BIT
2000, Vol. 40, No. 4, pp. 001–004
0006-3835/00/4004-0001 $15.00
© Swets & Zeitlinger

BOUND CONSTRAINED REGULARIZATION FOR ILL-POSED PROBLEMS *

D. DEMENTIEV[1] and J. G. NAGY[2] †

[1] *IST Department, Marshall University, Huntington, WV 25755, USA. email: dementiev@marshall.edu*

[2] *Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322, USA. email: nagy@mathcs.emory.edu*

Abstract.

We consider large scale ill-conditioned linear systems arising from discretization of ill-posed problems. Regularization is imposed through an (assumed known) upper bound constraint on the solution. An iterative scheme, requiring the computation of the smallest eigenvalue and corresponding eigenvector, is used to determine the proper level of regularization. In this paper we consider several computational issues involved in this approach, including the use of a Rayleigh quotient iteration for the eigenvalue/vector computation.

AMS subject classification: 65F10, 65F15, 65F22.

Key words: ill-posed problems, regularization, bound constraints, preconditioning.

1 Introduction

Discretization of large scale ill-posed problems result in linear systems of the form:

$$\mathbf{g} = K\mathbf{f} + \mathbf{n}\,, \tag{1.1}$$

where K is a large ill-conditioned matrix, $\mathbf{n}$ is a vector representing perturbations (such as noise) in the measured data, $\mathbf{g}$, and the aim is to compute a good approximation to the unknown vector $\mathbf{f}$. It is well known (cf. [5, 12]) that regularization is needed in order to avoid computing solutions that are corrupted by noise. Regularization can take many forms, such as Tikhonov regularization [9], truncated iterations [10], and truncated singular value decomposition [12]. In this paper we assume an upper bound, Δ, on the norm of the true solution, $\mathbf{f}$, is known, and enforce regularization by computing a solution of the constrained minimization problem:

$$\begin{array}{c} \min \frac{1}{2}\mathbf{f}^T K^T K\mathbf{f} - \mathbf{g}^T K\mathbf{f} \\ \text{subject to } ||\mathbf{f}||_2 \leq \Delta\,. \end{array} \tag{1.2}$$

*Received May 2002. Revised ??? Comunicated by ???

†This work was partially supported by the National Science Foundation under Grant DMS 00-75239.

We note that in many applications it may not be possible to find a good upper bound Δ, and the numerical methods considered in this paper may not be appropriate in these situations. However, if a good bound is known, then knowledge of this information should be exploited. Theoretical results characterizing properties of solutions of (1.2) have been presented by Gander [6]. In this paper, we are concerned with the development of efficient algorithms for computing solutions of (1.2).

For matrices of small dimension, solutions to (1.2) can be computed using the singular value decomposition (SVD) [7]. For large-scale problems, though, the SVD approach can be prohibitively expensive. Eldén [4] proposed a more efficient approach based on using the QR factorization, but it is still relatively expensive, and any special structure of the matrix K is not easily exploited in this scheme. Chan, Olkin and Cooley [3] consider the use of Newton and secant iteration methods to solve (1.2), but they assume an efficient solver for least squares problems exists. Golub and von Matt [8] propose a scheme based on Lanczos bidiagonalization for the case when the bound in (1.2) is equality.

The scheme we use to solve the bound constrained minimization problem (1.2) is based on recent work by Sorensen [19] and Rojas [17] (see also [18]). Our approach differs in some of the computational details, where we adapt some recent work by Björck, Heggernes and Matstoms [2] (see also [1]) for large total least squares problems. The method suggested by Rojas and Sorensen is based on the following lemma.

LEMMA 1.1. *The vector* $\mathbf{f}$ *is a solution to the bound constrained minimization problem (1.2) if and only if* $\mathbf{f}$ *is a solution to the linear system*

$$(K^T K - \lambda I)\mathbf{f} = K^T \mathbf{g}\,, \tag{1.3}$$

where $K^T K - \lambda I$ *is positive semi-definite,* $\lambda \leq 0$ *and* $\lambda(\Delta - ||\mathbf{f}||_2) = 0$.

Typically the linear system (1.3) is written with a nonnegative regularization parameter $\mu = -\lambda$. The nonstandard formulation used by Rojas and Sorensen allows for the problem of computing a regularized solution to be easily recast into a form that requires the computation of the smallest eigenvalue and associated eigenvector of a certain structured matrix. This relationship is outlined in section 2. In section 3 we describe a Rayleigh quotient iteration method (which is a modification of a scheme used in [1, 2] for large scale total least squares problems) to compute the smallest eigenvalue and corresponding eigenvector of the structured matrix. Some additional algorithmic considerations are discussed in section 4, and numerical results are reported in section 5.

2 An Eigenproblem

Suppose α is a fixed scalar. Then the solution to the bound constrained minimization problem is related to the solution of the following eigenproblem (cf. [19, 17]):

$$\begin{bmatrix} \alpha & -\mathbf{g}^T K \\ -K^T \mathbf{g} & K^T K \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{f} \end{bmatrix} = \lambda \begin{bmatrix} 1 \\ \mathbf{f} \end{bmatrix},$$

where the eigenvector has been normalized to have first component equal to one. We assume that the first component of the eigenvector is different from zero; in general, as we remark later, this will usually be the case. If the first component of the eigenvector is zero, then some modifications of the approach are needed; see [19, 17] for further details.

