Subspace Methods for Large Scale Nonlinear Equations and Nonlinear Least Squares [*]

Ya-xiang Yuan
State Key Laboratory of Scientific and Engineering Computing,
ICMSEC, AMSS, Chinese Academy of Sciences, Beijing 100190, China
yyx@lsec.cc.ac.cn

Abstract. In this paper, we study large scale nonlinear systems of equations and nonlinear least square problems. We present subspace methods for solving these two special optimization problems. The subspace methods have the characteristic to force the next iteration in a low dimensional subspace. The main technique is to construct subproblems in low dimensions so that the computation cost in each iteration can be reduced comparing to standard approaches. The subspace approach offers a possible way to handle large scale optimization problems which are now attracting more and more attention. Actually, quite a few known techniques can be viewed as subspace methods, such as conjugate gradient method, limited memory quasi-Newton method, projected gradient method, and null space method.

Keywords: Nonlinear Equations, Nonlinear Least Squares, Subspace, Trust Region, Line Search.

1. Introduction

In this paper, we consider two closely related problems of solving nonlinear equations:

$$F_i(x) = 0, \quad i = 1, ..., m, \tag{1}$$

where $F_i(x)$ are differentiable functions in $\Re^n$, and minimizing the least square norm of the vector $F(x) = (F_1(x), F_2(x), ..., F_m(x))^T$, namely

$$\min_{x \in \Re^n} \|F(x)\|_2^2 = \sum_{i=1}^{m} F_i(x)^2. \tag{2}$$

Problems (1) and (2) have various applications in physics, chemistry, biology, engineering, economics, finance, and many other fields. These two special optimization problems have been studied extensively and there are already many methods for solving them, such as Newton's method, inexact Newton's method, Gauss-Newton method, Levenberg-Marquardt method and trust region method, for example see (Ortega and Rheinboldt, 1970; Dennis and Schnale, 1993; Kelly, 1995; Nocedal

[*] this work was partially supported by the Chinese Academy of Sciences and the Chinese National Natural Science Foundation.

and Wright, 1999). In recent years, as many important practical problems involve in great number of variables(say, at the magnitude of millions of variables), large scale optimization problems, including large scale nonlinear equations and large scale nonlinear least squares have been attracting more and more attention from researchers, for example see (Toint, 1987; Gould et al., 2005; Gould and Toint, 2007). Most large scale problems have either sparse property or special structure, therefore special approaches, such as partial separability and structure-exploiting, should be and can be applied to such problems (Mizutani and Demmel, 2003; Bouaricha and Moré, 1997).

We consider another approach, using subspace techniques, to large scale problems. Our motivation is based on the following observation. The general approach for a nonlinear problem is to replace it by either a linear or quadratic or any other simpler problem in each iteration. Normally the simplified model is some kind of approximation to the original model, and both the original and the approximated model are defined for the same set of variables. Thus, for large scale nonlinear problems, most conventional methods will have a linear, or quadratic, or some other simpler problem in the same dimensional space, which means that at each iteration we need to solve a large scale linear (or quadratic or some other simple) problem. But, quite often, large scale linear systems are solved by subspace techniques, which means that an approximate solution for the large scale linear systems is generally an exact solution of a reduced linear system in a lower dimensional subspace, such as the Krylov subspace if the linear system is solved by the truncated conjugate gradient method(Golub and Van Loan, 1996; Steihaug, 1983; Toint, 1981; Yuan, 2000b). Hence, we can see that the standard linearization approach, for large scale nonlinear problems, in fact has two stage approximations: the first stage is to approximate a nonlinear problem by a linear problem, and the second one is to replace the large scale linear problem by a small scale linear problem. In this paper, we propose an approach which tries to avoid this detour. Our short-cut is to construct or define a small scale linear problem in a low dimension subspace directly, and to force such a small linear system approximating the original large scale nonlinear problem as good as possible. A small scale linear system normally can be expressed as a linear system defined in a proper low dimensional subspace. Thus, in each iteration of the methods we will propose in the following, we solve a low dimensional subspace subproblem. Other motivations for considering subspace approach can be found in Yuan (2007).

