A Large-Scale Trust-Region Approach to the Regularization of Discrete Ill-Posed Problems

Marielba Rojas

CRPC-TR98773
May 1998

Center for Research on Parallel Computation
Rice University
6100 South Main Street
CRPC - MS 41
Houston, TX 77005

Submitted September 1999; available as Rice Ph.D. thesis and as CAAM TR98-19

A Large–Scale Trust–Region Approach
to the Regularization of Discrete Ill–Posed Problems

Marielba Rojas

May 1998

TR98–19

RICE UNIVERSITY

A Large–Scale Trust–Region Approach to the Regularization of Discrete Ill–Posed Problems

by

Marielba Rojas

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Approved, Thesis Committee:

Danny C. Sorensen, Chairman
Professor of Computational and Applied Mathematics

William W. Symes
Professor of Computational and Applied Mathematics

John E. Dennis
Noah Harding Professor of Computational and Applied Mathematics

C. Sidney Burrus
Maxfield and Oshman Professor of Electrical and Computer Engineering

Houston, Texas

May, 1998

Abstract

A Large–Scale Trust–Region Approach to the Regularization of Discrete Ill–Posed Problems

by

Marielba Rojas

We consider the problem of computing the solution of large–scale discrete ill–posed problems when there is noise in the data. These problems arise in important areas such as seismic inversion, medical imaging and signal processing. We pose the problem as a quadratically constrained least squares problem and develop a method for the solution of such problem. Our method does not require factorization of the coefficient matrix, it has very low storage requirements and handles the high degree of singularities arising in discrete ill–posed problems. We present numerical results on test problems and an application of the method to a practical problem with real data.

Acknowledgments

I would like to thank my advisor Professor Danny C. Sorensen for the opportunity to work with him and benefit from his guidance and for giving me such an interesting problem to work on. Danny always set high standars for me and showed me with his example how to be a good researcher. He encouraged me to work independently and welcomed my ideas. He was patient and understanding during the process. I will always be indebted to him for all this.

I would also like to thank the other members of my committee, Professor John E. Dennis, Professor William W. Symes and Professor C. Sidney Burrus, for their interest in my work and for always being willing to help. Special thanks to Professor William W. Symes for providing me with the seismic applications and for helpful insight into that area.

It has been a privilege to share the enthusiasm for research and to benefit from the advice of all the members of my committee.

I acknowledge financial support from the Venezuelan National Science and Technology Council (CONICIT); from NSF, Grant No. EIA–9120008 Science and Technology Center for Research on Parallel Computation; and from ARPA, Grant No. ORA7453.01 A scalable Parallel Library for Numerical Linear Algebra.

During the development of this work I have benefited from the interaction with several people. I would specially like to acknowledge the collaboration with Sandra A. Santos in the initial stages of this work. I also thank Sandra for her friendship and help in difficult times.

I also wish to take this opportunity to thank Trond Steihaug for his help during this last year and his interest in my work, for contributing to my understanding of

the trust–region problem, for providing me with valuable reference material, and for sharing with me his contagious enthusiasm for academia. I also thank him for his friendship, for sharing his love for music with me, and for many enlighting conversations about life and dreams. Most of all, I would like to thank Trond for caring so much and for believing in me.

I wish to thank José Luis Farah for listening to my ideas, for his encouragement and for his friendship.

Thanks also to Keith Berrier and Trond Steihaug for reading early versions of this dissertation and for their valuable suggestions and comments.

I would like to acknowledge all the help I have received through the years from Michael Pearlman. Not only does Mike keep the computer systems running smoothly but he also makes the experience of working with computers a very pleasant one.

I have been blessed with many friends who made my stay at Rice a very nice experience even during difficult times. I would like to take this opportunity to thank all of them for their love and time.

I specially want to thank Haifa Nouaime for her friendship, support and wise advice, Elaine Dinan and Clifford Nolan for always being there for me and Gwyneth Owens Butera for being such a caring person and good friend.

I would also like to thank Graciela Lubertino, Debora Cores, Marcos Raydan, Aurelio Oliveira and Zeferino Parada who offered me their friendship and help when I came to Rice.

I also thank my friends Patricia Gómez from Venezuela and Ricardo Duarte from Brazil for keeping our friendship alive through the years and the distance.

Finally, I would like to thank my parents, Pablo and Andrea, my sister Carmen and my nephew Samuel for their love, for believing in me, and for giving me the strength to make this dream come true.

Contents

Illustrations

Tables

Chapter 1

Introduction

We are interested in the solution of discrete ill–posed problems which are ill–conditioned problems arising from the discretization of continuous ill–posed problems. For these problems the ill conditioning is an intrinsic feature that cannot be eliminated by any reformulation of the problem since it is a consequence of the properties of the underlying continuous operator.

Inverse problems are a natural source of ill–posed problems. In such problems, we want to determine the internal structure of a system from the response or behavior of the system.

An example of this kind of problem arises in oil exploration. In this field before actual drilling takes place, it is important to determine the composition of the subsurface in places where geological data or other information indicates that oil or other valuable hydrocarbons are likely to be found. The need for this information comes from the fact that drilling is a very expensive procedure.

The seismic inversion technique tries to determine the composition of the subsurface of the earth from the behavior of waves in the subsurface. To obtain data, seismic waves are generated, typically by an explosion, and their amplitude and direction are measured by receivers located on the surface.

The objective of the experiment is to recover the distribution of the slowness or inverse of wave propagation velocity in the subsurface. With this information and knowing the velocity with which waves travel in different media, experts can predict the composition of the subsurface.

Other sources of inverse problems are medical imaging and signal processing. Real applications usually give rise to large–scale discrete ill–posed problems. We are interested in the numerical treatment of such problems, for which standard methods fail to compute a good approximation to the solution. Therefore, special techniques known as *regularization methods* are needed to treat these problems.

The specific problem we want to solve is that of recovering the solution of the linear least squares problem

$$\min_{x \in \mathbb{R}^n} \quad \|Ax - b\|_2$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, for m, n large and the right–hand side b is an *exact* data vector such that $b \in \mathcal{R}(A)$, from the solution of

$$\min_{x \in \mathbb{R}^n} \quad \|Ax - \bar{b}\|_2$$

where $\bar{b} = b + \varepsilon s$, with $\varepsilon > 0$ and $s \in \mathbb{R}^m$ a random vector representing noise. Note that $\bar{b} \in \mathcal{R}(A)$ does not necessarily hold.

We pose the regularization problem as a quadratically constrained least squares problem which is equivalent to a trust–region subproblem. The use of methods for the trust–region subproblem for regularization is not common and it has been suggested by only a few authors and no efficient method for large–scale problems has been presented until now. Most of the suggested techniques do not use the known properties of the trust–region subproblem.

We develop a method for the solution of large–scale trust–region subproblems and we apply the method to the regularization of large–scale discrete ill–posed problems. Our method does not require explicit knowledge of the coefficient matrix, which is used in matrix–vector product only. The method also has very low storage requirements. Our method handles the high degree of singularities associated with discrete ill–posed problems in contrast with other methods for the large–scale trust–region subproblem that are not suitable for solving the problem in the presence of such singularities. We will present numerical results that show that our method computes solutions of large–scale trust–region subproblems at a low computational cost.

This dissertation consists of seven chapters. Chapters will be referenced by number and sections by the number of the chapter followed by a period and the section number. Subsections will be preceded by the symbol §. The content of each chapter is the following:

In Chapter 2 we present preliminary concepts, tools and results for the analysis of least squares problems.

In Chapter 3 we describe the properties of rank–deficient problems and discrete ill–posed problems and give an overview of numerical regularization methods for the solution of such problems.

In Chapter 4 we review the properties of the trust–region subproblem, establish the connection of this problem with the regularization problem and describe the special features of the problem in the discrete ill–posed case.

In Chapter 5 we present a method for the large–scale trust–region subproblem, establish the theoretical bases for the method and present preliminary numerical results when the method is used for the regularization of discrete ill–posed problems.

In Chapter 6 we present an example where we apply our method for the large–scale trust–region subproblem to the regularization of an inverse interpolation problem on real data.

In Chapter 7 we present concluding remarks and future work.

Chapter 2

Preliminaries

The purpose of this chapter is to introduce some notation and review some basic results for the least squares problem.

2.1 Notation

The following notation will be used throughout this work

- $\mathbb{R}$ will denote the real numbers.

- Unless otherwise specified, $\|\cdot\|$ will be the l_2 norm in $\mathbb{R}^n$ defined as $\|x\|_2 = \sqrt{x^T x}$, for $x \in \mathbb{R}^n$ and will also denote the induced matrix norm. The meaning will be clear from the context.

- $diag(\beta_1, \beta_2, \ldots, \beta_n)$ denotes a diagonal matrix whose diagonal elements are $\beta_i,\ i = 1, 2, \ldots, n$.

- † denotes the pseudoinverse of a matrix as defined in [25, Ch. 5].

- e_i denotes the ith canonical vector in a vector space that will be clear from the context.

- I_k denotes the identity matrix of dimension k. We will drop the subscript when the dimension is clear from the context.

- Given $A \in \mathbb{R}^{m\times n}$, $\mathcal{R}(A)$ and $\mathcal{N}(A)$ denote the range and the null space of A, respectively.

- Following [27, Chapter 9], we will use the notation $\mathcal{O}(t)$ to denote the asymptotic behavior of a real function $f(t)$. The formula

  $$f(t) = \mathcal{O}(g(t))$$

  means that there exists a constant C such that

  $$|f(t)| \leq C|g(t)|.$$

  Observe that when $f(t) = \mathcal{O}(1)$, this means that there exists a constant C such that $|f(t)| \leq C$.

2.2 The Least Squares Problem

The Least Squares (LS) problem has been thoroughly studied and is treated in most numerical linear algebra text books like [25] and [73], and also in specialized sources like the classic [42] and more recently in [4]. We state the least squares problem as

$$\min_{x \in \mathbb{R}^n} \quad \|Ax - b\|$$

where $A \in \mathbb{R}^{m \times n}$, $m \geq n$ and $b \in \mathbb{R}^m$.

Well known facts about this problem are:

- There is always a solution.

- The solution with minimum norm is unique.

- Any solution x satisfies the *Normal Equations*

  $$A^T Ax = A^T b \; . \tag{2.1}$$

2.3 Singular Value Decomposition (SVD)

The Singular Value Decomposition (SVD) of the coefficient matrix is an important tool for the analysis of the least squares problem. A decomposition $A = U\Sigma V^T$ is an SVD of A in short form if

$$U \in \mathbb{R}^{m\times n}, \quad V \in \mathbb{R}^{n\times n}$$
$$U^TU = I_n; \quad VV^T = V^TV = I_n$$
$$\Sigma = diag(\sigma_1, \ldots, \sigma_n)$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_n \geq 0$ and σ_i's are the *singular values* of A. The columns of U are the *left singular vectors* and the columns of V are the *right singular vectors* of A.

2.4 Least Squares Solution in terms of the SVD

Replacing A in (2.1) by its SVD yields

$$\begin{aligned} x &= V\Sigma^{\dagger}U^Tb \\ &= \sum_{i=1}^{r} \frac{u_i^Tb}{\sigma_i} v_i \end{aligned} \tag{2.2}$$

where r is the rank of A.

Note that (2.2) is also the expression for the solution of the linear system $Ax = b$, when A is a square nonsingular matrix.

2.5 Condition Number and Sensitivity of Least Squares Solution

We define the condition number of the matrix A with respect to the l_2 norm as

$$\kappa(A) = \frac{\sigma_1}{\sigma_r},$$

where σ_r is the smallest nonzero singular value of A.

The following result from [25, §5.3.8] provides a way of measuring the sensitivity of the solution of the least squares problem to changes in the data.

Theorem 2.1 ([25]) Suppose $x, r, \hat{x}$, and $\hat{r}$ satisfy

$$\begin{aligned} \|Ax - b\| &= \min \qquad & r &= b - Ax \\ \|(A + \delta A)\hat{x} - (b + \delta b)\| &= \min \qquad & r &= (b + \delta b) - (A + \delta A)\hat{x} \end{aligned}$$

where A and δA are in $\mathbb{R}^{m \times n}$ with $m \geq n$ and $0 \neq b$ and b are in $\mathbb{R}^{m}$. If

$$\epsilon = \max\left\{\frac{\|\delta A\|}{\|A\|}, \frac{\|\delta b\|}{\|b\|}\right\} < \frac{\sigma_r}{\sigma_1}$$

and

$$\sin(\theta) = \frac{\|r\|}{\|b\|} \neq 1,$$

then

$$\begin{aligned} \frac{\|\hat{x} - x\|}{\|x\|} &\leq \epsilon\left\{\frac{2\kappa(A)}{\cos(\theta)} + \tan(\theta)\kappa(A)^2\right\} + \mathcal{O}(\epsilon^2) & (2.3) \\ \frac{\|\hat{r} - r\|}{\|r\|} &\leq \epsilon(1 + 2\kappa(A))\min(1, m - n) + \mathcal{O}(\epsilon^2)\ . & (2.4) \end{aligned}$$

Proof See [25]. □

The right–hand side in (2.3) is proportional to the condition number of the matrix in the zero residual case, and to the *square* of the condition number in the nonzero residual case. A large condition number implies that the least squares solution will be unstable, i.e. very sensitive to perturbations in the data A, b. Such problems are called *ill–conditioned* problems.

Chapter 3

Numerical Regularization

The term numerical regularization encompasses a set of techniques for the numerical treatment of ill–conditioned problems, i.e. problems in which the coefficient matrix has a large condition number.

When we encounter ill–conditioned systems or least squares problems in practice, the usual recommendation is not to trust any solution obtained with the standard methods and to try to reformulate the problem in order to eliminate the ill conditioning.

There are cases however, for which the ill conditioning is an intrinsic feature of the problem. As a consequence, there is no reformulation of the problem that can eliminate the ill conditioning. This is the case for discrete ill–posed problems. In this work we are interested in numerical regularization techniques for the solution of large–scale discrete ill–posed problems.

In this chapter we describe the properties of ill–conditioned problems and review the available methods to treat these problems. In Section 3.1 we describe two classes of ill–conditioned problems, namely, rank–deficient and discrete ill–posed problems. In Section 3.2 we give an overview of the area of numerical regularization including the main approaches to regularization and the concepts of regularized solution and regularization parameter. In Section 3.3 we described the known criteria for estimating the regularization parameter. In Section 3.4 we present a summary of the main regularization methods for large–scale problems.

3.1 Ill–conditioned problems

We follow Hansen's classification of ill–conditioned problems in rank–deficient and discrete ill–posed problems [37]. We describe rank–deficient problems in §3.1.1 and discrete ill–posed problems in §3.1.2.

3.1.1 Rank–Deficient Problems

The main feature of this kind of problems is that there exists a clear gap between the small and large singular values of the coefficient matrices. Therefore, the determination of the numerical rank for these matrices is a well–conditioned problem.

Let A be the coefficient matrix for a rank–deficient problem, then we can summarize the properties of the problem as:

- The small singular values of A are well separated from the remaining singular values.
- The matrix A usually has a small cluster of small singular values.
- The problem of computing the numerical rank of the coefficient matrix is well conditioned, since the small singular values of the matrix are clearly separated from the rest.
- There is usually a reformulation that will eliminate the ill conditioning.

In Figure 3.1 we show a typical example of the singular value distribution for a problem of this type. We can observe in the figure the separation of the small singular values from the rest.

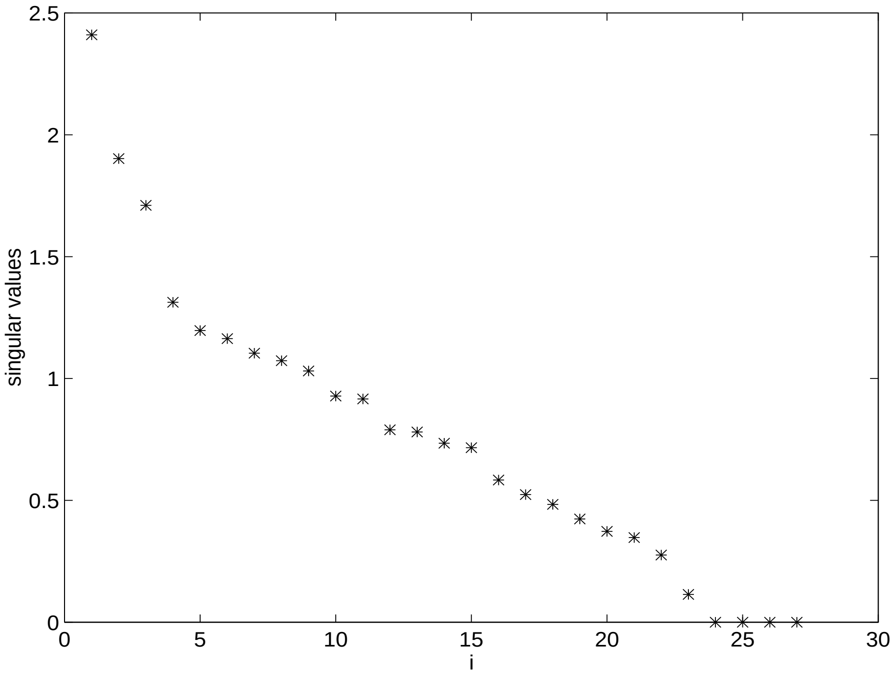

Figure 3.1 Distribution of Singular Values for a Rank–Deficient Problem.

3.1.2 Discrete Ill–Posed Problems

In 1923 Hadamard [28] introduced the concept of a well–posed problem as a problem for which there exists a solution, the solution is unique and the solution depends continuously on the data. If a problem did not satisfy any of these conditions he would say that the problem was ill posed, meaning that its formulation was incorrect. Hadamard believed that these problems would not arise in practice.

Since Hadamard's work however, ill–posed problems have appeared in practice mostly in connection with inverse problems in different areas such as seismic inversion, image processing and signal processing. The computational solution of such problems requires the discretization of the involved operators and functions and give rise to discrete ill–posed problems.

Let $A \in \mathbb{R}^{m \times n}$ come from the discretization of a continuous operator from an ill–posed problem. If we use a reasonably good discretization the matrix A will have properties that are the discrete counterpart of those of the underlying continuous operator.

We will discuss the properties of interest for a very common class of inverse problems, namely, Fredholm integral equations of the first kind with a square integrable kernel. We will follow [37] in this discussion. We can write an equation of this kind as

$$\int_0^1 K(s,t)f(t)dt = g(s), \quad 0 \le s \le 1, \tag{3.1}$$

where K is the kernel, g is the right–hand side and f is the unknown solution.

The main tool for the analysis of equations like (3.1) is the Singular Value Expansion of the kernel K. Any square integrable kernel K can be written as

$$K(s,t) = \sum_{i=1}^{\infty} \mu_i p_i(s) q_i(t),$$

where p_i, q_i are the singular functions of K and μ_i are the singular values of K. The singular functions are orthonormal with respect to the following inner product

$$\langle \phi, \psi \rangle = \int_0^1 \phi(t)\psi(t)dt.$$

A function f is square integrable if $\|f\| = \langle f, f \rangle^{\frac{1}{2}}$ exists.

Given a singular value expansion of K, a solution of (3.1) can be written as

$$f(t) = \sum_{i=1}^{\infty} \frac{\langle p_i, g \rangle}{\mu_i} q_i(t) \; \cdot$$

In order for a function f to be a square integrable solution of (3.1), it is necessary and sufficient that g satisfies the following condition [33].

Picard Condition (PC). The right–hand side g in (3.1) satisfies the PC if

$$\sum_{i=1}^{\infty} \left| \frac{\langle p_i, g \rangle}{\mu_i} \right|^2 < \infty, \quad \mu_i \neq 0 \; . \tag{3.2}$$

Suppose now that we use a reasonably good discretization to obtain $A \in \mathbb{R}^{m\times n}$, a discrete version of K. Let $A = U\Sigma V^T$ be a Singular Value Decomposition of A. Then we can summarize the properties of this discrete ill–posed problem as follows:

- The matrix A is ill–conditioned.
- The singular values of the matrix A decay gradually to zero.
- If the continuous problem satisfies the Picard Condition, then u_i^Tb decay gradually to zero as the index i increases.
- If u_i and v_i are the singular vectors of the matrix A, then u_i and v_i often have more sign changes in their components as the singular values σ_i decrease, i.e. the high–frequency components correspond to small singular values.
- There is usually no reformulation that can change these features.

In Figure 3.2 we present an example of the singular value distribution for a problem of this type (problem **heat** from the Regularization Tools package [36]). We can observe that there is no clear separation between the small singular values and the rest.

For the problems considered in this work we will assume that the singular values of the discrete operator A, are nonzero, although some of them can be very small. Therefore, A has rank n, and we can write the solution of $\min \|Ax - b\|$, $x \in \mathbb{R}^n$ in terms of the SVD of A as

$$x = \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i} v_i. \tag{3.3}$$

The norm of the least squares solution (3.3) is given by

$$\|x\| = \left[\sum_{i=1}^{n} \left(\frac{u_i^T b}{\sigma_i} \right)^2 \right]^{\frac{1}{2}}.$$

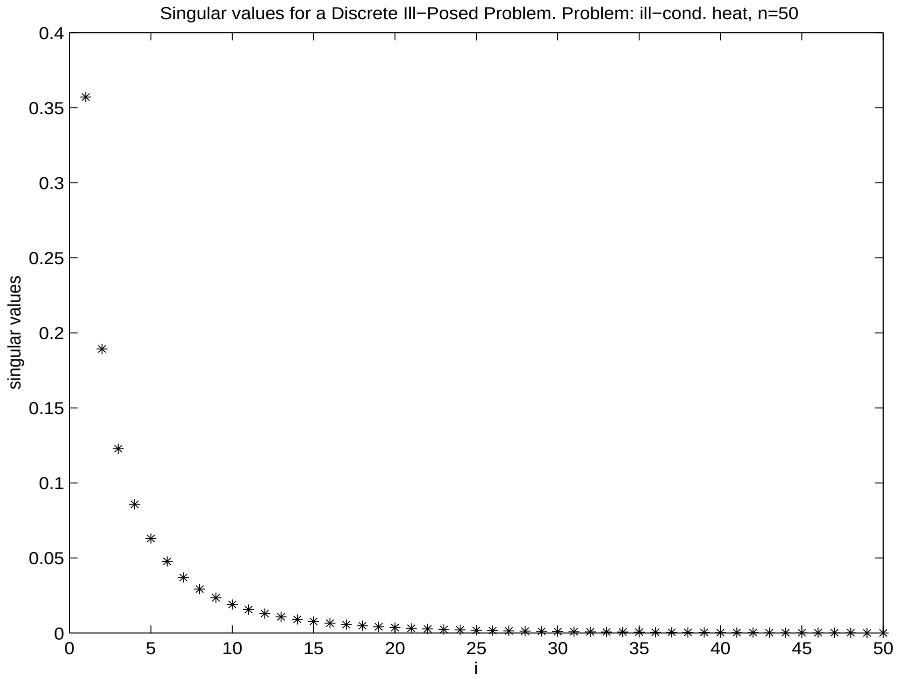

Figure 3.2 Distribution of Singular Values
for a Discrete Ill–Posed Problem.

Therefore, this norm will not be too large as long as $\mid u_i^T b \mid < \sigma_i$ for small σ_i.

In view of this observation and in connection with discrete ill–posed problems, the following condition is usually assumed.

Discrete Picard Condition (DPC)

$u_i^T b$ *on average decay to zero faster than* σ_i as the index i increases.

Hansen in [34] justified the need for this condition in order for some regularization methods, such as Tikhonov regularization and Truncated SVD, to be effective.

Varah in [76] observed that even when the DPC does not hold there might still be a solution in terms of a basis different from the basis of singular vectors. However, finding such basis is not an easy task.

The DPC is convenient in the sense that it indicates when it is effective to apply one of the standard regularization methods.

Note that the DPC is analogous to the Picard Condition for the continuous problem. For a further discussion of the relationship between the continuous and discrete operators of ill–posed problems we refer the reader to [33] and [37, Ch. 1].

As we mentioned in Chapter 1, we are interested in recovering x_*, the solution of

$$\min_{x \in \mathbb{R}^n} \quad \|Ax - b\| \tag{3.4}$$

from $\bar{x}_*$, the solution of

$$\min_{x \in \mathbb{R}^n} \quad \|Ax - \bar{b}\| \tag{3.5}$$

where $\bar{b} = b + \varepsilon s$, with $\varepsilon > 0$ and $s \in \mathbb{R}^m$ a random vector representing noise. In other words, we would like to solve problem (3.4) but instead of the exact right–hand side b, we have a perturbed right–hand side $\bar{b}$, and therefore the problem that we have is (3.5). The goal is to recover as much information as possible about x_* from $\bar{x}_*$.

The solution $\bar{x}_*$ of (3.5) is given by

$$\bar{x}_* = \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i} v_i + \sum_{i=1}^{n} \frac{\varepsilon u_i^T s}{\sigma_i} v_i \cdot$$

Thus, $\bar{x}_*$ consists of two terms, the first one is the actual solution of the unperturbed problem (3.4) and the second one is the contribution from the noise.

The difficulty in finding x_* from $\bar{x}_*$ stems from the contribution from the noise. If s is a vector of uncorrelated noise, $u_i^T s$ will be constant for all i, causing the ratios $\frac{u_i^T \bar{b}}{\sigma_i}$ to blow up. Figure 3.3 illustrates the behavior of $\frac{u_i^T b}{\sigma_i}$ and $\frac{u_i^T \bar{b}}{\sigma_i}$. In Figure 3.4 we show that in this case, $\bar{x}_*$ can differ considerably from x_*. In this particular example the relative error in the solution for noisy data with respect to the solution for exact data is 565.28. We can also observe in Figure 3.4 how $\bar{x}_*$ is dominated by high–frequency components. The example in Figures 3.3 and 3.4 is problem **deriv2**, $(m = n = 50)$ from [36].

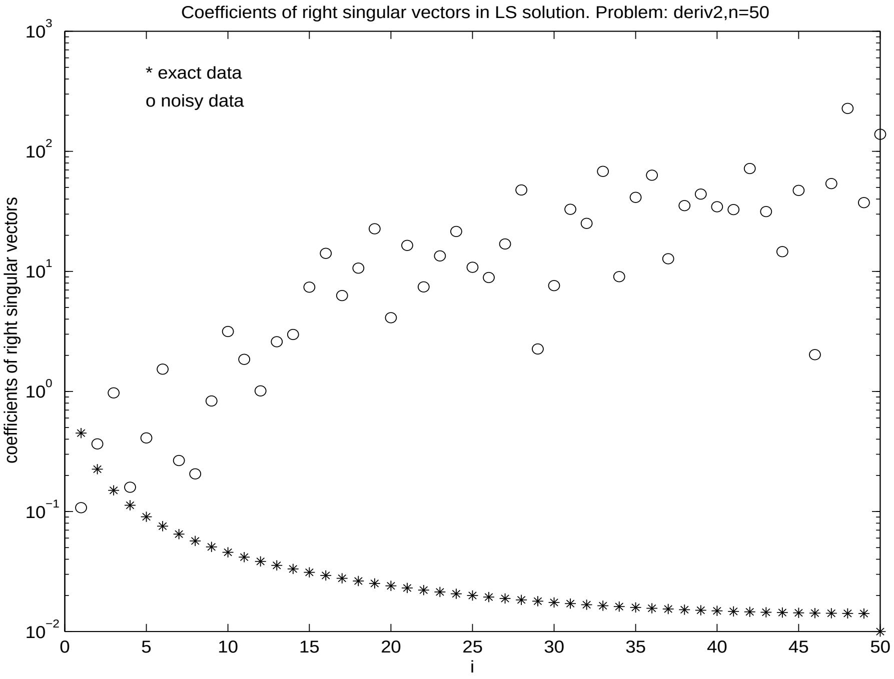

Figure 3.3 Coefficients of Right Singular Vectors in Least Squares solution, for exact and noisy data.

In the previous analysis we considered errors (noise) only in the right–hand side. In this work we will assume that the errors in the coefficient matrix, due for example to discretization or finite precision representation, are small in comparison with errors in the right–hand side. To take into account errors in both the coefficient matrix and the right–hand side, we should pose the problem as a Total Least Squares problem (see [75]), which is considerably more difficult and will not be treated in this work. The Total Least Squares problem has been used in connection with regularization in [17].

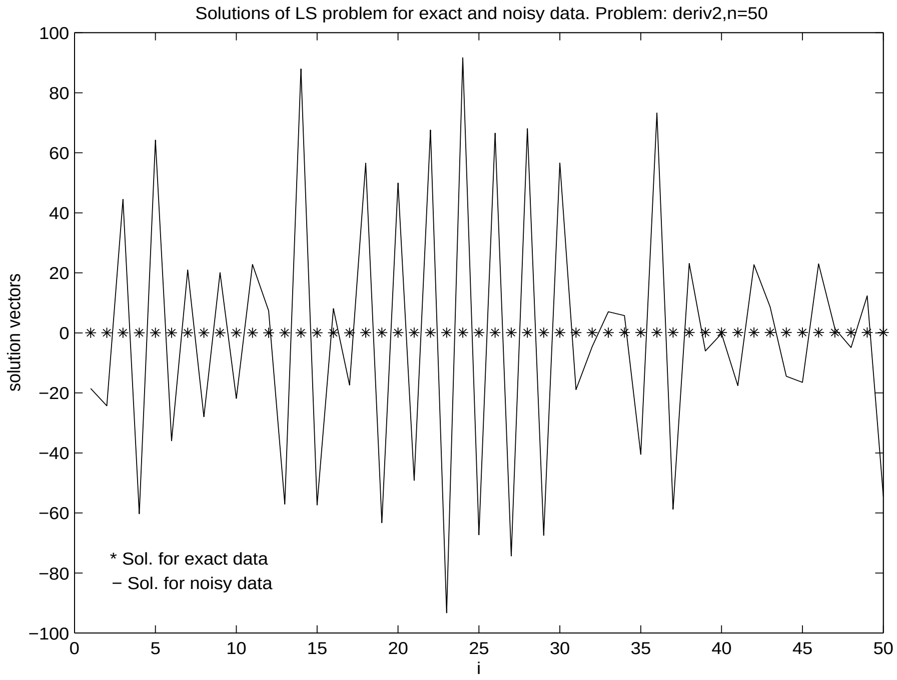

Figure 3.4 Solutions of a Least Squares Problem for exact and noisy data.

3.2 An Introduction to Numerical Regularization

Regularization is a technique that was originally devised for continuous ill–posed problems [74]. When the technique is extended to the discrete context it gives rise to *numerical regularization methods.* We limit our presentation to the latter case.

Numerical regularization methods seek to approximate the unknown solution of an ill–conditioned problem with exact data, by the solution of a related well–conditioned problem with perturbed data. The formulation of the well–conditioned problem includes information about the desired solution.

The additional information is usually expressed as a constraint on the solution of the least squares problem. Such constraint is of the form

$$\|Lx\| \leq \Delta \tag{3.6}$$

where L is typically the identity or a discretized derivative operator. In the latter case, the constraint is used to control the smoothness of the approximate solution. When $L = I$, the constraint controls the size of the solution.

