Trust Region Subproblems and Linear Least-Squares Regularization

by

Michael S. Froh

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics and Optimization

Waterloo, Ontario, 2003

I hereby declare that I am the sole author of this thesis.
I authorize the University of Waterloo to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Michael S. Froh

I authorize the University of Waterloo to reproduce this thesis by photocopying or other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Michael S. Froh

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Acknowledgements

I would like to thank my supervisor, Henry Wolkowicz for his guidance and support during my studies. I would like to thank Brian Borchers of New Mexico Tech for posing the questions which we try to answer in this document. I thank my readers, Levent Tunçel and Edward Vrscay for their time in reviewing this work. I would also like to thank my friends in other faculties for making me explain my research in non-mathematical terms, often providing me with insights I would most likely have otherwise overlooked.

Abstract

Solving an ill-conditioned linear least-squares problem, minimizing the norm of the residual vector, in practice often yields a solution which may differ significantly from the "true" solution, particularly when the right-hand side is subject to noise. By finding the solution of minimum norm, subject to some tolerance on the norm of the residual, we find a "smoother" solution, which, in practice, is closer to the true solution. In this work, we make use of the fact that this process of regularization is a special case of the Trust-Region Subproblem (TRS), and apply the work of Rendl and Wolkowicz to derive a new method for computing a regularized solution by computing an eigenpair. The technique exploits sparsity, and scales to large problems.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 What is Regularization?

1.1.1 Linear Least-Squares Problems

Consider the problem:

$$\min_x \|Gx - d\|_2 , \tag{1.1}$$

where $x \in \mathbb{R}^n$, $d \in \mathbb{R}^m$, and $G \in \mathbb{R}^{m \times n}$. Denote

$$G = \begin{bmatrix} g_1^T \\ g_2^T \\ \vdots \\ g_m^T \end{bmatrix} .$$

Note that squaring the objective function in (1.1) shows that the problem is equivalent to solving

$$\min_x \|Gx - d\|_2^2 \equiv \sum_{i=1}^{m} (g_i^T x - d_i)^2. \tag{1.2}$$

Hence, we refer to (1.1) as a *least-squares problem*. For a given vector x, we refer to $Gx - d$ as the *residual* of x, and refer to the norm of the residual, $\|Gx - d\|_2$, as the *discrepancy* of x.

As $\|\cdot\|_2^2$ is convex and differentiable, any solution to (1.2) is characterized by the *stationarity condition*:

$$0 = \nabla_x \|Gx - d\|_2^2 = 2G^T Gx - 2G^T d,$$

or simply

$$G^T Gx = G^T d, \tag{1.3}$$

known as the *normal equations*. If $m \geq n$ and G has full rank, then the normal equations have a unique solution, which may be found using standard techniques for solving systems of linear equations (e.g. finding a Cholesky factorization of $G^T G$ or a QR factorization of G). Otherwise, given a solution $\bar{x}$ to (1.1), the manifold of solutions is $\{\bar{x} + r : r \in \mathcal{N}(G)\}$.

In Chapter 2, we show that the solution to (1.1) with minimum norm is

$$x_\dagger = G^\dagger d,$$

where $G^\dagger$ represents the Moore-Penrose generalized inverse of G. We refer to $x_\dagger$ as the *best least-squares solution*. Some properties of the Moore-Penrose inverse and the best least-squares solution are discussed in greater detail in Chapter 2. Also, see e.g. [2] for more details.

1.1.2 Regularization of Linear Least-Squares Problems

In practice, the elements of d in (1.1) are often obtained from measurements which may be subject to error. This may be modeled by setting $d = d_{true} + e_d$, where d_{true} is the correct set of values to be used in (1.1) and $e_d \in \mathbb{R}^m$ is a vector of random variables representing the measurement errors. When attempting to fit data to a real-world problem, we will assume that the model, given correct data, will yield a perfect fit. In other words, it is assumed that $Gx = d_{true}$ is consistent. (This assumption also appears in [1].)

Recall from Section 1.1.1 that the best least-squares solution is given by

$$x_\dagger = G^\dagger d = G^\dagger (d_{true} + e_d).$$

A desired solution is given by $x_{true} = G^\dagger d_{true}$. Thus, the error in the minimum norm solution is given by

$$r = x_\dagger - x_{true} = G^\dagger (e_d).$$

In the case that G is ill-conditioned, $\|r\|$ may be larger than given tolerances would allow. Thus, the best least-squares solution $x_\dagger$ may be unacceptable, as it fits noisy data too closely. Therefore, it is helpful to reformulate the problem to try to obtain a solution of minimum norm, subject to some tolerance for discrepancy in the solution. In other words, we instead solve

$$\begin{aligned} \min_x \quad & \|x\|_2 \\ \text{s.t.} \quad & \|Gx - d\|_2 \leq \epsilon, \end{aligned} \tag{1.4}$$

given some $\epsilon \in \mathbb{R}_+$. The technique of allowing greater discrepancy in order to find a smoother solution is known as *regularization*. Note that

$$\|Gx_{true} - d\|_2 = \|Gx_{true} - d_{true} - e_d\|_2 = \|e_d\|_2 .$$

Thus, in order to potentially obtain x_{true} as a solution, we want to set $\epsilon = \|e_d\|_2$, the norm of the error in the data. This is known as the *discrepancy principle* (see e.g. [15]). Unfortunately, $\|e_d\|$ is likely unknown, so we use an estimate instead. If, for example, e_d is a vector of normally-distributed random variables with known standard deviations, we may use $\epsilon = E(\|e_d\|_2)$, the expected value of $\|e_d\|_2$.

Techniques for regularizing linear least-squares problems are discussed in more detail in Chapter 2.

1.2 What is the Trust Region Subproblem?

1.2.1 Motivation – Trust Region Methods

Consider the general unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1.5}$$

where f is assumed to be twice-differentiable and bounded below. We outline an iterative technique for solving (1.5) based on a quadratic approximation of f at the current iterate.

Algorithm 1.2.1 *Basic Trust Region Algorithm*
INPUT: *Initial point* x_0*, step quality thresholds* η_0, η_1 *satisfying* $0 < \eta_0 \leq \eta_1 < 1$ *and initial trust region radius* Δ_0.

1. *Set* $k = 0$.

2. *Determine a norm* $\|\cdot\|_k$, *and a model* $m_k(x)$ *that approximates* f *within some neighbourhood of* x_k.

3. *Find a step* s_k *such that* $\|s_k\| \leq \Delta_k$, *and* $m_k(x_k + s_k)$ *is "sufficiently" less than* $m_k(x_k)$.

4. *Evaluate*
$$\frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$
If the result is $\geq \eta_0$, *set* $x_{k+1} = x_k + s_k$. *If the result is greater than* η_1, *set* Δ_{k+1} *to be larger than* Δ_k *(otherwise set* $\Delta_{k+1} = \Delta_k$*). If the result is less than* η_0, *set* $x_{k+1} = x_k$, *and set* Δ_{k+1} *to be less than* Δ_k.

5. *Increment* k *by 1 and go to step 2.*

Note that the above algorithm does not define stopping criteria, nor does it describe details for the individual steps. For additional information on the Trust Region Subproblem, see e.g. [4].

The specific implementation of Step 3 is known as the *Trust Region Subproblem*. A common approach is to solve

$$\begin{array}{ll} \min & q(s) \\ \text{s.t.} & \|s\| \leq \Delta_k, \end{array} \tag{1.6}$$

where q is a quadratic approximation of f at x_k, with x_k translated to the origin. The l_2 norm is commonly used.

The Trust Region Subproblem is discussed in detail in Chapter 3.

1.3 Basic Optimization Theory

1.3.1 SDP Duality

Definition 1.3.1 *Let* $\mathcal{A} : \mathcal{E} \to \mathcal{F}$ *be a linear transformation where* $\mathcal{E}$, $\mathcal{F}$ *are inner-product spaces. We define the* adjoint of $\mathcal{A}$, *denoted by* $\mathcal{A}^* : \mathcal{F} \to \mathcal{E}$, *as a linear transformation*

satisfying

$$\langle \mathcal{A}(x), y \rangle_{\mathcal{F}} = \langle x, \mathcal{A}^*(y) \rangle_{\mathcal{E}} ,$$

for all $x \in \mathcal{E}$ *and* $y \in \mathcal{F}$.

In semidefinite programming, our variables are elements of $\mathcal{S}^n$, the space of $n \times n$ real symmetric matrices, to which we apply the Frobenius inner-product:

$$\langle X, Y \rangle_F = \text{trace}\,(XY).$$

The standard form of a semidefinite programming problem is

$$(P) \qquad \begin{array}{ll} \min & \text{trace}\,(CX) \\ \text{s.t.} & \mathcal{A}(X) = b \\ & X \succeq 0, \end{array} \tag{1.7}$$

where $X, C \in \mathcal{S}^n$, $\mathcal{A}$ is a linear transformation from $\mathcal{S}^n$ to $\mathbb{R}^m$, and $b \in \mathbb{R}^m$. The notation $X \succeq 0$ means that X is positive semidefinite. The dual program corresponding to (1.7) is

$$(D) \qquad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & \mathcal{A}^*(y) + S = C \\ & S \succeq 0, \end{array} \tag{1.8}$$

where $y \in \mathbb{R}^m$, $S \in \mathcal{S}^n$, and $\cdot^*$ denotes the adjoint operator. Weak SDP duality states that for any $\bar{X}$ feasible for (1.7) and $(\bar{y}, \bar{S})$ feasible for (1.8), it follows that

$$\text{trace}\,(C\bar{X}) - b^T y = \text{trace}\,(\bar{X}\bar{S}) \geq 0.$$

Definition 1.3.2 *A matrix* $\tilde{X} \in \mathcal{S}^n$ *is called a* Slater point for (1.7) *if* $\mathcal{A}(\tilde{X}) = b$, *and* $\tilde{X} \succ 0$.

Definition 1.3.3 *A pair* $(\tilde{y}, \tilde{S}) \in \mathbb{R}^m \times \mathcal{S}^n$ *is called a* Slater point for (1.8) *if* $\mathcal{A}^*(\tilde{y}) + \tilde{S} = C$ *and* $\tilde{S} \succ 0$.

If (1.8) has a Slater point, and the objective value of (1.8) is bounded above, then (1.7) has an optimal solution, and the optimal values of (1.7) and (1.8) are equal.

Chapter 2

Regularization Theory

Recall from Chapter 1 that the linear least-squares problem (1.1) is

$$\min_x \|Gx - d\|_2 ,$$

where $x \in \mathbb{R}^n$, $d \in \mathbb{R}^m$, and $G \in \mathbb{R}^{m \times n}$.

2.1 Discrete Regularization

2.1.1 Singular Value Decompositions and the Moore-Penrose Generalized Inverse

Given the matrix G as defined above, the *singular value decomposition* of G is:

$$G = U \Sigma V^T$$

where:

- U is an $m \times m$ orthogonal matrix whose column vectors form an orthonormal basis of $\mathbb{R}^m$ (the *data space*);
- V is an $n \times n$ orthogonal matrix whose column vectors form an orthonormal basis of $\mathbb{R}^n$ (the *model space*);

- Σ is an $m \times n$ matrix. The entries of Σ are nonnegative, and the (i,j) entry is given by

$$\Sigma_{ij} = \begin{cases} \sigma_i & \text{if } i = j, \text{ and} \\ 0 & \text{if } i \neq j. \end{cases}$$

  where σ_i is the i-th smallest *singular value* of G.

For more discussion of singular value decompositions, see, e.g. [1].

We may define a square diagonal matrix $\bar{\Sigma} = \text{Diag}\,(\sigma_1, \ldots, \sigma_p)$, where $\sigma_1, \ldots, \sigma_p$ are the strictly positive singular values of G (and we may define $\bar{U}$, $\bar{V}$ to be the column submatrices of U and V, respectively, corresponding to the p positive singular values of G). In particular, since any rows or columns discarded from Σ to obtain $\bar{\Sigma}$ are identically zero,

$$G = \bar{U}\bar{\Sigma}\bar{V}^T.$$

The columns of $\bar{U}$ form a basis for $\mathcal{R}(G)$, while the columns of $\bar{V}$ form a basis for $\mathcal{R}(G^T)$.

Note that $\bar{\Sigma}$ is a diagonal matrix with strictly positive diagonal entries, and is hence invertible. The Moore-Penrose inverse of G is defined to be:

$$G^\dagger = \bar{V}\bar{\Sigma}^{-1}\bar{U}^T. \tag{2.1}$$

Several properties of the Moore-Penrose generalized inverse should be noted based on this construction. In particular, given $G \in \mathbb{R}^{m\times n}$ and its Moore-Penrose generalized inverse $G^\dagger$, the following hold (see e.g. [2]):

1. $GG^\dagger G = G$,
2. $G^\dagger GG^\dagger = G^\dagger$,
3. $(GG^\dagger)^T = GG^\dagger$, and
4. $(G^\dagger G)^T = G^\dagger G$.

We refer to

$$x_\dagger = G^\dagger d$$

as the best least squares solution to (1.1), and obtain the following result (proved in e.g. [2]).

Theorem 2.1.1 *The best least squares solution $x_\dagger$ is a solution to* (1.1)*, and furthermore, given any solution x to* (1.1)*, we have*

$$\|x_\dagger\|_2 \leq \|x\|_2 .$$

Proof. Note that (1.1) is equivalent to

$$\min \left\| U\Sigma V^T x - d \right\|_2^2 .$$

Specifically, we order the columns of U and V, and the nonzero entries of Σ such that

$$\Sigma = \begin{bmatrix} \bar{\Sigma} & 0_{p\times(n-p)} \\ 0_{(m-p)\times p} & 0_{(m-p)\times(n-p)} \end{bmatrix},$$

where $0_{i\times j}$ is the $i \times j$ zero matrix. Furthermore, we define $\tilde{U}$ by $U = \begin{bmatrix} \bar{U} & \tilde{U} \end{bmatrix}$ and $\tilde{V}$ by $V = \begin{bmatrix} \bar{V} & \tilde{V} \end{bmatrix}$. As U is an orthogonal matrix, it is norm-invariant. Thus, (1.1) is further equivalent to

$$\min \left\| \Sigma(V^T x) - U^T d \right\|_2^2 .$$

If we define $y \in \mathbb{R}^n$ by $y = V^T x$, then we may rewrite the problem as

$$\min \left\| \Sigma y - U^T d \right\|_2^2$$

or

$$\min \left\| \bar{\Sigma}\bar{y} - \bar{U}^T d \right\|_2^2 + \left\| 0 - \tilde{U}^T d \right\|_2^2 , \tag{2.2}$$

where $\bar{y} \in \mathbb{R}^p$, and

$$y = \begin{bmatrix} \bar{y} \\ \tilde{y} \end{bmatrix}.$$

Note that y is a solution to (2.2) if and only if

$$\bar{y} = \bar{\Sigma}^{-1}\bar{U}^T d.$$

Furthermore, the solution to (2.2) with minimum norm would be the one satisfying $\tilde{y} = 0$. Thus, a minimum norm solution x to (1.1) is characterized by

$$\begin{bmatrix} \bar{y} \\ 0 \end{bmatrix} = V^T x = \begin{bmatrix} \bar{V}^T x \\ \tilde{V}^T x \end{bmatrix}.$$

Thus, $\tilde{V}^T x = 0$, and

$$\bar{V}^T x = \bar{y}. \tag{2.3}$$

Noting that

$$x = x_\dagger = \bar{V}\bar{\Sigma}^{-1}\bar{U}^T d = \bar{V}\bar{y}$$

solves (2.3) (since $\bar{V}^T\bar{V} = I_p$), we have the desired result. ■

Note that $x_\dagger$ may also be written as

$$x_\dagger = \sum_{i=1}^{p} \frac{u_i^T d}{\sigma_i} v_i, \tag{2.4}$$

where u_i, v_i are the i-th columns of U and V respectively.

