J. Math. Computer Sci., 24 (2022), 147–164

Online: ISSN 2008-949X

Journal of Mathematics and Computer Science

Journal Homepage: www.isr-publications.com/jmcs

Derivative-free SMR conjugate gradient method for constraint nonlinear equations

Abdulkarim Hassan Ibrahim[a], Kanikar Muangchoo[b,*], Nur Syarafina Mohamed[c], Auwal Bala Abubakar[d,e]

[a]Department of Mathematics, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), 126 Pracha-Uthit Road, Bang Mod, Thrung Khru, Bangkok 10140, Thailand.

[b]Department of Mathematics and Statistics, Faculty of Science and Technology, Rajamangala University of Technology Phra Nakhon (RMUTP), 1381 Pracharat 1 Road, Wongsawang, Bang Sue, Bangkok 10800, Thailand.

[c]Universiti Kuala Lumpur, Malaysian Institute of Industrial Technology, Malaysia.

[d]Department of Mathematical Sciences, Faculty of Physical Sciences, Bayero University, Kano, Kano, Nigeria.

[e]Department of Mathematics and Applied Mathematics, Sefako Makgatho Health Sciences University, Ga-Rankuwa, Pretoria, Medunsa-0204, South Africa.

Abstract

Based on the SMR conjugate gradient method for unconstrained optimization proposed by Mohamed et al. [N. S. Mohamed, M. Mamat, M. Rivaie, S. M. Shaharuddin, Indones. J. Electr. Eng. Comput. Sci., **11** (2018), 1188–1193] and the Solodov and Svaiter projection technique, we propose a derivative-free SMR method for solving nonlinear equations with convex constraints. The proposed method can be viewed as an extension of the SMR method for solving unconstrained optimization. The proposed method can be used to solve large-scale nonlinear equations with convex constraints because of derivative-free and low storage. Under the assumption that the underlying mapping is Lipschitz continuous and satisfies a weaker monotonicity assumption, we prove its global convergence. Preliminary numerical results show that the proposed method is promising.

Keywords: Nonlinear equations, conjugate gradient method, projection method, global convergence.

2020 MSC: 65L09, 65K05, 90C30.

1. Introduction

Mathematically, nonlinear systems of equations with convex constraint, can be express as

$$j(c) = 0, \ c \in \Theta, \tag{1.1}$$

where $j : \Theta \to \mathbb{R}$ is a continuous nonlinear mapping, and $\Theta \subseteq \mathbb{R}^n$ is a closed convex set. Nonlinear equations of the form (1.1) commonly appears in various applications such as financial forecasting problems [9], nonlinear compressed sensing [6], non-negative matrix factorisation [4, 25], economic equilibrium

*Corresponding author
Email address: kanikar.m@rmutp.ac.th (Kanikar Muangchoo)

doi: 10.22436/jmcs.024.02.06

Received: 2020-08-03 *Revised:* 2020-12-13 *Accepted:* 2021-01-02

problems [13] and many others. Consequently, a number of different iterative methods have been developed to solve (1.1). For instance, see [10, 11, 28, 32]. La Cruz [7] recently presented a spectral method which uses the residual vector as a search direction to solve large-scale systems of nonlinear equations involving monotone mapping. Solodov and Svaiter [30] suggested a method in which projection, proximal point and Newton method were combined. Motivated by the work of Solodov [30], Zhang and Zhou [34] proposed a spectral gradient projection method for solving nonlinear equation involving monotone mapping. Our interest in this paper, is on the conjugate gradient method. The conjugate gradient method is widely used for solving the unconstrained optimization problem. In the last years, various conjugate gradient methods for large-scale unconstrained optimization have been extended to solve (1.1). For instance, using the projection technique of Solodov [30], Ibrahim et al. [20] extended the hybrid LS-FR conjugate gradient method proposed by Djordjević [14] to solve nonlinear equation with convex constraint. At each iteration, the proposed method does not store any matrix. Also, inspired by the RMILL method [29] for unconstrained optimization, Fang [16] developed three derivative-free conjugate gradient procedures. For more articles on derivative-free algorithms, interested readers may refer to the recent papers [1–3, 18, 19, 21–24].

Quite recently, Mohammed et al. [27] designed a SMR conjugate gradient method for solving unconstrained optimization problem. Based on the efficiency performance of the SMR method and the projection technique of Solodov and Svaiter [30] we describe a class of new derivative-free conjugate gradient method for solving (1.1). The global convergence of the method is proved under the assumption that the underlying mapping is Lipschitz continuous and satisfies a weaker monotonicity assumption.

The remaining part of this paper is organized as follows. In the next section, we propose a derivative-free SMR method for solving the constraint nonlinear equation (1.1). Under mild assumption, the global convergence is established in Section 3. In Section 4, preliminary numerical results are presented to show that our method are efficient and promising. Finally, we have the conclusion. Throughout this manuscript, $\|\cdot\|$ denotes the Euclidean norm.

2. The method

In this section, we present our method for solving the nonlinear equation with convex constraints (1.1). Our method is based on the SMR conjugate gradient method which we will review below. The SMR conjugate gradient method proposed by Mohamed et al. [27] for solving the following unconstrained optimization problem

$$\min g(c),\ c \in \mathbb{R}^n,$$

where $g : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function. Let $\nabla g(c_t)$ denote the gradient of g at c_t. The Mohammed et al. [27] method generates a sequence of iterates $\{c_t\}$ by the following recursive formula

$$c_{t+1} = c_t + \alpha_t p_t,\ t \geqslant 0 \tag{2.1}$$

where c_t is the current iterative point and $c_0 \in \mathbb{R}^n$ is set to be a starting point of the sequence. From (2.1), $\alpha_t > 0$ is known as a step size and p_t is the search direction defined by the rule:

$$p_t := \begin{cases} -\nabla g(c_t), & \text{if } t = 0, \\ -\nabla g(c_t) + \beta_t^{SMR} p_{t-1}, & \text{if } t > 0, \end{cases}$$

where the conjugate gradient parameter β_t^{SMR} is defined as

$$\beta_t^{SMR} := \max\left\{0, \frac{\|\nabla g(c_t)\|^2 - |\nabla g(c_t)^T \nabla g(c_{t-1})|}{\|p_{t-1}\|^2}\right\}.$$

Based on the SMR method, we introduce a derivative-free projection method for solving (1.1). Our proposed method first generate a trial point d_t by the following relation:

$$d_t = c_t + \alpha_t p_t$$

and the search direction p_t is computed by

$$p_t := \begin{cases} -j_t, & \text{if } t = 0, \\ -j_t + \beta_t^{ESMR} p_{t-1}, & \text{if } t > 0, \end{cases} \tag{2.2}$$

where $j_t = j(c_t)$ and β_t^{ESMR} is defined as

$$\beta_t^{ESMR} := \max\left\{0, \frac{\|j_t\|^2 - |j_t^T j_{t-1}|}{\|p_{t-1}\|^2}\right\}. \tag{2.3}$$

It can be observed that the search direction p_t defined by (2.2) is likely not a descent direction for all t. To ensure the decency, we choose a vector from a subspace $\varphi_t = \{q \mid j_t^T q = 0\}$ to replace the second term $\beta_t^{ESMR} p_{t-1}$ of the direction (2.2). We obtain

$$p_t = -j_t + q, \;\; q \in \varphi_t.$$

For instance, if we choose $q = 0 \in \varphi_t$, the steepest descent direction is obtained. If we choose

$$q = \beta_t^{ESMR}\left(p_{t-1} - \frac{j_t^T p_{t-1}}{\|j_t\|^2} j_t\right) \in \varphi_t,$$

which is obviously motivated by the Gram-Schmidt (MGS) process, we get the direction used in [17, 31].

