	8	0.10841	7.24×10^{-7}	18	37	0.28111	1.99×10^{-8}	22	45	0.63837	9.98×10^{-7}
		x_2	6	8	0.12008	7.52×10^{-7}	19	39	0.35252	5.32×10^{-7}	23	47	0.59083	4.14×10^{-7}
		x_3	4	6	0.091808	7.79×10^{-7}	17	35	0.24494	5.52×10^{-7}	21	43	0.61762	6.85×10^{-7}
		x_4	6	8	0.10356	7.52×10^{-7}	19	39	0.26199	1.04×10^{-8}	23	47	0.58276	4.14×10^{-7}
		x_5	6	8	0.1118	6.19×10^{-7}	22	45	0.42086	5.64×10^{-8}	22	45	0.68866	8.53×10^{-7}
		x_6	6	8	0.13042	6.19×10^{-7}	20	41	0.30457	2.21×10^{-7}	22	45	0.57108	8.53×10^{-7}
	100,000	x_1	7	9	0.3991	5.17×10^{-7}	17	35	0.53229	5.58×10^{-8}	23	47	1.4183	5.64×10^{-7}
		x_2	7	9	0.33016	5.37×10^{-7}	19	39	0.70642	7.53×10^{-7}	23	47	1.2936	5.85×10^{-7}
		x_3	5	7	0.22423	5.57×10^{-7}	17	35	0.53979	7.8×10^{-7}	21	43	1.0728	9.69×10^{-7}
		x_4	7	9	0.31926	5.37×10^{-7}	18	37	0.59493	2.92×10^{-8}	23	47	1.3459	5.85×10^{-7}
		x_5	6	8	0.25384	8.75×10^{-7}	19	39	0.69354	1.21×10^{-8}	23	47	1.2682	4.82×10^{-7}
		x_6	6	8	0.37171	8.75×10^{-7}	20	41	0.62058	3.13×10^{-7}	23	47	1.3007	4.82×10^{-7}
P6	1000	x_1	3	5	0.010955	1.07×10^{-7}	23	47	0.01257	6.93×10^{-7}	19	39	0.015919	4.69×10^{-7}
		x_2	9	11	0.006	8.56×10^{-9}	23	47	0.012384	9.68×10^{-7}	19	39	0.01716	5.74×10^{-7}
		x_3	4	6	0.006474	1.15×10^{-8}	6	13	0.002199	2.21×10^{-7}	13	27	0.009428	8.36×10^{-7}
		x_4	11	13	0.008502	1.26×10^{-7}	23	47	0.013556	9.78×10^{-7}	19	39	0.012728	5.69×10^{-7}
		x_5	10	12	0.010102	9.99×10^{-9}	7	15	0.004183	2.36×10^{-7}	19	39	0.010011	4.42×10^{-7}
		x_6	10	12	0.007011	7.27×10^{-9}	7	15	0.006891	2.93×10^{-7}	19	39	0.009542	4.06×10^{-7}
	50,000	x_1	3	5	0.061255	7.6×10^{-7}	24	49	0.23097	2.6×10^{-8}	21	43	0.32775	5.28×10^{-7}
		x_2	11	13	0.13565	2.66×10^{-7}	25	51	0.23841	1.83×10^{-8}	21	43	0.34316	6.46×10^{-7}
		x_3	4	6	0.062462	8.1×10^{-8}	7	15	0.065855	8.32×10^{-9}	15	31	0.25529	9.4×10^{-7}
		x_4	14	16	0.20446	1.62×10^{-8}	26	53	0.26012	8.69×10^{-7}	21	43	0.31348	6.46×10^{-7}
		x_5	10	12	0.14657	7.62×10^{-8}	8	17	0.071141	8.94×10^{-9}	21	43	0.34669	5.05×10^{-7}
		x_6	10	12	0.15597	7.06×10^{-8}	8	17	0.091229	8.84×10^{-9}	21	43	0.35675	5.27×10^{-7}
	100,000	x_1	4	6	0.11184	5.71×10^{-9}	24	49	0.57769	3.68×10^{-8}	21	43	0.62764	7.47×10^{-7}
		x_2	14	16	0.44177	1.79×10^{-7}	25	51	0.45188	2.59×10^{-8}	21	43	0.6427	9.14×10^{-7}
		x_3	4	6	0.20305	1.15×10^{-7}	7	15	0.14039	1.18×10^{-8}	16	33	0.51687	5.3×10^{-7}
		x_4	14	16	0.30629	1.42×10^{-7}	27	55	0.49475	6.18×10^{-7}	21	43	0.65503	9.14×10^{-7}
		x_5	9	11	0.18947	2.07×10^{-8}	8	17	0.13416	1.26×10^{-8}	21	43	0.67669	7.15×10^{-7}
		x_6	9	11	0.25267	2.25×10^{-8}	8	17	0.16038	1.25×10^{-8}	21	43	0.67886	7.17×10^{-7}