Recall from Chapter 1 that the value of d is usually subject to some measurement error (i.e. $d = d_{true} + e_d$). This error can have a considerable effect on the determined value of $x_\dagger$. In particular, as $x_{true} = G^\dagger d_{true}$,

$$\begin{aligned} \|x_\dagger - x_{true}\|_2 &= \left\|G^\dagger(d - d_{true})\right\|_2 \\ &= \left\|G^\dagger e_d\right\|_2 \\ &\leq \left\|G^\dagger\right\| \|e_d\|_2 \\ &= \frac{1}{\sigma_p} \|e_d\|_2 . \end{aligned}$$

Note that if $e_d = \alpha u_p$, then by the application of (2.4),

$$G^\dagger e_d = \alpha \frac{u_p^T u_p}{\sigma_p} v_p = \frac{\alpha}{\sigma_p} v_p,$$

and so the above bound is tight. Furthermore, note that

$$\begin{aligned}
\|x_{true}\| &= \left\| \sum_{i=1}^{p} \frac{u_i^T d_{true}}{\sigma_i} v_i \right\|_2 \\
&\geq \left\| \frac{u_1^T d_{true}}{\sigma_1} v_1 \right\|_2 \\
&\geq \left\| \frac{\|d_{true}\|_2}{\sigma_1} v_1 \right\|_2 \\
&= \frac{\|d_{true}\|_2}{\sigma_1} \|v_1\|_2 \\
&= \frac{\|d_{true}\|_2}{\sigma_1}.
\end{aligned}$$

Combining the previous two results, we obtain

$$\frac{\|x_\dagger - x_{true}\|_2}{\|x_\dagger\|_2} \leq \frac{\sigma_1}{\sigma_p} \frac{\|e_d\|_2}{\|d_{true}\|_2}.$$

Thus, we see that the value of $\frac{\sigma_1}{\sigma_p}$ can be interpreted as a scaling factor for the relative error of the best least-squares solution. This value $\frac{\sigma_1}{\sigma_p}$ is known as the *condition number* of G, and is denoted by $\text{cond}\,(G)$. If a matrix has large condition number (i.e. the largest singular value is several orders of magnitude larger than the smallest singular value), then the matrix is said to be *ill-conditioned*.

2.1.2 The Truncated Singular Value Decomposition

As discussed at the end of the previous section, relatively small singular values may result in large errors in our least-squares solution.

We may approximate $G^\dagger$ with

$$G_l^\dagger = \bar{V}_l \bar{\Sigma}_l^{-1} \bar{U}_l^T$$

where $1 \leq l \leq p$, $\bar{U}_l, \bar{V}_l$ are the submatrices obtained from the first l columns of $\bar{U}$ and $\bar{V}$, respectively, and $\bar{\Sigma}_l = \text{Diag}\,(\sigma_1, \ldots, \sigma_l)$. Furthermore, we denote by $x_\dagger^l = G_l^\dagger d$ the *truncated singular value decomposition solution of rank* l (TSVD) of (1.1). This is an approximate

solution which satisfies

$$\frac{\left\|x_\dagger^l - x_{true}\right\|_2}{\left\|x_\dagger^l\right\|_2} \leq \frac{\sigma_1}{\sigma_l}\left\|d\right\|_2 \leq \frac{\sigma_1}{\sigma_p}\left\|d\right\|_2 .$$

Thus, by discarding successively larger singular values of G, we obtain progressively lower upper bounds on the norm of our solution, at the cost of increased inaccuracy.

Example 2.1.2 *Using the MATLAB* shaw *routine in [17], we generate* G, d_{true}, x_{true} *for a discretization of a Fredholm integral equation of the first kind (as described in [28]). We select the discretization to give* $G \in \mathbb{R}^{20\times 20}$ *and* $x, d \in \mathbb{R}^{20}$*. Note that for this problem,* G *has rank* 20*, and has condition number* 9.2692×10^{15}*, based on largest and smallest singular values of* 2.9934 *and* 3.2294×10^{-16}*, respectively (as found using MATLAB). We perturb* d_{true} *as* $d = d_{true} + e_d$ *for some vector* $e_d \in \mathbb{R}^{20}$ *of normally distributed random values with mean* 0 *and standard deviation* 0.1*. We then proceed to find* $x_\dagger^l$ *using* G *and* d*, for values of* l *between* 1 *and* 18*. The results are presented in Table 2.1 and graphed on a log-log scale in Figure 2.1. The roughly "L"-shaped distribution of approximate solutions arises frequently for ill-posed problems, and is discussed in greater detail in Section 2.2.*

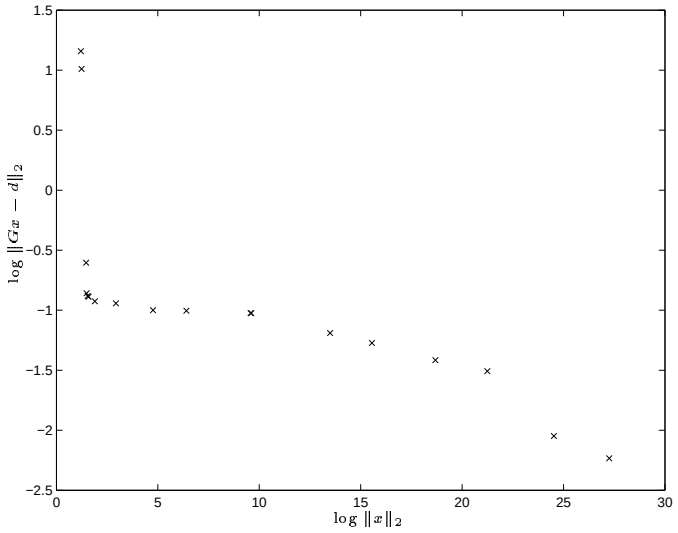


Figure 2.1: Solution points for Example 2.1.2, on a log-log scale.

Rank (l)	$\|Gx_\dagger^l - d\|$	$\|x_\dagger^l\|$
1	3.1794e+00	3.3585e+00
2	2.7570e+00	3.4651e+00
3	4.7863e-01	4.3471e+00
4	3.4775e-01	4.4267e+00
5	3.3921e-01	4.6135e+00
6	3.3884e-01	4.6364e+00
7	3.1808e-01	6.6373e+00
8	3.1292e-01	1.5666e+01
9	3.0980e-01	4.2594e+01
10	3.0806e-01	5.8060e+02
11	3.0295e-01	1.1010e+04
12	3.0027e-01	2.9154e+04
13	2.6087e-01	5.6641e+05
14	2.4009e-01	4.8848e+06
15	2.2587e-01	7.6133e+07
16	2.2371e-01	5.3159e+08
17	1.8319e-01	3.2011e+10
18	1.5722e-01	8.9940e+11

Table 2.1: TSVD solution and residual norms for 20×20 Shaw problem.

2.2 Regularization by Continuous Parameters

The previous section discussed a means of regularizing least-squares solutions, parametrized by the rank l of the truncated singular value decomposition. In this section, we introduce several parameters which may be varied continuously to obtain an infinite family of regularized solutions.

2.2.1 Basic Regularization Theory

When selecting from several available regularized solutions, we apply the *discrepancy principle* [1], which dictates that we select the solution that minimizes the norm of x, subject to a constraint on the accuracy of the model as applied to the data, parametrized by a tolerance δ (which should correspond to the norm of the error in d):

$$\begin{aligned} \epsilon = \quad & \min \quad \|x\| \\ & \text{s.t.} \quad \|Gx - d\| \leq \delta. \end{aligned} \tag{2.5}$$

Over varying values of δ, this problem yields various values for the optimal $\|x\|$ and $\|Gx - d\|$, which follow a curve of optimal values. Given a sufficiently ill-conditioned matrix G, if this curve is plotted on a log-log scale, the resulting graph often resembles an "L". Thus, this curve is called an L-curve (see e.g. [15]). In the two lemmas below, we show that the choice of δ above and subsequent solution for ϵ are consistent with the following:

$$\begin{aligned} \delta = \quad & \min \quad \|Gx - d\| \\ & \text{s.t.} \quad \|x\| \leq \epsilon. \end{aligned}$$

This result appears in an exercise in [1].

Lemma 2.2.1 *Suppose G $m \times n$ and $d \in \mathbb{R}^m$ are given. Let*

$$0 \leq \epsilon \leq \|G^\dagger d\| \tag{2.6}$$

and

$$\begin{aligned} \delta = \quad & \min \quad \|Gx - d\| \\ & \text{s.t.} \quad \|x\| \leq \epsilon. \end{aligned} \tag{2.7}$$

Then

$$\begin{array}{rll} \epsilon = & \min & \|y\| \\ & \text{s.t.} & \|Gy - d\| \leq \delta. \end{array} \tag{2.8}$$

Proof. Note that we can square the terms to simplify the differentiations.

Suppose that $0 < \epsilon < \|G^\dagger d\|$. First we look at program (2.7). The compactness of the feasible region guarantees that an optimum x^* exists. This is a convex program and the Slater constraint qualification (strict feasibility) holds. The first-order necessary optimality conditions imply that

$$2G^T Gx^* - 2G^T d + 2\lambda x^* = 0,$$

for some $\lambda \geq 0$. If $\lambda = 0$ then $G^T Gx^* = G^T d$, and so $\|x^*\| = \|G^\dagger d\|$, implying that x^* is infeasible for (2.7). Thus, the optimal Lagrange multiplier $\lambda > 0$ and so complementary slackness implies that the norm of the optimum $\|x^*\| = \epsilon$, i.e.

$$\|x^*\| = \epsilon, \quad \|Gx^* - d\| = \delta. \tag{2.9}$$

Now suppose that y^* is an optimum for (2.8). By (2.9), x^* is feasible for (2.8), and so

$$\|y^*\| \leq \epsilon, \quad \|Gy^* - d\| \leq \delta. \tag{2.10}$$

Now, y^* is feasible and optimal for (2.7). Thus, we cannot have $\|y^*\| < \epsilon$ or we would contradict the complementary slackness argument stated above for program (2.7).

If $0 = \epsilon \leq \|G^\dagger d\|$ then the only feasible point in (2.7) is $x = 0$. Thus, the optimal solution is $x^* = 0$ and $\delta = \|d\|$. In (2.8), the unconstrained minimizer $y = 0$ is feasible, and thus is optimal. Thus the result holds.

If $0 < \epsilon = \|G^\dagger d\|$, then $x = G^\dagger d$ is feasible for (2.7). As stated previously, this solution is an unconstrained minimizer for $\|Gx - d\|$, and hence is optimal for (2.7). Thus, $\delta = \|GG^\dagger d - d\|$. In (2.8), $y = G^\dagger d$ is feasible, and is the solution of least norm. Thus it is optimal. Hence, the optimal value for (2.8) will be $\|G^\dagger d\|$. ■

Lemma 2.2.2 *Suppose G $m \times n$ and $d \in \mathbb{R}^m$ are given. Let*

$$\|GG^\dagger d - d\| \leq \delta \leq \|d\|. \tag{2.11}$$

If ϵ is defined as in (2.8)*, then* (2.7) *holds.*

Proof. As in the proof of (2.2.1), we may square all values to simplify differentiation.

Suppose that $\|GG^\dagger d - d\| < \delta < \|d\|$. Consider first, the program (2.8). The point $y = G^\dagger d$ is strictly feasible, and so Slater's constraint qualification holds. The objective function is strictly convex, and so the existence of an optimum y^* is assured. The first-order necessary optimality conditions imply that

$$2y^* - 2\lambda G^T G y^* - 2\lambda G^T d = 0$$

for some $\lambda \geq 0$. If $\lambda = 0$ then $y^* = 0$. However, this would imply that $\|Gy^* - d\| = \|d\|$, which implies that y^* is infeasible. Thus, $\lambda > 0$. By complementary slackness, this implies that $\|Gy - d\| = \delta$. i.e.

$$\|y^*\| = \epsilon, \|Gy^* - d\| = \delta$$

Now, suppose that x^* is an optimum for (2.7). Since y^* above is feasible for (2.7), we obtain an upper-bound on $\|x^*\|$. Hence, we may infer the following conditions on x^*:

$$\|x^*\| \leq \epsilon, \|Gx^* - d\| \leq \delta.$$

Now, x^* is feasible and optimal for (2.8). Thus, we cannot have $\|Gx^* - d\| < \delta$, or we would contradict the complementary slackness argument made above.

Suppose that $\|GG^\dagger d - d\| = \delta \leq \|d\|$. Thus, a feasible solution to (2.8) would be $y = G^\dagger d$. Among solutions that maintain this level of accuracy, $G^\dagger d$ has the least norm. Hence this solution is optimal for (2.8), and so $\epsilon = \left\|G^\dagger d\right\|$. Thus, $x = G^\dagger d$ is feasible for (2.7). Since this is an unconstrained minimizer for $\|Gx - d\|$, it will also be an optimum for (2.7), and the optimal value will be $\|GG^\dagger d - d\|$.

Suppose that $\|GG^\dagger d - d\| < \delta = \|d\|$. Then, $y = 0$ is a feasible solution for (2.8). As it is also an unconstrained minimizer for $\|y\|$, we therefore have that $\epsilon = 0$. Thus, the only feasible point in (2.7) is $x = 0$, and the optimal value for (2.7) must therefore be $\|d\|$. ■

Thus, we see that under reasonable assumptions, (2.8) and (2.7) are equivalent problems.

2.2.2 Tikhonov Regularization

In this section, we consider the problem

$$\min_x \|Gx - d\|_2^2 + \alpha^2 \|x\|_2 , \tag{2.12}$$

varying the value of α. We will see that α is a regularization parameter, like ϵ and δ from the previous section.