This paper is organized as follows. In the next section, we will give some simple examples of algorithms that have certain subspace structures. In section 3, a model algorithm using subspace approach for nonlinear equations is given and some possibilities for choices of the subspaces are also discussed. In section 4, subspace techniques for nonlinear least squares are presented.

2. Examples of subspace approaches

In this section, we consider the general approach by subspace techniques. For simplicity, we consider the following unconstrained optimization problem:

$$\min_{x\in\Re^n} f(x) \tag{3}$$

where $f(x)$ is a smooth nonlinear function. The most simple and trivial example of subspace method is the gradient method:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k), \tag{4}$$

as the new iterate point x_{k+1} defined by the gradient method is the solution of the one dimensional problem

$$\min_{|\alpha|\le|\alpha_k|} f(x_k) - \alpha(\nabla f(x_k))^T \nabla f(x_k). \tag{5}$$

The objective function in the above line is just the one dimensional approximation (in the subspace $Span\{\nabla f(x_k)\}$) to the first order Taylor expansion of the original objective function (1).

Another obvious example of subspace method is the conjugate gradient method. Nonlinear conjugate gradient methods define the search direction d_k by

$$d_k = -g_k + \beta_k d_{k-1}, \tag{6}$$

where $g_k = \nabla f(x_k)$, and d_{k-1} is the search direction the previous iteration. Thus, no matter which nonlinear conjugate gradient method one uses, the algorithm will give the next iterate point in the subspace

$$x_k + Span\{g_k, d_{k-1}\}. \tag{7}$$

Therefore, Stoer and Yuan (1995) studied a successive 2-dimensional search algorithm, based on a model subproblem

$$\min_{d\in Span\{-g_k,d_{k-1}\}} \bar{Q}_k(d) \approx f(x_k+d), \tag{8}$$

where $\bar{Q}_k(d)$ is a quadratic function. Problem (8) is to minimize a 2 dimensional quadratic function. Except for the term depending on $g_k^T\nabla^2 f(x_k)g_k$, this quadratic function can be easily defined, as all the other terms are explicitly given or can be estimated by using the relation $\nabla^2 f(x_k)(x_k - x_{k-1}) \approx \nabla f(x_k) - \nabla f(x_{k-1})$. Therefore, in a two dimensional approach the approximate model is fixed except for one number, which is the reduced Hessian along the current gradient direction. This is also true for a higher dimensional subproblem provided the higher dimension subspace model in the $k-$ iteration is defined in a subspace being a subset of the subspace spanned by the steepest descent direction and all the previous steps $x_{i+1} - x_i (i = 1, ..., k-1)$.

The limited memory quasi-Newton method(Liu and Nocedal, 1989) also has the subspace property. In fact, one can show that limited memory quasi-Newton algorithms, no matter with line search or trust region, will always produce a step in the subspace

$$Span\{g_k, s_{k-1}, ..., s_{k-m}, y_{k-1}, ..., y_{k-m}\}, \tag{9}$$

where $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$. Indeed, a trust region algorithm using this subspace is given by Wang et al. (2004). Even for the standard quasi-Newton updates, we can also establish subspace property results. To be more precise, if the initial quasi-Newton matrix is a scale matrix, the standard quasi-Newton method (with line searches or trust regions) will generate the $k-$th iterate point in the subspaces

$$x_1 + Span\{g_1, g_2, ..., g_{k-1}\}. \tag{10}$$

More details can be found in (Gill and Leonard, 2001; Vlček and Lukček, 2002; Wang and Yuan, 2006).

We can give a model subspace algorithm for unconstrained optimization as follows.

ALGORITHM 1. *(A model subspace algorithm for unconstrained optimization)*

Step 1 Given x_1, Define $\mathcal{S}_1$, $\epsilon > 0$, $k := 1$.

Step 2 Solve a subspace subproblem:

$$\min_{d \in \mathcal{S}_k} \bar{Q}_k(d) = g_k^T d + \frac{1}{2} d^T B_k d \tag{11}$$

obtaining d_k. If $\|d_k\| \leq \epsilon$ then stop.