4. Applications in Image Deblurring

In this section, we apply the proposed Algorithm 1 to solve problems arising from compressive sensing, particularly image deblurring. Consider the following least square problem with ℓ_1-regularization term

$$\min_x \frac{1}{2}\|y - Ex\|_2^2 + \mu\|x\|_1, \tag{46}$$

where $x \in \mathbb{R}^n$ is the underlying images, $y \in \mathbb{R}^k$ is the observed images, $E \in \mathbb{R}^{k \times n}$ $(k << n)$, linear operator, is an $m \times n$ blurring matrix, and the parameter $\mu > 0$. Problem (46) is of great importance because it appears in many areas of applications arising from compressive sensing. Recently, problem (46) has been investigated by many researchers and different kinds of iterative algorithms have been proposed in the literature [3,32–35]. Many algorithms for solving (46) fall into two categories namely: algorithms that required differentiability assumption and algorithms that are derivative free. Since ℓ_1-norm is a nonsmooth function, algorithms that require the assumption of differentiabilty are not suitable for problem (46) in its original form. Consequently, either $\|x\|_1$ is approximated with some smooth function or problem (46) is reformulated into an equivalent problem. For instance, Figueiredo et al. [3] translate problem (46) into convex quadratic program as follows. For any $x \in \mathbb{R}^n$, we can find some vectors, say $u, v \in \mathbb{R}^n$ such that

$$x = u - v, \qquad u \geq 0, \quad v \geq 0,$$

where $u_i = \max\{0, x_i\}$, $v_i = \max\{0, -x_i\}$ for all $i = 1, 2, ..., n$. Thus, we can write $\|x\|_1 = e_n^T(u+v)$, where e_n is an $n-$dimensional vector with all elements one. Therefore, we can rewrite problem (46) as

$$\begin{aligned} &\min_{u,v} \frac{1}{2}\|y - E(u-v)\|_2^2 + \mu e_n^T(u+v). \\ &\text{s.t } \ u \geq 0, \ \ v \geq 0 \end{aligned} \tag{47}$$

Furthermore, if we let $q = [u \ \ v]^T$, then from Reference [3], we can write (47) as the following

$$\begin{aligned} &\min_{q} \frac{1}{2} q^T G q + c^T q, \\ &\text{s.t } \ q \geq 0, \end{aligned} \tag{48}$$

where $c = \mu e_{2n} + \begin{pmatrix} -b \\ b \end{pmatrix}$, $\quad b = E^T y, \quad G = \begin{pmatrix} E^T E & -E^T E \\ -E^T E & E^T E \end{pmatrix}$. It is not difficult to see that G is a positive semi-definite matrix.

In Reference [36], the resulted constrained minimization problem (48) is further translated into the following linear variable inequality problem

$$\begin{cases} \text{Find } q \in \mathbb{R}^n \text{ such that} \\ \langle Gq + c, \ q' - q \rangle \geq 0, \ \ \forall\, q' \geq 0. \end{cases} \tag{49}$$

Since the feasible region of q is $\mathbb{R}^n$, problem (49) is equivalent to the following linear complementary problem

$$\begin{cases} \text{Find } q \in \mathbb{R}^n \text{ such that} \\ q \geq 0, \\ Gq + c \geq 0, \\ \langle Gq + c, \ q \rangle = 0. \end{cases} \tag{50}$$

We can see that the point $q \in \mathbb{R}^n$ is a solution of the above linear complementary problem (50) if and only if it satisfies the following system of nonlinear equation

$$F(q) := \min\{q, Gq + c\} = 0. \tag{51}$$

The mapping F is a vector-valued and the "min" operator denotes the componentwise minimum of two vectors. Interestingly, Lemma 3 of Reference [37] and Lemma 2.2 of Reference [36] show that the mapping F satisfies Assumption 1 (ii) i.e., is Lipschitzian continuity and monotonicity. Therefore our proposed TSSP algorithm can be applied to solve it.