Definition 2.1. Let $\Theta \subseteq \mathbb{R}^n$ be a nonempty closed convex set. Then for any $y \in \mathbb{R}^n$, its projection onto Θ, denoted by $P_\Theta[y]$, is defined by

$$P_\Theta[y] := \arg\min\{\|y - x\| \;: x \in \Theta\}.$$

The projection operator P_Θ has a well-known property, that is, for any $y, x \in \mathbb{R}^n$ the following nonexpansive property hold

$$\|P_\Theta[y] - P_\Theta[x]\| \leqslant \|y - x\|. \tag{2.4}$$

In what follows, we state the iterative procedures/steps of our method.

Algorithm 1

Input. Set an initial point $c_0 \in \Theta$, the positive constants: $Tol > 0$, $r \in (0,1)$, $\varkappa \in (0,2)$, $a > 0, \mu > 0$. Set $t = 0$.

Step 0. Compute j_t. If $\|j_t\| \leqslant Tol$, stop. Otherwise, generate the search direction p_t using the following

$$p_t := \begin{cases} -j_t, & \text{if } t = 0, \\ -j_t + \beta_t^{ESMR}\left(p_{t-1} - \frac{j_t^T p_{t-1}}{\|j_t\|^2} j_t\right), & \text{if } t > 0, \end{cases} \tag{2.5}$$

where β_t^{ESMR} is computed by (2.3).

Step 1. Determine the step-size $\alpha_t = \max\{ar^m | m \geqslant 0\}$ such that

$$j(c_t + \alpha_t p_t)^T p_t \geqslant \mu \alpha_t \|p_t\|^2. \tag{2.6}$$

Step 2. Compute $d_t = c_t + \alpha_t p_t$, where d_t is a trial point.

Step 3. If $d_t \in \Theta$ and $\|j(d_t)\| = 0$, stop. Otherwise, compute the next iterate by

$$c_{t+1} = P_\Theta\left[c_t - \varkappa \frac{j(d_t)^T (c_t - d_t)}{\|j(d_t)\|^2} j(d_t)\right],$$

Step 4. Finally we set $t = t + 1$ and return to step 1.

---

3. Theoretical analysis

In this section, we obtain the global convergence property of Algorithm 1. We also make the following assumptions on the mapping j.

Assumption 3.1.

(i) The solution set of the constrained nonlinear (1.1), denoted by Θ^*, is nonempty.

(ii) The mapping j is Lipschitz continuous on $\mathbb{R}^n$. That is, there exists a constant $L > 0$ such that

$$\|j(\alpha) - j(\beta)\| \leqslant L\|\alpha - \beta\|, \ \forall \alpha, \beta \in \mathbb{R}^n. \tag{3.1}$$

(iii) For any $\beta \in \Theta^*$ and $\alpha \in \mathbb{R}^n$, it holds that

$$j(\alpha)^T(\alpha - \beta) \geqslant 0.$$

Lemma 3.2. *Let p_t be the search direction generated by Algorithm 1, then p_t is a sufficient descent direction. That is for all $t \geqslant 0$,*

$$j_t^T p_t = -c\|j_t\|^2, \ c > 0. \tag{3.2}$$

Proof. The proof follows. □

Lemma 3.3. *Let $\{p_t\}$ and $\{c_t\}$ be two sequences generated by Algorithm 1. Then, there exists a step size α_t satisfying the line search (2.6) for all $t \geqslant 0$.*

Proof. For any $m \geqslant 0$, suppose (2.6) does not hold for the iterate t_0-th, then we have

$$-j(c_{t_0} + ar^m p_{t_0})^T p_{t_0} < \mu a r^m \|p_{t_0}\|^2.$$

Thus, by the continuity of j and with $0 < r < 1$, it follows that by letting $m \to \infty$, we have

$$-j(c_{t_0})^T p_{t_0} \leqslant 0,$$

which contradicts (3.2).

□

Lemma 3.4. *Let the sequences $\{c_t\}$ and $\{d_t\}$ be generated by the Algorithm 1 method under Assumption 3.1, then*

$$\alpha_t \geqslant \max\left\{a, \frac{rc\|j_t\|^2}{(L+\mu)\|p_t\|^2}\right\}. \tag{3.3}$$

Proof. Let $\hat{\alpha}_t = \alpha_t r^{-1}$. Assume $\alpha_t \neq a$, from (2.6), $\hat{\alpha}_t$ does not satisfy (2.6). That is,

$$-j(c_t + \hat{\alpha}_t p_t)^T p_t < \mu\hat{\alpha}_t\|p_t\|^2.$$

From (3.1) and (3.2), it can be obviously seen that

$$\begin{aligned} c\|j_t\|^2 \leqslant -j_t^T p_t &= (j(c_t + \hat{\alpha}_t p_t) - j_t)^T p_t - j(c_t + \hat{\alpha}_t p_t)^T p_t \\ &\leqslant L\hat{\alpha}_t\|p_t\|^2 + \mu\hat{\alpha}_t\|p_t\|^2 \leqslant \hat{\alpha}_t(L+\mu)\|p_t\|^2. \end{aligned}$$

This gives the desired inequality (3.3). □

Lemma 3.5. *Suppose that Assumption 3.1 holds. Let $\{c_t\}$ and $\{d_t\}$ be sequences generated by the Algorithm 1, then for any solution c^* contained in the solution set Θ^* the inequality*

$$\|c_{t+1} - c^*\|^2 \leqslant \|c_t - c^*\|^2 - \mu^2\|c_t - d_t\|^4$$

holds. In addition, $\{c_t\}$ is bounded and

$$\sum_{t=0}^{\infty} \|c_t - d_t\|^4 < +\infty. \tag{3.4}$$

Proof. First, we begin by using the weakly monotonicity assumption (Assumption 3.1 (iii)) on the mapping j. Thus, for any solution $c^* \in \Theta^*$,

$$j(d_t)^T(c_t - c^*) \geqslant j(d_t)^T(c_t - d_t).$$

The above inequality together with (2.6) gives

$$j(c_t + \alpha_t p_t)^T(c_t - d_t) \geqslant \mu\alpha_t^2\|p_t\|^2 \geqslant 0. \tag{3.5}$$

From (2.4) and (3.5), we have the following

$$\begin{aligned}
\|c_{t+1} - c^*\|^2 &= \left\| P_\Theta\left[c_t - x\frac{j(d_t)^T(c_t - d_t)}{\|j(d_t)\|^2} j(d_t)\right] - c^*\right\|^2 \\
&\leqslant \left\| \left[c_t - x\frac{j(d_t)^T(c_t - d_t)}{\|j(d_t)\|^2} j(d_t)\right] - c^*\right\|^2 \\
&= \|c_t - c^*\|^2 - 2x\left(\frac{j(d_t)^T(c_t - d_t)}{\|j(d_t)\|^2}\right) j(d_t)^T(c_t - c^*) + x^2\left(\frac{j(d_t)^T(c_t - d_t)}{\|j(d_t)\|}\right)^2 \\
&= \|c_t - c^*\|^2 - 2x\left(\frac{j(d_t)^T(c_t - d_t)}{\|j(d_t)\|^2}\right) j(d_t)^T(c_t - d_t) + x^2\left(\frac{j(d_t)^T(c_t - d_t)}{\|j(d_t)\|}\right)^2 \\
&= \|c_t - c^*\|^2 - x(2 - x)\left(\frac{j(d_t)^T(c_t - d_t)}{\|j(d_t)\|}\right)^2 \\
&\leqslant \|c_t - c^*\|^2.
\end{aligned}$$