In the following we assume that $L = I$, whenever we use a constraint of the form (3.6). If the problem is given in general form ($L \neq I$) it is possible to transform it to the standard form ($L = I$) by means of the algorithms given in [14] and [37], or by a change of variable in case L is invertible.

Lin and Moré [45] have recently proposed a method that can be applied to quadratically constrained least squares problems, where the constraint is of the form (3.6). They assumed that if $L \neq I$ then L is invertible and they can apply a change of variables. They regard L as a scaling matrix that they compute so that it clusters the eigenvalues of $L^{-1}A^TAL$. This feature prevents the use of this method for discrete ill–posed problems since the strategy might change the spectrum of A^TA in undesirable ways. We will return to this topic in §3.4.3 when we discuss preconditioning in the context of discrete ill–posed problems.

To justify the use of (3.6), we must assume that the exact solution is smooth or that it is expected to have small norm. Regularization is also known as *smoothing* because it tries to damp nonsmooth components in the approximate solution. Note that as we discussed in §3.1.2, for discrete ill–posed problems, nonsmooth or high–frequency components usually correspond to small singular values of the coefficient matrix. Therefore, those components are magnified by the noise as we can observe in Figure 3.4.

We can also regard the regularization problem as a multi–objective optimization problem where we try to balance the accuracy of the approximate solution and the effect of the noise on the solution. This approach is taken in [40].

Every regularization method uses a parameter to control the effect that the noise in the data has on the approximate solution. This parameter is called the *regularization parameter* and its nature is different for each method. For example, in methods based on constrained minimization problems the parameter will be the Lagrange multiplier; in Tikhonov regularization it will be the penalty parameter; and in some iterative methods the number of iterations will play the role of the regularization parameter.

A complete regularization method has two aspects: the computation of the regularized solution and the computation of the regularization parameter. Some regularization methods are based on a priori estimate of the parameter followed by the computation of a regularized solution, others use a posteriori estimate of the parameter based on several approximate solutions. Yet another approach is a hybrid one, where the solution and the optimal regularization parameter are determined at the same time. The method of Björck, Grimme and Van Dooren [6] is the only one in this class. We discuss this method in §3.4.4.

In the rest of this section we summarize the main regularization techniques for both rank–deficient and discrete ill–posed problems. In Section 3.4 we concentrate on methods for the large–scale case. We present methods that are suitable for ill–conditioned linear systems and least squares problems. The nonlinear least squares case is treated in [77] and [16], for example.

For a more detailed description of the methods, we refer the reader to [37], [31] and [15] and the references therein. Early surveys of regularization methods appeared in [14], [5] and [76]. A common framework for the study of numerical regularization methods is proposed in [37] and [38], and more recently in [58].

3.2.1 Methods for Rank–Deficient Problems

As we mentioned before, the coefficient matrix of a rank–deficient problem has a singular value spectrum such that the smallest singular values are clearly separated from the rest. This feature makes it possible to determine the *numerical rank* of the matrix.

The *numerical ε–rank* $r_\varepsilon(A)$ of a matrix $A \in \mathbb{R}^{m\times n}$ is the number of columns that are numerically linearly dependent with respect to an error level ε, and it is defined as

$$r_\varepsilon(A) = \min_{||E|| \leq \varepsilon} \quad rank(A+E) \ .$$

In terms of the singular values of A we have the following relationship:

$$\sigma_{r_\varepsilon(A)} > \varepsilon \geq \sigma_{r_{\varepsilon(A)}+1} \ .$$

If $r_\varepsilon(A)$ is ill–determined, i.e. if it is too sensitive to perturbations on ε and on the singular values, then it is better to use regularization methods for problems with no gap in the singular spectrum.

The problem in consideration is

$$\min_{x \in \mathbb{R}^n} \quad ||Ax - \bar{b}||$$

where $\bar{b}$ contains noise.

The most common regularization strategy for rank–deficient problems consists of two steps

1. Replace the matrix A by a matrix of rank $k = r_\varepsilon(A)$, i.e. ignore the small singular values of A. The usual choice for this rank–k matrix is

$$A_k = \sum_{i=1}^{k} u_i \sigma_i v_i^T \ .$$

A_k is the closest rank–k matrix to A in the l_2 norm and in the Frobenius norm (see [25, Ch. 2, Theorem 2.5.2]).

2. Compute the approximate solution by

$$x_k = \sum_{i=1}^{k} \frac{u_i^T \bar{b}}{\sigma_i} v_i$$

which is known as the *truncated SVD solution.* This regularization strategy is known as *truncated SVD.*

Depending on the size of the problem, the SVD might be expensive to compute. An alternative approach in practice is to compute rank–revealing decompositions like the QR decomposition with column pivoting ([25, Ch. 5]). In this case the matrix we use to replace A in step 1 above is still close to A, although there are no results that characterize it as the closest one in some norm.

Even when less expensive decompositions than the SVD are used, the truncated SVD method is intended for small to medium–scale problems only.

Although the truncated SVD method does not use additional information about the desired solution, the method is still regarded as a regularization method because it stabilizes (*regularizes*) the solution, making it less sensitive to changes in the data.

3.2.2 Methods for Discrete Ill–Posed Problems

As we discussed before, posing a regularization problem involves the design of a new problem that includes some information about the desired solution. We present next the discrete version of Hansen's [37] most common regularization problems.

Let $A \in \mathbb{R}^{m \times n}$, $\bar{b} \in \mathbb{R}^m$.

The first problem is of the form

$$\begin{array}{ll} \min & \|Ax - \bar{b}\| \\ s.t. & x \in S \end{array} \tag{3.7}$$

where $S \subset \mathbb{R}^n$.

The second approach consists of solving the following quadratically constrained least squares problem

$$\begin{array}{ll} \min & \|Ax - \bar{b}\| \\ s.t. & {\scriptstyle \|Lx\| \leq \Delta} \end{array} \tag{3.8}$$

where $\Delta > 0$.

Another possible problem is

$$\begin{array}{ll} \min & \|x\| \\ s.t. & {\scriptstyle \|Ax - \bar{b}\| \leq \beta} \end{array} \tag{3.9}$$

where β is an estimate of the noise level in the data.

The last of the approaches is the most common regularization method, namely Tikhonov regularization where we solve the problem

$$\min \quad \|Ax - \bar{b}\|^2 + \lambda^2 \|Lx\|^2. \tag{3.10}$$

Alternative formulations for the Tikhonov regularization problem are

$$(A^T A + \lambda^2 I)x = A^T \bar{b}$$

i.e. find a zero for the gradient of the objective function in (3.10), and the damped least squares problem

$$\min_{x \in \mathbb{R}^n} \quad \left\| \begin{pmatrix} A \\ \lambda I \end{pmatrix} x - \begin{pmatrix} \bar{b} \\ 0 \end{pmatrix} \right\|_2 .$$

We will show in Chapter 4 that problem (3.8) is equivalent to a trust–region subproblem and we will use this regularization approach. Sorensen [69] and Gay

[20] derived a charaterization of the global solutions of this problem. Gander in [19] discussed the theoretical aspects of problems (3.8) and (3.9) in the ill–conditioned context.

We classify the regularization problems according to their sizes in small to medium–scale problems and large–scale problems. In this work we are interested in large–scale problems.

The solution of small to medium–scale discrete ill–posed problems can be obtained by means of direct methods since in this case it is affordable to compute factorizations of the matrices involved. Therefore we can use any direct method that solves one of the problems (3.7), (3.8), (3.9) or (3.10). One of the direct methods that we can used for solving problem (3.8) is the one presented by Moré and Sorensen [55]. Other direct methods for quadratically constrained problems are also discussed in Golub and Van Loan [25, Ch. 12]. For general direct regularization methods see for example [37] and the references therein.

The solution of large–scale discrete ill–posed problems requires iterative techniques both because of storage limitations and because the coefficient matrices are usually not available explicitly. In this case, we can use any iterative method that solves one of the problems (3.7), (3.8), (3.9) or (3.10). We summarize the main methods for the regularization of large–scale discrete ill–posed problems in Section 3.4. For other methods see [37] and the references therein. Before describing the methods, we review the main approaches for computing the regularization parameter in the next section.

3.3 Computation of the Regularization Parameter

Let us recall the multi–objective purpose of regularization, i.e. minimize the residual norm while minimizing the effect of perturbations in the data. Based on this observation we can identify two kinds of errors in any regularized solution: the *regularization error* introduced when minimizing the effect of perturbations in the data which usually causes the loss of meaningful information, and the *perturbation error* which is the error in the solution of the problem with perturbed data with respect to the solution of the problem with exact data. Regularization techniques try to balance these two kinds of error usually by means of a parameter known as the regularization parameter whose computation is also challenging.

There are two kinds of criteria for choosing the regularization parameter. Criteria in the first class assume some knowledge of the noise level in the data. The only criterion in this class is the Discrepancy Principle. The second class of criteria try to estimate the parameter assuming no knowledge of the noise level, but require several approximate solutions in order to estimate the regularization parameter. The Generalized Cross–Validation criterion and the L–curve criterion belong to the latter class.

3.3.1 Discrepancy Principle Criterion

When we know a bound for β, the noise level in the data, we can use this bound to implicitly compute a regularization parameter as the point where $\|Ax - \bar{b}\| \leq \beta$. The idea behind this criterion is that we cannot expect more accuracy in the approximate solution than the one present in the data. This criterion is attributed to Morozov [56].

3.3.2 Generalized Cross–Validation Criterion

This criterion was presented in [24] and discussed also in [79]. The idea of using cross–validation to compute the regularization parameter is the following. If we exclude a data point $\bar{b}_i$ and compute a regularized solution $\bar{x}_{reg,i}$ based on the reduced data vector, then if we use $\bar{x}_{reg,i}$ to compute an estimate of $\bar{b}_i$ we want this estimate to be good. While in ordinary cross–validation the ordering of the data counts, generalized cross–validation is invariant to orthogonal transformations of the data vector.

The regularization parameter is chosen as the minimizer of the Generalized Cross–Validation (GCV) function

$$GCV(\lambda) = \frac{\|A\bar{x}_{reg} - \bar{b}\|^2}{[trace(I - AA^{\#}(\lambda))]^2}$$

where $A^{\#}(\lambda)$ is the matrix that maps the data vector $\bar{b}$ onto the regularized solution $\bar{x}_{reg}$.

There are several difficulties associated with the GCV criterion. One is that sometimes the minimum of the function is difficult to compute numerically. The second one is that sometimes GCV cannot distinguish between the signal and correlated noise. A third problem is that the matrix $A^{\#}(\lambda)$ may not be available explicitly or may be difficult to compute, depending on the method we are using to compute the regularized solution. An example of this is the Conjugate Gradient Method on the Normal Equations (see 3.4.2).

3.3.3 L–curve Criterion

This criterion is based on the trade–off curve between the two goals of regularization: minimizing the residual norm while keeping the solution norm from being too large. The L–curve is a logarithmic plot of the solution norm versus the residual norm for each value of the regularization parameter. The name comes from the fact that this curve is L–shaped.

The optimal regularization parameter gives a solution that lies around the "corner" of the curve. Figure 3.5 shows the L–curve (and its corner) for problem **heat** from [36]. In this example, the curve is based on the values of the regularization parameter in Tikhonov regularization.

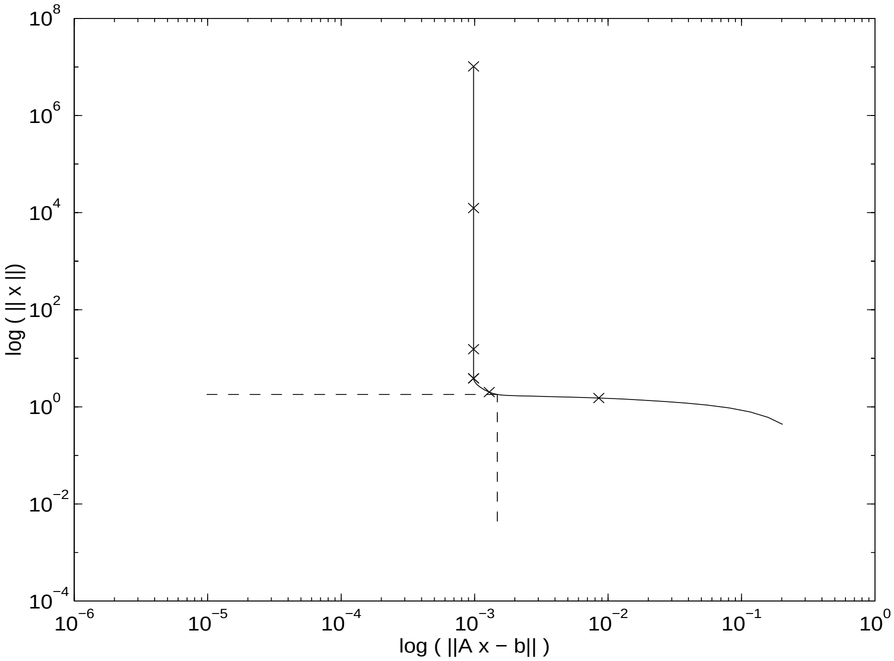

Figure 3.5 The L–curve for Tikhonov Regularization.

The use of this curve to estimate the regularization parameter has been studied in [35] and [38]. The idea is to interpolate the curve in order to estimate the “corner”. The L–curve criterion performs better than the GCV criterion when the noise in the data is correlated and comparably well for white noise. The advantage of the L–curve criterion over the GCV in the presence of correlated noise seems to come from the fact that the L–curve criterion uses information on both the residual norm and the solution norm, while the GCV criterion uses information on the residual norm only (see [38]).

A major drawback of the criterion is the need for several points on the curve to build the interpolant since the computation of such points may be expensive. Hanke

[30] and Vogel [78] point out other disadvantages of using this criterion to estimate the regularization parameter.

3.4 Regularization of Large–Scale Discrete Ill–Posed Problems

In this section we review the following methods for the regularization of large–scale discrete ill–posed problems: Landweber iteration in §3.4.1, Conjugate Gradient on the normal equations (CGLS) in §3.4.2, the method of Björck, Grimme and Van Dooren [6] in §3.4.4, the method of Golub and von Matt in §3.4.5 [26] and the method of Calvetti, Reichel and Zhang [8], [9] in §3.4.6. In §3.4.3 we discuss the preconditioning issue in the context of discrete ill–posed problems.

We recall that the problem we want to solve is that of recovering the solution of $\min \|Ax - b\|$, $x \in \mathbb{R}^n$ from the solution of $\min \|Ax - \bar{b}\|$, $x \in \mathbb{R}^n$, where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $\bar{b} = b + \varepsilon s$.

3.4.1 Landweber Iteration

This is an iterative method for solving the normal equations, in which the iterates are computed as

$$x_{k+1} = x_k + \omega A^T(\bar{b} - Ax_k) \quad k = 0, 1, 2, \ldots$$

where ω is a parameter. This method is not used in practice in its original form since it is not very efficient. A modification has been proposed in [29] to accelerate convergence.

3.4.2 The method of Conjugate Gradient on the Normal Equations

This approach consists of applying the Conjugate Gradient method to the normal equations

$$A^T A x = A^T \bar{b} \; .$$

An implementation of this method should avoid forming the matrix $A^T A$ since doing so may introduce large rounding errors (see [25, example 5.3.2 on p.225]). The resulting method is known as Conjugate Gradient on the Normal Equations (CGNR, CGLS). The method has been used successfully to solve some ill–conditioned least squares problems with noisy data due to an intrinsic regularization effect of the iteration.

To give an insight into this regularization effect, we first observe that the Conjugate Gradient method generates iterates in a Krylov subspace. In particular, for the CGLS method the iterates x_k belong to the Krylov subspace $\mathcal{K}_k(A^T A, A^T \bar{b})$ defined as

$$\mathcal{K}_k(A^T A, A^T \bar{b}) = \{A^T \bar{b}, (A^T A) A^T \bar{b}, \ldots, (A^T A)^{k-1} A^T \bar{b}\} \; .$$

Note that at each iteration, x_k is the solution of the following minimization problem

$$\begin{array}{ll} \min & \|Ax - \bar{b}\| \; . \\ s.t. & {\scriptstyle x \in \mathcal{K}_k(A^T A, A^T \bar{b})} \end{array}$$

Therefore by using the CGLS method we are solving a regularization problem of the form (3.7).

For some problems, $\mathcal{K}_k(A^T A, A^T \bar{b})$ approximates the subspace spanned by the right singular vectors v_i associated with the k largest singular values and therefore, x_k has components only in the direction of singular vectors associated with large singular values. As k increases however, singular vectors associated with small singular values enter in the approximation. This causes that contributions from the noise appear in the iterates which start to diverge from that point on. This behavior is known as *semiconvergence* and we illustrate it in Figure 3.6.

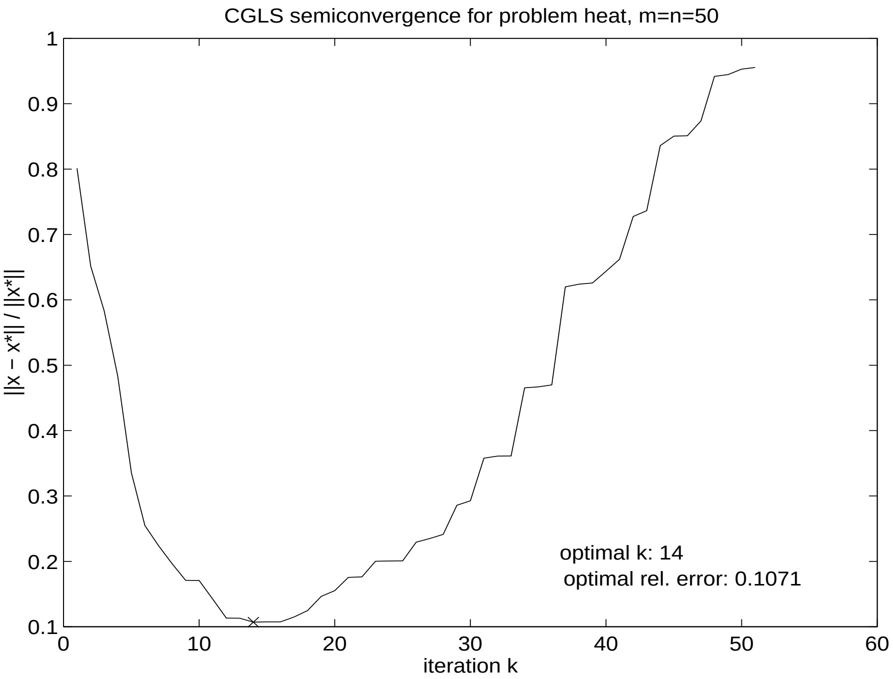

Figure 3.6 Semiconvergence behavior of the Conjugate Gradient Method on the Normal Equation (CGLS) on Discrete Ill–Posed Problems.

The semiconvergence behavior makes it necessary to stop the iteration before the effect of the noise appears. The number of iterations k plays the role of the regularization parameter in this case. The method is very sensitive to the value of k at which we stop, since at any particular iteration there is no warranty that *all* the singular vectors of interest have already converged when the iterates start to diverge, nor is there any warranty that *only* the singular vectors of interest have been computed. Two options for estimating k that were mentioned in [37] are the L–curve criterion and the Monte Carlo Cross–Validation procedure from [21]. As we discussed before, the method of Björck, Grimme and van Dooren [6] uses a Generalized Cross–Validation approach to estimate k.

In some cases, like for some problems arising in medical imaging (see [57], for example), it is known that all the singular vectors of interest converge first, so these problems are less sensitive to the choice of the maximum number of iterations k. The difficulty in those applications comes from the fact that the number of large singular values is large, requiring the design of efficient preconditioners to accelerate convergence.

The CGLS method is suitable for large–scale problems since it uses A and A^T only in matrix–vector products, it requires very little storage: five vectors, and it has fast convergence. The major disadvantage is the semiconvergence behavior that makes it necessary to have an accurate estimate of when to stop the iteration.

3.4.3 Preconditioned CGLS

The issue of preconditioning in connection with ill–conditioned problems is very delicate. The usual goal of preconditioning is to improve the condition of the coefficient matrix by either clustering its eigenvalues or making them close to one. In the ill–conditioned case however, it is not desirable to change the whole spectrum of the matrix.

If we precondition all the eigenvalues of A^TA (and therefore, the singular values of A) we would be computing iterates that contain contributions from the noise, since the preconditioner has probably mixed the small and large parts of the spectrum. Thus, in the ill–conditioned case we should precondition only the large part of the spectrum and leave the small part untouched. This fact has been observed before in [31], [37, Ch. 5], [32] and [57].

In general, it is not possible to distinguish a priori between the large and small parts of the spectrum of a matrix. This makes preconditioning for ill–conditioned problems a very difficult area and the object of current research.

For problems where we can identify large and small parts of the spectrum, it has been possible to build efficient preconditioners as the ones reported in [12] and [57]. These problems are characterized by having a highly structured coefficient matrix (Toeplitz matrix) for which circulant preconditioners have proved to be successful. Multilevel preconditioners for more general problems have been proposed in [41].

3.4.4 The method of Björck, Grimme and van Dooren

This method uses the normal equations approach for the solution of the least squares problems. The computation is based on the Lanczos Bidiagonalization process and on the regularization of the approximate solution in a Krylov subspace.

The method uses an Implicitly Restarted Lanczos Bidiagonalization process to compute a sequence of approximations to the left and right singular vectors of A. The implicit restart technique from [70] is adapted to the Lanczos Bidiagonalization process [59] and zero shifts are used to filter out small singular values.

In this method, full reorthogonalization of the Lanczos vectors is necessary at every step of the Lanczos process. The reorthogonalization makes it possible to use the Generalized Cross–Validation (GCV) function to estimate the regularization parameter k (in this case, the size of the factorization) when the noise level in the data is unknown.

The approximate solution computed by this method is a Truncated Singular Value Decomposition (TSVD) solution in a Krylov subspace of dimension k. This method can be used for the regularization of large–scale problems since it uses A and A^T only in matrix–vector products. The method is efficient when the minimum of the GCV function occurs at small values of k.

However, the method has several drawbacks. The first one is that it is necessary to store the two matrices U_k and V_k, with k unknown a priori. The second one is that the minimum of the GCV function can occur at a large value of k, as in the example in Figure 3.7 corresponding to problem **heat** from [36]. The bidiagonalization of the matrix becomes very expensive in this case, in both storage and computation. For this method, the increase in computational cost is higher than it would usually be, due to the reorthogonalization step. A third difficulty is that, as we discussed in §3.3.2, the GCV criterion for computing the regularization parameter is not reliable in certain cases.

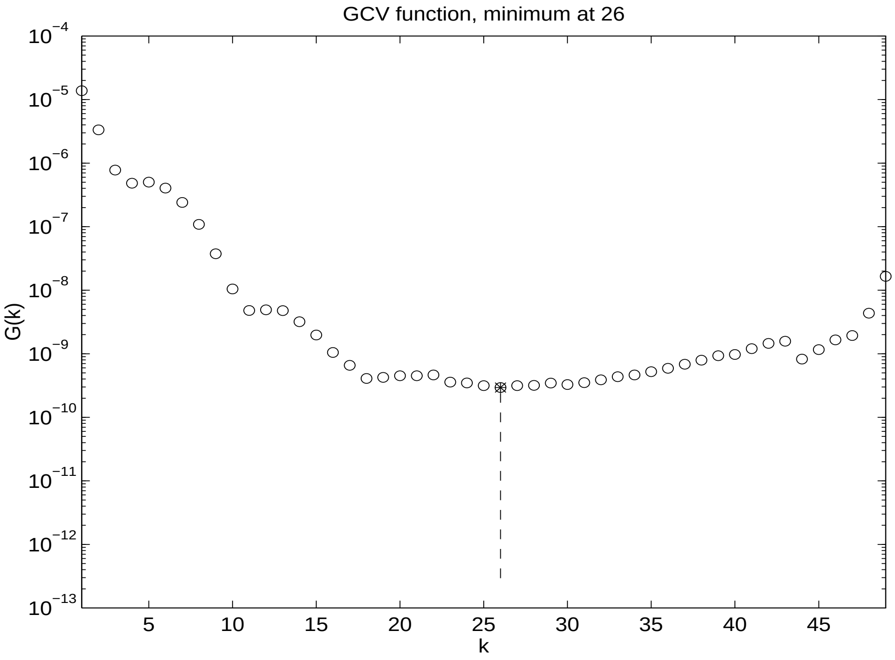

Figure 3.7 Generalized Cross–Validation function.

3.4.5 The method of Golub and von Matt

This method, presented in [26], poses the regularization problem as the following quadratically *equality* constrained least squares problem

$$\begin{array}{ll} \min & \|Ax - \bar{b}\| \\ s.t. & \|x\| = \Delta \end{array}$$

where $\bar{b}$ is a vector of noisy data.

The method relies upon the Lanczos Bidiagonalization [59] of the matrix A and the Cholesky factorization of small matrices, to compute a sequence of function pairs $(L_k(\lambda), U_k(\lambda))$ that bound the secular function

$$f(\lambda) = (A^T\bar{b})^T[(A^TA + \lambda I)^{\dagger}]^2(A^T\bar{b}) \ .$$

A zero–finding procedure is then applied to $L_k(\lambda)$ and $U_k(\lambda)$ to compute a new estimate for λ, the Lagrange multiplier associated with the minimization problem.

Among the advantages of this method are that it uses A and A^T only in matrix–vector products, and that the approximate solution x can be computed in an inexpensive way after the optimal λ has been found, since a partial bidiagonalization of A is available at that point.

We point out two main disadvantages associated with the method. The first one is that it is necessary to store the two matrices U_k and V_k, where k is not known a priori.

The second one is that the method works under the assumption that $\Delta < \|A^{\dagger}\bar{b}\|$. This assumption is not enough in the regularization context since it might still produce a solution with large norm. Instead, in order to use the method for regularization purposes we must ensure that Δ is such that $\Delta \leq \|A^{\dagger}b\| < \|A^{\dagger}\bar{b}\|$, where b is the exact data vector. This requires knowledge on the norm of the unknown solution $A^{\dagger}b$ of the unperturbed unconstrained problem.

3.4.6 The method of Calvetti, Reichel and Zhang

This method was presented in [8] and [9]. The method considers the problem of solving large ill–conditioned systems of equations. The method is based on expressing the regularized solution x_λ, which depends on the regularization parameter λ, as

$$x_\lambda = \varphi_\lambda(A)A^\dagger \bar{b},$$

where $\varphi_\lambda(A)$ is a polynomial in A that can be regarded as a filter function. The idea is to select an appropriate φ_λ to obtain a regularized solution.

The filter function is chosen by expanding known filter functions, like the Tikhonov filter function, in a basis of orthogonal polynomials and computing terms in the expansion until a prescribed accuracy is achieved.

The regularization parameter λ is computed in a very elegant way, by solving the following nonlinear equation in λ

$$||Ax_\lambda - \bar{b}||^2 - \beta^2 = 0,$$

where β is the noise level in the data. The solution of the nonlinear equation is obtained by a combination of the Bisection method and Newton's method or the Secant method.

This method is suitable for the solution of large–scale problems since the coefficient matrix and its transpose are used only in matrix–vector products. The method has been applied successfully to the regularization of discrete ill–posed problems from image reconstruction. Applications to more general unstructured problems have not been reported. The method requires knowledge of the noise level in the data and also of an interval that contains the eigenvalues of the matrix A.

Chapter 4

Regularization and the Trust–Region Subproblem

As we saw in the previous chapter, there are several possible approaches to regularization. Our approach uses a quadratically constrained least squares problem. In this chapter we describe this problem and show that it is equivalent to the Trust–Region Subproblem (TRS). We study the theoretical and computational aspects of the TRS and review the existent methods for the solution of this problem.

We present our regularization approach in Section 4.1. We study the TRS in Section 4.2. In Section 4.3 we study the TRS in the special case of discrete ill–posed problems. In Section 4.4 we review the methods for the TRS.

4.1 Trust–Region Subproblem Approach to Regularization

We pose the regularization problem as the following quadratically constrained least squares problem

$$\begin{array}{ll} \min & \frac{1}{2}\|Ax-\bar{b}\| \\ s.t. & \|x\|\leq\Delta \end{array} \tag{4.1}$$

where $A \in \mathbb{R}^{m\times n}$, $\bar{b} \in \mathbb{R}^m$, $\Delta \in \mathbb{R}$, $\Delta > 0$. The right hand side $\bar{b}$ satisfies $\bar{b} = b + \varepsilon s$, where $b \in \mathbb{R}^m$ is the *exact*, unknown data vector and $s \in \mathbb{R}^m$ is a random vector representing noise.

We observe now that the solutions of (4.1) are the same as those of the following problem

$$\begin{array}{ll} \min & \frac{1}{2}x^T Hx + g^T x \\ s.t. & \|x\|\leq\Delta \end{array} \tag{4.2}$$

with $H = A^T A$ and $g = -A^T \bar{b}$.

Problem (4.2) is a particular case of a problem known in optimization as the Trust–Region Subproblem, which arises in the context of globalization strategies for locally convergent methods, such as Newton's method.

We compute the solution to the regularization problem (4.1) by solving the TRS (4.2). We are interested in the special case of large–scale ill–conditioned TRS arising from the discretization of ill–posed problems.

We recall from §3.1.2 that for discrete ill–posed problems:

- $A \in \mathbb{R}^{m \times n}$, with m, n large,
- the condition number $\kappa_2(A)$ is large, i.e. A is ill–conditioned, and
- A has a large cluster of small singular values.

Therefore, the matrix $H = A^T A$ will be large, ill–conditioned and will have a large cluster of small eigenvalues. Observe that matrices of the form $H - \lambda I$ will also be large and ill–conditioned. Observe also that we should avoid forming $A^T A$ since this is expensive in terms of both storage and computation, and it may introduce large rounding errors (see [25, example 5.3.2 on p.225]).

Another issue that we must take into account when using (4.2) for regularization purposes, is that a difficult case associated with the TRS is very common for discrete ill–posed problems. We show this in Section 4.3.

The connection between problems (4.1) and (4.2) and the fact that most of the theoretical and computational aspects of the TRS are known, make this approach to regularization a very attractive one.

The use of methods for the TRS in the regularization context was suggested by Sorensen [71] and Rendl and Wolkowicz [66]. In [48] Martínez and Santos present a trust–region strategy which they apply to a regularization problem as an example. That particular strategy is not suitable for large–scale problems. Chan et al. [10] present a method for the TRS for the special case of ill–conditioned quadratically constrained least squares problems. The method works under the assumption $\Delta < \|A^{\dagger}\bar{b}\|$ and requires the solution of a sequence of systems of equations that will be large and ill–conditioned in the discrete ill–posed case.

It is interesting to note that there are very few references to methods for the TRS in the regularization literature and that methods proposed in that context for quadratically constrained least squares problems do not take into account all the properties of the TRS. We study these properties in the next section.

4.2 The Trust–Region Subproblem

As we mentioned before, Trust–Region Subproblems arise in optimization in the context of globalization strategies for locally convergent methods.

