Journal of Computational Mathematics, Vol.18, No.1, 2000,1–12.

SOLVING TRUST REGION PROBLEM IN LARGE SCALE OPTIMIZATION*1)

Bing-sheng He
(*Department of Mathematics, Nanjing University, Nanjing 210093, China*)

Abstract

This paper presents a new method for solving the basic problem in the "model-trust region" approach to large scale minimization: Compute a vector x such that $\frac{1}{2}x^THx + c^Tx = \min$, subject to the constraint $\|x\|_2 \le a$. The method is a combination of the CG method and a projection and contraction (PC) method. The first (CG) method with $x_0 = 0$ as the start point either directly offers a solution of the problem, or—as soon as the norm of the iterate greater than a, —it gives a suitable starting point and a favourable choice of a crucial scaling parameter in the second (PC) method. Some numerical examples are given, which indicate that the method is applicable.

Key words: Trust region problem, Conjugate gradient method, Projection and contraction method.

1. Introduction

Let H be a given $n \times n$ symmetric positive semidefinite matrix and $c \in R^n$. In this paper we consider the following quadratic programming with a simple quadratical constraint

$$\begin{aligned} \frac{1}{2}x^THx + c^Tx &= \min \\ \text{s.t} \quad \|x\|_2 &\le a, \end{aligned} \tag{1}$$

where the parameter a is prescribed. This problem occurs frequently in *trust region* method for unconstrained optimization [1]. A number of approaches for solving (1) have been proposed in the literature [2,3,6,12–16]. One technique is to approximate a Lagrange multiplier λ by Newton's method. The approximation of this parameter may be quite delicate, however, and involves the computation of a sequence of singular value decompositions [5]. Since the SVD is too costly for large matrices, the method is applicable only for small problems. The advanced interior point methods could also be used to solve problem (1) and seem attractive, because one can show that these methods converge at a polynomial rate, see e.g. [11]; however, each iteration of an interior point method has to solve a system of linear equations and therefore is rather expensive.

For large and sparse problems, Golub and von Matt [6] presented a method, which uses a series incomplete decompositions and yields a sequence of upper and lower bounds on the Lagrange multiplier and enables them to compute an approximate solution x from these bounds via solving a system of linear equations.

---

* Received February 12, 1996.

1) This work was supported by the National natural Science Foundation of P.R. China and the Natural Science Foundation of Province Jiangsu of China.

Recently, we proposed some projection and contraction (PC) methods [7,8,9] for solving a class of linear variational inequalities, which can be applied to solve problem (1). One of the PC method is rather inexpensive and very simple to realize, because each iteration of the method consists essentially of a matrix vector product and the method does not need to solve any systems of linear equations. However, the performance of the PC method strongly depends on the scaling of H and c.

In this paper, we show—in theory as well as for some numerical examples—that the Lagrange multiplier must be small if the PC method converges slowly. We will pay particular attention to the problem of ill-conditioning and propose an alternative simple (CG-PC) method for solving problem (1). The method start with $x^0 = 0$, if $a \geq \|H^+c\|$, the CG method solves the problems, otherwise, as soon as a $\|x^k\| > a$, we get the information about a proper scaling parameter for H and c, and switch to use some simple projection and contraction (PC) methods [9]. For large trust region problems, the presented method is as simple as the Goldstein's fundamental projection method [4], in addition, as the numerical results will show, it is almost as powerful as the method proposed by Golub and von Matt [6].

1.1. Outline and notation

The equivalent linear projection equation is given in Section 2. In Section 3 we briefly quote some convergence facts of the CG method [10] and the PC method [7,8,9]. In Section 4 we study the convergence behaviour of the PC method and analyze how to treat ill-conditioned problems. Further details of our method are given in Section 5. In Section 6 we present some numerical results.

We use the following notations. A superscript such as in x^k refers to specific vectors and k usually denotes the iteration index. By $\|v\|$ we denote the Euclidean norm of some vector v, by $\|v\|_G$ and the norm $(v^TGv)^{1/2}$ induced by a positive definite matrix G, and by $\|H\|$ we denote the spectral norm $\lambda_{\max}(H)$ of some symmetric matrix H. H^+ denotes the pseudoinverse of a matrix H. Finally, by x^* we denote the solution of the problems.

