Journal of Computational and Applied Mathematics 224 (2009) 11–19

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A family of derivative-free conjugate gradient methods for large-scale nonlinear systems of equations☆

Wanyou Cheng [a,*], Yunhai Xiao [b], Qing-Jie Hu [c]

[a] College of Software, Dongguan University of Technology, Dongguan 523000, PR china
[b] Institute of Applied Mathematics, College of Mathematics and Information Science, Henan University, Kaifeng, 475004, PR China
[c] Department of Information, Hunan Business College, Changsha, 410205, PR China

ARTICLE INFO

Article history:
Received 2 February 2007
Received in revised form 18 December 2007

MSC:
90C25
90C33

Keywords:
Conjugate gradient method
Nonmonotone line search
Global convergence

ABSTRACT

In this paper, we propose a family of derivative-free conjugate gradient methods for large-scale nonlinear systems of equations. They come from two modified conjugate gradient methods [W.Y. Cheng, A two term PRP based descent Method, Numer. Funct. Anal. Optim. 28 (2007) 1217–1230; L. Zhang, W.J. Zhou, D.H. Li, A descent modified Polak–Ribiére–Polyak conjugate gradient method and its global convergence, IMA J. Numer. Anal. 26 (2006) 629–640] recently proposed for unconstrained optimization problems. Under appropriate conditions, the global convergence of the proposed method is established. Preliminary numerical results show that the proposed method is promising.

1. Introduction

In this paper, we consider nonlinear systems of equations

$$g(x) = 0, \quad x \in R^n, \tag{1.1}$$

where $g(x)$ is a continuously differentiable mapping from R^n into itself. We are interested in the large-scale case for which the Jacobian of $g(x)$ is either not available or requires a low amount of storage.

Many methods for solving (1.1) fall into the Newton and quasi-Newton strategy [2–5,10,11,13,16,17,19,20]. These methods are attractive because they converge rapidly from a sufficiently good initial guess. They are typically unattractive for large-scale nonlinear systems of equations because they need to solve a linear system using the Jacobian matrix or an approximation of it.

Recently, the spectral gradient method [1] has been extended to solve large-scale nonlinear systems of equations [7,9,24]. La Cruz and Raydan [7] introduced a spectral algorithm (SANE). Global convergence is guaranteed by means of a variation of the nonmonotone line search in [12]. La Cruz, Martínez and Raydan [9] proposed a new derivative-free line search and developed the DF-SANE algorithm. Numerical experiments show that DF-SANE is very effective. Zhang and Zhou [24] combined the spectral gradient method [1] and project method [23] to solve nonlinear monotone equations. We refer to review papers [18,20] for a summary of nonlinear systems of equations.

The conjugate gradient methods are welcome methods for unconstrained optimization problems. They are particularly efficient for large-scale problems due to their simplicity and low storage [14]. However, the study of conjugate gradient

☆ This work was supported by the 973 project grant 2004CB719402 and the NSF project of China grant 10771057.
* Corresponding author. Tel.: +86 0731 8820195.
E-mail address: chengwanyou421@yahoo.com.cn (W. Cheng).

0377-0427/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2008.03.050

methods for large-scale nonlinear systems of equations is rare. This motivated the paper. Quite recently Zhang, Zhou and Li [25] proposed a three-term modified PRP method (TTPRP) and Cheng [6] proposed a two-term modified PRP method (TMPRP). The reported numerical results show they are competitive with the CG_DESCENT method [15]. As an attempt, we extend the two modified conjugate gradient methods to solve (1.1). An attractive feature of the algorithm is that the Jacobian of g is not fully used. Moreover, preliminary numerical results indicate that the proposed method is promising.

The paper is organized as follows. In the next section, we briefly recall conjugate gradient methods for unconstrained optimization problems and propose the algorithm. In Section 3, the global convergence of the algorithm is established. We report the numerical results in the last section.

Throughout the paper, we use $J(x)$ to denote the Jacobian matrix of g at x. We use $\|\cdot\|$ to denote the Euclidean norm of vectors. We denote by $\mathcal{N}$ the natural numbers.

2. Algorithm

In this section, we first focus on conjugate gradient methods for the unconstrained optimization problem

$$\min\{f(x) : x \in R^n\},$$

where $f : R^n - R$ is a continuously differentiable function and its gradient $\nabla f(x)$ is available. Nonlinear conjugate gradient methods generate a sequence $\{x_k\}$ by

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, \ldots$$

where the steplength α_k is determined by a line search rule and the search direction d_k is generated by

$$d_0 = -\nabla f(x_0), \qquad d_k = -\nabla f(x_k) + \beta_k d_{k-1}, \quad \forall\, k \geq 1,$$

where β_k is a scalar.

