A review on subspace methods for nonlinear optimization

Ya-xiang Yuan

Abstract. In this paper, we review various subspace techniques that have been used in constructing numerical methods for solving nonlinear optimization problems. As large scale optimization problems are attracting more and more attention in recent years, subspace methods are getting more and more important since they do not require solving large scale subproblems in each iteration. The essential parts of a subspace method are how to construct subproblems defined in lower dimensional subspaces and how to choose the subspaces in which the subproblems are defined. Various subspace methods for unconstrained optimization, constrained optimization, nonlinear equations and nonlinear least squares, and matrix optimization problems are given respectively, and different proposals are made on how to choose the subspaces.

Mathematics Subject Classification (2010). Primary 65K05; Secondary 90C30.

Keywords. numerical methods, nonlinear optimization, subspace techniques, subproblems.

1. Introduction

Nonlinear optimization problems have the following form:

$$\min_{x\in\Re^n} \quad f(x) \tag{1.1}$$

$$\text{subject to} \quad c_i(x) = 0, \qquad i = 1, ..., m_e\,, \tag{1.2}$$

$$c_i(x) \geq 0, \qquad i = m_e + 1, ..., m\,, \tag{1.3}$$

where m and m_e are integers satisfying $m \geq m_e \geq 0$, $f(x)$ and $c_i(x)(i = 1, ..., m)$ are real functions defined in $\Re^n$ and at least one of functions $f(x)$ and $c_i(x)(i = 1, ..., m)$ is nonlinear. If there is no constraint, namely $m = m_e = 0$, problem (1.1) is called an unconstrained optimization problem, otherwise problem (1.1)-(1.3) is called a constrained optimization problem.

Numerical methods for nonlinear optimization are iterative. At the $k-$th iteration, if the current iterate point x_k is not a solution, we try to compute a "better" point x_{k+1} and continue the process so that it will stop at a solution or generate a sequence which, hopefully, converges to a solution.

There are mainly two classes of numerical methods for nonlinear optimization. One class is line search methods in which the next iterate point is obtained by searching along a search direction. Namely, we let

$$x_{k+1} = x_k + \alpha_k d_k \tag{1.4}$$

Proceedings of the International Congress of Mathematicians, Seoul, 2014

where $d_k \in \Re^n$ is a search direction and $\alpha_k > 0$ is a step-length. The other class of methods are trust region algorithms, where a trial step s_k in a trust region is computed and then the algorithm decides whether the trail step should be accepted. The trust region is normally a small neighbourhood centered at the current iterate point x_k. Generally, the search direction or the trial step are obtained by solving a subproblem which is an approximation to the original nonlinear optimization problem. Convergence results of numerical methods for nonlinear optimization are normally based on the reduction of a penalty function. For example, the step-length α_k in a line search algorithm is chosen in such a way that sufficient reduction in the penalty function is achieved. Trial steps in a trust region algorithm will be accepted if the penalty function is reduced. A penalty function can be viewed as a combined measure for the two tasks of nonlinear optimization: reducing the objective function and satisfying the constraints. Another approach for ensuring global convergence of numerical methods for nonlinear optimization is the filter technique, which measures the constraint violation and objective function value as a two dimensional array. Detailed discussions on numerical methods for nonlinear optimization can be found in [32].

Due to their broad applications in many fields, large scale optimization problems are attracting more and more attention in recent years. However, even though the subproblems for computing search directions and trial steps are simpler than the original nonlinear optimization problems, they are still linear or quadratic problems large-scale in nature, as they are also defined in the same dimensional space as the original nonlinear problem. For example, in the $k-$th iteration, the sequential quadratic programming method for nonlinear optimization needs to solve the following quadratic programming subproblem:

$$\min_{x \in \Re^n} \quad Q_k(d) \tag{1.5}$$

$$\text{s. t.} \quad c_i(x_k) + d^T \nabla c_i(x_k) = 0, \qquad i = 1, ..., m_e\,, \tag{1.6}$$

$$c_i(x_k) + d^T \nabla c_i(x_k) \geq 0, \qquad i = m_e + 1, ..., m\,, \tag{1.7}$$

where $Q_k(d)$ is a quadratic approximation to the Lagrangian function. Though the above quadratic programming subproblem is simpler than the original nonlinear optimization problem, it is still large scale when the original nonlinear problem is large scale.

Therefore, it is important to study subspace techniques [9, 17, 41] due to the fact that subspace methods do not need to solve large scale subproblems in each iteration. In general, a subspace method searches in a lower dimensional subspace to obtain the search direction or the trust region step. Thus, in each iteration, we only need to solve a subproblem that is defined in a lower dimensional subspace.

In addition to the practical computation considerations, there are other reasons that motivated us to study numerical methods based on subspace techniques. First, let us consider a standard full space line search method. The search direction d_k is normally obtained by solving an approximation model based on the full space. For example, the search direction of the Newton's method is obtained by minimizing the second order Taylor expansion of a general nonlinear function in the whole space. Therefore, one can view that the computation of d_k is very aggressive as it is obtained through an optimistic approach by trusting the corresponding approximate model in the whole space. Once d_k is obtained, the line search procedure of computing the step-length α_k tries to minimize the one dimensional function $f(x_k + \alpha d_k)$. Thus, the computation of α_k is very conservative as it is obtained by searching in a one dimensional subspace. Thus, a standard full space line search algorithm swings between full space approximations and one-dimensional subspace searches.

Another motivation is from our long time studies on nonlinear conjugate gradient methods [10]. The search direction of a nonlinear conjugate gradient method for unconstrained optimization problem (1.1) has the form

$$d_k = -\nabla f(x_k) + \beta_k d_{k-1}\,, \tag{1.8}$$

where β_k is defined by certain conjugate conditions. Typical choices of β_k are as follows:

$$\beta_k^{HS} = \frac{g_{k+1}^T(g_{k+1}-g_k)}{d_k^T(g_{k+1}-g_k)}\,, \qquad \beta_k^{FR} = \frac{||g_{k+1}||_2^2}{||g_k||_2^2}\,, \tag{1.9}$$

$$\beta_k^{PRP} = \frac{g_{k+1}^T(g_{k+1}-g_k)}{||g_k||_2^2}\,, \qquad \beta_k^{DY} = \frac{||g_{k+1}||_2^2}{d_k^T(g_{k+1}-g_k)}\,. \tag{1.10}$$

We have two observations on the nonlinear conjugate gradient methods. Firstly, no matter which β_k is used, the new point $x_{k+1} = x_k + \alpha_k d_k$ is always in the 2-dimensional subspace $x_k + \text{span}\{-g_k, d_{k-1}\}$. Secondly, the conjugacy property is a good property only when it is associated with exact line searches. Therefore, instead of studying which formulae for β_k would lead to a good nonlinear conjugate gradient method, we should ask ourselves a different question: which point x in the two-dimensional space $x_k + \text{span}\{-g_k, d_{k-1}\}$ is the *best* point?