Thus, the sequence $\{\|c_t - c^*\|\}$ has a nonincreasing and convergent property. Therefore, this makes $\{c_t\}$ to be bounded and therefore the following holds

$$\sigma^2 \sum_{t=0}^{\infty} \|c_t - d_t\|^4 < \|c_0 - c^*\|^2 < +\infty.$$

□

Remark 3.6. Taking into account of the definition of d_t and also by (3.4), it can be deduced that

$$\lim_{t\to\infty} \alpha_t\|p_t\| = 0. \tag{3.6}$$

Theorem 3.7. *Suppose Assumption 3.1 holds. Let $\{c_t\}$ and $\{d_t\}$ be sequences generated by Algorithm 1, then*

$$\liminf_{t\to\infty} \|j_t\| = 0. \tag{3.7}$$

Proof. Suppose (3.7) is not valid, that is, there exist a constant say $s > 0$ such that $s \leqslant \|j_t\|$, $t \geqslant 0$. Then this along with (3.2) implies that

$$\|p_t\| \geqslant cs, \quad \forall t \geqslant 0.$$

It can be obviously seen from Lemma 3.5 and Remark 3.6, that the sequences $\{c_t\}$ and $\{d_k\}$ are bounded. In addition with the continuity of j, it further implies that $\{\|j_t\|\}$ is bounded by a constant say u. From (2.5), it follows that for all $t \geqslant 1$,

$$\begin{aligned}\|p_t\| &= \left\| -j_t + \beta_t^{ESMR}\left(p_{t-1} - \frac{j_t^T p_{t-1}}{\|j_t\|^2} j_t\right)\right\| \\ &= \left\| -j_t + \frac{\|j_t\|^2 - |j_t^T j_{t-1}|}{\|p_{t-1}\|^2}\left(p_{t-1} - \frac{j_t^T p_{t-1}}{\|j_t\|^2} j_t\right)\right\| \leqslant \|j_t\| + 2\frac{\|j_t\|^2 + \|j_t\|\|j_{t-1}\|}{\|p_{t-1}\|} \leqslant u + \frac{4u^2}{cs} \triangleq \gamma.\end{aligned}$$

It is worth mentioning since $\beta_t^{ESMR} = 0$, it is easy to see that p_{t-1} is still bounded. From (3.3), we have

$$\alpha_t\|p_t\| \geqslant \max\left\{a, \frac{rc\|j_t\|^2}{(L+\mu)\|p_t\|^2}\right\}\|p_t\| \geqslant \max\left\{acs, \frac{rcs^2}{(L+\mu)\gamma}\right\} > 0,$$

which contradicts (3.6). Hence (3.7) is valid. □

4. Numerical experiments

This section evaluates the numerical efficiency of the proposed algorithm using the Dolan and More performance profile [15]. The metrics taking into consideration using the Dolan and More performance profile includes; the number of iterations, the number of function evaluations and the CPU running time. The performance of Algorithm 1 is compared with the derivative-free iterative method for nonlinear monotone equations with convex constraints proposed in [26]. In what follows, we refer to the algorithm proposed in [26] as Algorithm 2. All codes were coded and implemented in Matlab environment.

- Control parameters: For Algorithm 1, we select $a = 1$, $r = 0.8$, $\mu = 10^{-4}$, $\varkappa = 1.2$, $Tol = 10^{-6}$. As for Algorithm 2, we select all parameters as in [26].
- Dimensions: $1000, 5000, 10,000, 50,000, 100,000$.
- Initial points: $c_1 = (0.1, 0.1, \dots, 0.1)^T$, $c_2 = (0.2, 0.2, \dots, 0.2)^T$, $c_3 = (0.5, 0.5, \dots, 0.5)^T$, $c_4 = (1.2, 1.2, \dots, 1.2)^T$, $c_5 = (1.5, 1.5, \dots, 1.5)^T$, $c_6 = (2, 2, \dots, 2)^T$, $c_7 = \text{rand}(0, 1)$.

The test problems with $j = (j_1, j_2, \cdots, j_n)$ are given below.

Problem 4.1 ([8])**.** Exponential function:

$$j_1(c) = e^{c_1} - 1, \quad j_i(c) = e^{c_i} + c_i - 1, \text{ for } i = 2, 3, \dots, n, \quad \text{and } \Theta = \mathbb{R}^n_+.$$

Problem 4.2 ([8])**.** Modified logarithmic function:

$$j_i(c) = \ln(c_i + 1) - \frac{c_i}{n}, \text{ for } i = 1, 2, 3, \dots, n,$$
$$\Theta = \left\{c \in \mathbb{R}^n : \sum_{i=1}^{n} c_i \leqslant n, c_i > -1, i = 1, 2, \cdots, n\right\}.$$

Problem 4.3 ([7])**.**

$$j_i(c) = \min\left(\min(|c_i|, c_i^2), \max(|c_i|, c_i^3)\right) \text{ for } i = 2, 3, \dots, n, \text{ and } \Theta = \mathbb{R}^n_+.$$

Problem 4.4 ([8]). Strictly convex function I:

$$j_i(c) = e^{c_i} - 1, \text{ for } i = 1, 2, \ldots, n, \quad \text{and} \quad \Theta = \mathbb{R}^n_+.$$

Problem 4.5 ([8]). Strictly convex function II:

$$j_i(c) = \frac{i}{n} e^{c_i} - 1, \text{ for } i = 1, 2, \ldots, n, \quad \text{and} \quad \Theta = \mathbb{R}^n_+.$$

Problem 4.6 ([5]). Tridiagonal exponential function:

$$\begin{aligned} j_1(c) &= c_1 - e^{\cos(h(c_1+c_2))}, \\ j_i(c) &= c_i - e^{\cos(h(c_{i-1}+c_i+c_{i+1}))}, \text{ for } i = 2, \ldots, n-1, \\ j_n(z) &= c_n - e^{\cos(h(c_{n-1}+c_n))}, \\ h &= \frac{1}{n+1}. \end{aligned}$$

Problem 4.7 ([33]). Nonsmooth function:

$$j_i(c) = c_i - \sin|c_i - 1|, \ i = 1, 2, 3, \ldots, n,$$
$$\Theta = \left\{ c \in \mathbb{R}^n : \sum_{i=1}^{n} c_i \leqslant n, c_i \geqslant -1, i = 1, 2, \cdots, n \right\}.$$

Problem 4.8 ([8]). The Trig exp function

$$j_1(c) = 3c_1^3 + 2c_2 - 5 + \sin(c_1 - c_2)\sin(c_1 + c_2),$$
$$j_i(c) = 3c_i^3 + 2c_{i+1} - 5 + \sin(c_i - c_{i+1})\sin(c_i + c_{i+1}) + 4c_i - c_{i-1}e^{c_{i-1}-c_i} - 3 \text{ for } i = 2, 3, \ldots, n-1,$$
$$j_n(z) = c_{n-1}e^{c_{n-1}-c_n} - 4c_n - 3, \text{ where } h = \frac{1}{m+1} \text{ and } \Theta = \mathbb{R}^n_+.$$

Problem 4.9 ([12]).

$$t_i = \sum_{i=1}^{n} c_i^2, \quad d = 10^{-5}, \quad j_i(c) = 2d(c_i - 1) + 4(t_i - 0.25)c_i, \ i = 1, 2, 3, \ldots, n, \ \text{and} \ \Theta = \mathbb{R}^n_+.$$

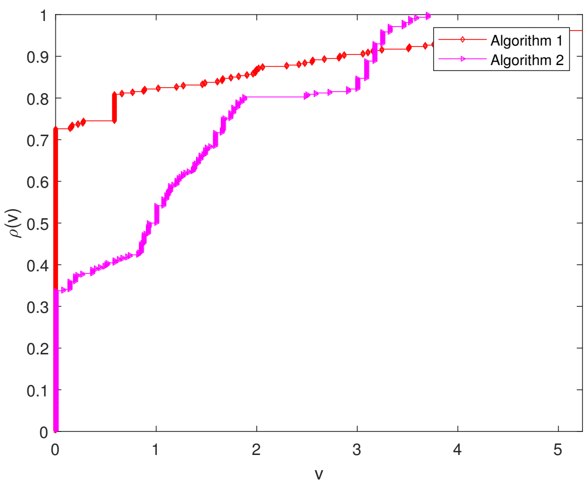

Figure 1: Performance profiles based on number of iterations.