Recently, Zhang, Zhou and Li [25] proposed the TTPRP method and the search direction has the form

$$d_0 = -\nabla f(x_0), \qquad d_k = -\nabla f(x_k) + \beta_k^{PRP} d_{k-1} - \eta_k^* y_{k-1}^*, \quad \forall\, k \geq 1, \tag{2.1}$$

where

$$\beta_k^{PRP} = \frac{\nabla f(x_k)^T y_{k-1}^*}{\|\nabla f(x_{k-1})\|^2}, \qquad \eta_k^* = \frac{\nabla f(x_k)^T d_{k-1}}{\|\nabla f(x_{k-1})\|^2} \quad \text{and} \quad y_{k-1}^* = \nabla f(x_k) - \nabla f(x_{k-1}).$$

It is clear to see that the search direction (2.1) satisfies

$$\nabla f(x_k)^T d_k = -\|\nabla f(x_k)\|^2. \tag{2.2}$$

Consequently, d_k is a sufficient descent direction of f at x_k. Cheng [6] proposed the TMPRP method. The search direction of the TMPRP method has the form

$$d_0 = -\nabla f(x_0), \qquad d_k = -(1 + \theta_k^*)\nabla f(x_k) + \beta_k^{PRP} d_{k-1}, \quad \forall\, k \geq 1, \tag{2.3}$$

where

$$\theta_k^* = \beta_k^{PRP} \frac{\nabla f(x_k)^T d_{k-1}}{\|\nabla f(x_k)\|^2}.$$

It is clear that the search direction (2.3) satisfies (2.2). The reported numerical results show that the two modified conjugate gradient methods perform better than the PRP method [21,22] and are competitive with the CG_DESCENT method [15].

As an attempt, we extend them to solve (1.1). We consider the search direction d_k^1 (denotes d_k determined by (2.1)) and the search direction d_k^2 (denotes d_k determined by (2.3)), a line combination

$$d_k = (1 - \lambda_k) d_k^1 + \lambda_k d_k^2, \tag{2.4}$$

where $\{\lambda_k\}$ is a bounded sequence. The direction (2.4) can be rewritten as

$$d_k = \begin{cases} -\nabla f(x_0), & \text{if } k = 0, \\ -(1 + \lambda_k \theta_k^*)\nabla f(x_k) + \beta_k^{PRP} d_{k-1} - (1 - \lambda_k)\eta_k^* y_{k-1}^*, & \text{if } k \geq 1. \end{cases} \tag{2.5}$$

We construct the search direction with the form (2.5) only from theoretical point of view. Observe that if we set $\lambda_k = 0$, then we get the TTPRP method, while $\lambda_k = 1$ yields the TMPRP method.

From now on, we pay attention to solving (1.1). Our method has the iterative form

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, \ldots$$

where α_k is the steplength that is determined by a nonmonotone line search which will be defined later and the search direction d_k has the following form

$$d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -(1 + \lambda_k \theta_k) g_k + \beta_k^{PRP} d_{k-1} - (1 - \lambda_k)\eta_k y_{k-1}, & \text{if } k \geq 1, \end{cases} \tag{2.6}$$

where

$$\beta_k^{PRP} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2}, \qquad \theta_k = \beta_k^{PRP} \frac{g_k^T d_{k-1}}{\|g_k\|^2}, \qquad \eta_k = \frac{g_k^T d_{k-1}}{\|g_{k-1}\|^2} \quad \text{and} \quad y_{k-1} = g_k - g_{k-1}.$$

Now we are ready to state the steps of our method for nonlinear systems of equations.

Algorithm 2.1 (*DF-SDCG*).

Step1. Given an initial point $x_0 \in R^n$ and a positive integer M. Let $0 < \rho_{\min} < \rho_{\max} < 1$, $0 < \sigma_{\min} < \sigma_{\max}$ and $\gamma_1, \gamma_2 > 0$ be given positive constants. Select a bounded sequence $\{\lambda_k\}$ and a positive sequence $\{\epsilon_k\}$ that satisfies $\sum_{k=0}^{\infty} \epsilon_k < \infty$. Set $d_0 = -g(x_0)$ and $k = 0$.

Step2. Chose an initial steplength σ_k such that $|\sigma_k| \in [\sigma_{\min}, \sigma_{\max}]$. Set $\alpha_+ = 1$ and $\alpha_- = 1$.

Step3. Nonmonotone line search.

`If`

$$\|g(x_k + \alpha_+\sigma_k d_k)\|^2 \le \max_{0\le j\le \min\{k,\ M-1\}} \|g(x_{k-j})\|^2 - \gamma_1\|\alpha_+\sigma_k g_k\|^2 - \gamma_2\|\alpha_+\sigma_k d_k\|^2 + \epsilon_k, \tag{2.7}$$

`then` define $\alpha_k = \alpha_+|\sigma_k|$, $d_k = sgn(\sigma_k)d_k$ and update $x_{k+1} = x_k + \alpha_k d_k$.