1.2. Basic observations

An immediate observation regarding problem (1) tells us, if the dimensions of the matrix H are small and a singular value decomposition of H can be computed in moderate time, then for $a \geq \|H^+c\|$ the solution $x = H^+c$ can be computed from the singular value decomposition of H, and for $a < \|H^+c\|$ problem (1) is equivalent to finding a value $\lambda > 0$ such that

$$c^T(H + \lambda I)^{-2}c - a^2 = 0. \tag{2}$$

(Given such λ, the solution x is given by $x = -(H + \lambda I)^{-1}c$.) The derivative with respect to λ of the right hand side is $-2x^T(H+\lambda I)^{-3}x$, and is thus computable directly from the singular value decomposition of $H = V^T\Sigma V$ by observing that $(H + \lambda I)^{-1} = V^T(\Sigma + \mu I)^{-1}V$. Thus some modifications of Newton's method seem appropriate for solving (1), see e.g. [13]. In the following we will assume that H is large and sparse, and does not allow a singular value decomposition in moderate time.

2. The Equivalent Projection Equation

The Lagrange function of problem (1) is

$$L(x, \lambda) = x^THx + 2c^Tx + \lambda(x^Tx - a^2), \tag{3}$$

which is defined on $R^n \times R_+$. The Kuhn-Tucker Theorem of convex programming tells us that x^* is a solution of (1) if and only if there exists a $\lambda^* \geq 0$, such that (x^*, λ^*)

satisfies

$$Hx + c + \lambda x = 0 \tag{4}$$

and

$$\begin{cases} \lambda \geq 0, \quad x^T x \leq a^2 \\ \lambda \cdot (x^T x - a^2) = 0. \end{cases} \tag{5}$$

Let

$$\Omega = \{x \in R^n \mid \|x\|_2 \leq a\}$$

and $P_\Omega(\cdot)$ denote the projection on Ω, i.e.,

$$P_\Omega(x) = \begin{cases} x & \text{if } \|x\| \leq a \\ \dfrac{ax}{\|x\|} & \text{otherwise.} \end{cases}$$

Note that the numerical computation of P_Ω is extremely simple. Moreover, the feasibility and complementarity condition (5) (in the case $x \neq 0$) can be written as

$$x = P_\Omega(x + \lambda x). \tag{6}$$

Substituting (4) in (6), we get the following linear projection equation (abbreviated to LPE)

$$\text{(LPE)} \qquad x = P_\Omega[x - (Hx + c)]. \tag{7}$$

It is easy to verify that (7) is equivalent to (4) and (5). Thus, solving problem (1) is equivalent to finding a zero point of the residue function

$$e(x) := x - P_\Omega[x - (Hx + c)]. \tag{8}$$

Therefore, we can view x as an approximation to the solution of (1), if $\|e(x)\| \leq \epsilon$ for a given $\epsilon > 0$. Note however, that $e(x)$ is not invariant under multiplication of H and c by some positive scalar, but clearly problem (1) is so. For example, if H and c are multiplied by some small positive scalar such that $\|c\| \leq \epsilon$, the point $x = 0$ satisfies $\|e(x)\| \leq \epsilon$. Similarly, the quantity $\|e(x)\|/\|c\|$ is not a reliable stopping criterion either, as it tends to zero (for fixed x and Ω) as H and c are multiplied by large positive scalars. In Section 5 we therefore propose a modified stopping criterion. In the following we denote the solution set by X^*. Since Ω is compact, X^* is nonempty.

3. Review of the CG Method and the PC Method

In this section, we summarize some facts from the CG method and the PC method.

3.1. The CG method

The CG method [10] for minimizing a convex quadratic function is an exact line search method

$$x^{k+1} = x^k - \alpha_k s^k,$$

in which

$$s^0 = g^0, \qquad \beta_0 = 0,$$
$$s^{k+1} = g^{k+1} + \beta_k s^k,$$
$$\beta_k = \frac{{g^{k+1}}^T g^{k+1}}{{g^k}^T g^k},$$
$$\alpha_k = \frac{{g^k}^T g^k}{{s^k}^T H s^k}$$

and

$$g = Hx + c.$$

It is well known [10], that if H is symmetric and positive semidefinite, the CG method for solving $Hx + c = 0$ terminates after $m \le n$ steps, and the following hold for all i, $1 \le i \le m$,

$$\begin{aligned} {s^i}^T H s^j &= 0, \\ {g^i}^T g^j &= 0, \qquad\qquad j = 0, 1, \ldots, i-1 \\ {g^i}^T s^j &= 0. \end{aligned}$$

Let $\tilde{x}$ denote $H^+ c$, it was proved [10] that for $k \le m$

$$\|x^{k+1} - \tilde{x}\| < \|x^k - \tilde{x}\|.$$

In following we prove another property of the CG method, which is useful for the construction of the presented method.