Locally convergent methods combined with such a globalization strategy are called Trust–Region Methods. In these methods we minimize a quadratic model of the objective function in a region where we 'trust' the model, i.e. a region where the quadratic model closely resembles the objective function. We adjust this *trust region* iteratively in a way that guarantees convergence to a solution. Trust–region methods require solving a sequence of Trust–Region Subproblems.

The origin of Trust–Region Methods is found in the works of Levenberg [44] and Marquardt [47]. Moré [54] gives an excellent survey of Trust–Region Methods which includes a review of methods for the TRS. In this section we study theoretical and computational aspects of the TRS and in Section 4.4 we review the methods available for solving this problem.

We define the Trust–Region Subproblem as

$$\begin{array}{ll} \min & \psi(x) \\ s.t. & {\scriptstyle \|x\|\leq\Delta} \end{array} \tag{4.3}$$

where $\psi(x) = \frac{1}{2}x^T Hx + g^T x$, $H \in \mathrm{I\!R}^{n\times n}$, $H = H^T$; $g \in \mathrm{I\!R}^n$; $\Delta \in \mathrm{I\!R}$, $\Delta > 0$.

An immediate observation is that there always exists a solution for this problem since we are minimizing a continuous functional on a compact set. Furthermore, the high degree of structure of the problem makes it possible to characterize its global solutions. This remarkable result was obtained independently by Gay [20] and Sorensen [69] and we present it in Lemma 4.1, where we follow [71] in the use of a nonstandard nonpositive multiplier.

Lemma 4.1 ([69]) A feasible vector x_* is a solution of (4.3) with corresponding Lagrange multiplier λ_* if and only if x_* and λ_* satisfy

(i) $(H - \lambda_* I)x_* = -g$.

(ii) $H - \lambda_* I$ is positive semidefinite.

(iii) $\lambda_* \leq 0$.

(iv) $\lambda_*(\|x_*\| - \Delta) = 0$.

Proof In the first part of the proof we show that *(i)–(iv)* are necessary conditions for $\{x_*, \lambda_*\}$ to be an optimal pair for (4.3).

Observe that if $\{x_*, \lambda_*\}$ is an optimal pair for problem (4.3), then *(i), (iii)* and *(iv)* hold since these are the Karush–Kuhn–Tucker necessary conditions (see [46, Ch. 10], for example) for a solution of (4.3). Therefore, it only remains to show that $H - \lambda_* I$ is positive semidefinite. We divide the analysis in two cases.

Case 1: $x_* \neq 0$.

Since x_* solves (4.3) it also solves

$$\begin{array}{ll} \min & \psi(x) \\ s.t. & {\scriptstyle \|x\|=\|x_*\|} \end{array} .$$

Therefore, $\psi(x) \geq \psi(x_*)$, $\forall x \in \mathbb{R}^n$ such that $\|x\| = \|x_*\|$, i.e.

$$\frac{1}{2}x^T Hx + g^T x \geq \frac{1}{2}x_*^T Hx_* + g^T x_*. \tag{4.4}$$

Now, by *(i)* we have

$$\begin{aligned} g^T x &= -x_*^T(H - \lambda_* I)x \quad \text{and} \\ g^T x_* &= -x_*^T(H - \lambda_* I)x_*. \end{aligned}$$

Substituting these expressions in (4.4) and after some algebraic manipulation, we obtain

$$\frac{1}{2}(x - x_*)^T(H - \lambda_* I)(x - x_*) \geq \frac{\lambda_*}{2}(x^T x - x_*^T x_*) = 0.$$

Observe that for any vector v in $\mathbb{R}^n$ such that $v \not\perp x_*$ it is possible to compute x such that $\|x\| = \|x_*\|$ and $v = (x - x_*)/\gamma$, for $\gamma \neq 0$. Therefore, $v^T(H - \lambda_* I)v \geq 0$. If $v \in \mathbb{R}^n$ is such that $v \perp x_*$, it is possible to construct a sequence of vectors $\{v_j\}$ such that $v_j \not\perp x_*$ with $v_j \to v$. And since $v_j \not\perp x_*$ we have that $v_j^T(H - \lambda_* I)v_j \geq 0$ and we must have $v^T(H - \lambda_* I)v \geq 0$. Therefore, $H - \lambda_* I$ is positive semidefinite.

Case 2: $x_* = 0$.

Then, *(i)* implies that $g = 0$ and x_* solves

$$\begin{array}{ll} \min & \frac{1}{2}x^T Hx \\ s.t. & {\scriptstyle \|x\| \leq \Delta} \end{array} .$$

Therefore $x^T Hx \geq 0, \forall x \in \mathbb{R}^n$, i.e. H is positive semidefinite and since $\lambda_* \leq 0$ is a necessary condition, then $H - \lambda_* I$ is positive semidefinite.

In the second part of the proof we show that *(i)–(iv)* are sufficient conditions for $\{x_*,\ \lambda_*\}$ to be an optimal pair for (4.3).

We will show that if $\{x_*, \lambda_*\}$ satisfies conditions *(i)–(iv)* then

(1) If $\lambda_* = 0$ and $\|x_*\| \leq \Delta$ then x_* solves (4.3).

(2) If $\lambda_* \leq 0$ and $\|x_*\| = \Delta$ then x_* solves (4.3).

Observe that *(i)* implies that

$$x_*^T(H - \lambda_* I)x_* + g^T x_*^T = 0,$$

and since $H - \lambda_* I$ is positive semidefinite then $\forall x \in \mathbb{R}^n$,

$$\begin{aligned} (x - x_*)^T(H - \lambda_* I)(x - x_*) &\geq 0 \\ &= x_*^T(H - \lambda_* I)x_* + g^T x_* \end{aligned}$$

and some manipulation yields

$$\psi(x) \geq \psi(x_*) + \frac{\lambda_*}{2}(x^T x - x_*^T x_*).$$

Therefore, if $\lambda_* = 0$ then $\psi(x) \geq \psi(x_*)$, $\forall x \in \mathbb{R}^n$ and if in addition $\|x_*\| \leq \Delta$ then x_* solves (4.3). If $\lambda_* \leq 0$ and $\|x_*\| = \Delta$ then since any $x \in \mathbb{R}^n$, $x \neq 0$ can be normalized such that $\|x\| = \Delta$, we have that $\psi(x) \geq \psi(x_*)$ and therefore, x_* solves (4.3) with the constraint binding at x_*.

This completes the proof. □

A solution of (4.3) lies either in the interior or on the boundary of the feasible set (trust region), i.e. the set $\{x|\ \|x\| \leq \Delta\}$. The TRS has no solutions on the boundary if and only if H is positive definite and $\|H^{-1}g\| < \Delta$ (see [69]). In this case, $x_* = -H^{-1}g$ with corresponding Lagrange multiplier $\lambda_* = 0$.

If the conditions for an interior solution are not satisfied, the solutions to problem (4.3) will be on the boundary of the feasible set. The cases when H is positive semidefinite and singular, or indefinite require special attention since in such cases the solution may not be unique. This situation is known as the *hard case* ([54], [55]) and we will discuss it in detail in §4.2.2. In §4.2.1 we will study the *secular equation*, which is an important tool for both the analysis and the design of methods for the TRS.

Before we proceed we need to introduce some notation which we will use in the rest of this section and also in the next chapter.

(i) We will denote the eigenvalues of H by $\delta_i, i = 1, \ldots, n$ with $\delta_1 \leq \delta_2 \leq \cdots \leq \delta_n$.

(ii) Let $k \leq n$ be the number of distinct eigenvalues of H, then $\mathcal{S}_i$ will denote the eigenspace corresponding to δ_i, $i = 1, \ldots, k$ i.e. $\mathcal{S}_i = \{q | Hq = \delta_i q\}$.

(iii) $H = QDQ^T$ will denote an eigendecomposition of H, i.e. $QQ^T = Q^TQ = I$ and $D = diag(\delta_1, \delta_2, \ldots, \delta_n)$.

(iv) $\gamma_i = q_i^T g$, i.e. γ_i will be the component of g in the direction of the ith eigenvector of H.

4.2.1 The Secular Equation

To introduce the secular equation let us first define the function $\phi(\lambda)$ and its derivative $\phi'(\lambda)$ as follows

$$\begin{aligned}
\phi(\lambda) &= g^T(H - \lambda I)^{\dagger} g = \sum_{i=1}^{n} \frac{{\gamma_i}^2}{\delta_i - \lambda} \\
\phi'(\lambda) &= g^T[(H - \lambda I)^{\dagger}]^2 g = \sum_{i=1}^{n} \frac{{\gamma_i}^2}{(\delta_i - \lambda)^2}
\end{aligned}$$

Observe that if $g \neq 0$, both $\phi(\lambda)$ and $\phi'(\lambda)$ are rational functions for which the eigenvalues of H are first and second order poles, respectively. Figures 4.1 and 4.2

illustrate the typical behavior of $\phi(\lambda)$ and $\phi'(\lambda)$. In these examples, $n = 3$ and the eigenvalues of H are $-2,\ 0, 2$.

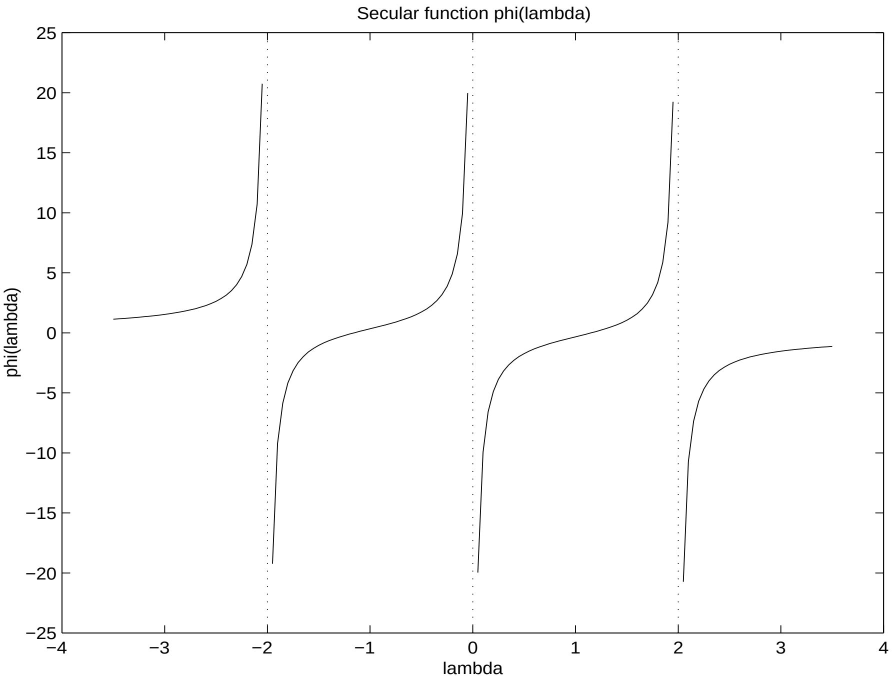

Figure 4.1 Secular Function $\phi(\lambda)$. Eigenvalues of Hessian: $-2, 0, 2$.

Computing boundary solutions for the TRS is closely related to finding the smallest value of λ that satisfies

$$\phi'(\lambda) = \Delta^2. \tag{4.5}$$

To see this, suppose that either g is not orthogonal to $\mathcal{S}_1$ or if g is orthogonal to $\mathcal{S}_1$ then $\Delta \leq \|(H - \delta_1 I)^\dagger g\|$. Then the smallest λ that satisfies (4.5) is such that $\lambda \leq \delta_1$. For such λ we have that $H - \lambda I$ is positive semidefinite and if we define $x = -(H - \lambda I)^\dagger g$ then $\phi'(\lambda) = \|x\|^2 = \Delta^2$. If in addition $\lambda \leq 0$, then Lemma 4.1 implies that such x is a boundary solution for the TRS. If $\lambda > 0$ then H is positive definite, $\|H^{-1}g\| < \Delta$ and there is a unique interior solution for the TRS.

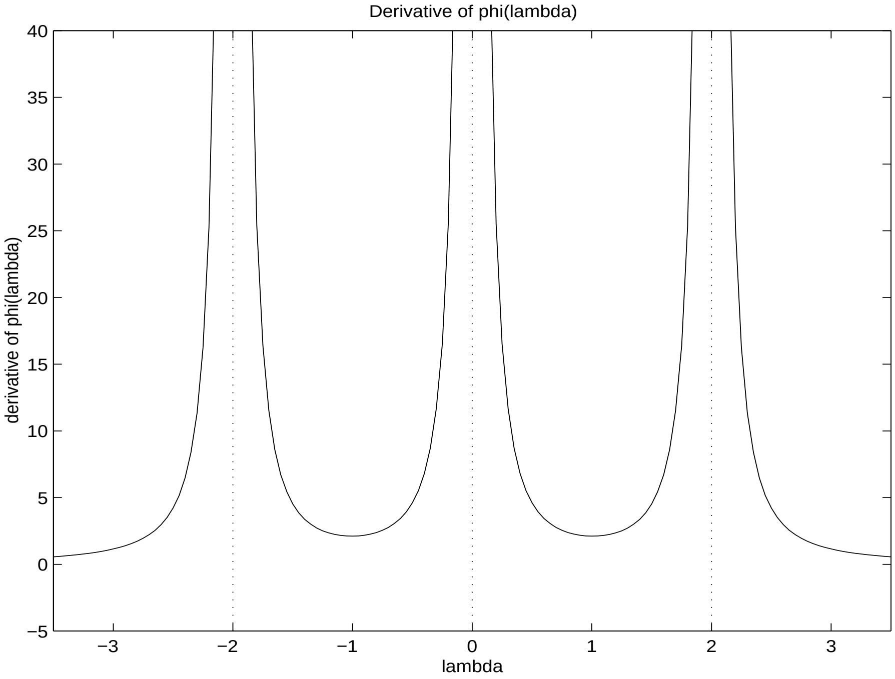

Figure 4.2 Secular Function $\phi'(\lambda)$. Eigenvalues of Hessian: $-2, 0, 2$.

A key observation about $\phi(\lambda)$ and $\phi'(\lambda)$ is that they are both strictly increasing in $(-\infty, \delta_1)$ if g is not orthogonal to $\mathcal{S}_1$. If g is orthogonal to $\mathcal{S}_i$ for $i = 1, 2, \ldots, \ell$, $1 \leq \ell < n$ then $\phi(\lambda)$ and $\phi'(\lambda)$ are strictly increasing in $(-\infty, \delta_{\ell+1})$.

Since Lemma 4.1 establishes that $H - \lambda I$ must be positive semidefinite at a solution of the TRS, we are interested in the unique solution of (4.5) in $(-\infty, \delta_1)$ if g is not orthogonal to $\mathcal{S}_1$, or in $(-\infty, \delta_1]$ if g is orthogonal to $\mathcal{S}_1$. Note that this solution corresponds to the smallest λ at which the line Δ^2 intersects $\phi'(\lambda)$.

A solution of (4.5) in $(-\infty, \delta_1)$ or in $(-\infty, \delta_1]$ will always exist as long as $\gamma_1 \neq 0$, or $\gamma_i = 0$, $i = 1, 2, \ldots, \ell$ for $1 \leq \ell < n$, and $\Delta \leq \|(H - \delta_1 I)^{\dagger} g\|$. In the latter case, the smallest λ that satisfies (4.5) lies in $(-\infty, \delta_1]$. Note that $\gamma_1 = 0$ if and only if g is orthogonal to $\mathcal{S}_1$.

Up to this point we have assumed that either $\gamma_1 \neq 0$ or $\gamma_1 = 0$ and $\Delta \leq \|(H - \delta_1 I)^\dagger g\|$. The case $\gamma_1 = 0$ for a general value of Δ is of special interest since it might give rise to the so–called hard case, which we present in §4.2.2.

Equations like (4.5) are known as *secular equations* and functions like $\phi(\lambda)$, $\phi'(\lambda)$ are sometimes called *secular functions.* The term seems to come from celestial mechanics (see [1]).

Secular equations appear in many contexts such as in the solution of certain eigenvalue problems, see for example [1], [3], [23], [7], [49], [51], [50] and [52]. These works are concerned with computing *all* the solutions of a secular equation and they usually assume that $\gamma_i \neq 0,\ i = 1, 2, \ldots, n$, therefore excluding the possibility of the *hard case.* This fact prevents the use of such methods for solving the general TRS.

The use of the secular equation (4.5) in connection with the TRS or related problems such as quadratically equality constrained least squares problems, can be traced back to [18]. Other works include [23], [22], [64],[65], [39], [53], [69] and [55].

Let us continue now with the study of the TRS, specifically with a special case: the hard case.

4.2.2 The Hard Case

The *hard case* refers to a special situation in which the boundary solution of problem (4.3) is not unique. The hard case can only occur when H is positive semidefinite and singular, or indefinite, when g is orthogonal to the eigenspace associated with the smallest eigenvalue of H (and possible to the eigenspaces associated with the next eigenvalues), and when $\Delta > \|(H - \delta_1 I)^\dagger g\|$. The precise statement is contained in the following result from [69].

Lemma 4.2 ([69]) Assume that g is orthogonal to $\mathcal{S}_1$ and let $p = -(H - \delta_1 I)^\dagger g$. If $\delta_1 \leq 0$ and $\|p\| \leq \Delta$, then the solutions of (4.3) consist of the set $\{x|x = p + z, z \in \mathcal{S}_1, \|x\| = \Delta\}$ with Lagrange multiplier $\lambda_* = \delta_1$.

Proof We will show first that given p as above and a vector $u \in \mathcal{S}_1$ with $\|u\| = 1$, if $x = p + \tau u$ with τ such that $|\tau| = \Delta^2 - \|p\|^2$, then $\|x\| = \Delta$. In order to prove this, we need to show first that p is orthogonal to $\mathcal{S}_1$.

Note that

$$p^T u = -g^T (H - \delta_1 I)^\dagger u. \tag{4.6}$$

Now, since $g \perp \mathcal{N}(H - \delta_1 I)$, this implies that $g \in \mathcal{R}(H - \delta_1 I)$ and since $(H - \delta_1 I)(H - \delta_1 I)^\dagger$ is an orthogonal projection onto $\mathcal{R}(H - \delta_1 I)$, we have

$$(H - \delta_1 I)(H - \delta_1 I)^\dagger g = g. \tag{4.7}$$

Substituting (4.7) into (4.6), we obtain

$$\begin{aligned} p^T u &= -g^T (H - \delta_1 I)^\dagger (H - \delta_1 I)(H - \delta_1 I)^\dagger u \\ &= -g^T (H - \delta_1 I)^\dagger (H - \delta_1 I)^\dagger (H - \delta_1 I) u \\ &= 0. \end{aligned}$$

Therefore, p is orthogonal to $\mathcal{S}_1$ and $\|p + \tau u\|^2 = \|p\|^2 + |\tau| \|u\|^2$. This implies that in order to have $\|x\| = \Delta$, with $x = p + \tau u$, τ must satisfy $|\tau| = \Delta^2 - \|p\|^2$.

Let τ be such that $|\tau| = \Delta^2 - \|p\|^2$. Let $z = \tau u$ and $x = p + z$. To prove that $\{x, \delta_1\}$ is an optimal pair for problem (4.3) in the hard case, we need to show that $\{x, \delta_1\}$ satisfies conditions *(i)–(iv)* of Lemma 4.1. Observe that $\{x, \delta_1\}$ satisfies conditions *(ii), (iii)* and *(iv)*, since δ_1 is the smallest eigenvalue of H, $\delta_1 \leq 0$ by hypothesis, and $\|x\| = \Delta$. Therefore, it only remains to show that the pair $\{x, \delta_1\}$ satisfies condition *(i)*, i.e. we must show that $(H - \delta_1 I)x = -g$. To see this, observe

$$\begin{aligned} (H - \delta_1 I)x &= (H - \delta_1 I)(p + z) \\ &= -(H - \delta_1 I)(H - \delta_1 I)^\dagger g + (H - \delta_1 I) z. \end{aligned}$$

By (4.7) we now have

$$(H - \delta_1 I)x = -g + (H - \delta_1 I)z$$

and since $z \in \mathcal{N}(H - \delta_1 I)$ we conclude

$$(H - \delta_1 I)x = -g,$$

which completes the proof. □

Observe that Lemma 4.2 still holds if $g \perp \mathcal{S}_i$, $i = 1, 2, \ldots, \ell$ with $1 \leq \ell \leq n$.

In Figure 4.3 we illustrate the situation in terms of the contours of $\psi(x)$ and the trust region in the positive semidefinite singular case for $n = 2$. In this case we have infinite solutions along the contour for which the quadratic has the same minimum value (dashed line in Figure 4.3).

For the indefinite case, we first show the *easy case* in Figure 4.4 where the trust–region radius is small enough so the the trust–region subproblem has a unique boundary solution. If the radius is larger than $\|(H - \delta_1 I)^{\dagger} g\|$, there will be two solutions in the direction of negative curvature. As Lemma 4.2 establishes these solutions are obtained adding a suitable multiple of an eigenvector corresponding to the smallest eigenvalue of H, to the vector $p = -(H - \delta_1 I)^{\dagger} g$. We illustrate this in Figure 4.5.

In Figure 4.6 we show the function $\phi'(\lambda)$ for the same example of Figure 4.2 for the case when g is orthogonal to $\mathcal{S}_1$. In this figure it is clear that since $\gamma_1 = 0$ in the hard case, δ_1 is no longer a pole of $\phi(\lambda)$ or $\phi'(\lambda)$.

We saw in Lemma 4.2 that the hard case can only occur when $g \perp \mathcal{S}_1$. In practice, g exactly orthogonal to $\mathcal{S}_1$ will rarely occur. The most common situation is that g is numerically orthogonal to $\mathcal{S}_1$, specially for discrete ill–posed problems, as we will show in Section 4.3.

We will refer to the situation when g is orthogonal or nearly orthogonal to $\mathcal{S}_1$ as a potential hard case and near or numerical potential hard case, respectively. We use

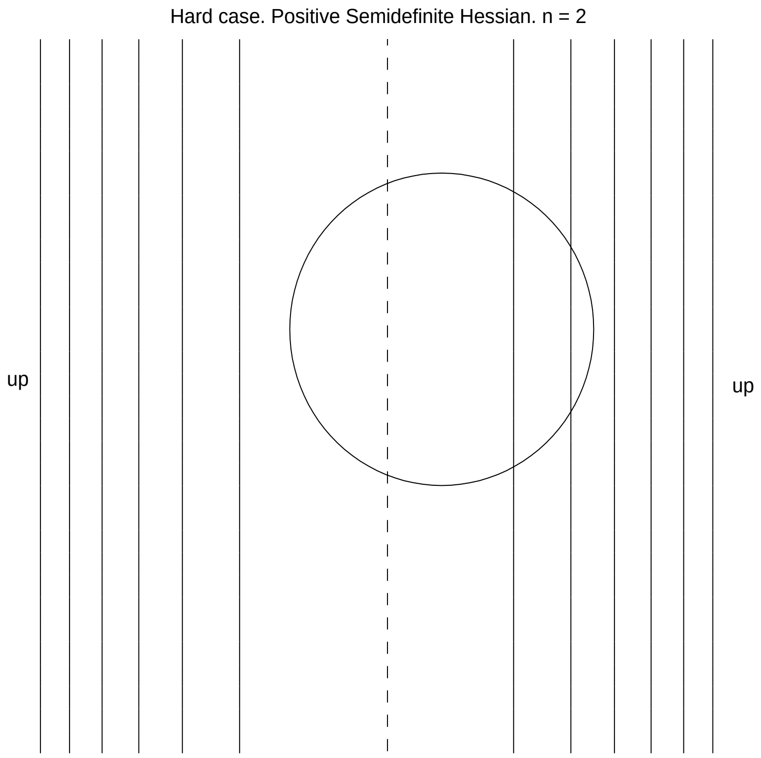

Figure 4.3 The Hard Case for a Positive Semidefinite and Singular Hessian.

the word potential in this case since even when g is orthogonal or nearly orthogonal to $\mathcal{S}_1$ the hard case or near hard case will only occur if in addition $\Delta > \|(H - \delta_1 I)^\dagger g$.

In a near potential hard case we find that although δ_1 is a pole of the secular functions, the coefficient γ_1 is very small. As a consequence, the functions become very steep close to this pole as we can see in Figure 4.7. This poses problems to the numerical computation of the smallest solution of (4.5). General methods for the TRS based on the secular equation must take into account this difficulty. Other methods will also encounter problems in the hard case.

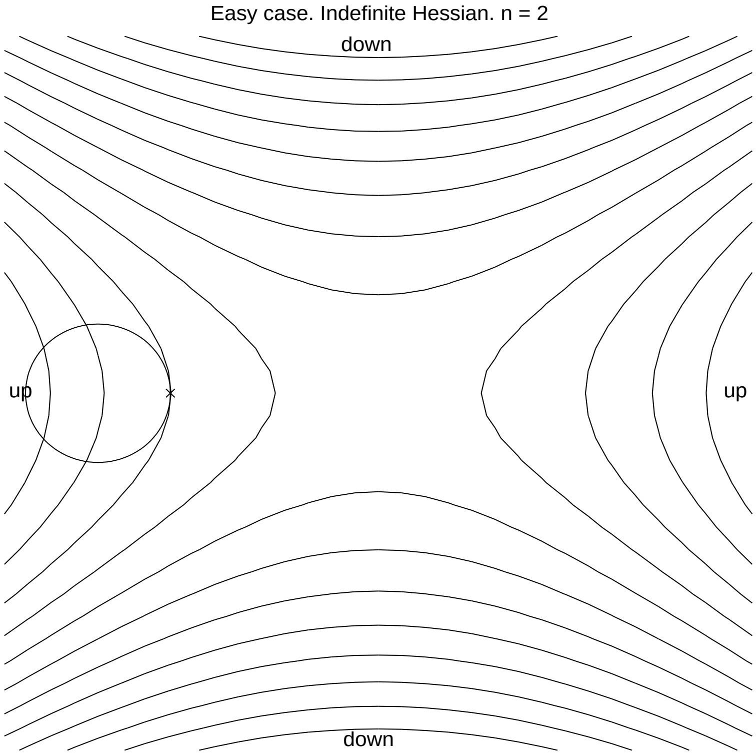

Figure 4.4 The Easy Case for an Indefinite Hessian.

4.3 The Discrete Ill–Posed Trust–Region Subproblem

In this section we show that the main feature of the TRS in the discrete ill–posed case is the occurrence of hard cases or potential hard cases.

When we encounter the hard case in optimization problems it is usually in its basic or *single* form, i.e. $g \perp \mathcal{S}_1$ only. We show in this section that not only can we expect potential hard cases to be the most common cases for discrete ill–posed problems, but they will also occur in a *multiple* form, i.e. $g \perp \mathcal{S}_i$ $i = 1, 2, \ldots, \ell$, $1 \leq \ell < n$, for $g \neq 0$.

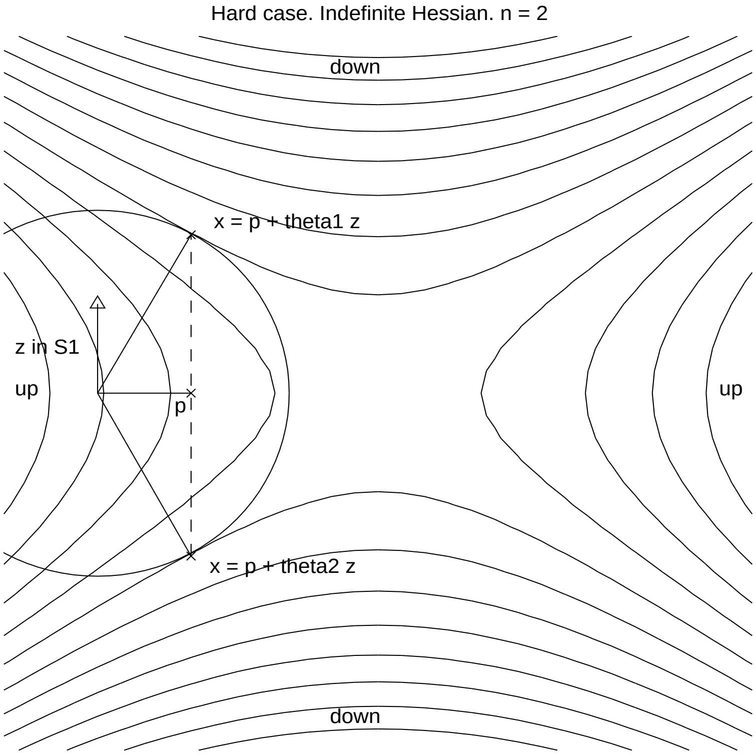

Figure 4.5 The Hard Case for an Indefinite Hessian.

In Figure 4.8 we show $\phi'(\lambda)$ for a problem of dimension $n = 7$, where a multiple potential hard case arises with $\ell = 4$. We can observe in the graph that in this case the smallest 4 eigenvalues of H $(-2, 0, 2, 3)$ are not poles of $\phi'(\lambda)$.

In Figure 4.9 we show a *near* potential hard case in its multiple instance, where we can observe the steepness of the functions near the poles.

We will show next that for discrete ill–posed problems we can expect g to be orthogonal or nearly orthogonal to $\mathcal{S}_i, i = 1, 2, \ldots, \ell$ for $1 \leq \ell < n$. The value ℓ is usually of the order of 20% to 90% of the problem dimension. Therefore, multiple potential hard cases or near multiple potential hard cases are likely to arise in this context.

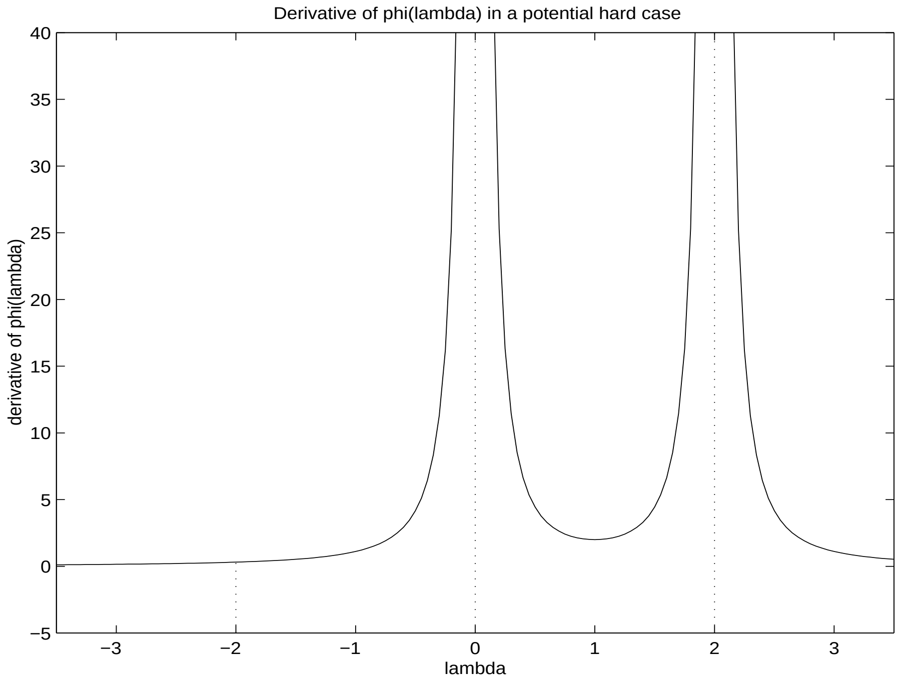

Figure 4.6 The Secular Function $\phi'(\lambda)$ in a Potential Hard Case.

