Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 17, No. 4, pp. 547–564, 2021

A SMOOTHING AND SCALING FLETCHER-REEVES TYPE CONJUGATE GRADIENT METHOD FOR SYSTEMS OF NONSMOOTH EQUATIONS

Yasushi Narushima* and Hiroshi Yabe

Abstract: We propose a Fletcher-Reeves (FR) type conjugate gradient method for solving systems of nonsmooth equations. The proposed method is based on a smoothing technique and the scaling FR method given by Zhang, Zhou, and Li (2006). The generated search direction is a descent direction for the merit function and we do not need any matrix to compute the search direction. We give an algorithm of the proposed method with the Armijo line search condition and show its global convergence property. Finally, we provide some numerical results.

Key words: *system of nonsmooth equations, smoothing method, scaling conjugate gradient method, global convergence*

Mathematics Subject Classification: *90C30, 90C06, 65H10*

1 Introduction

In this paper, we consider the following system of nonsmooth equations:

$$F(x) = 0, \qquad F : \boldsymbol{R}^n \to \boldsymbol{R}^n. \tag{1.1}$$

Throughout this paper, we assume that F is continuous, but not necessarily differentiable, and (1.1) has at least one solution. This system has many applications. For example, the variational inequality and the complementarity problems, involving many equilibrium problems in economics, can be reformulated as (1.1) (see [10] for example). Accordingly, Newton-like methods for solving (1.1) have been extensively studied (see [15, 23, 26–28, 30–32], for example). However, there has been a recent focus on matrix-free methods (such as residual methods and conjugate gradient methods) for solving large-scale systems of nonlinear equations (including the smooth cases and constrained cases). La Cruz et al. [19] proposed a spectral residual method with a nonmonotone line search technique, which was given by La Cruz and Raydan [18], for solving large-scale systems of smooth equations. By using a modification of La Cruz et al.'s line search, some authors developed conjugate gradient methods and variants [6, 20, 36]. Yuan et al. [35] treated a symmetric system of nonlinear equations, meaning that the Jacobian matrix of F is symmetric, and gave a conjugate gradient method with the Armijo line search. In their numerical experiments, they

*Corresponding author.

solved image restoration problems as an application of the symmetric system. Meanwhile, many researchers have dealt with monotone systems of nonsmooth equations (namely, F is monotone). Inspired by the projection method in [33], Zhan and Zhou [37] proposed a spectral residual projection method. Following their research, various conjugate gradient methods with the same projection technique have been proposed [4, 7, 11, 21, 34]. Moreover, in [2, 3], the authors presented conjugate gradient methods with the projection technique to solve constrained monotone systems of nonsmooth equations and applied the proposed methods to signal recovery and image restoration problems. Recent research of matrix-free methods for solving systems of equations can be categorized into two classes. The first is smooth and not necessarily monotone systems, and the second is nonsmooth and monotone systems. We emphasize that there are few existing studies on nonsmooth and nonmonotone systems.

To solve systems of nonsmooth equations, gradient methods such as Newton's method or Newton-like methods cannot be directly applied to problem (1.1). Thus, smoothing methods based on the following smoothing function are often used.

Definition 1.1. A function $\tilde{F} : \boldsymbol{R} \times \boldsymbol{R}^n \to \boldsymbol{R}^n$ is a smoothing function of F if $\tilde{F}$ is continuously differentiable on $\boldsymbol{R}_{++} \times \boldsymbol{R}^n$ and satisfies

$$\lim_{t \to +0} \tilde{F}(t, x) = \tilde{F}(0, x) = F(x)$$

for any x, where $\boldsymbol{R}_{++} = \{t \in \boldsymbol{R} \mid t > 0\}$.

Defining a function $H : \boldsymbol{R} \times \boldsymbol{R}^n \to \boldsymbol{R}^{1+n}$ by

$$H(t, x) = \begin{pmatrix} t \\ \tilde{F}(t, x) \end{pmatrix},$$

we solve the system of equations $H(t, x) = 0$ instead of (1.1). Moreover, we define a merit function $\Psi : \boldsymbol{R} \times \boldsymbol{R}^n \to \boldsymbol{R}$ by

$$\Psi(t, x) = \frac{1}{2}\|H(t, x)\|^2 = \frac{1}{2}\{t^2 + \|\tilde{F}(t, x)\|^2\}, \tag{1.2}$$

where $\|\cdot\|$ denotes the ℓ_2-norm. Then, (1.1) is equivalent to finding a global minimizer of the unconstrained optimization problem:

$$\min \Psi(t, x). \tag{1.3}$$

Note that Ψ is continuously differentiable on $\boldsymbol{R}_{++} \times \boldsymbol{R}^n$, but not necessarily continuously differentiable on the other region (namely, $t \leq 0$). Many researchers have proposed Newton's method or Newton-like methods based on (1.2) and (1.3), as those are reviewed in [32]. However, these methods need to store some matrices, and so cannot necessarily be applied to large-scale problems.

To develop an algorithm for solving large-scale problems, some smoothing conjugate gradient methods have been proposed by incorporating the smoothing technique into conjugate gradient methods for usual unconstrained optimization. Narushima [23] proposed a smoothing three-term Polak-Ribière-Polyak (PRP) type conjugate gradient (STPRP) method, based on the three-term PRP type conjugate gradient method given by Zhang, Zhou, and Li [38] which solves smooth unconstrained optimization problems. As scaling conjugate gradient methods are efficient for solving large-scale smooth unconstrained optimization problems, it is expected that smoothing and scaling conjugate gradient methods

can be efficient for solving nonsmooth equations (1.1). From this perspective, Narushima, Ootani, and Yabe [27] gave a smoothing and scaling PRP type conjugate gradient (SSPRP) method, based on the scaling PRP type conjugate gradient method given by Cheng [5]. However, the parameter of the SSPRP method includes a term whose denominator can be close to zero, causing numerical instability. Thus, we consider another type of smoothing and scaling conjugate gradient method such that a denominator of its parameter does not tend to zero. For this purpose, we propose a smoothing and scaling conjugate gradient method based on the scaling Fletcher-Reeves (FR) type conjugate gradient method by Zhang, Zhou, and Li [39].

This paper is organized as follows. In Section 2, we review usual conjugate gradient methods for solving smooth unconstrained optimization problems and smoothing conjugate gradient methods (namely, STPRP and SSPRP methods) for solving (1.1). In Section 3, we give an algorithm of our method. In Section 4, we prove the global convergence property of the proposed method. Finally, in Section 5, we present preliminary numerical results.

2 Preliminaries

In this section, we first recall conjugate gradient (CG) methods for solving the smooth unconstrained optimization problem:

$$\min f(z),$$

where $f : \boldsymbol{R}^\ell \to \boldsymbol{R}$ is a continuously differentiable function and its gradient ∇f is available. CG methods are iterative methods of the form:

$$z_{k+1} = z_k + \alpha_k d_k, \qquad d_k = \begin{cases} -\nabla f(z_k), & k = 0, \\ -\nabla f(z_k) + \beta_k d_{k-1}, & k \geq 1, \end{cases}$$

where $z_k \in \boldsymbol{R}^\ell$ is the kth approximation to a solution, α_k is a positive step size, and $d_k \in \boldsymbol{R}^\ell$ is a search direction. Since CG methods do not need to store matrices, they have been explored for solving large-scale unconstrained optimization problems. Choices of β_k are known to affect the numerical performance of the method. Well-known formulas for β_k are the Hestenes-Stiefel (HS) [16], Fletcher-Reeves (FR) [12], Polak-Ribière-Polyak (PRP) [29], and Dai-Yuan (DY) [8] formulas, which are respectively given by