The problem presented in (2.12) provides a continuous method for regularizing a linear least-squares problem. This method is known as *Tikhonov regularization*. Note that if $\alpha = 0$, (2.12) simply reduces to the original linear least-squares problem, (1.1), while as α approaches infinity, the optimal solution will tend towards zero. We may rewrite (2.12) as:

$$\min_x \left\| \begin{bmatrix} G \\ \alpha I \end{bmatrix} x - \begin{bmatrix} d \\ 0 \end{bmatrix} \right\|_2^2 . \tag{2.13}$$

If $\alpha \neq 0$, then the last n rows of $\begin{bmatrix} G \\ \alpha I \end{bmatrix}$ are linearly independent (and as stated above, $\alpha = 0$ corresponds to the case when no regularization is occurring). Thus, (2.13) is a full-rank linear least-squares problem, and hence may be solved by means of the normal equations, (1.3):

$$\begin{bmatrix} G^T & \alpha I \end{bmatrix} \begin{bmatrix} G \\ \alpha I \end{bmatrix} x = \begin{bmatrix} G^T & \alpha I \end{bmatrix} \begin{bmatrix} d \\ 0 \end{bmatrix},$$

which simplifies to

$$(G^T G + \alpha^2 I)x = G^T d. \tag{2.14}$$

Note that as this is a full-rank system, the normal equations have a unique solution. Applying the singular value decomposition to G, we may rewrite (2.14) as

$$(V \Sigma^T U^T U \Sigma V^T + \alpha^2 I)x = V \Sigma^T U^T d,$$

which we may rewrite as

$$(V \Sigma^T \Sigma V^T + \alpha^2 I)x = V \Sigma^T U^T d. \tag{2.15}$$

Note that the solution to this system is

$$x_\alpha = \sum_{i=1}^{p} \frac{\sigma_i^2}{\sigma_i^2 + \alpha^2} \frac{u_i^T d}{\sigma_i} v_i.$$

We can demonstrate that this solution is correct by substituting into (2.15):

$$\begin{aligned}
&(V\Sigma^T\Sigma V^T + \alpha^2 I) \textstyle\sum_{i=1}^{p} \frac{\sigma_i^2}{\sigma_i^2+\alpha^2} \frac{u_i^T d}{\sigma_i} v_i \\
&= \textstyle\sum_{i=1}^{p} \frac{\sigma_i^2}{\sigma_i^2+\alpha^2} \frac{u_i^T d}{\sigma_i} (V\Sigma^T\Sigma V^T + \alpha^2 I) v_i \\
&= \textstyle\sum_{i=1}^{p} \frac{\sigma_i^2}{\sigma_i^2+\alpha^2} \frac{u_i^T d}{\sigma_i} (V\Sigma^T\Sigma e_i + \alpha^2 I v_i) \\
&= \textstyle\sum_{i=1}^{p} \frac{\sigma_i^2}{\sigma_i^2+\alpha^2} \frac{u_i^T d}{\sigma_i} (\sigma_i^2 v_i + \alpha^2 I v_i) \\
&= \textstyle\sum_{i=1}^{p} \sigma_i u_i^T d v_i \\
&= V\Sigma^T U^T d.
\end{aligned}$$

Thus, we see that x_α is the unique solution to (2.13). Note the similarity between the formulation of x_α and the best least-squares solution (2.4). The terms in the summation are scaled by a *filter factor*

$$f_i = \frac{\sigma_i^2}{\sigma_i^2 + \alpha^2}.$$

For $\alpha \ll \sigma_i$, $f_i \approx 1$. If $\alpha \gg \sigma_i$, then $f_i \approx 0$. As α increases, the smaller singular values have a smaller effect on the solution obtained. Thus, we may think of this as a continuous version of TSVD regularization. Furthermore, we show in the next lemma that regularization by varying α is equivalent to regularization by varying ϵ (and hence δ) as discussed in the previous section (this result also appears in e.g. [18]).

Lemma 2.2.3 *For each choice of $\alpha \geq 0$, a solution to* (2.12) *is also a solution to* (2.7) *for some unique choice of $\epsilon \in [0, \left\|G^\dagger d\right\|_2]$.*

Proof. First, we consider the case when $\alpha = 0$. In this case, the problem reduces to the unconstrained least squares problem, which is in turn equivalent to the case where $\epsilon = \left\|G^\dagger d\right\|_2$.

We recall that the Lagrangian of (2.7) is

$$\mathcal{L}(x;\lambda) = x^T(G^TG + \lambda I)x - 2d^TGx + d^Td - \lambda\epsilon^2.$$

For any fixed $\lambda \geq 0$, $\mathcal{L}(x;\lambda)$ is a convex function in x. Thus, stationarity of the Lagrangian characterizes primal optimality. Hence optimal solutions are characterized by

$$(G^T G + \lambda I)x = G^T d,$$

for some choice of λ. Recall that a solution to (2.12) for $\alpha > 0$ is uniquely given by the normal equations (2.14). Setting $\lambda = \alpha^2$, we see that a solution x to (2.14) also satisfies stationarity of the Lagrangian of (2.7). Setting $\epsilon = \|x\|_2$, we obtain a unique choice of ϵ for each α. Conversely, it is shown in [31] that the optimal Lagrange multiplier λ is given uniquely. Hence, setting $\alpha = \sqrt{\lambda}$, we obtain a unique choice of α for each ϵ. ■

Example 2.2.4 *Using the same values of G and d from Example 2.1.2, we now apply Tikhonov regularization. We vary our choice of α between 10^{-15} and 10 (recall that the largest and smallest singular values of G are approximately 2.9934 and 3.2294×10^{-16}), respectively. The resulting values of $\|Gx_\alpha - d\|_2$ and $\|x_\alpha\|_2$ appear in Figure 2.2.*

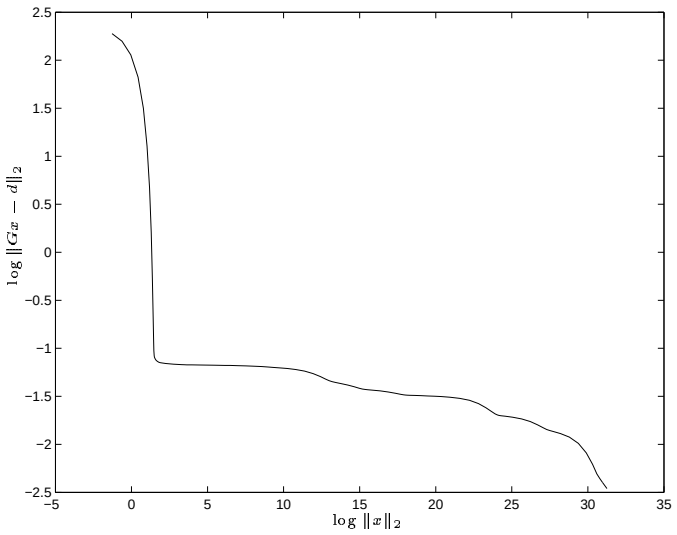


Figure 2.2: Solution points for Example 2.2.4, on a log-log scale.

In theory, we may vary α, until we obtain a solution x_α that satisfies the discrepancy principle. For small problems, this technique works well. As problems become large, the

cost of computing the SVD of G becomes prohibitive and it becomes necessary to employ other techniques to solve (2.12) directly. One such technique is described in Chapter 4.

Chapter 3

Trust Region Subproblem Theory

3.1 TRS Solutions

Recall from Chapter 1 that we wish to consider the Trust Region Subproblem (1.6), formulated as:

$$\begin{aligned} \min \quad & q(s) \\ \text{s.t.} \quad & \|s\|_2 \leq \Delta \\ & s \in \mathbb{R}^n, \end{aligned} \tag{3.1}$$

for some quadratic function q and a trust region radius Δ. In particular, we generally consider q of the form

$$q(s) := \frac{1}{2} s^T A s + a^T s,$$

for some $A \in S^n$ (where S^n is the space of $n \times n$ symmetric matrices) and $a \in \mathbb{R}^n$. Note that the feasible region is a compact set and the objective function is continuous. Hence, we have attainment of the optimum for (1.6). If q is convex, we may attain the optimum in the interior or on the boundary of the trust region. If q is not convex, the minimum will be attained on the boundary. To simplify differentiation, we rewrite (1.6) in the equivalent form

$$\begin{aligned} \min \quad & s^T A s + 2a^T s \\ \text{s.t.} \quad & s^T s \leq \Delta^2 \\ & s \in \mathbb{R}^n. \end{aligned} \tag{3.2}$$

Note that the Lagrangian of (3.2) is

$$\mathcal{L}(s;\lambda) = s^T A s + 2a^T s + \lambda(s^T s - \Delta^2).$$

Thus, s^* is a solution to (3.2) if and only if

$$\begin{array}{cc} \|s^*\| \leq \Delta & \text{(Primal Feasibility)} \\ \left.\begin{array}{c} A + \lambda^* I \succeq 0, \lambda^* \geq 0 \\ \nabla_s \mathcal{L}(s^*;\lambda^*) = 0 = 2As^* + 2a + 2\lambda^* s^* \end{array}\right\} & \text{(Dual Feasibility)} \\ \lambda^*(s^{*T}s^* - \Delta^2) = 0 & \text{(Complementary Slackness),} \end{array} \tag{3.3}$$

for some choice of optimal Lagrange multiplier λ^*. (see e.g. [11], [29]). We rewrite the dual feasibility condition (stationarity of the Lagrangian) as

$$(A + \lambda^* I)s^* = -a.$$

3.1.1 Dual-parametrized solutions for TRS

We can use the optimality conditions (3.3) to characterize the solution to (3.2) by the choice of optimal Lagrange multiplier. In particular, the stationarity of the Lagrangian, given by

$$(A + \lambda^* I)s = -a \tag{3.4}$$

suggests some methods for characterizing an optimal solution. In particular, if $A+\lambda^* I \succ 0$, then $s = (A + \lambda^* I)^{-1}a$ is the unique solution to (3.2) corresponding to λ^*. In general, we define

$$s(\lambda) = -(A + \lambda I)^{\dagger} a.$$

Applying a singular value decomposition to $A + \lambda I$,

$$A + \lambda I = U_p \Sigma_p V_p^T$$

and using the definition of the Moore-Penrose generalized inverse (2.1), we see that

$$\begin{aligned} (A + \lambda I)s(\lambda) &= -U_p \Sigma_p V_p^T \Sigma_p^{-1} U_p^T a \\ &= -U_p U_p^T a. \end{aligned}$$

Recall that the columns of U_p form a basis for $\mathcal{R}(A + \lambda I$. Thus, for some choice of λ, if $a \not\perp \mathcal{N}(A + \lambda I)$, then $s(\lambda)$ is the unique solution to (3.4). However, for $\lambda \neq \lambda^*$, $s(\lambda)$ will violate primal feasibility or complementary slackness (as λ^* is the unique value that satisfies (3.3). See [31].)

Recall that for λ to satisfy dual feasibility, it must satisfy $\lambda \geq -\lambda_1(A)$. If $a \not\perp \mathcal{N}(A - \lambda_1(A)I)$, then for any dual feasible λ, $s(\lambda)$ is the unique solution to the Lagrangian stationarity condition (since for any feasible λ, either $\lambda = -\lambda_1(A)$, or $A + \lambda I \succ 0$). This is known in the literature as the *easy case* (see e.g. [10]). Note that in the easy case, as $\lambda \searrow -\lambda_1(A)$, the smallest singular value of $A + \lambda I$ (which is also the smallest eigenvalue of $A + \lambda I$) will tend to zero. Since the column of U_p corresponding to this singular value is not orthogonal to a, it follows that as $\lambda \searrow -\lambda_1(A)$, $\|s(\lambda)\| \to \infty$. Hence, in the easy case, to maintain primal feasibility, we must have $\lambda^* > -\lambda_1(A)$.

Suppose on the other hand that $a \perp \mathcal{N}(A - \lambda_1(A)I)$. This situation is referred to in the literature as the *hard case* for TRS. In this case, we note that $s(\lambda)$ is still the unique solution to (3.4) for $\lambda > -\lambda_1(A)$. If the optimal Lagrange multiplier λ^* does satisfy $\lambda^* > -\lambda_1(A)$, we refer to this as (case 1) of the hard case (see e.g. [10]). If, however, $\lambda^* = -\lambda_1(A)$, $s(\lambda^*)$ is not the unique solution to (3.4), and hence may not yield an optimal solution to (3.2). We refer to this situation as (case 2) of the hard case. If $\|s(-\lambda_1(A)\|_2 = \Delta$ or $\lambda_1(A) = 0$, then the complementary slackness condition is satisfied, and all optimality conditions hold. (Note that primal feasibility is guaranteed by the fact that $\|s(\lambda)\|_2$ is a decreasing function for $\lambda > -\lambda_1(A)$. If $\|s(-\lambda_1(A))\|_2 > \Delta$, then $\lambda^* > -\lambda_1(A)$ and the hard case (case 1) would hold.) If, on the other hand, $-\lambda_1(A) > 0$ and $\|s(-\lambda_1(A))\|_2 < \Delta$, then the complementary slackness condition is violated. In this case, the optimal solution is $s(-\lambda_1(A)) + d$ for some $d \in \mathcal{N}(A - \lambda_1(A))$ such that $\|s(-\lambda_1(A) + \alpha d\|_2 = \Delta$.

We review the various cases presented above in Table 3.1. (Note that this table is reproduced from [10].)

At this point, we note that, except in the hard case (case 2), subcase (ii), $s(\lambda)$ will yield an optimal solution for (3.2). The remainder of this section is devoted to examples that demonstrate the various cases presented above.

Given a spectral decomposition $U\Lambda U^T$ for A, such that $\Lambda = \text{diag}\,(\lambda_n(A), \lambda_{n-1}(A) \ldots, \lambda_1(A))$

Easy Case	Hard Case (case 1)	Hard Case (case 2)
		$a \perp \mathcal{N}(A - \lambda_1(A)I)$
$a \not\perp \mathcal{N}(A - \lambda_1(A)I)$	$a \perp \mathcal{N}(A - \lambda_1(A)I)$	and $\lambda^* = -\lambda_1(A)$
(implies $\lambda^* > -\lambda_1(A)$)	and $\lambda^* > -\lambda_1(A)$	(i) $\|s(-\lambda_1(A))\|_2 = \Delta$ or $\lambda_1(A) = 0$
		(ii) $\|s(-\lambda_1(A))\|_2 < \Delta$ and $-\lambda_1(A) > 0$

Table 3.1: Different cases for the trust region subproblem.

with $\lambda_1(A) \leq \lambda_2(A) \leq \cdots \leq \lambda_n(A)$, it follows that

$$(A + \lambda I)^\dagger = U_p(\Lambda_p + \lambda I_p)^{-1} U_p^T,$$

where $p+1$ corresponds to the index of the first zero eigenvalue of $A+\lambda I$ in $\Lambda+\lambda I$. Hence, $s(\lambda)$ may be written as

$$s(\lambda) = -\sum_{i=1}^{p} \frac{u_i^T a}{\lambda_i(A) + \lambda} u_i,$$

where u_i corresponds to the i-th column of U. Recall that we must also satisfy $\|s(\lambda)\|_2 \leq \Delta$. Thus, for convenience, we define

$$\psi(\lambda) = \|s(\lambda)\|_2^2 = \sum_{i=1}^{n} \frac{(u_i^T a)^2}{(\lambda_i(A) + \lambda)^2}. \tag{3.5}$$

As we must maintain dual feasibility, we need only consider values of λ such that $\lambda \geq -\lambda_1(A)$ and $\lambda \geq 0$. We have a solution in the interior if $\lambda_1(A) > 0$ and $\psi(0) < \Delta^2$. Otherwise, we attempt to find some feasible λ such that $\psi(\lambda) = \Delta^2$. In all cases except the hard case (case 2) (ii), such a λ will exist. We demonstrate the easy case when $A \succ 0$ in Example 3.1.1, and the easy case when A is indefinite in Example 3.1.2. We then present an example of the hard case in Example 3.1.3.