The third motivation for us to study subspace algorithms is the famous limited memory quasi-Newton method. Quasi-Newton methods for nonlinear optimization use quadratic models in which the Hessian is a quasi-Newton matrix updated from iteration to iteration and satisfies the following quasi-Newton equation:

$$B_k s_{k-1} = y_{k-1}\,, \tag{1.11}$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$. An example of quasi-Newton update is the famous Broyden-Fletcher-Goldfarb-Shanno (BFGS) update:

$$\begin{aligned} B_k &= U^{BFGS}(B_{k-1}, s_{k-1}, y_{k-1}) \\ &= B_{k-1} - \frac{B_{k-1}s_{k-1}s_{k-1}^T B_{k-1}}{s_{k-1}^T B_{k-1} s_{k-1}} + \frac{y_{k-1}y_{k-1}^T}{s_{k-1}^T y_{k-1}}\,. \end{aligned} \tag{1.12}$$

For extremely large scale optimization problems, such as those derived from numerical weather prediction and data assimilation, we can not afford to store a full quasi-Newton matrix. To overcome such difficulties, Liu and Nocedal[21] proposed the limited memory BFGS method, which generates the quasi-Newton matrix by using the vectors s and y in the previous m iterations. Namely, $B_k^{(0)} = \sigma_k I$ and

$$B_k^{(i)} = U^{BFGS}(B_k^{(i-1)}, s_{k-m-1+i}, y_{k-m-1+i})\,,$$

for $i = 1, ..., m$. Eventually, the quasi-Newton matrix in the limited memory BFGS method has the following representation:

$$B_k = B_k^{(m)} = \sigma_k I + [S_k \quad Y_k]\, T_k \begin{bmatrix} S_k^T \\ Y_k^T \end{bmatrix}\,,$$

where T_k is a $2m \times 2m$ symmetric matrix and

$$[S_k \quad Y_k] = [s_{k-1}, s_{k-2}, ...s_{k-m}, y_{k-1}, y_{k-2}, ..., y_{k-m}] \in \Re^{n\times 2m}\,.$$

In a line search method we have $s_k = \alpha_k d_k = -\alpha_k B_k^{-1} g_k$ for some $\alpha_k > 0$, while in a trust region algorithm $s_k = -(B_k + \lambda_k I)^{-1} g_k$ for some $\lambda_k \geq 0$. Thus, in either case, we have

$$x_{k+1} - x_k \in \text{span}\{g_k, s_{k-1}, ..., s_{k-m}, y_{k-1}, ..., y_{k-m}\}\,. \tag{1.13}$$

This shows that limited memory quasi-Newton methods always produce a step in a lower dimensional subspace.

The block coordinate descent (BCD) method is a technique that is widely used in computational mathematics. From subspace point of view, the BCD method is a very special subspace method whose subspaces are spanned by coordinate directions. The method partitions the variables into a few blocks and then minimizes the objective function with respect to each block by fixing all other blocks at each iteration. It has been studied in convex programming [25], nonlinear programming [2], semidefinite programming [35], compressive sensing [11, 24], etc. A popular extension of the BCD method is the alternating direction method of multipliers (ADMM) by minimizing the augmented Lagrangian function blocks by blocks and then updating the Lagrangian multipliers. It dates back to optimization problems arising from partial differential equations (PDEs) [14–16], and has been applied to semidefinite programming [37], compressive sensing [40], distributed computation [5] and many other areas.

Parallel computation methods can also be viewed as subspace techniques. For example, the domain decomposition technique of Tai and Xu[39] decomposes the n dimensional space into p lower dimensional subspaces using the domain decomposition technique, and p processors search in parallel in the corresponding subspaces.

A general subspace approach requires

$$x_{k+1} - x_k \in \mathcal{S}_k\,, \tag{1.14}$$

where $\mathcal{S}_k$ is a subspace in $\Re^n$ with the good feature that the dimension τ_k of S_k being much less than n. An advantage of subspace approaches is that the subproblems for computing searching directions or trust region trial steps are defined in lower dimensional subspaces, which enables us to solve the corresponding subproblems quickly. Moreover, for many cases, we could show that subspace approaches attain good theoretical properties as full space models.

In a subspace method, the dimension of the subspace τ_k is either fixed or updated from iteration to iteration. $\mathcal{S}_{k+1}$ is normally updated from $\mathcal{S}_k$. Often $\mathcal{S}_{k+1}$ is obtained by adding some new directions $d_i^{(k)} (i = 1, ..., m)$:

$$\mathcal{S}_{k+1} = \text{span}\{\mathcal{S}_k, d_1^{(k)}, ..., d_m^{(k)}\}\,.$$

The directions $d_i^{(k)}$ to be added can be randomly generated or constructed based on the iteration information at the current iterate in order to improve the subspace. Sometimes, it is reasonable to remove some directions from the current subspace to avoid redundancy or to prevent the dimension of the subspace from increasing too rapidly. Moreover, it is reasonable for us to delete directions along which significant function reductions are not possible to obtain.

2. Subspace algorithms for unconstrained optimization

Consider a trust region algorithm for unconstrained optimization

$$\min_{x\in\Re^n} f(x)\,. \tag{2.1}$$

The trust region subproblem (TRS) is normally

$$\min_{d\in\Re^n} \quad Q_k(d) = g_k^T d + \frac{1}{2} d^T B_k d \tag{2.2}$$

$$\text{s. t.} \quad \|d\|_2 \le \Delta_k\,, \tag{2.3}$$

where $g_k = \nabla f(x_k)$, B_k is an approximate to $\nabla^2 f(x_k)$ and $\Delta_k > 0$ is the trust region bound.

When the approximate Hessian B_k is generated by quasi-Newton updates, the trust region subproblem has subspace properties. First, we have the following result

Lemma 2.1 ([34]). *Suppose $B_1 = \sigma I$, $\sigma > 0$. The matrix updating formula is any one chosen from amongst SR1, PSB and Broyden family, and B_k is the k-th updated matrix. s_k is the solution of TRS, $x_{k+1} = x_k + s_k$, $g_k = \nabla f(x_k)$. Let $\mathcal{G}_k = span\{g_1, g_2, \cdots, g_k\}$. Then $s_k \in \mathcal{G}_k$ and for any $z \in \mathcal{G}_k$, $w \in \mathcal{G}_k^{\perp}$, we have*

$$B_k z \in \mathcal{G}_k, \quad B_k u = \sigma u\,. \tag{2.4}$$

The above lemma shows that quasi-Newton matrices have very nice subspace properties. Similar results for line search QN methods are given by Gill and Leonard[13].

From the above lemma, it is not difficult to prove the following theorem.

Theorem 2.2 ([34]). *If $\mathcal{S}_k = span\{g(x_1), ..., g(x_k)\}$. The subspace trust region algorithm will generate the same sequences as the full space trust region quasi-Newton algorithm for unconstrained optimization if the $B_1 = \sigma I$ and B_k is updated by SR1, PSB and Broyden's family.*

Based on the above results, a subspace trust region quasi-Newton method for large scale unconstrained optimization is presented by Wang and Yuan[34].

Now, we discuss a special trust region subproblem which makes good use of subspace properties. If we replace the $||.||_2$ by a general norm $||.||_W$ in (2.3), we obtain a general TRS subproblem

$$\min_{s\in\Re^n} \quad g^T s + \frac{1}{2} s^T B s \tag{2.5}$$

$$\text{s. t.} \quad \|s\|_W \le \Delta\,, \tag{2.6}$$

where $\|.\|_W$ is any norm in $\Re^n$. A natural question is which norm $||.||_W$ we should use. Intuitively, we should choose the norm $\|.\|_W$ properly so that the trust region subproblem can easily be solved by using the corresponding subspace properties of the objective function $g^T s + \frac{1}{2} s^T B s$. Assume that B is a limited memory quasi-Newton matrix which is expressed as $B = \sigma I + PDP^T$, where $P \in \Re^{n\times l}$ satisfies $P^T P = I$. If we define a cylinder norm:

$$\|s\|_P = \max\{\|P^T s\|_\infty,\ \|P_\perp^T s\|_2\}\,, \tag{2.7}$$

where $P_\perp^T$ is the projection onto the space orthogonal to range(P). Due to the definition of $||.||_P$, the solution s of the P norm trust region subproblem

$$\min_{s\in\Re^n} \quad g^T s + \frac{1}{2} s^T B s \tag{2.8}$$

$$\text{s. t.} \quad \|s\|_P \le \Delta\,, \tag{2.9}$$

can be expressed by $Ps_1 + P_\perp s_2$, where s_1 is the solution of the bound-constrained quadratic programming problem

$$\min_{s\in\Re^l} \quad s^T(P^T g) + \frac{1}{2} s^T(\sigma I + D)s \tag{2.10}$$

$$\text{s. t.} \quad \|s\|_\infty \le \Delta\,, \tag{2.11}$$

and s_2 is solution of the 2-norm constrained quadratic programming problem

$$\min_{s\in\Re^{n-l}} \quad s^T(P_\perp^T g) + \frac{1}{2}\sigma s^T s \tag{2.12}$$

$$\text{s. t.} \quad \|s\|_2 \le \Delta\,. \tag{2.13}$$

It is easy to see that both s_1 and s_2 have closed form solutions:

$$(s_1)_i = \begin{cases} \frac{-(P^T g)_i}{\sigma + D_{ii}} & \text{if} \quad |(P^T g)_i| < (\sigma + D_{ii})\Delta\,, \\ \Delta \text{sign}(-(P^T g)_i) & \text{otherwise}\,, \end{cases} \tag{2.14}$$