Image Deblurring Experiment

We tested the performance of the two-step TSSP algorithm in restoring some blurred images in comparison with the one-step spectral gradient method for ℓ_1 problems in compressed sensing (SGCS) [36]. The images used for the experiment are the well-known gray test images namely: Lena, House, Pepper, Camera man and Barbara where the size of each image is given in Table 5. The following metrics are used to assess the performance and quality of restoration by each algorithm tested: number of iteration (ITER), CPU time in seconds (TIME), signal-to-noise-ratio (SNR) which is defined as

$$\text{SNR} = 20 \times \log_{10}\left(\frac{\|\bar{x}\|}{\|x - \bar{x}\|}\right),$$

and the structural similarity (SSIM) index that measure the similarity between the original image and the restored image [38] for each of the two experiments. The MATLAB implementation of the SSIM index can be obtained at http://www.cns.nyu.edu/~lcv/ssim/. To achieve fairness in comparison, each code was run from same initial point $x_0 = E^T y$ and terminate when

$$\frac{|f_k - f_{k-1}|}{|f_{k-1}|} < 10^{-5},$$

where f_k is the merit function evaluation at x_k, with $f(x_k) = \frac{1}{2}\|y - Ex_k\|_2^2 + \mu\|x_k\|_1$. The parameters used for both TSSP and SGCS in this experiment come from Reference [36] except for $c = 1$ in the line search (1) and $\alpha_k = (0.999^k) \times (10^5 + \|F(x_0)\|^2)$.

The original, blurred and restored images by each algorithm are given in Figure 4. The two tested algorithms restored the blurred images successfully with different speed and level of quality. The results of the restoration by each algorithm are reported in Table 5. We see from Table 5 that TSSP restored all the five images with less ITER. Taking TIME into consideration, we see that though the SGCS is faster in restoring two of the images (i.e., Camera man and Barbara), TSSP is faster in restoring the remaining three images (i.e., Lena, House and Pepper). In addition, the SNR and SSIM values recorded by each algorithm revealed that TSSP restored the five blurred images with slightly better quality than SGCS except for Camera man. Taking everything together, this experiment shows that the two-step TSSP can deal with the ℓ_1 regularization problems effectively and can be a favourable alternative for image reconstruction.

Table 5. Test results for TSSP and SGCS in image restoration.

		TSSP				SGCS			
Image	**Size**	**ITER**	**TIME(s)**	**SNR**	**SSIM**	**ITER**	**TIME(s)**	**SNR**	**SSIM**
Lena	256 × 256	113	8.84	24.25	0.90	218	11.09	23.70	0.90
House	256 × 256	121	16.53	22.86	0.87	235	17.47	23.61	0.87
Pepper	256 × 256	100	8.69	27.58	0.89	167	12.05	27.02	0.89
Camera man	256 × 256	21	2.20	20.33	0.84	28	2.19	20.56	0.84
Barbara	512 × 512	22	12.69	19.16	0.76	23	11.08	19.16	0.76