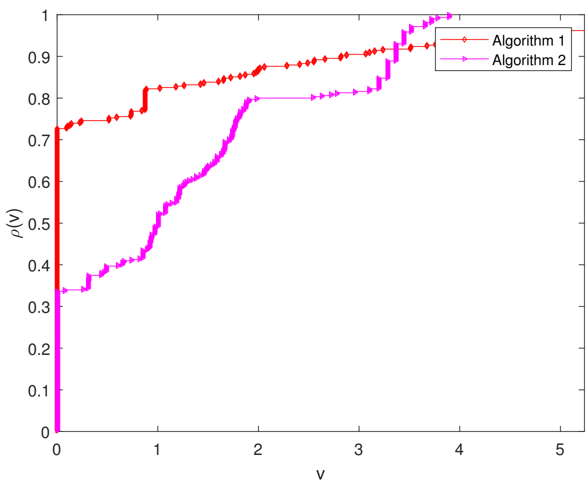

Figure 2: Performance profiles based on number of function evaluations.

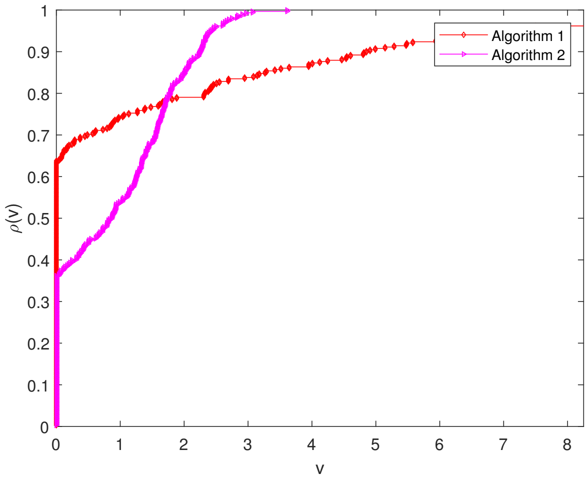

Figure 3: Performance profiles based on CPU time (in seconds).

The assessment results are given in Tables 1-9 of the appendix section. In Tables 1-9, ”dm” denotes the dimension, ”inp” denotes the initial point, ”it” denotes the number of iteration, ”nf” denotes the number of function evaluation, and ”tm” denotes the CPU running time. Figure 1 displays the performance profile of the number of iterations. It can be seen that Algorithm 1 is the top curve. Algorithm 1 outperforms Algorithm 2, with Algorithms 1 been able to solve 72% of the test problems with less number of iterations while, Algorithm 2 was able to solve around 32%. Similarly, in Figure 2, Algorithm 1 achieved less number of function evaluations compared to Algorithm 2. Figure 3 is the performance profile measured by the CPU time. The top curve is the Algorithm 1 that solved the most problems in a time that was within a factor v of the best time. In particular, the Algorithm 1 solves about 62% of the test problems with the least CPU time, while Algorithm 2 solves around 39%. Based on the above comparisons, it indicates that the Algorithm 1 outperforms Algorithm 2 well-known methods for all metric, that is the number of iteration, the total number of function evaluations and the CPU time.

5. Conclusion

We have proposed a derivative-free SMR conjugate gradient method for constraint nonlinear equation. The search direction of the proposed method satisfies the sufficient descent condition. The global convergence is proved under the assumption that the underlying operator is Lipschitz continuous and satisfies a weaker monotonicity condition. Numerical experiment shows that the proposed method is efficient.

Acknowledgment

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

References

[1] A. B. Abubakar, A. H. Ibrahim, A. B. Muhammad, C. Tammer, *A modified descent Dai-Yuan conjugate gradient method for constraint nonlinear monotone operator equations*, Appl. Anal. Optim., **4** (2020), 1–24. 1

[2] A. B. Abubakar, P. Kumam, A. H. Ibrahim, J. Rilwan, *Derivative-free hs-dy-type method for solving nonlinear equations and image restoration*, Heliyon, **6** (2020), 1–14.

[3] A. B. Abubakar, J. Rilwan, S. E. Yimer, A. H. Ibrahim, I. Ahmed, *Spectral three-term conjugate descent method for solving nonlinear monotone equations with convex constraints*, Thai J. Math., **18** (2020), 501–517. 1

[4] M. W. Berry, M. Browne, A. N. Langville, V. Pauca, R. J. Plemmons, , Comput. Statist. Data Anal., **52** (2007), 155–173. 1

[5] Y. Bing, G. Lin, *An efficient implementation of Merrill's method for sparse or partially separable systems of nonlinear equations*, SIAM J. Optim., **1** (1991), 206–221. 4.6

[6] T. Blumensath, *Compressed sensing with nonlinear observations and related nonlinear optimization problems*, IEEE Trans. Inform. Theory, **59** (2013), 3466–3474. 1

[7] W. L. Cruz, *A spectral algorithm for large-scale systems of nonlinear monotone equations*, Numer. Algorithms, **76** (2017), 1109–1130. 1, 4.3

[8] W. L. Cruz, J. M. Martínez, M. Raydan, *Spectral residual method without gradient information for solving large-scale nonlinear systems of equations*, Math. Comp., **75** (2006), 1429–1448. 4.1, 4.2, 4.4, 4.5, 4.8

[9] Z. Dai, X. Dong, J. Kang, L. Hong, *Forecasting stock market returns: New technical indicators and two-step economic constraint method*, North Am. J. Econ. Finance, **53** (2020), 1–12. 1

[10] J. E. Dennis, Jr., J. J. Moré, *A characterization of superlinear convergence and its application to quasi-newton methods*, Math. Comp., **28** (1974), 549–560. 1

[11] J. E. Dennis, Jr., R. B. Schnabel, *numerical methods for unconstrained optimization and nonlinear equations*, Prentice Hall, Englewood Cliffs, (1983). 1

[12] Y. Ding, Y. Xiao, J. Li, *A class of conjugate gradient methods for convex constrained monotone equations*, Optimization, **66** (2017), 2309–2328. 4.9

[13] S. P. Dirkse, M. C. Ferris, *Mcplib: A collection of nonlinear mixed complementarity problems*, Optim. Methods Softw., **5** (1995), 319–345. 1

[14] S. S. Djordjević, *New Hybrid Conjugate Gradient Method As A Convex Combination of Ls and Fr Methods*, Acta Math. Sci. Ser. B (Engl. Ed.), **39** (2019), 214–228. 1

[15] E. D. Dolan, J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program., **91** (2002), 201–213. 4

[16] X. Fang, *A class of new derivative-free gradient type methods for large-scale nonlinear systems of monotone equations*, J. Inequal. Appl., **2020** (2020), 1–13. 1

[17] D. Feng, M. Sun, X. Wang, *A family of conjugate gradient methods for large-scale nonlinear equations*, J. Inequal. Appl., **2017** (2017), 1–8. 2

[18] A. H. Ibrahim, A. I. Garba, H. Usman, J. Abubakar, A. B. Abubakar, *Derivative-free rmil conjugate gradient algorithm for convex constrained equations*, Thai J. Math., **18** (2020), 212–232. 1

[19] A. H. Ibrahim, P. Kumam, A. B. Abubakar, J. Abubakar, A. B. Muhammad, *Least-square-based three-term conjugate gradient projection method for ℓ_1-norm problems with application to compressed sensing*, Mathematics, **8** (2020), 1–21. 1

[20] A. H. Ibrahim, P. Kumam, A. B. Abubakar, W. Jirakitpuwapat, J. Abubakar, *A hybrid conjugate gradient algorithm for constrained monotone equations with application in compressive sensing*, Heliyon, **6** (2020), 1–17. 1

[21] A. H. Ibrahim, P. Kumam, A. B. Abubakar, U. B. Yusuf, J. Rilwan, *Derivative-free conjugate residual algorithms for convex constraints nonlinear monotone equations and signal recovery*, J. Nonlinear Convex Anal., **21** (2020), 1959–1972. 1

[22] A. H. Ibrahim, P. Kumam, A. B. Abubakar, U. B. Yusuf, S. E. Yimer, K. O. Aremu, *An efficient gradient-free projection algorithm for constrained nonlinear equations and image restoration*, Aims Math., **6** (2021), 235–260.