Note that this eigenvalue problem implies that

$$-K^T\mathbf{g} + K^TK\mathbf{f} = \lambda\mathbf{f}\,,$$

or equivalently

$$(K^TK - \lambda I)\mathbf{f} = K^T\mathbf{g}\,.$$

Thus, according to Lemma 1.1, the bound constrained minimization problem (1.2) is solved if we can find a scalar α such that the bordered matrix

$$B_\alpha = \begin{bmatrix} \alpha & -\mathbf{g}^TK \\ -K^T\mathbf{g} & K^TK \end{bmatrix}$$

has an eigenvalue/eigenvector pair λ, $\begin{bmatrix} 1 \\ \mathbf{f} \end{bmatrix}$ satisfying:

- $K^TK - \lambda I$ is positive semi-definite,
- $\lambda \leq 0$, and
- $\lambda(\Delta - ||\mathbf{f}||_2) = 0$.

If we can iteratively adjust α so that such an eigenpair can be found, then we have an approach to solve the regularization problem (1.2).

At first glance this approach appears to be computationally expensive. However, the following important observations allow us to restrict our search of eigenpairs to the smallest eigenvalue of B_α.

1. Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{n+1}$ be the eigenvalues of the $(n+1)\times(n+1)$ bordered matrix B_α, and $0 < \delta_1 \leq \delta_2 \leq \cdots \leq \delta_n$ be the eigenvalues of the $n \times n$ matrix K^TK. The interlacing property of eigenvalues [7] implies that
$$\lambda_1 \leq \delta_1 \leq \lambda_2 \leq \delta_2 \leq \cdots \leq \lambda_n \leq \delta_n \leq \lambda_{n+1}\,.$$
2. For any choice of α, the interlacing property implies that $\lambda_j > 0$ for $j \geq 2$. Moreover $K^TK - \lambda_j I$ is not positive semi-definite for $j \geq 2$.
3. Moreover, for any α, the interlacing property implies that if $\lambda = \lambda_1$, then $K^TK - \lambda I$ is always positive semi-definite.

Since we need to find an eigenvalue satisfying $\lambda \leq 0$ and such that $K^TK - \lambda I$ is positive semi-definite, these observations suggest that we need only consider the smallest eigenvalue and corresponding eigenvector of B_α. Thus, an outline of a scheme to compute a solution to the regularized problem (1.2) is:

Choose an initial α
`for` $k = 0, 1, 2, \ldots$
- Compute the smallest eigenvalue λ and corresponding (normalized) eigenvector $\begin{bmatrix} 1 \\ \mathbf{f} \end{bmatrix}$
- if $\lambda \leq 0$ and $\lambda(\Delta - ||\mathbf{f}||_2) = 0$, stop.
- otherwise, adjust α and continue

`end`

Let us briefly explain why we can choose the first entry of the eigenvector of B_α to be equal to 1. In the general case we look for the solution of

$$B_\alpha \mathbf{x} = \lambda \mathbf{x} \tag{2.1}$$

in the form

$$\mathbf{x} = \begin{bmatrix} \mu \\ \mathbf{f} \end{bmatrix},$$

that is,

$$\begin{bmatrix} \alpha & -\mathbf{g}^T K \\ -K^T \mathbf{g} & K^T K \end{bmatrix} \begin{bmatrix} \mu \\ \mathbf{f} \end{bmatrix} = \lambda \begin{bmatrix} \mu \\ \mathbf{f} \end{bmatrix}$$

which is the same as the linear system

$$\left\{ \begin{array}{rcl} (\alpha - \lambda)\mu & = & \mathbf{g}^T K \mathbf{f} \\ (K^T K - \lambda I)\mathbf{f} & = & \mu \mathbf{g}^T K \end{array} \right. .$$

If we assume that the first entry of the solution can be zero (i.e. $\mu = 0$), then this system becomes the following one:

$$\left\{ \begin{array}{rcl} \mathbf{g}^T K \mathbf{f} & = & 0 \\ (K^T K - \lambda I)\mathbf{f} & = & 0 \end{array} \right. .$$

However, since we are looking for the solution of this linear system under the assumption that $\lambda < 0$, then matrix $K^T K - \lambda I$ is positive definite. Thus the linear subsystem $(K^T K - \lambda I)\mathbf{f} = 0$ has only one solution $\mathbf{f} = 0$. So, the solution of the original problem (2.1) is $\mathbf{0}$ and this is not an eigenvector. This contradiction proves that $\mu \neq 0$, or without loss of generality, we can assume $\mu = 1$.

Clearly the efficiency of this approach depends on how efficiently we can compute the smallest eigenvalue and eigenvector of B_α, and on how quickly we can iteratively update α in order to converge to the optimal λ. These issues are discussed in the following sections.

3 Solving the Eigenproblem

Since we are considering large-scale problems, we use an iterative method to compute the smallest eigenvalue and eigenvector of B_α. In particular, we would like a scheme in which matrix-vector multiplications are the most expensive part of each iteration. Sorensen [19] and Rojas [17] use the implicitly restarted Lanczos method as implemented in ARPACK [15].

In this paper, we consider an alternative approach which uses a Rayleigh quotient iteration (RQI). The motivation for using RQI comes from the fact that the eigenvalue problem we want to solve is very similar to the one that arises in the closed form expression of the basic solution for total least squares problems (see [13, Theorem 2.7]). Björck [1] proposed using an RQI method to solve the eigenvalue problem, and Björck, Heggernes and Matstoms [2] provided a further analysis of the scheme. In particular, they suggest that the RQI approach can be used to solve a wider range of problems than Lanczos type algorithms, particularly because linear systems arising in the iteration can be solved by a *preconditioned* conjugate gradient method. In the TLS context, Kamm and Nagy [14] show that preconditioning can have a dramatic effect on the speed of convergence when the coefficient matrix has a Toeplitz structure.