Let $A \in \mathbb{R}^{m\times n}$, $H = A^T A$ and $g = -A^T\bar{b}$, where $\varepsilon \geq 0$, $\bar{b} = b + \varepsilon s$ with $b, s \in \mathbb{R}^m$. The vector s is a random vector representing noise. Let $A = U\Sigma V^T$ be a Singular Value Decomposition of A. Then

$$\begin{aligned} H &= V\Sigma^2 V^T, \quad \text{and} \\ g &= -V\Sigma U^T\bar{b} \\ &= -V\ diag(\sigma_1, \ldots, \sigma_{n-1}, \sigma_n)\ U^T\bar{b}\ . \end{aligned}$$

Notice that $\sigma_i = \sqrt{\delta_{n-i+1}}$, since we number the eigenvalues and the singular values in opposite ways.

Suppose first that $\ell = 1$ and let us show that we can expect g to be nearly orthogonal to $\mathcal{S}_1$. Suppose that δ_1 has multiplicity k. Therefore, $\mathcal{S}_1 = \{v_{n-k+1}, \ldots, v_n\}$.

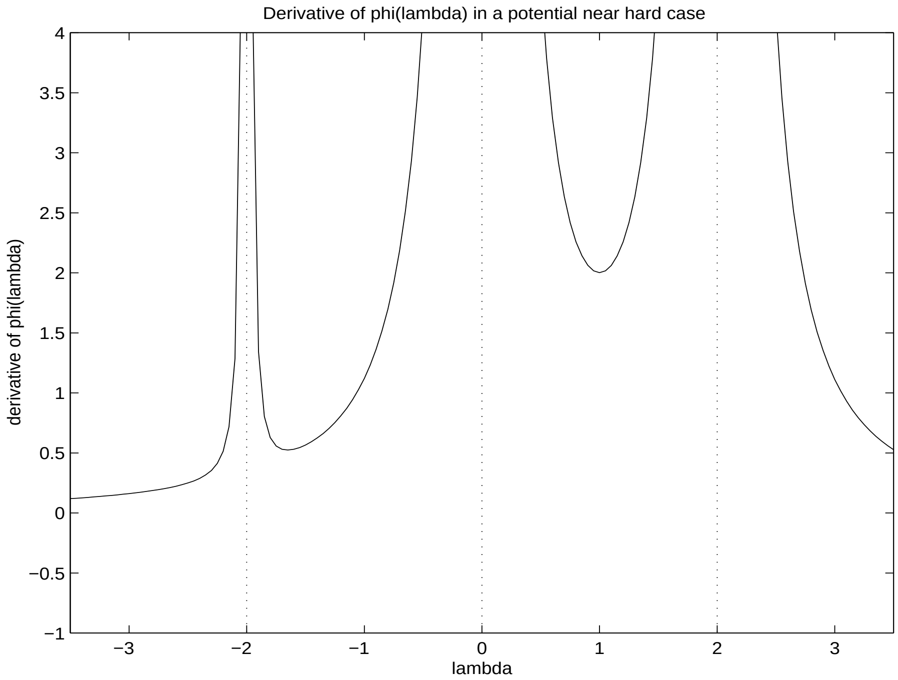

Figure 4.7 The Secular Function $\phi'(\lambda)$ in a *Near* Potential Hard Case.

Let $x = v_j, \ n-k+1 \leq j \leq n$. Then

$$g^T x \quad = \quad -\bar{b}^T U diag(\sigma_1, \ldots, \sigma_{n-1}, \sigma_n) V^T v_j$$

and since $v_i^T v_j = 0, \ i \neq j$ and $v_j^T v_j = 1$, we have

$$\begin{aligned} g^T x &= -\sigma_n u_j^T \bar{b} \\ &= -\sigma_n u_j^T (b + \varepsilon s) \\ &= -\sigma_n (u_j^T b + \varepsilon u_j^T s) \ . \end{aligned}$$

Let us see now that we can expect $g^T x$ to be small in discrete ill–posed problems. In order to do this, we will analyze both the exact and noisy data cases.

In the exact data case, $\varepsilon = 0$ and we saw in §3.1.2 that if the underlying continuous problem satisfies the Picard Condition then $u_j^T b, \ n-k+1 \leq j \leq n$ are small and since σ_n is also small, we have that g is nearly orthogonal to $\mathcal{S}_1$ in this case.

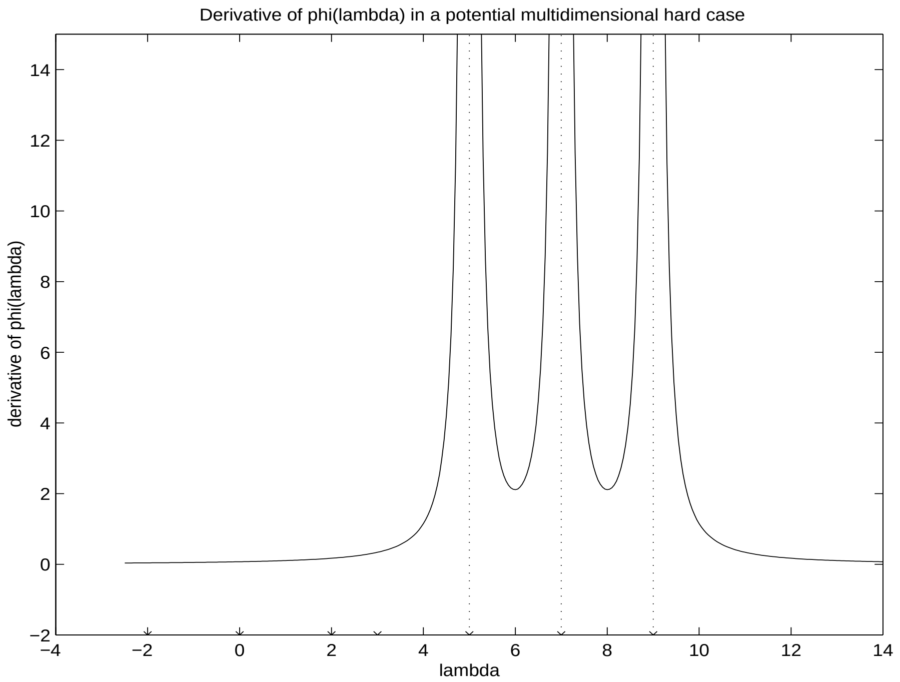

Figure 4.8 Multiple Potential Hard Case.

For noisy data, $\varepsilon \neq 0$ and $g^T x$ might not be small since although $\sigma_n u_j^T b$ will be small as before, the term $\varepsilon u_j^T s$ might be large enough to compensate for the small σ_n. For most severely ill–conditioned problems however, σ_n is so close to zero that even if $u_j^T s$ is large, g will still be nearly orthogonal to $\mathcal{S}_1$.

In Table 4.1 we illustrate the situation with problems from the Regularization Tools package [36]. In these examples the multiplicity of δ_1 is one and therefore $x = v_n$. We show an exact data case for problem **heat**, $n = 50$, a noisy data case for the same problem where $g^T x$ is still small and a noisy data case for problem **deriv2**, $n = 50$ in which $g^T x$ is not too small.

A similar argument to the one used for $\mathcal{S}_1$ can be used to show that g is nearly orthogonal to $\mathcal{S}_i$, $i = 2, \ldots, \ell$ when the cluster of small singular values of A is of size ℓ. To illustrate this, we computed the values $g^T v_i$, $i = 1, 2, \ldots, n$ for problem

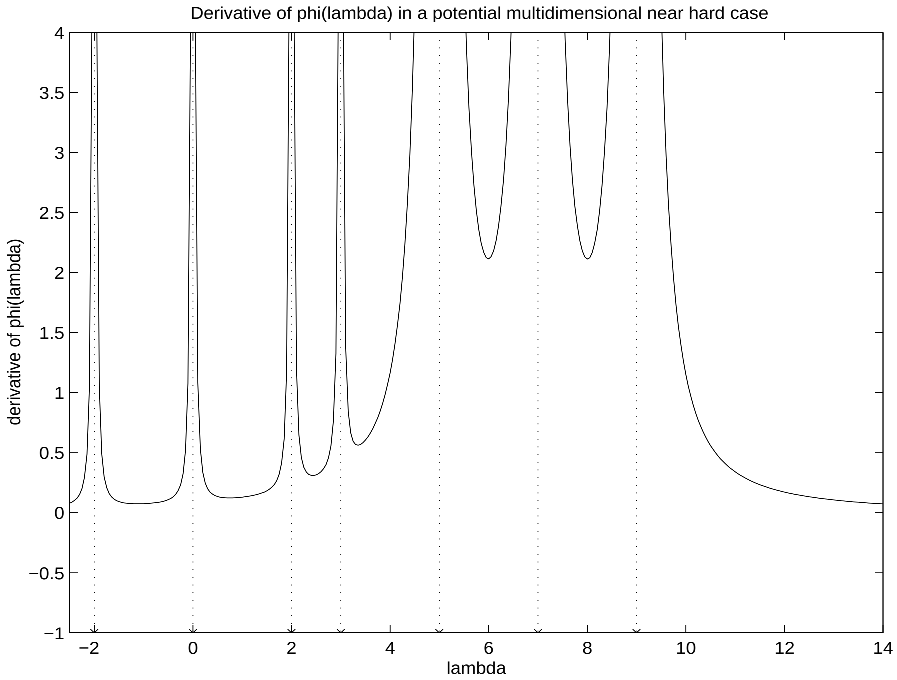

Figure 4.9 *Near* Multiple Potential Hard Case.

Problem	σ_1	$u_n^T b$	$\varepsilon u_n^T s$	$g^T v_n$
heat	1.19×10^{-27}	4.59×10^{-17}	0.0	$5.4621e - 44$
heat	1.19×10^{-27}	4.59×10^{-17}	-0.0061	-7.38×10^{-30}
deriv2	3.33×10^{-5}	-3.33×10^{-7}	0.0051	-1.68×10^{-7}

Table 4.1 Near orthogonality of g with respect to $\mathcal{S}_1$.

foxgood from [36], $n = 300$, with noisy data. In Figure 4.10 we show the absolute value of $g^T v_i$ in logarithmic scale and observe that up to $\ell = 292$ all the values are of magnitude less than or equal to 10^{-15}.

The above discussion implies that in order to use methods for the TRS for the regularization of discrete ill–posed problems, we must take into account that multiple potential hard cases are very likely to arise in this context. Until now, none of the

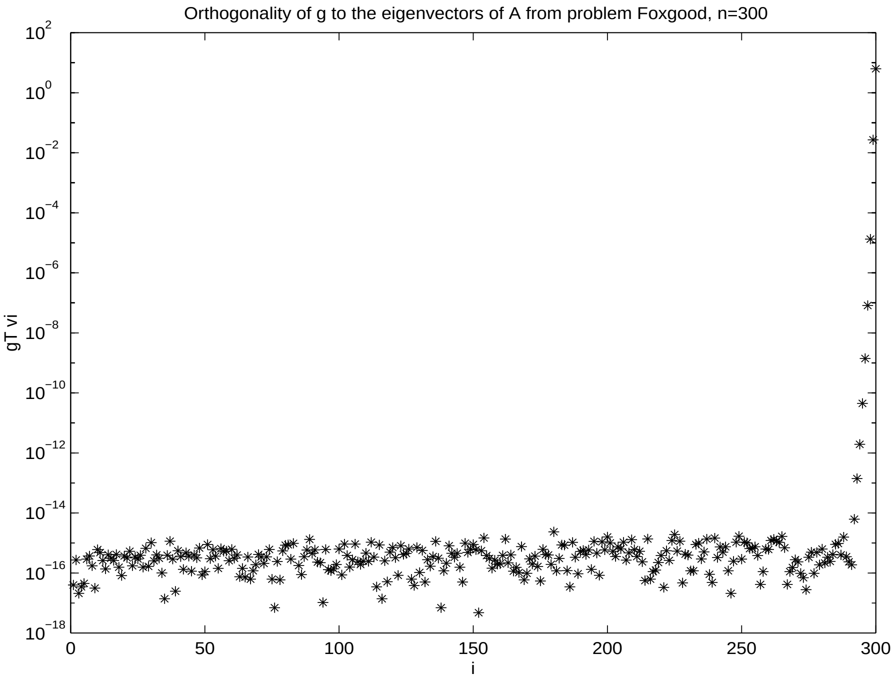

Figure 4.10 Orthogonality of g with respect to the eigenvectors of a discretized ill–posed operator.

existent methods for the TRS has considered this issue. We present an overview of those methods in the next section.

4.4 Methods for the Trust–Region Subproblem

Several methods have been proposed for the solution of the trust–region subproblem, most of them for small to medium scale problems. The main methods in this category are the dogleg methods of Powell [63] and [62], the double dogleg method of Dennis and Mei [13] and the method of Moré and Sorensen [55]. The first two methods work within the context of optimization algorithms and do not provide an optimal solution whereas the method of Moré and Sorensen does. The method of Moré and Sorensen is the method of choice for trust–region subproblems for which the factorization of matrices of the form $H - \lambda I$ is affordable.

Until recently, the only method available for the solution of large–scale trust–region subproblems was the method of Steihaug [72]. The method uses the Preconditioned Conjugate Gradient method to compute an approximate solution for the trust–region subproblem in a Krylov subspace. Although this method does not provide optimal solutions, it has been shown to be very efficient in the context of optimization algorithms. However, the method cannot be applied to discrete ill–posed problems even in that context, since it does not handle the hard case which is a very common case for these problems as we showed in Section 4.3. Another limitation of this method is the need for a preconditioner which is still an open issue for discrete ill–posed problems, as we discussed in §3.4.3.

The new methods for the large–scale trust–region subproblem are the methods of Sorensen [71], Rendl and Wolkowicz [66], Santos and Sorensen [68] and Rojas, Santos and Sorensen [67]. All these methods pose the trust–region subproblem in terms of a parameterized eigenvalue problem with a matrix closely related to the Hessian H.

The methods in [71], [68] and [67] are based on a rational interpolation strategy to update the parameter of the eigenvalue problem. The eigenvalue problems are solved with the Implicitly Restarted Lanczos Method [70]. We will discuss the differences among these approaches in Chapter 5.

In [66], the trust–region subproblem is relaxed to a semidefinite program. The strategy for updating the parameter is a dual simplex method in the standard case and a primal simplex method in the hard case. The eigenvalue problems are solved with a block–Lanczos procedure.

In Chapter 5 we present a new method which is based on [67]. Among the contributions of this new method is the treatment of the hard case for which we have developed a complete theory. These features allow us to solve any kind of trust–region subproblems, including those arising from the regularization of discrete ill–posed problems.

Chapter 5

A Method for the Large–Scale Trust–Region Subproblem

We recall the definition of the Trust–Region Subproblem (TRS) from the previous chapter

$$\begin{array}{ll} \min & \psi(x) \\ s.t. & {\scriptstyle \|x\| \leq \Delta} \end{array} \qquad (5.1)$$

where $\psi(x) = \frac{1}{2}x^T Hx + g^T x$, $H \in \mathbb{R}^{n \times n}$, $H = H^T$; $g \in \mathbb{R}^n$; $\Delta \in \mathbb{R}$, $\Delta > 0$.

In this chapter we present a method for the large–scale trust–region subproblem, for which the Hessian H is large and might not be available explicitly. These features prevent the use of methods like [55] that require the factorization of matrices of the form $H - \lambda I$.

Our method is based on a formulation of the TRS as a parameterized eigenvalue problem. In this setting the computation of a solution for the TRS depends on computing an optimal value for the parameter. The computation of this optimal parameter requires the solution of a sequence of large–scale eigenvalue problems for a matrix closely related to H. We solve these eigenproblems by means of the Implicitly Restarted Lanczos Method [70]; as a consequence, we use the matrix H only in matrix–vector products and we have low storage requirements.

In our method, we assume that $g \neq 0$. If $g = 0$ then solving the TRS reduces to finding an eigenvector x corresponding to the smallest eigenvalue of H, and normalizing it such that $\|x\| = \Delta$.

In Section 5.1 we introduce the basic idea of the method when the hard case is not present. In Section 5.2 we give the characterization of the hard case in terms of the parameterized eigenvalue problem. In Section 5.3 we present the main contributions of this work, which consist of the necessary tools to properly handle the hard case. In Section 5.4 we describe the main algorithm and in Section 5.5 we describe each component in detail. In Section 5.6 we present the convergence results. In Section 5.7 we present numerical results on test problems.

5.1 Motivation

Observe that if we define the bordered matrix B_α as

$$B_\alpha = \begin{pmatrix} \alpha & g^T \\ g & H \end{pmatrix} \tag{5.2}$$

then the following relationship holds

$$\frac{\alpha}{2} + \psi(x) = \frac{1}{2}(1, x^T) B_\alpha \begin{pmatrix} 1 \\ x \end{pmatrix} .$$

If we now let $y = (1, x^T)^T$, we can write problem (5.1) as

$$\min_{\substack{s.t. \quad y^T y \leq 1+\Delta^2 \\ e_1^T y = 1}} \quad \tfrac{1}{2} y^T B_\alpha y \; . \tag{5.3}$$

This formulation of the problem suggests that we can compute a solution for the TRS from an eigenpair of B_α, as long as the required normalization of the eigenvector can take place. This follows from the observation that if we eliminate the constraint on the first component of the solution, then problem (5.3) is equivalent to the minimization of a Rayleigh quotient whose solution is an eigenvector associated with the smallest eigenvalue of the matrix B_α.

The formulation of the TRS in terms of a parameterized eigenvalue problem was first used in [71] and later in [66], [68] and [67].

At this point let us introduce some notation and also recall the one from Chapter 4:

(i) We will denote the eigenvalues of B_α by λ_i, $i = 1, \ldots, n+1$, with $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{n+1}$.

(ii) We will denote the eigenvalues of H by $\delta_i, i = 1, \ldots, n$ with $\delta_1 \leq \delta_2 \leq \cdots \leq \delta_n$.

(iii) Let $k \leq n$ be the number of distinct eigenvalues of H, then $\mathcal{S}_i$ will denote the eigenspace corresponding to δ_i, $i = 1, \ldots, k$ i.e. $\mathcal{S}_i = \{q | Hq = \delta_i q\}$.

(iv) $H = QDQ^T$ will denote an eigendecomposition of H, i.e. $QQ^T = Q^TQ = I$ and $D = diag(\delta_1, \delta_2, \ldots, \delta_n)$.

(v) $\gamma_i = q_i^T g$, i.e. γ_i will be the component of g in the direction of the ith eigenvector of H, $i = 1, 2, \ldots, n$.

Let us now see how we can use the solution of the parameterized eigenproblems to solve the TRS. In order to do this, suppose that $\{\lambda, (1, x^T)^T\}$ is an eigenpair of B_α, i.e.

$$\begin{pmatrix} \alpha & g^T \\ g & H \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} = \lambda \begin{pmatrix} 1 \\ x \end{pmatrix}, \tag{5.4}$$

which is equivalent to

$$\alpha - \lambda = -g^T x \quad \text{and} \quad (H - \lambda I)x = -g.$$

Suppose that $\lambda \neq \delta_i$, $i = 1, 2, \ldots, n$. Then x satisfies $x = -(H - \lambda I)^{-1} g$ and we have

$$\begin{aligned} \alpha - \lambda &= g^T (H - \lambda I)^{-1} g = \sum_{i=1}^{n} \frac{{\gamma_i}^2}{\delta_i - \lambda}, \quad \text{and} \\ x^T x &= g^T (H - \lambda I)^{-2} g = \sum_{i=1}^{n} \frac{{\gamma_i}^2}{(\delta_i - \lambda)^2}. \end{aligned}$$

This implies that given the eigenpair $\{\lambda, (1, x^T)^T\}$ of B_α, we can compute the values of the functions ϕ and ϕ', defined in §4.2.1, at the eigenvalue λ as

$$\phi(\lambda) = -g^T x \tag{5.5}$$

$$\phi'(\lambda) = x^T x \tag{5.6}$$

i.e. these values are readily available from an eigenvector of the form $(1, x^T)^T$. We show next that we are interested in an eigenvector corresponding to the smallest eigenvalue of B_α.

As a consequence of Cauchy's Interlace Theorem (see for example [60, Section 10–1]) and also from equation (5.5), we have

$$\lambda_1 \leq \delta_1 \leq \lambda_2 \leq \cdots \leq \lambda_n \leq \delta_n \leq \lambda_{n+1}.$$

This relationship implies that the matrix $H - \lambda_1 I$ is always positive semidefinite, independently of the value of α.

This observation leads to the following idea for solving the TRS in terms of a sequence of parameterized eigenvalue problems.

Given an initial guess for the parameter α, compute λ_1, the smallest eigenvalue of B_α and a corresponding eigenvector v_1. If $\lambda_1 < \delta_1$ and if we can normalize v_1 so that $v_1 = (1, x^T)^T$, then we can evaluate $\phi(\lambda_1)$, $\phi'(\lambda_1)$ by means of (5.5) and (5.6).

Now, if we find an α such that the eigenpair $\{\lambda_1, (1, x^T)^T\}$ for B_α satisfies $\phi'(\lambda_1) = \Delta^2$ with $\phi(\lambda) = \alpha - \lambda$, then we will have $(H - \lambda_1 I)x = -g$ and $\lambda_1(\Delta - \|x\|) = 0$ with $H - \lambda_1 I$ positive semidefinite. Therefore, if $\lambda_1 \leq 0$ then Lemma 4.1 implies that x is a boundary solution for the TRS.

We can update the parameter by computing the value of α that satisfies the following nonlinear equation

$$\frac{1}{\sqrt{\phi'(\lambda)}} = \frac{1}{\Delta} \tag{5.7}$$

since once we have computed an eigenpair $\{\lambda, (1, x^T)^T\}$ of B_α, we can replace λ by $\alpha + g^T x$ and solve for α. Hebden [39] and Reinsch [65] pointed out the numerical advantages of solving (5.7) instead of (4.5). We can solve equation (5.7) by Newton's method or the Secant method. However, Newton's method requires the solution of linear systems with coefficient matrices of the form $H - \lambda I$ and the cost of solving these systems might be very high for large–scale problems.

Since (5.7) is an equation in one unknown, another possibility for updating the parameter α is by means of local rational approximations to $\phi(\lambda)$ and $\phi'(\lambda)$. This approach has the advantage of taking into account the rational structure of these functions. The use of rational interpolation in this context was discussed in [39] and [69].

In this approach, we construct rational interpolants $\hat{\phi}(\lambda)$ and $\hat{\phi}'(\lambda)$ for $\phi(\lambda)$ and $\phi'(\lambda)$, respectively. Then we compute $\bar{\lambda}$ such that $\dfrac{1}{\hat{\phi}'(\bar{\lambda})} = \dfrac{1}{\Delta^2}$ and update the parameter as $\alpha = \bar{\lambda} + \hat{\phi}(\bar{\lambda})$. This choice of the parameter needs to be safeguarded to guarantee convergence.

Sorensen in [71] reports that this strategy for updating α is more efficient than the use of the Secant method to solve (5.7). Rational interpolation is used in [71], [68], [67] and also in the present work. We describe the interpolation schemes in §5.5.3 and the safeguarding mechanism in §5.5.4.

If in the process of updating α we find $\lambda_1 > 0$ and $\|x\| < \Delta$ then we can claim that the problem has an interior solution. To see this, recall from Section 4.2 that the TRS has no solutions on the boundary if and only if H is positive definite and $\|H^{-1}g\| < \Delta$. We now show that these two conditions are satisfied if $\lambda_1 > 0$ and $\|x\| < \Delta$.

Observe that $0 < \lambda_1$ implies that H is positive definite since $\lambda_1 \leq \delta_1$. This also implies that H is nonsingular.

Observe now that $\phi'(\lambda_1) = \|(H - \lambda_1 I)^\dagger g\|^2 = \|x\|^2 < \Delta^2$ and recall that $\phi'(\lambda)$ is strictly increasing in $(-\infty, \delta_1)$. Therefore, since $\lambda_1 > 0$ and $\phi'(\lambda_1) < \Delta^2$, then $\phi'(0) = \|H^\dagger g\|^2 < \Delta^2$. Note that H nonsingular implies $H^\dagger = H^{-1}$ and therefore $\|H^{-1}g\| < \Delta$.

The computation of interior solutions for general problems requires the solution of linear systems. For discrete ill–posed problems, these systems are very ill–conditioned and other strategies are needed in this case. We discuss our approach for computing interior solutions for these problems in §5.5.6.

Up to this point in our discussion we have assumed that we can normalize an eigenvector associated with λ_1 to have first component one, and that $\lambda_1 \neq \delta_i, \;\; i = 1, 2, \ldots, n$. The strategy we presented for solving the TRS will break down if the normalization cannot be carried out or if $\lambda_1 = \delta_i$, $1 \leq i \leq n$. In Section 5.2 we will see the connection between this situation and the hard case. In Sections 5.2 and 5.3 we develop the necessary tools for the treatment of this case.

5.2 Characterization of the Hard Case

We recall that a potential hard case arises whenever g is orthogonal to the eigenspace associated with the smallest eigenvalue of H.

In this section we present a characterization of potential hard cases in terms of the parameterized eigenvalue problem introduced in Section 5.1. We show that a potential hard case is related to the case when we cannot normalize an eigenvector associated with the smallest eigenvalue of B_α, and we develop the bases for the numerical treatment of this case.

The first result shows a very interesting relationship between the first components of the eigenvectors of B_α, and the orthogonality of g with respect to the eigenspaces of H.

Lemma 5.1 For any $\alpha \in \mathbb{R}$ and any $q \in \mathcal{S}_i$, $1 \leq i \leq n$, $\{\delta_i, (0, q^T)^T\}$ is an eigenpair of B_α if and only if g is orthogonal to $\mathcal{S}_i$.

Proof The proof follows from the observation that $g \perp \mathcal{S}_i$ and $Hq = \delta_i q$ are equivalent to

$$\begin{pmatrix} \alpha & g^T \\ g & H \end{pmatrix} \begin{pmatrix} 0 \\ q \end{pmatrix} = \delta_i \begin{pmatrix} 0 \\ q \end{pmatrix} .$$

□

If δ_i are eigenvalues of H for $i = 1, 2, \ldots, \ell$ with $1 \leq \ell < n$ in Lemma 5.1, then this result establishes that for any value of α in a potential hard case we have that the δ_i's are eigenvalues of B_α with eigenvectors that have zero first components. Lemma 5.2 shows that for each i, $1 \leq i \leq n$, such that $g \perp \mathcal{S}_i$, there exists a special value of α for which B_α will also have an eigenvector with first component one, associated with δ_i.

Lemma 5.2 Suppose that g is orthogonal to $\mathcal{S}_i$, $1 \leq i \leq n$, and let $p = -(H - \delta_i I)^\dagger g$. The pair $\{\delta_i, (1, p^T)^T\}$ is an eigenpair of B_α if and only if $\alpha = \tilde{\alpha}_i$ where $\tilde{\alpha}_i = \delta_i - g^T p$.

Proof First we observe that $g \perp \mathcal{S}_i$ implies that $g \in \mathcal{R}(H - \delta_i I)$ and therefore

$$(H - \delta_i I)p = -(H - \delta_i I)(H - \delta_i I)^\dagger g = -g, \tag{5.8}$$

since $(H - \delta_i I)(H - \delta_i I)^\dagger$ is an orthogonal projection onto $\mathcal{R}(H - \delta_i I)$.

Suppose now that $\tilde{\alpha}_i = \delta_i - g^T p$ with $p = -(H - \delta_i I)^\dagger g$. Then

$$\begin{pmatrix} \tilde{\alpha}_i & g^T \\ g & H \end{pmatrix} \begin{pmatrix} 1 \\ p \end{pmatrix} = \begin{pmatrix} \tilde{\alpha}_i + g^T p \\ g + Hp \end{pmatrix} = \delta_i \begin{pmatrix} 1 \\ p \end{pmatrix},$$

since by definition of $\tilde{\alpha}_i$ we have that $\tilde{\alpha}_i + g^T p = \delta_i$ and by (5.8), $g + Hp = \delta_i p$.

To prove the other implication, let $p = -(H - \delta_i I)^\dagger g$ and suppose that $\{\delta_i, (1, p^T)^T\}$ is an eigenpair of B_α, i.e.

$$\begin{pmatrix} \alpha & g^T \\ g & H \end{pmatrix} \begin{pmatrix} 1 \\ p \end{pmatrix} = \delta_i \begin{pmatrix} 1 \\ p \end{pmatrix}.$$

It follows directly from this relationship that $\alpha = \tilde{\alpha}_i = \delta_i - g^T p$. □

Corollary 5.1 Suppose that g is orthogonal to $\mathcal{S}_i$, $1 \le i \le n$, and let $\mathcal{Z}_i(\alpha) = \{z \in \mathbb{R}^{n+1} \mid B_\alpha z = \delta_i z\}$. If $\tilde{\alpha}_i = \delta_i + g^T p$ with $p = -(H - \delta_i I)^\dagger g$ then $\dim \mathcal{Z}_i(\tilde{\alpha}_i) = \dim \mathcal{S}_i + 1$ and for any other value of α, $\dim \mathcal{Z}_i(\alpha) = \dim \mathcal{S}_i$. Moreover, if $\{q_1, \ldots, q_r\}$ is an orthogonal basis for $\mathcal{S}_i$ then

$$\left\{ \begin{pmatrix} 1 \\ p \end{pmatrix}, \begin{pmatrix} 0 \\ q_1 \end{pmatrix}, \ldots, \begin{pmatrix} 0 \\ q_r \end{pmatrix} \right\}$$

is an orthogonal basis for $\mathcal{Z}_i(\tilde{\alpha}_i)$ and

$$\left\{ \begin{pmatrix} 0 \\ q_1 \end{pmatrix}, \ldots, \begin{pmatrix} 0 \\ q_r \end{pmatrix} \right\}$$

is an orthogonal basis for $\mathcal{Z}_i(\alpha)$, for $\alpha \neq \tilde{\alpha}_i$.

Lemmas 5.1 and 5.2 are straightforward generalizations of Lemmas 3 and 4 from [67], respectively, where the results are established for $i = 1$ only. The proof of Lemma 5.1 is similar to the one of Lemma 3. The proof of Lemma 5.2 follows a different strategy from the proof of Lemma 4.

These lemmas provide the main tools to prove Theorem 5.1, which establishes that there always exists an eigenvector of B_α with first component one. This eigenvector will correspond to the smallest eigenvalue of B_α, except in potential hard cases where, depending on the value of α, it will correspond to the ith eigenvalue of B_α, for $1 \le i \le n$.

