SCIENCE CHINA
Mathematics

• ARTICLES •

November 2010 Vol. 53 No. 11: 2907–2915
doi: 10.1007/s11425-010-4056-x

Sparse two-sided rank-one updates for nonlinear equations

CHENG MingHou & DAI YuHong*

State Key Laboratory of Scientific and Engineering Computing,
Institute of Computational Mathematics and Scientific/Engineering Computing,
Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190*, China*
Email: chengmh@lsec.cc.ac.cn, dyh@lsec.cc.ac.cn

Received January 20, 2009; accepted January 14, 2010

Abstract The two-sided rank-one (TR1) update method was introduced by Griewank and Walther (2002) for solving nonlinear equations. It generates dense approximations of the Jacobian and thus is not applicable to large-scale sparse problems. To overcome this difficulty, we propose sparse extensions of the TR1 update and give some convergence analysis. The numerical experiments show that some of our extensions are superior to the TR1 update method. Some convergence analysis is also presented.

Keywords TR1 update, Broyden rank-one update, sparsity

MSC(2000): 47J05, 49M99

Citation: Cheng M H, Dai Y H. Sparse two-sided rank-one updates for nonlinear equations. Sci China Math, 2010, 53(11): 2907–2915, doi: 10.1007/s11425-010-4056-x

1 Introduction

Extensive analysis, both computational and theoretical, has verified that quasi-Newton method is highly successful for the solution of nonlinear equations and the minimization of nonlinear functions. However, as expertise in solving these problems has grown, the problem size, determined by the number of variables, has also grown, leading to storage problems on digital computers. Fortunately, a large-scale problem usually has known sparsity. It then remains to study how to extend some known quasi-Newton methods by making use of the sparsity.

For the problem of solving a system of n nonlinear equations in n real variables

$$F(x) = 0, \tag{1}$$

Griewank and Walther [4] proposed the two-sided rank-one (TR1) update. It is the generalization of Broyden's symmetric rank-one (SR1) update [1]. The q-superlinear convergence of the method is proved in [7]. The TR1 method is to obtain d_k by solving $B_k d_k = -F(x_k)$ and to update $x_{k+1} = x_k + \alpha_k d_k$, in which α_k is computed by some line search. The approximation B_k to the Jacobian is updated at each step in accordance with

$$B_{k+1} = B_k + \frac{(F'(x_{k+1}) - B_k)s_k\sigma_k^{\mathrm{T}}(F'(x_{k+1}) - B_k)}{\sigma_k^{\mathrm{T}}(F'(x_{k+1}) - B_k)s_k}, \tag{2}$$

*Corresponding author

where s_k equals $x_{k+1} - x_k$, and $F'(x_{k+1})$ is the Jacobian at the point x_{k+1}. The parameter σ_k can be chosen to be $(F'(x_{k+1}) - B_k)s_k$, $F(x_{k+1})$ or $(F(x_{k+1}) - F(x_k))/\alpha_k - B_k s_k$ for a sequence $\alpha_k \subset (0, 1]$ with $\lim_{i\to\infty} \alpha_k = 1$ (see [7,8]). The factor α_k is the step length at the k-th iteration. In (2), it does not need to compute $F'(x_{k+1})$. The forward and reverse modes of automatic differentiation (AD) provide the possibility to compute $F'(x)u$ and $v^{\mathrm{T}}F'(x)$ exactly within machine accuracy for given vectors x, u and v. The updating formula (2) fulfills the direct tangent condition

$$B_{k+1}s_k = F'(x_{k+1})s_k \tag{3}$$

and the adjoint tangent condition

$$\sigma_k^{\mathrm{T}} B_{k+1} = \sigma_k^{\mathrm{T}} F'(x_{k+1}). \tag{4}$$

This is the reason why it is called as the two-sided rank-one update.

As we can see, the second term in the right-hand side of (2) is the outer product of the column vector $(F'(x_{k+1}) - B_k)s_k$ and the row vector $\sigma_k^{\mathrm{T}}(F'(x_{k+1}) - B_k)$. Since the components of the vectors are generally not zeros, we see from (2) that the TR1 update generates a dense approximation of the Jacobian. However, in many large nonlinear systems, particularly those difference equations arising from nonlinear differential equations, most elements of the Jacobian are known to be zeros. For example, if B_k has a band structure, we wish it can be updated with the same sparsity. Then it needs much less storage and d_k needs fewer computations. Therefore the sparse extension of (2) will be interesting. This motivates the study of this paper.

This paper is organized as follows. Section 2 describes some sparse extensions of the TR1 update. Convergence analysis for a special case is given in Section 3. In Section 4, we present some numerical results. Finally, we give some remarks in Section 5.