$$\beta_k^{HS} = \frac{\nabla f(z_k)^T \widehat{y}_{k-1}}{d_{k-1}^T \widehat{y}_{k-1}}, \qquad \beta_k^{FR} = \frac{\|\nabla f(z_k)\|^2}{\|\nabla f(z_{k-1})\|^2},$$
$$\beta_k^{PRP} = \frac{\nabla f(z_k)^T \widehat{y}_{k-1}}{\|\nabla f(z_{k-1})\|^2}, \qquad \beta_k^{DY} = \frac{\|\nabla f(z_k)\|^2}{d_{k-1}^T \widehat{y}_{k-1}},$$

where $\widehat{y}_{k-1} = \nabla f(z_k) - \nabla f(z_{k-1})$. Recent development of CG methods and their global convergence properties is reviewed in [14, 24]. A weakness of CG methods is that many CG methods do not necessarily satisfy the descent condition $\nabla f(z_k)^T d_k < 0$. To overcome this weakness, some researchers recently proposed three-term or scaling CG methods that always generate descent search directions (for example, see [1,5,13,17,22,25,38–40]). Zhang, Zhou, and Li proposed a three-term PRP method in [38] and a scaling FR method in [39], respectively given by

$$d_k = -\nabla f(z_k) + \beta_k^{PRP} d_{k-1} - \frac{\nabla f(z_k)^T d_{k-1}}{\|\nabla f(z_{k-1})\|^2} \widehat{y}_{k-1}, \tag{2.1}$$

$$d_k = -\frac{d_{k-1}^T \widehat{y}_{k-1}}{\|\nabla f(z_{k-1})\|^2} \nabla f(z_k) + \beta_k^{FR} d_{k-1}. \tag{2.2}$$

Cheng [5] gave the following modified PRP method:

$$\begin{aligned} d_k &= -\nabla f(z_k) + \beta_k^{PRP} \left(I - \frac{\nabla f(z_k) \nabla f(z_k)^T}{\|\nabla f(z_k)\|^2} \right) d_{k-1} \\ &= -\left(1 + \beta_k^{PRP} \frac{\nabla f(z_k)^T d_{k-1}}{\|\nabla f(z_k)\|^2} \right) \nabla f(z_k) + \beta_k^{PRP} d_{k-1}. \end{aligned} \tag{2.3}$$

It can be easily verified that these methods always satisfy the sufficient descent condition in the sense that $\nabla f(z_k)^T d_k = -\|\nabla f(z_k)\|^2 < 0$ whenever $\nabla f(z_k) \neq 0$. We note that (2.2) can be written as

$$d_k = -\left(1 + \frac{\nabla f(z_k)^T d_{k-1}}{\|\nabla f(z_{k-1})\|^2} \right) \nabla f(z_k) + \beta_k^{FR} d_{k-1}. \tag{2.4}$$

by using the relation $\nabla f(z_k)^T d_k = -\|\nabla f(z_k)\|^2$. This form will be referred to in subsequent sections.

Next, we review two types of smoothing CG methods for solving (1.3) to therefore solve (1.1). To this end, we introduce some notations and relations. For any Fréchet-differentiable mapping $G : \boldsymbol{R}^n \to \boldsymbol{R}^m$, we denote its transposed Jacobian at $x \in \boldsymbol{R}^n$ by $\nabla G(x) \in \boldsymbol{R}^{n \times m}$. Note that, when $m = 1$, the term $\nabla G(x) \in \boldsymbol{R}^n$ refers to the gradient vector of G at x. The gradient of a smoothing function $\tilde{F}$ of F is given by

$$\nabla \tilde{F}(t, x) = \begin{pmatrix} \nabla_t \tilde{F}(t, x) \\ \nabla_x \tilde{F}(t, x) \end{pmatrix}.$$

Also, by (1.2), we have

$$\nabla \Psi(t, x) = \begin{pmatrix} \nabla_t \Psi(t, x) \\ \nabla_x \Psi(t, x) \end{pmatrix} = \begin{pmatrix} t + \nabla_t \tilde{F}(t, x) \tilde{F}(t, x) \\ \nabla_x \tilde{F}(t, x) \tilde{F}(t, x) \end{pmatrix}. \tag{2.5}$$

Note that $\nabla_t \tilde{F}(t, x) \in \boldsymbol{R}^{1 \times n}$ is a row vector, and so $\nabla_t \tilde{F}(t, x) \tilde{F}(t, x)$ is a scalar, while $\nabla_x \tilde{F}(t, x) \tilde{F}(t, x) \in \boldsymbol{R}^n$ is a column vector. We often write $(t, x) \in \boldsymbol{R} \times \boldsymbol{R}^n$ instead of $(t, x^T)^T \in \boldsymbol{R}^{1+n}$. For simplicity, we set $v = (t, x)$.

Narushima [23] proposed the STPRP method, which is an iterative method of the form:

$$v_{k+1} = v_k + \alpha_k d_k, \tag{2.6}$$

where $v_k \in \boldsymbol{R}^{1+n}$ is the kth approximation to a solution of (1.3), α_k is a positive step size, and $d_k \in \boldsymbol{R}^{1+n}$ is a search direction. Similar to $v = (t, x)$, we use the symbol $v_k = (t_k, x_k)$. In addition, we often use the conventional abbreviation:

$$\tilde{F}_k = \tilde{F}(v_k),$$

and we adopt the same manner for the other functions. The search direction in (2.6) is given by

$$d_k = \begin{pmatrix} \bar{t} \gamma_k - t_k \\ \tilde{d}_k \end{pmatrix}, \tag{2.7}$$

where

$$\gamma_k = \gamma(v_k), \qquad \gamma(v) = \bar{\gamma} \min\{1, \Psi(v)\}, \tag{2.8}$$

and $\bar{\gamma}$ and $\bar{t}$ are positive constants such that $\bar{\gamma}\bar{t} < 1$. Here, $\bar{t}\gamma_k - t_k \in \boldsymbol{R}$ is a search direction associated with the variable t, and $\tilde{d}_k \in \boldsymbol{R}^n$ is a search direction associated with the variable x. Moreover, by modifying the three-term PRP method (2.1), $\tilde{d}_k \in \boldsymbol{R}^n$ is defined by the following:

If $\nabla_x \Psi_k = 0$, then

$$\tilde{d}_k = 0. \tag{2.9}$$

Else,

$$\tilde{d}_k = \begin{cases} -\zeta_k \nabla_x \Psi_k & k = 0, \\ -\zeta_k \nabla_x \Psi_k + \beta_k^{PRP'} \tilde{d}_{k-1} - \dfrac{\nabla_x \Psi_k^T \tilde{d}_{k-1}}{\|\nabla \Psi_{k-1}\|^2} y_{k-1} & k \geq 1, \end{cases} \tag{2.10}$$

where $\eta \in (0, 1)$ is a constant,

$$\zeta_k = \begin{cases} 1, & \eta \|\nabla_x \Psi_k\|^2 \geq \nabla_t \tilde{F}_k \tilde{F}_k (\bar{t}\gamma_k - t_k), \\ 1 + \dfrac{\nabla_t \tilde{F}_k \tilde{F}_k (\bar{t}\gamma_k - t_k)}{\|\nabla_x \Psi_k\|^2}, & \text{otherwise}, \end{cases} \tag{2.11}$$

$$\beta_k^{PRP'} = \frac{\nabla_x \Psi_k^T y_{k-1}}{\|\nabla \Psi_{k-1}\|^2},$$

and $y_{k-1} = \nabla_x \Psi_k - \nabla_x \Psi_{k-1}$. Narushima [23] showed the global convergence of the STPRP method with an Armijo type line search.