Example 3.1.1 *Consider the TRS problem given by the following choices of A and a:*

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{bmatrix}, \quad a = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

This example demonstrates the case when A is positive definite. In this case, the solution to (3.4) *is uniquely defined for any feasible λ. We plot $\psi(\lambda)$ versus λ, and show the resulting graph in Figure 3.1. The bold curve represents the possible (λ, Δ^2) pairs that may yield a solution to the problem (i.e. values of (λ, Δ^2) such that complementary slackness, primal feasibility, and dual feasibility would all hold). The bold vertical line occurs at $\lambda = 0$, as complementary slackness allows solutions in the interior of the trust region only when $\lambda = 0$. Thus, if $\Delta^2 > \psi(0)$, the solution $s(0)$ is in the interior. If, however, $\lambda > 0$, the solution must lie on the boundary, and hence we must satisfy $\psi(\lambda) = \Delta^2$.*

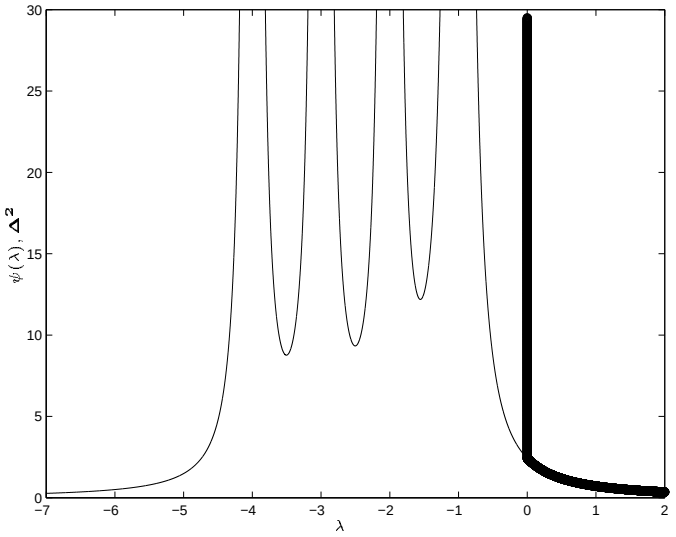


Figure 3.1: Values of $\psi(\lambda)$ and $\Delta^2(\lambda)$ versus λ in Example 3.1.1.

Note that $\psi(\lambda)$ will be discontinuous at each point where $\lambda = -\lambda_i(A)$ for some $i \in \{1, 2, \ldots, n\}$. If $u_i^T a \neq 0$, then $\psi(\lambda) \to \infty$ as $\lambda \to \lambda_i(A)$. This is the easy case, which we present in the following example.

Example 3.1.2 *Consider the TRS problem given by the following choices of A and a:*

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}, \quad a = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

As A is indefinite, our TRS objective is not convex, and hence our solution will lie on the boundary of the trust region (i.e. $\psi(\lambda) = \Delta^2$). Furthermore, to maintain $A + \lambda I$ positive semidefinite, valid solutions must also satisfy $\lambda \geq 1$. The valid (λ, Δ^2) pairs are shown in the bold curve in figure 3.2.

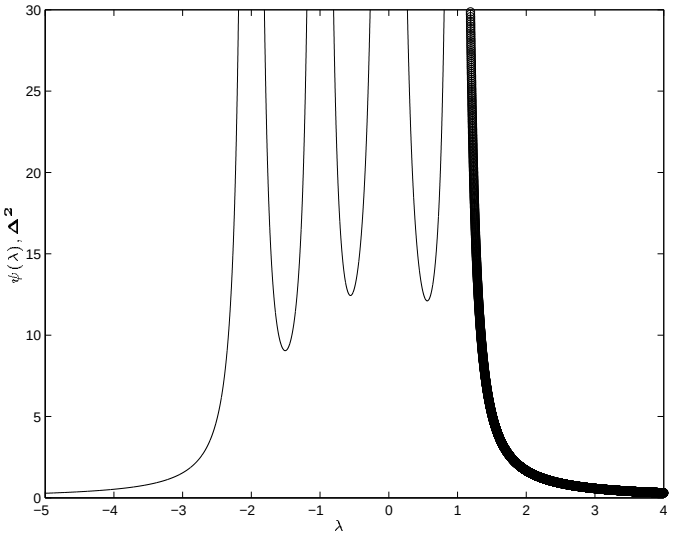


Figure 3.2: Values of $\psi(\lambda)$ and $\Delta^2(\lambda)$ versus λ in Example 3.1.2.

On the other hand, if $u_i^T a = 0$ for all u_i in the eigenspace corresponding to $\lambda_1(A)$, then $\frac{(u_i^T a)^2}{(\lambda_1(A)+\lambda)^2} \to 0$ as $\lambda \to -\lambda_1(A)$. Thus, $\psi(\lambda)$ will not have a pole at $-\lambda_1(A)$. This is the hard case, which we present in the following example.

Example 3.1.3 *Consider the following slight variation on Example 3.1.2:*

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}, \quad a = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Note that in this case, $u_1^T a = 0$, *and* $\lambda_1(A)$ *is a simple eigenvalue. Thus, we have* $\lambda \geq -\lambda_1(A) = 1$, *but* $\psi(\lambda)$ *has no pole at* $\lambda = 1$. *In fact,* $\psi(\lambda)$ *attains a maximum value of approximately* 2.34 *on the domain* $\lambda \geq 1$. *As feasible* λ *must satisfy* $\lambda > 0$, *complementary slackness implies that the solution to TRS must lie on the boundary. If* $\Delta^2 < \lim_{\lambda \to -\lambda_1(A)} \psi(\lambda)$, *a solution to* $\psi(\lambda) = \Delta^2$ *exists for some* $\lambda > -\lambda_1(A)$. *This is the hard case (case 1). In this case,* $s(\lambda)$ *would still be the unique solution to the stationarity condition* (3.4). *If* $\Delta^2 = \lim_{\lambda \to -\lambda_1(A)} \psi(\lambda)$, *we would have* $\|s(-\lambda_1(A)\|_2 = \Delta$, *and the hard case (case 2) (i) would hold. If, however,* $\Delta^2 > \lim_{\lambda \to -\lambda_1(A)} \psi(\lambda)$, *the hard case (case 2) (ii) holds, and* $s(\lambda)$ *is not a valid solution for any feasible* λ *(in particular, there is no feasible* λ *such that* $\psi(\lambda) = \Delta^2$.*) See Figure 3.3 for the graph of* $\psi(\lambda)$ *versus* λ.

We may generalize the ideas presented in Example 3.1.3 to arbitrary problems for which the hard case holds. In particular, there will be some finite critical value

$$\Delta_{cri}^2 := \lim_{\lambda \searrow -\lambda_1(A)} \psi(\lambda)$$

such that for all $\Delta^2 > \Delta_{cri}^2$, there will be no solution to $\psi(\lambda) = \Delta^2$.

3.1.2 Newton's Method for TRS

In addition to the hard case, numerical difficulties may arise when $u_i^T a$ is nearly zero for all i such that $\lambda_i(A) = \lambda_1(A)$. This situation is sometimes referred to as the *near-hard case*. In this case, whereas $\psi(\lambda)$ has a pole at $\lambda = -\lambda_1(A)$, it may be highly nonlinear near this pole. This behaviour may give rise to slow convergence of Newton's method applied to finding a solution to $\psi(\lambda) = \Delta^2$. For example, in Figure 3.4 we present plots of $\psi(\lambda)$

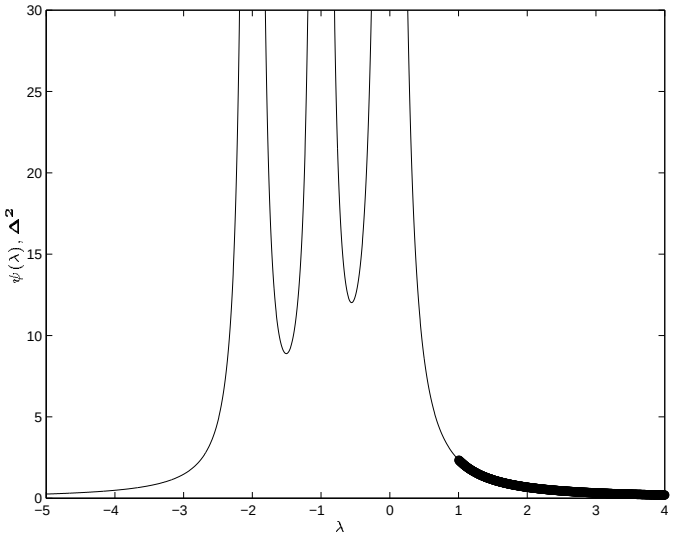


Figure 3.3: Values of $\psi(\lambda)$ and $\Delta^2(\lambda)$ versus λ in Example 3.1.3.

versus λ for modified versions of Example 3.1.2 with different values of a_1. Note that as a_1 becomes smaller, the function becomes increasingly nonlinear near the pole.

An alternative is therefore to apply Newton's method to solving $\phi(\lambda) = 0$), where

$$\phi(\lambda) = \frac{1}{\sqrt{\psi(\lambda)}} - \frac{1}{\Delta} = \frac{1}{\|s(\lambda)\|} - \frac{1}{\Delta}.$$

Note that as $\psi(\lambda)$ was positive for all λ, and had poles at $-\lambda_i(A)$ for each $i \in \{1 \ldots, n\}$, $1/\sqrt{\psi(\lambda)}$ will be nonnegative for all λ, and have roots at $-\lambda_i(A)$. The behaviour of ϕ may be seen in Figure 3.5, which features a plot of $1/\sqrt{\psi(\lambda)}$ versus λ, using the same sets of parameters as Figure 3.4. Some properties of ϕ are described in the following Lemma (which appears in [4]).

Lemma 3.1.4 *If $a \neq 0$, and $\lambda > \lambda_1(A)$, then the function $\phi(\lambda)$ defined above has first and second derivative given by*

$$\phi'(\lambda) = -\frac{s(\lambda)^T \nabla_\lambda s(\lambda)}{\|s(\lambda)\|_2^3}$$

and

$$\phi''(\lambda) = \frac{3\left[(s(\lambda)^T \nabla_\lambda s(\lambda))^2 - \|s(\lambda)\|_2^2 \|\nabla_\lambda s(\lambda)\|_2^2\right]}{\|s(\lambda)\|_2^5}$$

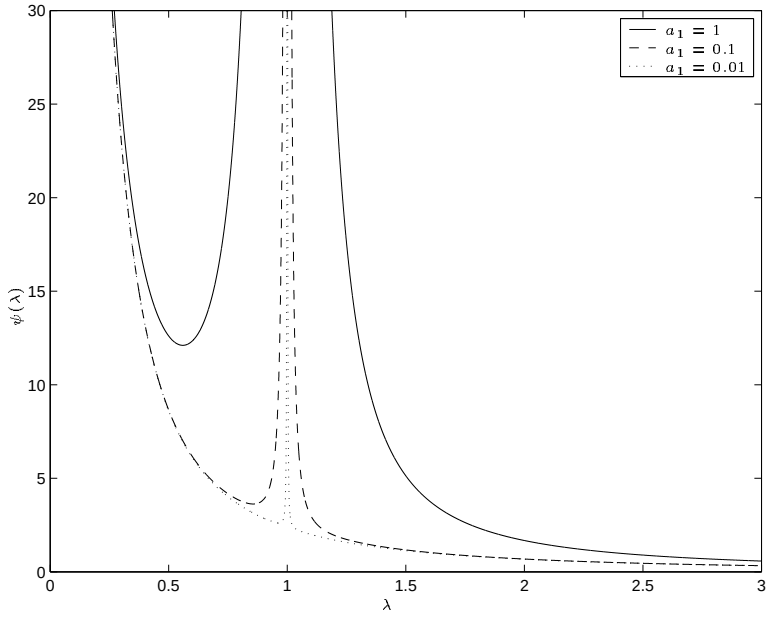


Figure 3.4: Plots of $\psi(\lambda)$ for different values of a_1 in Example 3.1.2.

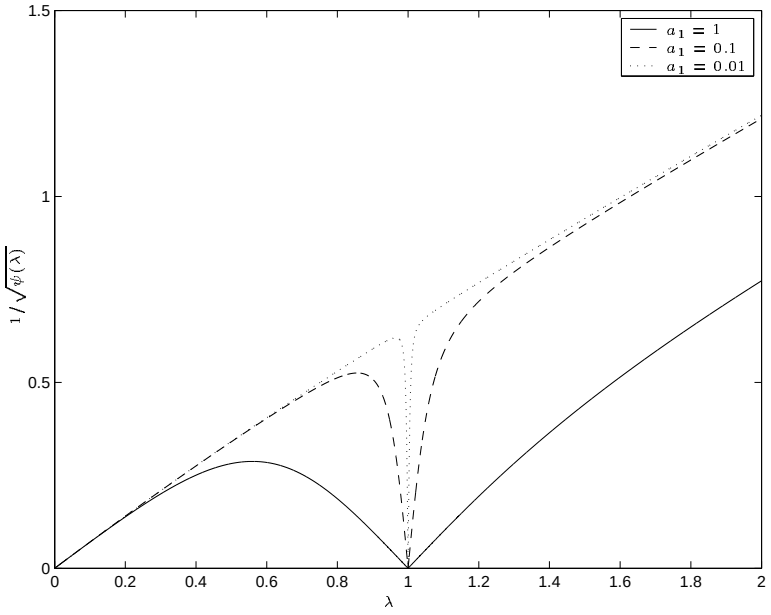


Figure 3.5: Plots of $1/\sqrt{\psi(\lambda)}$ for different values of a_1 in Example 3.1.2.

respectively, where $\nabla_\lambda s(\lambda) = -(A + \lambda I)^{-1} s(\lambda)$.