`Else if`

$$\|g(x_k - \alpha_-\sigma_k d_k)\|^2 \le \max_{0\le j\le \min\{k,\ M-1\}} \|g(x_{k-j})\|^2 - \gamma_1\|\alpha_-\sigma_k g_k\|^2 - \gamma_2\|\alpha_-\sigma_k d_k\|^2 + \epsilon_k,$$

`then` define $\alpha_k = \alpha_-|\sigma_k|$, $d_k = -sgn(\sigma_k)d_k$ and update $x_{k+1} = x_k + \alpha_k d_k$.

`else` choose $\alpha_{+\text{new}} \in [\rho_{\min}\alpha_+, \rho_{\max}\alpha_+]$, $\alpha_{-new} \in [\rho_{\min}\alpha_-, \rho_{\max}\alpha_-]$. Replace $\alpha_+ = \alpha_{+\text{new}}$, $\alpha_- = \alpha_{-new}$ and go to Step 3.

`End if`

Step4. Convergence check.

Step5. Update d_{k+1} by (2.6), set $k = k + 1$ and go to step 2.

Remark. (1) We can see that the line search (2.7) is a modified form that was used in [9]. (2) Since $\epsilon_k > 0$, after a finite number of reductions of α_+ the condition (2.7) necessarily holds. So the line search process, i.e., Step 3 of Algorithm 2.1, is well defined.

3. Convergence analysis

This section is devoted to the global convergence of Algorithm 2.1. We first make some assumptions.

Assumption 3.1. (i) The level set $\Omega = \{x|\ \|g(x)\| \le \sqrt{\|g(x_0)\|^2 + \eta}\}$ is bounded, where η is a positive constant such that $\sum_{k=0}^{\infty} \epsilon_k \le \eta$.

(ii) In some neighborhood Γ of Ω, the nonlinear mapping $g(x)$ has continuous partial derivatives and is Lipschitz continuous, namely, there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \le L\|x - y\|, \quad \forall x, y \in \Gamma. \tag{3.1}$$

The assumption indicates that there exists a positive constant γ such that

$$\|g(x)\| \le \gamma, \quad \forall\, x \in \Omega. \tag{3.2}$$

Before we proceed with the convergence analysis, we firstly state some preliminary definitions. Define $V_0 = \|g(x_0)\|^2$ and

$$V_k = \max\{\|g(x_{(k-1)M+1})\|^2, \ldots, \|g(x_{kM})\|^2\}, \quad \forall\, k = 1, 2, \ldots.$$

Let $\nu_{(k)} \in \{(k-1)M + 1, \ldots, kM\}$ be such that for all $k = 1, 2, \ldots,$

$$\|g(x_{\nu(k)})\|^2 = V_k.$$

Proceeding similarly as in the proofs of Proposition 2 and Proposition 3 in [9], we get the following two lemmas.

Lemma 3.1. *For all $k, l \in \mathcal{N}$, we have*

$$\|g(x_{kM+l})\|^2 \le \|g(x_{\nu(k)})\|^2 + \eta.$$

Lemma 3.2. *Suppose that Assumption 3.1 holds. Let $\{x_k\}$ be generated by Algorithm 2.1. Then we have*

$$\lim_{k\to\infty} \|\alpha_{\nu(k)-1} d_{\nu(k)-1}\| = 0 \tag{3.3}$$

and

$$\lim_{k\to\infty} \|\alpha_{\nu(k)-1} g_{\nu(k)-1}\| = 0. \tag{3.4}$$

From now on we define $K = \{\nu(1)-1, \nu(2)-1, \nu(3)-1, \ldots\}$. The following lemma shows that the search direction d_k is bounded if the current point x_k is not the solution of (1.1).

Lemma 3.3. *Suppose that Assumption 3.1 holds and d_k is determined by (2.6). If there exists a constant $\epsilon > 0$ such that*

$$\|g_k\| \geq \epsilon, \quad \forall\, k \in K, \tag{3.5}$$

then there exists a positive constant M such that

$$\|d_k\| \leq M, \quad \forall\, k \in K. \tag{3.6}$$

Proof. From the definition of d_k in (2.6), (3.5) and Assumption 3.1, we have

$$\begin{aligned}
\|d_k\| &\leq \|g_k\| + |\lambda_k||\theta_k|\|g_k\| + |(\lambda_k + 1 - \lambda_k)\beta_k^{PRP}|\|d_{k-1}\| + |1-\lambda_k|\|\eta_k y_{k-1}\| \\
&\leq \|g_k\| + |\lambda_k||\beta_k^{PRP}|\frac{\|g_k\|\|d_{k-1}\|}{\|g_k\|^2}\|g_k\| + |\lambda_k||\beta_k^{PRP}|\|d_{k-1}\| + |1-\lambda_k|\left(|\beta_k^{PRP}|\|d_{k-1}\| + \frac{\|g_k^T d_{k-1}\|\|y_{k-1}\|}{\|g_{k-1}\|^2}\right) \\
&\leq \|g_k\| + 2|\lambda_k||\beta_k^{PRP}|\|d_{k-1}\| + 2|1-\lambda_k|\frac{\|g_k\|\ \|y_{k-1}\|}{\|g_{k-1}\|^2}\|d_{k-1}\| \\
&\leq \|g_k\| + 2(|\lambda_k| + |1-\lambda_k|)\frac{\|y_{k-1}\|\|g_k\|}{\|g_{k-1}\|^2}\|d_{k-1}\| \\
&\leq \gamma + 2(|\lambda_k| + |1-\lambda_k|)\frac{\gamma L\alpha_{k-1}\|d_{k-1}\|}{\epsilon^2}\|d_{k-1}\|.
\end{aligned}$$