5.3 Treatment of the Hard Case

In this section we present the theoretical bases for our treatment of potential hard cases, when we might not be able to normalize an eigenvector associated with the smallest eigenvalue of B_α to have first component one.

Our first result, Theorem 5.1, was essentially presented in [67] but we present it here in a more general way and give a different proof that emphasizes the importance of the result.

Theorem 5.1 Let $\lambda(\alpha)$ be the smallest solution of the equation

$$\phi(\lambda) = \lambda - \alpha \; . \tag{5.9}$$

Then, for any value of α, $\lambda(\alpha)$ is an eigenvalue of B_α that has a corresponding eigenvector that can be normalized to have first component one.

Proof Suppose first that g is orthogonal to $\mathcal{S}_i$, $i = 1, 2, \ldots, \ell$ with $1 \leq \ell < n$. We show next that for any $\alpha \in \mathbb{R}$ there exists an eigenvector of B_α that can be normalized to have first component one.

Since $\gamma_i = 0, \;\; i = 1, 2, \ldots, \ell$ it follows that

$$\begin{aligned} \phi(\lambda) &= g^T (H - \lambda I)^\dagger g \\ &= \sum_{j=\ell+1}^{n} \frac{{\gamma_i}^2}{\delta_j - \lambda}. \end{aligned}$$

Let $\lambda(\alpha)$ be the smallest solution of equation (5.9). Then $\lambda(\alpha) \in (-\infty, \delta_{\ell+1})$, and since $\phi(\lambda)$ is strictly increasing on its domain and $\lambda - \alpha$ is a decreasing straight line, we conclude that $\lambda(\alpha)$ is unique.

Since $\lambda(\alpha)$ depends continuously on α, so does $p(\alpha) = -(H - \lambda(\alpha)I)^\dagger g$ and also $v(\alpha) = (1, p(\alpha)^T)^T$. Moreover, $v(\alpha)$ is an eigenvector of B_α associated with $\lambda(\alpha)$. To see this, consider

$$\begin{pmatrix} \alpha & g^T \\ g & H \end{pmatrix} \begin{pmatrix} 1 \\ p(\alpha) \end{pmatrix} = \begin{pmatrix} \alpha + g^T p(\alpha) \\ g + Hp(\alpha) \end{pmatrix}$$

and note that

$$\begin{aligned} \alpha + g^T p(\alpha) &= \alpha - g^T (H - \lambda(\alpha) I)^{\dagger} g \\ &= \alpha - \phi(\lambda(\alpha)) \\ &= \lambda(\alpha), \quad \text{by definition of } \lambda(\alpha). \end{aligned}$$

Now, $g \perp \mathcal{S}_i$, $i = 1, 2, \ldots, \ell$ implies that $g \in \mathcal{R}(H - \lambda I)$ for $\lambda \in (-\infty, \delta_{\ell+1})$. Therefore $g \in \mathcal{R}(H - \lambda(\alpha) I)$ and we can conclude that $(H - \lambda(\alpha) I)p = -g$ as in the proof of Lemma 5.2. From this it follows that

$$g + Hp(\alpha) = \lambda(\alpha) p(\alpha).$$

Therefore, $B_\alpha v(\alpha) = \lambda(\alpha) v(\alpha)$.

Suppose now that g is not orthogonal to $\mathcal{S}_1$. Then $\lambda(\alpha) \in (-\infty, \delta_1)$ is the smallest eigenvalue of B_α and Lemma 5.1 implies that any eigenvector associated with $\lambda(\alpha)$ can be normalized to have first component one, independently of the value of α. □

Depending on the value of α, we can characterize the first ℓ or $\ell + 1$ eigenvalues of B_α. This is established in the following lemma.

Lemma 5.3 Let $\lambda(\alpha)$ be the smallest solution of (5.9) and let $v(\alpha)$ be the corresponding eigenvector as in Theorem 5.1. Let λ_j, $j = 1, 2, \ldots, n+1$, be the eigenvalues of B_α in increasing order. Define $\tilde{\alpha}_i$ as in Lemma 5.2. Let $1 \le \ell < n$ and suppose $g \perp \mathcal{S}_i$, $i = 1, 2, \ldots, \ell$. Then:

(1) If $\alpha = \tilde{\alpha}_i$, $1 \le i \le \ell$ then $\lambda_i = \lambda(\alpha) = \delta_i$ and $\lambda_j = \delta_j$, $j = 1, 2, \ldots, \ell$. Moreover the multiplicity of λ_i is equal to the multiplicity of δ_i plus one.

(2) If $\alpha < \tilde{\alpha}_1$ then $\lambda_1 = \lambda(\alpha)$ and $\lambda_{j+1} = \delta_j$, $j = 1, 2, \ldots, \ell$.

(3) If $\tilde{\alpha}_{i-1} < \alpha < \tilde{\alpha}_i$, $i = 1, 2, \ldots, \ell$ then $\lambda_i = \lambda(\alpha)$, $\lambda_j = \delta_j$ for $j = 1, \ldots, i-1$, and $\lambda_{j+1} = \delta_j$ for $j = i+1, \ldots, \ell$.

(4) If $\alpha > \tilde{\alpha}_\ell$ then $\lambda_{\ell+1} = \lambda(\alpha)$ and $\lambda_j = \delta_j$, $j = 1, 2, \ldots, \ell$.

Proof These results are a direct consequence of Cauchy's Interlace Theorem and the properties of the functions $\phi(\lambda)$ and $\lambda - \alpha$. □

In Figure 5.1 we present an example to illustrate Lemma 5.3. In this example $n = 7$, $\ell = 4$, the eigenvalues of H are $\{-2, 0, 2, 3, 5, 7, 9\}$ and $\alpha = \tilde{\alpha}_3$. Therefore the first four eigenvalues of $B_{\tilde{\alpha}_3}$ are $-2, 0, 2$ and 3. The eigenvalue 2 has multiplicity two.

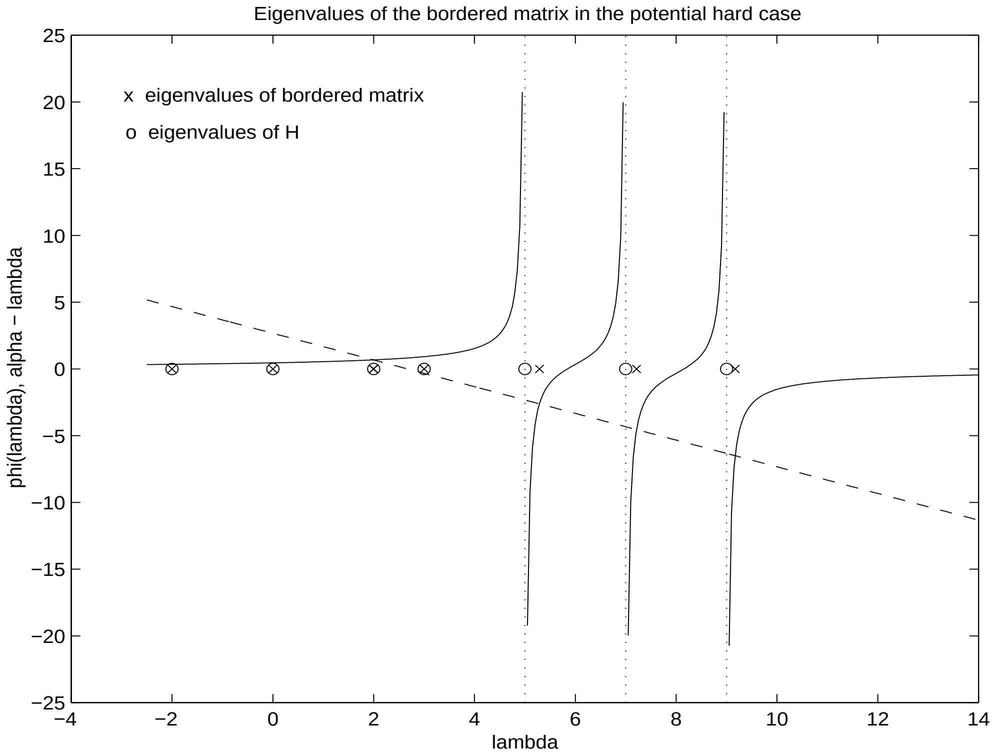

Figure 5.1 Eigenvalues of the Bordered Matrix in a Potential Hard Case.

Theorem 5.1 along with Theorem 5.2, constitute the key results for our treatment of potential hard cases.

Our next result, Lemma 5.4, is a technical result used in the proof of Theorem 5.2.

Lemma 5.4 Let $z_i = (\nu_i, \hat{z}_i^T)^T \in \mathbb{R}^{n+1}$, $i = 1, 2, \ldots, k$ and define the matrices $Z = [z_1 \ \ z_2 \ \ \cdots \ \ z_k]$ and $\hat{Z} = [\hat{z}_1 \ \ \hat{z}_2 \ \ \cdots \ \ \hat{z}_k]$. Assume

(i) $B_\alpha z_1 = \lambda_1 z_1$, with λ_1 the smallest eigenvalue of B_α.

(ii) $Z^T Z = I_k$.

If $e_1^T Z Z^T e_1 \geq \dfrac{1}{1+\Delta^2}$ then $\exists\, t \in \mathbb{R}^k$ with $\|t\| = 1$ such that

(1) $\dfrac{\|Zt\|^2}{(e_1^T Z t)^2} = 1 + \Delta^2$

(2) $\alpha + 2\psi(\hat{x}) = \hat{\lambda}(1+\Delta^2)$, where $\hat{\lambda} = t^T Z^T B_\alpha Z t$ and $\hat{x} = \dfrac{\hat{Z}t}{e_1^T Z t}$.

Proof In order to prove (1), consider

$$\frac{\|Zt\|^2}{(e_1^T Z t)^2} = 1 + \Delta^2 \tag{5.10}$$

and let us derive the conditions for this equation to have a solution $t \in \mathbb{R}^k$, with $t \neq 0$. Observe that we can rewrite equation (5.10) as

$$\begin{aligned} t^T Z^T Z t &= (1+\Delta^2)\,(e_1^T Z t)^2 \\ &= (1+\Delta^2)\,(t^T Z^T e_1 e_1^T Z t) \end{aligned}$$

which is equivalent to

$$t^T \, [I_k - (1+\Delta^2)\; Z^T e_1 e_1^T Z]\; t = 0 \tag{5.11}$$

since $Z^T Z = I_k$ by hypothesis.

Observe that for equation (5.11) to have a solution $t \neq 0$, the matrix $M = I - (1+\Delta^2)\ Z^T e_1 e_1^T Z$ has to be indefinite or positive semidefinite and singular. In order to see when this holds, let us study the eigenvalues of M.

The eigenpairs $\{\lambda_i, y_i\}$ of the matrix $M = I - (1+\Delta^2)\ Z^T e_1 e_1^T Z$ are given by

$$\begin{aligned} \lambda_1 &= 1 - (1+\Delta^2)\ e_1^T Z Z^T e_1, \quad y_1 = Z^T e_1 \\ \lambda_i &= 1, \quad y_i \in \mathbb{R}^k \quad \text{such that} \quad y_i \perp Z^T e_1, \quad i = 2, \ldots, k\ . \end{aligned}$$

Thus, the matrix M will be indefinite if $\lambda_1 < 0$ and positive semidefinite and singular if $\lambda_1 = 0$, i.e. equation (5.11) will have nontrivial solutions if $\lambda_1 = 1 - (1+\Delta^2)\ e_1^T Z Z^T e_1 \leq 0$.

Therefore, if $e_1^T Z Z^T e_1 \geq \dfrac{1}{1+\Delta^2}$ then $\exists\ t \in \mathbb{R}^k,\ t \neq 0$, so that (5.10) is satisfied. Note that since $t \neq 0$, we can normalize it to have $\|t\| = 1$. It is straightforward to see now that for such t, $\|Zt\| = 1$. Therefore

$$(e_1^T Z t)^2 = \frac{1}{1+\Delta^2}\cdot \tag{5.12}$$

Since this implies that $e_1^T Z t \neq 0$, we can claim (1), i.e. $\dfrac{\|Zt\|^2}{(e_1^T Z t)^2} = 1 + \Delta^2$.

To see that (2) holds, note that

$$\begin{aligned} \alpha + 2\psi(\hat{x}) &= (1\ \ \hat{x}^T) \begin{pmatrix} \alpha & g^T \\ g & H \end{pmatrix} \begin{pmatrix} 1 \\ \hat{x} \end{pmatrix} \\ &= (1\ \ \hat{x}^T)\ B_\alpha \begin{pmatrix} 1 \\ \hat{x} \end{pmatrix} \end{aligned}$$

and since $(1, \hat{x}^T)^T = \dfrac{1}{(e_1^T Z t)} Z t$, it follows that

$$\begin{aligned} \alpha + 2\psi(\hat{x}) &= \frac{1}{(e_1^T Z t)^2} t^T Z^T B_\alpha Z t \\ &= \hat{\lambda} \frac{1}{(e_1^T Z t)^2}\cdot \end{aligned}$$

Now if $e_1^T Z Z^T e_1 \geq \dfrac{1}{1+\Delta^2}$ then using (5.12) we have

$$\alpha + 2\psi(\hat{x}) \quad = \quad \hat{\lambda}(1+\Delta^2)$$

as claimed. □

The following Lemma now establishes a way of computing the solutions for equations of the form of equation (5.11).

Lemma 5.5 Let $\beta \in \mathbb{R}$, $z \in \mathbb{R}^n$. The equation

$$t^T[I - \beta z z^T]t = 0 \tag{5.13}$$

in t with $t \in \mathbb{R}^n$, has $2(n-1)$ nontrivial solutions if the matrix $M = I - \beta z z^T$ is indefinite and one nontrivial solution if M is positive semidefinite and singular.

Proof Let $P \in \mathbb{R}^{n \times n}$ be such that $P^T z = \|z\| e_1$ with $P^T P = I$ and apply this orthogonal transformation to the matrix M to obtain

$$P^T[I - \beta z z^T]P = I - \beta \|z\|^2 e_1 e_1^T.$$

Therefore the solutions of equation (5.13) in this new basis are the solutions of

$$y^T \begin{pmatrix} -\Theta & 0 \\ 0 & I \end{pmatrix} y = 0 \tag{5.14}$$

where $\Theta = -1 + \beta \|z\|^2 e_1 e_1^T$.

The nontrivial solutions of (5.13) are then given by $t = Py$ where

(1) $y = (1, \sqrt{\Theta} e_i^T)^T$ or $y = (-1, \sqrt{\Theta} e_i^T)^T$ with e_i the ith canonical vector in $\mathbb{R}^{n-1}$, $i = 1, 2, \ldots, n-1$, if M is indefinite, i.e. if $\Theta > 0$, or

(2) $y = e_1$, if M is positive semidefinite and singular, i.e. if $\Theta = 0$.

Therefore equation (5.13) has $2(n-1)$ nontrivial solutions if M is indefinite and one nontrivial solution if M is positive semidefinite and singular. □

Now we are ready to present our main result.

Theorem 5.2 Let λ_1 be the smallest eigenvalue of B_α with corresponding eigenvector $z_1 = (\nu_1, \hat{z}_1^T)^T$. Let λ_i, $i = 2, \ldots, k$ be *any* $k-1$ of the remaining n eigenvalues of B_α with corresponding eigenvectors $z_i = (\nu_i, \hat{z}_i^T)^T$.

Define $Z = [z_1 \;\; z_2 \;\; \cdots \;\; z_k]$ and $\hat{Z} = [\hat{z}_1 \;\; \hat{z}_2 \;\; \cdots \;\; \hat{z}_k]$. Let $\eta > 0$.

If

(i) $\exists \; t = (\tau_1 \; \tau_2 \; \ldots \; \tau_k)^T$, $\|t\| = 1$ such that $\dfrac{\|Zt\|^2}{(e_1^T Z t)^2} = 1 + \Delta^2$.

(ii) $0 \le \sum_{i=2}^{k} (\lambda_i - \lambda_1) \; \tau_i{}^2 \; (1 + \Delta^2) \le -2\eta\psi(\hat{x})$,
for $\hat{x} = \dfrac{\hat{Z}t}{e_1^T Z t}$.

Then

$$\psi(x_*) \le \psi(\hat{x}) \le \frac{1}{1+\eta}\psi(x_*)$$

where x_* is a boundary solution for the TRS (5.1) with $\psi(x_*) \le 0$.

Proof Observe that because x_* is a boundary solution of (5.1), we have that $\psi(x_*) \le \psi(x)$, $\forall \; x \in \mathbb{R}^n$ such that $\|x\| = \Delta$. Therefore, in order to prove that $\psi(x_*) \le \psi(\hat{x})$, it will suffice to show that $\|\hat{x}\| = \Delta$.

First notice that

$$\frac{Zt}{e_1^T Z t} = \frac{1}{e_1^T Z t} \begin{pmatrix} e_1^T Z \\ \hat{Z} \end{pmatrix} t$$

which implies that

$$\begin{aligned} \left\| \frac{Zt}{e_1^T Zt} \right\|^2 &= \frac{t^T(Z^T e_1 e_1^T Z)t + t^T \hat{Z}^T \hat{Z} t}{(e_1^T Zt)^2} \\ &= 1 + \frac{t^T \hat{Z}^T \hat{Z} t}{(e_1^T Zt)^2} && (5.15) \\ &= 1 + \Delta^2, \quad \text{by } (i). && (5.16) \end{aligned}$$

Since $\|\hat{x}\|^2 = \frac{t^T \hat{Z}^T \hat{Z} t}{(e_1^T Zt)^2}$, (5.15) and (5.16) imply that $\|\hat{x}\| = \Delta$ and therefore, $\psi(x_*) \leq \psi(\hat{x})$.

Let us now prove the other part of the inequality.

Observe that $\alpha + 2\psi(x_*) = (1 \quad x_*^T)\ B_\alpha\ (1, x_*^T)^T$. Thus, by Rayleigh quotient properties

$$\alpha + 2\psi(x_*) \ \geq \ \lambda_1 \|(1, x_*^T)^T\|^2.$$

And since $\|x_*\| = \Delta$, it follows that $\|(1, x_*^T)^T\|^2 = 1 + \Delta^2$, and therefore

$$\alpha + 2\psi(x_*) \geq \lambda_1(1 + \Delta^2). \qquad (5.17)$$

Now, observe that $\{\lambda_i, z_i\}$ satisfies conditions *(i)–(ii)* in Lemma 5.4 and observe also that by hypothesis we have $t \in \mathbb{R}^k$ with $\|t\| = 1$ and $\left\| \frac{Zt}{e_1^T Zt} \right\|^2 = 1 + \Delta^2$. Therefore, as in Lemma 5.4 we have that $\hat{\lambda} = \sum_{i=1}^{k} {\tau_i}^2 \lambda_i$ and $\hat{x} = \frac{\hat{Z}t}{e_1^T Zt}$ satisfy

$$\begin{aligned} \alpha + 2\psi(\hat{x}) &= \hat{\lambda}\ (1 + \Delta^2) \\ &= \sum_{i=1}^{k} {\tau_i}^2 \lambda_i\ (1 + \Delta^2) \end{aligned}$$

and since $\sum_{i=2}^{k} {\tau_i}^2 = 1$, we have

$$\begin{aligned} \alpha + 2\psi(\hat{x}) &= \left[\left(1 - \sum_{i=2}^{k} {\tau_i}^2 \right) \lambda_1 + \sum_{i=2}^{k} {\tau_i}^2 \lambda_i \right] (1 + \Delta^2) \\ &= \left[\lambda_1 + \sum_{i=2}^{k} (\lambda_i - \lambda_1) {\tau_i}^2 \right] (1 + \Delta^2) \end{aligned}$$

Therefore

$$\begin{aligned} \alpha + 2\psi(\hat{x}) - \sum_{i=2}^{k}(\lambda_i - \lambda_1)\tau_i^{\,2}(1+\Delta^2) &= \lambda_1(1+\Delta^2) \\ &\leq \alpha + 2\psi(x_*), \quad \text{by (5.17).} \end{aligned}$$

If $\sum_{i=2}^{k}(\lambda_i - \lambda_1)\tau_i^{\,2}(1+\Delta^2) \leq -2\eta\psi(\hat{x})$ holds, then

$$\begin{aligned} \alpha + 2\psi(\hat{x}) + 2\eta\psi(\hat{x}) &\leq \alpha + 2\psi(x_*) \\ \alpha + 2(1+\eta)\psi(\hat{x}) &\leq \alpha + 2\psi(x_*) \end{aligned}$$

and we can conclude $\psi(\hat{x}) \leq \frac{1}{1+\eta}\psi(x_*)$.

Therefore $\psi(x_*) \leq \psi(\hat{x}) \leq \frac{1}{1+\eta}\psi(x_*)$ as claimed. □

It follows directly from this result that

$$\begin{aligned} 0 \leq \psi(\hat{x}) - \psi(x_*) \leq -\frac{\eta}{1+\eta}\psi(x_*) \\ 0 \leq |\psi(\hat{x}) - \psi(x_*)| \leq \frac{\eta}{1+\eta}|\psi(x_*)| \ . \end{aligned} \tag{5.18}$$

The inequality (5.18) implies that under the conditions of Theorem 5.2, $\psi(\hat{x})$ will be arbitrarily close to $\psi(x_*)$. We will call such $\hat{x}$ a quasi–optimal solution for (5.1).

The following corollary is of particular interest when solving large–scale problems. In this case we typically use $k = 2$ since the use of more vectors may be prohibitive.

Corollary 5.2 If $k = 2$ in Theorem 5.2 then the necessary eigenpairs are one associated with the smallest eigenvalue of B_α, and a second one associated with *any* of the remaining eigenvalues of B_α.

Theorem 5.2 and Lemma 5.4 provide the tools for approximating the vector $x = p + \tau z$ from Lemma 4.2, as a combination of eigenvectors of H. Theorem 5.2 also establishes conditions under which the vector x computed in this way is a quasi–optimal solution for problem (5.1).

Moreover, Theorem 5.2 guarantees that if we are working with only two vectors and the second smallest eigenvalue belongs to a cluster, as in discrete ill–posed problems, we can still build the special vector $\hat{x}$ from an eigenvector associated with the smallest eigenvalue of B_α, and an eigenvector associated with *any* eigenvalue of the cluster, not necessarily the second smallest. If the first eigenvalue and $k-1$ of the remaining ones and the corresponding eigenvectors of B_α are available for $k > 2$, we can use all of them to build a quasi–optimal solution for the TRS. This strategy might yield faster convergence than the one that uses only two eigenpairs and has yet to be tested.

Lemma 5.4 and Theorem 5.2 are generalizations of Lemmas 6 and 7 from [67], respectively. The differences are that in [67] the results are proved for two vectors only and in Lemma 7 these vectors are required to be eigenvectors associated with the two smallest eigenvalues of B_α. Our results hold for k vectors, $2 \leq k < n$ and in Theorem 5.2 we require an eigenpair associated with the smallest eigenvalue of B_α, and *any* other $k-1$ of the remaining eigenpairs.

Observe that Lemmas 5.4 and 5.5, respectively, provide a way of computing the vectors $\hat{x}$ and t needed in Theorem 5.2. We use Theorem 5.2, Lemma 5.4 and Lemma 5.5 in the design of one of the stopping criteria in our method. We describe this stopping rule in §5.5.6.

In Sections 5.4 and 5.5 we will present the complete description of our method for the Large–Scale Trust–Region Subproblem. Our method is the result of the evolution of the ideas originally presented in [71], [68] and [67]. We extended the ideas in the previous works to handle the high degree of singularities present in discrete ill–posed problems. This lead to the development of a complete theory for the treatment of the hard case.

The treatment of discrete ill–posed problems was not possible with the previous methods for several reasons.

In [71] there is a separate treatment of the hard case and the easy case. The iteration for the hard case has slower convergence than the one for the easy case. That work did not present robust safeguarding strategies and did not consider the possibility of a multiple hard case.

The method in [68] uses a different interpolation scheme from the one in [71], introduces a safeguarding strategy for the parameter and presents a unified treatment of all cases. The treatment of the hard case and the stopping criteria are the weak aspects of that method.

The method in [67] is based on [68] but uses a more robust safeguarding strategy and makes important contributions to the treatment of the hard case. It also presents more robust stopping criteria than the ones in [68]. This work does not consider the possibility of a multiple hard case.

Although our method is based on [67], we have introduced new elements in the treatment of the hard case, the safeguarding strategies and the stopping criteria. We have also designed an approach to handle interior solutions in the special case of discrete ill–posed problems.

5.4 Algorithm

In this section we present the general algorithm as Algorithm 5.4.1 (Figure 5.2). In Section 5.5 we describe each of the components of the algorithm in detail.

In the description of Algorithm 5.4.1. we use the following definitions and assumptions:

- The vector g satisfies $g \neq 0$. If $g = 0$ we compute an eigenvector x associated with the smallest eigenvalue of H and normalize it such that $\|x\| = \Delta$.
- $\alpha_* = \lambda_* - g^T x_*$, where $\{\lambda_*, x_*\}$ is the optimal pair from Lemma 4.1, except when there is only an interior solution where we define $x_* = -(H - \lambda_* I)^\dagger g$ such that $\|x_*\| = \Delta$.
- The values α_L, α_U are such that $\alpha_L \leq \alpha_* \leq \alpha_U$. These values are used to safeguard the parameter α and they are updated at each iteration. We describe the safeguarding strategy in §5.5.4.
- δ_1 is the smallest eigenvalue of H and δ_U is an upper bound for δ_1. The bound δ_U is used for safeguarding purposes and it is updated at each iteration. We describe this safeguarding strategy in §5.5.4.
- At iteration k, the pairs $\{\lambda_1,\ (\nu_1, u_1^T)^T\}$, $\{\lambda_2,\ (\nu_2, u_2^T)^T\}$ are eigenpairs of $B_{\alpha_{(k)}}$, where λ_1 is the smallest eigenvalue and λ_2 is a value equal or close to the second smallest eigenvalue.
- The pair $\{\lambda_{(k)},\ (\nu_{(k)}\ u_{(k)}^T)^T\}$ is used to compute $\{\lambda_{(k)}, x_{(k)}\}$, the current approximation to the optimal pair $\{\lambda_*, x_*\}$ and also to compute interpolation values.

  When used to compute $\{\lambda_{(k)}, x_{(k)}\}$, $\{\lambda_{(k)},\ (\nu_{(k)}\ u_{(k)}^T)^T\}$ is always equal to $\{\lambda_1,\ (\nu_1, u_1^T)^T\}$.

  When $\{\lambda_{(k)},\ (\nu_{(k)}\ u_{(k)}^T)^T\}$ is used to compute interpolation values, it might be equal to $\{\lambda_2,\ (\nu_2, u_2^T)^T\}$. This is decided in Step 3.4 of the algorithm. The criterion for choosing the interpolation values is described in §5.5.2.

Algorithm 5.4.1 Method for the Large–Scale Trust–Region Subproblem.
Input: $\varepsilon_\Delta,\ \varepsilon_{Int},\ \varepsilon_{HC},\ \varepsilon_\nu,\ \varepsilon_\alpha,\ \varepsilon_0 \in (0,1)$; $\Delta \in \mathbb{R},\ \Delta > 0$; $g \in \mathbb{R}^n,\ g \neq 0$; and $H \in \mathbb{R}^{n\times n}$ with $H = H^T$, or a procedure for computing Hv for $v \in \mathbb{R}^n$.
Output: an approximation to an optimal pair $\{\lambda_*, x_*\}$ for problem (5.1), given in Lemma 4.1.

```
1. Initial step
    1.2 Compute initial safeguards $\delta_U \geq \delta_1$ and $\alpha_U \geq \alpha_*$
    1.3 Compute $\alpha_{(0)} = \min\{0, \alpha_U\}$
    1.4 Compute $\{\lambda_1,\ (\nu_1, u_1^T)^T\}$, $\{\lambda_2,\ (\nu_2, u_2^T)^T\}$ two eigenpairs of $B_{\alpha_{(0)}}$,
        with $\lambda_1$ the smallest eigenvalue and $\lambda_2$ equal or close to the
        second smallest eigenvalue
    1.5 Set $\{\lambda_{(0)},\ (\nu_{(0)}\ u_{(0)}^T)^T\} = \{\lambda_1,\ (\nu_1, u_1^T)^T\}$
    1.6 Compute initial safeguard $\alpha_L \leq \alpha_*$
2. $k = 0$
3. while not convergence $(\{\lambda_{(k)},\ (\nu_{(k)}\ u_{(k)}^T)^T\})$ do
    3.1 Compute $\delta_U = \min\left\{\delta_U, \dfrac{u_1^T H u_1}{u_1^T u_1}\right\}$, if $u_1 \neq 0$
    3.2 Adjust $\alpha_{(k)}$ and compute $\{\lambda_1,\ (\nu_1, u_1^T)^T\}$, $\{\lambda_2,\ (\nu_2, u_2^T)^T\}$ two
        eigenpairs of $B_{\alpha_{(k)}}$ with $\lambda_1$ the smallest eigenvalue and $\lambda_2$ equal or
        close to the second smallest eigenvalue, until either $\nu_1$ or $\nu_2$ is large
    3.3 Select interpolation values $\{\lambda_{(k)},\ (\nu_{(k)}\ u_{(k)}^T)^T\}$ from
        $\{\lambda_1,\ (\nu_1, u_1^T)^T\}$ and $\{\lambda_2,\ (\nu_2, u_2^T)^T\}$
    3.4 if $(k = 0)$ then
            Compute $\alpha_{(k+1)}$ by one–point interpolation scheme from §5.5.3.
        else
            Compute $\alpha_{(k+1)}$ by two–point interpolation scheme from §5.5.3.
        end if
    3.5 Safeguard $\alpha_{(k+1)}$
    3.6 Compute $\{\lambda_1,\ (\nu_1, u_1^T)^T\}$, $\{\lambda_2,\ (\nu_2, u_2^T)^T\}$ two eigenpairs of $B_{\alpha_{(k+1)}}$,
        with $\lambda_1$ the smallest eigenvalue and $\lambda_2$ equal or close to the
        second smallest eigenvalue
    3.7 Set $\{\lambda_{(k+1)},\ (\nu_{(k+1)}\ u_{(k+1)}^T)^T\} = \{\lambda_1,\ (\nu_1, u_1^T)^T\}$
    3.8 Update safeguards $\alpha_L$ and $\alpha_U$
    3.9 $k = k + 1$
   end while
4. Compute $\lambda_*$, $x_*$ as in §5.5.6
```

Figure 5.2 Method for the Large–Scale Trust–Region Subproblem.

5.5 Components

In this section we describe each component of Algorithm 5.4.1. For each component we describe the theoretical aspects and discuss implementation issues.

Table 5.1 is intended as a guide to relate the steps of the algorithm with each description.

Steps	Discussed in
1.4 and 3.6 Solution of Eigenproblems	§5.5.1
3.3 Selection of Interpolation Values	§5.5.2
3.4 Interpolating Schemes	§5.5.3
1.2, 1.6, 3.5 and 3.8 Safeguarding	§5.5.4
3.2 Adjustment of α	§5.5.5
Stopping Criteria	§5.5.6

Table 5.1 Index of the Components of the Trust–Region Subproblem Method

5.5.1 Solution of Eigenproblems

We are interested in problems where the matrix H is large and may not be available explicitly. Instead a procedure for computing Hv is available, or in the special case when $H = A^TA$, procedures for computing A^Tw and Av are available.