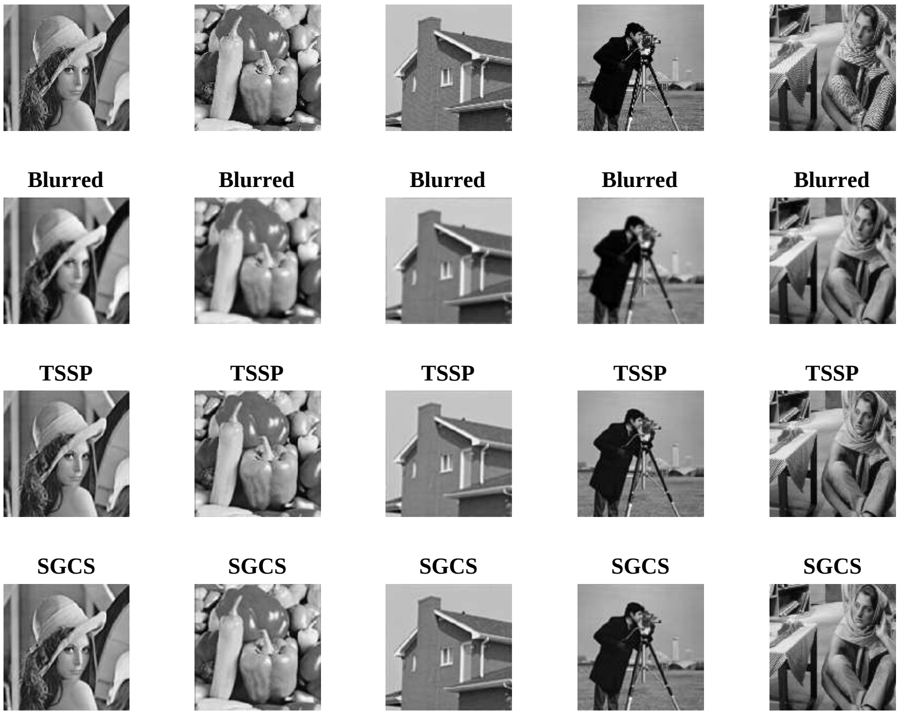

Figure 4. The original images (first row), the blurred images (second row), the restored images by methods TSSP (third row) and SGCS (last row).

5. Conclusions

This paper presents an efficient derivative-free iterative algorithm called TSSP for nonlinear monotone equations. It utilizes a two-step approach that incorporates the well-known BB parameters with a projection strategy. We showed that the TSSP converges globally under the Lipschitzian and monotonicity assumptions. Also, we proposed a new line search that is more general than the one proposed by Cheng in Reference [24], able to include the line search by Cheng as a special case. Preliminary numerical results reported in Table 2 shows that the parameter c introduced in the new line search defined by (1) may have some effect on the performance of the proposed algorithm. Numerical results presented revealed that the proposed TSSP algorithm has computational advantage and performs better than the two existing algorithms in References [17,25]. These results indicate that the two-step BB-like algorithm is superior to the existing one-step BB-like algorithms, especially on solving nonlinear equations. It is worth emphasizing that the TSSP algorithm improves existing results on monotone nonlinear equations. The results obtained in Section 3 show that the TSSP algorithm possessed excellent numerical performances with evidence of efficiently solving all the test problems considered with minimal number of iterations and function evaluations. The numerical results reported from the experiments of deblurring two-dimensional images from their limited measurements have shown that the two-step algorithm TSSP competes favorably with the one-step SGCS algorithm. Future work includes the extension of TSSP algorithm on different forms of optimization frameworks and applications such as nonlinear least-squares problems [39,40], neural networks [41,42] and machine learning [43,44].

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

Funding: The authors acknowledge the financial support provided by the Petchra Pra Jom Klao Doctoral Scholarship for Ph.D. program of King Mongkut's University of Technology Thonburi (KMUTT) and Center of Excellence in Theoretical and Computational Science (TaCS-CoE), KMUTT. The first author was supported by the Petchra Pra Jom Klao Ph.D. Research Scholarship from King Mongkut's University of Technology Thonburi (Contract No. 42/2560).

Acknowledgments: The authors would like to thank the anonymous referees for their useful comments. This project was supported by Center of Excellence in Theoretical and Computational Science (TaCS-CoE) Center under Computational and Applied Science for Smart Innovation research Cluster (CLASSIC), Faculty of Science, KMUTT. This work was done while the first author visits Yunnan University of Finance and Economics, Kunming China.