Since $\{\lambda_k\}$ is a bounded sequence, we get from (3.3) that there exist a constant $q \in (0, 1)$ and an integer n_0 such that for all $k > n_0$ with $k \in K$

$$2(|\lambda_k| + |1-\lambda_k|)\frac{\gamma L\alpha_{k-1}\|d_{k-1}\|}{\epsilon^2} < q.$$

Hence for any $k > n_0$ with $k \in K$, we have

$$\begin{aligned}
\|d_k\| &\leq \gamma + q\|d_{k-1}\| \\
&\leq \gamma(1 + q + q^2 + \cdots + q^{k-n_0+1}) + q^{k-n_0}\|d_{n_0}\| \\
&\leq \frac{\gamma}{1-q} + \|d_{n_0}\|.
\end{aligned}$$

Setting $M = \max\{\|d_1\|, \|d_2\|, \ldots, \|d_{n_0}\|, \frac{\gamma}{1-q} + \|d_{n_0}\|\}$, we deduce (3.6). $\square$

Lemma 3.4. *Suppose that Assumption 3.1 holds and d_k is determined by (2.6). If there exists a constant $\epsilon > 0$ such that for all $k \in K$*

$$\|g_k\| \geq \epsilon, \tag{3.7}$$

then we have

$$\lim_{k\to\infty} \beta_k^{PRP}\|d_{k-1}\| = 0 \tag{3.8}$$

and

$$\lim_{k\to\infty} \lambda_k\theta_k g_k + (1-\lambda_k)\eta_k y_k = 0. \tag{3.9}$$

Proof. First, by (3.1)–(3.3), (3.6) and (3.7), we have

$$|\beta_k^{PRP}|\|d_{k-1}\| = \frac{|g_k^T(g_k - g_{k-1})|}{\|g_{k-1}\|^2}\|d_{k-1}\| \leq \frac{\|g_k\|L\alpha_{k-1}\|d_{k-1}\|^2}{\|g_{k-1}\|^2} \to 0 \quad \text{as } k \to \infty,$$

which shows (3.8).

Second, from (3.1)–(3.3), (3.6) and (3.7), we have

$$\lambda_k\theta_k g_k = \lambda_k\beta_k^{PRP}\frac{g_k^T d_{k-1}}{\|g_k\|^2}g_k = \lambda_k\frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2}\frac{g_k^T d_{k-1}}{\|g_k\|^2}g_k \le \lambda_k\frac{L\alpha_{k-1}\|d_{k-1}\|^2}{\|g_{k-1}\|^2} \to 0 \quad \text{as } k\to\infty \tag{3.10}$$

and

$$(1-\lambda_k)\eta_k y_k = (1-\lambda_k)\frac{g_k^T d_{k-1}}{\|g_{k-1}\|^2}y_k \le (1-\lambda_k)\frac{\|g_k\|L\alpha_{k-1}\|d_{k-1}\|^2}{\|g_{k-1}\|^2} \to 0 \quad \text{as } k\to\infty. \tag{3.11}$$

Hence, (3.10) and (3.11) imply (3.9). □

The following theorem establishes the global convergence of Algorithm 2.1. It is similar to Theorem 1 of [9].

Theorem 3.1. *Suppose that Assumption 3.1 holds. Let $\{x_k\}$ be generated by Algorithm 2.1. Then we have*

$$\liminf_{k\to\infty}\|g_k\| = 0, \tag{3.12}$$

or every limit point x^* of $\{x_k\}_K$ satisfies

$$g(x^*)^T J(x^*) g(x^*) = 0. \tag{3.13}$$

In particular, if g is strict, namely, g or $-g$ is strictly monotone, then the whole sequence $\{x_k\}$ converges to the unique solution of equation of (1.1).

Proof. Let x^* be any limit point of $\{x_k\}_K$ and let $K_1\subset K$ be an infinite index set such that $\lim_{k\in K_1} x_k = x^*$. By (3.4), we have $\lim_{k\in K_1}\|\alpha_k g_k\| = 0$.

`Case I:` If $\lim_{k\in K_1}\sup\alpha_k \ne 0$, then there exists an infinite index set $K_2\subset K_1$ such that $\{\alpha_k\}_{K_2}$ is bounded away from zero. By (3.4), we have $\lim_{k\in K_2}\|g(x_k)\| = 0$. Since g is continuous and $\lim_{k\in K_2} x_k = x^*$, we have (3.12).