Step 3 Carry out line search to obtain $\alpha_k > 0$, set

$$x_{k+1} = x_k + \alpha_k d_k. \tag{12}$$

Step 4 Generate $\mathcal{S}_{k+1}$ and $\bar{Q}_{k+1}(d)$.

Step 5 $k := k+1$, *Go to Step 2.*

The above algorithm is only a frame for constructing subspace methods. It needs to have additional implementation details. For example, we need to choose a line search technique to ensure the global convergence of the algorithm. How to choose the subspace $\mathcal{S}_{k+1}$ and how to update the quadratic model $\bar{Q}_k(d)$ are also important for the efficiency of the algorithm.

Algorithm 1 is a slight modification of the standard quasi-Newtion algorithm for unconstrained optimization (for example, see (Conn et al., 2000; Fletcher, 1987; Sun and Yuan, 2006)). The main difference between them is the constraint requiring the search direction d_k to be in the subspace $\mathcal{S}_k$. Thus, the essential issue of a subspace algorithm is how to choose the subspace $\mathcal{S}_k$. Yuan (2007) discussed some possible choices, such as $Span\{-g_k, s_{k-1}, ..., s_{k-m}\}$ and $Span\{-g_k, y_{k-1}, ..., y_{k-m}\}$.

3. Subspace techniques for nonlinear equations

Now, we consider subspace techniques for nonlinear equations system (1). Suppose that at the $k-$th iteration, we have the current iterate point x_k and a subspace $\mathcal{S}_k$. Denote the number of dimension of $\mathcal{S}_k$ to be i_k and $q_1^{(k)}, q_2^{(k)}, ..., q_{i_k}^{(k)}$ are a set of linearly independent vectors in $\mathcal{S}_k$. We would like to find the next iterate point x_{k+1} in such a way that the increment $x_{k+1} - x_k$ is in the subspace $\mathcal{S}_k$. Thus, we would like to have

$$F_i(x_k + Q_k z) = 0, \quad i = 1, ..., m, \tag{13}$$

for $z \in \Re^{i_k}$, where

$$Q_k = [\ q_1^{(k)},\ q_2^{(k)},\ ...,\ q_{i_k}^{(k)}\ ]\ . \tag{14}$$

The linearized system for (13) is

$$F_i(x_k) + z^T Q_k^T \nabla F_i(x_k) = 0, \quad i = 1, ..., m, \tag{15}$$

which can be written as

$$F(x_k) + J_k Q_k z = 0, \tag{16}$$

where J_k is the Jacobian matrix of $F(x)$ at x_k:

$$J_k = J(x_k) = [\ \nabla F_1(x_k),\ \nabla F_2(x_k),\ ...,\ \nabla F_m(x_k)\ ]^T. \tag{17}$$

However, linear system (16) may not have a solution as the number of equations may not be the same as the number of the variables. Therefore, we can consider the reduced system

$$P_k^T [\ F(x_k) + J_k Q_k z\ ] = 0, \tag{18}$$

where

$$P_k = [\ p_1^{(k)},\ p_2^{(k)},\ ...,\ p_{i_k}^{(k)}\ ]\ , \tag{19}$$

is a full column rank matrix. We do not need the full Jacobian matrix J_k for solving the reduced linear system (18). Instead, we only require an approximate matrix $M_k \approx P_k^T J_k Q_k$, which is a square matrix in $\Re^{i_k \times i_k}$. It is easy to see that normally M_k has much fewer elements than the Jacobian matrix J_k. Thus, a general subspace method for nonlinear equations can be described as follows.

ALGORITHM 2. *(A general subspace algorithm for nonlinear equations)*

Step 1 *Given x_1, $\epsilon > 0$, $k := 1$.*

Step 2 *Generate P_k, Q_k and $M_k \approx P_k^T J(x_k) Q_k$;*
Compute the vector z_k by solving:

$$P_k^T F(x_k) + M_k z = 0, \tag{20}$$

Set $d_k = Q_k z_k$. If $\|d_k\| \le \epsilon$ then stop.

Step 3 *Carry out a line search, obtaining a stepsize α_k and set*

$$x_{k+1} = x_k + \alpha_k d_k. \tag{21}$$

Step 4 *$k := k + 1$, Go to Step 2.*

For the very special case when $m = n$ and $F(x)$ is linear, if we set $P_k = Q_k = e_k \in \Re^{n\times 1}$ and if we set $\alpha_k = 1$, the first n iterations of the above algorithm is exactly one Gauss-Seidel iteration.