2 Sparse extensions of two-sided rank-one update

We consider problem (1) whose Jacobian is sparse. We adopt the same notations as in [11] to describe the sparsity of the Jacobian. Denote

$$V_i := \{v \in R^n :\ e_j^{\mathrm{T}} v = 0,\ \forall j,\ \text{such that } e_i^{\mathrm{T}} F'(x) e_j = 0 \text{ for all } x\},$$

where e_i is the unit vector whose i-th element is one and the others are zeros. It is easy to see that V_i is the set of vectors that have the same sparsity as the i-th row of $F'(x)$. For any vector $s \in R^n$, the orthogonal projection operator S^{V_i} onto the subspace V_i is defined by

$$S^{V_i}(s) := \begin{cases} s_j, & \text{if} \quad v_j \neq 0, \\ 0, & \text{if} \quad v_j = 0. \end{cases}$$

Denote the sparsity of the Jacobian as

$$SP(F') := \{M \in R^{n\times n} : M^{\mathrm{T}} e_i \in V_i, 1 \leqslant i \leqslant n\}.$$

Our problem is

$$\begin{aligned} \min_{B\in R^{n\times n}} \quad & \|B - B_k\|_F \\ \text{s.t.} \quad & Bs_k = F'(x_{k+1})s_k, \\ & \sigma_k^{\mathrm{T}} B = \sigma_k^{\mathrm{T}} F'(x_{k+1}), \\ & B \in SP(F'). \end{aligned} \tag{5}$$

There are some difficulties in solving (5) because of the existence of the two linear constraints. For this reason, we consider (5) with only one linear constraint at the first stage. To do so, we list the following lemma given by Walter [11]. Lemma 2.1 is one basis to obtain the sparse extensions of the TR1 update.

Lemma 2.1. *Let $c \in R$ and $v \in V_i \backslash \{0\}$. Then the unique solution to*

$$\begin{aligned} \min_{x \in R^n} \quad & \|x\|_2 \\ \text{s.t.} \quad & v^{\mathrm{T}} x = c \end{aligned} \tag{6}$$

is $x = cv / \sum_{i \in I} v_i^2$, where $I \in \{i : 1 \leqslant i \leqslant n, v_i \neq 0\}$.

By Lemma 2.1, we can obtain one sparse extension of the TR1 update, in which only the first linear constraint of (5) is considered.

Theorem 2.2. *Given $B_k \in SP(F')$, the unique solution to*

$$\begin{aligned} \min_{B \in R^{n \times n}} \quad & \|B - B_k\|_F \\ \text{s.t.} \quad & B s_k = F'(x_{k+1}) s_k, \\ & B \in SP(F') \end{aligned} \tag{7}$$

is

$$B_{k+1} = B_k + \sum_{i=1}^{n} e_i e_i^{\mathrm{T}} \frac{(F'(x_{k+1}) - B_k) s_k S^{V_i}(s_k^{\mathrm{T}})}{S^{V_i}(s_k^{\mathrm{T}}) S^{V_i}(s_k)}. \tag{8}$$

Proof. According to the definition of F-norm, we have

$$\|B - B_k\|_F^2 = \sum_{i=1}^{n} \|e_i^{\mathrm{T}}(B - B_k)\|_2^2.$$

Hence the problem (7) can be derived into n subproblems,

$$\begin{aligned} \min_{B \in R^{n \times n}} \quad & \|e_i^{\mathrm{T}}(B - B_k)\|_2 \\ \text{s.t.} \quad & e_i^{\mathrm{T}} B s_k = e_i^{\mathrm{T}} F'(x_{k+1}) s_k, \\ & e_i^{\mathrm{T}} B \in V_i \end{aligned}$$

for $i = 1, 2, \ldots, n$, and each subproblem is of the form (6). Applying Lemma 2.1 to get the solution of each subproblem and summing them together, we know the solution of (7) is exactly (8).

Each term of the series in (8) is a matrix with only one nonzero row. The matrix B_{k+1} can be updated from B_k row by row. Each row has the same sparsity as that of the Jacobian. Consequently, B_{k+1} inherits the sparsity structure of the whole Jacobian.

Similarly to Theorem 2.2, we can obtain the corresponding sparse variant of the TR1 method if only the second linear constraint of (5) is concerned.

Theorem 2.3. *Given $B_k \in SP(F')$, the unique solution to*

$$\begin{aligned} \min_{B \in R^{n \times n}} \quad & \|B - B_k\|_F \\ \text{s.t.} \quad & \sigma_k^{\mathrm{T}} B = \sigma_k^{\mathrm{T}} F'(x_{k+1}), \\ & B \in SP(F') \end{aligned}$$

is

$$B_{k+1} = B_k + \sum_{i=1}^{n} \frac{S^{\bar{V}_i}(\sigma_k) \sigma_k^{\mathrm{T}} (F'(x_{k+1}) - B_k)}{S^{\bar{V}_i}(\sigma_k^{\mathrm{T}}) S^{\bar{V}_i}(\sigma_k)} e_i e_i^{\mathrm{T}}, \tag{9}$$

where $\bar{V}_i := \{v \in R^n : e_j^{\mathrm{T}} v = 0, \ \forall j, \text{ such that } e_j^{\mathrm{T}} F'(x) e_i = 0, \text{ for all } x\}$ describes the sparsity of the i-th column of the Jacobian.