Proof. First we establish some preliminary results which will be used later in the proof. Recall that

$$(A + \lambda I)s(\lambda) = a.$$

Differentiating both sides of the above equation with respect to λ we get

$$s(\lambda) + (A + \lambda I)\nabla_\lambda s(\lambda) = 0, \tag{3.6}$$

and hence

$$\nabla_\lambda s(\lambda) = -(A + \lambda I)^{-1} s(\lambda).$$

Differentiating both sides of (3.6), we obtain

$$\nabla_\lambda s(\lambda) + \nabla_\lambda s(\lambda) + (A + \lambda I)\nabla^2_{\lambda\lambda} s(\lambda) = 0,$$

and so

$$\nabla^2_{\lambda\lambda} s(\lambda) = -2(A + \lambda)^{-1} \nabla s(\lambda).$$

Taking the l_2 inner product of both sides of this equation with $s(\lambda)$, we obtain

$$\begin{aligned} s(\lambda)^T \nabla^2_{\lambda\lambda} s(\lambda) &= -2s(\lambda)^T (A + \lambda)^{-1} \nabla_\lambda s(\lambda) \\ &= -2\nabla_\lambda s(\lambda)^T (A + \lambda)^{-1} s(\lambda) \\ &= 2\nabla_\lambda s(\lambda)^T \nabla_\lambda s(\lambda) \\ &= 2 \left\| \nabla_\lambda s(\lambda) \right\|_2^2 . \end{aligned} \tag{3.7}$$

Having obtained these preliminary results, we proceed to find $\phi'(\lambda)$:

$$\begin{aligned} \phi'(\lambda) &= \tfrac{d}{d\lambda} \left[(s(\lambda)^T s(\lambda))^{-1/2} - \tfrac{1}{\Delta} \right] \\ &= -\tfrac{1}{2} (s(\lambda)^T s(\lambda))^{-3/2} \tfrac{d}{d\lambda} s(\lambda)^T s(\lambda) \\ &= \tfrac{s(\lambda)^T \nabla_\lambda s(\lambda)}{\|s(\lambda)\|_2^3}. \end{aligned}$$

Differentiating further, we find an intermediate expression for $\phi''(\lambda)$:

$$\phi''(\lambda) = -s(\lambda)^T \nabla_\lambda s(\lambda) \tfrac{d}{d\lambda} \left(s(\lambda)^T s(\lambda) \right)^{-3/2} - \left(s(\lambda)^T s(\lambda) \right)^{-3/2} \tfrac{d}{d\lambda} s(\lambda)^T \nabla_\lambda s(\lambda).$$

Evaluating individual terms in the above expression, we find that

$$\begin{aligned} s(\lambda)^T \nabla_\lambda s(\lambda) \tfrac{d}{d\lambda} \left(s(\lambda)^T s(\lambda)\right)^{-3/2} &= s(\lambda)^T \nabla_\lambda s(\lambda) \left[\tfrac{-3}{2} \left(s(\lambda)^T s(\lambda)\right)^{-5/2} \left(2 s(\lambda)^T \nabla_\lambda s(\lambda)\right)\right] \\ &= -3 \tfrac{\left(s(\lambda)^T \nabla_\lambda s(\lambda)\right)^2}{\|s(\lambda)\|_2^5}, \end{aligned}$$

and (performing the substitution given in (3.7))

$$\begin{aligned} \left(s(\lambda)^T s(\lambda)\right)^{-3/2} \tfrac{d}{d\lambda} s(\lambda)^T \nabla_\lambda s(\lambda) &= \left(s(\lambda)^T s(\lambda)\right)^{-3/2} \left[\nabla_\lambda s(\lambda)^T \nabla_\lambda s(\lambda) + s(\lambda)^T \nabla_{\lambda\lambda}^2 s(\lambda)\right] \\ &= \tfrac{\|s(\lambda)\|_2^2}{\|s(\lambda)\|_2^5} \left[\|\nabla_\lambda s(\lambda)\|_2^2 + 2 \|\nabla_\lambda s(\lambda)\|_2^2\right] \\ &= 3 \tfrac{\|s(\lambda)\|_2^2 \|\nabla_\lambda s(\lambda)\|_2^2}{\|s(\lambda)\|_2^5}. \end{aligned}$$

Hence, we find that

$$\phi''(\lambda) = 3 \frac{\left(s(\lambda)^T \nabla_\lambda s(\lambda)\right)^2 - \|s(\lambda)\|_2^2 \|\nabla_\lambda s(\lambda)\|_2^2}{\|s(\lambda)\|_2^5}.$$

■

Note that in Lemma 3.1.4, we can write $\phi'(\lambda)$ as

$$\phi'(\lambda) = \frac{s(\lambda)^T (A + \lambda I) s(\lambda)}{\|s(\lambda)_2^3\|}.$$

As $(A + \lambda I)$ is positive definite for $\lambda > -\lambda_1(A)$, we see that $\phi(\lambda)$ is strictly increasing. Furthermore, the Cauchy-Schwartz inequality implies that

$$s(\lambda)^T \nabla_\lambda s(\lambda) \leq \|s(\lambda)\|_2 \|\nabla_\lambda s(\lambda)\|_2 ,$$

and so $\phi''(\lambda) \leq 0$. Thus $\phi(\lambda)$ is a concave function.

With the derivative of $\phi(\lambda)$, we can solve for $\phi(\lambda) = 0$ using Newton's method. In particular, given some estimate λ, we find the next iterate as

$$\lambda_{new} = \lambda - \phi(\lambda)/\phi'(\lambda).$$

3.2 Dual Formulations of TRS

To simplify calculations, we multiply the objective value of (3.2) by 2, yielding

$$\begin{array}{ll} \min & s^T A s + 2a^T s \\ \text{s.t.} & \|s\|_2^2 \leq \Delta^2. \end{array} \tag{3.8}$$

The Lagrangian of (3.8) is therefore

$$\begin{aligned} \mathcal{L}(s,\lambda) &= s^T A s + 2a^T s + \lambda(s^T s - \Delta^2) \\ &= s^T(A+\lambda I)s + 2a^T s - \lambda\Delta^2. \end{aligned}$$

The following result was shown in [31].

Theorem 3.2.1 *Strong duality holds for* (3.8)*, i.e.*

$$\mu^* = \min_s \max_{\lambda \geq 0} \mathcal{L}(s,\lambda) = \max_{\lambda \geq 0} \min_s \mathcal{L}(s,\lambda).$$

In particular, it was shown in Section 3.1.1 that a solution to

$$\min_s \mathcal{L}(s,\lambda), \tag{3.9}$$

for some fixed $\lambda \geq 0$, where $\lambda \geq -\lambda_1(A)$ is

$$s(\lambda) = -(A+\lambda I)^\dagger a.$$

Furthermore, if $A+\lambda I$ is singular and $a \not\perp \mathcal{N}(A+\lambda I)$, we may select $d \in \mathcal{N}(A+\lambda I)$ such that $d^T a < 0$. Evaluating $\mathcal{L}(\alpha d, \lambda)$ with $\alpha \to \infty$, we find that the dual functional (3.9) is unbounded. If $A+\lambda I$ is singular and $a \perp \mathcal{N}(A+\lambda I)$ then the dual functional corresponds to $s(\lambda)+d$ for any $d \in \mathcal{N}(A+\lambda I)$. If this occurs for our optimal choice of λ, say λ^* (i.e. $\lambda^* = -\lambda_1(A+\lambda I)$), then consider the sequence $\lambda_k \to \lambda^T *$, with $A+\lambda_k I \succ 0$. In this case, the corresponding $s(\lambda_k)$ will be $-(A+\lambda_k I)^{-1}a$, and $s(\lambda_k) \to s(\lambda)$. Thus, the Lagrangian dual problem of (3.8) becomes

$$(D) \qquad \begin{array}{ll} \mu^* = \sup & h(\lambda) \\ \text{s.t.} & A+\lambda I \succ 0, \\ & \lambda \geq 0, \end{array} \tag{3.10}$$

where

$$\begin{aligned} h(\lambda) &= a^T(A+\lambda I)^\dagger(A+\lambda I)(A+\lambda I)^\dagger a - 2a^T(A+\lambda I)^\dagger a - \lambda\Delta^2 \\ &= -a^T(A+\lambda I)^\dagger a - \lambda\Delta^2. \end{aligned}$$

The strong duality between (3.8) and (3.10) allows us to derive another dual form by homogenizing (3.8) (as in [26]):

$$\begin{aligned} \mu^* &= \min_{\|s\|_2 \le \Delta, y_0^2 = 1} s^T A s + 2y_0 a^T s \\ &= \max_t \min_{\|s\|_2 \le \Delta, y_0^2 = 1} s^T A s + 2y_0 a^T s + t y_0^2 - t \\ &\ge \max_t \min_{\|s\|_2^2 + y_0^2 \le \Delta^2 + 1} s^T A s + 2y_0 a^T s + t y_0^2 - t \\ &\ge \max_{t \in \mathbb{R}, \lambda \ge 0} \min_{s, y_0} s^T A s + 2y_0 a^T s + t y_0^2 - t + \lambda(\|s\|_2^2 + y_0^2 - \Delta^2 - 1) \\ &= \max_{r \in \mathbb{R}, \lambda \ge 0} \min_{s, y_0} s^T A s + 2y_0 a^T s + r y_0^2 - r + \lambda(\|s\|_2^2 - \Delta^2) \\ &= \max_{\lambda \ge 0} \left(\max_r \min_{s, y_0} s^T A s + 2y_0 a^T s + r y_0^2 - r + \lambda(\|s\|_2^2 - \Delta^2) \right) \\ &= \max_{\lambda \ge 0} \min_{s, y_0^2 = 1} s^T A s + 2y_0 a^T s + \lambda(\|s\|_2^2 - \Delta^2) \\ &= \mu^*. \end{aligned}$$

The above uses the substitution $r = t + \lambda$. The last equality follows from the strong duality result from Theorem 3.2.1.

Using the third expression in the above chain, we have a further dual problem to (3.8):

$$\mu^* = \max_t \min_{\|s\|_2^2 + y_0^2 \le \Delta^2 + 1} (y_0, s^T) D(t) \begin{pmatrix} y_0 \\ s \end{pmatrix} - t,$$

where

$$D(t) = \begin{bmatrix} t & a^T \\ a & A \end{bmatrix}.$$

Suppose t^* is the optimal choice of t in the above problem. If $\lambda_1(D(t^*)) > 0$, then $s = 0$ and $y_0 = 0$. Thus, the optimum will occur at the minimum t such that $\lambda_1(D(t)) > 0$. However, this minimum will not be attained, since $\lambda_1(D(\cdot))$ is a continuous function. Thus, we may assume that for an optimal choice of t^*, $\lambda_1(D(t^*)) \le 0$. Under this assumption, we are

assured that for some optimal choice of s and y_0, $\|s\|_2^2 + y_0^2 = \Delta^2 + 1$, and so the dual problem becomes

$$\begin{aligned} \mu^* = \max_t \quad & (\Delta^2+1)\lambda_1(D(t)) - t \\ \text{s.t.} \quad & \lambda_1(D(t)) \leq 0. \end{aligned}$$

By introducing an additional variable λ, we may reformulate this dual problem as the following semidefinite program:

$$(DSDP) \qquad \begin{aligned} \mu^* = \max \quad & -(\Delta^2+1)\lambda - t \\ \text{s.t.} \quad & D(t) + \lambda I \succeq 0 \\ & \lambda \geq 0. \end{aligned} \tag{3.11}$$

We may rewrite (3.11) as a dual semidefinite program in standard form:

$$\begin{aligned} \max \quad & \begin{pmatrix} -1, & -\Delta^2 - 1 \end{pmatrix} \begin{pmatrix} t \\ \lambda \end{pmatrix} \\ \text{s.t.} \quad & -te_1e_1^T - \lambda I + S = D(0) \\ & S \succeq 0 \\ & \lambda \geq 0, \end{aligned}$$

where e_1 is the first standard basis vector, and we apply the fact that $D(t) = D(0) + te_1e_1^T$. Thus, the corresponding primal SDP is

$$\begin{aligned} \min \quad & \operatorname{trace}(D(0)X) \\ \text{s.t.} \quad & \operatorname{trace}(-Xe_1e_1^T) = -1 \\ & \operatorname{trace}(-XI) \leq -\Delta^2 - 1 \\ & X \succeq 0, \end{aligned}$$

which we rewrite in the more convenient form

$$(PSDP) \qquad \begin{aligned} \mu^* = \min \quad & \operatorname{trace}(D(0)X) \\ \text{s.t.} \quad & \operatorname{trace} X \leq \Delta^2 + 1 \\ & X_{11} = 1 \\ & X \succeq 0. \end{aligned} \tag{3.12}$$

We justify the equality in the objective values between (3.12) and (3.11) by observing that (3.11) has a Slater point, and is bounded above. Hence, strong SDP duality holds.

Note that (3.12) is, in fact, a semidefinite relaxation of the homogenized form of (3.8). If we select X given by

$$X = \begin{pmatrix} y_0 \\ s \end{pmatrix} \begin{pmatrix} y_0 & s \end{pmatrix} = \begin{bmatrix} y_0^2 & y_0 s^T \\ y_0 s & ss^T \end{bmatrix},$$

where $y_0^2 = 1$, then we observe that

$$\begin{aligned} \text{trace}\,(D(0)X) &= \text{trace} \begin{bmatrix} y_0 a^T s & a^T s s^T \\ y_0^2 a + As & y_0 a s^T + A s s^T \end{bmatrix} \\ &= y_0 a^T s + \text{trace}\,(y_0 a s^T) + \text{trace}\,(A s s^T) \\ &= s^T A s + 2 y_0 a^T s. \end{aligned}$$

Furthermore, $\text{trace}\, X = 1 + \|s\|_2^2$. As X is defined as the outer product of a vector with itself, we also have that $X \succeq 0$. Hence, any feasible solution of (3.8) maps to a feasible solution for (3.12), and the objective value is preserved. In fact, the following theorem (presented in [26], and reworded here for clarity) shows that optimal solutions to the primal-dual SDP pair can be used to obtain optimal solutions to (3.8) and (3.10).

Theorem 3.2.2 *Suppose that* (λ^*, t^*) *and* $X^* = \begin{bmatrix} 1 & y^{*T} \\ y^* & \bar{X}^* \end{bmatrix}$ *are optimal for* (3.11) *and* (3.12) *respectively. Then* $\mu^* := -\lambda^*(\Delta^2 + 1) - t^* = -\lambda^* \Delta^2 - a^T y^*$, *and* $\|y^*\|_2 \leq \Delta$.