The derivation of the RQI method is very similar to that given in [1], but we present it here for completeness. For the eigenvalue problem

$$\begin{bmatrix} \alpha & -\mathbf{g}^T K \\ -K^T\mathbf{g} & K^T K \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{f} \end{bmatrix} = \lambda \begin{bmatrix} 1 \\ \mathbf{f} \end{bmatrix},$$

the Rayleigh quotient is

$$\begin{aligned} \rho(\mathbf{f}) &= \frac{\begin{bmatrix} 1 & \mathbf{f}^T \end{bmatrix} \begin{bmatrix} \alpha & -\mathbf{g}^T K \\ -K^T\mathbf{g} & K^T K \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{f} \end{bmatrix}}{\begin{bmatrix} 1 & \mathbf{f}^T \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{f} \end{bmatrix}} \\ &= \frac{\alpha - 2\mathbf{g}^T K\mathbf{f} + \mathbf{f}^T K^T K\mathbf{f}}{1 + \mathbf{f}^T\mathbf{f}} \\ &= \frac{\alpha - \mathbf{g}^T\mathbf{g} + \mathbf{r}^T\mathbf{r}}{1 + \mathbf{f}^T\mathbf{f}}, \end{aligned}$$

where $\mathbf{r} = \mathbf{g} - K\mathbf{f}$. The RQI method for our problem can be stated as (see [7]):

$\mathbf{f}_0$ given
`for` $j = 0, 1, 2, \ldots$
  $\lambda^{(j)} = \rho(\mathbf{f}_j)$
  Solve $(B_\alpha - \lambda^{(j)} I) \begin{bmatrix} 1 \\ \mathbf{f}_{j+1} \end{bmatrix} = \beta_j \begin{bmatrix} 1 \\ \mathbf{f}_j \end{bmatrix}$
`end`

The linear system we need to solve at each step of RQI is:

$$\begin{bmatrix} \alpha - \lambda^{(j)} & -\mathbf{g}^T K \\ -K^T\mathbf{g} & K^T K - \lambda^{(j)} I \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{f}_{j+1} \end{bmatrix} = \beta_j \begin{bmatrix} 1 \\ \mathbf{f}_j \end{bmatrix}.$$

Multiplying both sides of this relation by

$$\begin{bmatrix} 1 & \mathbf{g}^T K (K^T K - \lambda^{(j)} I)^{-1} \\ 0 & I \end{bmatrix}$$

we obtain

$$\begin{bmatrix} \tau_j & 0 \\ -K^T\mathbf{g} & K^TK \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{f}_{j+1} \end{bmatrix} = \beta_j \begin{bmatrix} 1 \\ 1+\mathbf{z}_j^T\mathbf{f}_j \end{bmatrix} .$$

where $\mathbf{z}_j = (K^TK - \lambda^{(j)}I)^{-1}K^T\mathbf{g}$ and $\tau_j = \alpha - \lambda^{(j)} - \mathbf{g}^TK\mathbf{z}_j$. From this relation, we see that

$$\beta_j = \frac{\tau_j}{1+\mathbf{z}_j^T\mathbf{f}_j} \tag{3.1}$$

$$\mathbf{f}_{j+1} = \mathbf{z}_j + \beta_j\mathbf{u}_j , \tag{3.2}$$

where $\mathbf{u}_j = (K^TK - \lambda^{(j)}I)^{-1}\mathbf{f}_j$. If we now define the residuals

$$\mathbf{p}_j = -K^T\mathbf{r}_j - \lambda^{(j)}\mathbf{f}_j \tag{3.3}$$

$$\gamma_j = -\mathbf{g}^T\mathbf{r}_j + \lambda^{(j)} + \mathbf{g}^T\mathbf{g} - \alpha , \tag{3.4}$$

then we can rewrite $\mathbf{z}_j$ and τ_j as

$$\mathbf{z}_j = \mathbf{f}_j + \mathbf{w}_j \tag{3.5}$$

$$\tau_j = \mathbf{z}_j^T\mathbf{p}_j - \gamma_j + \alpha - \mathbf{g}^T\mathbf{g} \tag{3.6}$$

Using the relations (3.1–3.6) in the RQI method, we obtain the following algorithm:

```
Assume an initial f is given
r = g - Kf
α_g = α - g^T g
λ^(0) = (α_g + r^T r)/(1 + f^T f)
for j = 0, 1, 2, ...
    p = -K^T r - λ^(j) f
    γ = -g^T r + λ^(j) - α_g
    Solve (K^T K - λ^(j) I)w = -p
    z = f + w
    τ = z^T p - γ
    β = τ/(1 + z^T f)
    Solve (K^T K - λ^(j) I)u = f
    f = z + βu
    r = g - Kf
    λ^(j) = (α_g + r^T r)/(1 + f^T f)
end
```

The advantages of using this RQI approach include the fact that RQI is cubically convergent, and that we can solve the linear systems involving $K^TK - \lambda^{(j)}I$ using a preconditioned conjugate gradient method; see [1, 2] for further details.