`Case II:` If

$$\lim_{k\in K_1}\alpha_k = 0. \tag{3.14}$$

From (3.14), we can suppose that in Algorithm 2.1 step k (i.e., the step which generates x_{k+1}) α_+ and α_- were adapted m_k ($m_k > 1$) times in the line search process. Let α_k^+ and α_k^- be the values of α_+ and α_- respectively in the last unsuccessful steplength. By the choice of $\alpha_{+\text{new}}$ and α_{-new} in Step 3 of Algorithm 2.1, we have that

$$\alpha_k \ge \rho_{\min}^{m_k}$$

for all $k > k_0$ with $k\in K_1$. By (3.14), we have $\lim_{k\in K_1} m_k = \infty$. From the choice of $\alpha_{+\text{new}}$ and α_{-new}, we have

$$\alpha_k^+ \le \rho_{\max}^{m_k-1}$$

and

$$\alpha_k^- \le \rho_{\max}^{m_k-1}.$$

Since $\rho_{\max} < 1$ and $\lim_{k\in K_1} m_k = \infty$, we get

$$\lim_{k\in K_1}\alpha_k^+ = \lim_{k\in K_1}\alpha_k^- = 0.$$

By the line search rule, we have

$$\begin{aligned}\|g(x_k+\alpha_k^+\sigma_k d_k)\|^2 - \|g_k\|^2 &\ge \|g(x_k+\alpha_k^+\sigma_k d_k)\|^2 - \max_{0\le j\le\min\{k,\,M-1\}}\|g_k\|^2\\ &> -\gamma_1\|\alpha_k^+\sigma_k g_k\|^2 - \gamma_2\|\alpha_k^+\sigma_k d_k\|^2 + \epsilon_k\\ &> -\gamma_1\|\alpha_k^+\sigma_k g_k\|^2 - \gamma_2\|\alpha_k^+\sigma_k d_k\|^2\end{aligned}$$

and

$$\begin{aligned}\|g(x_k-\alpha_k^-\sigma_k d_k)\|^2 - \|g_k\|^2 &\ge \|g(x_k-\alpha_k^-\sigma_k d_k)\|^2 - \max_{0\le j\le\min\{k,\,M-1\}}\|g_k\|^2\\ &> -\gamma_1\|\alpha_k^-\sigma_k g_k\|^2 - \gamma_2\|\alpha_k^-\sigma_k d_k\|^2 + \epsilon_k\\ &> -\gamma_1\|\alpha_k^-\sigma_k g_k\|^2 - \gamma_2\|\alpha_k^-\sigma_k d_k\|^2.\end{aligned}$$

By using (3.2) and (3.6), we have

$$\|g(x_k+\alpha_k^+\sigma_k d_k)\|^2 - \|g_k\|^2 > -C(\alpha_k^+)^2 \tag{3.15}$$

Table 4.1
Numerical results

Problem(dim)	Method	Iter	Nfunc	Time	Dim	Iter	Nfunc	Time
expo1	dfsane	5	5	0.0000	expo1	2	2	0.0000
(1000)	dfsdcg1	4	8	0.0000	(10000)	1	2	0.0000
	dfsdcg2	3	6	0.0000		1	2	0.0000
	dfsdcg3	4	8	0.0000		1	2	0.0000
lin1	dfsane	1	2	0.0000	lin1	1	2	0.0156
(1000)	dfsdcg1	1	2	0.0000	(10000)	1	2	0.0156
	dfsdcg2	1	2	0.0000		1	2	0.0156
	dfsdcg3	1	2	0.0000		1	2	0.0156
expo3	dfsane	13	18	0.0000	expo3	16	22	0.0156
(100)	dfsdcg1	15	34	0.0000	(1000)	33	86	0.0313
	dfsdcg2	33	77	0.0156		52	142	0.0468
	dfsdcg3	35	94	0.0156		22	62	0.0156
fdtvf	dfsane	183	726	0.0000	fdtvf	150	544	0.0000
(99)	dfsdcg1	513	2877	1.6876	(200)	279	791	0.0313
	dfsdcg2	215	777	0.0156		250	1155	0.0798
	dfsdcg3	147	451	0.0000		–	–	–
fukushima	dfsane	42	68	0.0000	fukushima	524	2062	0.0156
(9)	dfsdcg1	5	10	0.0000	(49)	22	44	0.0000
	dfsdcg2	5	10	0.0000		22	44	0.0000
	dfsdcg3	5	10	0.0000		22	44	0.0000
rosen	dfsane	3	3	0.0000	rosen	3	3	0.0156
(100)	dfsdcg1	1	2	0.0000	(10000)	1	2	0.0000
	dfsdcg2	1	2	0.0000		1	2	0.0000
	dfsdcg3	1	2	0.0000		1	2	0.0000
arosen	dfsane	1	1	0.0000	arosen	1	1	0.0000
(1000)	dfsdcg1	2	4	0.0000	(10000)	2	4	0.0000
	dfsdcg2	2	4	0.0000		2	4	0.0000
	dfsdcg3	2	4	0.0000		2	4	0.0000
chandra	dfsane	6	6	0.0000	chandra	6	6	7.0156
(100)	dfsdcg1	6	12	0.0000	(5000)	8	16	19.2500
	dfsdcg2	2	4	0.0000		4	8	9.5938
	dfsdcg3	5	10	0.0000		7	14	17.1560
powell	dfsane	2	12	0.0000	powell	2	12	0.0313
(1000)	dfsdcg1	1	2	0.0000	(10000)	1	2	0.0156
	dfsdcg2	1	2	0.0000		1	2	0.0156
	dfsdcg3	1	2	0.0000		1	2	0.0156
powella	dfsane	17	49	0.0313	powella	17	49	0.0625
(1000)	dfsdcg1	6	16	0.0000	(5000)	6	16	0.0156
	dfsdcg2	6	17	0.0156		6	18	0.0313
	dfsdcg3	6	16	0.0000		7	19	0.0313