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

References

1. Iusem, N.A.; Solodov, V.M. Newton-type methods with generalized distances for constrained optimization. *Optimization* **1997**, *41*, 257–278. [CrossRef]
2. Zhao, Y.B.; Li, D. Monotonicity of fixed point and normal mappings associated with variational inequality and its application. *SIAM J. Optim.* **2001**, *11*, 962–973. [CrossRef]
3. Figueiredo, M.A.T.; Nowak, R.D.; Wright, S.J. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE J. Sel. Top. Signal Process.* **2007**, *1*, 586–597. [CrossRef]
4. Barzilai, J.; Borwein, J.M. Two-point step size gradient methods. *IMA J. Numer. Anal.* **1988**, *8*, 141–148. [CrossRef]
5. Dai, Y.H.; Al-Baali, M.; Yang, X. A Positive Barzilai–Borwein-Like Stepsize and an Extension for Symmetric Linear Systems. *Numer. Anal. Optim.* **2015**, *41*, 59–75.
6. Dai, Y.H; Huang, Y.; Liu, X.W. A family of spectral gradient methods for optimization. *Comput. Optim. Appl.* **2019**, *74*, 43–65. [CrossRef]

7. Birgin, E.G.; Martínez, J.M.; Raydan, M. Spectral projected gradient methods: Review and perspectives. *J. Stat. Softw.* **2014**, *60*, 1–21. [CrossRef]
8. Fletcher, R. On the Barzilai-Borwein method. *Optim. Control Appl.* **2005**, *60*, 235–256.
9. Dai, Y.H. A new analysis on the Barzilai-Borwein gradient method. *J. Oper. Res. Soc. China* **2013**, *41*, 187–198. [CrossRef]
10. Dai, Y.H.; Kou, C. A Barzilai-Borwein conjugate gradient method. *Sci. China Math.* **2016**, *59*, 1511–1524. [CrossRef]
11. Raydan, M. On the Barzilai and Borwein choice of steplength for the gradient method. *IMA J. Numer. Anal.* **1993**, *13*, 321–326. [CrossRef]
12. Raydan, M. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM J. Optim.* **1997**, *7*, 26–33. [CrossRef]
13. La Cruz, W.; Raydan, M. Nonmonotone spectral methods for large-scale nonlinear systems. *Optim. Methods Softw.* **2003**, *18*, 583–599. [CrossRef]
14. La Cruz, W.; Martínez, J.M.; Raydan, M. Spectral residual method without gradient information for solving large-scale nonlinear systems of equations. *Math. Comput.* **2006**, *75*, 1429–1448. [CrossRef]
15. Solodov, M.V.; Svaiter, B.F. A globally convergent inexact Newton method for systems of monotone equations. In *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*; Springe: Boston, MA, USA, 1998; pp. 355–369.
16. Zhang, L.; Zhou, W. Spectral gradient projection method for solving nonlinear monotone equations. *J. Comput. Appl. Math.* **2006**, *96*, 478–484. [CrossRef]
17. Yu, Z.; Lin, J.; Sun, J.; Xiao, Y.; Liu, L.; Li, Z. Spectral gradient projection method for monotone nonlinear equations with convex constraints. *Appl. Numer. Math.* **2009**, *59*, 2416–2423. [CrossRef]
18. Wang, C.; Wang, Y.; Xu, C. A projection method for a system of nonlinear monotone equations with convex constraints. *Math. Methods Oper. Res.* **2007**, *66*, 33–46. [CrossRef]
19. Mohammad, H.; Abubakar, A.B. A positive spectral gradient-like method for large-scale nonlinear monotone equations. *Bull. Comput. Appl. Math.* **2017**, *5*, 97–113.
20. Awwal, A.M.; Kumam, P.; Abubakar, A.B.; Wakili, A.; Pakkaranang, N. A New Hybrid Spectral Gradient Projection Method for Monotone System of Nonlinear Equations with Convex Constraints. *Thai J. Math.* **2018**, *16*, 125–147.
21. Abubakar, A.B.; Kumam, P.; Mohammad, H. A note on the spectral gradient projection method for nonlinear monotone equations with applications. *Comput. Appl. Math.* **2020**. [CrossRef]
22. Mann, W.R. Mean value methods in iteration. *Proc. Am. Math. Soc.* **1953**, *4*, 506–510. [CrossRef]
23. Ishikawa, S. Fixed points by a new iteration method. *Proc. Am. Math. Soc.* **1974**, *44*, 147–150. [CrossRef]
24. Cheng, W. A PRP type method for systems of monotone equations. *Math. Comput. Model.* **2009**, *50*, 15–20. [CrossRef]
25. Liu, J.; Duan, Y. Two spectral gradient projection methods for constrained equations and their linear convergence rate. *J. Inequal. Appl.* **2015**, *1*, 8. [CrossRef]
26. Dolan, E.D.; Moré, J.J. Benchmarking optimization software with performance profiles. *Math. Program. Ser.* **2002**, *91*, 201–213. [CrossRef]
27. La Cruz, W.; Martínez, J.M.; Raydan, M. *Spectral Residual Method without Gradient Information for Solving Large-Scale Nonlinear Systems: Theory and Experiments*; Technical Report RT-04-08; Universidad Central de Venezuela: Caracas, Venezuela, 2004; p. 37. Available online: http://kuainasi.ciens.ucv.ve/mraydan/download_papers/TechRep.pdf (accessed on 15 April 2020).
28. Abubakar, A.B.; Kumam, P.; Mohammad, H.; Awwal, A.M. An Efficient Conjugate Gradient Method for Convex Constrained Monotone Nonlinear Equations with Applications *Mathematics* **2019**, *7*, 767.
29. La Cruz, W. A spectral algorithm for large-scale systems of nonlinear monotone equations. *Numer. Algorithms* **2017**, *76*, 1109–1130. [CrossRef]
30. Awwal, A.M.; Kumam, P.; Abubakar, A.B. A modified conjugate gradient method for monotone nonlinear equations with convex constraints. *Appl. Numer. Math.* **2019**, *145*, 507–520. [CrossRef]
31. Yu, G.; Niu, S.; Ma, J. Multivariate spectral gradient projection method for nonlinear monotone equations with convex constraints *J. Ind. Manag. Optim.* **2013**, *9*, 117–129. [CrossRef]