4 Updating α

In this section our goal is to find α such that the normalized eigenvector $[1, \mathbf{f}^T]^T$ corresponding to the smallest eigenvalue λ_α of the matrix B_α satisfies

the equation:

$$||\mathbf{f}||_2 = \Delta.$$

Rendl and Wolkowicz [16] presented the following bounds for the optimal parameter α_{opt}

$$\delta_1 - \frac{||K^T\mathbf{g}||}{\Delta} \leq \alpha_{opt} \leq \delta_1 + ||K^T\mathbf{g}||\Delta, \tag{4.1}$$

where δ_1 is the smallest eigenvalue of K^TK.

However it would be very expensive to find the exact value of δ_1 or even a good approximation to it. For this reason, we will use the following estimates for δ_1 (see, for example, [17])

$$\delta_1 \leq \left(\frac{||K^T\mathbf{x}||}{||\mathbf{x}||}\right)^2$$

where $\mathbf{x}$ is a random vector, and

$$\delta_1 \geq \lambda_1(\min(0, \left(\frac{||K^T\mathbf{x}||}{||\mathbf{x}||}\right)^2 + ||K^T\mathbf{g}||\Delta))$$

where λ_1 is the minimal eigenvalue of B_α for the corresponding α. The last inequality holds because of the interlacing property. Thus, we have computable bounds for the optimal value of $\alpha \in [\alpha_l, \alpha_r]$:

$$\alpha_r = \left(\frac{||K^T\mathbf{x}||}{||\mathbf{x}||}\right)^2 + ||K^T\mathbf{g}||\Delta,$$

$$\alpha_l = \lambda_1(\min(0, \alpha_r)) - \frac{||K^T\mathbf{g}||}{\Delta}.$$

4.1 Adjusting α

One approach to find the optimal value α_{opt} is based on the function $\phi(\lambda)$ defined by

$$\phi(\lambda) = \mathbf{g}^T K \left(K^TK - \lambda I\right)^{-1} K^T\mathbf{g}.$$

If the function $\phi(\lambda)$ is known, then the optimal parameter α can be found in the form

$$\alpha = \lambda + \phi(\lambda). \tag{4.2}$$

The value of the function can be computed for each λ, however we do not have an analytical expression for $\phi(\lambda)$. Because of this, Rojas, Santos and Sorensen [18, 17] suggested to use some approximations to the function $\phi(\lambda)$.

In particular, they approximate $\phi(\lambda)$ using the function $\tilde{\phi}(\lambda)$ which has the form

$$\tilde{\phi}(\lambda) = \frac{\gamma^2}{\delta - \lambda}, \tag{4.3}$$

where γ^2 and δ are to be found. It is very reasonable to look for an approximation having this form because if all eigenvalues δ_i are known, then the function $\phi(\lambda)$ can be written in the form

$$\phi(\lambda) = \sum_{i=1}^{n} \frac{\gamma_i^2}{\delta_i - \lambda}, \tag{4.4}$$

where γ_i are the expansion coefficients of $-K^T\mathbf{g}$ in the eigenvector basis.

Another approximation for the function $\phi(\lambda)$ suggested by Rojas [17] has the form

$$\tilde{\phi}(\lambda) = \frac{\gamma^2}{\delta - \lambda} + \beta(\delta - \lambda) + \eta, \tag{4.5}$$

where the coefficients β, γ^2, δ, and η are to be determined.

To find the undetermined coefficients of the function $\tilde{\phi}(\lambda)$ the following properties of the original function are used:

$$\phi(\lambda) = \mathbf{g}^T K \mathbf{f}_\lambda \quad \text{and} \quad \phi'(\lambda) = \mathbf{f}_\lambda^T \mathbf{f}_\lambda, \tag{4.6}$$

where the vector $[1, \mathbf{f}_\lambda^T]^T$ is the eigenvector of B_α corresponding to the eigenvalue λ.

Note: we are assuming that the approximation function $\tilde{\phi}(\lambda)$ also satisfies these properties. We also would like to note that for the approximation (4.3) one has to know only the minimal eigenvalue of B_α, however for (4.5) it is not enough to know only one pair $(\lambda_1, \mathbf{f}_{\lambda_1})$, because there are four unknown parameters and only two conditions. Therefore, one has to know two pairs $(\lambda_1, \mathbf{f}_{\lambda_1})$ and $(\lambda_2, \mathbf{f}_{\lambda_2})$ to find all unknown parameters.

4.2 An alternative method for updating α

Aiming to find the same optimal value of the parameter α we consider an alternative approach. Let us recall that the our goal is to find such a value α_{opt} that satisfies the equation $||\mathbf{f}_{\lambda_\alpha}||_2 = \Delta$. Thus, the main idea of the method is to solve the equation $||\mathbf{f}_{\lambda_\alpha}||_2 - \Delta = 0$ over the interval $[\alpha_l, \alpha_r]$.