Theorem 1. *If the CG method start with $x^0 = 0$, then for all $k < m$*

$$\|x^{k+1}\| > \|x^k\|$$

Proof. Using

$$x^k = -(\alpha_0 s^0 + \cdots + \alpha_{k-1} s^{k-1})$$

we have

$$\begin{aligned} \|x^{k+1}\|^2 &= \|x^k - \alpha_k s^k\|^2 \\ &= \|x^k\|^2 + 2\alpha_k {s^k}^T [\alpha_0 s^0 + \cdots + \alpha_{k-1} s^{k-1}] + \alpha_k^2 \|s^k\|^2. \end{aligned}$$

Note that for $j < k$,

$${s^j}^T s^k = {s^j}^T (g^k + \beta_{k-1} s^{k-1}) = \beta_{k-1} \cdots \beta_j {s^j}^T s^j.$$

Since for all i, $0 \le i \le m$, $\alpha_i > 0$ and $\beta_i \ge 0$, the assertion is proved.

3.2. The PC method

In [9] we suggested some basic projection and contraction methods (abbreviated PC methods) for solving linear projection equation (7). The search directions of the PC methods in [9] are constructed based on the residue $e(x)$ of the projection equation. The iterative scheme of these methods is

$$x^{k+1} = x^k - \rho(x^k) Q e(x^k), \tag{10}$$

with different matrices Q and steplength ρ. For trust region problem in large scale optimization, only the method without matrix factorization is in consideration, hence, we report a method of the steepest descent type that have been defined in [9].

In the PC methods of S-D type

$$Q = I, \quad \rho(x) = \frac{\|e(x)\|^2}{e(x)^T (I + H) e(x)}. \tag{11}$$

As was shown in [9], the method has the following contractive property.

Theorem 2. (Theorem 2 in [9]) *The sequence* $\{x^k\}$ *generated by PC method for* (LPE) *satisfies*

$$\begin{aligned}\|x^{k+1}-x^*\|_G^2 \le \|x^k-x^*\|_G^2 - \rho(x^k)\cdot\|e(x^k)\|^2 \\ - 2\rho(x^k)\cdot(x^k-x^*)H(x^k-x^*) \qquad \forall x^*\in X^*\end{aligned} \tag{12}$$

where

$$G = I + H. \tag{13}$$

Using Theorem 2, it is easy to prove the following convergence result.

Corollary 1. (Theorem 6 in [9]) *The sequence* $\{x^k\}$ *generated by PC method converges to a solution point* x^**. Moreover, if* H *is positive definite, then the method is globally linearly convergent.*

4. The Convergence Behaviour of the PC Method

Let (x^*,λ^*) be the solution of the Lagrange equation (4) which satisfies the feasibility and complementarity condition (5). In the case that $\|H^+c\| > a$, it is clear that $\|x^*\| = a$ and $\lambda^* > 0$. First, we give the following lemma.

Lemma 1. *Let* $\Omega = \{x\in R^n \mid \|x\|\le a\}$, $u,v\in R^n$, $a<\|v\|<\|u\|$, $(u-v)\perp v$ *and* $\|u-v\|<\|u\|-a$. *Then we have*

$$\|P_\Omega(u+z)-P_\Omega(u)\| \le \|P_\Omega(v)-P_\Omega(u)\|, \qquad \forall z\in R^n \text{ with } \|z\|\le\|u-v\| \tag{14}$$

and

$$\frac{\|P_\Omega(v)-P_\Omega(u)\|}{\|v-u\|} \le \frac{a}{\|v\|}. \tag{15}$$

Proof. The arguments follow immediately from the elementary geometry, and the details are omitted.

From (14) and (15) follows that

$$\|P_\Omega(u+z)-P_\Omega(u)\| \le \frac{a\|z\|}{\sqrt{\|u\|^2-\|z\|^2}} \quad \forall z\in R^n \ \text{ with } \ \|z\|\le\|u\|-a. \tag{16}$$

The following theorem shows that, if $\lambda^*>0$ is sufficiently large, then $e(x)$ is large relative to $\|x-x^*\|$ and thus by Theorem 2, the PC method converges fast.

Theorem 3. *Let* H *be positive semidefinite,* (x^*,λ^*) *be the solution of the Lagrange function with* $\|x^*\|=\alpha$ *and* $\lambda^*>0$. *If* $\|I-H\|\le 1+\delta\lambda^*$, *for some* $\delta\le 0.8$ *then*

$$\|e(x)\| \ge \frac{(4-5\delta)\lambda^*}{5+4\lambda^*}\|x-x^*\| \tag{17}$$

for all $x\in X := \{x\in R^n \mid \|(I-H)(x-x^*)\|\le 0.6\lambda^* a\}$.