Since scaling CG methods are often very efficient for large-scale problems, Narushima, Ootani and Yabe [27] presented the SSPRP method by proposing another $\tilde{d}_k$ based on the scaling PRP method (2.3). Specifically, they used the following $\tilde{d}_k$ instead of (2.10):

$$\tilde{d}_k = \begin{cases} -\zeta_k \nabla_x \Psi_k, & k = 0. \\ -\left(\zeta_k + \beta_k^{PRP'} \dfrac{\nabla_x \Psi_k^T \tilde{d}_{k-1}}{\|\nabla_x \Psi_k\|^2} \right) \nabla_x \Psi_k + \beta_k^{PRP'} \tilde{d}_{k-1}, & k \geq 1. \end{cases} \tag{2.12}$$

They showed the global convergence of the SSPRP method with an Armijo type line search, which is same as STPRP. In their numerical comparison of these methods, the SSPRP method could not outperform the STPRP method. We suggest this is because $\|\nabla_x \Psi_k\|^2$ is included in the denominator of (2.12), and $\|\nabla_x \Psi_k\|^2$ can approach zero, even if $\|\nabla_x \Psi_k\|^2 \neq 0$ for all k. Thus, the method could be unstable. Note that $\|\nabla_x \Psi_k\|^2$ is also included in ζ_k, but $\zeta_k = 1$ is almost always chosen in our numerical experiments, and so we expect that $\|\nabla_x \Psi_k\|^2$ in ζ_k will not be a major cause of numerical instability. Considering the above arguments, we propose a smoothing and scaling CG method that does not involve $\|\nabla_x \Psi_k\|^2$ in the denominators in $\tilde{d}_k$.

3 Proposed Method

In this section, we develop a new smoothing and scaling CG method with the same framework as the STPRP method. Specifically, based on the scaling FR method (2.4), we propose the following $\tilde{d}_k$ instead of (2.10):

$$\tilde{d}_k = \begin{cases} -\zeta_k \nabla_x \Psi_k & k = 0, \\ -(\zeta_k + \theta_k) \nabla_x \Psi_k + \beta_k \tilde{d}_{k-1} & k \geq 1, \end{cases} \tag{3.1}$$

where

$$\beta_k = \frac{\|\nabla_x \Psi_k\|^2}{\|\nabla \Psi_{k-1}\|^2}, \quad \text{and} \quad \theta_k = \frac{\nabla_x \Psi_k^T \tilde{d}_{k-1}}{\|\nabla \Psi_{k-1}\|^2}. \tag{3.2}$$

We now introduce the set:

$$\Omega = \{v \mid t \geq \bar{t}\gamma(v)\}, \tag{3.3}$$

which is originally given in [31]. Note that if $\{v_k\} \subset \Omega$, then definition (2.8) yields the following statements:

- The search direction associated with the variable t is nonpositive, namely,

$$\bar{t}\gamma_k - t_k \leq 0 \tag{3.4}$$

  holds.

- If $\Psi_k \neq 0$ holds, then we have $t_k > 0$ for all k.

- If t_k approaches zero, then Ψ_k also approaches zero.

Therefore, we construct our algorithm such that the generated sequence $\{v_k\}$ is included in Ω. To establish $\{v_k\} \subset \Omega$, we need relations $0 < t_{k+1} \leq t_k$ and $\Psi_{k+1} < \Psi_k$. The next proposition is proved in [23, Proposition 2.1].

Proposition 3.1. *Assume that $v_k \in \Omega$ and $t_k > 0$. Then $0 < t_k + \alpha(\bar{t}\gamma_k - t_k) \leq t_k$ holds for any $\alpha \in (0, 1]$.*

Note that $\tilde{F}(v_k) \neq 0$ was assumed in [23, Proposition 2.1] but it was not used in the proof. Proposition 3.1 implies that $0 < t_{k+1} \leq t_k$ holds for any step size $\alpha_k \in (0, 1]$. To establish $\Psi_{k+1} < \Psi_k$, it is important that the search direction (2.7) with (2.9) and (3.1) is a descent search direction of the merit function, namely, $\nabla \Psi_k^T d_k < 0$ for all k. The next lemma is useful to show that the search direction becomes a descent direction for the merit function.

Lemma 3.2. *Assume that $\nabla_x \Psi_k \neq 0$. Then, $\tilde{d}_k$ in (3.1) satisfies*

$$\nabla_x \Psi_k^T \tilde{d}_k = -\zeta_k \|\nabla_x \Psi_k\|^2. \tag{3.5}$$

Proof. When $k = 0$, we immediately have (3.5). Thus, we consider the case $k \geq 1$. Multiplying both sides of (3.1) by $\nabla_x \Psi_k^T$ from the left, it follows from (3.2) that

$$\nabla_x \Psi_k^T \tilde{d}_k = -\left(\zeta_k + \frac{\nabla_x \Psi_k^T \tilde{d}_{k-1}}{\|\nabla \Psi_{k-1}\|^2}\right) \|\nabla_x \Psi_k\|^2 + \frac{\|\nabla_x \Psi_k\|^2}{\|\nabla \Psi_{k-1}\|^2} \nabla_x \Psi_k^T \tilde{d}_{k-1} = -\zeta_k \|\nabla_x \Psi_k\|^2,$$

and hence, (3.5) holds. □

To establish that the search direction (2.7) with (2.9) and (3.1) is a descent search direction, we give the following lemma.

Lemma 3.3. *Consider any point $v \in \Omega$ such that $t \in (0, 1]$ and $\nabla_x \tilde{F}(v)$ is nonsingular. If $\nabla_x \Psi(v) = 0$, then we have*

$$\bar{t}\gamma(v) - t < 0. \tag{3.6}$$

Proof. It follows from (3.3) that $\bar{t}\gamma(v) - t \leq 0$, and so we consider the case $\bar{t}\gamma(v) - t = 0$. Since $\nabla_x \Psi(v) = \nabla_x \tilde{F}(v)\tilde{F}(v) = 0$ and $\nabla_x \tilde{F}(v)$ is nonsingular, the relation $\tilde{F}(v) = 0$ holds. Therefore, we have from (1.2), (2.8), $t = \bar{t}\gamma(v)$ and $\bar{\gamma}\bar{t} < 1$ that

$$t = \bar{\gamma}\bar{t}\min\{1, \Psi(v)\} < \Psi(v) = \frac{1}{2}t^2.$$

This contradicts $0 < t \leq 1$, and hence, (3.6) holds. □

We now prove that the search direction (2.7) with (2.9) and (3.1) is a descent search direction.

Proposition 3.4. *Assume that $v_k \in \Omega$ and $0 < t_k \leq 1$. If $\nabla_x \tilde{F}_k$ is nonsingular, then the following holds*:

$$\nabla \Psi_k^T d_k \leq -(1-\eta)\|\nabla_x \Psi_k\|^2 + t_k(\bar{t}\gamma_k - t_k) < 0. \tag{3.7}$$

Proof. It follows from (2.5) and (2.7) that

$$\nabla \Psi_k^T d_k = \nabla_x \Psi_k^T \tilde{d}_k + t_k(\bar{t}\gamma_k - t_k) + \nabla_t \tilde{F}_k \tilde{F}_k(\bar{t}\gamma_k - t_k). \tag{3.8}$$

Now we consider two cases corresponding to (2.9) and (3.1).

The case $\nabla_x \Psi_k = 0$:
Since $\nabla_x \tilde{F}_k$ is nonsingular and $\nabla_x \Psi_k = \nabla_x \tilde{F}_k \tilde{F}_k = 0$, we have $\tilde{F}_k = 0$. Therefore, from (3.8), $\nabla_x \Psi_k = 0$ and $\tilde{F}_k = 0$, we have

$$\nabla \Psi_k^T d_k = t_k(\bar{t}\gamma_k - t_k) = -(1-\eta)\|\nabla_x \Psi_k\|^2 + t_k(\bar{t}\gamma_k - t_k).$$

It follows from Lemma 3.3 that $\bar{t}\gamma_k - t_k < 0$ holds. Thus, (3.7) is obtained.