$i = 1, ..., l$, and

$$s_2 = -\min\left(\frac{1}{\sigma}\,,\ \frac{\Delta}{||P_\perp^T g||}\right) P_\perp^T g\,. \tag{2.15}$$

Numerical results based on a trust region algorithm that uses the P-norm trust region subproblem are given by [6].

In a general line search type subspace algorithm for unconstrained optimization, we obtain the search direction by solving a subproblem defined in the subspace:

$$\min_{d\in\mathcal{S}_k} m_k(d)\,, \tag{2.16}$$

where $m_k(d)$ is an approximation to $f(x_k + d)$ for d in the subspace $\mathcal{S}_k$ It would be desirable that the approximation model $m_k(d)$ has the following properties: it is easy to minimize in the subspace $\mathcal{S}_k$, it is a good approximation to f and the solution of the subspace subproblem will yield a sufficient reduction in the original objective function f.

It is natural to use quadratic approximations to the objective function. This leads to quadratic models in subspaces. Let $dim(\mathcal{S}_k) = \tau_k$ and

$$\mathcal{S}_k = \text{span}\{p_1, p_2, ..., p_{\tau_k}\}\,.$$

Define $P_k = [p_1, p_2, ..., p_{\tau_k}]$. Thus, the subspace condition $d \in \mathcal{S}_k$ is satisfied if we let $d = P_k\bar{d}$ for $\bar{d} \in \Re^{\tau_k}$. The quadratic function $Q_k(d)$ defined in the subspace can be expressed as a function $\bar{Q}_k$ in a lower dimension space $\Re^{\tau_k}$: $Q_k(d) = \bar{Q}_k(\bar{d})$.

Now, we discuss possible choices for the subspace $\mathcal{S}_k$. First, we consider the special subspace

$$\mathcal{S}_k = \text{span}\{-g_k, s_{k-1}, ..., s_{k-m}\}\,. \tag{2.17}$$

In this case, any vector d in the subspace $\mathcal{S}_k$ has the following form:

$$d = \alpha g_k + \sum_{i=1}^{m} \beta_i s_{k-i} = (-g_k, s_{k-1}, \cdots, s_{k-m})\bar{d} \tag{2.18}$$

where $\bar{d} = (\alpha, \beta_1, \cdots, \beta_m)^T \in \Re^{m+1}$. By using the secant conditions, we estimate all the second order terms of the Taylor expansion of $f(x_k + d)$ in the subspace $\mathcal{S}_k$

$$s_{k-i}^T \nabla^2 f(x_k) s_{k-j} \approx s_{k-i}^T y_{k-j}\,, \quad s_{k-i}^T \nabla^2 f(x_k) g_k \approx y_{k-i}^T g_k\,, \tag{2.19}$$

except one term $g_k^T \nabla^2 f(x_k) g_k$. Therefore, it is reasonable to use the following quadratic model in the subspace $\mathcal{S}_k$:

$$\bar{Q}_k(\bar{d}) = (-\|g_k\|^2, g_k^T s_{k-1}, \cdots, g_k^T s_{k-m})\bar{d} + \frac{1}{2}\bar{d}^T \bar{B}_k \bar{d}\,, \tag{2.20}$$

where

$$\bar{B}_k = \begin{pmatrix} \rho_k & -g_k^T y_{k-1} & \cdots & -g_k^T y_{k-m} \\ -g_k^T y_{k-1} & y_{k-1}^T s_{k-1} & \cdots & y_{k-m}^T s_{k-1} \\ \vdots & \vdots & \ddots & \vdots \\ -g_k^T y_{y-m} & y_{k-m}^T s_{k-1} & \cdots & y_{k-m}^T s_{k-m} \end{pmatrix} \tag{2.21}$$

with $\rho_k \approx g_k^T \nabla^2 f(x_k) g_k$. Hence, once we have a good estimate to the term $g_k^T \nabla^2 f(x_k) g_k$, we obtain a good quadratic model in the subspace $\mathcal{S}_k$.

There are different ways to choose ρ_k. Similarly to Stoer and Yuan[31], we let

$$\rho_k = 2\frac{(s_{k-1}^T g_k)^2}{s_{k-1}^T y_{k-1}}\,, \tag{2.22}$$

due to the fact that the mean value of $\cos^2(\theta)$ is $\frac{1}{2}$, which gives

$$g_k^T \nabla^2 f(x_k) g_k = \frac{1}{\cos^2\theta_k}\frac{(s_{k-1}^T \nabla^2 f(x_k) g_k)^2}{s_{k-1}^T \nabla^2 f(x_k) s_{k-1}} \approx 2\frac{(s_{k-1}^T g_k)^2}{s_{k-1}^T y_{k-1}}\,, \tag{2.23}$$

where θ_k is the angle between $(\nabla^2 f(x_k))^{\frac{1}{2}} g_k$ and $(\nabla^2 f(x_k))^{\frac{1}{2}} s_{k-1}$. Another way to estimate $g_k^T(\nabla^2 f(x_k))g_k$ is to replace $\nabla^2 f(x_k)$ by a quasi-Newton matrix. We can also obtain ρ_k by computing an extra function value $f(x_k + tg_k)$ and setting

$$\rho_k = \frac{2(f(x_k + tg_k) - f(x_k) - t\|g_k\|_2^2)}{t^2}\,. \tag{2.24}$$

By letting the second order curvature along g_k to be the average of those along s_{k-i} $(i = 1, ..., m)$, we get

$$\rho_k = \frac{\|g_k\|_2^2}{m}\sum_{i=1}^{m}\frac{s_{k-i}^T y_{k-i}}{s_{k-i}^T s_{k-i}}\,. \tag{2.25}$$

Suppose $g_k^T \nabla^2 f(x_k) g_k = \rho$, we have $d(\rho) =$

$$(-g_k, s_{k-1}, \ldots, s_{k-m}) \begin{pmatrix} \rho & -g_k^T y_{k-1} & \cdots & -g_k^T y_{k-m} \\ -g_k^T y_{k-1} & y_{k-1}^T s_{k-1} & \cdots & y_{k-m}^T s_{k-1} \\ \vdots & \vdots & \ddots & \vdots \\ -g_k^T y_{y-m} & y_{k-m}^T s_{k-1} & \cdots & y_{k-m}^T s_{k-m} \end{pmatrix}^{-1} \begin{pmatrix} -\|g_k\|^2 \\ g_k^T s_{k-1} \\ \cdots \\ g_k^T s_{k-m} \end{pmatrix}$$

Using

$$(B + \rho e e^T)^{-1} = B^{-1} - \frac{\rho}{1 + \rho e^T B^{-1} e} B^{-1} e e^T B \,,$$

we could show that the solution set is on a line:

$$d(\rho) = d(+\infty) + \alpha(\rho)\hat{d} \,.$$

Thus, instead of estimating an ideal ρ, we can carry out a line search for ρ to achieve sufficient reduction in the objective function.