Proof. First,

$$\begin{aligned}\|e(x)\| &= \|x-x^*-\{P_\Omega[x-(Hx+c)]-x^*\}\| \\ &\ge \|x-x^*\| - \|P_\Omega[x-(Hx+c)]-x^*\|.\end{aligned} \tag{18}$$

Using $x^*-(Hx^*+c)=(1+\lambda^*)x^*$ and $x^*=P_\Omega[(1+\lambda^*)x^*]$ we have

$$\|P_\Omega[x-(Hx+c)]-x^*\| = \|P_\Omega[(1+\lambda^*)x^*+(I-H)(x-x^*)]-P_\Omega[(1+\lambda^*)x^*]\|.$$

Under the assumptions $\|(I-H)(x-x^*)\| \le 0.6\lambda^* a$ and by using (16) follows that

$$\begin{aligned}&\|P_\Omega[(1+\lambda^*)x^* + (I-H)(x-x^*)] - P_\Omega[(1+\lambda^*)x^*]\| \\ &\le \frac{1}{\sqrt{(1+\lambda^*)^2 - (0.6\lambda^*)^2}} \cdot \|(I-H)(x-x^*)\|.\end{aligned}$$

Finally, using $\|I-H\| \le 1+\delta\lambda^*$ we get

$$\begin{aligned}\|P_\Omega[x-(Hx+c)] - x^*\| &\le \frac{1}{1+0.8\lambda^*} \cdot (1+\delta\lambda^*)\|x-x^*\| \\ &= \frac{5+5\delta\lambda^*}{5+4\lambda^*} \cdot \|x-x^*\|.\end{aligned} \tag{19}$$

Substituting (19) in (18) we get the assertion of this theorem.

On the application of the PC method of S-D type

In the method of steepest descent, each iteration essentially consists of one projection on Ω (when computing $e(x)$), and the computation of Hx and $He(x)$. These operations are extremely cheap. Nevertheless, Theorem 3 implies that, when $\lambda^* > 0$ is sufficiently large, even if H is ill-conditioned and the method of steepest descent is used, then locally the sequence $\{x^k\}$ converges to the solution x^* rapidly.

This result can even be globalized in the following sense. Notice that problem (1) and hence also (7) is invariant under multiplication of H and c by some positive constant. Hence we may assume that the norm $\|H\|$ of H is adjustable according to our needs, and the assumption of Theorem 3 can trivially be satisfied by scaling H such that $\|H\| \le 1$ which implies that $\|I-H\| \le 1$, i.e. $\delta = 0$ in Theorem 3. (This follows since H is positive semidefinite). In this case, in the method of the steepest descent $\rho(x) \ge \frac{1}{2}$. Using $G = I + H$, (12) and (17), it follows that

$$\begin{aligned}\|x^{k+1}-x^*\|_G^2 &\le \|x^k - x^*\|_G^2 - \frac{1}{2}\|e(x^k)\|^2 - (x^k-x^*)^T H (x^k - x^*) \\ &\le \|x^k - x^*\|_G^2 - \frac{1}{2}\frac{(4\lambda^*)^2}{(5+4\lambda^*)^2}\|x^k-x^*\|^2 - \|x^k-x^*\|_H^2 \\ &\le \Big(1 - \frac{1}{2}\cdot\frac{(4\lambda^*)^2}{(5+4\lambda^*)^2}\Big)\|x^k-x^*\|_G^2,\end{aligned} \tag{20}$$

for all x with $\|x-x^*\| \le 0.6\lambda^*\alpha$. Now observe that for $\|H\| \le 1$ it follows that $\|x^*\|_G \le \alpha\sqrt{2}$, and since $\|x^k - x^*\|_{(I-H)^2} \le \|x^k - x^*\|_G \le \sqrt{2}$, the second assumption of Theorem 3 (namely $x^k \in X$, i.e. $\|(I-H)(x^k - x^*)\| \le 0.6\lambda^*\alpha\}$) is satisfied by all iterates x^k if the steepest descent method is started at $x^0 = 0$ and λ^* satisfies $\lambda^* \ge 5\sqrt{2}/3 \approx 2.4$. From (20) we also obtain that in this case the rate of convergence is at least 0.787 that is after at most 10 iterations the error is reduced by a factor of 10. Such rate of convergence is sufficient for most purposes since each iteration is very cheap.

Summarizing we observe that if the steepest descent method does not converge rapidly, then $\lambda^* < 2.4$ must hold true (independent of the condition number of H). This bound "$\lambda^* < 2.4$" is certainly weaker than what we expect in practice. Indeed, in our numerical examples below, much smaller values of λ^* still allowed a reasonable rate of convergence of the steepest descent method.

On the PC method of the S-D type with scaling

As we have just seen, the numerical solution of problem (1) might be difficult, only when $H + \lambda^* I$ is ill-conditioned that is when λ^* is small. In the following we illustrate how to treat such ill-conditioned problems.