For general nonlinear equations, we can also choose $P_k = Q_k$. However, such a choice may not always ensure that the search direction is a descent direction for the given merit function of the nonlinear equations. Suppose we use the L_∞ penalty function as the merit function, namely

$$P_\infty(x) = \|F(x)\|_\infty, \tag{22}$$

we can sort the violations in descent order:

$$|F_{j_1}(x_k)| \ge |F_{j_2}(x_k)| \ge ... \ge |F_{j_m}(x_k)|. \tag{23}$$

One possible way for choosing P_k is

$$P_k = [\ e_{j_1}, e_{j_2}, ..., e_{j_{i_k}}\ ]. \tag{24}$$

The following result is easily seen.

LEMMA 3. *Suppose* $|F_{j_{i_k+1}}(x_k)| < \|F(x_k)\|_\infty$, *if the search direction* d_k *satisfies*

$$P_k^T[\ F(x_k) + J_k d_k\ ] = 0, \tag{25}$$

then d_k *is a descent direction of* $\|F(x)\|_\infty$ *at* x_k.

Proof. Under the condition $|F_{j_{i_k+1}}(x_k)| < \|F(x_k)\|_\infty$, relation (25) implies that

$$\begin{aligned} \frac{d}{d\alpha}(\|F(x_k+\alpha d_k)\|_\infty)|_{\alpha=0} &= \frac{d}{d\alpha}\left(\max_{1\le i\le i_k}|F_{j_i}(x_k+\alpha d_k|)\right)|_{\alpha=0} \\ &= -\|F(x_k)\|_\infty < 0. \end{aligned} \tag{26}$$

This shows that d_k is a descent direction of $\|F(x)\|_\infty$. Q.E.D.

Choice (24) is also reasonable if the merit function is $\|F(x)\|_1 = \sum_{i=1}^m |F_i(x)|$, though in this case there is no theoretical guarantee that d_k will be a descent direction any more.

Now, we turn to the issue of choosing Q_k, the subspace for the search direction. A straightforward generalization to the Gauss-Seidel method is to choose $Q_k = P_k$. Namely, we can use

$$Q_k = [\ e_{j_1}, e_{j_2}, ..., e_{j_{i_k}}\ ], \tag{27}$$

where j_i are defined by (23). We can also choose different Q_k. Consider the type of general Q_k consisting of coordinate directions:

$$Q_k = [\ e_{l_1}, e_{l_2}, ..., e_{l_{i_k}}\ ], \tag{28}$$

where $l_j(j = 1, ..., i_k)$ are a subset of $\{1, 2, ..., n\}$. Such a choice of the subset indicating that a trial step in the subspace spanned by these coordinate directions will make the linearized functions with index set $\mathcal{I}_k = \{j_{i_1}, j_{i_2}, .., j_{i_k}\}$ zero. In order to make the linearized functions with index set $\{1, 2, ..., m\} \setminus \mathcal{I}_k$ having less increases, it is natural to require $\|d_k\|_2$ as small as possible. Intuitively, it is more or less equivalent to requiring the matrix

$$(P_k^T J_k Q_k)^{-1} \tag{29}$$

as small as possible. But, estimation of $(P_k^T J_k Q_k)^{-1}$ is not easy. Therefore, we suggest that Q_k can be chosen so that $\|P_k^T J_k Q_k\|$ is as large as possible for some norm. For example, Q_k can be so chosen in order to maximize the sum of the absolute values of all the elements of $P_k^T J_k Q_k$.