Similar to (2.17), a slightly different subspace is

$$\mathcal{S}_k = \text{span}\{-g_k, y_{k-1}, ..., y_{k-m}\} \,. \tag{2.26}$$

In this case, any vector in $\mathcal{S}_k$ is represented as

$$d = \alpha g_k + \sum_{i=1}^{m} \beta_i y_{k-i} = W_k \bar{d} \tag{2.27}$$

where $W_k = [-g_k, y_{k-1}, ..., y_{k-m}] \in \Re^{n \times (m+1)}$. The Newton's step in the subspace $\mathcal{S}_k$ is $W_k \bar{d}_k$ with

$$\bar{d}_k = -\left[W_k^T \nabla^2 f(x_k) W_k\right]^{-1} W_k^T \nabla f(x_k) \,. \tag{2.28}$$

Thus, the remaining issue we need to consider is to obtain a good estimate of $\bar{d}_k$, using the fact that all the elements of $\left[W_k^T (\nabla^2 f(x_k))^{-1} W_k\right]$ is known except one entry $g_k \nabla^2 f(x_k)^{-1} g_k$.

Due to the property of (1.13), it is reasonable to use

$$\mathcal{S}_k = \text{span}\{-g_k, s_{k-1}, ..., s_{k-m}, y_{k-1}, ..., y_{k-m}\} \,. \tag{2.29}$$

This subspace is used by [33] where a subspace trust region limited memory quasi-Newton method is presented.

Now, we consider subspaces spanned by coordinate directions. Such subspaces have sparsity structures. First, let us sort $|(g_k)_i|$ by the descending order

$$|(g_k)_{i_1}| \geq |(g_k)_{i_2}| \geq |(g_k)_{i_3}| \geq \cdots \,. \tag{2.30}$$

We call the subspace

$$\mathcal{S}_k = \text{span}\{e_{i_1}, e_{i_2}, ..., e_{i_\tau}\} \tag{2.31}$$

the *τ-steepest coordinates subspace*. One good property of the steepest coordinates subspace is that the steepest descent direction in the subspace is a sufficiently descent direction, namely

$$\min_{d \in \mathcal{S}_k} \frac{d^T g_k}{||d||_2 ||g_k||_2} \leq -\frac{\tau}{n} \,. \tag{2.32}$$

If $(g_k)^2_{i_{\tau+1}} \leq \epsilon \sum_{j=1}^{\tau} (g_k)^2_{i_j}$, we obtain the following estimate:

$$\min_{d \in \mathcal{S}_k} \frac{d^T g_k}{||d||_2 ||g_k||_2} \leq -\frac{1}{\sqrt{1+\epsilon(n-\tau)}}\,. \tag{2.33}$$

By sequentially adding steepest coordinate directions into the subspace, we obtain a *sequential steepest coordinates search* (SSCS) technique. As an example, let us consider applying the sequential steepest coordinates search to the minimization of a convex quadratic function

$$Q(x) = g^T x + \frac{1}{2} x^T B x\,.$$

Algorithm 2.3. (Sequential steepest coordinates search for quadratic functions)

Step 1 *Given* x_1. $k := 1$.

Step 2 *Compute* $g_k = \nabla Q(x_k)$, *if* $||g_k|| = 0$ *then stop;*
Choose $i_k = \arg\min_i \{|(g_k)_i|\}$.

Step 3 *Let* $S_k = \mathrm{span}\{e_{i_1}, ..., e_{i_k}\}$,
Find $x_{k+1} = \arg\min_{x \in x_1 + S_k} Q(x)$;
Go to Step 2.

The sequential steepest coordinates search could be used to obtain an approximate sparse solution of linear least square problems. For example, consider the following sparsity constraint linear least squares problem:

$$\min_{x \in \Re^n} \quad ||Ax - b||_2^2 \tag{2.34}$$

$$\text{s. t.} \quad ||x||_0 \leq r\,, \tag{2.35}$$

where $A \in \Re^{n \times m}$, $b \in \Re^m$, r is a positive integer less than n, and $||x||_0$ is the number of non-zero elements of vector x. If Algorithm 2.3 is applied to $\min Q(x) = \frac{1}{2}||Ax - b||_2^2$, it will give a greedy algorithm for (2.34)-(2.35).

Algorithm 2.4. (SSCS for linear least squares)

Step 1 $x_1 = 0$, $g = A^T b$, $i_1 = \arg\max\{|(g)_i|\}$, $p_1 = e_{i_1}$, *given* $\epsilon > 0$.

Step 2 $\alpha_k = \arg\min_\alpha Q(x_k + \alpha p_k)$,
$x_{k+1} = x_k + \alpha_k p_k$,

Step 3 *If* $k \geq r$ *then stop;* $g := g - \alpha A^T A p_k$;
If $||g||_2 \leq \epsilon$ *then stop;*

Step 4 *let* $i_{k+1} = \arg\max_i \{|(g)_i|\}$;
let $p_{k+1} \in \mathrm{span}\{p_1, ..., p_k, e_{i_{k+1}}\}$ *conjugate to* $p_1, ..., p_k$.

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

If $\epsilon = 0$, the solution obtained by the above algorithm is a local solution of problem (2.34)-(2.35). Let $S(r, A, b)$ be the set of all global solutions of (2.34)-(2.35), we are interested in studying what conditions would imply $x_{r+1} \in S(r, A, b)$. If $A = I$, it is easily to see that $x_{r+1} \in S(r, I, b)$. For general A, if $r = 1$ or 2 we have the following results.

Lemma 2.5. *Let $A = (a_1, ..., a_n)$. If $||a_i|| = 1$ for all i, the iterate point x_{k+1} obtained by the SSCS algorithm has the following properties:*

(1) $x_2 \in S(1, A, b)$*;*

(2) *There exists a $y \in S(2, A, b)$ such that x_3 and y share one non-zero element index.*

The subproblems in the SSCS algorithm have the form

$$\min_{x \in \Re^n} \quad ||Ax - b||_2^2 \tag{2.36}$$

$$\text{s. t.} \quad x_i = 0\,, \quad i \in I_k \tag{2.37}$$

for some active set I_k. Thus, general subspaces spanned by coordinate directions for sparsity constraint problems should have the form $\mathcal{S}_k = \{d \,|\, d_i = 0, i \in I_k\}$. Such subspaces are used by many methods for compressive sensing. One particular optimization model in compressive sensing is the l_0 minimization problem

$$\min_{x \in \Re^n} \quad ||x||_0 \tag{2.38}$$

$$\text{s. t.} \quad Ax = b\,. \tag{2.39}$$

For more detailed discussions, please refer to [38] and the references given there.

Another possible subspace is the *steepest descent $\tau-$subspace*, which is a τ dimensional subspace which forces τ elements of the gradient vector to be zero. Instead of requiring the whole vector $g(x) = 0$, which is the optimality condition for $\min f(x)$, we require τ elements of $g(x)$ to be zero, namely

$$\bar{g}(x) = ((g(x))_{i_1}, (g(x))_{i_2}, ..., (g(x))_{i_\tau})^T = 0\,,$$

at the current iteration. This should be achievable by searching in a subspace spanned by τ coordinate directions, since there are only τ equations. Let the Jacobian of $\bar{g}(x)$ to be $\bar{A}(x)$, a Newton's step d satisfies

$$(\bar{A}(x_k))^T d + \bar{g}(x_k) = 0\,. \tag{2.40}$$

Because the above system has τ equations with n unknowns, it is possible to consider d in any subspace spanned by τ coordinate directions. There are C_n^m such choices, and we call the one which makes the length of the solution of (2.40) in the subspace the shortest as the steepest descent τ-subspace. Intuitively, this subspace has the nice property of forcing τ elements of the gradient vector to zero by moving a $\tau-$coordinate step as small as possible. However, such a definition of the subspace seems to be too theoretical and may not be easy to be implemented in practice, as it needs to solve linear least squares problem with linear constraints and a sparsity constraint:

$$\begin{aligned} \min_{d \in \Re^n} \quad & ||d||_2^2 \\ \text{s. t.} \quad & (\bar{A}(x_k))^T d + \bar{g}(x_k) = 0, \quad ||d||_0 = \tau\,. \end{aligned}$$

3. Subspace techniques for constrained optimization

Now we consider subspace techniques for constrained optimization. In order to simplify the presentation, instead of considering the general problem (1.1)-(1.3), we focus on the equality

constrained problem:

$$\min_{x\in\Re^n} \quad f(x) \tag{3.1}$$

$$\text{s. t.} \quad c(x) = 0\,, \tag{3.2}$$

where $c(x) = (c_1(x), \cdots, c_m(x))^T$.