The case $\nabla_x \Psi_k \neq 0$:
Note that the case $k = 0$ can be proven in the same way as the case $k \geq 1$. Hence, we consider only the case $k \geq 1$ in the following. If $\eta\|\nabla_x \Psi_k\|^2 \geq \nabla_t \tilde{F}_k \tilde{F}_k(\bar{t}\gamma_k - t_k)$, then it follows from (2.11), (3.4), (3.5), (3.8) and $\nabla_x \Psi_k \neq 0$ that

$$\begin{aligned} \nabla \Psi_k^T d_k &= -\|\nabla_x \Psi_k\|^2 + t_k(\bar{t}\gamma_k - t_k) + \nabla_t \tilde{F}_k \tilde{F}_k(\bar{t}\gamma_k - t_k) \\ &\leq -(1-\eta)\|\nabla_x \Psi_k\|^2 + t_k(\bar{t}\gamma_k - t_k) < 0. \end{aligned}$$

On the other hand, if $\eta\|\nabla_x \Psi_k\|^2 < \nabla_t \tilde{F}_k \tilde{F}_k(\bar{t}\gamma_k - t_k)$, then from (2.11), (3.4), (3.5), (3.8), and $\nabla_x \Psi_k \neq 0$, we have

$$\begin{aligned} \nabla \Psi_k^T d_k &= -\left(1 + \frac{\nabla_t \tilde{F}_k \tilde{F}_k(\bar{t}\gamma_k - t_k)}{\|\nabla_x \Psi_k\|^2}\right)\|\nabla_x \Psi_k\|^2 + t_k(\bar{t}\gamma_k - t_k) + \nabla_t \tilde{F}_k \tilde{F}_k(\bar{t}\gamma_k - t_k) \\ &\leq -\|\nabla_x \Psi_k\|^2 + t_k(\bar{t}\gamma_k - t_k) < -(1-\eta)\|\nabla_x \Psi_k\|^2 + t_k(\bar{t}\gamma_k - t_k) < 0. \end{aligned}$$

Therefore, the proof is complete. □

Now we are ready to give an algorithm of the scaling and smoothing FR (SSFR) method satisfying $0 < t_{k+1} \leq t_k$ and $\Psi_{k+1} < \Psi_k$.

Algorithm SSFR.

Step 0. Choose $\bar{t} \in (0, 1]$, $\bar{\gamma} \in (0, 1)$, $\sigma \in (0, 1)$., and $\delta \in (0, 1)$. Set $t_0 = \bar{t}$, and give an initial point $v_0 = (t_0, x_0) \in \Omega$. Let $k := 0$.

Step 1. If $\|F(x_k)\| = 0$, then stop.

Step 2. Compute d_k by (2.7) with (2.9) and (3.1).

Step 3. Find the smallest nonnegative integer ℓ satisfying

$$\Psi(v_k + \sigma^\ell d_k) \leq \Psi(v_k) + \delta\sigma^\ell \nabla\Psi_k^T d_k, \tag{3.9}$$

and set $\alpha_k = \sigma^\ell$.

Step 4. Update v_{k+1} by (2.6).

Step 5. Set $k := k + 1$, and return to Step 1.

In Step 0, an initial point v_0 must be chosen so that it belongs to Ω. This is easily done by taking $t_0 = \bar{t}$ because $(\bar{t}, x_0) \in \Omega$ for any $x_0 \in \boldsymbol{R}^n$. In Step 3, we can find the integer ℓ because Ψ is continuously differentiable and d_k is a descent search direction (namely, $\nabla\Psi_k^T d_k < 0$ for all k). Also in Step 3, we use the Armijo condition, which diverges from the STPRP and SSPRP methods.

Although we adopt the bisection method in Step 3, we can also use another procedure:

Step 3'. *Set $\ell = 0$ and $\alpha^{(\ell)} = 1$.*

Step 3.1. *If*

$$\Psi(v_k + \alpha^{(\ell)} d_k) \leq \Psi(v_k) + \delta\alpha^{(\ell)} \nabla\Psi_k^T d_k,$$

set $\alpha_k = \alpha^{(\ell)}$ and go to Step 4.

Step 3.2. *Choose $\sigma^{(\ell)} \in [\sigma_{\min}, \sigma_{\max}]$ and set $\alpha^{(\ell+1)} = \sigma^{(\ell)}\alpha^{(\ell)}$.*

Step 3.3. *Let $\ell := \ell + 1$, and return to Step 3.1.*

Here, $\sigma_{\min}$ and $\sigma_{\max}$ are positive constants such that $0 < \sigma_{\min} \leq \sigma_{\max} < 1$. If we set $\sigma_{\min} = \sigma_{\max} = \sigma$, then Step 3' reduces to the original Step 3. On the other hand, if we set

$$\sigma^{(\ell)} = \max\left\{\sigma_{\min},\ \min\left\{\sigma_{\max},\ \frac{0.5\alpha^{(\ell)}\nabla\Psi_k^T d_k}{\Psi_k + \alpha^{(\ell)}\nabla\Psi_k^T d_k - \Psi(v_k + \alpha^{(\ell)} d_k)}\right\}\right\},$$

then it becomes the quadratic interpolation line search. We will show the global convergence of Algorithm SSFR with Step 3 only for simplicity. However, we can show the global convergence of Algorithm SSFR with Step 3' in the same way.

4 Global Convergence

In this section, we prove the global convergence of Algorithm SSFR. Let $\{v_k\}$ be a sequence generated by Algorithm SSFR. We make the following assumption.

Assumption 4.1.

A1. For any positive constant t, the following holds:

$$\lim_{\|x\|\to\infty} \|\tilde{F}(t, x)\| = \infty.$$

A2. In some neighborhood $\mathcal{N}$ of the level set $\mathcal{L}$ at the initial point v_0 :

$$\mathcal{L} = \{v \in \boldsymbol{R}^{1+n} \mid \Psi(v) \leq \Psi(v_0),\ t \in [0,1]\},$$

$\nabla\Psi$ is Lipschitz continuous, i.e., there exists a positive constant L such that

$$\|\nabla\Psi(u) - \nabla\Psi(v)\| \leq L\|u - v\| \qquad \text{for all } u, v \in \mathcal{N}.$$

A3. For any $v \in (\boldsymbol{R}_{++} \times \boldsymbol{R}^n) \cap \Omega$, it holds that $\nabla_x \tilde{F}(v)$ is nonsingular.

Assumption A1 ensures the compactness of the level set $\mathcal{L}$. If F is level-bounded, namely, the level set of $\|F(x)\|$ is bounded at any point, Assumption A1 seems reasonable. Moreover, if we employ a regularization technique, for example, using $G(t,x) := \tilde{F}(t,x) + tx$ instead of $\tilde{F}(t,x)$ as a smoothing function, Assumption A1 could be satisfied under weaker conditions than the level boundedness of F.

Remark 4.1. From the decreasing property of $\{\Psi_k\}$, we have $\{v_k\} \subset \mathcal{L}$. Thus, it follows from Assumption A1 (namely, the boundedness of $\mathcal{L}$) that $\{v_k\}$ is bounded.

Under Assumption 4.1, we obtain the following proposition, which is originally given in [23]. The proof is identical, and so we omit it here.

Proposition 4.2 ([23, Proposition 2.3])**.** *Suppose Assumption* 4.1 *holds. Then, we have* $\{v_k\} \subset \Omega$.

Remark 4.3. Proposition 4.2 guarantees $\{v_k\} \subset \Omega$, which implies from Proposition 3.1 that $0 < t_{k+1} \leq t_k$ holds for all $k \geq 0$. Thus, there exists the limit value of the sequence $\{t_k\}$.

Remark 4.4. From Propositions 3.1 and 4.2, we have $\{v_k\} \subset (\boldsymbol{R}_{++} \times \boldsymbol{R}^n) \cap \Omega$. Thus, from Assumption A3, it holds that $\nabla_x \tilde{F}_k$ is nonsingular for any k.