and

$$\|g(x_k - \alpha_k^- \sigma_k d_k)\|^2 - \|g_k\|^2 > -C(\alpha_k^-)^2, \tag{3.16}$$

where $C = (\gamma_1 \gamma^2 + \gamma_2 M^2)\sigma_{\max}^2$. From (3.15), we obtain

$$\frac{\|g(x_k + \alpha_k^+ \sigma_k d_k)\|^2 - \|g_k\|^2}{\alpha_k^+} > -C\alpha_k^+. \tag{3.17}$$

By the mean-value theorem and (3.17), there exists a $\xi_k \in (0, 1)$ such that

$$\sigma_k \langle 2J(x_k + \xi_k \alpha_k^+ \sigma_k d_k)^T g(x_k + \xi_k \alpha_k^+ \sigma_k d_k), d_k\rangle > -C\alpha_k^+.$$

By Step 2 of the algorithm we have that $\sigma_k > 0$ for infinitely many indices or $\sigma_k < 0$ for infinitely many indices. If $\sigma_k > 0$ for many indices $k \in K_3 \subset K_1$, the last inequality implies, for $k \in K_3$ with $k \geq k_0$,

$$\langle 2J(x_k + \xi_k \alpha_k^+ \sigma_k d_k)^T g(x_k + \xi_k \alpha_k^+ \sigma_k d_k), g(x_k) + \lambda_k \theta_k g(x_k) - (1 - \lambda_k)\eta_k y_{k-1} - \beta_k^{PRP} d_{k-1}\rangle < C\frac{\alpha_k^+}{\sigma_{\min}}. \tag{3.18}$$

Using (3.3), (3.8) and (3.9), taking limits in (3.18), we obtain

$$g(x^*)^T J(x^*) g(x^*) \leq 0.$$

Table 4.2
Numerical results

Problem(dim)	Method	Iter	Nfunc	Time	Dim	Iter	Nfunc	Time
trig	dfsane	30	62	0.0469	trig	14	32	0.0625
(1000)	dfsdcg1	7	18	0.0000	(5000)	6	12	0.0313
	dfsdcg2	5	10	0.0000		5	18	0.0469
	dfsdcg3	6	12	0.0000		2	6	0.0313
trigs	dfsane	4	8	0.0000	trigs	6	12	0.0156
(1000)	dfsdcg1	5	14	0.0000	(5000)	5	15	0.0156
	dfsdcg2	5	14	0.0000		5	15	0.0313
	dfsdcg3	5	14	0.0000		5	15	0.0313
sing	dfsane	11	17	0.0000	sing	12	20	0.0313
(1000)	dfsdcg1	11	22	0.0000	(10000)	11	22	0.0469
	dfsdcg2	9	18	0.0000		9	18	0.0313
	dfsdcg2	8	16	0.0000		8	16	0.0156
loga	dfsane	5	5	0.0000	loga	5	5	0.0156
(1000)	dfsdcg1	4	8	0.0000	(10000)	4	8	0.0313
	dfsdcg2	4	8	0.0000		4	8	0.0156
	dfsdcg3	4	8	0.0000		4	8	0.0156
broydt	dfsane	14	16	0.0000	broydt	17	17	0.0000
(500)	dfsdcg1	14	28	0.0000	(5000)	15	30	0.0313
	dfsdcg2	14	28	0.0000		15	30	0.0313
	dfsdcg3	14	28	0.0000		15	30	0.0313
trigexp	dfsane	9	11	0.0000	trigexp	7	9	0.0000
(100)	dfsdcg1	9	24	0.0000	(10000)	11	26	0.0156
	dfsdcg2	9	24	0.0000		7	18	0.0000
	dfsdcg3	9	24	0.0000		9	22	0.0000
fun15	dfsane	–	–	–	fun15	–	–	–
(1000)	dfsdcg1	–	–	–	(5000)	–	–	–
	dfsdcg2	–	–	–		–	–	–
	dfsdcg3	–	–	–		–	–	–
econvex1	dfsane	5	5	0.0000	econvex1	5	5	0.0156
(100)	dfsdcg1	6	12	0.0000	(10000)	6	12	0.0469
	dfsdcg2	4	8	0.0000		4	8	0.0313
	dfsdcg3	5	10	0.0000		5	10	0.0313
econvex2	dfsane	40	42	0.0156	econvex2	68	132	0.2188
(1000)	dfsdcg1	47	94	0.0313	(5000)	48	96	0.1719
	dfsdcg2	49	98	0.0313		50	100	0.1719
	dfsdcg3	47	94	0.0313		48	96	0.1456
fun18	dfsane	5	7	0.0000	fun18	5	7	0.0156
(399)	dfsdcg1	3	6	0.0000	(9000)	3	6	0.0156
	dfsdcg2	2	4	0.0000		3	6	0.0156
	dfsdcg3	2	4	0.0000		3	6	0.0156