The sequential quadratic programming method (SQP) is an important numerical method for solving constrained optimization. The main idea of the SQP method is to solve the nonlinearly constrained problem (3.1)-(3.2) by successively minimizing quadratic approximations to the Lagrangian function subject to the linearized constraints. The search direction d_k of a line search type SQP method is obtained by solving the following quadratic programming subproblem

$$\min_{d\in\Re^n} \quad Q_k(d) = g_k^T d + \frac{1}{2} d^T B_k d \tag{3.3}$$

$$\text{s. t.} \quad c(x_k) + A_k^T d = 0\,, \tag{3.4}$$

where $A_k = \nabla c(x_k)$ and B_k is an approximation to the Hessian of the Lagrangian function. The SQP step d_k can be decomposed into two parts $d_k = h_k + v_k$ where $v_k \in \text{range}(A_k)$ and $h_k \in \text{null}(A_k^T)$. Thus, v_k is a solution of the linearized constrained constraints (3.4) in the range space of A_k, while h_k is the minimizer of the quadratic function $Q_k(v_k + d)$ in the null space of A_k^T.

One good property of the SQP method is that it converges superlinearly, namely when x_k is close to a KKT point x^* we have the following relation

$$x_k + d_k - x^* = o(\|x_k - x^*\|)\,. \tag{3.5}$$

But, the superlinearly convergent step d_k may lead to a point that seems "bad" as it may increase both the objective function and the constraint violations. The famous Marotos effect shows that it is possible for the SQP step d_k to have both $f(x_k + d_k) > f(x_k)$ and $\|c(x_k + d_k)\| > \|c(x_k)\|$, even though (3.5) holds. A remedy for overcoming the Marotos effect is the second order correction step method[12, 26], where the step is obtained by resolving the quadratic programming subproblem with the constraints (3.4) are replaced by

$$c(x_k + d_k) + A_k^T(d - d_k) = 0 \tag{3.6}$$

because the left hand side of (3.6) is a better approximation to $c(x_k + d)$ near the point $d = d_k$. Since the change of the constraints is a second order term, the new step can be viewed as the SQP step d_k adding a second order correction step $\hat{d}_k$. For detailed discussions on the SQP method and the second order correction step, please see [32].

Now, let us examine the second order correction step from subspace point of views. The second order correction step $\hat{d}_k$ is a solution of

$$\min_{d\in\Re^n} \quad Q_k(d_k + d) \tag{3.7}$$

$$\text{s. t.} \quad c(x_k + d_k) + A_k^T d = 0\,. \tag{3.8}$$

Assume that the QR factorization of A_k is $[Y_k, Z_k]\begin{bmatrix} R_k \\ 0 \end{bmatrix}$ and R_k is nonsingular. Thus, the second order correction step is represented as $\hat{d}_k = \hat{v}_k + \hat{h}_k$, where $\hat{v}_k = -Y_k R_k^{-T} c(x_k + d_k)$

and $\hat{h}_k$ is the minimizer of

$$\min_{h \in \mathbf{null}(A_k^T)} \quad Q(d_k + \hat{v}_k + h)\,. \tag{3.9}$$

Since d_k is the SQP step, it follows that $g_k + B_k d_k \in \text{range}(A_k)$, which implies that the minimization problem (3.9) is equivalent to

$$\min_{h \in \mathbf{null}(A_k^T)} \quad \frac{1}{2}(\hat{v}_k + h)^T B_k (\hat{v}_k + h)\,. \tag{3.10}$$

If $Y_k^T B_k Z_k = 0$, we have that $\hat{h}_k = 0$, which shows that the second order correction step $\hat{d}_k \in \text{range}(A_k)$ is also a range space step. In this case, the second order correction uses two range space steps and one null space step. This is an undesirable property because a range space step is a fast convergent step as it is a Newton's step while a null space step is normally a slower convergent step due to the fact that it is normally a quasi-Newton step because B_k is generally a quasi-Newton approximation to the Hessian of the Lagrangian function. Hence, examining the SQP method with subspace properties helps us to understand the insights of the method. Intuitively, it would be more reasonable to have two steps in the slower space with one step in the fast space. Thus, it might be better to investigate a modified SQP method with a correction step $\hat{d}_k \in \text{null}(A_k^T)$.

We can also consider subspaces other than the null space and the range space. In general, a subspace SQP method obtains the search direction d_k by solving a QP in a subspace:

$$\min_{d \in \Re^n} \quad Q_k(d) \tag{3.11}$$

$$\text{s. t.} \quad c_k + A_k^T d = 0, \qquad d \in \mathcal{S}_k\,, \tag{3.12}$$

where $\mathcal{S}_k$ is a subspace. Lee[20] considered the following choice:

$$\mathcal{S}_k = \text{span}\{-g_k, d_1, ..., d_{k-1}, -\nabla c_{k_i}\}\,,$$

where $|c_{k_i}| = \|c_k\|_\infty$.

In some trust region algorithms for constrained optimization, the subproblem that needs to be solved in each iteration is the Celis-Dennis-Tapia subproblem[7]

$$\min_{d \in \Re^n} \quad Q_k(d) = g_k^T d + \frac{1}{2} d^T B_k d \tag{3.13}$$

$$\text{s. t.} \quad ||c_k + A_k^T d||_2 \le \xi_k, \quad ||d||_2 \le \Delta_k\,. \tag{3.14}$$

Recently, It is shown that the CDT subproblem has certain subspace properties[18]:

Lemma 3.1 ([18]). *Let $S_k = span\{Z_k\}$, $Z_k^T Z_k = I$, $span\{A_k, g_k\} \subset S_k$ and $B_k u = \sigma u$, $\forall u \in S_k^\perp$. Then the CDT subproblem is equivalent to*

$$\min_{\bar{d} \in \Re^r} \quad \bar{Q}_k(\bar{d}) = \bar{g}_k^T \bar{d} + \frac{1}{2} \bar{d}^T \bar{B}_k \bar{d} \tag{3.15}$$

$$s.\ t. \quad ||c_k + \bar{A}_k^T \bar{d}||_2 \le \xi_k, \quad ||\bar{d}||_2 \le \Delta_k\,, \tag{3.16}$$

where $\bar{g}_k = Z_k^T g_k$, $\bar{B}_k = Z_k^T B_k Z_k$ and $\bar{A}_k = Z_k^T A_k$.

Based on the above result, a subspace version of the Powell-Yuan trust algorithm[28] was given in [18].

Subspace techniques can also be used with other methods for constrained optimization. For example, interior methods for nonlinearly constrained optimization basically use a Newton's step to the KKT system based on the log-barrier function. If we solve the derived linear system in a lower dimensional subspace, it will give us a subspace version of an interior point method.