Note that if $\|x - (Hx + c)\| < \alpha$, then $e(x) = Hx + c$, and problem (1) is related with the problem of solving a linear equation

$$Hx + c = 0. \tag{21}$$

When H is ill-conditioned, the iterative scheme of Levenberg-Marquardt for solving equation (21) is

$$x^{k+1} = x^k - d(x^k), \tag{22}$$

where

$$d(x) = (\sigma I + H)^{-1}(Hx + c) \tag{23}$$

and $\sigma > 0$. Let $\mu = \dfrac{1}{\sigma}$, then $d(x)$ in (23) can be written as

$$(I + \mu H)^{-1} \cdot \mu(Hx + c). \tag{24}$$

Based on this idea, instead of taking H and c in (8) and (11), we should use μH and μc in the practical computation, i.e., we take the following iterative scheme:

$$x^{k+1} = x^k - (I + \mu H)^{-1}\{x^k - P_\Omega[x^k - \mu(Hx^k + c)]\}. \tag{25}$$

From Theorem 2 follows that, for all $\mu > 0$, the sequence $\{x^k\}$ generated by recursion (25) satisfies

$$\|x^{k+1} - x^*\|^2_{(I+\mu H)^2} \le \|x^k - x^*\|^2_{(I+\mu H)^2} - \|e(x^k, \mu)\|^2, \tag{26}$$

where

$$e(x, \mu) = x - P_\Omega[x - \mu(Hx + c)].$$

Inequality (26) will *not* guarantee a fast rate of convergence if $\|e(x^k, \mu)\|$ is small compared to $\|x^k - x^*\|_{(I+\mu H)^2}$. In the following we show that (locally) the parameter $\mu = 1/\lambda^*$ is a "good" choice in that for this choice of μ inequality (26) implies a fast linear convergence if x^k is already near the optimal solution x^*.

For $\mu = 1/\lambda^*$ we obtain

$$\|x - x^*\|^2_{(I+\mu H)^2} = \|(I + \mu H)(x - x^*)\|^2 = \|x + \mu Hx + \mu c\|^2.$$

We compare the above distance of x and $-\mu(Hx + c)$ with $\|e(x, \mu)\|$ and set $v = x + \mu(Hx + c)$,

$$\|e(x, \mu)\| = \|x - P_\Omega[x - \mu(Hx + c)]\| = \|x - P_\Omega[2x - v]\|.$$

Now, for x near x^* we have $\|x\| \approx a$ and $x \approx -\mu(Hx + c)$. Furthermore, in this case x is nearly orthogonal to v. Up to a relative error that tends to zero as x approaches x^* we may therefore conclude that

$$\|x - P_\Omega[2x - \nu]\| \ge \frac{1}{2}\|\nu\|,$$

i.e. $\|e(x,\mu)\| \geq \frac{1}{2}\|x - x^*\|_{(I+\mu H)^2}$. But in this case, inequality (26) implies convergence at the rate of $\sqrt{0.75}$, that is after at most 16 iterations the error $\|x^k - x^*\|_G$ is reduced by a factor of 10.

We note again, that if the ideal parameter $\mu = 1/\lambda^*$ was known, and also a factorization of $I + \mu H$ was given, then the optimal solution could be computed directly from this factorization. However, λ^* is unknown, and the factorization of $I + \mu H$ is too costly, therefore, the above analysis is only a motivation of how to aim at an alternative scheme (25).

In order to avoid the expensive factorization of $I + \mu H$, instead of the iterative scheme (25), we take the recursion

$$x^{k+1} = x^k - \rho(x^k, \mu)\{x^k - P_\Omega[x^k - \mu(Hx^k + c)]\}, \tag{27}$$

where

$$\rho(x^k, \mu) = \frac{\|e(x^k, \mu)\|^2}{e(x^k, \mu)^T(I + \mu H)e(x^k, \mu)}.$$

We call the method a *PC method of S-D type with scaling.* Again from Theorem 2 follows that, the sequence $\{x^k\}$ generated by recursion (27) satisfies

$$\|x^{k+1} - x^*\|^2_{(I+\mu H)} \leq \|x^k - x^*\|^2_{(I+\mu H)} - \rho(x^k, \mu)\|e(x^k, \mu)\|^2. \tag{28}$$

5. The Implementation of CG-PC Method

In general, we don't know whether $\|H^+c\| < a$. Therefore, we begin the solution process of (1) by taking starting point $x^0 = 0$, and solve the linear system $Hx + c = 0$ by using CG method. From Theorem 1 we know that the iterates $\{x^l\}$; $l = 0, \ldots, n$, satisfies $\|x^{j+1}\| > \|x^j\|$. This means, that $\|H^+c\| > a$ (hence $\|x^*\| = a$ and $\lambda^* > 0$) if and only if there is a $l \leq n$ with $\|x^l\| > a$. We use the iterative step of the CG method until an iterate $\|x^l\| > a$ or the iteration number l equal to n.