Using (3.16) and proceeding in the same way, we obtain

$$g(x^*)^T J(x^*) g(x^*) \geq 0.$$

The last two inequalities imply (3.13). If $\sigma_k < 0$ for infinitely many indices, proceeding in an analogous way, we also deduce (3.13). □

4. Numerical experiments

In this section, we tested DF-SDCG and compared it with DF-SANE in [9]. We tested 20 nonlinear monotone equations that were described in [8]. The DF-SDCG code was written in Fortran77 and in double precision arithmetic. The DF-SANE code was provided by Professor Raydan. The programs were carried out on a PC (`CPU 1.6GHz, 256M memory`) with the Windows operation system.

We implemented DF-SDCG with the following parameters: $M = 1$, $\rho_{\min} = 0.1$, $\rho_{\max} = 0.5$, $\sigma_{\min} = 10^{-10}$, $\sigma_{\max} = 10^{10}$, $\gamma_1 = \gamma_2 = 10^{-4}$ and $\epsilon_k = \frac{\|g(x_0)\|}{(1+k)^2}$ for all k. We choose $\alpha_{+\text{new}}$ and α_{-new} in the same way as those in [9]. We choose σ_k in the same way as that in [25]. To be more precise, the initial steplength in Step 2 of DF-SDCG is $\sigma_k = \frac{-g_k^T d_k}{d_k^T z_k}$, where

$$z_k = \frac{g(x_k + \epsilon d_k) - g(x_k)}{\epsilon}, \quad \epsilon = 10^{-8}.$$

Table 4.3
Number of problems for which each method is a winner

Method	Iter	Nfunc	Time
dfsane	12	21	29
dfsdcg1	19	13	27
dfsdcg2	26	13	28
dfsdcg3	23	17	30

If $|\sigma_k| \notin [\sigma_{\min}, \sigma_{\max}]$, we set $\sigma_k = 1$. In both DF-SANE and DF-SDCG we stop the process when the following inequality is satisfied

$$\frac{\|g(x_k)\|}{\sqrt{n}} \leq e_a + e_r \frac{\|g(x_0)\|}{\sqrt{n}},$$

where $e_a = 10^{-5}$ and $e_r = 10^{-4}$. For each test problem, we perform the following four algorithms:

- `dfsane`: DF-SANE in [9];
- `dfsdcg1`: DF-SDCG with $\lambda_k = 1$;
- `dfsdcg2`: DF-SDCG with $\lambda_k = 0$;
- `dfsdcg3`: DF-SDCG with $\lambda_k = 0.5$.

We implemented DF-SANE with the following parameters: $nexp = 2$, $\sigma_{\min} = 10^{-10}$, $\sigma_{\max} = 10^{10}$, $\sigma_0 = 1$, $\tau_{\min} = 0.1$, $\tau_{\max} = 0.5$, $\gamma = 10^{-4}$, $M = 10$, $\eta_k = \frac{\|g(x_0)\|}{(1+k)^2}$ for all k. In Tables 4.1 and 4.2, we report the name of the problem (problem), the dimension of the problem (dim), the number of iterations (iter), the number of function evaluations (including the additional functional evaluations that DF-SDCG uses for approximating initial steplength σ_k) (Nfunc) and the CPU time in seconds (time). We claim that the method fails, and use the symbol '-', when some of the following options hold:

(a) the number of iterations is greater than or equal to 1000; or
(b) the number of backtracking at some line search is greater than or equal to 50.

In addition, the results from Tables 4.1 and 4.2 are summarized in Table 4.3. In Table 4.3 we report the number of problems for which each method is a winner with respect to the number of iterations, number of function evaluations and CPU time.