Proof. It suffices to rewrite the problem with respect to B_{k+1}^{T} and B_k^{T}. Then we can deduce (9) similarly to the proof of Theorem 2.2.

In formula (9), each term of the series is a matrix with only one nonzero column. The matrix B_{k+1} can be updated from B_k column by column and each column has the same sparsity as that of the Jacobian. As a result, B_{k+1} inherits the sparsity structure of the whole Jacobian.

Further, based on the above analysis of the formulas (8) and (9), we can propose two other sparse modifications of the TR1 update (2). The first sparse formula is to update B_k row by row as (8),

$$B_{k+1} = B_k + \sum_{i=1}^{n} e_i e_i^{\mathrm{T}} \frac{(F'(x_{k+1}) - B_k)s_k S^{V_i}(\tilde{\sigma}_k^{\mathrm{T}})}{S^{V_i}(\tilde{\sigma}_k^{\mathrm{T}})s_k}, \tag{10}$$

where $\tilde{\sigma}_k^{\mathrm{T}} = \sigma_k^{\mathrm{T}}(F'(x_{k+1}) - B_k)$. The second sparse formula is to update B_k column by column as (9),

$$B_{k+1} = B_k + \sum_{i=1}^{n} \frac{S^{\bar{V}_i}(\tilde{s}_k)\sigma_k^{\mathrm{T}}(F'(x_{k+1}) - B_k)}{\sigma_k^{\mathrm{T}} S^{\bar{V}_i}(\tilde{s}_k)} e_i e_i^{\mathrm{T}}, \tag{11}$$

where $\tilde{s}_k = (F'(x_{k+1}) - B_k)s_k$.

Thus we have obtained four sparse formulas. We can see that (8) and (10) satisfy the direct tangent condition (3) and update B_k row by row. The formulas (9) and (11) satisfy the adjoint tangent condition (4) and update B_k column by column. They all generate sparse approximations and therefore are applicable to the solution of the large-scale sparse nonlinear equations as in [9,12]. Furthermore, the formulas (10) and (11) have the least change property for the special choice of σ_k, which enables us to give the convergence analysis that will be done in the next section.

Numerically, the four sparse formulas are easy to compute. When the denominator is zero, we will not update the corresponding row or column. However, none of them is the solution of (5), because they only satisfy (3) or (4). Hence, we are interested in some sparse update that satisfies (3) and (4) simultaneously.

Our idea is to solve (5) via Lagrangian function. Denoting $E = B_{k+1} - B_k$, $a = (F'(x_{k+1}) - B_k)s_k$, $b = (F'(x_{k+1}) - B_k)^{\mathrm{T}}\sigma_k$, $s = s_k$ and $\sigma = \sigma_k$, we rewrite (5) as follows,

$$\begin{aligned} \min_{E\in R^{n\times n}} \quad & \|E\|_F \\ \text{s.t.} \quad & \sum_{j=1}^{n} E_{ij} S^{V_i}(s)_j = a_i, \\ & \sum_{j=1}^{n} E_{ji} S^{\bar{V}_i}(\sigma)_j = b_i, \qquad \text{for } \ i = 1, 2, \ldots, n, \end{aligned} \tag{12}$$

where S^{V_i} and $S^{\bar{V}_i}$ are defined before. The Lagrangian function associated with (12) is given by

$$L(E, \lambda, \bar{\lambda}) = \frac{1}{2}\|E\|_F^2 - \sum_{i=1}^{n} \lambda_i \left(\sum_{j=1}^{n} E_{ij} S^{V_i}(s)_j - a_i \right) - \sum_{i=1}^{n} \bar{\lambda}_i \left(\sum_{j=1}^{n} E_{ji} S^{\bar{V}_i}(\sigma)_j - b_i \right).$$

Differentiation with respect to E_{ij} shows that the multipliers λ and $\bar{\lambda}$ must satisfy the equation

$$\frac{\partial L(E, \lambda, \bar{\lambda})}{\partial E_{ij}} = E_{ij} - \lambda_i S^{V_i}(s)_j - \bar{\lambda}_j S^{\bar{V}_j}(\sigma)_i = 0 \tag{13}$$

for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, n$. Substituting (13) into the linear constraints of (12) yields

$$\begin{cases} \displaystyle\sum_{j=1}^{n} (\lambda_i S^{V_i}(s)_j + \bar{\lambda}_j S^{\bar{V}_j}(\sigma)_i) S^{V_i}(s)_j = a_i, \\ \displaystyle\sum_{j=1}^{n} (\lambda_j S^{V_j}(s)_i + \bar{\lambda}_i S^{\bar{V}_i}(\sigma)_j)^{\mathrm{T}} S^{\bar{V}_i}(\sigma)_j = b_i, \end{cases} \tag{14}$$