There are many subspace techniques for bound-constrained problems, where the constraints are

$$l \leq x \leq u\,, \tag{3.17}$$

where l and u are two given vectors in $\Re^n$. For example, A subspace adaptation of the Coleman-Li trust region and interior method[8] is proposed for solving large-scale bound-constrained minimization problems[3], and another subspace version of the Coleman-Li trust region algorithm was presented in [41]. Ni and Yuan[27] proposes a subspace limited memory quasi-Newton method for solving large-scale optimization with bound constraints (3.17), in which the limited memory quasi-Newton method is used to update the variables with indices outside of the active set, while the projected gradient method is used to update the active variables.

4. Subspace techniques for nonlinear equations and nonlinear least squares

In this subsection, we consider systems of nonlinear equations

$$F_i(x) = 0, \quad i = 1, ..., m; \quad x \in \Re^n\,, \tag{4.1}$$

and nonlinear least squares:

$$\min_{x\in\Re^n} \sum_{i=1}^{m} (F_i(x))^2\,. \tag{4.2}$$

Because nonlinear least squares problem (4.2) is a special unconstrained optimization problem, all the subspace techniques discussed in Section 2 can be applied. Due to the special structures of nonlinear equations and nonlinear least squares, there are special subspace approaches. For example, several implementations of Newton-like iteration schemes based on Krylov subspace projection methods for solving nonlinear equations are considered in [4]. The Gauss-Seidel iteration for linear equations can be extended for nonlinear equations. In the following, we will discuss some possible subspace approaches including incomplete sum, partition of variables, and steepest descent $\tau-$subspace.

First, we explain the technique of incomplete sum for nonlinear least squares. At iteration k, we minimize the sum of squares of some selected terms instead of all terms. Namely, define an index set J_k which is a subset of $\{1, ..., m\}$, and consider

$$\min_{x\in\Re^n} \sum_{i\in J_k} (F_i(x))^2\,. \tag{4.3}$$

The incomplete sum approach works quite well for certain class of problems, for example the distance geometry problem which has lots of applications including protein structure prediction, where the nonlinear least squares of all the terms would have lots of local minimizers[30].

For nonlinear equations, the incomplete approach is to ignore some equations. Instead of requiring the original system (4.1), we consider

$$F_i(x) = 0, \quad i \in J_k\,, \tag{4.4}$$

which is an incomplete set of equations. It is easy to see the incomplete approach is a subspace technique. Define the vector

$$F = \begin{pmatrix} F_1(x) \\ F_2(x) \\ \vdots \\ F_m(x) \end{pmatrix} \in \Re^m\,.$$

To solve the nonlinear equations (4.1) is to find a x at which F maps to the origin. Let P_k^T be a mapping from R^m to a lower dimensional subspace, solving the reduced system

$$P_k^T F(x) = 0 \tag{4.5}$$

is exactly replacing $F = 0$ by requiring its mapping to the subspace spanned by P_k to be zero. In particular, if the columns of P_k are chosen to be coordinate vectors $\{e_i, i \in J_k\}$, we obtain the incomplete set of equations (4.4).

Now, we consider partition of variables, which is clearly a subspace technique. Let I_k be a subset of $\{1, ..., n\}$. We partition the variables into two group $x = (\bar{x}\,, \hat{x})$, where $\bar{x} = \{x_i, i \in I_k\}$ and $\hat{x} = \{x_i\,,\ i \notin I_k\}$. At the $k-$th iteration, we fix the variables $\hat{x}$ and allow $\bar{x}$ to change in order to obtain a better iterate point. To be exact, we try to solve

$$\min_{\bar{x} \in \Re^{|I_k|}} \sum_{i=1}^{m} (F_i(\bar{x}, \hat{x}_k))^2\,. \tag{4.6}$$

The above problem has fewer variables. It is easy to see that partition of variables use special subspaces that spanned by coordinate directions. An obvious generalization of partition of variables is decomposition of the space which uses subspaces spanned by any given directions. For example, assume that we have i_k vectors $\{q_1^{(k)}, q_2^{(k)}, ..., q_{i_k}^{(k)}\}$ which spans $\mathcal{S}_k$. Similar to (4.6), we consider the subspace subproblem

$$\min_{d \in \mathcal{S}_k} \sum_{i=1}^{m} (F_i(x_k + d))^2\,. \tag{4.7}$$

When the above subproblem is combined with the reduced system technique, it gives the general subspace subproblem for nonlinear least squares

$$\min_{d \in \mathcal{S}_k} ||P_k^T F(x_k + d)||_2^2\,. \tag{4.8}$$

For nonlinear equations, a similar subproblem is

$$P_k^T F(x_k + Q_k z) = 0, \tag{4.9}$$

where $Q_k = [\ q_1^{(k)},\ q_2^{(k)},\ ...,\ q_{i_k}^{(k)}\ ]$ and $P_k = [\ p_1^{(k)},\ p_2^{(k)},\ ...,\ p_{i_k}^{(k)}\ ]$. Let J_k be the Jacobian of F at x_k, the linearized system for subproblem (4.9) is

$$P_k^T [\ F(x_k) + J_k Q_k z\ ] = 0\,. \tag{4.10}$$

Of course, the efficiency of such an approach depends on how to select P_k and Q_k. We can borrow ideas from subspace techniques for large scale linear systems[29]. Instead of using (4.10), we construct a subproblem of the following form:

$$P_k^T F(x_k) + \hat{J}_k z = 0\,, \tag{4.11}$$

where $\hat{J}_k \in \Re^{i_k \times i_k}$ is an approximation to $P_k^T J_k Q_k$. The reason for preferring (4.11) over (4.10) is that in (4.11) we do not need the Jacobian matrix J_k, whose size is normally significantly larger than that of $\hat{J}_k$.

The $\tau-$steepest descent coordinate subspace discussed in Section 2 can also be extended to nonlinear equations and nonlinear least squares. Here we only discuss nonlinear equations. Assume we have

$$|F_{i_1}(x_k)| > \cdots > |F_{i_\tau}(x_k)| > \cdots \tag{4.12}$$

at the $k-$th iteration. A direct extension of the $\tau-$ steepest descent coordinate subspace method discussed in Section 2 would solve

$$F_{i_j}(x_k) + d^T \nabla F_{i_j}(x_k) = 0 \quad j = 1, ..., \tau\,. \tag{4.13}$$

in the subspace spanned by the corresponding coordinate directions $\{e_{i_j}, \quad j = 1, ..., \tau\}$. This approach is reasonable if $F(x)$ is a monotone operator. For general nonlinear functions $F(x)$, it seems that we should replace e_{i_j} by the coordinate direction which is the steepest descent coordinate direction of the function $F_{i_j}(x)$ at x_k. Namely, we should replace i_j by an index l_j such that

$$l_j = \text{argmax}_{t=1,...,n} \left| \frac{\partial F_{i_j}(x_k)}{\partial (x)_t} \right| .$$

However, such a choice may lead to one l_j for two different j, which makes subproblem (4.13) has no solution in the subspace spanned by $\{e_{l_1}, ..., e_{l_\tau}\}$.

A good subspace spanned by $\tau-$ coordinate directions might be the steepest descent $\tau-$subspace as discussed in Section 2, which should contain the shortest vector d from all solutions of (4.13) satisfying $||d||_0 = \tau$. However, such a subspace is not easy to obtain, an approximation could be derived by finding τ row indices of the matrix $[\nabla F_{i_1}(x_k), ..., \nabla F_{i_\tau}(x_k)]$ such that the corresponding $\tau \times \tau$ sub-matrix Γ_k makes $||(\Gamma_k)^{-1}||$ as small as possible.

More detailed discussions on subspace methods for nonlinear equations and nonlinear least squares are given in [42].