(**Scheme 1**)

$x^0 = 0$, While $\|x^l\| \leq a$ and $l < n$ compute x^{l+1} by a CG step.

If $\|x^l\| < a$, $l = 0, \ldots, n$, then the iterate x^n generated by CG method solves the linear system $Hx + c = 0$ and is a solution of the problem (1). Otherwise, as soon as $\|x^l\| > a$, $(l \leq n)$, we take $\tilde{x}^0 = ax^l/\|x^l\|$ as the new starting point and turn to use recursion scheme (27) of the PC method. The remaining question is how to choose a parameter μ in (27). From the analysis in Section 4, we know, that for the local fast convergence, an ideal choice of the parameter μ is $\mu = 1/\lambda^*$. Unfortunately, we do not know λ^*—if we did, we could compute x^* directly by solving the linear equation $(H + \lambda^* I)x + c = 0$.

It seems reasonable to view $\tilde{x}^0$ as an approximation of x^*. Because $\lambda^* x^* = -(Hx^* + c)$, we take

$$\mu = \frac{a}{\|H\tilde{x}^0 + c\|}$$

as the parameter in practical computation and use the following iterative scheme:

(**Scheme 2**)

$$\begin{aligned} &x^0 = \tilde{x}^0, \\ &e(x^k, \mu) = x^k - P_\Omega[x^k - \mu(Hx^k + c)], \\ &x^{k+1} = x^k - \frac{\|e(x^k, \mu)\|^2}{e(x^k)^T(I + \mu H)e(x^k, \mu)}\, e(x^k, \mu), \qquad k = 0, 1, \ldots \end{aligned}$$

Because $\|x^*\| = a$, the stopping criterion for iterative Scheme 2 is if both

$$\left|\frac{\|x\| - a}{a}\right| \leq \varepsilon, \quad \text{and} \quad \max\left\{\frac{\|e(x)\|}{\alpha}, \ \frac{\|e(x)\|}{\sqrt{a\|c\|}}\right\} \leq \varepsilon$$

are satisfied for some tolerance $\epsilon > 0$. Clearly as the norm of the optimal solution is a, we have to divide the vector $e(x)$ by a to obtain some sort of relative error. On the other hand we have seen following definition (5) that $e(x)$ also varies with the scaling of H and c, so $e(x)$ should be divided by the minimum of 1 and $\|c\|$. Since in our examples below $a\|c\|$ was always greater than 1 we only took the square root which resulted in the above stopping criterion.

A summary of CG-PC method is given in following:

The CG-PC Algorithm for quadratic problems

given K and $\varepsilon > 0$.

$done$:= **false**

$l := 0$

$x := 0$

while not $\|x\| < a$ **and** $l < n$ **do**

  Compute a new iterate x via a CG step

  $l := l + 1$

end

if $l = n$ **then** $done$:=**true**

else

  $k := 0$

  $x := \dfrac{ax}{\|x\|}$

  $\mu := \dfrac{a}{\|Hx + c\|}$

end

while not $done$ **and** $k \leq K$ **do**

  $e(x, \mu) = x - P_\Omega[x - \mu(Hx + c)]$

  $\rho(x, \mu) = \dfrac{\|e(x, \mu)\|^2}{e(x, \mu)^T (I + \mu H) e(x, \mu)}$

  $x := x - \rho(x, \mu) e(x, \mu)$

  $k := k + 1$

  if $\max\left\{\dfrac{\|e(x)\|}{\sqrt{a\|c\|}}, \dfrac{\|e(x)\|}{a}, \left|\dfrac{\|x\| - a}{a}\right|\right\} \leq \varepsilon$ $done$:=**true**

end

---

6. Numerical Results

In this section we present numerical results for the trust region problem in large scale optimization. We form test problems of type (1) with $H = A^T A$ and $c = -A^T b$ as in [6]: The matrix A is constructed synthetically such that it has a prescribed distribution of its singular values. This is accomplished by setting

$$A := U \Sigma V^T,$$

where

$$U = I_m - 2\frac{uu^T}{\|u\|_2^2},$$
$$V = I_n - 2\frac{vv^T}{\|v\|_2^2},$$

are Householder matrices and

$$\Sigma = \mathrm{diag}(\sigma_k)$$

is a $m \times n$ diagonal matrix. The vectors u, v and b contain pseudorandomnumbers:

$$\begin{aligned} u_1 &= 13846 \\ u_i &= (31416u_{i-1} + 13846) \bmod 46261 \quad i = 2, \ldots, m \\ v_1 &= 13846 \\ v_j &= (42108v_{j-1} + 13846) \bmod 46273 \quad j = 2, \ldots, n \\ b_1 &= 13846 \\ b_i &= (45278b_{i-1} + 13846) \bmod 46219 \quad i = 2, \ldots, m. \end{aligned}$$

As in [6], we take $\varepsilon = 5 \cdot 10^{-6}$ and only test the problems with $a < \|H^+c\|$.