for $i = 1, 2, \ldots, n$. This is a linear system of λ and $\bar{\lambda}$ and can be written as

$$\begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix} \begin{pmatrix} \lambda \\ \bar{\lambda} \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}, \tag{15}$$

where $A_1 = \mathrm{diag}[S^{V_1}(s)^{\mathrm{T}}s, \ldots, S^{V_n}(s)^{\mathrm{T}}s]$, $A_2 = (S^{\bar{V}_j}(\sigma)_i S^{V_i}(s)_j)_{n\times n}$, $A_3 = A_2^{\mathrm{T}}$, $A_4 = \mathrm{diag}[S^{\bar{V}_1}(\sigma)^{\mathrm{T}}\sigma, \ldots, S^{\bar{V}_n}(\sigma)^{\mathrm{T}}\sigma]$. Therefore, to solve (5), we can first compute the solutions λ and $\bar{\lambda}$ of (15). Substituting them into (13), we can get

$$E = \sum_{i=1}^{n} (\lambda_i e_i S^{V_i}(s)^{\mathrm{T}} + \bar{\lambda}_i S^{\bar{V}_i}(\sigma) e_i^{\mathrm{T}}). \tag{16}$$

Then $B_{k+1} = B_k + E$ is obtained. Unfortunately, the coefficient matrix of (15) is rank deficient and its rank is less than or equal to $2n-1$. To circumvent this difficulty, we solve a sparse linear least squares problem at each step which differs from the symmetric case in [10, 12]. In real computations, we solve the least squares problem by the Bi-CG method (see [3, 5]).

3 Convergence analysis

In this section, we give some convergence analysis of the sparse TR1 updates (11). For this purpose, we consider the choice $\sigma_k = (F'(x_{k+1}) - B_k)s_k$. As in [4], the formula (2) reduces to the adjoint tangent rank-one update

$$B_{k+1} = B_k + \frac{\sigma_k \sigma_k^{\mathrm{T}}(F'(x_{k+1}) - B_k)}{\sigma_k^{\mathrm{T}}\sigma_k}. \tag{17}$$

The formula (17) satisfies (4) and the following least change condition

$$\|B_{k+1} - B_k\|_F \leqslant \left\| \frac{\sigma_k \sigma_k^{\mathrm{T}}}{\sigma_k^{\mathrm{T}}\sigma_k} \right\|_2 \|B - B_k\|_F = \|B - B_k\|_F.$$

for any B satisfying $\sigma_k^{\mathrm{T}} B = \sigma_k^{\mathrm{T}} F'(x_{k+1})$. In this case, (11) satisfies the least change condition. Based on the convergence analysis of least change secant methods presented by Dennis and Walker in [2], we can get similar conclusions of our updating formula (11). The main difference is that to obtain B_{k+1} we make use of the adjoint tangent condition instead of the secant condition. We firstly analyze the properties of B_{k+1} and get the bounded deterioration inequality.

B_{k+1} is required to lie in the intersection of the two affine subspaces $SP(F')$ and $\mathcal{L}(u_k, \sigma_k) = \{M \in R^{n\times n} : M^{\mathrm{T}}\sigma_k = u_k\}$, where $u_k = F'(x_{k+1})^{\mathrm{T}}\sigma_k$. Denote $\mathcal{A} = SP(F')$ and $\mathcal{L} = \mathcal{L}(u_k, \sigma_k)$. The new sparse approximation B_{k+1} is the unique solution of

$$\min_{B\in\mathcal{M}(\mathcal{A},\mathcal{L})} \|B - B_k\|_F,$$

as described in [2]. $\mathcal{M}(\mathcal{A}_1, \mathcal{A}_2)$ is the set of elements of $\mathcal{A}_1$ for which the distance to $\mathcal{A}_2$ in F-norm is minimal for any affine subspaces $\mathcal{A}_1$, $\mathcal{A}_2 \subseteq R^{n\times n}$. We also note that $\mathcal{L}$ can be written as

$$\mathcal{L} = \left\{ P_{\mathcal{N}}^{\perp}\left(\frac{u_k\sigma_k^{\mathrm{T}}}{\sigma_k^{\mathrm{T}}\sigma_k}\right) + M : M \in \mathcal{N} \right\},$$

where $\mathcal{N} = \mathcal{N}(\sigma_k) = \{M \in R^{n\times n} : M^{\mathrm{T}}\sigma_k = 0\}$ is the subspace of annihilators of σ_k. Unless indicated otherwise, the projection which maps onto a given affine subspace is denoted by "P", with the subspace or affine subspace indicated as a subscript. The projection orthogonal to this projection is indicated by "$P^{\perp}$". Similarly, one can write $\mathcal{A} = \{B_N + M : M \in \mathcal{S}\}$, where the "normal" $B_N \in \mathcal{S}^{\perp}$.