Let us define a new function $n(\alpha)$ by

$$n(\alpha) = ||\mathbf{f}_{\lambda_\alpha}||_2 - \Delta,$$

where λ_α is the smallest eigenvalue of B_α and $[1, \mathbf{f}_{\lambda_\alpha}^T]^T$ is the corresponding eigenvector. Now we need to solve the equation $n(\alpha) = 0$ numerically. In order to do this we approximate the function $n(\alpha)$ by approximating $||\mathbf{f}_{\lambda_\alpha}||$. To do this we recall that by the second part of (4.6) $||\mathbf{f}_{\lambda_\alpha}|| = \sqrt{\phi'(\lambda_\alpha)}$ and could be approximated by a rational function of λ. However, since it is more convenient for us to approximate it by a rational function of α, we use

$$\tilde{n}(\alpha) = \frac{a}{b - \alpha} + c,$$

where the coefficients a, b, and c are to be found. This approximation is similar to (4.5) and, in a small neighborhood of the root of the equation $n(\alpha) = 0$, we

claim that they are essentially the same. This can be seen by observing that near the root, $\phi'(\lambda) \approx \Delta$, and so differentiating equation (4.2) with respect to α we obtain

$$\frac{d\lambda}{d\alpha} = \frac{1}{1+\phi'(\lambda)} \approx \frac{1}{1+\Delta^2}.$$

That is, λ_α depends linearly on α, and it is therefore reasonable to consider a rational approximation of $n(\alpha)$.

Since we need to find three coefficients we have to know values of the new function $\tilde{n}(\alpha)$ at three different points. Thus, knowing values of $n(\alpha)$ at points α_l, α_r and an intermediate point α_i we are able to find the unknown coefficients and then solve

$$\frac{a}{b-\alpha} + c = 0$$

for α. Doing this we can conclude that the next intermediate pont α_{next} can be expressed by

$$\alpha_{next} = b + \frac{a}{c}, \tag{4.7}$$

where

$$\begin{aligned} b &= \left(\alpha_r - \alpha_i \tfrac{n(\alpha_l)-n(\alpha_i)}{\alpha_l-\alpha_i} \tfrac{\alpha_r-\alpha_l}{n(\alpha_r)-n(\alpha_l)}\right) \left(1 - \tfrac{n(\alpha_l)-n(\alpha_i)}{\alpha_l-\alpha_i} \tfrac{\alpha_r-\alpha_l}{n(\alpha_r)-n(\alpha_l)}\right)^{-1} \\ a &= -\tfrac{n(\alpha_l)-n(\alpha_i)}{\alpha_l-\alpha_i}(\alpha_l-b)(\alpha_i-b) \\ c &= n(\alpha_l) - \tfrac{a}{\alpha_l-b} \end{aligned} \tag{4.8}$$

The iteration method and safeguarding. At each step of the iteration process using the points $(\alpha_l, n(\alpha_l))$, $(\alpha_i, n(\alpha_i))$, and $(\alpha_r, n(\alpha_r))$ and expressions (4.8) we evaluate the next approximation to the optimal parameter and update the points. If the current step of the process looks like the one on Figure 4.1, then the old α_i becomes the new α_l, α_{next} becomes the new α_i, and α_r stays unchanged.

When updating α one has to make sure that the next approximation α_{next} is located in the correct subinterval of the interval $[\alpha_l, \alpha_r]$. It might happen, due to roundoff errors, that $n(\alpha_i)$ is not between $n(\alpha_l)$ and $n(\alpha_r)$. In this case the approximation function will behave as shown in Figure 4.2. In situations such as this, we choose α_{next} by using the secant method to find the solution numerically

$$\alpha_{next} = \alpha_i - (\alpha_r - \alpha_i)\frac{n(\alpha_i)}{n(\alpha_r)}.$$

Choosing the initial point α_i. Starting the iteration process to find the optimal value of α we are not able to find all unknown coefficients in the approximation $\tilde{n}(\alpha)$. To do this we need to have a value of the function at one more point. We will call this point the initial one. Choosing the initial point we would like to get a good initial approximation to the optimal value. We can do that if we choose $\alpha = \mathbf{g}^T K K^T \mathbf{g}$ as the initial approximation. This

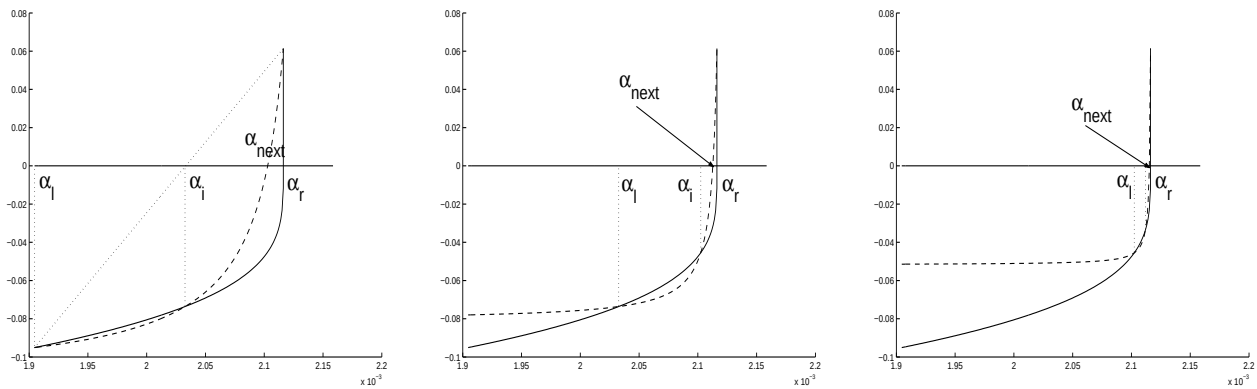


Figure 4.1: One step of the iteration process.