[23] A. H. Ibrahim, P. Kumam, W. Kumam, *A family of derivative-free conjugate gradient methods for constrained nonlinear equations and image restoration*, IEEE Access, **8** (2020), 162714–162729.

[24] A. H. Ibrahim, K. Muangchoo, A. B. Abubakar, A. D. Adedokun, H. Mohammed, *Spectral conjugate gradient like method for signal reconstruction*, Thai J. Math., **18** (2020), 2013–2022. 1

[25] D. D. Lee, H. S. Seung, *Algorithms for Non-negative Matrix Factorization*, Adv. Neural Inf. Process. Syst., (2001), 556–562. 1

[26] J. Liu, Y. Feng, *A derivative-free iterative method for nonlinear monotone equations with convex constraints*, Numer. Algorithms, **82** (2019), 245–262. 4

[27] N. S. Mohamed, M. Mamat, M. Rivaie, S. M. Shaharuddin, *Global convergence of a new coefficient nonlinear conjugate gradient method*, Indones. J. Electr. Eng. Comput. Sci., **11** (2018), 1188–1193. 1, 2, 2

[28] L. Q. Qi, J. Sun, *A nonsmooth version of Newton's method*, Math. Programming, **58** (1993), 353–367. 1

[29] M. Rivaie, M. Mamat, L. W. June, I. Mohd, *A new class of nonlinear conjugate gradient coefficients with global convergence properties*, Appl. Math. Comput., **218** (2012), 11323–11332. 1

[30] M. V. Solodov, B. F. Svaiter, *A new projection method for variational inequality problems*, SIAM J. Control Optim., **37** (1999), 765–776. 1

[31] M. Sun, J. Liu, *Three derivative-free projection methods for nonlinear equations with convex constraints*, J. Appl. Math. Comput., **47** (2015), 265–276. 2

[32] N. Yamashita, M. Fukushima, *On the rate of convergence of the Levenberg-Marquardt method*, Comput. [Suppl.], **15** (2001), 239–249. 1

[33] Z. Yu, J. Lin, J. Sun, Y. Xiao, L. Liu, Z. Li, *Spectral gradient projection method for monotone nonlinear equations with convex constraints*, Appl. Numer. Math., **59** (2009), 2416–2423. 4.7

[34] L. Zhang, W. Zhou, *Spectral gradient projection method for solving nonlinear monotone equations*, J. Comput. Appl. Math., **196** (2006), 478–484. 1

Appendix

Table 1: Numerical result for Problem 4.1.

		Algorithm 1				Algorithm 2			
dm	inp	it	nf	tm	nm	it	nf	tm	nm
1000	c_1	3	11	0.22387	0	16	64	0.041872	3.45E-07
	c_2	2	7	0.02453	0	16	64	0.021963	7.03E-07
	c_3	3	11	0.019411	0	17	68	0.020096	6.22E-07
	c_4	2	7	0.008967	0	18	72	0.026791	4.54E-07
	c_5	2	7	0.019794	0	18	72	0.022601	3.65E-07
	c_6	2	7	0.010403	0	18	72	0.020692	3.80E-07
	c_7	19	75	0.087602	0	17	68	0.017729	7.12E-07
5000	c_1	2	7	0.035455	0	16	64	0.073457	7.61E-07
	c_2	2	7	0.021848	0	17	68	0.062864	5.15E-07
	c_3	2	7	0.020868	0	18	72	0.065321	4.63E-07
	c_4	2	7	0.020796	0	19	76	0.072119	3.38E-07
	c_5	2	7	0.089639	0	18	72	0.064029	8.12E-07
	c_6	2	7	0.025163	0	18	72	0.10743	8.10E-07
	c_7	72	287	1.0216	0	18	72	0.063575	5.37E-07
10000	c_1	2	7	0.065315	0	17	68	0.11957	3.55E-07
	c_2	2	7	0.02504	0	17	68	0.096546	7.27E-07
	c_3	2	7	0.022788	0	18	72	0.10444	6.55E-07
	c_4	2	7	0.11322	0	19	76	0.10849	4.77E-07
	c_5	2	7	0.041054	0	20	80	0.12003	4.52E-07
	c_6	2	7	0.030381	0	19	76	0.14566	5.51E-07
	c_7	244	975	11.9443	0	18	72	0.099343	7.51E-07
50000	c_1	2	7	0.10549	0	17	68	0.41409	7.93E-07
	c_2	2	7	0.13639	0	18	72	0.42281	5.44E-07
	c_3	2	7	0.13746	0	19	76	0.39061	4.86E-07
	c_4	2	7	0.1435	0	20	80	0.43511	9.70E-07
	c_5	2	7	0.11929	0	22	88	0.5831	8.63E-07
	c_6	2	7	0.18758	0	23	92	0.64319	8.62E-07
	c_7	77	307	7.5127	0	19	76	0.43164	5.61E-07
100000	c_1	2	7	0.21193	0	18	72	0.75865	3.76E-07
	c_2	2	7	0.18253	0	18	72	0.85139	7.69E-07
	c_3	2	7	0.17591	0	19	76	0.8479	6.88E-07
	c_4	2	7	0.20545	0	23	92	1.0896	3.63E-07
	c_5	2	7	0.381	0	23	92	1.043	9.61E-07
	c_6	2	7	0.20449	0	26	104	1.3727	3.39E-07
	c_7	141	563	32.7692	0	20	80	0.8401	7.79E-07

Table 2: Numerical result for Problem 4.2.

		Algorithm 1				Algorithm 2			
dm	inp	it	nf	tm	nm	it	nf	tm	nm
1000	c_1	7	22	0.057927	1.58E-09	13	51	0.010627	7.68E-07
	c_2	7	22	0.012172	2.12E-09	15	59	0.013658	3.49E-07
	c_3	6	19	0.005371	7.52E-09	16	63	0.018126	6.98E-07
	c_4	8	25	0.025657	1.95E-09	18	71	0.048427	3.52E-07
	c_5	6	19	0.009033	8.43E-09	18	71	0.019308	5.13E-07
	c_6	9	28	0.008674	1.04E-09	18	71	0.016487	8.59E-07
	c_7	11	35	0.010357	6.79E-07	17	67	0.014273	4.49E-07
5000	c_1	6	20	0.017838	2.97E-07	14	55	0.052256	5.44E-07
	c_2	6	20	0.051993	4.05E-07	15	59	0.057179	7.63E-07
	c_3	6	19	0.016847	9.12E-10	17	67	0.052789	5.12E-07
	c_4	7	23	0.025412	3.74E-07	18	71	0.074039	7.73E-07
	c_5	6	19	0.017764	1.42E-09	19	75	0.061007	3.75E-07
	c_6	7	22	0.014747	7.12E-09	19	75	0.061742	6.27E-07
	c_7	12	38	0.033328	6.79E-07	17	67	0.07213	9.83E-07
10000	c_1	5	16	0.024083	9.23E-09	14	55	0.11764	7.66E-07
	c_2	6	20	0.075326	3.06E-07	16	63	0.081869	3.55E-07
	c_3	6	19	0.032398	4.32E-10	17	67	0.10339	7.23E-07
	c_4	7	24	0.029118	2.82E-07	19	75	0.096522	3.63E-07
	c_5	6	20	0.028677	7.38E-10	19	75	0.11756	5.29E-07
	c_6	7	22	0.1375	4.21E-09	19	76	0.10636	9.51E-07
	c_7	11	38	0.077944	1.97E-07	18	71	0.10495	4.63E-07
50000	c_1	7	26	0.1369	1.84E-07	15	59	0.34239	5.78E-07
	c_2	9	34	0.12631	3.87E-07	16	63	0.37365	7.92E-07
	c_3	6	21	0.081129	5.88E-07	18	71	0.40685	5.36E-07
	c_4	10	37	0.15593	3.60E-07	21	84	0.45662	3.43E-07
	c_5	7	25	0.093439	1.16E-07	21	84	0.45842	4.72E-07
	c_6	8	28	0.10706	7.93E-07	21	84	0.50529	4.77E-07
	c_7	10	37	0.16389	9.96E-07	19	75	0.4452	3.45E-07
100000	c_1	7	26	0.59493	2.56E-07	15	59	0.62158	8.17E-07
	c_2	9	34	0.29254	5.47E-07	17	67	0.6947	3.76E-07
	c_3	6	21	0.31395	7.65E-07	18	72	0.77096	9.65E-07
	c_4	10	37	0.39074	5.09E-07	22	88	1.0101	8.28E-07
	c_5	7	25	0.34098	1.55E-07	22	88	1.0098	8.18E-07
	c_6	9	32	0.24409	1.09E-07	22	88	0.99244	7.87E-07
	c_7	11	41	0.37031	1.15E-07	20	80	0.92908	5.48E-07