32. Daubechies, I.; Defrise, M.; De Mol, C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math. A J. Issued Courant Inst. Math. Sci.* **2004**, *57*, 1413–1457. [CrossRef]
33. Hale, E.T.; Yin, W.; Zhang, Y. *A Fixed-Point Continuation Method for ℓ_1-Regularized Minimization with Applications to Compressed Sensing*; CAAM TR07-07; Rice University: Houston, TX, USA, 2007; Volume 43.
34. Beck, A.; Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2009**, *2*, 183–202. [CrossRef]
35. Wright, S.J.; Nowak, R.D.; Figueiredo, M.A.T. Sparse reconstruction by separable approximation. *IEEE Trans. Signal Process.* **2009**, *57*, 2479–2493. [CrossRef]
36. Xiao, Y.; Wang, Q.; Hu, Q. Non-smooth equations based methods for l_1-norm problems with applications to compressed sensing. *Nonlinear Anal. Theory Methods Appl.* **2011**, *74*, 3570–3577. [CrossRef]
37. Pang, J.S. Inexact Newton methods for the nonlinear complementarity problem. *Math. Program.* **1986**, *36*, 54–71. [CrossRef]
38. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef]
39. Awwal, A.M.; Kumam, P.; Mohammad, H. Iterative algorithm with structured diagonal Hessian approximation for solving nonlinear least squares problems. *arXiv* **2020**, arXiv:2002.01871.
40. Mohammad, H.; Waziri, M.Y. Structured two-point stepsize gradient methods for nonlinear least squares. *J. Optim. Theory Appl.* **2019**, *181*, 298–317. [CrossRef]
41. Li, G.; Zeng, Z. A neural-network algorithm for solving nonlinear equation systems. In Proceedings of the 2008 International Conference on Computational Intelligence and Security, Suzhou, China, 13–17 December 2008; Volume 1, pp. 20–23.
42. Li, G.; Zeng, Z. A new neural network for solving nonlinear projection equations. *Neural Netw.* **2007**, *20*, 577–589.
43. Zhu, Z.; Wang, H.; Zhang, B. A spectral conjugate gradient method for nonlinear inverse problems. *Inverse Probl. Sci. Eng.* **2018**, *26*, 1561–1589. [CrossRef]
44. Bottou, L.; Cutis, F.E.; Nocedal, J. Optimization Methods for Large-Scale Machine Learning. *SIAM Rev.* **2018**, *60*, 223–311. [CrossRef]