choice can be explained if we remember that $\mathbf{g}$ is given by (1.1). If we solve the pure system with no noise in it, then $\mathbf{g} = K\mathbf{f}$ and in this case the optimal parameter $\alpha_{opt} = \mathbf{g}^T KK^T \mathbf{g}$. It is easy to see that in this case the $\mathbf{f}$-part of the eigenvector $[1, \mathbf{f}^T]^T$ corresponding the minimal eigenvalue $\lambda = 0$ of the matrix $B_{\alpha_{opt}}$ is the solution of the original system (see, for example, [2]). If the noise level is not too large, say $||n|| \leq \varepsilon ||\mathbf{g}||$ (where $\varepsilon \approx 10^{-2}$–$10^{-3}$), we can conclude that $\alpha = \mathbf{g}^T KK^T \mathbf{g}$ is a very reasonable approximation. Extensive numerical experiments indicate that $n(\alpha_i) > 0$. We are not able to prove this as a fact, but evaluating function $n(\alpha)$ at the point α_i before the point α_r gives us an opportunity to avoid an extra step of the iteration method.

5 Numerical Results

In this section we present the numerical results using the algorithms described in this paper. The first set of experiments are similar to these used be Rojas [17]. In particular, we consider several test problems (listed in the first column of Table 1) from Hansen's Regularization Tools package [11]. The dimension of all problems was chosen to be $n = 300$. The right-hand side is

$$\bar{\mathbf{g}} = \mathbf{g} + \varepsilon \frac{\mathbf{r}}{||\mathbf{r}||},$$

where $\mathbf{g}$ is the "pure" right-hand side, that is $K\mathbf{f}_{exact} = \mathbf{g}$, $\mathbf{r}$ is a random vector with components uniformly distributed in $[0, 1]$, and ε is the noise level. The trust-region radius Δ was chosen to be equal to $||\mathbf{f}_{exact}||$.

Table 1 shows the numerical results of the described algorithm applied to specific problems. These results compare favorably with those presented in [17]. Moreover, we have also used Newton and secant methods suggested in [3], and we have found the approach used in this paper to be more robust. That is, although the Newton and/or secant methods sometimes require fewer matrix-

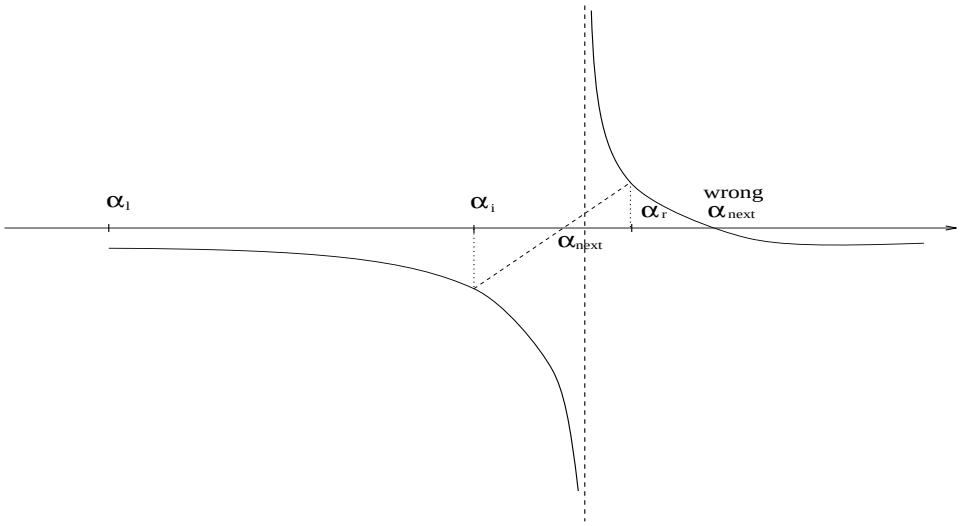


Figure 4.2: Safeguarding α_{next}.

Problem	Δ	$\frac{\|f-f_{exact}\|}{\|f_{exact}\|}$	TR its	RQI its	CGLS its	MV prods
baart	1.2533	0.123742	4	20	185	469
deriv2	0.5514	0.028558	11	36	1009	2213
foxgood	9.9999	0.036772	3	12	60	183
wing	0.5774	0.602906	8	27	212	557
phillips	2.9999	0.028761	4	15	221	523
shaw	17.2893	0.090364	6	24	263	852

Table 5.1: No preconditioner

vector multiplications, the solutions can be significantly worse than those obtained by the presented method. Table 2 shows the relative errors and required number of matrix-vector products for the same set of test problems. Note, that only for the *phillips* test problem is the same precision achieved. Relative errors for all of the other problems are substantially larger than those obtained with our approach.

In the next set of experiments we illustrate that preconditioning can be effective in reducing the computational cost of the approach used in this paper. The test problem we use is a simulation of an application in which a degraded gamma ray spectrum is to be restored [20, 21]. The true solution $\mathbf{f}_{exact}$ (i. e., the original spectrum) is shown on Figure 5.1-a, and the noise free right hand side $\mathbf{g} = K\mathbf{f}_{exact}$ (i. e., the degraded spectrum) is shown on Figure 5.1-b. The matrix K is a symmetric Toeplitz matrix with entries in the first column given