Table 3: Numerical result for Problem 4.3.

		Algorithm 1				Algorithm 2			
dm	inp	it	nf	tm	nm	it	nf	tm	nm
1000	c_1	2	6	0.047463	0	2	6	0.004935	0
	c_2	2	6	0.004985	0	2	6	0.005237	0
	c_3	2	6	0.007876	0	2	6	0.004172	0
	c_4	3	11	0.007798	0	2	6	0.003795	0
	c_5	3	11	0.007161	0	2	6	0.00432	0
	c_6	3	11	0.008865	0	2	6	0.004887	0
	c_7	3	10	0.008971	0	2	6	0.005036	0
5000	c_1	2	6	0.012999	0	2	6	0.010986	0
	c_2	2	6	0.015079	0	2	6	0.011065	0
	c_3	2	6	0.01514	0	2	6	0.01249	0
	c_4	3	11	0.051959	0	2	6	0.010016	0
	c_5	3	11	0.023692	0	2	6	0.012051	0
	c_6	3	11	0.04786	0	2	6	0.009662	0
	c_7	3	10	0.029942	0	2	6	0.01144	0
10000	c_1	2	6	0.023004	0	2	6	0.021632	0
	c_2	2	6	0.023611	0	2	6	0.019348	0
	c_3	2	6	0.020701	0	2	6	0.032078	0
	c_4	3	11	0.046703	0	2	6	0.01331	0
	c_5	3	11	0.029767	0	2	6	0.045565	0
	c_6	3	11	0.030729	0	2	6	0.015755	0
	c_7	3	10	0.024323	0	2	6	0.028824	0
50000	c_1	2	6	0.075796	0	2	6	0.15535	0
	c_2	2	6	0.078935	0	2	6	0.097748	0
	c_3	2	6	0.18919	0	2	6	0.14607	0
	c_4	3	11	0.23517	0	2	6	0.084355	0
	c_5	3	11	0.14586	0	2	6	0.08018	0
	c_6	3	11	0.23072	0	2	6	0.21474	0
	c_7	3	10	0.10979	0	2	7	0.13907	0
100000	c_1	2	6	0.15384	0	2	6	0.21944	0
	c_2	2	6	0.16301	0	2	6	0.15044	0
	c_3	2	6	0.15059	0	2	6	0.13129	0
	c_4	3	11	0.31484	0	2	6	0.10428	0
	c_5	3	11	0.53464	0	2	6	0.094099	0
	c_6	3	11	0.28063	0	2	6	0.13453	0
	c_7	3	10	0.21087	0	2	7	0.22669	0

Table 4: Numerical result for Problem 4.4.

		Algorithm 1				Algorithm 2			
dm	inp	it	nf	tm	nm	it	nf	tm	nm
1000	c_1	2	7	0.021528	0	15	60	0.011562	5.13E-07
	c_2	2	7	0.004349	0	16	64	0.011605	3.59E-07
	c_3	2	7	0.005195	0	16	64	0.009487	9.42E-07
	c_4	2	7	0.006398	0	15	60	0.012216	6.44E-07
	c_5	2	7	0.00616	0	17	68	0.017389	3.91E-07
	c_6	2	7	0.006822	0	17	68	0.017897	7.89E-07
	c_7	41	163	0.091031	0	17	68	0.015525	5.18E-07
5000	c_1	2	7	0.014377	0	16	64	0.050295	3.86E-07
	c_2	2	7	0.019962	0	16	64	0.12969	8.02E-07
	c_3	2	7	0.02006	0	17	68	0.04733	7.00E-07
	c_4	2	7	0.020821	0	16	64	0.16294	4.74E-07
	c_5	2	7	0.053722	0	17	68	0.075782	8.74E-07
	c_6	2	7	0.023505	0	19	76	0.28728	5.11E-07
	c_7	57	227	0.3848	0	18	72	0.077147	3.67E-07
10000	c_1	2	7	0.018114	0	16	64	0.15223	5.46E-07
	c_2	2	7	0.015551	0	17	68	0.11014	3.76E-07
	c_3	2	7	0.019111	0	17	68	0.10562	9.90E-07
	c_4	2	7	0.050521	0	19	76	0.091896	3.70E-07
	c_5	2	7	0.11758	0	18	72	0.10508	4.15E-07
	c_6	2	7	0.049299	0	19	76	0.25366	7.22E-07
	c_7	155	619	3.5227	0	18	72	0.073714	5.08E-07
50000	c_1	2	7	0.068523	0	17	68	0.34355	4.04E-07
	c_2	2	7	0.11952	0	17	68	0.41732	8.40E-07
	c_3	2	7	0.06705	0	18	72	0.50331	7.39E-07
	c_4	2	7	0.076185	0	20	80	0.46207	6.25E-07
	c_5	2	7	0.13758	0	20	80	0.41103	8.13E-07
	c_6	2	7	0.090147	0	22	88	0.45026	9.65E-07
	c_7	261	1043	19.5919	0	19	76	0.61574	6.74E-07
100000	c_1	2	7	0.1258	0	17	68	0.9023	5.71E-07
	c_2	2	7	0.12298	0	18	72	0.62902	3.98E-07
	c_3	2	7	0.14553	0	19	76	0.55966	9.57E-07
	c_4	2	7	0.17899	0	22	88	0.79304	3.99E-07
	c_5	2	7	0.15521	0	24	96	0.87754	3.66E-07
	c_6	2	7	0.17655	0	26	104	0.86724	3.55E-07
	c_7	516	2063	86.4794	0	19	76	0.5515	9.54E-07

Table 5: Numerical result for Problem 4.5.