The following theorems can be proven like the corresponding theorems in [2]. So we pass most of the proofs over. In Theorem 3.1, we list some properties of the elements of $\mathcal{M}(\mathcal{A}, \mathcal{L})$ like Theorem 2.3 in [2].

Theorem 3.1. *Given that vectors σ_k,$u_k \in R^n$ with $\sigma_k \neq 0$, an affine subspace $\mathcal{A} \subseteq R^{n\times n}$. Set $P_{\mathcal{M}(\mathcal{A},\mathcal{L})}B = B_+$ for $B \in R^{n\times n}$. Then $\mathcal{M}(\mathcal{A}, \mathcal{L})$ is an affine subspace of $R^{n\times n}$ with parallel subspace $\mathcal{S}\cap\mathcal{N}$; in particular*

$$\mathcal{M}(\mathcal{A}, \mathcal{L}) = \left\{ (I - P_{\mathcal{S}}P_{\mathcal{N}})^{-1}B_N + (I - P_{\mathcal{S}}P_{\mathcal{N}})^{-1}P_{\mathcal{S}}P_{\mathcal{N}}^{\perp}\left(\frac{u_k\sigma_k^{\mathrm{T}}}{\sigma_k^{\mathrm{T}}\sigma_k}\right) + M : M \in \mathcal{S}\cap\mathcal{N} \right\}.$$

If $G,\bar{G} \in \mathcal{M}(\mathcal{A},\mathcal{L})$, *then* $P_{\mathcal{N}}^{\perp}G = P_{\mathcal{N}}^{\perp}\bar{G}$, *i.e.,* $Gs_k = \bar{G}s_k$. *Furthermore, if* $G \in \mathcal{M}(\mathcal{A},\mathcal{L})$, *then* $P_{\mathcal{S}}P_{\mathcal{N}}^{\perp}G = P_{\mathcal{S}}P_{\mathcal{N}}^{\perp}(\frac{u_k\sigma_k^{\mathrm{T}}}{\sigma_k^{\mathrm{T}}\sigma_k})$ *and, if* $B \in R^{n\times n}$, *then* $B_+ = P_{\mathcal{S}\cap\mathcal{N}}B + P_{\mathcal{S}}P_{\mathcal{N}}^{\perp}G$. *If* $G \in \mathcal{M}(\mathcal{A},\mathcal{L})$, *then*

$$\|B_+ - M\|_F \leqslant \|P_{\mathcal{S}\cap\mathcal{N}}(B-M)\|_F + \|P_{\mathcal{S}\cap\mathcal{N}}^{\perp}(G-M)\|_F.$$

Proof. According to the definitions of $\mathcal{A}$ and $\mathcal{L}$, we can easily get the similar conclusions as in [2].

Based on Theorem 3.1, we give the convergence analysis of the method considered here in the case of $B_* = F'(x_*)$.

Theorem 3.2. *Let* F *be differentiable in an open convex neighborhood* Ω *of a point* $x_* \in R^n$ *for which* $F(x_*) = 0$, *and* $F'(x_*)$ *be nonsingular, let* $\gamma \geqslant 0$, *and* $p \in (0,1]$, *be such that, for* $x \in \Omega$, $\|F'(x)-F'(x_*)\|_F \leqslant \gamma\|x-x_*\|_2^p$, *and let* $B_* = F'(x_*)$, *so there exists an* r_* *for which* $\|I-B_*^{-1}F'(x_*)\|_F \leqslant r_* < 1$. *Also assume that the choice rule* χ *for* $F'(x_k)^{\mathrm{T}}\sigma_k$ *has the property with* $\mathcal{A}$ *that there exists an* $\alpha \geqslant 0$ *such that for any* x, $x_+ \in \Omega$ *and any* $F'(x_k)^{\mathrm{T}}\sigma_k \in \chi(x,x_+)$, *one has*

$$\|P_{\mathcal{S}\cap\mathcal{N}(s)}^{\perp}(G-B_*)\|_F \leqslant \alpha\psi(x,x_+)^p$$

for $\forall\ G^{\mathrm{T}} \in \mathcal{M}(\mathcal{A},\mathcal{L})$, *where* $\psi(x,x_+) = \max\{\|x-x_*\|_2, \|x_+-x_*\|_2\}$. *Under these hypotheses, if* $r \in (r_*,1)$, *then there are positive constants* ε_r, δ_r *such that for* $x_0 \in R^n$ *and* $B_0 \in R^{n\times n}$ *satisfying* $\|x_0-x_*\|_2 < \varepsilon_r$ *and* $\|B_0-B_*\|_F < \delta_r$, *any sequence* $\{x_k\}$ *defined by* $\{x_{k+1} = x_k - B_k^{-1}F(x_k),\ B_k \in \{B_{k-1},(B_{k-1})_+\}\}$ *satisfies* $\|x_{k+1}-x_*\|_2 \leqslant r\|x_k-x_*\|_2$ *for* $k = 0,1,2,\ldots$, *where* $(B_{k-1})_+$ *is the least-change adjoint secant update of* B_{k-1}. *Furthermore,* $\|B_k^{-1}\|_F$ *and* $\|B_k\|_F$ *are uniformly bounded.*