The code was written in FORTRAN. The calculations have been performed on a 486 personal computer (without Weitek coprecessor). In following tables l indicates the step number by using CG method until $\|x^l\| > a$, k denotes the iteration number by PC method until the convergence criterion was met. Since the termination criterion of PC method was checked every fifth iteration, the iteration numbers (k) are multiples of 5.

Example 1. In the first set of test problems we take $m = 2000$, $n = 1000$ and set $\sigma_k = \cos\frac{k\pi}{n+1} + 1, k = 1, \ldots, n$. The singular values of matrix $H = A^TA$ tend to cluster at the endpoints of the interval $[0, 4]$. $\mathrm{Cond}(H) = \mathrm{Cond}(A^TA) = 1.65 \cdot 10^{11}$ and $\|H^+c\| = 4.37 \cdot 10^9$. Table 1 presents the test results for problems with $\alpha = 1000, \ldots, 10^7$, respectively.

Table 1.1 Results of CG-PC method with scaling

m	n	$\|H^+c\|$	a	l	μ	k
2000	1000	$4.37 \cdot 10^9$	10^4	1	0.01	10
2000	1000	$4.37 \cdot 10^9$	10^5	1	0.14	15
2000	1000	$4.37 \cdot 10^9$	10^6	7	11.05	120
2000	1000	$4.37 \cdot 10^9$	$2 \cdot 10^6$	20	47.17	375
2000	1000	$4.37 \cdot 10^9$	$3 \cdot 10^6$	36	97.13	690
2000	1000	$4.37 \cdot 10^9$	$5 \cdot 10^6$	63	215.32	1100
2000	1000	$4.37 \cdot 10^9$	$8 \cdot 10^6$	111	573.93	1765
2000	1000	$4.37 \cdot 10^9$	10^7	146	867.98	2625

Indeed, a proper choice of parameter μ in PC method scheme (27) is necessary for fast convergence. For the same problems, if the problem is somewhat ill-conditioned and we take $\mu \equiv 1$, the method needs more than 5 times of the iterations that the method with scaling needs. We report the test results in Table 1.2.

Table 1.2 Numerical results of CG-PC method without scaling

m	n	$\|H^+c\|$	a	l	μ	k
2000	1000	$4.37 \cdot 10^9$	$2 \cdot 10^6$	20	1	1495
2000	1000	$4.37 \cdot 10^9$	$3 \cdot 10^6$	36	1	3445
2000	1000	$4.37 \cdot 10^9$	$5 \cdot 10^6$	63	1	8690
2000	1000	$4.37 \cdot 10^9$	$8 \cdot 10^6$	111	1	> 15000
2000	1000	$4.37 \cdot 10^9$	10^7	146	1	> 20000

The Goldstein's fundamental projection method [4]

$$x^{k+1} = P_\Omega[x^k - \alpha(Hx^k + c)]$$

with a constant step length $0 < \alpha < 2/\|H\|$ is proved to be convergent. Since the singular values of the matrix A in our test problems tend to cluster at the endpoints of the interval $[0, 2]$, the constant step length can be taken in $(0, 0.5]$ (because $\|H\| < 4$). The iteration of fundamental projection method is very cheap. However, as Table 1.3 shows, it converges much slowly than CG-PC method.

Table 1.3 Numerical results of Fundamental Projection Method

m	n	$\|H^+c\|$	a	l	$k(\alpha = 0.5)$	$k(\alpha = 0.25)$
2000	1000	$4.37 \cdot 10^9$	$2 \cdot 10^6$	20	2160	2900
2000	1000	$4.37 \cdot 10^9$	$3 \cdot 10^6$	36	5120	6610
2000	1000	$4.37 \cdot 10^9$	$5 \cdot 10^6$	63	11325	16610
2000	1000	$4.37 \cdot 10^9$	$8 \cdot 10^6$	111	> 20000	> 20000
2000	1000	$4.37 \cdot 10^9$	10^7	146	> 20000	> 20000

Example 2. The second set of test problems is also the same in [6]. We take $m = 20000$, $n = 10000$ and set $\sigma_k = \exp\frac{-k}{1000}$, $k = 1, \ldots, n$. In these problems, $\text{Cond}(H) = \text{Cond}(A^TA) = 4.85 \cdot 10^8$ and $\|H^+c\| = 1.29 \cdot 10^9$.