The following lemma is also given by [23].

Lemma 4.5 ([23, Lemma 3.1])**.** *Suppose Assumption* 4.1 *holds. If* $\lim_{k\to\infty} t_k \neq 0$*, then there exists a positive constant* c_1 *such that*

$$\|\nabla\Psi_k\| \geq c_1$$

holds for all k.

We give the following lemma for showing the global convergence theorem.

Lemma 4.6. *Suppose Assumption* 4.1 *holds. If* $\lim_{k\to\infty} t_k \neq 0$*, then there exists a positive constant* c_2 *such that*

$$|\nabla\Psi_k^T d_k| \geq c_2$$

holds for all k.

Proof. By Remark 4.3 and the assumption of the lemma, there exists the positive limit $\lim_{k\to\infty} t_k \equiv \widehat{t} > 0$. Let us define a set $\widehat{\Omega}$ by

$$\widehat{\Omega} = ([\widehat{t}, 1] \times \boldsymbol{R}^n) \cap \Omega \cap \mathcal{L}. \tag{4.1}$$

Because $\mathcal{L}$ is compact by Assumption A1, Ω is closed, and $[\widehat{t}, 1] \times \boldsymbol{R}^n$ is closed, we find that $\widehat{\Omega}$ is compact. To show this lemma by contradiction, we assume that $\liminf_{k\to\infty} |\nabla\Psi_k^T d_k| = 0$.

Then, it follows from (3.4), (3.7) and $v_k \in \widehat{\Omega}$ that there exists a subsequence $K \subset \boldsymbol{N}$ and a limit point $\widehat{v}$ such that

$$\lim_{k\in K, k\to\infty} v_k = \widehat{v} \in \widehat{\Omega}, \quad \nabla_x \Psi(\widehat{v}) = 0, \quad \widehat{t}(\bar{t}\gamma(\widehat{v}) - \widehat{t}) = 0.$$

Since $\widehat{t} > 0$ and $\nabla_x \tilde{F}(\widehat{v})$ is nonsingular, this contradicts Lemma 3.3. Therefore, the proof is complete. □

We now show the global convergence theorem.

Theorem 4.7. *Suppose Assumption* 4.1 *holds. Then, the sequence* $\{v_k\}$ *has at least one accumulation point, and* $\lim_{k\to\infty} t_k = 0$ *holds. Moreover, any accumulation point* $v^* = (0, x^*)$ *satisfies* $H(v^*) = 0$*, and so* x^* *is a solution of* (1.1).

Proof. By Remark 4.1, $\{v_k\}$ has at least one accumulation point. Also from Remark 4.3, there exists the limit $\lim_{k\to\infty} t_k \equiv \widehat{t} \geq 0$. In order to prove $\widehat{t} = 0$ by contradiction, we assume that $\widehat{t} > 0$.

We have from (3.9) that

$$\Psi(v_k) - \Psi(v_{k+1}) \geq \delta\alpha_k |\nabla\Psi_k^T d_k|. \tag{4.2}$$

Since $\{\Psi_k\}$ is bounded below and nonincreasing, there exists a limit of $\{\Psi_k\}$. Therefore, Lemma 4.6 yields $\lim_{k\to\infty} \alpha_k = 0$. Then, the line search rule (3.9) with $\ell = 0$ is not satisfied for all sufficiently large k. Hence, we have

$$\delta\sigma^{-1}\alpha_k \nabla\Psi_k^T d_k < \Psi(v_k + \sigma^{-1}\alpha_k d_k) - \Psi_k. \tag{4.3}$$

On the other hand, by the mean-value theorem and Assumption A2, there exists a positive constant $\tau \in (0, 1)$ such that the following relations hold:

$$\begin{aligned}
\Psi(v_k + \sigma^{-1}\alpha_k d_k) - \Psi_k &= \sigma^{-1}\alpha_k \nabla\Psi(v_k + \tau\sigma^{-1}\alpha_k d_k)^T d_k \\
&= \sigma^{-1}\alpha_k \nabla\Psi_k^T d_k + \sigma^{-1}\alpha_k (\nabla\Psi(v_k + \tau\sigma^{-1}\alpha_k d_k) - \nabla\Psi_k)^T d_k \\
&\leq \sigma^{-1}\alpha_k \nabla\Psi_k^T d_k + L\sigma^{-2}\alpha_k^2 \|d_k\|^2.
\end{aligned}$$

Accordingly, (4.3) yields

$$\alpha_k \geq \frac{(1-\delta)\sigma|\nabla\Psi_k^T d_k|}{L\|d_k\|^2}. \tag{4.4}$$

It follows from (4.2) that

$$\sum_{k=0}^{\infty} \alpha_k |\nabla\Psi_k^T d_k| < +\infty,$$

which implies from (4.4) that

$$\sum_{k=0}^{\infty} \frac{(\nabla\Psi_k^T d_k)^2}{\|d_k\|^2} < +\infty. \tag{4.5}$$

It follows from Assumption A3, the compactness of $\widehat{\Omega}$ in (4.1) and the continuity of $\nabla_x \tilde{F}$ on $\widehat{\Omega}$ that there exists a positive constant c_3 such that

$$\|\nabla_x \tilde{F}_k^{-1}\| \leq c_3$$

holds for any k, where $\|\cdot\|$ denotes the matrix norm induced by the ℓ_2 vector norm. Accordingly, from (2.11), $\bar{t}\bar{\gamma}_k = \bar{t}\bar{\gamma}\min\{1, \Psi_k\} < 1$ and $t_k \le 1$, we have

$$\begin{aligned}
\zeta_k &\le 1 + \frac{|\nabla_t \tilde{F}_k \tilde{F}_k(\bar{t}\bar{\gamma}_k - t_k)|}{\|\nabla_x \tilde{F}_k \tilde{F}_k\|} \\
&= 1 + \frac{\|\nabla_x \tilde{F}_k^{-1}\|\ |\nabla_t \tilde{F}_k \tilde{F}_k(\bar{t}\bar{\gamma}_k - t_k)|}{\|\nabla_x \tilde{F}_k^{-1}\|\|\nabla_x \tilde{F}_k \tilde{F}_k\|} \\
&\le 1 + \frac{\|\nabla_x \tilde{F}_k^{-1}\|\|\nabla_t \tilde{F}_k\|\|\tilde{F}_k\||\bar{t}\bar{\gamma}_k - t_k|}{\|\nabla_x \tilde{F}_k^{-1}\nabla_x \tilde{F}_k \tilde{F}_k\|} \\
&\le 1 + c_3(1+\bar{t})\|\nabla_t \tilde{F}_k\|.
\end{aligned}$$

Since $\widehat{\Omega}$ is compact and $\nabla_t \tilde{F}$ is continuous on $\widehat{\Omega}$, there exists a positive constant $\bar{\zeta}$ such that $\zeta_k \le \bar{\zeta}$ holds for all k. If $\nabla_x \Psi_k = 0$, then (2.9) implies $\|\tilde{d}_k\| = 0$. Otherwise, we have from (3.1) and (3.5) that

$$\begin{aligned}
\|\tilde{d}_k\|^2 &= \beta_k^2\|\tilde{d}_{k-1}\|^2 - 2(\zeta_k + \theta_k)\nabla_x \Psi_k^T \tilde{d}_k - (\zeta_k + \theta_k)^2\|\nabla_x \Psi_k\|^2 \\
&= \beta_k^2\|\tilde{d}_{k-1}\|^2 + 2\zeta_k(\zeta_k + \theta_k)\|\nabla_x \Psi_k\|^2 - (\zeta_k + \theta_k)^2\|\nabla_x \Psi_k\|^2 \\
&= \beta_k^2\|\tilde{d}_{k-1}\|^2 + (\zeta_k^2 - \theta_k^2)\|\nabla_x \Psi_k\|^2 \\
&\le \beta_k^2\|\tilde{d}_{k-1}\|^2 + \bar{\zeta}^2\|\nabla_x \Psi_k\|^2.
\end{aligned}$$