- *If* $\lambda_1(D(t^*))$ *is simple, then* $s^* := y^*$ *is optimal for* (3.8).
- *Otherwise,* $\lambda_1(D(t^*))$ *is not simple. Consider the matrix factorization for* X^* *given by* $X^* = TT^T$, *where* T *is* $(n+1) \times r$ *and full column rank. For every* $v \in \mathbb{R}^r$ *such that* $v \neq 0$ *and* $Tv = \begin{pmatrix} 0 \\ \eta_v \end{pmatrix}$, *we have* $\eta_v \in \mathcal{N}(A - \lambda^* I)$. *Furthermore, setting*

$$\alpha := \frac{\Delta^2 - \|y^*\|_2^2}{y^{*T}\eta_v + sgn(y^{*T}\eta_v)\sqrt{(y^{*T}\eta_v)^2 + (\Delta^2 - \|y^*\|_2^2)}},$$

we have that $s^* := y^* \pm \alpha \eta_v$ *is an optimal solution to* (3.8).

Given x^* defined in the appropriate case above, we have that the unique optimal Lagrange multiplier for s^* in (3.8) *is $\lambda^* = \lambda_1(D(t^*))$. Also, $\tilde{X} := \begin{pmatrix} 1 \\ s^* \end{pmatrix} (1\ s^*)$ is optimal for* (3.12), *and $\begin{pmatrix} 1 \\ s^* \end{pmatrix}$ is an eigenvector for $\lambda_1(D(t^*))$.*

Chapter 4

Regularization using TRS Theory

4.1 Fundamentals

4.1.1 Formulation

Recall from Chapter 2 that our least-squares problem is

$$\min_x \|Gx - d\|_2 ,$$

for some matrix G and vector d. We obtain regularized solutions by minimizing the norm of the solution subject to a constraint on the norm of the residual:

$$\begin{array}{rl} \min_x & \|x\|_2 \\ \text{s.t.} & \|Gx - d\|_2 \leq \delta. \end{array}$$

It was also shown that this formulation was equivalent to

$$\begin{array}{rl} \min_x & \|Gx - d\|_2 \\ \text{s.t.} & \|x\|_2 \leq \epsilon, \end{array}$$

for appropriate choices of δ and ϵ. If we square the objective function and constraint in this last formulation, we obtain

$$\begin{array}{rl} \min_x & \|Gx - d\|_2^2 = x^T(G^TG)x - 2d^TGx + d^Td \\ \text{s.t.} & \|x\|_2^2 \leq \epsilon^2. \end{array}$$

We then subtract $d^T d$ from the objective function (which does not affect the optimality of any given solution). The resulting problem is a special case of the Trust Region Subproblem, with $A = G^T G$ and $a = -G^T d$:

$$\begin{aligned} \mu^* = \min \quad & x^T G^T G x - 2 d^T G \\ \text{s.t.} \quad & \|x\|_2^2 \leq \epsilon^2. \end{aligned} \tag{4.1}$$

Note, in particular that $G^T G \succeq 0$. Hence, the hard case, case (2) (ii) cannot hold. Recall from the dual formulation (3.11), that

$$\begin{aligned} \mu^* = \max \quad & -(\epsilon^2 + 1)\lambda - t \\ \text{s.t.} \quad & D(t) + \lambda I \succeq 0 \\ & \lambda \geq 0, \end{aligned} \tag{4.2}$$

where in this case,

$$D(t) = \begin{bmatrix} t & -d^T G \\ -G^T d & G^T G \end{bmatrix}.$$

This dual achieves optimality for $\lambda^* = -\lambda_1(D(t^*))$. Furthermore, we know by Theorem 3.2.2 that a vector in the eigenspace of $\lambda_1(D(t^*))$, normalized to have first component 1, yields an optimal solution to (4.1).

Let $t^*(\epsilon), \lambda^*(\epsilon)$ be an optimal solution to (4.2) for a given choice of ϵ. We have already shown that λ^* is optimal for the TRS dual program:

$$\begin{aligned} \mu^* = \max \quad & -d^T G (G^T G + \lambda I)^\dagger G^T d - \lambda \epsilon^2 \\ \text{s.t.} \quad & G^T G + \lambda I \succeq 0. \end{aligned}$$

Furthermore, we have established in Chapter 2 that a given choice of $\lambda^* \geq 0$ uniquely yields the parameter α such that $\alpha^2 = \lambda^*$, and an optimal solution x^* to

$$\min x^T G^T G x - 2 d^T G x + \lambda(\|x\|_2^2 - \epsilon^2)$$

is also an optimal solution for

$$\min \|Gx - d\|_2^2 + \alpha^2 \|x\|_2^2 .$$

Hence, for every value of $\lambda \geq 0$, we can uniquely specify the regularization parameter. Note though, that our choice of λ arises as $\lambda = -\lambda_1(D(t))$. Thus, it remains to show that

each choice of t yields a unique value for $\lambda_1(D(t))$. Recall that (4.2) may be written in the unconstrained form:

$$\max_t k(t) = (\epsilon^2 + 1)\lambda_1(D(t)) - t.$$

As this form does attain a finite optimum, stationarity must hold at optimality. In other words,

$$k'(t) = (\epsilon^2 + 1)\frac{d}{dt}\lambda_1(D(t)) - 1 = 0,$$

and hence,

$$\frac{d}{dt}\lambda^*(t) = \frac{d}{dt} - \lambda_1(D(t)) = -\frac{1}{\epsilon^2 + 1} < 0.$$

Note that $k'(t)$ is not defined only when the multiplicity of the smallest eigenvalue of $D(t)$ changes (shown in [26]). However, we will show later in this section that we may restrict our choice of t such that $\lambda_1(D(t)) < 0 \leq \lambda_1(G^T G)$, and hence the multiplicity of $\lambda_1(D(t))$ is constantly 1. Therefore, we see that for every increase in t^*, our optimal choice of λ^* decreases, as does α^* and δ^*, whereas ϵ^* increases. Thus, we can obtain regularized solutions to a linear least squares problem by varying t, and obtaining an eigenvector $y = \begin{pmatrix} 1 \\ x \end{pmatrix}$ corresponding to $\lambda_1(D(t))$ (where the resulting vector x is a regularized least squares solution).

We wish to find appropriate values for t, such that we actually find valid regularized solutions. Observe first that from our unconstrained dual formulation, we have

$$\begin{aligned} k(t^*) &= (\epsilon^{*2} + 1)\lambda_1(D(t^*)) - t^* \\ &= \mu^* \\ &= x^{*T} G^T G x^* - 2d^T G x^* \\ &= \|Gx - d\|_2^2 - d^T d \\ &= \delta^{*2} - d^T d. \end{aligned}$$

Since our regularization parameter δ is bounded below by $\left\|GG^\dagger d - d\right\|_2$, we therefore have that

$$t^* \leq d^T d - \left\|GG^\dagger d - d\right\|_2^2 + (\epsilon^{*2} + 1)\lambda_1(D(t^*)) \leq d^T d - \left\|GG^\dagger d - d\right\|_2^2$$

(where this last inequality follows from the fact that $\lambda_1(D(t^*)) \leq 0$). Hence we obtain an

upper bound on t^*. Note also that we may rewrite this upper bound as:

$$\begin{aligned} d^T d - \left\| GG^\dagger d - d \right\|_2^2 &= d^T d - d^T (GG^\dagger)^T GG^\dagger d + 2d^T GG^\dagger d - d^T d \\ &= -d^T GG^\dagger GG^\dagger d + 2d^T GG^\dagger d \\ &= -d^T GG^\dagger d + +2d^T GG^\dagger d \\ &= d^T GG^\dagger d. \end{aligned}$$

The following result shows that this bound is tight:

Theorem 4.1.1

$$\lim_{t \to d^T GG^\dagger d} \lambda_1(D(t)) = 0.$$

Proof. First, we note that $\lambda_1(D(t)) \geq 0$ if and only if $D(t) \succeq 0$. Taking the Schur complement of $D(t)$ we note that

$$\begin{aligned} D(t) \succeq 0 &\Leftrightarrow G^T G - \tfrac{1}{t} G^T dd^T G \succeq 0 \\ &\Leftrightarrow G^T G^{\dagger T} G^T GG^\dagger G - \tfrac{1}{t} G^T G^{\dagger T} G^T dd^T GG^\dagger G \succeq 0 \\ &\Leftrightarrow G^T (GG^\dagger)^T GG^\dagger G - \tfrac{1}{t} G^T (GG^\dagger)^T dd^T (GG^\dagger) G \succeq 0 \\ &\Leftrightarrow G^T GG^\dagger GG^\dagger G - \tfrac{1}{t} G^T (GG^\dagger) dd^T (GG^\dagger)^T G \succeq 0 \\ &\Leftrightarrow G^T GG^\dagger G - \tfrac{1}{t} G^T (GG^\dagger d)(GG^\dagger d)^T G \succeq 0 \\ &\Leftrightarrow G^T (GG^\dagger - \tfrac{1}{t} (GG^\dagger d)(GG^\dagger d)^T) G \succeq 0. \end{aligned}$$

Now, since all of the columns of $GG^\dagger$ and $(GG^\dagger d)(GG^\dagger d)^T$ are in $\mathcal{R}(G)$, it follows that

$$D(t) \succeq 0 \Leftrightarrow GG^\dagger - \frac{1}{t}(GG^\dagger d)(GG^\dagger d)^T \succeq 0.$$

Now we define an orthonormal basis for $\mathbb{R}^m$ as follows: the first basis element is a vector in the direction of $GG^\dagger d$, and the next $p-1$ elements are constructed to yield an orthonormal basis of size p for $\mathcal{R}(G)$. The remaining $m-p$ basis vectors are constructed to yield our orthonormal basis for $\mathbb{R}^m$. Let U be a suitable change-of-basis matrix for this system. Thus,

$$GG^\dagger - \frac{1}{t}(GG^\dagger d)(GG^\dagger d)^T = U^T \begin{bmatrix} 1 - \frac{d^T GG^\dagger d}{t} & 0 & 0 \\ 0 & I_{p-1} & 0 \\ 0 & 0 & 0_{m-p} \end{bmatrix} U,$$

where 0_{m-p} is the $(m-p) \times (m-p)$ zero matrix. Thus, we have that $\lambda_1(D(t)) \geq 0$ if and only if $t \geq d^T GG^\dagger d$. The limit follows from the continuity of $\lambda_1(D(\cdot))$. ■

We further obtain the following corollary which shows that t is an appropriate choice for a regularization parameter:

Corollary 4.1.2 *Every choice of t in the interval $(-\infty, d^T GG^\dagger d]$ yields a solution x^* which solves*

$$\min \|Gx - d\|_2^2 + \alpha^2 \|x\|_2^2 \tag{4.3}$$

for some unique α on the interval $[0, \infty)$. Furthermore, for each choice of $\alpha^2 \in [0, \infty)$, we can find some $t \in (-\infty, d^T GG^\dagger d]$ such that the solution x^* corresponding to t minimizes (4.3).

Proof. Above, we have shown that t is in one-to-one correspondence with $\lambda_1(D(t))$ if $\epsilon \neq 0$. Thus, for $\epsilon \neq 0$, every choice of $t \leq d^T GG^\dagger d$ yields a unique choice of $\lambda_1(D(t)) \leq 0$. Note that $\lambda_1(D(t)) = -\lambda^* = -\alpha^2$. Hence, if $\epsilon \neq 0$, every choice of t in the relevant interval yields a unique choice of $\alpha^2 \geq 0$. Note that $\epsilon \to 0$ as $\alpha^2 \to \infty$. Note, though that $\alpha^2 \to \infty$ only when $\lambda_1(D(t)) \to -\infty$, which only occurs as $t \to -\infty$. Thus, this one-to-one correspondence holds for every choice of $t \in (-\infty, d^T GG^\dagger d]$.

Note that on the domain $(-\infty, d^T GG^\dagger d]$, $\lambda_1(D(\cdot))$ is a monotonically increasing function which we have already shown maps to values between $-\infty$ and 0. Thus, we may conclude that $\lambda_1(D(\cdot))$ is an isomorphism mapping $(-\infty, d^T GG^\dagger d]$ to $(-\infty, 0]$. Hence, we may define an isomorphism $\alpha^2(\cdot) := -\lambda_1(D(\cdot))$ which maps $(-\infty, d^T GG^\dagger d]$ to $[0, \infty)$. ■

Thus, we observe that t is an appropriate choice for a regularization parameter. At this point, we present a brief review of the regularization parameters we have examined so far:

4.1.2 Methods

The L-curve for a least-squares problem may be found by obtaining the minimum eigenvalue (and corresponding eigenvector) of $D(t)$ for many values of t. However, particularly given a large matrix G, the eigensolver may take a considerable time to converge. Thus, it

Parameter	**Valid Domain**	**Behaviour as t increases**
t	$\left(-\infty, d^T GG^\dagger d\right]$	increases
α^2	$[0, \infty)$	decreases
λ	$[0, \infty)$	decreases
ϵ	$\left[0, \left\|G^\dagger d\right\|_2\right]$	increases
δ	$\left[\left\|GG^\dagger d - d\right\|_2, \|d\|_2\right]$	decreases

Table 4.1: Summary of regularization parameters.

is advantageous to select values of t which are close to the corner of the L-curve, as this is presumed to be the region of interest (as explained in Chapter 2). We use various heuristics to attempt to isolate the corner of the L-curve, preferably minimizing the number of eigenvalue problems which must be solved.

We have the following preferences for the points we compute:

1. we should compute as few points to the left of the corner of the L-curve as possible;
2. we should compute as few points to the right of the corner of the L-curve as possible;
3. we should compute points as near the corner of the L-curve as possible;
4. we should compute enough points that the shape of the L-curve is clear.

An algorithm which implements these techniques is presented as a flowchart, with corresponding MATLAB code, in Appendix A.

Finding a sufficiently low starting value for t

Assuming we have some estimate $\|\tilde{e}_d\|$ on the norm of the measurement error $d - d_{true}$, we may assume that points to the left of the corner of the L-curve are those with residual greater than $\|\tilde{e}_d\|$. With $\delta \geq \|\tilde{e}_d\|$,

$$t_{low} \leq d^T GG^\dagger d + \lambda_1(D(t_{low}))(\epsilon(t_{low})^2 + 1) - \|\tilde{e}_d\|^2 . \tag{4.4}$$

As $\lambda_1(D(t_{low}))$ and $\epsilon(t_{low})$ are dependent on t_{low}, and $\lambda_1(D(t))(\epsilon(t)^2+1) < 0$ for $t < d^T GG^\dagger d$, it is not possible to guarantee that a given choice of t will satisfy (4.4). Thus, we initially set

$$t = d^T GG^\dagger d - \beta \left\|\tilde{e}_d\right\|^2,$$

for some value $\beta \geq 1$. Using this choice of t, we may then find the minimum eigenpair for $D(t)$, and also determine $\epsilon(t)$ and $\delta(t)$. If $\delta(t) \geq \|\tilde{e}_d\|$ then we use this choice of t as our starting value, t_low. Otherwise, set

$$t' = d^T GG^\dagger d - \beta(\frac{\|\tilde{e}_d\|}{\delta(t)})[\|\tilde{e}_d\|^2 - \lambda_1(D(t))(\epsilon(t)^2+1)].$$

Now we compute the minimum eigenpair for $D(t')$, and repeat this process until we find a suitable choice of t_{low}.