As in [71], [68] and [67], we use the Implicitly Restarted Lanczos Method (IRLM) [70] as implemented in [43] to solve the eigenproblems. This method has the following advantages:

- H is used only in matrix–vector products. Note that in the special case when $H = A^T A$, it is not necessary to form this matrix explicitly.

- The storage requirement is of order $n \times k$, for k small and fixed.

- Fast convergence to the extreme eigenvalues if they are separated from the rest.

It is worth noting that for small values of Δ, the smallest eigenvalue of B_α will be separated from the rest and we can expect the IRLM to be very efficient in computing that eigenvalue.

However, the Lanczos method will have difficulties in computing the smallest eigenvalue of B_α if it belongs to a cluster (see [25, Ch. 9]). If the second smallest eigenvalue belongs to a cluster this does not pose any problem for our method, since we can build a quasi–optimal solution using any eigenvalue from the cluster as we discussed at the end of Section 5.2.

In discrete ill–posed problems we encounter several difficulties with respect to the solution of the eigenproblems. The first one is that the smallest eigenvalues of the matrix B_α are clustered even when Δ is not too large. This is due to the facts that potential hard cases are common and that the smallest eigenvalues of H are clustered. We still need to address the treatment of clustered eigenvalues in our implementation.

A second difficulty is due to the fact that the smallest eigenvalue of H is usually very close to zero. Again, for values of Δ that are not too large we will have that the smallest eigenvalue of B_α will also be close to zero. Any method based on matrix–vector products of the form $B_\alpha v$, will annihilate components precisely in the direction of the eigenvectors of interest. To avoid this problem we apply a Tchebyshev polynomial preconditioner that maps the smallest eigenvalue to a meaningful value.

Another issue is the computation of eigenvectors corresponding to eigenvalues with multiplicity greater than one. Recall that to update the parameter α in Step

3.4 we require the smallest eigenvalue of the bordered matrix and a corresponding eigenvector with significant first component. As we saw in Section 5.3, the smallest eigenvalue of B_α can have multiplicity greater than one with only one of the associated eigenvectors having a large first component. It is desirable that the eigensolver computes such eigenvector. The Implicitly Restarted Lanczos Method does not guarantee the computation of this eigenvector as a result of this, we could be adjusting the parameter α more than necessary in Step 3.2.

5.5.2 Selection of Interpolation Values

Given the two eigenpairs of $B_{\alpha_{(k)}}$, $\{\lambda_1,\ (\nu_1, u_1^T)^T\}$ and $\{\lambda_2,\ (\nu_2, u_2^T)^T\}$, we use the values $\lambda_{(k)} = \lambda_1$ and $x_{(k)} = \dfrac{u_1}{\nu_1}$ as interpolation values as long as ν_1 is different from zero.

In practice, ν_1 might be small in near potential hard cases so that dividing by this value might produce large errors. For example, if $\alpha > \tilde{\alpha}_1$ in a potential hard case all the eigenvectors associated with λ_1 will have small first component, as the results from Section 5.3 establish. Moreover, even if $\alpha \leq \tilde{\alpha}_1$ the eigensolver might not compute the eigenvector with large first component.

When ν_1 is too small, we use the pair $\{\lambda_2,\ (\nu_2, u_2^T)^T\}$ to compute the interpolation values. In Step 3.2 we adjust α in a way that guarantees that either ν_1 or ν_2 is sufficiently large. We discuss the adjustment procedure in §5.5.5.

Figure 5.3 presents the procedure for choosing the interpolation values from the two given eigenpairs of the bordered matrix. In this procedure, $\varepsilon_\nu \in (0, 1)$.

Note that if the test is not satisfied this means that ν_1 is large and there is no need to update $\{\lambda_{(k)},\ (\nu_{(k)}\ u_{(k)}^T)^T\}$ since it is already equal to $\{\lambda_1,\ (\nu_1, u_1^T)^T\}$ after Step 1.5 or Step 3.8 of the algorithm.

```
Step 3.3 Selection of Interpolation Values.
    if $(\|g\|\ |\nu_1| \le \varepsilon_\nu\sqrt{1-{\nu_1}^2})$ and $(|\lambda_1-\alpha|\ |\nu_1| \le \sqrt{\varepsilon_\nu})$ then
        Set $\{\lambda_{(k)},\ (\nu_{(k)}\ u_{(k)}^T)^T\} = \{\lambda_2,\ (\nu_2, u_2^T)^T\}$
    end if
```

Figure 5.3 Procedure for the Selection of the Interpolation Values.

The previous test is motivated as follows. Observe that at any iteration we have $(H-\lambda_1 I)u_1 = -g\nu_1$, with $\|(\nu_1, u_1^T)^T\| = 1$, and therefore

$$\frac{\|(H-\lambda_1 I)u_1\|}{\|u_1\|} = \frac{\|g\|\ |\nu_1|}{\sqrt{1-{\nu_1}^2}}.$$

Thus, $\|g\|\ |\nu_1| \le \varepsilon_\nu\sqrt{1-{\nu_1}^2}$ implies that $\|(H-\lambda_1 I)u_1\| \le \varepsilon_\nu\|u_1\|$, which can be made scale independent by choosing $\varepsilon_\nu = \varepsilon\|H\|$. Also observe that $|\lambda_1-\alpha|\ |\nu_1| \le \sqrt{\varepsilon_\nu}$ implies that $g^T u_1$ is small.

Therefore, if the test is satisfied then $\{\lambda_1, u_1\}$ is an approximate eigenpair of H with $g^T u_1$ small, which is in agreement with Lemma 5.1. We save the best approximation to a vector in $\mathcal{S}_1$, i.e. the best u_1, to build a solution if the hard case is detected (see §5.5.6) and the minimum length of the safeguarding interval has been reached.

Observe that $g^T u_1$ small implies that g is numericallly orthogonal to $\mathcal{S}_1$ if $\mathcal{S}_1 = \{u_1\}$.

5.5.3 Interpolating Schemes

As we discussed in Section 5.1 we update the parameter α by means of a local rational approximation to the functions $\phi(\lambda)$ and $\phi'(\lambda)$. This approach takes advantage of the rational structure of these functions and has proved to be very efficient in practice while yielding local superlinear convergence of the iteration (see §5.6.2).

As in [67] we use a two–point interpolating scheme in the main iteration. To start the iteration we require two initial sets of interpolation values. The first set is chosen from the eigenpairs of $B_{\alpha_{(0)}}$ computed in Step 1.4. The second set is obtained by building a one–point interpolant based on the first set of values.

We describe the one–point and the two–point interpolating schemes next.

One–point interpolating scheme

The interpolant is a rational function of the form

$$\hat{\phi}(\lambda) = \frac{\gamma^2}{\delta - \lambda}.$$

We now show how to compute γ^2 and δ.

Let λ_0 be the smallest eigenvalue of B_{α_0} with corresponding eigenvector $(1, x_0^T)^T$, i.e.

$$\alpha - \lambda_0 = -g^T x_0 \quad \text{and} \quad (H - \lambda_0 I)x_0 = -g.$$

We require that $\hat{\phi}(\lambda)$ and $\hat{\phi}'(\lambda)$ satisfy the following conditions:

(i) $\hat{\phi}(\lambda_0) = \phi(\lambda_0) = -g^T x_0.$

(ii) $\hat{\phi}'(\lambda_0) = \phi'(\lambda_0) = x_0^T x_0.$

A straightforward calculation now yields

$$\delta = \lambda_0 - \frac{g^T x_0}{x_0^T x_0} \quad \text{and} \quad \gamma^2 = \frac{(g^T x_0)^2}{x_0^T x_0}.$$

The following expression yields $\bar{\lambda}$ such that $\hat{\phi}'(\bar{\lambda}) = \Delta^2$

$$\bar{\lambda} = \delta \ + \ \frac{g^T x_0}{\|x_0\|\Delta}$$

and some algebraic manipulation leads to the formula for α_1

$$\alpha_1 = \bar{\lambda} + \hat{\phi}(\bar{\lambda}) = \alpha_0 + \frac{\alpha_0 - \lambda_0}{\|x_0\|}\left(\frac{\Delta - \|x_0\|}{\Delta}\right)\left(\Delta + \frac{1}{\|x_0\|}\right). \tag{5.19}$$

Note that this interpolation scheme could also be used in the main iteration. However, as observed in [71], this choice yields a linearly convergent method and will not be used. Instead we will update the parameter through a two–point interpolating scheme which yields faster convergence. We describe this scheme next.

Two–point interpolating scheme

To build the two–point interpolant at iteration $k+1$, $k \geq 1$, we use the interpolation values at iteration $k-1$ and at iteration k, namely $\{\lambda_{(k-1)},\ (\nu_{(k-1)}\ u^T_{(k-1)})^T\}$ and $\{\lambda_{(k)},\ (\nu_{(k)}\ u^T_{(k)})^T\}$.

From these values we compute,

$$x_{(i)} = \frac{u_{(i)}}{\nu_{(i)}}$$
$$\phi_i = \phi(\lambda_{(i)}) = -g^T x_{(i)}$$
$$\phi'_i = \phi'(\lambda_{(i)}) = x^T_{(i)} x_{(i)}$$

for $i = k-1,\ k$.

The rational interpolant that we use in the two–point scheme is of the form

$$\hat{\phi}(\lambda) = \frac{\gamma^2}{\delta - \lambda} + \eta,$$

for which $\hat{\phi}'(\lambda) = \dfrac{\gamma^2}{(\delta-\lambda)^2}$. Since we know the values ϕ'_{k-1}, ϕ'_k we can compute γ^2 and δ, which are given by

$$\gamma^2 = \frac{(\lambda_{(k)} - \lambda_{(k-1)})^2 \|x_{(k-1)}\|^2 \|x_{(k)}\|^2}{(\|x_{(k)}\| - \|x_{(k-1)}\|)^2} \qquad \text{and} \qquad \delta = \frac{\lambda_{(k)}\|x_{(k)}\| - \lambda_{(k-1)}\|x_{(k-1)}\|}{\|x_{(k)}\| - \|x_{(k-1)}\|}\ .$$

Knowing γ^2 and δ, we can compute $\bar{\lambda}$ such that $\dfrac{1}{\hat{\phi}'(\bar{\lambda})} = \dfrac{1}{\Delta^2}$. This value can be shown to be

$$\bar{\lambda} = \frac{\lambda_{(k-1)}\|x_{(k-1)}\|(\|x_{(k)}\| - \Delta) + \lambda_{(k)}\|x_{(k)}\|(\Delta - \|x_{(k-1)}\|)}{\Delta(\|x_{(k)}\| - \|x_{(k-1)}\|)}\ . \tag{5.20}$$

Recall now that the parameter α is updated using the formula $\alpha = \bar{\lambda} + \hat{\phi}(\bar{\lambda})$. To evaluate $\hat{\phi}(\bar{\lambda})$ we need to choose a value for η. Our choice is

$$\eta = \left(\frac{\lambda_{(k)} - \bar{\lambda}}{\lambda_{(k)} - \lambda_{(k-1)}} \right) \eta_{k-1} + \left(\frac{\bar{\lambda} - \lambda_{(k-1)}}{\lambda_{(k)} - \lambda_{(k-1)}} \right) \eta_k$$

where we define $\eta_i = \phi_i - \dfrac{\gamma^2}{\delta - \lambda_{(i)}}$, for $i = k-1,\ k$. This choice makes use of the rest of the information available, namely $\phi_{k-1},\ \phi_k$.

After some manipulation and defining $\omega \equiv \dfrac{\lambda_{(k)} - \bar{\lambda}}{\lambda_{(k)} - \lambda_{(k-1)}}$, the formula for updating α can be expressed as

$$\begin{aligned}
\alpha_{(k+1)} &= \bar{\lambda} + \omega\phi_{k-1} + (1-\omega)\phi_k \\
&+ \frac{\|x_{(k-1)}\|\|x_{(k)}\|(\|x_{(k)}\| - \|x_{(k-1)}\|)}{\omega\|x_{(k)}\| + (1-\omega)\|x_{(k-1)}\|} \; \frac{(\lambda_{(k-1)} - \bar{\lambda})(\lambda_{(k)} - \bar{\lambda})}{(\lambda_{(k)} - \lambda_{(k-1)})} \\
&= \omega\alpha_{(k-1)} + (1-\omega)\alpha_{(k)} \\
&+ \frac{\|x_{(k-1)}\|\|x_{(k)}\|(\|x_{(k)}\| - \|x_{(k-1)}\|)}{\omega\|x_{(k)}\| + (1-\omega)\|x_{(k-1)}\|} \; \frac{(\lambda_{(k-1)} - \bar{\lambda})(\lambda_{(k)} - \bar{\lambda})}{(\lambda_{(k)} - \lambda_{(k-1)})} \qquad (5.21)
\end{aligned}$$

where $\alpha_i = \lambda_{(i)} + \phi_i$, $i = k-1, k$.

5.5.4 Safeguarding

There are two quantities that we need to safeguard in Algorithm 5.4.1. These are the parameter α and $\bar{\lambda}$, the value that satisfies $\hat{\phi}'(\bar{\lambda}) = \Delta^2$. We describe the safeguarding strategies next.

Safeguarding $\bar{\lambda}$

Since there is the possibility that we will use an eigenpair of B_α associated with the second smallest eigenvalue or with a value close to it to build the interpolants, it is possible that $\bar{\lambda} > \delta_1$. Using such $\bar{\lambda}$ to compute $\alpha = \bar{\lambda} + \hat{\phi}(\bar{\lambda})$ might yield values of α that are far away from the optimal one. Another situation that requires safeguarding $\bar{\lambda}$ is the occurrence of a zero or small denominator in (5.20).

We set $\bar{\lambda} = \delta_U$, where $\delta_U \geq \delta_1$, if $\bar{\lambda} > \delta_U$ or if we cannot safely compute $\bar{\lambda}$ by means of (5.20).

To obtain an initial value for δ_U we observe that any Rayleigh quotient is an upper bound for δ_1. Therefore we define the initial δ_U as:

(a) $\delta_U = \min\ e_i^T H e_i$ for $1 \leq i \leq n$, if the diagonal elements of H are available, or

(b) $\delta_U = \dfrac{v^T H v}{v^T v}$ for a random vector $v \in \mathbb{R}^n$, $v \neq 0$, otherwise.

We update this bound in Step 3.1 of Algorithm 5.4.1 by

$$\delta_U = \min\left\{\delta_U, \frac{u_1^T H u_1}{u_1^T u_1}\right\},$$

if $u_1 \neq 0$, where $(\nu_1, u_1^T)^T$ is an eigenvector corresponding to the smallest eigenvalue of $B_{\alpha_{(k)}}$.

Observe that $\bar{\lambda} > \delta_U$ might occur because we used an eigenpair associated with the second smallest eigenvalue of B_α to compute the interpolation values. As we saw in §5.5.2, this choice of interpolation values implies two things. First, that u_1 is a good approximation to an eigenvector of H associated with δ_1, i.e. that $\dfrac{u_1^T H u_1}{u_1^T u_1}$ is a sharp estimate for δ_1. And second, that $g^T u_1$ is small which in turns implies that we are in a potential hard case if $\mathcal{S}_1 = \{u_1\}$.

Therefore, we set $\bar{\lambda} = \delta_U$ precisely when δ_U is close to δ_1. Such choice of $\bar{\lambda}$ will produce a value of α close to the value $\tilde{\alpha}_1$ from Lemma 5.2, which is the optimal value in potential hard cases. In other words, in case we need to use the safeguarding strategy we will obtain a value of the parameter that is close to the optimal.

Safeguarding α

To assure global convergence of the iteration we must safeguard the parameter $\alpha_{(k+1)}$, after Step 3.4 of Algorithm 5.4.1 or during the computation of α by (5.21) in Step 3.4. As in [68] and [67], we maintain a safeguarding interval $[\alpha_L, \alpha_U]$ containing the value α_*. If $\alpha_{(k+1)} \notin [\alpha_L, \alpha_U]$, we attempt a linear adjustment of the parameter and if this value is still not contained in the safeguarding interval we take the midpoint of the interval as the new value of the parameter. We present the procedure in Figure 5.4.

```
Step 3.5 Safeguarding of $\alpha$
    if ($\alpha_{(k+1)} \notin [\alpha_L, \alpha_U]$) then
        if ($k = 0$ or $\|x_{(k)}\| < \|x_{(k-1)}\|$) then
            $\alpha_{(k+1)} = \delta_U + \phi_k + \phi'_k(\delta_U - \lambda_{(k)})$
        else
            $\alpha_{(k+1)} = \delta_U + \phi_{k-1} + \phi'_{k-1}(\delta_U - \lambda_{(k-1)})$
        end if
        if ($\alpha_{(k+1)} \notin [\alpha_L, \alpha_U]$) then
            $\alpha_{(k+1)} = \dfrac{\alpha_L + \alpha_U}{2}$
        end if
    end if
```

Figure 5.4 Procedure for the Safeguarding of the Parameter α.

where, as in §5.5.3, we define $x_{(i)} = \dfrac{u_{(i)}}{\nu_{(i)}}$, $\phi_i = \phi(\lambda_{(i)}) = -g^T x_{(i)}$ and $\phi'_i = \phi'(\lambda_{(i)}) = x_{(i)}^T x_{(i)}$, for $i = k-1,\ k$.

The initial values we use for α_L and α_U are based on the bounds introduced in [66], but we wish to present a more straightforward derivation here.

First we show that $\alpha_* \leq \delta_U + \|g\|\Delta$.

Let $\{\lambda_*, x_*\}$ be the optimal pair from Lemma 4.1, except when there is only an interior solution in which case we define $x_* = -(H - \lambda_* I)^\dagger g$ such that $\|x_*\| = \Delta$. We define α_* as

$$\alpha_* = \lambda_* - g^T x_*,$$

therefore

$$\alpha_* - \lambda_* \leq \mid \alpha_* - \lambda_* \mid \leq \|g\|\|x_*\|$$

and $\|x_*\| \leq \Delta$ implies

$$\alpha_* \leq \lambda_* \ + \ \|g\|\Delta.$$

Since $\lambda_* \leq \delta_1 \leq \delta_U$, we conclude

$$\alpha_* \leq \delta_U \ + \ \|g\|\Delta.$$

Now we prove that $\delta_1 - \dfrac{\|g\|}{\Delta} \leq \alpha_*$.

For an optimal pair $\{\lambda_*, x_*\}$ defined as above, we have that

$$\begin{aligned} \Delta^2 &= \phi'(\lambda_*) \\ &= \sum_{i=1}^{n} \frac{{\gamma_i}^2}{(\delta_i - \lambda_*)^2} \end{aligned}$$

and therefore

$$\begin{aligned} \Delta^2 &\leq \sum_{i=1}^{n} \frac{{\gamma_i}^2}{(\delta_1 - \lambda_*)^2} \\ &= \frac{\|g\|^2}{(\delta_1 - \lambda_*)^2} \end{aligned}$$

which implies

$$\delta_1 - \lambda_* \leq \frac{\|g\|}{\Delta}$$
$$\delta_1 - \frac{\|g\|}{\Delta} \leq \lambda_*$$

and since $\lambda_* \leq \alpha_*$ by Rayleigh quotient properties, we can now claim

$$\delta_1 - \frac{\|g\|}{\Delta} \ \leq \ \alpha_*. \tag{5.22}$$

In practice, the computation of δ_1 can be expensive but we can avoid it by observing that (5.22) still holds if we replace δ_1 with δ_L such that $\delta_L \leq \delta_1$.

An initial lower bound for δ_1 is available after Step 1.4 in Algorithm 5.4.1 since λ_1, the smallest eigenvalue of $B_{\alpha_{(0)}}$, satisfies $\lambda_1 \leq \delta_1$. Therefore we can compute an initial value for α_L as

$$\alpha_L = \lambda_1 - \frac{\|g\|}{\Delta}.$$

In Figure 5.5 we present the procedure for updating the safeguards corresponding to Step 3.8 of Algorithm 5.4.1.

```
Step 3.8  Update of Safeguards.
  if ($\nu_{(k)} = \nu_1$ and $u_{(k)} = u_1$) then
     if ($\|x_{(k)}\| > \Delta$) then
          $\alpha_U = \alpha$
     else
          if ($\|x_{(k)}\| < \Delta$) then
              $\alpha_L = \alpha$
          end if
     end if
  else
       $\alpha_U = \alpha$
  end if
```

Figure 5.5 Procedure for the Update of the Safeguards.

To see that with this procedure for updating the safeguarding interval the value α_* remains in $[\alpha_L, \alpha_U]$, first note that if we are using the pair $\{\lambda_1,\ (\nu_1, u_1^T)^T\}$ to compute $\{\lambda_{(k)}, x_{(k)}\}$, then $\lambda_{(k)} \leq \delta_1$. Now recall that the function $\phi(\lambda)$ is strictly increasing in either $(-\infty, \delta_1)$ in the easy case, or $(-\infty, \delta_1]$ in a potential hard case; and also that $\phi'(\lambda_{(k)}) = \|x_{(k)}\|^2$. Therefore, if $\|x_{(k)}\| < \Delta$ then $\alpha_{(k)} < \alpha_*$ and if $\|x_{(k)}\| > \Delta$ then $\alpha_{(k)} > \alpha_*$.

Moreover, if we are using the pair $\{\lambda_2,\ (\nu_2, u_2^T)^T\}$ to compute $\{\lambda_{(k)}, x_{(k)}\}$, then the value $\alpha_{(k)}$ that yields such eigenpair is an upper bound for α_*, since $\lambda_2 > \delta_1$.

5.5.5 Adjustment of α

As we saw in §5.5.2, at iteration k the values $\{\lambda_{(k)},\ (\nu_{(k)}\ u_{(k)}^T)^T\}$ are chosen as an eigenpair associated with the smallest eigenvalue of $B_{\alpha_{(k)}}$ or with a value equal or close to the second smallest eigenvalue of this matrix. The choice between the two eigenpairs is based on the magnitude of the first component of each eigenvector and therefore we must guarantee that one of the two eigenvectors will have a sufficiently large first component. The following procedure shows our approach to adjust α to satisfy this requirement. The procedure follows the one in [67] and we present it in Figure 5.6.

```
Step 3.2 Adjustment of $\alpha$.
    while ($\nu_1$ small or $\nu_2$ small) do
      1. $\alpha_U = \alpha_{(k)}$
      2. $\alpha = \frac{\alpha_L + \alpha_U}{2}$
      3. Compute $\{\lambda_1,\ (\nu_1, u_1^T)^T\}$, $\{\lambda_2,\ (\nu_2, u_2^T)^T\}$ two
         eigenpairs of $B_{\alpha_{(k)}}$, with $\lambda_1$ the smallest eigenvalue and
         $\lambda_2$ equal or close to the second smallest eigenvalue
    end while
```

Figure 5.6 Procedure for the Adjustment of the Parameter α.

We consider ν_1 or ν_2 small according to the criterion we use in the selection of the interpolation values (see §5.5.2).

Note that Theorem 5.1 implies that the only case in which an eigenvector corresponding to the smallest eigenvalue of B_α has small first component is in a potential hard case when $\alpha > \alpha_*$. Therefore the interval $[\alpha_L, \alpha_U]$ after Step 1 in the while loop will still contain α_*.

After Step 2, either $\alpha_L \leq \alpha < \alpha_*$ or $\alpha_U \geq \alpha > \alpha_*$. If $\alpha_L \leq \alpha < \alpha_*$ then Theorem 5.1 implies that ν_1 will be large. If we always get $\alpha_U \geq \alpha > \alpha_*$ then, since we reduce the interval $[\alpha_L, \alpha_U]$ at each iteration, we have that eventually α will get close enough to α_* (from the right) and Theorem 5.1 implies that ν_2 will be large.

In practice we must also check if the minimum length for the safeguarding interval is reached in this process. In this case, the algorithm finishes and a solution is computed as indicated in §5.5.6.

5.5.6 Stopping Criteria

Given the values $\{\lambda_{(k)},\ (\nu_{(k)}\ u_{(k)}^T)^T\}$ we check for a boundary solution, an interior solution or a quasi–optimal solution according to Theorem 5.2. We can also stop the iteration if a maximum number of iterations has been reached or if the length of the safeguarding interval is too small.

In the tests, we check with $\nu_{(k)}$ and $u_{(k)}$ instead of $x_{(k)} = \dfrac{u_{(k)}}{\nu_{(k)}}$ because at that point in the algorithm we do not know if this division is well defined, i.e. $\nu_{(k)}$ could be zero. In practice, $\nu_{(k)}$ could be very small and dividing by such value will produce large errors.

(1) Boundary Solution.

A boundary solution is found if

$$(|\ \|u_{(k)}\| - \Delta|\nu_{(k)}|\ |\ \leq\ \varepsilon_\Delta * \Delta|\nu_{(k)}|) \quad \text{and} \quad (\lambda_{(k)} \leq 0).$$

If this condition is satisfied the solution is

$$\lambda_* = \lambda_{(k)} \quad \text{and} \quad x_* = \frac{u_{(k)}}{\nu_{(k)}}.$$

(2) Interior Solution.

Let $\varepsilon_{Int} \in (0,1)$, then an interior solution is detected if

$$(\|u_{(k)}\| < \Delta|\nu_{(k)}|) \quad \text{and} \quad (\lambda_{(k)} > -\varepsilon_{Int}) \quad \text{and} \quad (H \quad \text{is positive definite}).$$

Where H is declared to be positive definite if *any* of the $\lambda_{(i)}$ for $i = 1, 2, \ldots, k$, satisfies $\lambda_{(i)} > -\varepsilon_0$, since this implies that $\delta_1 \geq \lambda_{(i)} > -\varepsilon_0$. We emphasize that we check the condition on $\lambda_{(i)}$ at a point in the algorithm where this value corresponds to the smallest eigenvalue of B_{α_i}.

We know from Chapter 4 that an interior solution is given by $x_{INT} = -H^{-1}g$ with corresponding Lagrange multiplier $\lambda_{INT} = 0$. We recall from Section 5.4 that for the purpose of computing the safeguarding interval in the interior solution case, we redefine x_* as $x_* = -(H - \lambda_* I)^{\dagger} g$ such that $\|x_*\| = \Delta$, and therefore, $\alpha_* = \lambda_* - g^T x_*$.

We can guarantee that we detect the existence of an interior solution since $\alpha_* > -g^T x_{INT}$, and the safeguarding interval is reduced in a way that guarantees that we eventually compute λ_1 positive with $\|(H - \lambda_1 I)^{\dagger} g\| < \Delta$.

An interior solution satisfies the linear system $Hx_{INT} = -g$. Therefore when we detect the existence of an interior solution we should, in principle, solve such system. The Conjugate Gradient Method (see for example [25, Ch. 10]) is a natural choice for solving this system in the large–scale case. However, for discrete ill–posed problems this approach amounts to applying the Conjugate Gradient Method to the normal equations with a very ill–conditioned matrix. For the reasons exposed in §3.4.2 we want to avoid this computation.

We use a different strategy that consists of checking if $\lambda_{(k)}$, which is the smallest eigenvalue of B_α, is sufficiently small. If so, we set $\lambda_* = 0$ and $x_* = \frac{u_{(k)}}{\nu_{(k)}}$. Note that $x_* = -(H - \lambda_{(k)} I)^{-1} g$ and for $\lambda_{(k)}$ close to zero such x_* will be a good approximation to the unconstrained minimizer given by $-H^{-1}g$. However, if there is a high level of noise in the data, this unconstrained minimizer is the least squares, unregularized solution of the perturbed problem and it is contaminated by the noise in the data This solution will not be a good approximation to the desired solution of the original problem (see §3.1.2).

If $\lambda_{(k)}$ is not small enough, we reduce the radius Δ and start the algorithm again. This approach is different from the one in [71], [68] and [67] where the interior solution is computed by solving the system $Hx = -g$.

(3) Quasi–optimal Solution.

To declare that a quasi–optimal solution has been found, we first compute $\hat{\lambda}$ and $\hat{x}$ as in Lemma 5.4, provided that the conditions of the lemma are satisfied.

If $\hat{\lambda}$ and $\hat{x}$ satisfy condition *(ii)* of Theorem 5.2, we declare that a quasi–optimal solution has been found and set $\lambda_* = \hat{\lambda}$ and $x_* = \hat{x}$.

(4) The interval $[\alpha_L, \alpha_U]$ is too small.

If $|\alpha_U - \alpha_L| \leq \varepsilon_\alpha \max\{|\alpha_L|, |\alpha_U|\}$ then we stop the iteration and set $\lambda_* = \lambda_{(k)}$. If $\nu_{(k)}$ is too small or zero, we cannot compute a solution. This situation can arise in practice because the eigensolver might not provide the eigenvector with significant first component that the theory guarantees. We have not encountered this case in our experiments.

If $\nu_{(k)}$ is large enough, we set $p = \dfrac{u_{(k)}}{\nu_{(k)}}$ and if $|\|p\| - \Delta| > \varepsilon_\Delta * \Delta$ then we set $x_* = p$.

If $\|p\| < \Delta$ then we are in the hard case and α_* is within ε_α of $\tilde{\alpha}_1$. In this case we compute x_* as $x_* = p + \tau u_1$ such that $\|x_*\| = \Delta$. The vector u_1 is an approximate eigenvector associated with the smallest eigenvalue of H. This vector is available in potential hard cases, after Step 3.3 of the algorithm.

Of the two possible choices for τ, we choose the one with smaller magnitude since this value minimizes $\psi(p+\tau u_1)$ (see [55, p. 558]). This choice of τ is given by

$$\tau = \frac{\Delta^2 - \|p\|^2}{p^T u_1 + sign(p^T u_1)\sqrt{(p^T u_1)^2 - (\Delta^2 - \|p\|^2)}} .$$

5.6 Convergence Properties

In this section we present the convergence results for Algorithm 5.4.1. In §5.6.1 we prove that the iterates generated by Algorithm 5.4.1 are well defined, in §5.6.2 we present the local convergence theory and in §5.6.3 the global convergence theory of the method.

5.6.1 The Iterates are Well Defined

> **Lemma 5.6 ([67])** The iterates generated by Algorithm 5.4.1 are well defined.

Proof In order to define the current iterate $x_{(k)}$ in Algorithm 5.4.1, we must ensure that we can safely normalize an eigenvector corresponding to either the smallest eigenvalue or a value equal or close to the second smallest eigenvalue of $B_{\alpha_{(k)}}$, to have first component one. This is accomplished in Step 3.2 where we adjust the parameter $\alpha_{(k)}$ until one of these two eigenvectors can be normalized to have first component one.

We described Step 3.2 and discussed its convergence properties in §5.5.5. Theorem 5.1 guarantees that the adjusting procedure in Step 3.2 yields a value of α such that for the smallest eigenvalue or a value equal or close to the second smallest eigenvalue of B_α, there exists a corresponding eigenvector with significant first component.