Since $\widehat{\Omega}$ is compact and $\nabla_x \Psi$ is continuous on $\widehat{\Omega}$, there exists a positive constant c_4 such that $\|\nabla_x \Psi_k\| \le c_4$. Hence,

$$\|\tilde{d}_k\|^2 \le \beta_k^2\|\tilde{d}_{k-1}\|^2 + \bar{\zeta}^2 c_4^2.$$

Note that the above relation holds for the case $\nabla_x \Psi_k = 0$ (namely, $\|\tilde{d}_k\| = 0$). Therefore, we obtain

$$\|\tilde{d}_k\|^2 \le \left(\prod_{i=1}^{k} \beta_i^2\right)\|\tilde{d}_0\|^2 + \bar{\zeta}^2 c_4^2 \left(1 + \sum_{j=2}^{k}\prod_{i=j}^{k} \beta_i^2\right) \tag{4.6}$$

for all k. Since from (2.5) we have $\|\nabla_x \Psi_k\| \le \|\nabla \Psi_k\|$ for all k, (3.2) and Lemma 4.5 yield

$$\prod_{i=j}^{k} \beta_i = \frac{\|\nabla_x \Psi_j\|^2}{\|\nabla \Psi_{j-1}\|^2} \cdot \frac{\|\nabla_x \Psi_{j+1}\|^2}{\|\nabla \Psi_j\|^2} \cdots \frac{\|\nabla_x \Psi_{k-1}\|^2}{\|\nabla \Psi_{k-2}\|^2} \cdot \frac{\|\nabla_x \Psi_k\|^2}{\|\nabla \Psi_{k-1}\|^2} \le \frac{\|\nabla_x \Psi_k\|^2}{\|\nabla \Psi_{j-1}\|^2} \le \frac{c_4^2}{c_1^2},$$

for any k and j such that $1 \le j \le k$. Therefore, (4.6) implies that

$$\|\tilde{d}_k\|^2 \le \frac{c_4^4}{c_1^4}\|\tilde{d}_0\|^2 + \bar{\zeta}^2 c_4^2 \left(1 + \sum_{j=2}^{k} \frac{c_4^4}{c_1^4}\right) \le \frac{c_4^4}{c_1^4}\|\tilde{d}_0\|^2 + \bar{\zeta}^2 c_4^2 \max\left\{1, \frac{c_4^4}{c_1^4}\right\} k \equiv c_5 + c_6 k,$$

where $c_5 = \frac{c_4^4}{c_1^4}\|\tilde{d}_0\|^2$ and $c_6 = \bar{\zeta}^2 c_4^2 \max\{1, c_4^4/c_1^4\}$. Then, it follows from $\bar{t}\bar{\gamma} < 1$, $0 \le t_k \le 1$, $v_k \in \Omega$, (2.7) and (2.8) that

$$\|d_k\|^2 = \|\tilde{d}_k\|^2 + (\bar{t}\bar{\gamma}_k - t_k)^2 \le \|\tilde{d}_k\|^2 + 1 \le c_5 + c_6 k + 1.$$

Thus, from Lemma 4.6, we obtain

$$\frac{\|d_k\|^2}{(\nabla \Psi_k^T d_k)^2} \le \frac{c_5 + c_6 k + 1}{c_2^2},$$

which contradicts (4.5), and so $\lim_{k\to\infty} t_k = \widehat{t} = 0$ holds. Therefore, from (2.8), (3.3), and $\{v_k\} \in \Omega$, we obtain $\lim_{k\to\infty} \Psi_k = 0$. Thus, it follows from (1.2) that any accumulation point v^* satisfies $H(v^*) = 0$. □

5 Numerical Results

In this section, we give some preliminary numerical results of the proposed method. The program was coded in MATLAB R2020a, and computations were carried out on HP Z4 G4 Workstation (Intel(R) Xenon(R) W-2133, @3.60GHz, 3.60GHz) with 32.0GB RAM running Windows 10 Pro for Workstations. To compare our method with other methods, we tested the following methods:

SSFR	:	Algorithm SSFR with the quadratic interpolation,
SSPRP	:	Smoothing and scaling PRP method [27] with the quadratic interpolation,
STPRP	:	Smoothing three-term PRP method [23] with the quadratic interpolation,
SNewton	:	Smoothing Newton method [31] with the bisection method.

As shown in [27], the quadratic interpolation is suitable for SSPRP and STPRP, and the bisection method is suitable for SNewton. Also, in our experiments, the quadratic interpolation is suitable for SSFR. The smoothing Newton method solves $H(v) = 0$, and its search direction d_k is obtained by solving the following Newton equation:

$$H(v_k) + \nabla H(v_k)^T d_k = \gamma(v_k)\bar{v}, \tag{5.1}$$

where $\bar{v} = (\bar{t}, 0, \ldots, 0)^T \in \boldsymbol{R}^{1+n}$. The sequence $\{v_k\}$ generated by SNewton is included in the set Ω. In our method, we set the parameters $\bar{t} = \min\{0.1, 1/\sqrt{n}\}$, $\bar{\gamma} = 0.9$, $\eta = 0.1$, $\sigma_{\min} = 0.1$, $\sigma_{\max} = 0.9$, and $\delta = 0.001$. The stopping criterion was

$$\|F(x_k)\| \leq 10^{-5}.$$

The algorithms also stop when the number of iterations exceeds 1000, or a numerical overflow occurs.

We solved problem (1.1) with $F(x) = [F_1(x), \ldots, F_n(x)]^T$ and

$$\begin{aligned}
P1 : F_i(x) &= \begin{cases} e^{(x_i^2 + x_{i+1}^2)^{1/2}} - 1, & i \text{ is odd}, \\ x_{i-1} - x_i, & i \text{ is even}, \end{cases} \\
P2 : F_i(x) &= \begin{cases} e^{(x_i^2 + x_{i+1}^2)^{1/2}} - 1, & i \text{ is odd}, \\ \min\{x_{i-1}, x_i\}, & i \text{ is even}, \end{cases} \\
P3 : F_i(x) &= \begin{cases} \max\{0, x_i + x_{i+1}^2 + 2\} - 2, & i \text{ is odd}, \\ (x_i^2 + x_{i+1}^2)^{1/2}, & i \text{ is even}, \end{cases} \\
P4 : F_i(x) &= \begin{cases} e^{(x_i^2 + x_{i+1}^2)^{1/2}} - 1, & i \text{ is odd}, \\ \max\{x_{i-1}, x_i\}, & i \text{ is even}, \end{cases} \\
P5 : F_i(x) &= \begin{cases} e^{|\max\{x_i, x_{i+1}\}|} - 1, & i \text{ is odd}, \\ \min\{x_{i-1}, x_i\}, & i \text{ is even}, \end{cases} \\
P6 : F_i(x) &= n - 1 + e^{|x_i|} - \sum_{j=1}^{n} \cos x_j.
\end{aligned}$$

The above functions have the unique solution $x^* = (0, \ldots, 0)^T$. For P1–P6, we used the

following smoothing functions:

$$\begin{aligned}
&f(\alpha) = \sqrt{\alpha} \quad (\alpha > 0) &&\longrightarrow \quad \tilde{f}(t,\alpha) = \sqrt{\alpha + t^2},\\
&f(\alpha) = |\alpha| &&\longrightarrow \quad \tilde{f}(t,\alpha) = \sqrt{\alpha^2 + t^2},\\
&f(\alpha,\beta) = \max\{\alpha,\beta\} &&\longrightarrow \quad \tilde{f}(t,\alpha,\beta) = \frac{1}{2}\left(\alpha+\beta+\sqrt{(\alpha-\beta)^2+t^2}\right),\\
&f(\alpha,\beta) = \min\{\alpha,\beta\} &&\longrightarrow \quad \tilde{f}(t,\alpha,\beta) = \frac{1}{2}\left(\alpha+\beta-\sqrt{(\alpha-\beta)^2+t^2}\right).
\end{aligned}$$

We set the dimension $n = 1000, 3000$ and 5000. Note that the Jacobian matrices $\nabla H(v)$ with P1–P5 have sparse structures. We made use of the sparsity when the Newton equation (5.1) was solved in SNewton. For P1–P5, we randomly chose (for each dimension) 100 initial points in the rectangle $[-5,5]^n$. For P6, we randomly chose 100 initial points in the rectangle $[-1,1]^n$. Thus, in total, we tested 1800 instances.