Similar to unconstrained optimization, we can use the previous search directions. Consider at the $k-th$ iteration, we already have $k-1$ previous directions $s_i(i = 1, 2, ..., k-1)$. We can let

$$Q_k = [\ s_1, s_2, ..., s_{k-1}, u_k\ ] \tag{30}$$

where u_k, not in the subspace $Span\{s_1, s_2, ..., s_{k-1}\}$, is a new direction that we should add to the subspace in the $k-$th iteration, otherwise the iterations may be trapped into a lower dimensional subspace. There are many ways to choose u_k. It can be randomly generated or be set to a coordinate unit vector. u_k can also be the residual vector $-F(x_k)$ if $m = n$. One good property of (30) is that we can take advantage of the relations

$$J_k s_i = J_k(x_{i+1} - x_i) \approx F(x_{i+1}) - F(x_i) = y_i. \tag{31}$$

Therefore we can let

$$M_k = P_k^T[\ y_1, y_2, ..., y_{k-1}, v_k\ ], \tag{32}$$

with v_k being an approximation to $J_k u_k$. One way to approximate $v_k \approx J_k u_k$ is by computing an additional function value $F(x_k + u_k)$ and setting

$$v_k = F(x_k + u_k) - F(x_k). \tag{33}$$

Actually we do not need to know the whole vector v_k in order to compute the search direction d_k. All we need is a shorter vector $P_k^T v_k$. In the special case $m = n$, we can let $P_k = Q_k$, then the first $k-1$ elements of $P_k^T v_k$ can be obtained by (31). Thus, we only require to approximate one number $u_k^T J_k u_k$. If u_k is a coordinate direction e_j, this number $u_k^T J_k u_k$ can be easily approximated by an additional calculation of $F_j(x_k + e_j)$, namely

$$e_j^T J_k e_j \approx F_j(x_k + e_j) - F_j(x_k)\ . \tag{34}$$

It is also possible to allow that P_k and Q_k have different dimensions. For example, at iteration k, we can choose two positive integers r_1 and r_2, and let $P_k \in \Re^{n\times r_1}$ and $Q_k \in \Re^{n\times r_2}$. If both r_1 and r_2 are much less than n, the linear system (18) is much smaller than the Newton's equation $F(x_k) + J_k d = 0$. Without requiring $r_1 = r_2$ would make (18) have no solution or have infinitely many solutions. One way is to find a least squares solution of (18), which will be very similar to the method that we discussed in the next section. Another possible way is to try to minimize $\|P_k^T[F(x_k) + J_k Q_k z]\|_1$ or $\|P_k^T[F(x_k) + J_k Q_k z]\|_\infty$.

4. Subspace techniques for nonlinear least squares

In this section, we consider the nonlinear least square problems (2). Similar to the previous section, we let

$$\mathcal{S}_k = Span\{q_1^{(k)}, q_2^{(k)}, ..., q_{i_k}^{(k)}\} \tag{35}$$

and define Q_k by (14). The second order Taylor expansion of $\|F(x)\|_2^2$ is

$$\|F(x_k)+J_kd\|_2^2+d^TW_kd, \tag{36}$$

where $W_k \in \Re^{n\times n}$ is defined by

$$\sum_{i=1}^{m} F_i(x_k)\nabla^2 F_i(x_k). \tag{37}$$

If we consider vectors d in the subspace $\mathcal{S}_k$, we would get the quadratic model

$$\bar{Q}_k(z)=\|F(x_k)+J_kQ_kz\|_2^2+z^TB_kz, \tag{38}$$

where $B_k \in \Re^{i_k\times i_k}$ approximates the reduced matrix

$$Q_k^TW_kQ_k=\sum_{i=1}^{m} F_i(x_k)Q_k^T\nabla^2 F_i(x_k)Q_k. \tag{39}$$

We can give a subspace algorithm for nonlinear least squares as follows.

ALGORITHM 4. *(A subspace trust region algorithm for nonlinear least squares)*

Step 1 Given $x_1 \in \Re^n$, $\Delta_1 > 0$, Choose matrices Q_1 and B_1 , Given $\epsilon > 0$, $k := 1$.

Step 2 Solve the subspace subproblem:

$$\min_{z\in\Re^{|i_k|}} \quad \bar{Q}_k(z)=\|F(x_k)+J_kQ_kz\|_2^2+z^TB_kz \tag{40}$$

$$s.\ t. \quad \|z\|_2 \le \Delta_k, \tag{41}$$

obtaining z_k, set $s_k = Q_kz_k$. If $\|s_k\| \le \epsilon$ then stop.