We now address the superlinear convergence of the method. It is worth pointing out that Theorem 3.3 applies to any sequence $\{x_k\}$ generated by the method and not just a sequence started from a sufficiently good x_0 and B_0 as required by Theorem 3.2.

Theorem 3.3. *Suppose that the hypotheses of Theorem* 3.2 *hold and that for some* $x_0 \in R^n$ *and* $B_0 \in R^{n\times n}$, x_k *is a sequence defined by*

$$x_{k+1} = x_k - B_k^{-1}F(x_k), \quad u_k \in \chi(x_k,x_{k+1}), \quad B_{k+1} = (B_k)_+,$$

which converges q-linearly to x_*, *where* $(B_k)_+$ *is the least-change adjoint tangent update of* B_k. *Set* $e_k = x_k - x_*$ *for* $k = 0,1,2,\ldots$ *Then*

$$\lim_{k\to\infty}\frac{\|e_{k+1}\|_2}{\|e_k\|_2} = 0.$$

4 Framework of algorithm and numerical experiments

In this section, based on the sparse extensions of the two-sided rank-one updates, we give the framework of the algorithm for solving nonlinear systems.

Algorithm 4.1. **Step 1** Initialize the starting point x_0, the approximate matrix B_0 of the Jacobian and set $k = 0, \epsilon > 0$; Compute the error $err = \|F(x_0)\|_2$;

Step 2 Test the stop criterion. If $err < \epsilon$ is not satisfied, go to Step 3;

Step 3 Solve $B_kp_k = -F(x_k)$ to get s_k;

Step 4 Implement some line search strategies to get the step length t_k;

Step 5 Compute $x_{k+1} = x_k + t_kp_k$ and the error $err = \|F(x_{k+1})\|_2$;

Step 6 Obtain a new sparse approximation B_{k+1} of the Jacobian; Set $k := k+1$ and go to Step 2.

Different choices for B_{k+1} in Step 6 result in different algorithms. If B_{k+1} is generated by (2), (10), (11) and (8), we call the corresponding algorithm by TR1, Alg.4.1a, Alg.4.1b, Alg.4.1d, respectively. Alg.4.1c corresponds to the choice of B_{k+1} based on (15) and (16). As for the choice of σ_k, we choose $\sigma_k = (F'(x_{k+1}) - B_k)s_k$. Hence, the formula (9) is equal to (11).

In each experiment, we employ the following termination criterion:

$$\|F(x_k)\|_2 \leqslant 10^{-10}$$

Table 1 Results for Problem 1

n	method	N_1	N_m	m	R
30	TR1	2.180596e + 001	4.658708e − 010	16	0.666894
	Alg.4.1a	2.180596e + 001	3.769557e − 010	12	0.896857
	Alg.4.1b	2.180596e + 001	4.977453e − 010	11	0.967415
	Alg.4.1c	2.180596e + 001	7.107384e − 010	12	0.873905
	Alg.4.1d	2.180596e + 001	4.853470e − 010	12	0.887710
300	TR1	6.150610e + 001	6.728408e − 010	16	0.685063
	Alg.4.1a	6.150610e + 001	5.506191e − 010	11	1.004370
	Alg.4.1b	6.150610e + 001	4.867070e − 010	11	1.009241
	Alg.4.1c	6.150610e + 001	6.048830e − 010	12	0.917271
	Alg.4.1d	6.150610e + 001	4.853156e − 010	12	0.925241
3000	TR1	1.919844e + 002	6.784660e − 010	16	0.715734
	Alg.4.1a	1.919844e + 002	5.506191e − 010	11	1.049310
	Alg.4.1b	1.919844e + 002	4.867070e − 010	11	1.054182
	Alg.4.1c	1.919844e + 002	6.385849e − 010	12	0.956504
	Alg.4.1d	1.919844e + 002	4.853156e − 010	12	0.966437

and the backtracking line search strategy to obtain the step length t_k. As in [1, 9], Broyden's mean convergence rate $R = \frac{1}{m}\log_{10}(\frac{N_1}{N_m})$ is computed in each case, where m, N_1 and N_m are the total number of function evaluations, the initial and final Euclidean norms of $F(x)$, respectively. The numerical experiments are done by using Matlab v7.0 on Core(TM)2 PC with Windows-XP.