□

5.6.2 Local Convergence

We established in the previous section that the iterates generated by Algorithm 5.4.1 are well defined. In this section we present the local convergence properties of the algorithm.

We will use Lemmas 5.7 and 5.8, in the proof of Theorem 5.3 which contains the local convergence result.

> **Lemma 5.7 ([67])** Let $\lambda_* \leq \delta_1$ be the Lagrange multiplier corresponding to a boundary solution of problem (5.1). Let $\lambda_{(i+1)}$ be the *(i+1)*th iterate computed by Algorithm 5.4.1 using the two–point interpolating scheme given by (5.21) to update α. Then, there exists a neighborhood $\mathcal{B}$ of λ_*, $\mathcal{B} = \{\lambda \mid |\lambda - \lambda_*| < r\}$, such that if $\lambda_{(i-1)},\ \lambda_{(i)} \in \mathcal{B}$ then $\lambda_{(i+1)}$ satisfies
>
> $$\lambda_{(i+1)} - \lambda_* = (\lambda_{(i-1)} - \lambda_*)(\lambda_{(i)} - \lambda_*)\mathcal{O}(1). \tag{5.23}$$

Proof See [67]. □

> **Lemma 5.8** Let $\lambda_* \leq \delta_1$ be the Lagrange multiplier corresponding to a boundary solution of problem (5.1). Let $\{\lambda_{(k)}\}$ be the sequence of iterates generated by Algorithm 5.4.1 using the two–point interpolating scheme given by (5.21) to update α. Then
>
> $$g \in \mathcal{R}(H - \lambda_{(k)}I) \quad \text{and} \quad g \in \mathcal{R}(H - \lambda_* I).$$

Proof First we prove that $g \in \mathcal{R}(H - \lambda_{(k)}I)$. We consider two cases.

Case 1: $g \not\perp \mathcal{S}_i$, $1 \leq i \leq n$. Then $\forall k$, $\lambda_{(k)} < \delta_1$, $H - \lambda_{(k)}I$ is nonsingular and $g \in \mathcal{R}(H - \lambda_{(k)}I)$.

Case 2: $g \perp \mathcal{S}_i$, $i = 1, 2, \ldots, \ell$, $1 \leq \ell < n$. Then, if $\lambda_{(k)} \neq \delta_i$, $i = 1, 2, \ldots, \ell$ then $H - \lambda_{(k)} I$ is nonsingular and $g \in \mathcal{R}(H - \lambda_{(k)} I)$. If $\lambda_{(k)} = \delta_i$, for some i such that $1 \leq i \leq \ell$, then since $g \perp \mathcal{N}(H - \delta_i I)$ we have that $g \in \mathcal{R}(H - \lambda_{(k)} I)$.

To prove that $g \in \mathcal{R}(H - \lambda_* I)$ observe that $\lambda_* \leq \delta_1$. If $\lambda_* < \delta_1$ then $H - \lambda_* I$ is nonsingular and therefore $g \in \mathcal{R}(H - \lambda_* I)$. If $\lambda_* = \delta_1$ then $g \perp \mathcal{N}(H - \lambda_* I)$ must hold and hence $g \in \mathcal{R}(H - \lambda_* I)$. □

Observe that under the conditions of Lemma 5.8 and because H is symmetric, we have

$$\begin{aligned} (H - \lambda_{(k)} I)(H - \lambda_{(k)} I)^\dagger g &= (H - \lambda_{(k)} I)^\dagger (H - \lambda_{(k)} I) g = g \qquad (5.24) \\ \text{and} \quad (H - \lambda_* I)(H - \lambda_* I)^\dagger g &= (H - \lambda_* I)^\dagger (H - \lambda_* I) g = g, \qquad (5.25) \end{aligned}$$

since $(H - \lambda_{(k)} I)(H - \lambda_{(k)} I)^\dagger$ and $(H - \lambda_{(k)} I)^\dagger (H - \lambda_{(k)} I)$ are orthogonal projections onto $\mathcal{R}(H - \lambda_{(k)} I)$, and $(H - \lambda_* I)(H - \lambda_* I)^\dagger$ and $(H - \lambda_* I)^\dagger (H - \lambda_* I)$ are orthogonal projections onto $\mathcal{R}(H - \lambda_* I)$.

Theorem 5.3 ([67]) Let $\lambda_* \leq \delta_1$ be the Lagrange multiplier corresponding to a boundary solution of problem (5.1). Let $\{\lambda_{(k)}\}$ be the sequence of iterates generated by Algorithm 5.4.1 using the two–point interpolating scheme given by (5.21) to update α. There exists a neighborhood $\mathcal{B}$ of λ_* such that if $\lambda_{(i-1)}$, $\lambda_{(i)} \in \mathcal{B}$ then $\{\lambda_{(k)}\}$, $k \geq i - 1$, remains in $\mathcal{B}$ and converges superlinearly to λ_*. Moreover, if $x_{(k)} = -(H - \lambda_{(k)} I)^\dagger g$ then the sequence $\{x_{(k)}\}$, beginning with $x_{(i-1)}$, $x_{(i)}$, converges superlinearly to x_* with $\|x_*\| = \Delta$ if $\lambda_* < \delta_1$, or to $p = -(H - \delta_1 I)^\dagger g$ with $\|p\| \leq \Delta$ for $\lambda_* = \delta_1$.

Proof First we show that $\{\lambda_{(k)}\}$ converges to λ_* and that the rate of convergence is superlinear.

Let $\rho_i = \lambda_{(i)} - \lambda_*$. Let $\mathcal{B}_r = \{\lambda \mid |\lambda - \lambda_*| < r\}$ be the neighborhood of Lemma 5.7 and suppose that $\lambda_{(i-1)},\ \lambda_{(i)} \in \mathcal{B}_r$. Then, (5.23) holds and implies that there exists a constant $c > 0$ such that

$$|\rho_{i+1}| \leq c\ |\rho_i|\ |\rho_{i-1}|. \tag{5.26}$$

Let $\hat{r} = \min\{r, \frac{1}{2c}\}$, define $\mathcal{B}_{\hat{r}} = \{\lambda \mid |\lambda - \lambda_*| < \hat{r}\}$ and observe that $\mathcal{B}_{\hat{r}} \subset \mathcal{B}_r$. If $\lambda_{(i-1)},\ \lambda_{(i)} \in \mathcal{B}_{\hat{r}}$ we have that (5.26) holds and since $c\ |\rho_{i-1}| \leq \frac{1}{2}$ then $|\rho_{i+1}| \leq \frac{1}{2}|\rho_i|$. Therefore $\lambda_{(k)} \in \mathcal{B}_r$ for $k \geq i-1$, which implies that $|\rho_{k+1}| \leq c\ |\rho_k|\ |\rho_{k-1}|$, for $k \geq i$ with $|\rho_{k+1}| \leq \frac{1}{2}|\rho_k|$. Therefore ρ_k goes to zero as k goes to infinity. We can conclude now that $\{\lambda_{(k)}\}$ remains in $B_{\hat{r}}$ and converges to λ_*.

To see that the convergence is superlinear, observe that by (5.23) for $k \geq i$ we have

$$\frac{|\lambda_{(k+1)} - \lambda_*|}{|\lambda_{(k)} - \lambda_*|} = |\lambda_{(k-1)} - \lambda_*|\mathcal{O}(1),$$

which goes to zero as k goes to infinity.

In the second part of the proof we show that the sequence $\{x_{(k)}\}$ converges to the vector $y = -(H - \lambda_* I)^\dagger g$ such that if $\|y\| = \Delta$ and $\lambda_* < \delta_1$ then $y = x_*$, and if $\|y\| \leq \Delta$ and $\lambda_* = \delta_1$ then $y = p$. We then show that the rate of convergence is superlinear.

Let us study $x_{(k)} - y$ which is given by

$$x_{(k)} - y = (H - \lambda_* I)^\dagger g - (H - \lambda_{(k)} I)^\dagger g. \tag{5.27}$$

Observe that by (5.24) we can write

$$(H - \lambda_* I)^\dagger g = (H - \lambda_* I)^\dagger (H - \lambda_{(k)} I)(H - \lambda_{(k)} I)^\dagger g, \tag{5.28}$$

and by (5.25)

$$(H - \lambda_{(k)} I)^\dagger g = (H - \lambda_{(k)} I)^\dagger (H - \lambda_* I)^\dagger (H - \lambda_* I) g \tag{5.29}$$

which is equivalent to

$$(H - \lambda_{(k)} I)^{\dagger} g = (H - \lambda_* I)^{\dagger} (H - \lambda_* I)(H - \lambda_{(k)} I)^{\dagger} g \tag{5.30}$$

which follows after substituting the SVD of the matrices in (5.29) and changing the order of the diagonal matrices of the SVD's.

Substituting (5.28) and (5.30) into (5.27), we obtain

$$\begin{aligned} x_{(k)} - y &= (H - \lambda_* I)^{\dagger} ((H - \lambda_{(k)} I) - (H - \lambda_* I))(H - \lambda_{(k)} I)^{\dagger} g \\ &= -(H - \lambda_* I)^{\dagger} (\lambda_{(k)} - \lambda_*)(H - \lambda_{(k)} I)^{\dagger} g. \end{aligned}$$

Taking norms on both sides we have

$$\|x_{(k)} - y\| = |\lambda_{(k)} - \lambda_*| \; \|(H - \lambda_* I)^{\dagger} (H - \lambda_{(k)} I)^{\dagger} g\|. \tag{5.31}$$

Since $\lambda_{(k)} \rightarrow \lambda_*$, we have that $x_{(k)} \rightarrow x_*$ if $\lambda_* < \delta_1$ and $\|y\| = \Delta$, or $x_{(k)} \rightarrow p$ if $\lambda_* = \delta_1$ and $\|y\| \leq \Delta$.

To see the superlinear convergence observe that (5.23) and (5.31) imply that

$$\frac{\|x_{(k+1)} - y\|}{\|x_{(k)} - y\|} = |\lambda_{(k-1)} - \lambda_*| \mathcal{O}(1).$$

This completes the proof. □

5.6.3 Global Convergence

Theorem 5.4 ([67]) Algorithm 5.4.1 is globally convergent.

Proof The goal of Algorithm 5.4.1 is to solve the trust–region subproblem by either determining the existence of an interior solution, or by computing an optimal value α_* for the parameter α, such that the solution to the parameterized eigenvalue problem for B_{α_*} can be used to compute a boundary solution for the trust–region subproblem.

The global convergence of Algorithm 5.4.1 is achieved by means of a safeguarding interval that contains the optimal parameter α_*.

We first recall that the initial safeguarding interval $[\alpha_L, \alpha_U]$ contains α_* (see §5.5.4). Starting with that interval, the updating procedure for α_L and α_U, either in Step 3.2 (see §5.5.5) or in Step 3.8 (see §5.5.4), guarantees that α_* remains in the interval and that the safeguarding interval is reduced at each iteration.

Therefore, since $\alpha = \lambda - g^T x$, after a finite number of steps either the sequence $\{\lambda_{(k)}\}$ reaches the neighborhood of λ_* of Theorem 5.3 that guarantees convergence, or the length of the safeguarding interval $|\alpha_U - \alpha_L|$ goes to zero with $\alpha_L \leq \alpha_* \leq \alpha_U$. □

5.7 Numerical Results

In this section we present the numerical results obtained when we apply Algorithm 5.4.1 to the problem

$$\begin{array}{ll} \min & \frac{1}{2}x^T H x + g^T x \\ s.t. & \|x\| \leq \Delta \end{array}$$

with $H = A^T A$ and $g = -A^T \bar{b}$.

The test problems are from the Regularization Tools package of Hansen [36].

Our program is a Matlab implementation of Algorithm 5.4.1. The eigenproblems are solved with the ARPACK version ([43]) of the Implicitly Restarted Lanczos Method (IRLM) of Sorensen [70]. Our Matlab code uses the IRLM by means of a Mexfile interface with the ARPACK library.

We carried out all the experiments on a Sun Ultrasparc 2. The floating point arithmetic is IEEE standard double precision with machine precision of $\epsilon_M \equiv 2^{-52} \approx 2.2204 \cdot 10^{-16}$.

The required accuracy for the eigenpairs was 10^{-2}, but the IRLM usually computed the eigenpairs at a higher accuracy. The initial vector for the Lanczos factorization is a randomly generated vector that remains fixed in all the experiments. In §5.7.1 and §5.7.2, the number of basis vectors for the IRLM was limited to nine. Seven shifts (i.e. seven matrix–vector products) were applied on each implicit restart.

For the experiments in §5.7.3, the number of basis vectors and the number of shifts varied for three different sets of problems.

We apply a Tchebyshev polynomial filter in order to recover very small eigenvalues. We use a polynomial of degree ten, which implies that each matrix–vector product in the IRLM amounts to nine matrix–vector products with the original matrix. In order to use a Tchebyshev polynomial we need an estimate of the largest eigenvalue of the bordered matrix. Since for discrete ill–posed problems, the largest eigenvalue is well separated from the rest, we expect the IRLM to be very efficient in computing such value. We observed that even for large problems it usually takes only one iteration of the IRLM to compute that eigenvalue.

An alternative to the use of a Tchebyshev polynomial filter is the use of Leja points as shifts, since they were reported in [2] to be very efficient for recovering very small eigenvalues. We did not try this option.

Table 5.2 contains the values of tolerances used in all the experiments.

The meaning of the tolerances in Table 5.2 is the following:

- ε_Δ determines the accuracy of a boundary solution (see §5.5.6).
- $-\varepsilon_{Int}$ is a lower bound on an acceptable value of λ for an interior solution (see §5.5.6).
- ε_{HC} determines the accuracy of a quasi–optimal solution (see §5.5.6).
- $-\varepsilon_0$ is used as threshold to determine that a value is positive (see §5.5.6).

Tolerance	Value
ε_{Δ}	10^{-4}
ε_{Int}	10^{-8}
ε_{HC}	10^{-8}
ε_{0}	10^{-10}
ε_{α}	10^{-8}
ε_{ν}	10^{-2}

Table 5.2 Values of tolerances for experiments with the Trust–Region Subproblem Method.

- ε_α determines the minimum allowed length of the safeguarding interval $[\alpha_L, \alpha_U]$ (see §5.5.6).
- ε_ν is used to determine when the first component of an eigenvector of B_α is small (see §5.5.2).

The purpose of this section is to illustrate some of the properties of the Trust–Region Subproblem Method described in the previous sections. We show superlinear convergence in §5.7.1, the computation of interior solutions in §5.7.2 and finally the results for all the problems in the Regularization Tools package in §5.7.3.

In all cases, x_{IP} denotes a discretization of the exact solution of the inverse problem.

5.7.1 Superlinear Convergence

In Table 5.3 we show the superlinear convergence behavior of the method when it is applied to problem **phillips** from [36].

Problem **phillips** is a discretization of a classic test problem presented by D.L. Phillips in [61], consisting of a Fredholm integral equation of the first kind. The kernel

Iter k	$\frac{\|\Delta - \|x_{(k)}\|\|}{\Delta}$
1	1.8188e-02
2	4.0480e-03
3	1.1766e-03
4	2.3746e-04
5	2.9231e-06

Table 5.3 Superlinear Convergence.

K, the solution f and the right–hand side g are given by

$$\begin{aligned} K(s,t) &= \varphi(s-t) \\ f(t) &= \varphi(t) \\ g(s) &= (6-|s|)\left(1+\tfrac{1}{2}\cos\left(\frac{\pi s}{3}\right)+\frac{9}{2\pi}\sin\left(\frac{\pi s}{3}\right)\right) \end{aligned}$$

where the function $\varphi(x)$ is defined as

$$\varphi(x) = \begin{cases} 1+\cos\left(\frac{\pi x}{3}\right), & |x| < 3 \\ 0 & |x| \geq 3 \end{cases}$$

and the integration intervals are $[-6, 6]$. Problem **phillips** from [36] is a discretization of this continuous problem.

We chose the dimension of the problem as $n = 300$ and therefore the discretized operator A is a 300×300 matrix. The right–hand side is $\bar{b} = b + \epsilon r$, where b is a discretization of the right hand side g, $\epsilon = 0.01$ and r is random vector with components uniformly distributed in $[0, 1]$. The noise level is $\epsilon\|r\| = 0.0994$.

We chose the trust–region radius as $\Delta = \|x_{IP}\| = 2.9999$. We obtained a relative error in the approximation of order 10^{-2}. In Figure 5.7 we show x_{IP} and x, the solution of the trust–region subproblem computed with our Matlab implementation of Algorithm 5.4.1.

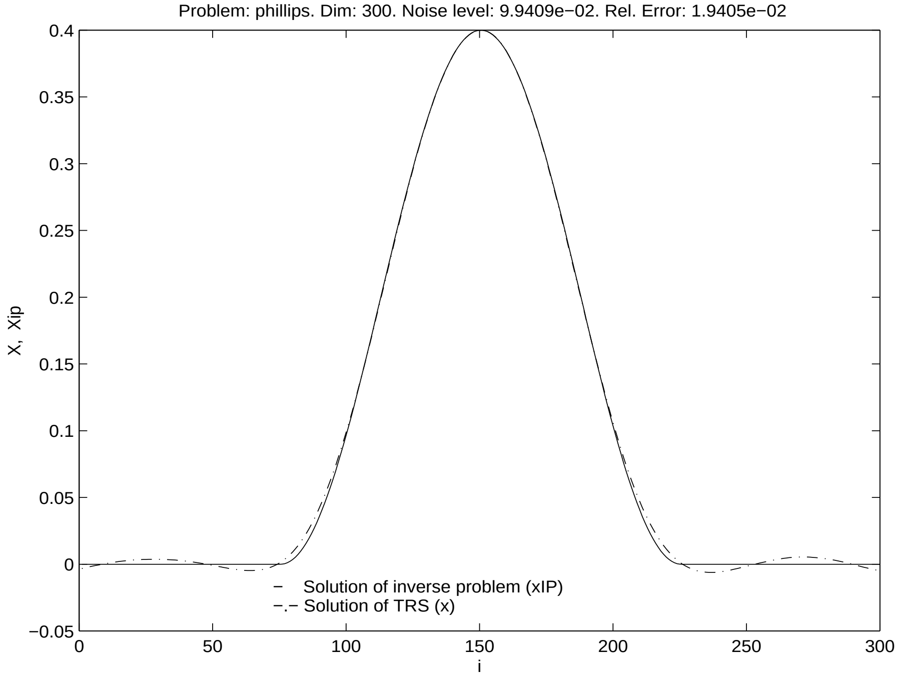

Figure 5.7 Boundary Solution for problem **phillips**.

5.7.2 Interior Solutions

As we discussed in §5.5.6, when there is noise in the right–hand side, an interior solution corresponds to an unregularized solution which might differ considerably from the desired solution (see §3.1.2).

The situation is different for the exact data case where there is no noise in the right–hand side. In this case, it might be better to overestimate the value of the trust–region radius since an interior solution will usually be detected for a small value of λ, especially for severely ill–posed problems where the smallest eigenvalue of H is very close to zero. As we discussed in §5.5.6, a small value of λ implies that $-(H - \lambda I)^{\dagger}g$ is a good approximation to the interior solution.

We have chosen problem **foxgood** from [36] to illustrate this situation. The problem comes from the discretization of a Fredholm equation of the first kind with both integration intervals equal to $[0,1]$, with kernel K and right–hand side g given by

$$K(s,t)=\sqrt{s^2+t^2}, \quad g(s)=\frac{1}{3}((1+s^2)^{\frac{3}{2}}-s^3)$$

and with solution $f=t$. This is an artificial discrete severely ill–posed problem which does not satisfy the Discrete Picard Condition (see 3.1.2).

For $n=300$, we have that $\|x_{IP}\|=10$. If we choose $\Delta=20$, for example, and work with exact data, then our method finds an interior solution with an accuracy of 10^{-2}. We show x_{IP} and x, the computed solution of the trust–region subproblem in Figure 5.8.

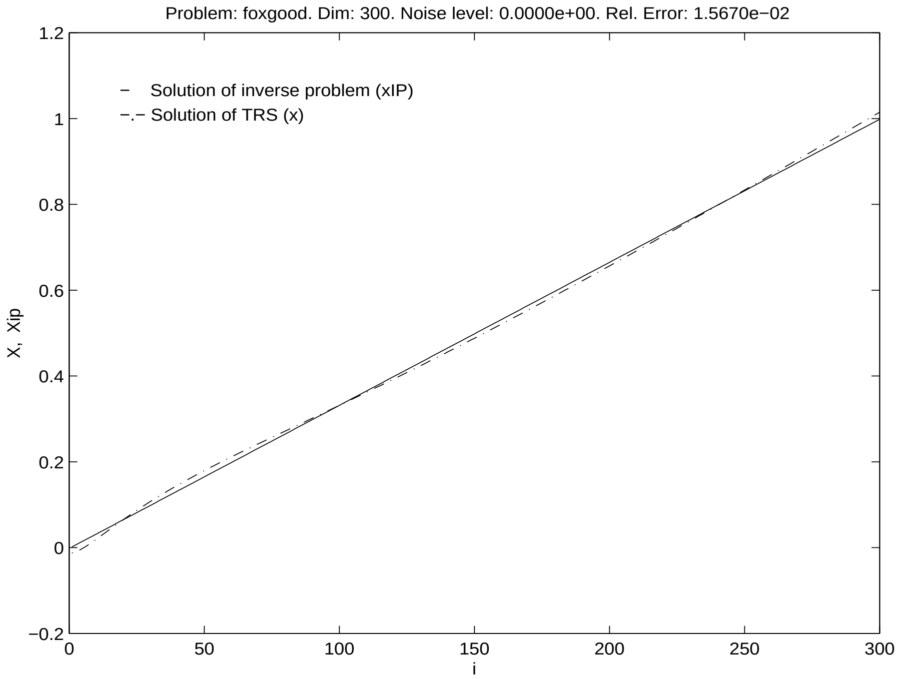

Figure 5.8 Interior solution for problem **foxgood** with *exact* data.

5.7.3 Results for Test Problems

In Tables 5.4, 5.5 and 5.6 we show the behavior of our method on all the problems from [36]. We report the number of matrix–vector products and the number of iterations used by the TRS method to compute a boundary or quasi–optimal solution. When the solution to the original problem was available we used its norm as trust–region radius.

Two important observations can be made from Tables 5.4, 5.5 and 5.6. The first one is that we solve the trust–region subproblem in all cases. The second one is that even though the computed solution is optimal or quasi–optimal and $\Delta = \|x_{IP}\|$, sometimes this solution is not a very good approximation to the exact solution, as we can observe in problems **baart** and **deriv2**. In these cases, the difficulty might come from the fact that we are not using the appropriate constraint. As we mentioned in Section 3.2, for some problems the constraint should be of the form $\|Lx\| \leq \Delta$ where L is a discrete derivative operator. This type of constraint controls the smoothness of the solution rather than the norm of the solution itself. An example of this situation is the application we present in Chapter 6. The trust–region subproblem should be formulated according to the features of each class of problem.

Another observation is that the number of matrix–vector products might be large with respect to the dimension of the problem. We want to point out that this number does not seem to increase as the dimension increases and in some cases it actually decreases. That this is the case can be observed in the three problems of dimension 1000 in Table 5.5, namely **heat**, **phillips** and **shaw**. We observe that the number of matrix–vector products is of the same order as for dimension 300, and for problem **heat** it actually decreases by a factor of two.

In Chapter 6 we present an example of high dimension where the number of matrix–vector products is low. In Tables 5.4, 5.5 and 5.6 BS stands for Boundary Solution, QO for Quasi–optimal Solution, IS for Interior Solution, HC for Hard Case.

In the first set of problems the number of basis vectors was limited to five with three shifts on each implicit restart. We show the results in Table 5.4.

Problem	Dim.	Δ	$\|\|x\|\|$	Exit by	$\frac{\|\|x-x_{IP}\|\|}{\|\|x_{IP}\|\|}$	MV Prods.	Iter.
baart	300	1.2533	1.2533	BS,QO	1.7723e-01	491	9
deriv2	300	0.5773	0.5773	HC	1.8506e+00	1181	13
foxgood	300	10.0000	9.9999	BS	4.3303e-02	389	7
spikes	300	31.9687	18.9296	IS	8.1813e-01	447	7
ursell	300	10.0000	9.9999	BS	–	589	7
wing	300	0.5774	0.5774	BS	6.8749e-01	524	10

Table 5.4 Results of the Trust–Region Subproblem Method for test problems from the Regularization Tools package. First set of problems.

For the second set of problems the number of basis vectors was limited to nine with seven shifts on each implicit restart. Table 5.5 contains these results.

Problem	Dim.	Δ	$\|\|x\|\|$	Exit by	$\frac{\|\|x-x_{IP}\|\|}{\|\|x_{IP}\|\|}$	MV Prods.	Iter.
Ill **heat**	300	4.2631	4.2631	QO	3.7838e-01	2479	15
Ill **heat**	1000	7.7829	7.7829	QO	6.4598e-01	1480	9
Well **heat**	300	4.2631	4.2631	BS,QO	7.1192e-02	1933	6
ilaplace	195	2.7629	2.7631	BS,QO	1.6841e-01	1192	12
parallax	300	5.0000	5.0000	QO	–	958	10
phillips	300	2.9999	2.9999	BS	1.9405e-02	697	7
phillips	1000	3.0000	3.0000	BS,QO	2.6030e-02	751	7
shaw	300	17.2893	17.2892	BS,QO	5.4469e-02	859	9
shaw	1000	31.5659	31.5665	BS	5.3534e-02	859	9

Table 5.5 Results of the Trust–Region Subproblem Method for test problems from the Regularization Tools package. Second set of problems.

Problem **spikes** required the use of seventeen Lanczos vectors (fifteen shifts per restart) in order to be solved. Table 5.6 shows the result.

Problem	Dim.	$\Delta = \|\|x_{IP}\|\|$	$\|\|x\|\|$	Exit	$\frac{\|\|x-x_{IP}\|\|}{\|\|x_{IP}\|\|}$	M–V	Iter.
spikes	300	31.9687	31.9687	QO	1.1809e+00	1584	8

Table 5.6 Results of the Trust–Region Subproblem Method for problem **spikes** from the Regularization Tools package.

We also observe that even when the relative error in the solution of the trust–region subproblem is large, in most cases this solution behaves similarly to the solution of the original problem. We illustrate this fact in Figures 5.9 and 5.10, for problems **baart** and **ilaplace**, respectively. In some practical problems an approximation of this quality provides enough information about the desired solution.

In the next chapter we present the results obtained when we applied our method to a large–scale problem. The problem is an inverse interpolation problem and the data are real samples of the depths at different points of the Sea of Galilee.

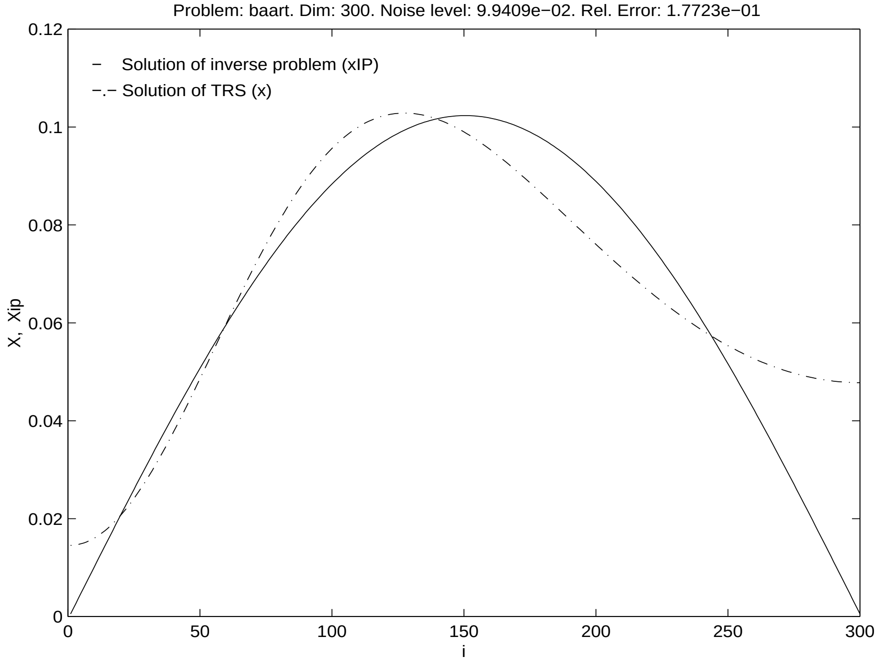

Figure 5.9 Trust–Region Subproblem solution for problem **baart**.

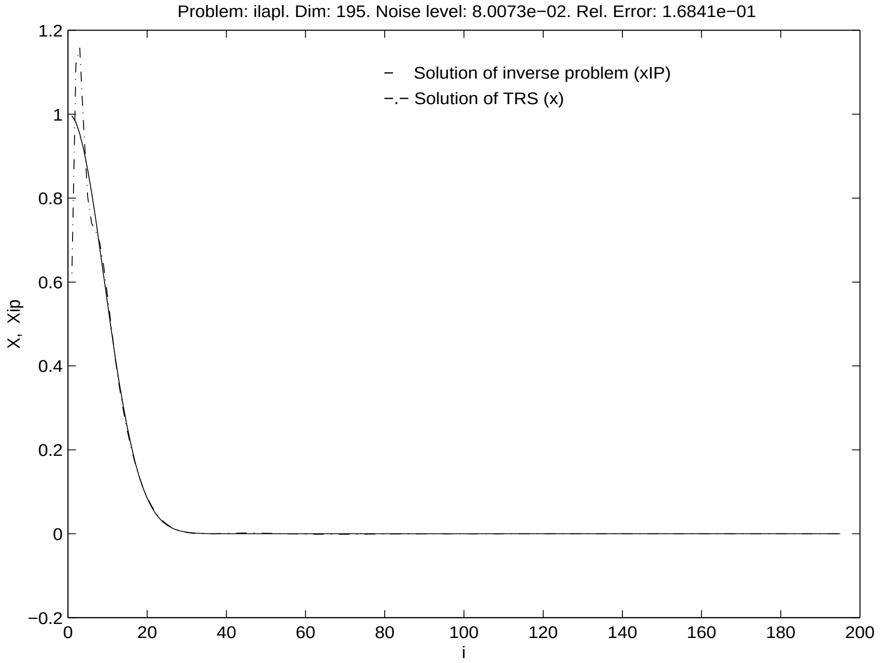

Figure 5.10 Trust–Region Subproblem solution for problem **ilaplace** (Inverse Laplacian).

Chapter 6

An Inverse Interpolation Problem

The linear interpolation problem as defined by Clærbout [11], consists of finding the values of a function at arbitrary points given the values of the function on a regular grid of points. If we construct the interpolant represented by the matrix $A \in \mathbb{R}^{m\times n}$ based on a regular grid of points, and we have that $c \in \mathbb{R}^n$ contains the irregular spaced points, then we compute the values of the function at c as $Ac = z$.

A more complex problem is the inverse interpolation problem: finding the values of the function on a regular grid of points from which we can extract the values of the function at irregularly spaced points by linear interpolation.