To compare our method with the other methods, we adopted the performance profiles of Dolan and Moré [9]. For n_s solvers and n_p problems, the performance profile $P : \boldsymbol{R} \to [0,1]$ is defined as follows:

> Let $\mathcal{P}$ and $\mathcal{S}$ be the set of problems and the set of solvers, respectively. For each problem $p \in \mathcal{P}$ and for each solver $s \in \mathcal{S}$, we define $t_{p,s}$ = computing time (similarly for the number of iterations) required to solve problem p by solver s. The performance ratio is given by $r_{p,s} = t_{p,s}/\min_{s\in\mathcal{S}} t_{p,s}$. Then, the performance profile is defined by $P(\tau) = \frac{1}{n_p}\mathrm{size}\{p \in \mathcal{P} | r_{p,s} \le \tau\}$, for all $\tau > 0$, where size$A$, for any set A, stands for the number of the elements in that set. Note that $P(\tau)$ is the probability for solver $s \in \mathcal{S}$ such that a performance ratio $r_{p,s}$ is within a factor $\tau > 0$ of the best possible ratio.

The performance profiles of our numerical results are shown in Figures 1–6. We see from Figures 1, 2, 4, and 6 that SSFR performed better than or at least comparably with the other tested methods. By contrast, Figures 3 and 5 show that SSFR is not superior to the other methods. Summarizing the above arguments, SSFR is very efficient, but does not always perform better than the other tested methods.

It is known that SNewton converges quickly to a solution in the neighborhood of the solution. However, in our numerical experiments, the initial points were chosen randomly, and hence might not belong to the neighborhood. In addition, it is not possible to clarify the local behavior of the tested methods only by using performance profiles. Accordingly, to compare the numerical behaviors of errors for SNewton with those for the other tested methods, we give additional figures, Figures 7 and 8, which show $\log(\|x_k - x^*\|)$ at each iteration for P1 and P2 ($n = 5000$), respectively. The errors for SSFR, SSPRP, and STPRP decrease quickly in both cases, but the convergence rates of the methods seem linear. On the other hand, although SNewton converges q-superlinearly in the neighborhood of the solutions, it needs more iterations to approach the neighborhood than the other tested methods (especially for P2). To incorporate the merit of SNewton into SSFR, we tested a hybrid method (called Hybrid). Hybrid uses SSFR while $\|F(x_k)\| \ge 1$ and switches to SNewton when $\|F(x_k)\| < 1$. Figures 7 and 8 show that Hybrid performed very well. In particular, in Figure 8, we see that the sequences generated by SSFR, SSPRP and STPRP are a little stagnant near the solution, whereas the sequence generated by Hybrid converges very quickly to the solution. In our numerical experiments, we adopted a simple switching rule. Developing efficient switching techniques will be part of our further study.

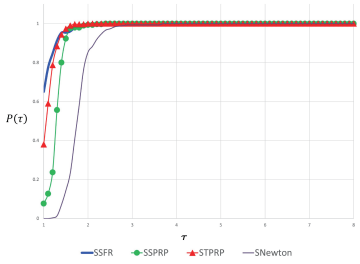

Figure 1: Performance profile for P1

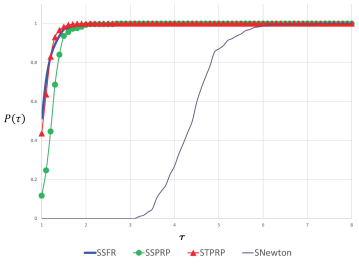

Figure 2: Performance profile for P2

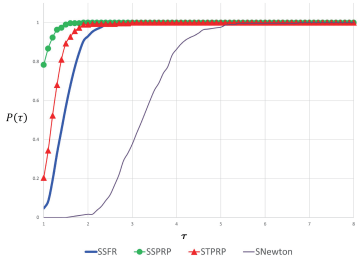

Figure 3: Performance profile for P3

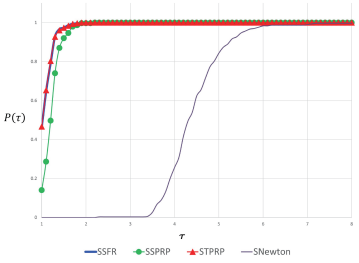

Figure 4: Performance profile for P4

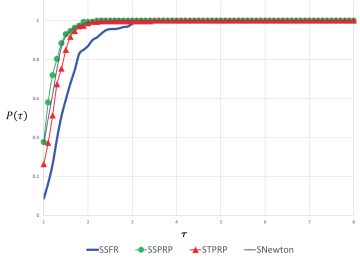

Figure 5: Performance profile for P5

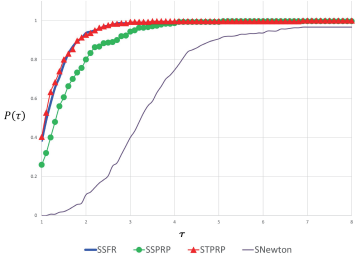

Figure 6: Performance profile for P6

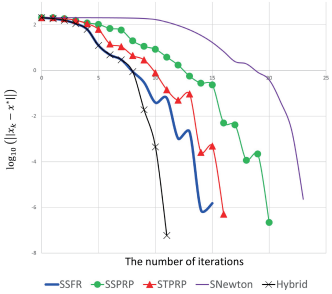

Figure 7: Numerical behavior for P1

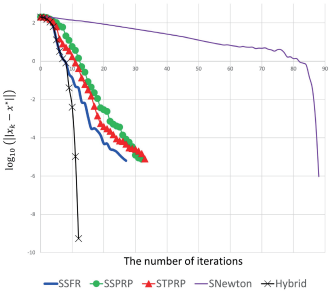

Figure 8: Numerical behavior for P2

6 Conclusion

In this paper, we proposed a smoothing and scaling Fletcher-Reeves type conjugate gradient method for solving a system of nonsmooth equations. The proposed method always generates descent search directions for the merit function. We showed the global convergence properties of the proposed method with the Armijo line search condition. In numerical experiments, we compared the proposed method with existing methods. The numerical results demonstrated that the proposed method is efficient and at least comparable with existing methods. We also tested a hybrid method, which combined the proposed method with the smoothing Newton method, and showed promising results. Thus, developing efficient switching techniques is an area of further study. Other future areas of study are to incorporate scaling techniques into the proposed method to accelerate the method, and/or to extend the proposed method to constrained systems of nonsmooth equations.

Acknowledgements

The authors would like to thank the associate editor of the journal and the anonymous referees for their valuable comments, which helped us to improve the quality of this paper. This research was supported in part by JSPS KAKENHI (grant number 18K11179 and 20K11698).

References

[1] M. Al-Baali, Y. Narushima and H. Yabe, A family of three-term conjugate gradient methods with sufficient descent property for unconstrained optimization, *Comput. Optim. Appl.* 60 (2015) 89–110.