Step 3 Define

$$r_k=\frac{\|F(x_k)\|_2^2-\|F(x_k+s_k)\|_2^2}{\bar{Q}_k(0)-\bar{Q}_k(z_k)}. \tag{42}$$

Set

$$x_{k+1}=\begin{cases} x_k+s_k & \textit{if } f(x_k+s_k)<f(x_k);\\ x_k & \textit{otherwise.}\end{cases} \tag{43}$$

$$\Delta_{k+1}=\begin{cases} \frac{1}{2}\|z_k\|_2 & \textit{if } r_k<0.1;\\ 2\Delta_k & \textit{if } r_k>0.9 \textit{ and } 2\|z_k\|>\Delta_k\ ;\\ \Delta_k & \textit{otherwise.}\end{cases} \tag{44}$$

Step 4 Generate Q_{k+1} and B_{k+1}.

Step 5 $k := k + 1$, Go to Step 2.

The above algorithm is a modification of the standard trust region algorithm for nonlinear least squares(Yuan, 1998; Yuan, 2000a) by using our subspace approach. If the subspace $\mathcal{S}_k$ contains a descent direction for function $\|F(x)\|_2^2$ at x_k, we can get a lower bound for the predicted reduction $\bar{Q}_k(0) - \bar{Q}_k(z_k)$, which is stated as follows.

LEMMA 5. *If there is a nonzero vector $\bar{z}_k$ such that $\bar{d}_k = Q_k \bar{z}_k \in \mathcal{S}_k$ and that*

$$\bar{d}_k^T J_k^T F(x_k) < 0, \tag{45}$$

and if z_k is the minimizer of (40)-(41), then

$$\bar{Q}_k(0) - \bar{Q}_k(z_k) \geq \frac{1}{2}\xi_\Delta(x_k) \min\left\{1, \frac{\xi_\Delta(x_k)}{2\|B_k\|_2 \Delta_k^2}\right\}, \tag{46}$$

where

$$\xi_{\Delta_k}(x_k) = \max_{\|\alpha \bar{z}_k\|_2 \leq \Delta_k} [\|F(x_k)\|_2^2 - \|F(x_k) - \alpha J_k Q_k \bar{z}_k\|_2^2]. \tag{47}$$

Proof. By definition of z_k, we have that

$$\bar{Q}_k(0) - \bar{Q}_k(z_k) \geq \bar{Q}_k(0) - \min_{\|\alpha \bar{z}_k\|_2 \leq \Delta_k} \bar{Q}_k(\alpha \bar{z}_k). \tag{48}$$

From the convexity of $\|F(x_k) + J_k Q_k z\|_2^2$, similar to Powell (1970) and Yuan (1998), we have that

$$\bar{Q}_k(0) - \min_{\|\alpha \bar{z}_k\|_2 \leq \Delta_k} \bar{Q}_k(\alpha \bar{z}_k) \geq \frac{1}{2}\xi_\Delta(x_k) \min\left\{1, \frac{\xi_\Delta(x_k)}{2\|B_k\|_2 \Delta_k^2}\right\}, \tag{49}$$

where $\xi_{\Delta_k}(x_k)$ is defined by (47). Q.E.D.

In particularly, if $-J_k^+ F(x_k) \in \mathcal{S}_k$ and $-J_k^+ F(x_k)/\|J_k^+ F(x_k)\|_2$ is one column of Q_k, then

$$\bar{Q}_k(0) - \bar{Q}_k(z_k) \geq \frac{1}{2}\eta_\Delta(x_k) \min\left\{1, \frac{\eta_\Delta(x_k)}{2\|B_k\|_2 \Delta_k^2}\right\}, \tag{50}$$

where

$$\eta_{\Delta_k}(x_k) = \max_{\|\alpha J_k^+ F(x_k)\|_2 \leq \Delta_k} [\|F(x_k)\|_2^2 - \|F(x_k) - \alpha J_k J_k^+ F(x_k)\|_2^2]. \tag{51}$$

Using (51), we can establish the global convergence of Algorithm 4. In the above algorithm 4, the trust region constraint is imposed on the reduced variables z instead of the original variables x. Thus, the actual length of the trial step s_k depends on both the length of z_k and the matrix Q_k. For example, if we double the matrix Q_k, the trial step s_k will be also doubled. Thus it is reasonable to choose Q_k in such a way that all the columns of Q_k are unit-length vectors. For example, s_i can be replace by $s_i/\|s_i\|_2$. For numerical reasons, Q_k can be chosen such that $Q_kQ_k^T$ is a projection from $\Re^n$ to $\mathcal{S}_k$.