		Algorithm 1				Algorithm 2			
dm	inp	it	nf	tm	nm	it	nf	tm	nm
1000	c_1	21	80	0.24362	4.32E-07	19	75	0.011016	6.70E-07
	c_2	30	113	0.03131	7.77E-07	19	75	0.020435	6.02E-07
	c_3	61	240	0.11984	1.94E-07	20	79	0.013558	8.17E-07
	c_4	78	310	0.37303	2.03E-07	20	80	0.013375	4.14E-07
	c_5	107	425	0.25091	3.70E-07	20	80	0.013224	3.51E-07
	c_6	199	792	0.74997	5.00E-07	21	84	0.016755	3.89E-07
	c_7	115	459	0.24034	9.24E-07	29	115	0.024595	9.45E-07
5000	c_1	55	212	0.24991	1.67E-07	20	79	0.046749	6.26E-07
	c_2	34	131	0.17639	8.67E-07	20	79	0.067916	5.64E-07
	c_3	117	466	1.0295	2.48E-07	21	83	0.044109	7.12E-07
	c_4	149	594	1.6382	1.85E-07	21	84	0.056726	3.38E-07
	c_5	239	956	3.1091	9.65E-08	21	84	0.050799	4.47E-07
	c_6	319	1272	4.0241	3.52E-07	21	84	0.05015	6.59E-07
	c_7	212	844	2.144	9.57E-07	29	115	0.076333	3.95E-07
10000	c_1	56	217	0.63431	2.05E-07	20	79	0.097986	9.79E-07
	c_2	46	179	0.56127	4.30E-07	20	79	0.10212	8.67E-07
	c_3	183	729	3.3313	4.24E-07	22	87	0.11089	4.07E-07
	c_4	205	819	4.1869	1.98E-07	23	92	0.094223	4.76E-07
	c_5	388	1550	10.1934	1.07E-07	21	84	0.096989	7.05E-07
	c_6	439	1755	10.8848	9.10E-07	21	84	0.08221	5.31E-07
	c_7	294	1174	6.1579	2.53E-07	23	92	0.096241	9.31E-07
50000	c_1	86	336	4.2367	3.00E-07	23	92	0.37093	4.69E-07
	c_2	77	304	4.0044	9.55E-08	23	92	0.3743	4.37E-07
	c_3	444	1773	51.5065	4.96E-07	22	88	0.34276	8.93E-07
	c_4	483	1932	73.1707	1.11E-07	24	96	0.40374	5.83E-07
	c_5	NaN	NaN	NaN	NaN	24	96	0.44806	5.87E-07
	c_6	NaN	NaN	NaN	NaN	23	92	0.42548	8.28E-07
	c_7	525	2096	72.6984	8.52E-08	26	104	0.44434	4.70E-07
100000	c_1	118	467	18.4649	4.85E-07	24	96	0.79669	8.11E-07
	c_2	153	607	27.5528	1.46E-07	24	96	0.77883	7.59E-07
	c_3	364	1449	72.9234	7.58E-07	23	92	0.70613	4.30E-07
	c_4	564	2254	127.6077	2.47E-07	25	100	0.80085	3.79E-07
	c_5	NaN	NaN	NaN	NaN	25	100	0.85008	5.83E-07
	c_6	NaN	NaN	NaN	NaN	26	104	0.8879	3.96E-07
	c_7	NaN	NaN	NaN	NaN	24	96	0.7836	9.28E-07

Table 6: Numerical result for Problem 4.6.

		Algorithm 1				Algorithm 2			
dm	inp	it	nf	tm	nm	it	nf	tm	nm
1000	c_1	9	36	0.040786	8.24E-07	18	72	0.038391	4.82E-07
	c_2	9	36	0.009086	7.93E-07	18	72	0.01727	4.64E-07
	c_3	9	36	0.010292	6.98E-07	18	72	0.017788	4.08E-07
	c_4	9	36	0.009416	4.78E-07	17	68	0.023552	8.34E-07
	c_5	9	36	0.015915	3.83E-07	17	68	0.015462	6.69E-07
	c_6	9	36	0.009282	2.26E-07	17	68	0.016267	3.94E-07
	c_7	9	36	0.00898	7.01E-07	18	72	0.01953	4.12E-07
5000	c_1	10	40	0.030792	1.85E-07	19	76	0.074066	3.58E-07
	c_2	10	40	0.032701	1.78E-07	19	76	0.073348	3.44E-07
	c_3	10	40	0.030782	1.57E-07	18	72	0.071951	9.14E-07
	c_4	10	40	0.041305	1.07E-07	18	72	0.068405	6.26E-07
	c_5	9	36	0.032659	8.61E-07	18	72	0.079743	5.02E-07
	c_6	9	36	0.029737	5.08E-07	17	68	0.075313	8.83E-07
	c_7	10	40	0.033865	1.58E-07	18	72	0.081145	9.22E-07
10000	c_1	10	40	0.058927	2.62E-07	21	84	0.19184	4.00E-07
	c_2	10	40	0.053023	2.52E-07	21	84	0.17222	3.85E-07
	c_3	10	40	0.056913	2.22E-07	20	80	0.17532	5.83E-07
	c_4	10	40	0.063291	1.52E-07	18	72	0.14316	8.85E-07
	c_5	10	40	0.052705	1.22E-07	18	72	0.13431	7.10E-07
	c_6	9	36	0.044005	7.18E-07	18	72	0.13244	4.19E-07
	c_7	10	40	0.067819	2.23E-07	20	80	0.17845	5.88E-07
50000	c_1	10	40	0.20709	5.85E-07	24	96	0.82383	7.08E-07
	c_2	10	40	0.18537	5.63E-07	24	96	0.77604	6.81E-07
	c_3	10	40	0.22992	4.96E-07	23	92	0.74561	7.26E-07
	c_4	10	40	0.21739	3.40E-07	21	84	0.64599	5.18E-07
	c_5	10	40	0.2151	2.72E-07	21	84	0.66372	4.16E-07
	c_6	10	40	0.18323	1.61E-07	18	72	0.59112	9.36E-07
	c_7	10	40	0.2104	5.00E-07	23	92	0.73695	7.32E-07
100000	c_1	10	40	0.42899	8.28E-07	29	116	2.308	5.93E-07
	c_2	10	40	0.39985	7.96E-07	28	112	2.191	6.09E-07
	c_3	10	40	0.40936	7.01E-07	26	104	1.9182	6.39E-07
	c_4	10	40	0.43014	4.80E-07	23	92	1.604	7.03E-07
	c_5	10	40	0.45867	3.85E-07	22	88	1.4573	3.66E-07
	c_6	10	40	0.39766	2.27E-07	20	80	1.2933	5.97E-07
	c_7	10	40	0.41225	7.08E-07	26	104	1.9593	6.44E-07

Table 7: Numerical result for Problem 4.7.

		Algorithm 1				Algorithm 2			
dm	inp	it	nf	tm	nm	it	nf	tm	nm
1000	c_1	5	20	0.02835	3.24E-07	17	68	0.015365	6.92E-07
	c_2	5	20	0.00849	1.43E-07	17	68	0.013959	4.34E-07
	c_3	5	20	0.02459	1.68E-08	5	20	0.007568	4.50E-08
	c_4	6	24	0.009991	9.16E-09	18	72	0.016767	8.82E-07
	c_5	6	24	0.008829	1.23E-08	19	76	0.015507	8.09E-07
	c_6	6	23	0.010687	1.04E-07	18	71	0.018398	5.23E-07
	c_7	14	56	0.019819	8.59E-08	19	76	0.016317	4.01E-07
5000	c_1	5	20	0.020684	7.25E-07	18	72	0.05213	5.59E-07
	c_2	5	20	0.019194	3.20E-07	17	68	0.0552	9.70E-07
	c_3	5	20	0.021205	3.75E-08	5	20	0.018794	1.01E-07
	c_4	6	24	0.023976	2.05E-08	19	76	0.057299	7.14E-07
	c_5	6	24	0.020556	2.75E-08	20	80	0.068109	6.56E-07
	c_6	6	23	0.021564	2.32E-07	19	75	0.05434	4.22E-07
	c_7	12	48	0.045511	1.54E-07	19	76	0.071732	9.23E-07
10000	c_1	6	24	0.03415	5.12E-09	18	72	0.092741	7.90E-07
	c_2	5	20	0.036905	4.52E-07	18	72	0.096713	4.95E-07
	c_3	5	20	0.038781	5.31E-08	5	20	0.022365	1.42E-07
	c_4	6	24	0.033419	2.90E-08	20	80	0.14743	3.66E-07
	c_5	6	24	0.031172	3.89E-08	20	80	0.099693	9.28E-07
	c_6	6	23	0.029565	3.28E-07	21	84	0.12772	4.36E-07
	c_7	19	76	0.10999	1.97E-08	20	80	0.11279	4.60E-07
50000	c_1	6	24	0.1595	1.15E-08	19	76	0.37052	6.42E-07
	c_2	6	24	0.15396	5.06E-09	19	76	0.36384	4.02E-07
	c_3	5	20	0.099927	1.19E-07	5	20	0.092183	3.18E-07
	c_4	6	24	0.11421	6.48E-08	21	84	0.45906	8.23E-07
	c_5	6	24	0.13876	8.70E-08	21	84	0.43542	7.14E-07
	c_6	6	23	0.11134	7.35E-07	21	84	0.46337	9.75E-07
	c_7	15	60	0.35451	2.17E-07	21	84	0.45523	3.79E-07
100000	c_1	6	24	0.22565	1.62E-08	20	80	0.77259	7.45E-07
	c_2	6	24	0.22915	7.15E-09	19	76	0.68732	5.69E-07
	c_3	5	20	0.19636	1.68E-07	5	20	0.22767	4.50E-07
	c_4	6	24	0.23499	9.16E-08	22	88	1.4285	4.22E-07
	c_5	6	24	0.27449	1.23E-07	22	88	0.90341	7.50E-07
	c_6	7	27	0.26052	5.19E-09	22	88	0.87859	5.00E-07
	c_7	12	48	0.53364	1.96E-08	20	80	0.85813	6.66E-07