5. Subspace techniques for matrix optimization

Matrix optimization problems have stimulated lots of researches in recent years due to their broad applications. The one million dollar Netflix prize problem[1] may be formulated as the following problem

$$\min_{X \in \Re^{n \times m}} \quad \text{rank}(X) \tag{5.1}$$

$$\text{s. t.} \quad (X)_{ij} = M_{ij}\,, \quad (i,j) \in \mathcal{T}\,, \tag{5.2}$$

where $\mathcal{T}$ is a subset of $\{(i,j) \mid i = 1, ..., n;\ j = 1, ..., m\}$, and $M_{ij}((i,j) \in \mathcal{T})$ are given data. A second example of matrix optimization problem is the semidefinite programming

problem

$$\min_{X\in\Re^{n\times n}} \quad \langle C\,,\,X\rangle \tag{5.3}$$

$$\text{s. t.} \quad \langle A_i,\,X\rangle \geq b_i, \quad i=1,...,m, \tag{5.4}$$

$$X \succeq 0, \tag{5.5}$$

where $\langle X\,,\,Y\rangle = trace(X^TY)$. Another example is solving the Kohn-Sham equation in density functional theory from physics and quantum chemistry, where the total energy of a system needs to be minimized. This leads to the minimization of a nonlinear matrix function with orthogonality constraints:

$$\min_{X\in\Re^{n\times m}} \quad E(X) \tag{5.6}$$

$$\text{s. t.} \quad X^TX = I\,, \tag{5.7}$$

where $E(X)$ is the energy function [22, 36].

A general nonlinear matrix optimization has the following form

$$\min_{X\in\mathcal{X}} \quad f(X) \tag{5.8}$$

$$\text{s. t.} \quad c(X) = 0, \tag{5.9}$$

where $f:\Re^{n\times m}\to\Re$, $\mathcal{X}\subseteq\Re^{n\times m}$ and $c:\Re^{n\times m}\to\Re^p$. The constraints have been split into the set $\mathcal{X}$ and the general constraints $c(X)=0$ according to their structures and roles in the targeted subspace subproblems. For example, some simple constraints such as orthogonality and positive semidefiniteness can be put in $\mathcal{X}$ and the subspace subproblems still have a computable closed form solution. Specifically, for a suitably chosen subspace $\mathcal{S}_k\subset\Re^{n\times m}$, $m_k(X)\approx f(X)$ and an linear operator $\mathcal{A}_k$ such that $\mathcal{A}_k(X)\approx c(X)$ for $X\in\mathcal{S}_k$, the subspace subproblem is:

$$\min_{X\in\mathcal{S}_k\cap\mathcal{X}} \quad m_k(X) \tag{5.10}$$

$$\text{s. t.} \quad \mathcal{A}_k(X) = 0. \tag{5.11}$$

Then, a model subspace algorithm for the general matrix optimization problem (5.8)-(5.9) can be given as follows.

Algorithm 5.1. (Model subspace method for nonlinear matrix optimization)

Step 1 *Given X_1. Let $k:=1$.*

Step 2 *If X_k is a stationary point of (5.8)-(5.9) then stop.*
Choose a low-dimensional subspace $\mathcal{S}_k\subset\Re^{n\times m}$,
build an approximate model $m_k(X)\approx f(X)$ for $X\in\mathcal{S}_k$, and an linear operator $\mathcal{A}_k$ such that $\mathcal{A}_k(X)\approx c(X)$ for $X\in\mathcal{S}_k$.

Step 3 *Solve* (5.10)-(5.11) *to obtain $\hat{X}$.*

Step 4 *Choose a suitable map $h(X):\Re^{n\times m}\to\Re^{n\times m}$ to construct the next iteration: $X_{k+1}:=h(\hat{X}_k)$; Go to Step 2.*

Most of the techniques for choosing subspaces in subsection 2.1 can be extended here. For example, we can choose the subspace mainly spanned by the gradients at the first k iterations:

$$\mathcal{S}_k = \text{span}\{X_k, \nabla f(X_1), \cdots, \nabla f(X_k)\}, \tag{5.12}$$

or use the conjugate gradient type subspace

$$\mathcal{S}_k = \text{span}\{\nabla f(x_k), X_k, X_{k-1}\}. \tag{5.13}$$

There are various ways for defining subspaces when the matrix optimization problems have special structures. For example, for the low rank matrix optimization problems we can search in subspaces of low dimensional manifolds of low rank matrices. In particular, consider the following problem

$$\min_{X \in \Re^{n\times p}} \quad ||\mathcal{A}(X) - b||_2^2 \tag{5.14}$$

$$\text{s. t.} \quad \text{rank}(X) \le r. \tag{5.15}$$

One special subspace is

$$\mathcal{S}_k = \{X_k + Y \,|\, \text{rank}(Y) \le \tau\}. \tag{5.16}$$

If $\tau = 1$, we update the iterate matrix with the increment being a rank-1 matrix.

Computing the dominate singular value decomposition of a given matrix $A \in \Re^{n\times m}$ leads to a matrix optimization problem with orthogonality constraints:

$$\max_{X \in \Re^{n\times p}} \quad ||A^T X||_F^2 \tag{5.17}$$

$$\text{s. t.} \quad X^T X = I. \tag{5.18}$$

Let $\mathcal{X} = \{X \in \Re^{n\times p} \mid X^T X = I\}$ and $c(X) = \emptyset$. The locally optimal block preconditioned conjugate gradient method (LOBPCG) [19] chooses $h(\cdot)$ as the identity map and the following conjugate gradient type of subspace:

$$\mathcal{S}_k = \text{span}\{X_{k-1}, X_k, AA^T X_k\}, \tag{5.19}$$

The corresponding subspace problem is a $3p$-dimensional generalized eigenvalue problem which can be solved fast due to the fact that $p \ll \min\{n, m\}$. The limited memory block Krylov subspace optimization method (LMSVD, [23]) selects the subspace

$$\mathcal{S}_k = \text{span}\{X_k, X_{k-1}, ..., X_{k-q}\} \tag{5.20}$$

with an adaptive way to adjust the size of $\mathcal{S}_k$ and takes

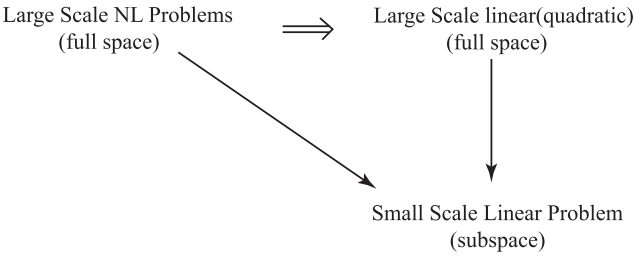
$$h(X) := \text{orth}(AA^T X), \tag{5.21}$$

which reduces the probability to be trapped by saddle points of (5.17)-(5.18). A general global convergence analysis for both LOBPCG and LMSVD is established in [23] by requiring some minimal assumptions.

6. Summary

In this paper, we review subspace techniques for nonlinear optimization. Compared to full space algorithms which normally convert nonlinear problems to linear/quadratic systems without reducing the size of the problem, subspace algorithms aim to to take a short-cut from large scale nonlinear problem to small scale linear/quadratic systems. This is illustrated by the following diagram:

Full-space via Sub-space



Subspace techniques are suitable for problems where function values are difficult to compute and problems that are highly nonlinear for which normally line searches are very expensive. Though we have given quite a few suggestions on how to choose subspaces, there are still many issues to be investigated further, including how to balance between null space and range space for constrained optimization for null-space type methods and how to choose subspaces depending on constraints for general subspace methods for constrained optimization.

The subspace techniques discussed in the paper show that large scale problems can be approximated by lower dimensional subspace subproblems, and we believe that the nice properties of subspace techniques will enable them to play an important role in the development of numerical methods for large scale optimization.

Acknowledgements. The author's research is partially supported by NSFC grants 11331012 and 11321061. I am very grateful to my former students Xin Liu, Zaiwen Wen and Cong Sun for their helpful comments on the first draft of the paper. I also thank my daughter Yuan Yuan for her polishing the English of the paper.

References

[1] J. Bennett and S. Lanning, *The Netflix prize*, Proceedings of KDD Cup and Workshop, 2007.