Instead of imposing the trust region constraint to z_k, we can also directly require the trial step in the original space to satisfy the trust region condition. Namely we can modify the statements of Algorithm 4 slightly so that s_k is the solution of

$$\min_{s\in\mathcal{S}_k} \quad \hat{Q}_k(s) = \|F(x_k) + J_k s\|_2^2 + s^T\hat{B}_k s, \tag{52}$$

$$\text{s. t.} \quad \|s\|_2 \le \Delta_k. \tag{53}$$

Thus any orthogonal basis of $\mathcal{S}_k$ consists of a matrix Q_k. We can choose the Q_k properly so that the subproblem (40)-(41) is easy to solve and the approximate matrix $B_k = Q_k^T\hat{B}_kQ_k$ is easy to obtain.

Similar to unconstrained optimization, an obvious choice is

$$\mathcal{S}_k = Span\{-J_k^+F(x_k), s_{k-1}, s_{k-2}, ..., s_2, s_1\}. \tag{54}$$

Other choices are also possible, for example we can replace $-J_k^+F(x_k)$ in (54) by a randomly generated unite vector or any descent coordinate direction. One thing that is different from the previous section is the approximation of B_k. Because this matrix is an approximation to the projection (or reduction) of the Hessian matrix (37), it looks like that we will have to use some second order information of $F(x)$ which seems impossible. However, similar to the techniques discussed in the previous section, if we use (54), most elements of B_k can be expressed by $s_i\nabla^2F_t(x_k)s_j$. Thus, all we need to do is trying to use the function values $F_t(x_j)$, $F_t(x_{j+1})$, $F_t(x_i)$ and $F_t(x_{i+1})$ to get a good estimate to the term $s_i\nabla^2F_t(x_k)s_j$.

In the objective function in (40), we can replace the residual of the linearized equations $F(x_k) + J_kQ_kz$ by a reduced residual, which would be a shorter vector. For example, we can consider the following subproblem

$$\min_{z\in\Re^{|i_k|}} \quad \bar{Q}_k(z) = \|\ P_k^T[F(x_k) + J_kQ_kz]\ \|_2^2\ +\ z^TB_kz \tag{55}$$

$$\text{s. t.} \quad \|z\|_2 \le \Delta_k, \tag{56}$$

where $P_k \in \Re^{m\times p_k}$ with $p_k << m$. One special case is that all the columns of P_k are the canonical vectors e_i, which means that only a selection of p_k terms are used in the sum of squares. Similar to the previous section, we can select the terms with large residuals at the current iteration.

5. Discussions

We have presented subspace methods for solving large scale nonlinear equations and nonlinear least squares problems. These methods are specially designed for large scale problems as they only solve subproblems in subspace with dimension much less than the dimension of the original large scale problems. Such methods differ from the standard optimization methods because they require the trial step or the search direction in a low dimensional subspace. The key issues for such methods are how to choose the subspaces and how to obtain the approximate model in corresponding subspaces quickly. In the paper we give some suggestions on these two issues. Of course there are definitely many other possibilities except what we have discussed above. And moreover we have not yet tested our ideas by numerical examples. This indicates that further studies are needed on subspace methods.

References

A. Bouaricha and J.J. Moré, “Impact of Partial Separability on Large-Scale Optimization”, *Computational Optimization and Applications*, Vol. 7, No. 1, pp. 27-40, 1997.

A.R. Conn, N.J.M. Gould and Ph.L. Toint, *Trust-Region Methods*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2000.

J.E. Dennis and R.B. Schnable *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (SIAM, Philadelphia, 1993).

R. Fletcher, *Practical Methods of Optimization, Second Edition*, John Wiley and Sons, 1987.

P.E. Gill and M.W. Leonard, “Reduced-Hessian quasi-Newton methods for unconstrained optimization”, *SIAM J. Optim.* , Vol. 12, pp. 209-237, 2001.