Updating t

We establish some step size s. A reasonable choice is to initially set

$$s = \frac{d^T GG^\dagger d - t_{low}}{m},$$

for some positive integer m. In general, we update $t \leftarrow t+s$. Note that we likely want more than m points for our graph of the L-curve, but initially want to take larger steps (since small changes in t only result in large changes to solutions once t is close to $d^T GG^\dagger d$). If $t+s \geq d^T GG^\dagger d$, then before updating t, we update $s \leftarrow s/h$, for some $h > 1$. Note that as t approaches $d^T GG^\dagger d$, $\lambda_1(D(t))$ approaches zero. As the smallest eigenvalue of $G^T G$ may be zero or numerically indistinguishable from zero, sufficiently small values of $\lambda_1(D(t))$ may correspond to near hard-case solutions. In the near-hard case, our eigenvalue solver may not converge, or may yield inaccurate solutions. Thus, if $\lambda_1(D(t)) > -\iota$ for some small choice of ι (within two or three orders of magnitude of the machine epsilon), we are too close to $d^T GG^\dagger d$. In this case, we revert to our previous choice of t, and decrease the step size.

Focusing on the corner of the L-curve

Suppose we update t as $t' := t + s$, for some step size s. We associate the points to the left of the corner of the L-curve with small changes in $\epsilon(t)$ with each update. To the right

of the corner, small changes in t will result in large changes in $\epsilon(t)$. Thus, we fix some threshold $\eta > 1$, and assume that if

$$\epsilon(t') > \eta\epsilon(t),$$

then t' describes a point to the right of the corner of $L-curve$. Thus, we discard t', update $s \leftarrow s/k$ for some integer k, and find a new $t' := t + s$.

4.2 Discussion

As mentioned in Section 4.1.1, the norm-constrained least squares problem is a special case of TRS. The particular structure of the problem allows us to avoid some of the more difficult aspects of TRS, discussed in Chapter 3.

First, as mentioned in Section 4.1.1, we recall that $G^T G \succeq 0$, and hence the hard case, case (2) (ii) cannot hold. Furthermore, for any choice of t such that $\lambda_1(D(t)) < 0$, it follows that $G^T G - \lambda_1(D(t)) \succ 0$.

Chapter 5

Numerical Results and Applications

5.1 A Simple Example

Example 5.1.1 *Recall the Shaw problem presented in Examples 2.1.2 and 2.2.4. We now solve this problem using our dual TRS technique described in Chapter 4. Recall that we our expected error is* $\sqrt{20}/10$. *Thus, we use this as an estimate for the norm of the residual at the corner of L-curve. We set our parameters for the algorithm as:* $i_{max} = 15$, $\eta = 1.25$, $h = 5$, $k = 5$, *and* $\beta = 1.5$. *The norm and residual for the* 15 *computed solutions are shown in Table 5.1, and are graphed on a log-log scale in Figure 5.1.*

5.2 Large Sparse Least-squares Problems

A major benefit of the algorithm presented in Chapter 4 is the fact each iteration involves the solution of a single eigenvalue problem, a task which requires only a series of matrix-vector multiplications, and which can fully exploit sparsity in the matrix G. Additionally at each step the algorithm uses the previously computed eigenvector as an estimate for the new eigenvector. Given sufficiently small changes in t, the difference between consecutive solutions is small, making the estimate quite accurate. Thus, the algorithm is suitable for regularizing large-scale sparse problems efficiently.

Point	$\|Gx_\dagger^l - d\|$	$\|x_\dagger^l\|$
1	4.9708e-01	4.1679e+00
2	4.5072e-01	4.2146e+00
3	4.0903e-01	4.2644e+00
4	3.7452e-01	4.3187e+00
5	3.5019e-01	4.3830e+00
6	3.4651e-01	4.4005e+00
7	3.4277e-01	4.4259e+00
8	3.3738e-01	4.4920e+00
9	3.3560e-01	4.5271e+00
10	3.3320e-01	4.5893e+00
11	3.2967e-01	4.7222e+00
12	3.2404e-01	5.0998e+00
13	3.2258e-01	5.2531e+00
14	3.2103e-01	5.4581e+00
15	3.1946e-01	5.7400e+00

Table 5.1: Values of $\|Gx - d\|_2$ and $\|x\|_2$ for Example 5.1.1.

5.2.1 Deblurring noisy images

Image deblurring is an example of a large, sparse regularization problem which occurs in the real world. In the examples we present here, the images are two-dimensional grayscale figures. In order to perform operations on them, we must represent these images as vectors.

We begin with a sample 100×100 image generated by the "blur" command in P.C. Hansen's regularization tools package ([17]). This image is shown in Figure 5.2.

This image is converted to a vector by column-stacking, i.e. if the image is given by the matrix $X = (x_1, x_2, \ldots, x_{100})$, where $x_i \in \mathbb{R}^{100}$ describes the brightness of the pixels in

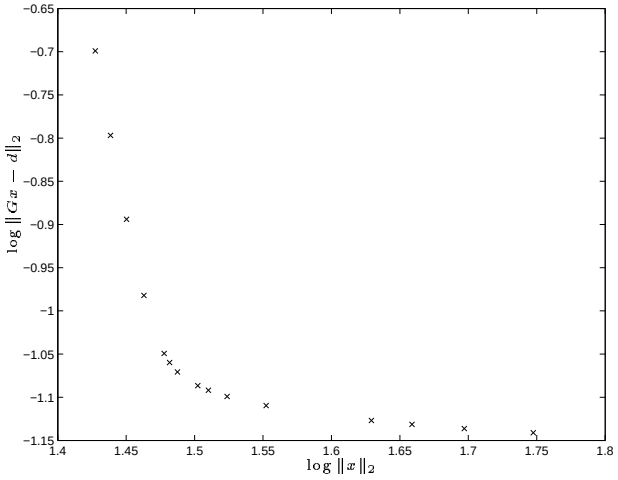


Figure 5.1: Points computed by dual regularization algorithm for Example 5.1.1.

the i-th column of the image, then the resulting vector is:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{100} \end{bmatrix}.$$

Note that $x \in \mathbb{R}^{10000}$. The "blur" command creates a 10000×10000 blur matrix G, as follows: given a blur bandwidth b and a smoothness parameter σ, construct a vector $z \in \mathbb{R}^{100}$ by

$$z_i = \begin{cases} e^{-(i-1)^2/2\sigma^2} & 1 \leq i \leq b \\ 0 & b < i \leq 100 \end{cases}.$$

Next, let A be the 100×100 symmetric Toeplitz matrix given by

$$A = \begin{bmatrix} z_1 & z_2 & \dots & \dots & \dots & \dots & z_{100} \\ z_2 & z_1 & z_2 & z_3 & \dots & \dots & z_{99} \\ z_3 & z_2 & z_1 & z_2 & z_3 & \dots & z_{98} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ z_{100} & z_{99} & z_{98} & \dots & \dots & \dots & z_1 \end{bmatrix}.$$

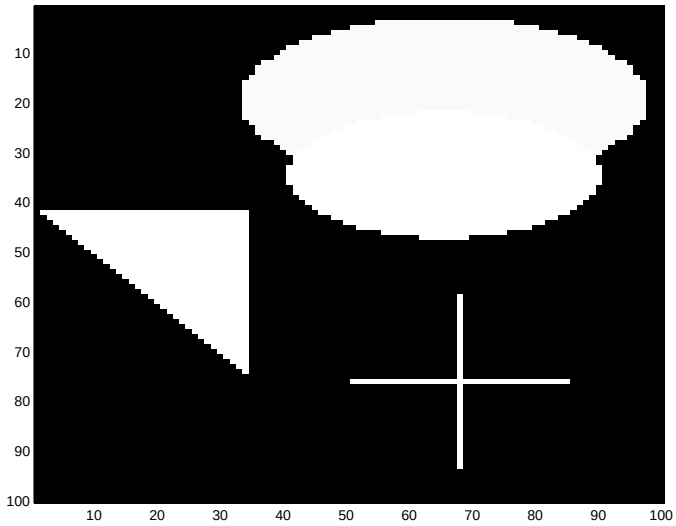


Figure 5.2: Original image for deblurring example.

Let $G = A \otimes A$, where $\otimes$ denotes the Kronecker product. We then compute the blurred image d_{true} as $Gx = d_{true}$. For our example, we use a bandwidth of 5 and a σ-value of 5. The blurred image is shown in Figure 5.3.

We then construct $d = d_{true} + e_d$, where $e_d \in \mathbb{R}^{10000}$ is a vector of normally-distributed random values, generated with mean 0 and standard deviation 0.05. Thus, the expected value of $\|e_d\|$ is 5. The resulting blurred image with this added noise is shown in Figure 5.4.

Next, we apply the algorithm from chapter 4. To demonstrate overly smoothed solutions, we set $\|\tilde{e}_d\| = 6$. Other parameters are $\beta = 2$, $\eta = 1.25$, $h = 5$, $k = 5$, and $i_{max} = 10$. The resulting L-curve is shown in Figure 5.5. The t-values used, as well as resulting norms and residual norms, are shown in Table 5.2.1. The images corresponding to some of computed solutions are shown in Figures 5.6 through 5.12.

5.3 Applications

For the problem in Example 5.1.1, it would be simpler to compute the full singular value decomposition of G and apply Tikhonov regularization (as we did in Example 2.2.4). In

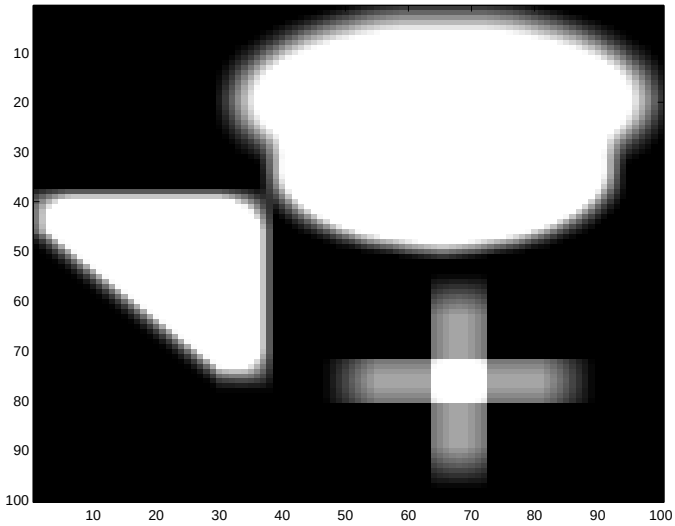


Figure 5.3: Blurred image for deblurring example.

this case, each point could be computed by summing a collection of vectors, based on the filter factors given by our regularization parameter α. Assuming a full singular value decomposition is available, it is not efficient to solve an eigenvalue problem at each iteration.

However, as shown in Section 5.2.1, our eigenvalue-based approach is able to produce regularized solutions to large least-squares problems in a matter of minutes, using consumer hardware. Computing a full singular value decomposition for a 10000×10000 matrix is not currently feasible, and hence applying Tikhonov regularization by filter factors is not possible. The eigenvalue-based approach is also able to exploit sparsity in G. Hence, the algorithm appears well-suited to large sparse problems.

5.4 Limitations

5.4.1 Performance

Whereas the algorithm provides a means for computing solutions to large sparse linear least-squares problems, its performance is poor compared to other modern techniques for solving these problems. For example, the curve in Figure 5.5 required approximately 20

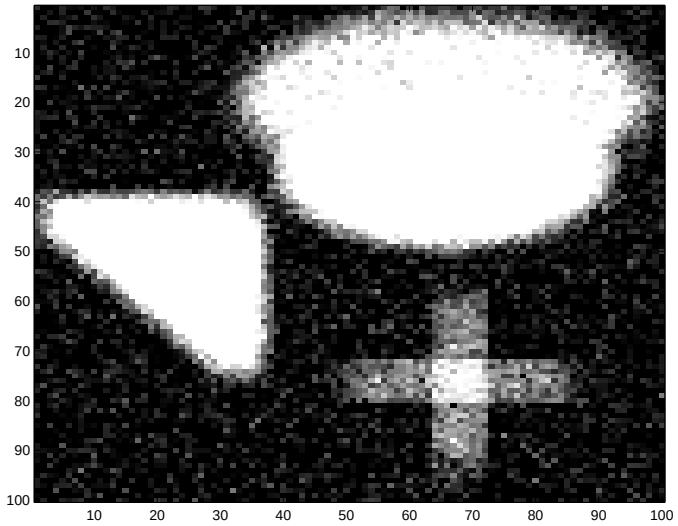


Figure 5.4: Blurred image for deblurring example, with noise added.

minutes to compute in MATLAB on an AMD Athlon XP-2000+ (including a 10 minute calculation to find a point that was far to the right of the corner of the curve, which was discarded and resulted in a reduction in step size; see Section 4.1.2). By comparison, the curve in Figure 5.13, with 50 points, was computed with Hansen's MATLAB implementation of the LSQR algorithm (see [17] and [24]) in under 2 seconds on the same machine.

In particular, for larger values of t, solving for the eigenpair for $\lambda_1(D(t)$ seems to require more time (based on experimental observations). Furthermore, for values of t which yield solutions far to the right of the corner of the L-curve, the eigenvalue calculation takes much longer, while the points obtained are simply discarded by the algorithm and the step size is reduced.

5.4.2 Finding an appropriate regularization parameter

The algorithm presented makes use of the discrepancy principal to find a suitable lower bound on the values of t. We find a value of t such that $x(t)$ satisfies $\|Gx(t) - d\|_2 > \|\tilde{e}_d\|_2$ for the expected error $\tilde{e}_d$. In practice, for many applications, the variance of the noise is not known (see e.g. [9]). In theory, one could select an arbitrarily large error estimate, and compute points until the shape of the L-curve begins to appear. The eigenvalue-based

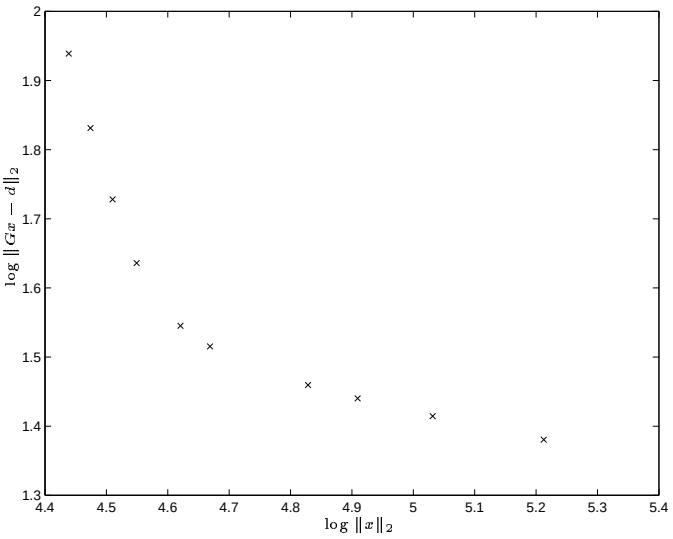


Figure 5.5: Points computed by dual regularization algorithm for deblurring example.

regularization algorithm appears to compute points to the left of the corner of the L-curve more efficiently.