From Table 4.3, we observe that DF-SDCG requires less iterations and more function evaluations than DF-SANE. We also observe from Tables 4.1 and 4.2 that in most cases the number of function evaluations is twice the number of iterations. This implies that the initial steplength σ_k has the advantage of being accepted often, but has the disadvantage of needing an additional function evaluation at each iteration. This leads to more function evaluations for DF-SDCG. However, as far as the CPU time is concerned, DF-SDCG has almost the same performance as DF-SANE, which is important for solving large-scale problems. To sum up, the results from Tables 4.1–4.3 show that DF-SDCG provides an efficient method for solving large-scale nonlinear systems of equations. Looking for a proper steplength to improve the efficiency of DF-SDCG will be a future topic for us.

Acknowledgments

We thank Professor Dong-Hui Li for his careful reading of the manuscript and his suggestions and comments on the paper. We are indebted to an anonymous referee for his helpful suggestions which improved the quality of this paper. We are also very grateful to Professor Raydan for providing us the test problems and the DF-SANE codes.

References

[1] J. Barzilai, J.M. Borwein, Two-point step size gradient methods, IMA J. Numer. Anal. 8 (1988) 141–148.
[2] E.G. Birgin, N.K. Krejic, J.M. Martínez, Globally convergent inexact quasi-Newton methods for solving nonliear systems, Numer. Algorithms. 32 (2003) 249–260.
[3] S. Bellavia, B. Morini, A globally convergent Newton-GMRES subspace method for systems of nonlinear equations, SIAM J. Sci. Comput. 23 (2001) 940–960.
[4] P.N. Brown, Y. Saad, Convergence theory of nonlinear Newton-Krylov algorithms, SIAM J. Optim. 4 (1994) 297–330.
[5] C.G. Broyden, A class of methods for solving nonlinear simultaneous equations, Math. Comput. 19 (1965) 577–593.
[6] W.Y. Cheng, A two term PRP based descent Method, Numer. Funct. Anal. Optim. 28 (2007) 1217–1230.
[7] W.La Cruz, M. Raydan, Nonmonotone spectral methods for large-scale nonlinear systems, Optim. Methods Softw. 18 (2003) 583–599.
[8] W.La Cruz, J.M. Martínez, M. Raydan, Spectral residual method without gradient information for solving large-scale nonlinear systems: Theory and experiments, Technical Report RT-04-08, Dpto. de Computation, UCV, 2004.
[9] W.La Cruz, J.M. Martínez, M. Raydan, Spectral residual method without gradient information for solving large-scale nonlinear systems of equations, Math. Comput. 75 (2006) 1429–1448.
[10] J.E. Dennis Jr., R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.
[11] M. Gasparo, A nonmonotone hybrid method for nonlinear systems, Optim. Methods Softw. 13 (2000) 79–94.
[12] L. Grippo, F. Lampariello, S. Lucidi, A nonmonotone line search technique for Newton's method, SIAM J. Numer. Anal. 23 (1986) 707–716.
[13] A. Griewank, The 'global' convergence of Broyden-like methods with suitable line search, J. Austral. Math. Soc. Ser. B 28 (1986) 75–92.
[14] W.W. Hager, H. Zhang, A survey of nonlinear conjugate gradient methods, Pacific J. Optim. 2 (2006) 35–58.

[15] W.W. Hager, H. Zhang, A new conjugate gradient method with guaranteed descent and an efficent line search, SIAM J. Optim. 16 (2005) 170–192.
[16] D.H. Li, M. Fukushima, A derivative-free line search and global convergence of Broyden-like method for nonlinear equations, Optim. Methods Softw. 13 (2000) 583–599.
[17] D.H. Li, M. Fukushima, A globally and superlinearly convergent Gauss-Newton based BFGS methods for symmetric nonlinear equations, SIAM J. Numer. Anal. 37 (1999) 152–172.
[18] D.H. Li, W.Y. Cheng, Recent progress in the global convergence of quasi-Newton methods for nonlinear equations, Hokkaido Math. J. 36 (2007) 729–743.
[19] J.M. Martínez, A family of quasi-Newton methods for nonlinear equations with direct secant updates of matrix factorizations, SIAM J. Numer. Anal. 27 (1990) 1034–1049.
[20] J.M. Martínez, Practical quasi-Newton methods for solving nonlinear systems, J. Comput. Appl. Math. 124 (2000) 97–121.
[21] E. Polak, G. Ribière, Note surla convergence des mèthodes de diretions conjuguèes, Rev Francaise Imformat Recherche Opertionelle. 16 (1969) 35–43.
[22] B.T. Polak, The conjugate gradient method in extreme problems, USSR Comp. Math. Math. Phys. 9 (1969) 94–112.
[23] M.V. Solodov, B.F. Svaiter, A globally convergent inexact Newton method for systems of monotone equations, in: M. Fukushima, L. Qi (Eds.), Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, Kluwer Academic Publishers, 1999, pp. 355–369.
[24] L. Zhang, W.J. Zhou, Spectral gradient projection method for solving nonlinear monotone equations, J. Comput. Appl. Math. 196 (2006) 478–484.
[25] L. Zhang, W.J. Zhou, D.H. Li, A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence, IMA J. Numer. Anal. 26 (2006) 629–640.