Because the problems size is too large for a personal computer, and our main interest is to test the efficiency of the presented method, for this set of test problems, instead of solving

$$\min\{\|U\Sigma V^T\xi - b\| \mid \|\xi\| \le \alpha\}$$

we let $x = V^T\xi$ and solve the numerical equivalent problem

$$\min\{\|\Sigma x - U^Tb\| \mid \|x\| \le \alpha\}$$

In other words, we solve problems of form (1) with $H = \Sigma^T\Sigma$ and $c = -\Sigma^TU^Tb$. The test numerical results are given in Table 2.

Table 2 Results of CG-PC method with scaling

m	n	$\|H^+c\|$	a	l	μ	k
20000	10000	$1.29 \cdot 10^{10}$	10^4	1	0.02	10
20000	10000	$1.29 \cdot 10^{10}$	10^5	1	0.19	15
20000	10000	$1.29 \cdot 10^{10}$	10^6	1	3.37	20
20000	10000	$1.29 \cdot 10^{10}$	10^7	12	86.58	165
20000	10000	$1.29 \cdot 10^{10}$	10^8	116	2572.47	1575

For the same problems, we see that the CG-PC method is as powerful as the method in [6]. Although the CG-PC method needs about three times iterations that the method in [6] needs, but each iteration of the CG-PC method is very cheap. In addition, we don't need solving a linear least squares system, as the method in [6] finally did. Therefore, it seems that the CG-PC method is an alternative acceptable method for solving trust region problem in large scale optimization.

7. Conclusions

We presented the CG-PC methods for solving trust region problem in large scale optimization. The method is simple to implement and more efficient than other cheap methods (e.g. fundamental projection method) in literature. The numerical results show, that the CG-PC method produce an acceptable (see the relative residue-error $\frac{\|e(x)\|}{a}$ and the relative norm-error $|\frac{\|x\|}{a} - 1|$ of x), approximate solution x in a moderate number of iterations, even if the problems are somewhat ill-conditioned.

However, we would like to point out, that the recursion (27) without matrix factorization belongs to PC methods of the steepest descent type [9], which is not appropriate for solving ill-conditioned problems.

Acknowledgment. The author would like to thank Dr. F. Jarre for some valuable suggestions.

References

[1] J.E. Dennis, R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Printice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

[2] W. Gander, Least squares with a quadratic constraint, *Numerische Mathematik*, **36** (1981), 291–307.

[3] W. Gander, G.H. Golub, U. von Matt, A constrained eigenvalue problem, *Linear Algebra Appl.*, **114/115** (1989), 815–839.

[4] A.A. Goldstein, Convex programming in Hilbert space, *Bull. Amer. Math. Soc.*, **70** (1964), 709–710.

[5] G.H. Golub, C.F. Van Loan, Matrix Computations, 2nd Ed, The Johns Hopkins University Press, Baltimore, 1989.

[6] G.H. Golub, U. von Matt, Quadratically constrained least squares and quadratic problems, *Numerische Mathmatik*, **59** (1990), 561–580.

[7] B.S. He, A projection and contraction method for a class of linear complementarity problems and its application in convex quadratic programming, *Applied Mathematics and Optimization*, **25** (1992), 247–262.

[8] B.S. He, A new method for a class of linear variational inequalities, *Mathematical Programming*, **66** (1994), 137–144.

[9] B.S. He, Solving a class of linear projection equations, *Numerische Mathematik,* **68** (1994), 71–80.

[10] M. Hestenes, Conjugate Direction Methods in Optimization, Springer-Verlag, New York, Heidelberg, Berlin, 1980.

[11] F. Jarre, On the convergence of the method of analytic centers when applied to convex quadratic programs, *Mathematical Programming*, **49** (1991), 341–358.

[12] J.J. Moré, D.C. Sorensen, Computing a trust region step, *SIAM J. Scientific and Statistic Computing,* **4** (1983) 553–572.

[13] J.J. Moré, Generalizations of the trust region problem, *Optimi. Methods Software*, **2** (1993), 189–209.

[14] C.H. Reinsch, Smoothing by spline Functions, II, *Numerische Mathmatik*, **16** (1971), 451–454.

[15] T. Steihaug, The conjugate gradient method and trust regions in large scale optimization, *SIAM J. Numer. Anal.*, **20** (1983), 626–637.

[16] Y. Yuan, On a subproblem of trust region algorithms for constrained optimization, *Mathematical Programming*, **47** (1990), 53–63.