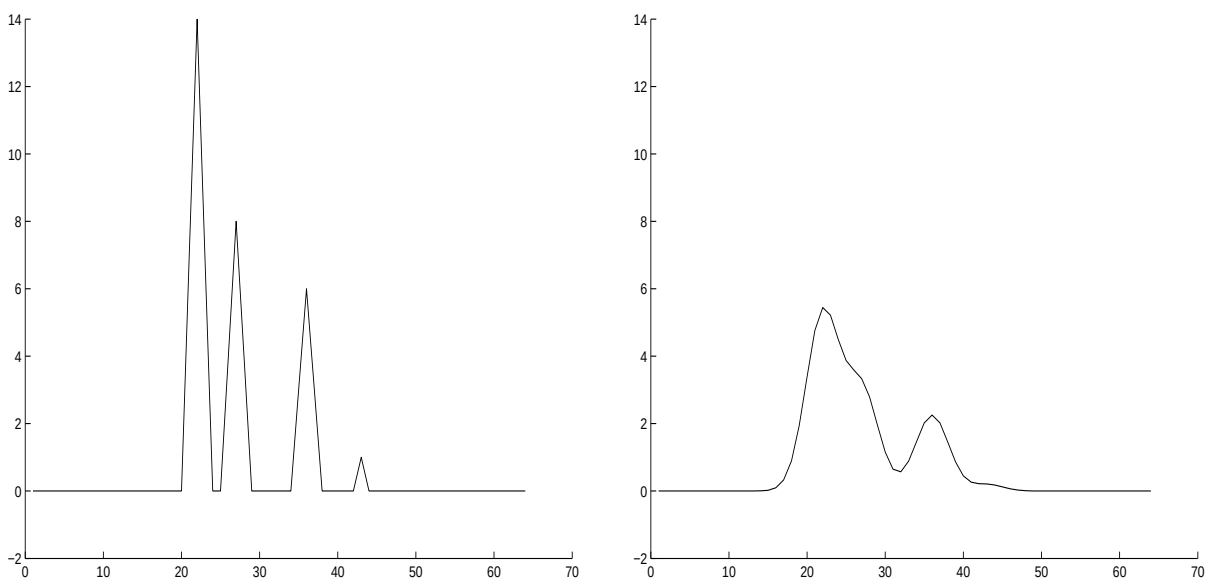
Problem	Newton method		Secant method	
	$\frac{\|\|f-f_{exact}\|\|}{\|\|f_{exact}\|\|}$	**MV prods**	$\frac{\|\|f-f_{exact}\|\|}{\|\|f_{exact}\|\|}$	**MV prods**
baart	0.350582	43	0.341526	25
deriv2	0.771426	79	0.609696	41
foxgood	0.295256	61	0.036871	37
phillips	0.028086	65	0.028320	65
shaw	0.167104	77	0.166490	371

Table 5.2: Results for Newton and secant method ($\varepsilon = 0.01$).

by

$$K(j,1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp[\frac{-(j-1)^2}{2\sigma^2}]$$

where $\sigma = 2.0$. Noise was added to $\mathbf{g}$, as in the previous set of experiments, for varying values of ε.



a) Original signal. b) Degraded signal.

Figure 5.1: Original and degraded signals.

Circulant preconditioners were constructed for the linear systems of the form

$$\left(K^T K - \lambda I\right) \mathbf{x} = \mathbf{y}$$

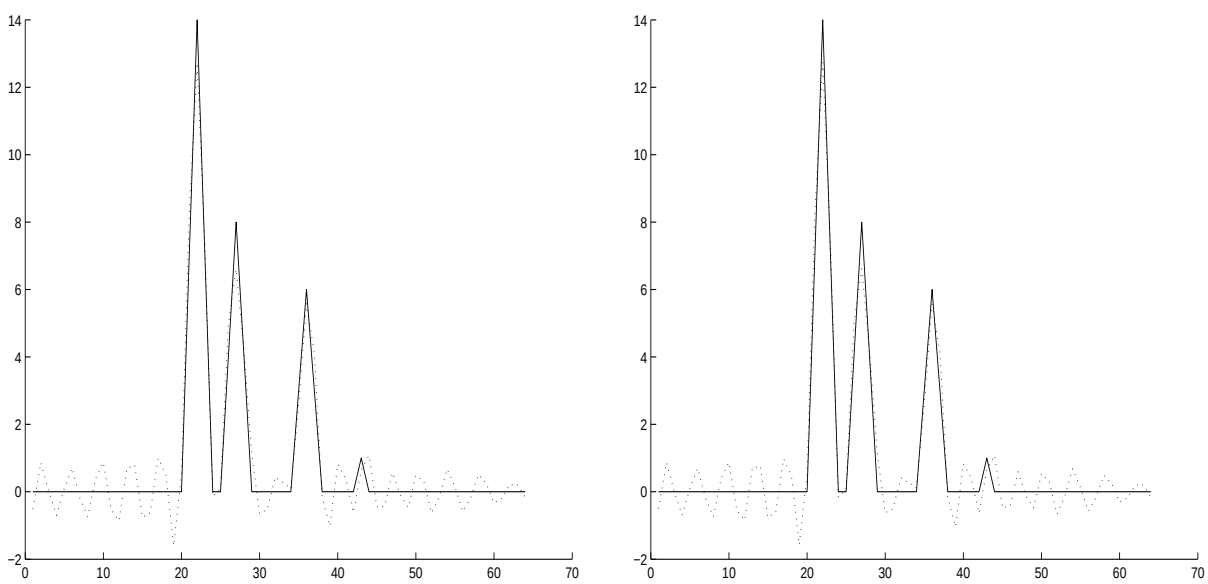
that arise in the RQI methods; see [22] for further details.