We can pose the inverse interpolation problem as the following least squares problem

$$\min_{c \in \mathbb{R}^n} \quad \|Ac - z\|$$

where $A \in \mathbb{R}^{m\times n}$ and $z \in \mathbb{R}^m$. The vector c represents the solution, i.e. the function values on a regular grid, the vector z contains the function values at irregular spaced points. The matrix A represents the linear interpolant.

To illustrate how our method works on this problem we will use the example of reconstructing a depth map of the Sea of Galilee.

The data consists of triples x_i, y_i, z_i, representing coordinates on the plane and depth, respectively, of the Sea of Galilee. The number of triples is 132044 and they were obtained by measuring the depth of the sea at different points, from a vessel.

Many sources of noise were present when the samples were obtained, including malfunctioning equipment that measured zero depths at points in the middle of the sea.

A plot of the original data also shows the tracks of the vessel used to take the samples. These tracks are the straight lines that we can observe in Figure 6.1. As Clærbout points out in [11] a good image of the sea should not show these lines.

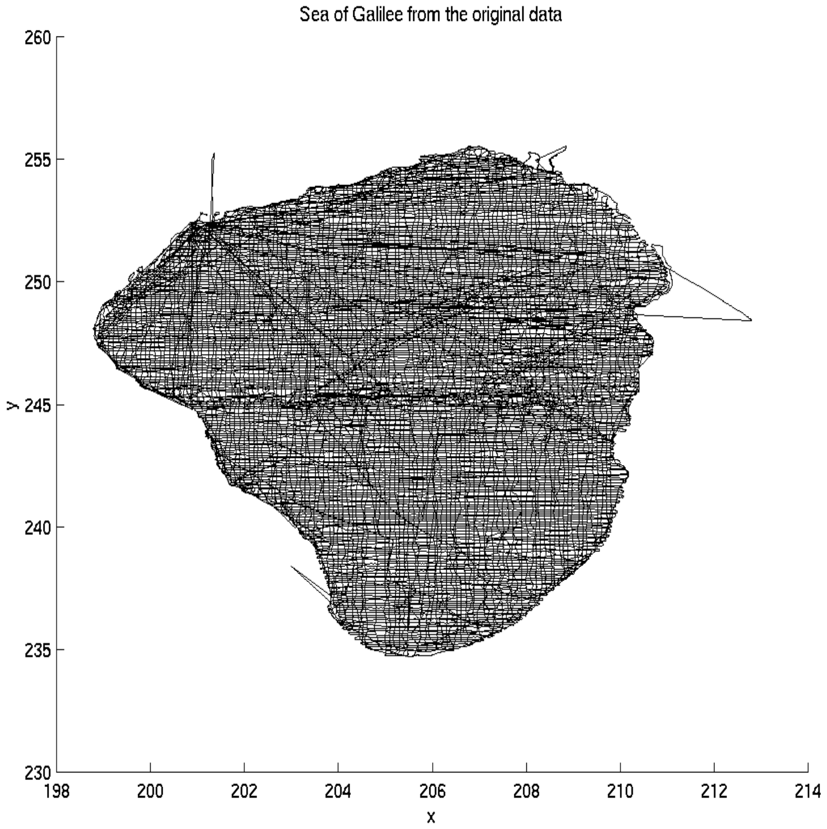

Figure 6.1 Sea of Galilee from the original data.

In our case the dimension of the grid is $n1 \times n2$ with $n1 = n2 = 201$ and therefore the matrix A is an $m \times n$ matrix, with $m = 132044$ and $n = 40401$. This matrix is ill–

conditioned and it is not available explicitly, instead we have procedures to evaluate Av and A^Tw for $v \in \mathbb{R}^n$ and $w \in \mathbb{R}^m$.

Initially, we posed the trust region subproblem as

$$\begin{array}{ll} \min & \frac{1}{2}c^TA^TAc - (A^Tz)^Tc \ . \\ s.t. & \|c\| \leq \Delta \end{array} \qquad (6.1)$$

This approach yields the solution shown in Figure 6.2 corresponding to $\Delta = 6000$, which gave the best image among several trial values for the trust region radius. This image still shows the tracks of the vessel and it does not show some of the features reported in [11]. This result indicated that it was necessary to introduce a stronger constraint on the smoothness of the solution.

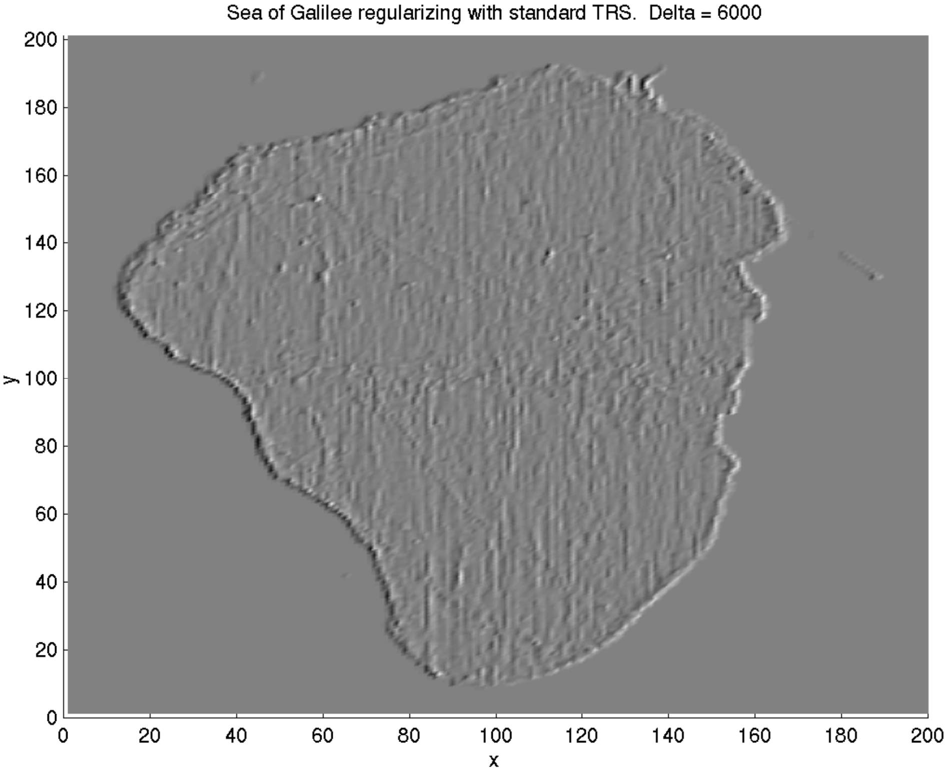

Figure 6.2 Sea of Galilee. Regularizing with standard TRS.

We then introduced a constraint on the smoothness of the solution of problem (6.1), obtaining the following trust region subproblem

$$\begin{array}{ll} \min & \frac{1}{2}c^T L^{-p} A^T A L^{-p} c - (L^{-p} A^T z)^T c \\ s.t. & {\scriptstyle ||L^p c|| \leq \Delta} \end{array} \qquad (6.2)$$

where L is a Helmholtz operator. Using this approach we produced the image shown in Figure 6.3 for $\Delta = 26000$ and $p = 0.3$. In this image we are able to identify some of the features that Clærbout recovered, such as some ancient shores that are now submerged on the southwest part of the sea and some shelves on the northeast part.

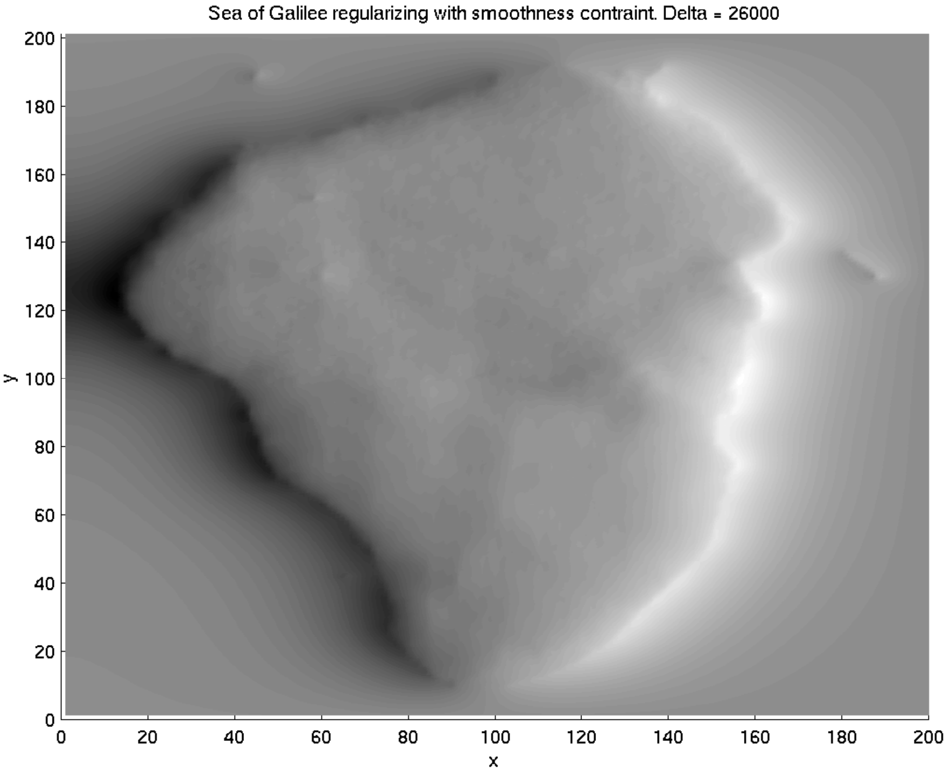

Figure 6.3 Sea of Galilee. Regularizing with constraint on smoothness.

After computing this solution we went back to our initial approach, but applying the smoother L^{-1} to the solution of (6.1). We call this approach *post smoothing*.

We obtained the image in Figure 6.4 using the post smoothing approach, for $\Delta = 23423$ which is the norm of x when $\|Lx\| = 26000$. The image is very similar to the one in Figure 6.3.

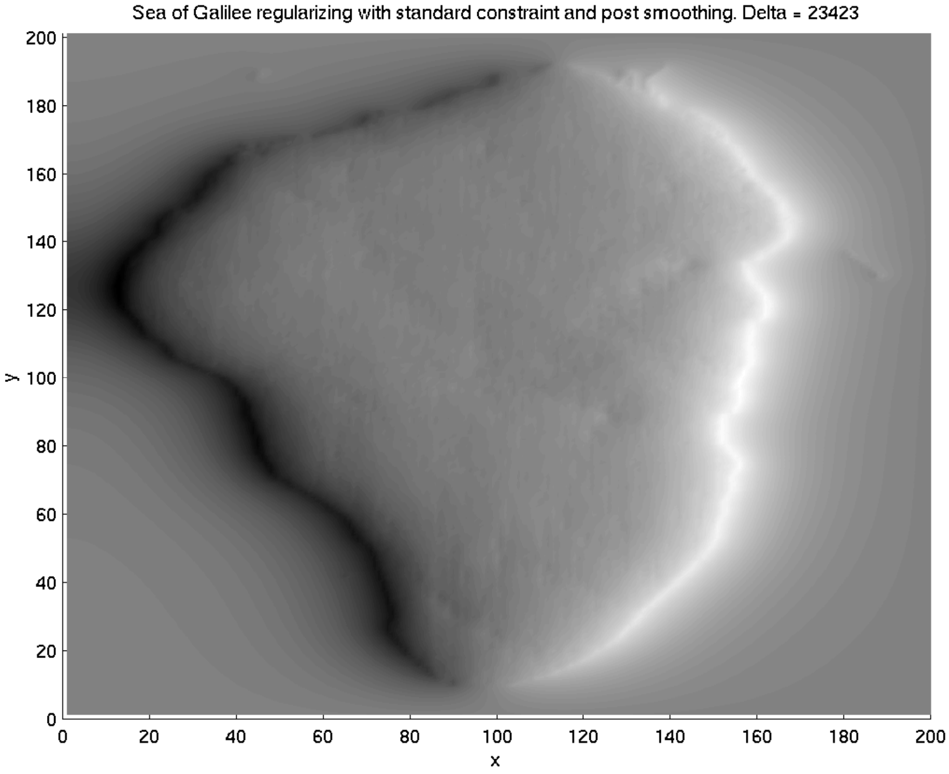

Figure 6.4 Sea of Galilee. Regularizing with standard constraint and post smoothing.

We also tried the postsmoothing approach for $\Delta = 6000$ and even in this case we recover the features that we mentioned before. We can observe that in Figure 6.5.

From the computational point of view, it is more efficient to solve problem (6.1) combined with the post smoothing technique than to solve problem (6.2). There are two reasons for this difference in efficiency. The first one is that the matrix–vector products with the matrix in (6.2) are more expensive than the matrix–vector products with A^TA. The second reason is that the smallest eigenvalues of $L^{-p}A^TAL^{-p}$

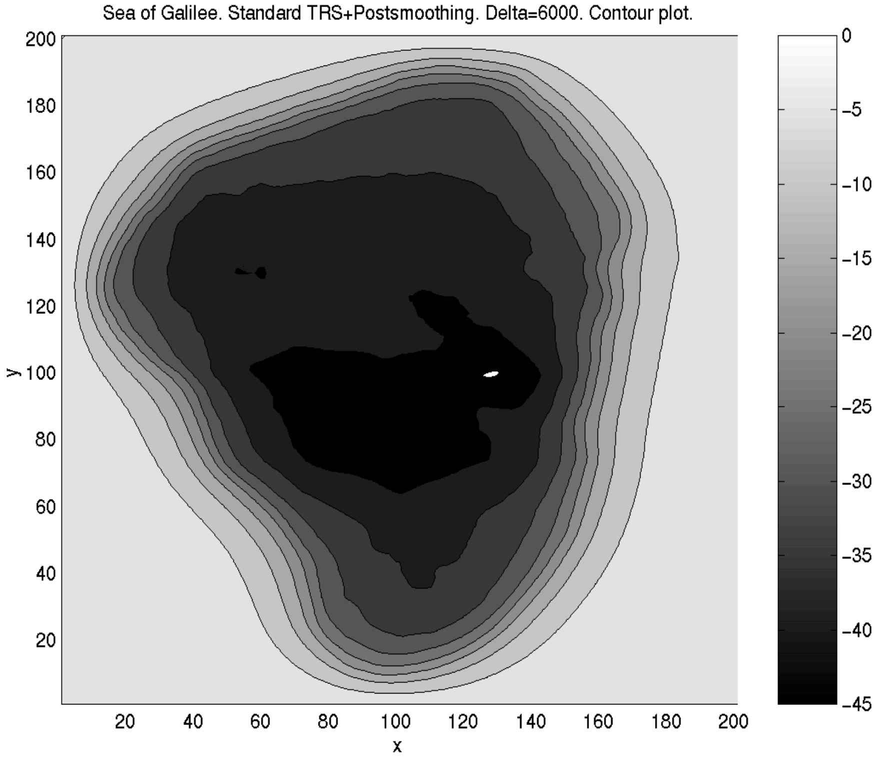

Figure 6.5 Sea of Galilee. Regularizing with standard constraint and post smoothing. Contour Plot.

are clustered so the IRLM uses many iterations trying to compute these eigenvalues and produces a low accuracy approximation to them. This in turn causes our method to use the safeguarding strategy most of the time.

In Table 6.1 we report the number of matrix–vector products and the CPU time used when we compute the regularized solution using the two approaches, namely: a trust region subproblem with a constraint on the smoothness of the solution and a trust region subproblem combined with the post smoothing technique. The times do not include the time of processing the data, we report them to illustrate the difference in cost of the matrix–vector products. Note that even though solving problem (6.2) is more expensive than solving problem (6.1) combined with post smoothing, the number of iterations is relatively low in both approaches.

Although the technique of post smoothing is very efficient computationally as we can see in Table 6.1, further study is needed to determine the physical meaning of this approach.

In all the experiments in this chapter, we used the same values for the tolerances as in Section 5.7 except for ε_{Δ} which is 10^{-3} in this case. The size of the Lanczos basis is five and we applied three shifts on each implicit restart. Therefore, the storage requirement is essentially five vectors of length 40401. This is the same storage requirement as for the Conjugate Gradient Method.

Problem Sea of Galilee. $n = 40401$	Type of Solution	Δ	$\|x_*\|$	TRS Iter.	M–V Prods.	CPU time (min.)
TRS with post smoothing	BS,QO	23423	23418.58	4	508	3.64
Constraint on smoothness	QO	26000	24777.97	10	1030	115.99

Table 6.1 Performance results of the TRS Method for an Inverse Interpolation Problem.

Chapter 7

Concluding Remarks

In this dissertation we have presented a method for the solution of general large–scale trust–region subproblems. Our method handles any level of singularities in the trust–region subproblem, including those arising from the regularization of discrete ill–posed problems.

Our method effectively computes the solution of the trust region subproblem and the Lagrange multiplier associated with that solution. The method does not require the explicit availability of the coefficient matrix and has low storage requirements.

We have developed the theoretical properties and a computer implementation of the method. We have successfully applied the method to the regularization of discrete ill–posed problems of medium to large scale.

We applied the method to an inverse interpolation problem on real data for which we computed a solution for the trust region subproblem at a very low computational cost.

After the numerical experiments, we reached the conclusion that the use of the quadratically constrained least squares approach for the regularization of discrete ill–posed problems requires careful study of each class of problem in order to formulate the appropriate trust region subproblem. The use of an inappropriate trust region subproblem might yield not so good approximations to the desired solution even when relevant information about this solution is available.

Many issues remain to be studied after this work. Some of these issues are the following:

- The development of efficient software that will allow the application of the method to larger problems.
- The implementation of the k vector version of the method for $k > 2$.
- The use of the TRS method in the context of optimization algorithms.
- Combination of the TRS method with a technique for estimating the trust region radius.
- Comparison with other methods for the large–scale trust–region subproblem.
- Comparison with other methods for the regularization of discrete ill–posed problems.
- Design of new preconditioning techniques for the efficient computation of clustered eigenvalues.
- Implementation of new deflation techniques to assure that the eigensolver computes an eigenvector with significant first component.

With this work we have contributed to the area of optimization providing a new tool for solving large–scale trust–region subproblems and to the area of regularization where our method can be applied successfully to obtain regularized solutions by solving large–scale quadratically constrained ill–conditioned least squares problems.

Bibliography

[1] J. Anderson. A secular equation for the eigenvalues of a diagonal matrix perturbation. *Lin. Alg. Appl.*, 246:49–70, 1996.

[2] J. Baglama, D. Calvetti, and L. Reichel. Iterative methods for computing a few eigenvalues of a large symmetric matrix. *BIT*, 36:400–421, 1996.

[3] R.H. Bartels, G.H. Golub, and M.A. Saunders. Numerical techniques in mathematical programming. In J.B. Rosen, O.L. Mangasarian, and K. Ritter, editors, *Nonlinear Programming*, pages 123–176. Academic Press, 1970.

[4] Å. Björck. *Numerical Methods for Least Squares Problems.* SIAM, Philadelphia, 1996.

[5] Å. Björck and L. Eldén. Methods in numerical algebra for ill–posed problems. Technical Report LiTH–MAT–R–33–1979, Department of Mathematics, Linköping University, Sweden, 1979.

[6] Å. Björck, E. Grimme, and P. Van Dooren. An implicit shift bidiagonalization algorithm for ill–posed systems. *BIT*, 34:510–534, 1994.

[7] J.R. Bunch, C.P. Nielsen, and D.C. Sorensen. Rank–one modification of the symmetric eigenproblem. *Numer. Math.*, 31:31–48, 1978.

[8] D. Calvetti, L. Reichel, and Q. Zhang. Iterative exponential filtering for large discrete ill–posed problems. To appear in Numer. Math., 1997.

[9] D. Calvetti, L. Reichel, and Q. Zhang. Iterative solution methods for large linear discrete ill–posed problems. To appear in Applied and Computational Control, Signals and Systems I, ed. B.N. Datta, 1997.

[10] T.F. Chan, J.A. Olkin, and D.W. Cooley. Solving quadratically constrained least squares using black box solvers. *BIT*, 32:481–495, 1992.

[11] J. Clærbout. Geophysical Exploration Mapping: Environmental soundings image enhancement. Available from *http://sepwww.stanford.edu/sep/jon/index.html*, February 1998.

[12] N. Clinthorne, T. Pan, P. Chiao, W. Rogers, and J. Stamos. Preconditioning methods for improved convergence rates in iterative reconstructions. *IEEE Trans. Med. Imag.*, 12(1):78–83, 1993.

[13] J.E. Dennis and H.H. Mei. Two new unconstrained optimization algorithms which use function and gradient values. *J. Opt. Theory Appl.*, 28(4):453–482, 1979.

[14] L. Eldén. Algorithms for the regularization of ill–conditioned least squares problems. *BIT*, 17:134–145, 1977.

[15] H. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems.* Kluwer Academic Publishers, Dordrecht, 1996.

[16] J. Eriksson. *Optimization and regularization of nonlinear least squares problems.* Ph.D. thesis, Umeå University, Umeå, Sweden, 1996.

[17] R.D. Fierro, G.H. Golub, P.C. Hansen, and D.P. O'Leary. Regularization by truncated total least squares. *SIAM J. Sci. Comput.*, 18(4):1223–1241, 1997.

[18] G.E. Forsythe and G.H. Golub. On the stationary values of a second–degree polynomial on the unit sphere. *J. Soc. Indust. Appl. Math.*, 13(4):1050–1068, 1965.

[19] W. Gander. Least squares with a quadratic constraint. *Numer. Math.*, 36:291–307, 1981.

[20] D. Gay. Computing optimal locally constrained steps. *SIAM J. Sci. Stat. Comput.*, 2(2):186–197, 1981.

[21] D.A. Girard. A fast 'Monte–Carlo Cross–Validation' procedure for large least squares problems with noisy data. *Numer. Math.*, 56:1–23, 1989.

[22] S.M. Goldfeld, R.E. Quandt, and H.F. Trotter. Maximization by quadratic hill–climbing. *Econometrica*, 34:541–551, 1966.

[23] G.H. Golub. Some modified matrix eigenvalue problems. *SIAM Review*, 15(2):318–334, 1973.

[24] G.H. Golub, M.T. Heath, and G. Wahba. Generalized cross–validation as a method for choosing a good Ridge parameter. *Technometrics*, 21(2):215–223, 1979.

[25] G.H. Golub and C.F. Van Loan. *Matrix Computations.* The John Hopkins University Press, Baltimore, second edition, 1989.

[26] G.H. Golub and U. von Matt. Quadratically constrained least squares and quadratic problems. *Numer. Math.*, 59:561–580, 1991.

[27] R.L. Graham, D.E. Knuth, and O. Patashnik. *Concrete Mathematics.* Addison–Wesley Publishing Company, Menlo Park, California, 1991.

[28] J. Hadamard. *Lectures on Cauchy's problem in linear partial differential equations.* Yale University Press, New Haven, 1923.

[29] M. Hanke. Accelerated Landweber iterations for the solution of ill–posed equations. *Numer. Math.*, 60:341–373, 1991.

[30] M. Hanke. Limitations of the L–curve method in ill–posed problems. *BIT*, 36(2):287–301, 1996.

[31] M. Hanke and P.C. Hansen. Regularization methods for large–scale problems. *Surv. Math. Ind.*, 3:253–315, 1993.

[32] M. Hanke, J. Nagy, and R. Plemmons. Preconditioned iterative regularization for ill–posed problems. In L. Reichel, A. Ruttan, and R. Varga, editors, *Numerical Linear Algebra*, Berlin, 1992. Walter de Gruyter & Co. Proceedings of the Conference in Numerical Linear Algebra and Scientific Computation, Kent (Ohio), 1992.

[33] P.C. Hansen. Computation of the Singular Value Expansion. *Computing*, 40:185–199, 1988.

[34] P.C. Hansen. The Discrete Picard Condition for discrete ill–posed problems. *BIT*, 30:658–672, 1990.

[35] P.C. Hansen. Analysis of discrete ill–posed problems by means of the L–curve. *SIAM Review*, 34(4):561–580, 1992.

[36] P.C. Hansen. Regularization Tools: a MATLAB package for analysis and solution of discrete ill–posed problems. *Numer. Algo.*, 6:1–35, 1994. Software available from *http://www.imm.dtu.dk/documents/users/pch/Regutools/regutools.html.*

[37] P.C. Hansen. *Rank–Deficient and Discrete Ill–Posed Problems.* Doctoral Dissertation, UNI•C, Technical University of Denmark, Lyngby, Denmark, 1996.

[38] P.C. Hansen and D.P. O'Leary. The use of the L–curve in the regularization of discrete ill–posed problems. *SIAM J. Sci. Comput.*, 14(6):1487–1503, 1993.

[39] M.D. Hebden. An algorithm for minimization using exact second derivatives. Technical Report T.P. 515, Atomic Energy Research Establishment, Harwell, England, 1973.

[40] L. Kaufman and A. Neumaier. Image reconstruction through regularization by envelope guided Conjugate Gradients. Technical Report 4–14, AT&T Bell Laboratories, 1994.

[41] J.T. King. Multilevel algorithms for ill–posed problems. *Numer. Math.*, 61:311–334, 1992.

[42] C.L. Lawson and R.J. Hanson. *Solving Least Squares Problems.* Classics in Applied Mathematics. SIAM, Philadelphia, 1995.

[43] R.B. Lehoucq, D.C. Sorensen, and C. Yang. *ARPACK User's Guide: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods.* SIAM, Philadelphia, 1998. Software available from *http://www.caam.rice.edu/software/ARPACK/UG/ug.html.*

[44] K. Levenberg. A method for the solution of certain non–linear problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944.

[45] C-J. Lin and J.J. Moré. Incomplete Cholesky factorizations with limited memory. Technical Report MCS–P682–0897, Mathematics and Computer Science Division, Argonne National Laboratory. Argonne, Illinois, August 1997.

[46] D. Luenberger. *Linear and Nonlinear Programming.* Addison–Wesley, Reading, Massachusetts, second edition, 1984.

[47] D.W. Marquardt. An algorithm for least–squares estimation of nonlinear parameters. *J. Soc. Indust. Appl. Math.*, 11(2):431–441, 1963.

[48] J.M. Martínez and S.A. Santos. A trust region strategy for minimization on arbitrary domains. To appear in *Mathematical Programming*, 1995.

[49] A. Melman. Numerical solution of a secular equation. *Numer. Math.*, 69:483–493, 1995.

[50] A. Melman. A numerical comparison of methods for solving secular equations. *J. Comp. App. Math.*, 86(1):237–249, 1997.

[51] A. Melman. A unifying convergence analysis of second–order methods for secular equations. *Math. Comp.*, 66(217):333–344, 1997.

[52] A. Melman. Analysis of third–order methods for secular equations. *Math. Comp.*, 67(221):271–286, 1998.

[53] J.J. Moré. The Levenberg-Marquardt algorithm: Implementation and theory. In G. A. Watson, editor, *Numerical Analysis. Proceedings Biennial Conference Dundee 1977*, volume 630 of *Lecture Notes in Mathematics*, pages 105–116. Springer–Verlag, Berlin, 1978.

[54] J.J. Moré. Recent developments in algorithms and software for trust region methods. In A. Bachem, M. Grötschel, and B. Korte, editors, *Mathematical Programming: the state of the art, Bonn 1982*, pages 258–287. Springer–Verlag, New York, 1983.

[55] J.J. Moré and D.C. Sorensen. Computing a trust region step. *SIAM J. Sci. Stat. Comput.*, 4(3):553–572, 1983.

[56] V.A. Morozov. On the solution of functional equations by the method of regularization. *Sov. Math. Doklady*, 167(3):414–417, 1966.

[57] J. Nagy, V. Pauca, R. Plemmons, and T. Torgersen. Space–varying restoration of optical images. Technical report, Dept. of Mathematics and Computer Science, Wake Forest University, Winston–Salem, North Carolina, March 1996.

[58] A. Neumaier. Solving ill–conditioned and singular linear systems: a tutorial on regularization. *SIAM Review*, 40(3):636–666, 1998.

[59] C.C. Paige and M.A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Softw.*, 8(1):43–71, 1982.

[60] B.N. Parlett. *The Symmetric Eigenvalue Problem.* Prentice–Hall, Englewood Cliffs, NJ, 1980.

[61] D.L. Phillips. A technique for the numerical solution of certain integral equations of the first kind. *J. ACM*, 9:84–97, 1962.

[62] M.J.D. Powell. A hybrid method for nonlinear equations. In P. Rabinowitz, editor, *Numerical Methods for Nonlinear Algebraic Equations*, pages 87–114. Gordon and Breach Science Publishers, 1970.

[63] M.J.D. Powell. A new algorithm for unconstrained optimization. In J.B. Rosen, O.L. Mangasarian, and K. Ritter, editors, *Nonlinear Programming*, pages 31–65. Academic Press, 1970.

[64] C.H. Reinsch. Smoothing by spline functions. *Numer. Math.*, 10:177–183, 1967.

[65] C.H. Reinsch. Smoothing by spline functions. II. *Numer. Math.*, 16:451–454, 1971.

[66] F. Rendl and H. Wolkowicz. A semidefinite framework for trust region subproblems with applications to large scale minimization. *Math. Prog.*, 77(2):273–299, 1997.

[67] M. Rojas, S.A. Santos, and D.C. Sorensen. A matrix–free algorithm for the large–scale trust–region subproblem. In preparation.

[68] S.A. Santos and D.C. Sorensen. A new matrix–free algorithm for the large–scale trust–region subproblem. Technical Report 95–20, Department of Computational and Applied Mathematics, Rice University, July 1995.

[69] D.C. Sorensen. Newton's method with a model trust region modification. *SIAM J. Numer. Anal.*, 19(2):409–426, 1982.

[70] D.C. Sorensen. Implicit application of polynomial filters in a k–step Arnoldi method. *SIAM J. Matrix Anal. Appl.*, 13(1):357–385, 1992.

[71] D.C. Sorensen. Minimization of a large–scale quadratic function subject to a spherical constraint. *SIAM J. Optim.*, 7(1):141–161, 1997.

[72] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Numer. Anal.*, 20(3):626–637, 1983.

[73] G.W. Stewart. *Introduction to Matrix Computations.* Academic Press, New York, 1973.

[74] A.N. Tikhonov and V.Y. Arsenin. *Solutions of ill–posed problems.* Halsted Press, New York, 1977.

[75] S. Van Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis*, volume 9 of *Frontiers in Applied Mathematics.* SIAM, Philadelphia, 1991.

[76] J. Varah. A practical examination of some numerical methods for linear discrete ill–posed problems. *SIAM Review*, 21(1):100–111, 1979.

[77] C.R. Vogel. A constrained least squares regularization method for nonlinear ill–posed problems. *SIAM J. Control and Optim.*, 28(1):34–49, 1990.

[78] C.R. Vogel. Non–convergence of the L–curve regularization parameter selection method. To appear in Inverse Problems, 1996.

[79] G. Wahba. *Spline Models for Observational Data.* SIAM, Philadelphia, 1990.