The testing problems are as follows:

Problem 1. Broyden tridiagonal function [1]:

$$\begin{aligned} F_1(x) &= -(3+\alpha x_1)x_1 + 2x_2 - \beta, \\ F_i(x) &= x_{i-1} - (3+\alpha x_i)x_i + 2x_{i+1} - \beta, \quad i = 2, 3, \ldots, n-1, \\ F_n(x) &= x_{n-1} - (3+\alpha x_n)x_n - \beta. \end{aligned} \tag{18}$$

The parameters were chosen to be $\alpha = -0.5$ and $\beta = 1$.

Problem 2. Trigonometric-exponential system [6]:

$$\begin{aligned} F_1(x) &= 3x_1^2 + 2x_2 - 5 + \sin(x_1 - x_2)\sin(x_1 + x_2), \\ F_i(x) &= 3x_i^2 + 2x_{i+1} - 5 + \sin(x_i - x_{i+1})\sin(x_i + x_{i+1}) + 4x_i \\ &\quad - x_{i-1}\exp(x_i - 1) - x_i - 3, \qquad i = 2, 3, \ldots, n-1, \\ F_n(x) &= 4x_n - x_{n-1}\exp(x_{n-1} - x_n) - 3. \end{aligned} \tag{19}$$

For each problem, we choose three values for the dimension: $n = 30,\ 300,\ 3000$. The initial points are chosen to be $-3 \times \text{ones}(n, 1)$ and $-2 \times \text{ones}(n, 1)$, respectively. The initial Jacobian approximation B_0 can be computed by the automatic differentiation code AD in Matlab.

The numerical results show that the sparse TR1 updates become increasingly desirable as n increases. The four sparse algorithms have better performance than the TR1 algorithm. Furthermore, Alg.4.1a and Alg.4.1b are better than Alg.4.1c, although the sparse formula of Alg.4.1c is required to satisfy both of (3) and (4). In Alg.4.1c, we just choose an approximate solution to (5), because we have to solve the sparse linear least squares problem derived from (15). Hence, it satisfies neither (3) nor (4) in practical computations. The numerical results indicate that Alg.4.1d also has better performance than the TR1 algorithm, but is not better than the formulas (10) and (11).

Table 2 Results for Problem 2

n	Method	N_1	N_m	m	R
30	TR1	1.555635e + 001	7.589990e − 010	10	1.031167
	Alg.4.1a	1.555635e + 001	1.016641e − 010	9	1.242749
	Alg.4.1b	1.555635e + 001	1.718375e − 010	9	1.217421
	Alg.4.1c	1.555635e + 001	5.666206e − 010	9	1.159846
	Alg.4.1d	1.555635e + 001	6.128601e − 010	9	1.156061
300	TR1	3.635932e + 001	1.958495e − 010	11	1.024427
	Alg.4.1a	3.635932e + 001	1.016641e − 010	9	1.283716
	Alg.4.1b	3.635932e + 001	1.718375e − 010	9	1.258389
	Alg.4.1c	3.635932e + 001	5.666206e − 010	9	1.200814
	Alg.4.1d	3.635932e + 001	6.128601e − 010	9	1.197028
3000	TR1	1.101000e + 002	5.316855e − 010	10	1.131613
	Alg.4.1a	1.101000e + 002	1.016641e − 010	9	1.337180
	Alg.4.1b	1.101000e + 002	1.718375e − 010	9	1.311852
	Alg.4.1c	1.101000e + 002	5.666206e − 010	9	1.254277
	Alg.4.1d	1.101000e + 002	6.128601e − 010	9	1.250492

5 Conclusions and discussion

In this paper, we proposed four sparse quasi-Newton formulas of the TR1 update to solve sparse nonlinear equations. We also give an implicit method based on (15) and (16) to obtain the sparse formula of the TR1 update. For some special cases, we made the corresponding convergence analysis. The numerical experiments show that the sparse TR1 updates are superior to the TR1 update when the problem has sparse Jacobian. Hence, the sparse TR1 quasi-Newton updates are applicable to large-scale sparse nonlinear equations, although the update based on (15) and (16) needs to solve the sparse linear equations at each step.

At the end of this section, we give more discussion about the problem (5) from the other two different viewpoints. The first one is that we just try to get a feasible solution of the constraints of (5), which means to solve the following system,

$$\begin{cases} B_{k+1}s_k = F'(x_{k+1})s_k, \\ \sigma_k^{\mathrm{T}} B_{k+1} = \sigma_k^{\mathrm{T}} F'(x_{k+1}), \\ B_{k+1} \in SP(F'). \end{cases} \tag{20}$$

We can rewrite (20) to a system of linear equations with respect to the nonzero elements of B_{k+1}. The size of the coefficient matrix is $2n \times m$ where m is the number of nonzeros in B_{k+1}. If $m > 2n$, its rank is less than or equal to $2n - 1$. We should solve the corresponding linear least squares problem at each step.