[2] S. Aji, P. Kumam, A.M. Awwal, M.M. Yahaya and K. Sitthithakerngkiet, An efficient DY-type spectral conjugate gradient method for system of nonlinear monotone equations with application in signal recovery, *AIMS Math.* 6 (2021) 8078–8106.

[3] S. Aji, P. Kumam, P. Siricharoen, A.B. Abubakar and M.M. Yahaya, A modified conjugate descent projection method for monotone nonlinear equations and image restoration, *IEEE Access* 8 (2020) 158656–158665.

[4] A.M. Awwal, P. Kumam and A.B. Abubakar, Spectral modified Polak-Ribiére-Polyak projection conjugate gradient method for solving monotone systems of nonlinear equations, *Appl. Math. Comput.* 362 (2019) ID: 124514.

[5] W. Cheng, A two-term PRP-based descent method, *Numer. Funct. Anal. Optim.* 28 (2007) 1217–1230.

[6] W. Cheng, Y. Xiao and Q.-J. Hu, A family of derivative-free conjugate gradient methods for large-scale nonlinear systems of equations, *J. Comput. Appl. Math.* 224 (2009) 11–19.

[7] Z. Dai, X. Chen and F. Wen, A modified Perry's conjugate gradient method-based derivative-free method for solving large-scale nonlinear monotone equations, *Appl. Math. Comput.* 270 (2015) 378–386.

[8] Y.H. Dai and Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, *SIAM J. Optim.* 10 (1999) 177–182.

[9] E.D. Dolan and J.J. Moré, Benchmarking optimization software with performance profiles, *Math. Program.* 91 (2002) 201–213.

[10] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Springer, New York, 2003.

[11] X. Fang, A class of new derivative-free gradient type methods for large-scale nonlinear systems of monotone equations, *J. Inequal. Appl.* (2020) ID: 93(2020).

[12] R. Fletcher and C.M. Reeves, Function minimization by conjugate gradients, *Comput. J.* 7 (1964), 149–154.

[13] J.A. Ford, Y. Narushima and H. Yabe, Multi-step nonlinear conjugate gradient methods for unconstrained minimization, *Comput. Optim. Appl.* 40 (2008) 191–216.

[14] W.W. Hager and H. Zhang, A survey of nonlinear conjugate gradient methods, *Pac. J. Optim.* 2 (2006), 35–58.

[15] S. Hayashi, N. Yamashita and M. Fukushima, A combined smoothing and regularization method for monotone second-order cone complementarity problems, *SIAM J. Optim.* 15 (2005) 593–615.

[16] M.R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Res. Natl. Bur. Stand.* 49 (1952) 409–436.

[17] H. Kobayashi, Y. Narushima and H. Yabe, Descent three-term conjugate gradient methods based on secant conditions for unconstrained optimization, *Optim. Methods Softw.* 32 (2017) 1313–1329.

[18] W. La Cruz and M. Raydan, Nonmonotone spectral methods for large-scale nonlinear systems, *Optim. Methods Softw.* 18 (2003) 583–599.

[19] W. La Cruz, J.M. Martínez and M. Raydan, Spectral residual method without gradient information for solving large-scale nonlinear systems of equations, *Math. Comput.* 75 (2006) 1429–1448.

[20] M. Li, A Polak-Ribiére-Polyak method for solving large-scale nonlinear systems of equations and its global convergence, *Appl. Math. Comput.* 248 (2014) 314–332.

[21] Q. Li and D.H. Li, A class of derivative-free methods for large-scale nonlinear monotone equations, *IMA J. Numer. Anal.* 31 (2011) 1625–1635.

[22] W. Nakamura, Y. Narushima and H. Yabe, Nonlinear conjugate gradient methods with sufficient descent properties for unconstrained optimization, *J. Ind. Manag. Optim.* 9 (2013) 595–619.

[23] Y. Narushima, A smoothing conjugate gradient method for solving systems of nonsmooth equations, *Appl. Math. Comput.* 219 (2013) 8646–8655.

[24] Y. Narushima and H. Yabe, A survey of sufficient descent conjugate gradient methods for unconstrained optimization, *SUT J. Math.* 50 (2014) 167–203.

[25] Y. Narushima, H. Yabe and J.A. Ford, A three-term conjugate gradient method with sufficient descent property for unconstrained optimization, *SIAM J. Optim.* 21 (2011) 212–230.

[26] Y. Narushima, H. Ogasawara and S. Hayashi, A smoothing method with appropriate parameter control based on Fischer-Burmeister function for second-order cone complementarity problems, *Abstr. Appl. Anal.* 2013 (2013) ID: 830698.

[27] Y. Narushima, R. Ootani and H. Yabe, Global convergence of smoothing and scaling PRP type conjugate gradient methods for systems of nonsmooth equations, *Trans. Oper. Res. Soc. Jpn.* 59 (2016) 160–183. (in Japanese)

[28] Y. Narushima, N. Sagara and H. Ogasawara, A smoothing Newton method with Fischer-Burmeister function for second-order cone complementarity problems, *J. Optim. Theory Appl.* 149 (2011) 79–101.

[29] J. Nocedal and S.J. Wright, *Numerical Optimization (Second Edition), Springer Series in Operations Research*, Springer, New York, 2006.

[30] L. Qi and J. Sun, A nonsmooth version of Newton's method, *Math. Program.* 58 (1993) 353–367.

[31] L. Qi, D. Sun and G. Zhou, A new look at smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequalities, *Math. Program.* 87 (2000), 1–35.

[32] L. Qi and D. Sun, A survey of some nonsmooth equations and smoothing Newton methods, in: *Progress in Optimization*, A. Eberhard, R. Hill, D. Ralph and B.M. Glover (eds.), Springer, Boston, 1999, pp. 121–146.

[33] M.V. Solodov and B.F. Svaiter, A globally convergent inexact Newton method for systems of monotone equations, in: *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, M. Fukushima and L. Qi (eds.), Springer, Boston, 1998, pp. 355–369.

[34] M.Y. Waziri, K. Ahmed and J. Sabi'u, A family of Hager-Zhang conjugate gradient methods for system of monotone nonlinear equations, *Appl. Math. Comput.* 361 (2019) 645–660.

[35] G. Yuan, T. Li and W. Hu, A conjugate gradient algorithm for large-scale nonlinear equations and image restoration problems, *Appl. Numer. Math.* 147 (2020) 129–141.

[36] G. Yuan and M. Zhang, A three-terms Polak-Ribiére-Polyak conjugate gradient algorithm for large-scale nonlinear equations, *J. Comput. Appl. Math.* 286 (2015) 186–195.

[37] L. Zhang and W. Zhou, Spectral gradient projection method for solving nonlinear monotone equations, *J. Comput. Appl. Math.* 196 (2006) 478–484.

[38] L. Zhang, W. Zhou and D.H. Li, A descent modified Polak-Ribière-Polyak conjugate gradient method and its global convergence, *IMA J. Numer. Anal.* 26 (2006) 629–640.

[39] L. Zhang, W. Zhou and D.H. Li, Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search, *Numer. Math.* 104 (2006) 561–572.

[40] L. Zhang, W. Zhou and D.H. Li, Some descent three-term conjugate gradient methods and their global convergence, *Optim. Methods Softw.* 22 (2007) 697–711.

Manuscript received 10 November 2015
revised 17 August 2021
accepted for publication 23 August 2021

Yasushi Narushima
Department of Industrial and Systems Engineering, Keio University
3-14-1 Hiyoshi, Kouhoku-ku, Yokohama 223-8522, Japan
E-mail address: narushima@ae.keio.ac.jp

Hiroshi Yabe
Center for Data Science, Tokyo University of Science
1-3 Kagurazaka, Shinjuku-ku, Tokyo 162-8601, Japan
E-mail address: yabe@rs.tus.ac.jp