G.H. Golb and Ch.F. Van Loan, *Matrix Computations, 3rd Edition*, (Johns Hopkins University Press, Baltimore and London, 1996).

N.I.M. Gould, D. Orban and Ph.L. Toint, “Numerical methods for large-scale nonlinear optimization”, *Acta Numerica*, pp. 299-361, 2005

N.I.M. Gould and Ph.L. Toint, “FILTRANE, a Fortran 95 filter-trust-region package for solving nonlinear least-squares and nonlinear feasibility problems”, *ACM Transactions on Mathematical Software (TOMS)*, Vol. 33, No. 1, pp. 3-25, 2007.

C.T. Kelly, *Iterative Methods for Linear and Nonlinear Equations* (SIAM, Philadelphia, 1995).

D.C. Liu and J. Nocedal, “On the limited memory BFGS method for large scale optimization”, *Math. Program.*, Vol. 45, pp. 503-528, 1989.

E. Mizutani and J.W. Demmel, “On structure-exploiting trust-region regularized nonlinear least squares algorithms for neural-network learning”, *Neural Networks*, Vol. 16, No. 5-6, pp. 745-753, 2003.

J. Nocedal and S.J. Wright, *Numerical Optimization* (Springer, New York, 1999).

J.M. Ortega and W.C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, (Academic Press, New York and London, 1970).

M.J.D. Powell, "A new algorithm for unconstrained optimization", in: J.B. Rosen, O.L. Mangasarian and K. Ritter, eds., *Nonlinear Programming*, pp. 31-66, Academic Press, New York, 1970.

T. Steihaug, "The conjugate gradient method and trust regions in large scale optimization", *SIAM J. Numerical Analysis*, Vol. 20, pp. 626-637, 1983.

J. Stoer and Y. Yuan, "A subspace study on conjugate gradient algorithms", *ZAMM Z. angew. Math. Mech.*, Vo. 75, pp. 69-77, 1995.

W.Y. Sun and Y. Yuan, *Optimization Theory and Mehtods: Nonlinear Programming*, Springer Optimization and Its Application, Vol. 1, Springer, 2006.

Ph.L. Toint, "Towards an efficient sparsity exploiting Newton method for minimization", in: I. Duff, ed. *Sparse Matrices and Their Uses*, pp. 57-88, (Academic Press, 1981).

Ph.L. Toint, "On large scale nonlinear least squares calculations", *SIAM J. Sci. Stat. Comput.*, Vol. 8, No. 3, pp. 416-435, 1987.

J. Vlček and L. Lukšan, "New variable metric methods for unconstrained minimization covering the large-scale case", Technical report No. V876, Institue of Computer Science, Academy of Sciences of the Czech Republic, October 2002.

Z.H. Wang, Z.W. Wen and Y. Yuan, "A subspace trust region method for large scale unconstrained optimization", in; Y. Yuan, ed. *Numerical Linear Algebra and Optimization*, pp. 264-274, Science Press, 2004.

Z.H. Wang and Y. Yuan, "A Subspace implementation of quasi-Newton trust region methods for unconstrained optimization", *Numerische Mathematik* Vol. 104, pp. 241-269, 2006.

Y. Yuan, "Trust region algorithms for nonlinear equations", *Information* Vol. 1, pp. 7-20, 1998.

Y. Yuan, A review of trust region algorithms for optimization, in ICM99: Proceedings of the Fourth International Congress on Industrial and Applied Mathematics, J. M. Ball and J. C. R. Hunt, eds., Oxford University Press,Oxford, 2000, pp. 271–282. (2000a)

Y. Yuan, "On the truncated conjugate gradient method", *Math. Program.*, Vol. 86, pp. 561-571, 2000. (2000b)

Y. Yuan, "Subspace techniques for nonlinear optimization", in: R. Jeltsch, D.Q. Li and I. H. Sloan, eds., *Some Topics in Industrial and Applied Mathematics (Series in Contemporary Applied Mathematics CAM 8)* pp. 206-218. (Higher Education Press. Beijing, 2007).