The second one is to find such a good combination of the sparse formulas (10) and (11) that it can satisfy both (3) and (4). If the combined formula is

$$B_{k+1} = B_k + \sum_{i=1}^{n} c_i e_i e_i^{\mathrm{T}} \frac{(F'(x_{k+1}) - B_k)s_k S^{V_i}(\tilde{\sigma}_k^{\mathrm{T}})}{S^{V_i}(\tilde{\sigma}_k^{\mathrm{T}})s_k} + \sum_{i=1}^{n} d_i \frac{S^{V_i}(\tilde{s}_k)\sigma_k^{\mathrm{T}}(F'(x_{k+1}) - B_k)}{\sigma_k^{\mathrm{T}} S^{\bar{V}_i}(\tilde{s}_k)} e_i e_i^{\mathrm{T}}, \tag{21}$$

we should compute the coefficients c_i and d_i for $i = 1, 2, \ldots, n$ such that (3) and (4) are satisfied. This problem also equals to a linear system as (15), but the coefficient matrix is

$$A_1 = \mathrm{diag}(a), \quad A_2 = \left[\frac{s_1 b_1 S^{\bar{V}_1}(a)}{\sigma^{\mathrm{T}} S^{\bar{V}_1}(a)}, \ldots, \frac{s_n b_n S^{\bar{V}_n}(a)}{\sigma^{\mathrm{T}} S^{\bar{V}_n}(a)} \right], \quad A_3 = \left[\frac{\sigma_1 a_1 S^{V_1}(b)}{s^{\mathrm{T}} S^{V_1}(b)}, \ldots, \frac{\sigma_1 a_1 S^{V_1}(b)}{s^{\mathrm{T}} S^{V_1}(b)} \right]$$

and

$$A_4 = \mathrm{diag}(b),$$

where a, b, s and σ are defined in Section 2. The coefficient matrix is rank deficient, whose rank is less than or equal to $2n-1$. Hence, we also need to solve a corresponding least squares problem at each step.

For the two methods, we also did some numerical experiments. But the performance is not better than that of the updating method based on (15) and (16). Why are the three implicit methods not so good? There are two possible reasons to explain this. The first one is that the solution of the corresponding linear equations is not chosen to be more beneficial to decrease the value of $\|F(x)\|_2$. The second one is that we do not know what kind of choice of σ_k is better.

As for the choices of σ_k, we proposed two choices. The first choice is $\sigma_k = F(x_{k+1}) - F(x_k)$. Intuitively, it means to update the rows of the quasi-Newton matrix whose corresponding element function values are with large change. The second choice $\sigma_k = F'(x_{k+1})s_k$ is some approximation to the first one. But the performance of the two choices are not satisfactory.

Acknowledgements This work is partly supported by National Natural Science Foundation of China (Grant Nos. 10571171, 10831006) and Chinese Academy of Sciences Knowledge Innovation Grant (Grant No. kjcx-yw-s7-03). The authors cordially thank the referees for their careful reading and helpful comments.

References

1 Broyden C G. A class of methods for solving nonlinear simultaneous equations. Math Comp, 1965, 19: 577–593
2 Dennis Jr J E, Walker H F. Convergence theorems for least change secant update methods. SIAM J Numer Anal, 1981, 18: 949–987
3 Fletcher R. Conjugate gradient methods for indefinite systems. In: Watson G, ed. Proc Dundee Biennial Conf on Numerical Analysis. New York: Springer, 1975
4 Griewank A, Walther A. On constrained optimization by adjoint based quasi-Newton methods. Optim Methods Softw, 2002, 17: 869–889
5 Lanczos C. Solution of systems of linear equations by minimized iteration. J Res Nat Bur Stand, 1952, 49: 33–53
6 Luksan L. Inexact trust region method for large sparse system of nonlinear equations. J Optim Theory Appl, 1994, 81: 569–590
7 Schlenkrich S, Griewank A, Walther A. Local convergence analysis of TR1 updates for solving nonlinear equations. MATHEON, Preprint 337, 2006
8 Schlenkrich S, Walther A, Griewank A. AD-based quasi-Newton methods for the integration of stiff ODEs. Lect Notes Comput Sci Eng, 50. New York: Springer, 2005, 89–98
9 Schubert L K. Modification of a quasi-Newton method for nonlinear equations with a sparse Jacobian. Math Comp, 1970, 24: 27–30
10 Toint Ph L. On sparse and symmetric matrix updating subject to a linear equation. Math Comp, 1997, 31: 954–961
11 Walter A. Improvement of incomplete factorizations by a sparse secant method. Konrad-Zuse-Zentrum Berlin, Preprint SC 90–12, 1990
12 Yuan Y, Sun W. Optimization Theory and Methods. Beijing: Academic Press, 1995