Table 3 provides a summary of numerical results for this test problem using different noise level ε. As we can see, preconditioning substantially reduces the number of matrix-vector products (we count each preconditioner solve as one matrix-vector product).

ε	**Precond**	$\frac{\lVert f - f_{exact} \rVert}{\lVert f_{exact} \rVert}$	**TR its**	**RQI its**	**CGLS its**	**MV prods**
0.001	No	0.18051011	6	22	2210	4751
	Yes	0.17749838	7	26	920	4071
0.003	No	0.22681155	6	23	2145	4554
	Yes	0.22821271	5	20	650	2896
0.01	No	0.24978938	7	23	2156	4471
	Yes	0.24978938	6	23	643	2851
0.03	No	0.36806460	8	33	3383	6937
	Yes	0.39216796	7	26	651	2813

Table 5.3: Using preconditioner.

Figure 5.2 illustrates the solutions of the problem computed without using a preconditioner (a) and using the preconditioner (b). The dotted solutions are the approximate ones. We can see that both approximate solutions are very close to the real one inside the interval and a little noisy when it comes to the boundaries.



a) No preconditioner used. b) Using preconditioner.

Figure 5.2: Comparing with the true solution.

6 Concluding Remarks

We have considered an approach for solving bound constrained regularization problems proposed recently by Sorensen [19] and Rojas [17], which requires computing the smallest eigenvalue of structured matrices. We have shown that a Rayleigh quotient method can be effectively used for this computation. The advantage of this approach is that preconditioning can be used to reduce the computational effort. Moreover, we have proposed an alternative method for updating the alpha parameter. Numerical experiments indicate that our approach is competitive with the Sorensen-Rojas scheme. However, it is important to note that there still is not one method that is optimal for all problems. It would be useful to do a comprehensive study of the various existing approaches, e.g., those discussed in [3, 8, 17, 19] as well as our approach, on a variety of ill-posed problems. We plan to carry out such a study in our future work.

REFERENCES

1. Å. Björck. Newton and Rayleigh quotient methods for total least squares problems. In S. Van Huffel, editor, *Recent Advances in Total Least Squares Techniques and Errors–in–Variables Modeling: Proceedings of the Second International Workshop on Total Least Squares and Errors–in–Variables Modeling*, pages 149–160. SIAM, Philadelphia, 1997.
2. Å. Björck, P. Heggernes, and P. Matstoms. Methods for large scale total least squares problems. Technical report, Department of Mathematics, Linköping University, Sweden, 1999.
3. T. F. Chan, J. A. Olkin, and D. W. Cooley. Solving quadratically constrained least squares using black box solvers. *BIT*, 32:481–495, 1992.
4. L. Eldén. Algorithms for the regularization of ill-conditioned least squares problems. *BIT*, 17:134–145, 1977.
5. H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems.* Kluwer Academic Publishers, Dordrecht, 2000.
6. W. Gander. Least squares with a quadratic constraint. *Numer. Math.*, 36:291–307, 1981.
7. G. H. Golub and C. Van Loan. *Matrix Computations, second edition.* Johns Hopkins Press, 1989.
8. G. H. Golub and U. von Matt. Quadratically constrained least squares and quadratic problems. *Numer. Math.*, 59:561–580, 1991.
9. C. W. Groetsch. *The Theory of Tikhonov Regularization for Fredholm Integral Equations of the First Kind.* Pitman, Boston, 1984.
10. M. Hanke. *Conjugate Gradient Type Methods for Ill-Posed Problems.* Pitman Research Notes in Mathematics, Longman Scientific & Technical, Harlow, Essex, 1995.
11. P. C. Hansen. Regularization tools: A Matlab package for the analysis and solution of discrete ill-posed problems. *Numerical Algorithms*, 6:1–35, 1994.

12. P. C. Hansen. *Rank-deficient and discrete ill-posed problems.* SIAM, Philadelphia, PA, 1997.

13. S. Van Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis.* SIAM, Philadelphia, 1991.

14. J. Kamm and J. G. Nagy. A total least squares method for Toeplitz systems of equations. *BIT*, 38:560–582, 1998.

15. R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods.* SIAM, Philadelphia, PA, 1998.

16. F. Rendl and H. Wolkowicz. A semidefinite framework for trust-region subproblems with applications to large-scale minimization. *Math. Programming*, 77:273–299, 1997.

17. M. Rojas. *A large-scale trust-region approach to the regularization of discrete ill-posed problems.* PhD thesis, Rice University, Houston, TX, 1998.

18. M. Rojas, S. A. Santos, and D. C. Sorensen. A new matrix-free algorithm for the large-scale trust-region subproblem. Technical Report 99-19, Dept. of Computational and Applied Mathematics, Rice University, 1999.

19. D. Sorensen. Minimization of a large-scale quadratic function subject to a spherical constraint. *SIAM J. Optim.*, 7:141–161, 1997.

20. H. Joel Trussell. Convergence criteria for iterative restoration methods. IEEE Transactions on Acoustic, Speech, and Signal Processing, Vol. ASSP-31, NO. 1, 1983.

21. H. Joel Trussell and Mehmet R. Civanlar. The feasible solution in signal restoration. IEEE Transactions on Acoustic, Speech, and Signal Processing, Vol. ASSP-32, NO. 2, 1984.

22. J. Kamm and J. G. Nagy. A total least square method for Toeplitz systems of equations. BIT, Vol. 38, No. 3, pp. 560–582, 1998.