Table 8: Numerical result for Problem 4.8.

		Algorithm 1				Algorithm 2			
dm	inp	it	nf	tm	nm	it	nf	tm	nm
1000	c_1	13	48	0.28499	NaN	36	144	0.19199	6.34E-07
	c_2	206	824	2.7554	3.27E-07	35	140	0.17837	9.13E-07
	c_3	71	284	1.2452	3.58E-07	35	140	0.1615	7.34E-07
	c_4	NaN	NaN	NaN	NaN	33	132	0.14206	2.30E-07
	c_5	NaN	NaN	NaN	NaN	31	124	0.14692	8.06E-07
	c_6	4	14	0.033612	NaN	24	96	0.13093	9.72E-07
	c_7	25	97	0.32507	NaN	29	116	0.13481	2.49E-07
5000	c_1	199	796	11.1914	9.18E-07	34	136	0.72482	8.36E-07
	c_2	10	37	0.56969	NaN	34	136	0.7476	7.93E-07
	c_3	NaN	NaN	NaN	NaN	34	136	0.73337	6.18E-07
	c_4	NaN	NaN	NaN	NaN	31	124	0.67258	3.90E-07
	c_5	21	79	1.0588	NaN	30	120	0.69797	8.11E-07
	c_6	28	103	1.2647	NaN	24	96	0.52499	7.51E-07
	c_7	8	29	0.23404	NaN	24	96	0.54913	2.91E-07
10000	c_1	NaN	NaN	NaN	NaN	34	136	1.3986	6.78E-07
	c_2	NaN	NaN	NaN	NaN	34	136	1.3769	6.42E-07
	c_3	61	244	5.0283	3.56E-07	33	132	1.3628	7.57E-07
	c_4	125	500	11.3843	8.81E-07	30	120	1.2608	3.94E-07
	c_5	17	63	1.3152	NaN	30	120	1.2407	5.57E-07
	c_6	275	1098	24.2426	4.14E-07	24	96	0.99676	7.21E-07
	c_7	107	425	14.0895	NaN	27	108	1.1274	3.65E-07
50000	c_1	41	159	18.506	NaN	34	136	6.0333	6.35E-07
	c_2	37	144	29.7738	NaN	33	132	5.9281	6.12E-07
	c_3	113	451	52.1981	4.09E-07	32	128	6.1533	7.22E-07
	c_4	11	41	4.0036	NaN	24	96	5.3594	3.36E-07
	c_5	22	84	13.935	NaN	29	116	6.3535	5.83E-07
	c_6	12	44	5.9726	NaN	31	124	8.4474	7.91E-07
	c_7	11	42	3.1947	NaN	26	104	10.049	3.36E-07
100000	c_1	40	156	51.3509	NaN	33	132	16.1737	8.00E-07
	c_2	5	17	2.7955	NaN	33	132	14.174	7.49E-07
	c_3	126	503	115.4434	4.01E-07	40	160	17.8687	9.75E-07
	c_4	55	215	64.8415	NaN	30	120	12.3492	9.85E-07
	c_5	5	17	2.8864	NaN	28	112	10.6799	9.46E-07
	c_6	NaN	NaN	NaN	NaN	26	104	10.1937	9.05E-07
	c_7	10	38	7.026	NaN	30	120	11.5463	2.57E-07

Table 9: Numerical result for Problem 4.9.

		Algorithm 1				Algorithm 2			
dm	inp	it	nf	tm	nm	it	nf	tm	nm
1000	c_1	10	34	0.025676	1.06E-07	11	42	0.00812	2.67E-07
	c_2	10	34	0.006497	1.06E-07	11	42	0.008082	2.67E-07
	c_3	10	34	0.006769	1.06E-07	11	42	0.009859	2.67E-07
	c_4	10	34	0.006506	1.06E-07	11	42	0.008471	2.67E-07
	c_5	10	34	0.006569	1.06E-07	11	42	0.01137	2.67E-07
	c_6	10	34	0.005753	1.06E-07	12	46	0.009152	2.67E-07
	c_7	10	34	0.006338	1.06E-07	11	42	0.007492	2.67E-07
5000	c_1	7	25	0.017159	6.89E-08	8	31	0.032027	1.59E-07
	c_2	7	25	0.016181	6.89E-08	8	31	0.030725	1.59E-07
	c_3	7	25	0.019768	6.89E-08	8	31	0.025114	1.59E-07
	c_4	7	25	0.015824	6.89E-08	9	35	0.036701	1.59E-07
	c_5	7	25	0.018099	6.89E-08	9	35	0.032259	1.59E-07
	c_6	7	25	0.019327	6.89E-08	9	35	0.047628	1.59E-07
	c_7	7	25	0.019407	6.89E-08	8	31	0.035816	1.59E-07
10000	c_1	6	22	0.036419	8.13E-08	11	43	0.074354	7.22E-07
	c_2	6	22	0.041491	8.13E-08	11	43	0.077893	7.22E-07
	c_3	6	22	0.037815	8.13E-08	11	43	0.08316	7.22E-07
	c_4	6	22	0.038765	8.13E-08	12	47	0.10473	7.22E-07
	c_5	6	22	0.039708	8.13E-08	13	51	0.14345	7.22E-07
	c_6	6	22	0.033133	8.13E-08	13	51	0.13929	7.22E-07
	c_7	6	22	0.038275	8.13E-08	11	43	0.10201	7.22E-07
50000	c_1	5	19	0.18677	1.41E-07	10	40	0.36425	7.59E-07
	c_2	5	19	0.15923	1.41E-07	10	40	0.35055	7.59E-07
	c_3	5	19	0.18752	1.41E-07	11	44	0.4686	7.59E-07
	c_4	5	19	0.16679	1.41E-07	13	52	0.67096	7.59E-07
	c_5	5	19	0.15913	1.41E-07	14	56	0.83025	7.59E-07
	c_6	5	19	0.16621	1.41E-07	16	64	1.0331	7.59E-07
	c_7	5	19	0.17155	1.41E-07	11	44	0.4487	7.59E-07
100000	c_1	6	23	0.49529	2.10E-07	9	36	0.65953	2.19E-07
	c_2	6	23	0.49149	2.10E-07	9	36	0.66909	2.19E-07
	c_3	6	23	0.46091	2.10E-07	11	44	1.0605	2.19E-07
	c_4	6	23	0.55906	2.10E-07	14	56	1.7111	2.19E-07
	c_5	6	23	0.47421	2.10E-07	16	64	2.2975	2.19E-07
	c_6	6	23	0.47836	2.10E-07	18	72	3.0334	2.19E-07
	c_7	6	23	0.48422	2.10E-07	11	44	1.0618	2.19E-07