[2] D.P. Bertsekas. *Nonlinear Programming*, Athena Scientific, Belmont, Massachusetts, 1999.

[3] M.A. Branch, T.F. Coleman, and Y.Y. Li, *A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems*, SIAM J. Sci. Comput. **21** (2007), 1–23.

[4] P. Brown and Y. Saad, *Hybrid Krylov methods for nonlinear systems of equations*, SIAM Journal on Scientific and Statistical Computing **11** (1990), 450–481.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Foundations and Trends in Machine Learning **3** (2011), 1–122

[6] O. Burdakov, L.J. Gong, Y.X. Yuan, and S. Zikrin, *On efficiently combining limited memory and trust-region techniques*, Report, AMSS, CAS, 2013.

[7] Celis, M.R., Dennis, J.E., and Tapia, R.A., *A trust region algorithm for nonlinear equality constrained optimization*, in: P.T. Boggs, R.H. Byrd, and R.B. Schnabel, eds., Numerical Optimization (SIAM, Philadelphia, 1985), 71–82.

[8] T. Coleman and Y.Y. Li, *An interior point trust region approach for nonlinear minimization subject to bounds*, SIAM J. Optimization **6** (1996), 418–445.

[9] A. Conn, N. Gould, A. Sartenaer, and Ph. Toint, *On iterated-subspace methods for nonlinear optimization*, in: J. Adams and J.L. Nazareth, eds., Linear and Nonlinear Conjugate Gradient-Related Methods (1996), 50–79.

[10] Y. H. Dai and Y.X. Yuan, *Nonlinear Conjugate Gradient Methods* (in Chinese), Shanghai Science and Technology Publisher, Shanghai, 2000.

[11] M. Elad, B. Matalon, and M. Zibulevsky, *Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization*, Appl. Comput. Harmon. Anal. **23** (2007) 346–367

[12] R. Fletcher, *Second order correction for nondifferentiable optimization*, in: G.A. Watson, ed., Numerical Analysis, Springer-Verlag, Berlin, (1982), 85–115.

[13] P.E. Gill and M.W. Leonard, *Reduced-Hessian quasi-Newton methods for unconstrained optimization*, SIAM J. Optim. **1** (2001), 209–237.

[14] R. Glowinski and A. Marrocco, *Sur l'approximation, par éléements finis d'ordre un, et la résolution, par pénalisation-dualité, d'une classe de problèmes de Dirichlet non linéaires*, Laboria, 1975.

[15] M. Fortin and R. Glowinski, *Augmented Lagrangian Methods*, Vol. 15 of Studies in Mathematics and its Applications, North-Holland Publishing Co., Amsterdam, 1983. Applications to the numerical solution of boundary value problems, Translated from the French by B. Hunt and D. C. Spicer.

[16] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-splitting Methods in Nonlinear Mechanics*, Vol. 9 of SIAM Studies in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1989.

[17] N.I.M. Gould, D. Orban, and Ph.L. Toint, *Numerical methods for large-scale nonlinear optimization*, Acta Numerica (2005), 299–361.

[18] Grapiglia, G.N., Yuan, J.Y., and Yuan, Y.X., *A subspace version of the Powell-Yuan trust-region algorithm for equality constrained optimization*, J. Operations Research Society of China **1** (2013), 425– 451.

[19] A. V. Knyazev, *Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput. **23** (2001), 517–541.

[20] Lee, J.H., *A Subspace Algorithm for Nonlinear Equality Constrained Optimization*, Ph.D. thesis, ICMSEC, AMSS, Chinese Academy of Science, Beijing, 2009.

[21] D.C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale optimization*, Mathematical Programming **45** (1989), 503–528.

[22] X. Liu, X. Wang, Z.W. Wen, and Y.X. Yuan, *On the convergence of the self-consistent field iteration in Kohn-Sham density function theory*, accepted by SIAM Journal on Matrix Analysis and Applications (2014).

[23] X. Liu, Z.W. Wen and Y. Zhang, *Limited memory block Krylov subspace optimization for computing dominant singular value decompositions*, SIAM Journal on Scientific Computing **35** (2013), A1641-A1668

[24] Y.Y. Li and S. Osher, *Coordinate descent optimization for ℓ^1 minimization with application to compressed sensing; a greedy algorithm*, Inverse Probl. Imaging **3** (2009), 487–503

[25] Z. Q. Luo and P. Tseng, *On the convergence of the coordinate descent method for convex differentiable minimization*, J. Optim. Theory Appl. **72** (1992), 7–35

[26] D.Q. Mayne and E. Polak, *A superlinearly convergent algorithm for constrained optimization problems*, Math. Prog. Study **16** (1982), 45–61.

[27] Q. Ni and Y.X. Yuan, *A subspace limited memory quasi-Newton algorithm for large-scale nonlinear bound constrained optimization*, Mathematics of Computations **66** (1997), 1509–1520.

[28] M.J.D. Powell and Y.X. Yuan, *A trust region algorithm for equality constrained optimization*, Mathematical Programming **49** (1991), 189–211.

[29] Y. Saad, *Iterative Methods for Sparse Linear Systems: 2nd Ed.*, SIAM, Philedalphia, 2003.

[30] A. Sit, Z.J. Wu, and Y.X. Yuan, *A geometric buildup algorithm for the solution of the distance geometry problem using least-squares approximation*, Bulletin of Mathematical Biology **71** (2009), 1914–1933.

[31] J. Stoer and Y.X. Yuan, *A subspace study on conjugate gradient algorithms*, ZAMM Z. angew. Math. Mech. **75** (1995), 69–77.

[32] W.Y. Sun and Y.X. Yuan, *Optimization Theory and Methods: Nonlinear Programming*, Springer Verlag, New York, 2006.

[33] Z.H. Wang, Z.W. Wen, and Y.X. Yuan, *A subspace trust region method for large scale unconstrained optimization*, in: Y.X. Yuan eds. Numerical Linear Algebra and Optimization (Science Press, Beijing/NewYork, 2004), pp. 265–274.

[34] Z.H. Wang and Y.X. Yuan, *A subspace implementation of quasi-Newton trust region methods for unconstrained optimization*, Numerische Mathematik **104** (2006), 241–269.

[35] Z.W. Wen, D. Goldfarb, and K. Scheinberg, *Block coordinate descent methods for semidefinite programming*, in: Handbook on Semidefinite, Conic and Polynomial Optimization International Series in Operations Research & Management Science **166** (2012), 533–564.

[36] Z.W. Wen, A. Milzarek, M. Ulbrich, and H.C. Zhang, *Adaptive regularized self-consistent field iteration with exact Hessian for electronic structure calculation*, SIAM Journal on Scientific Computing **35** (2013), pp. A1299–A1324.

[37] Z.W. Wen, D. Goldfarb, and W. Yin, *Alternating direction augmented Lagrangian methods for semidefinite programming*, Mathematical Programming Computation **2** (2010), 203–230

[38] Z.W. Wen, W. Yin, D. Goldfarb, and Y. Zhang, *A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation*, SIAM Journal on Scientific Computing **32** (2010), 1832–1857.

[39] X.C. Tai and J.C. Xu, *Global and uniform convergence of subspace correction methods for some convex optimization problems*, Math. Comp. **71** (2002), 105–124.

[40] J. Yang and Y. Zhang, *Alternating direction algorithms for l_1-problems in compressive sensing*, SIAM journal on scientific computing **33** (2011), 250–278.

[41] Y.X. 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) (Higher Education Press. Beijing, 2007), pp. 206–218.

[42] ———, *Subspace methods for large scale nonlinear equations and nonlinear least squares*, Optimization and Engineering **10** (2009), 207–218.

Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Zhong Guan Cun Donglu 55, Beijing 100190, China

E-mail: yyx@lsec.cc.ac.cn