Currently, the only stopping criterion for the algorithm is achieving the maximum number of iterations, i_{max}. It is assumed that the first iteration begins near the corner of the L-curve, on the left side. Furthermore, the heuristic which restricts the change in solution norm between subsequent iterations intuitively keeps computed points near the corner. However, as values of t approach the upper bound, the cost of each iteration appears to increase. Thus, it would be desirable to identify the optimal regularization parameter without spending very many iterations to the right of the L-curve.

Several techniques exist to identify the optimal regularization parameter. We have already discussed the discrepancy principal. Another technique is that of *generalized cross-validation*, or GCV (see [5] and [12]). Applying GCV to our eigenvalue-based regularization algorithm, we would select t to minimize

$$\frac{\|Gx(t) - d\|_2^2}{(\mathcal{T}(t))^2},$$

where

$$\mathcal{T}(t) = \text{trace}\,(I - G(G^T G + \lambda_1 (D(t))^2 I)^{-1} G^T).$$

Point	t	$\|Gx(t)-d\|_2$	$\|x(t)\|_2$	$\log\|Gx(t)-d\|_2$	$\log\|x(t)\|_2$	time (s)
1	1170.6	6.9514	84.6676	1.9389	4.4387	7.62
2	1209.6	6.2417	87.7257	1.8313	4.4742	7.59
3	1248.6	5.6300	90.9315	1.7281	4.5101	12.18
4	1287.5	5.1345	94.5645	1.6360	4.5493	12.14
5	1326.5	4.6894	101.5635	1.5453	4.6207	25.73
6	1334.3	4.5516	106.5735	1.5155	4.6688	34.8
7	1342.1	4.3045	125.0357	1.4597	4.8286	71.1
8	1343.7	4.2213	135.5584	1.4401	4.9094	89.17
9	1345.2	4.1149	153.1565	1.4146	5.0315	125.41
10	1346.8	3.9771	183.5408	1.3806	5.2124	193.52

Table 5.2: Result data for deblurring example.

The computation of $\mathcal{T}(t)$ generally involves the construction of a full matrix, which in the case of large sparse G causes computational difficulties.

The technique for finding an optimal regularization parameter presented throughout this work, albeit intuitively, is the idea of identifying the "corner" of the L-curve. Hansen and O'Leary (see [16]) defined the "corner" as the point on the L-curve with maximum curvature. The L-curve criterion for selecting the regularization parameter is often more robust that the GCV method when dealing with correlated errors (see [15]).

5.4.3 Potential future work

As mentioned in the previous section, the LSQR method appears to find a collection of regularized solutions much faster than our eigenvalue-based regularization algorithm. However, the LSQR method used did not provide a choice of points to compute. Thus, it may be worthwhile to first compute a discrete sketch of the L-curve using LSQR, and then use the eigenvalue-based approach near the corner of the L-curve, in the hopes of finding a better solution (as the eigenvalue approach, like Tikhonov regularization, can compute any point on the L-curve).

Furthermore, there may be some value in adding to the algorithm the capability to

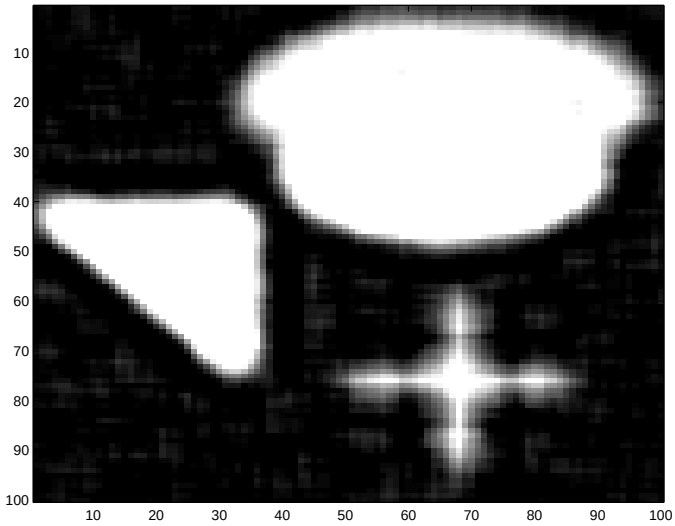


Figure 5.6: Deblurred solution corresponding to $t = 1170.6$.

allow it to compute the curvature of the L-curve at each point. In theory, the algorithm could use a binary search, or some other search technique, to find the point of maximum curvature to a high accuracy.

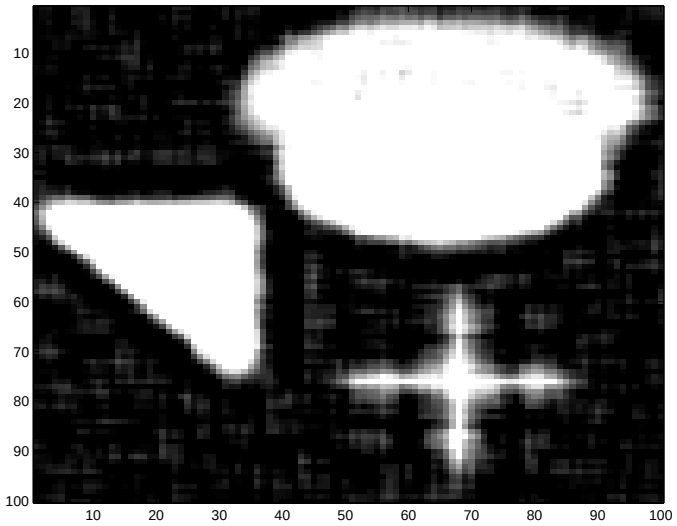


Figure 5.7: Deblurred solution corresponding to $t = 1248.6$.

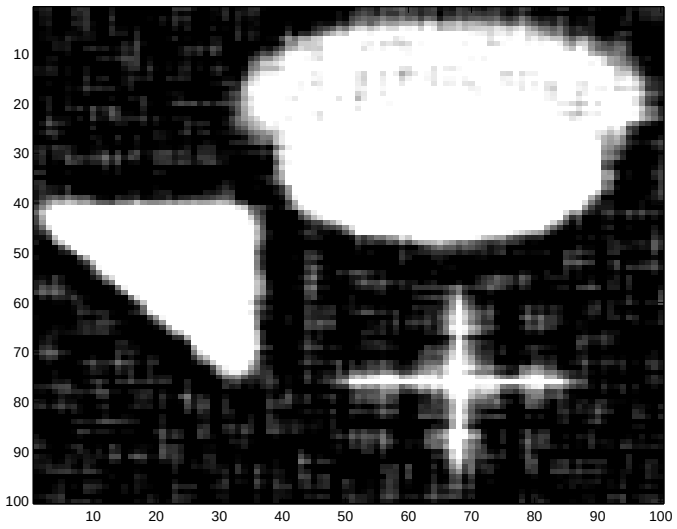


Figure 5.8: Deblurred solution corresponding to $t = 1287.5$.

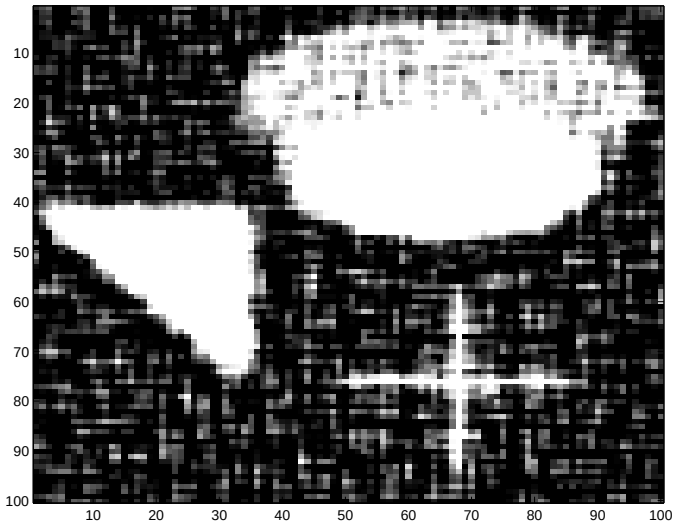


Figure 5.9: Deblurred solution corresponding to $t = 1326.5$.

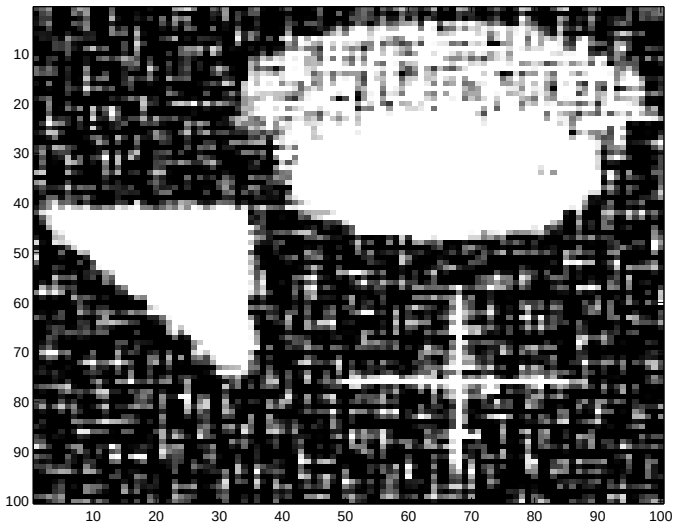


Figure 5.10: Deblurred solution corresponding to $t = 1334.3$.

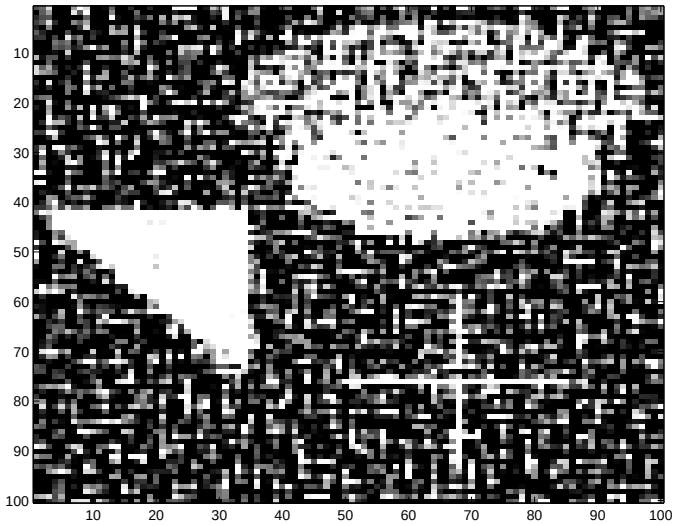


Figure 5.11: Deblurred solution corresponding to $t = 1342.1$.

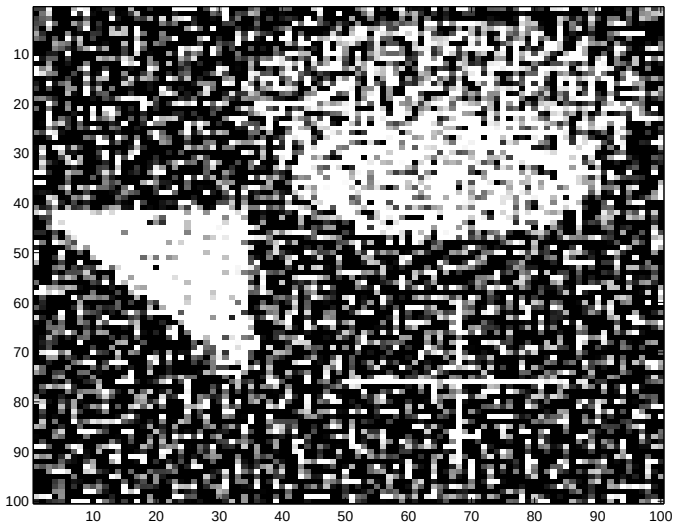


Figure 5.12: Deblurred solution corresponding to $t = 1345.2$.

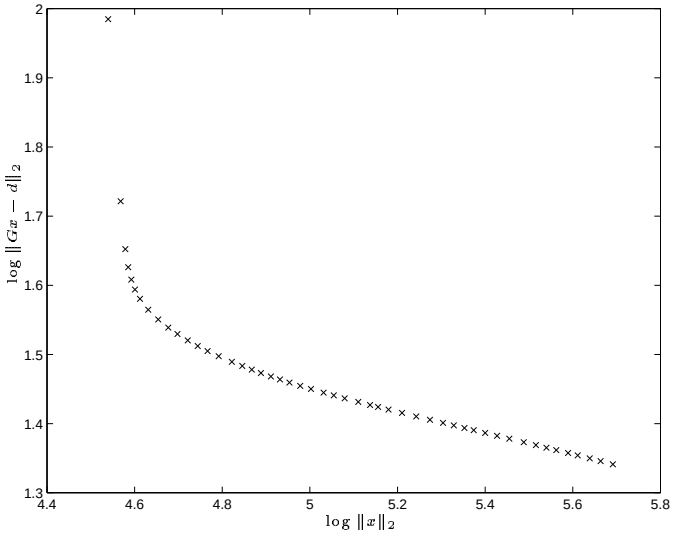


Figure 5.13: Points computed by LSQR for deblurring example.

Appendix A

A Regularization Algorithm

MATLAB code for newlplot.m

```
function newlplot(G,d,err_est,beta,tlow,tup)
global x counter;

ub = d'*d;
%ub = d'*(G*(G\d));
n = size(G,2);
eigenvals = [];
eigenvecs = [];
xvals = [];
normx = [];
resid = [];
tvals = [];
lambdas = [];
counter=0;
%OPTIONS.tol= 100*eps; % need accuracy here    was eps ??????
OPTIONS.tol= 1e-7; % need accuracy here    was eps ??????
OPTIONS.disp = 0; % no display of the output for eigs.m
OPTIONS.maxit = max(500,n);
```

```
OPTIONS.issym = 1;
starttime = cputime;
Dhandle2 = @Dfun2;
Dhandle = @Dfun;
if nargin > 4,
    lb = tlow;
    ub = tup;
    stepsize = (ub-lb)/5;
else
lb = ub - beta*err_est^2;
%OPTIONS.v0 = ones(n+1,1);
fixflag=1;
% Try initial iteration to see how closely lb matches desired behaviour
while fixflag,
    inittime = cputime;
    [va,lambda,flageigs]=eigs(Dhandle,n+1,1,'SA',OPTIONS,G,d,lb);
    disp(['Preliminary eigenvalue calculation time: ', ...
        num2str(cputime-inittime)]);
    vatemp = va*(1/va(1));
    xval = vatemp(2:n+1);
    curresid = norm(G*xval-d);
    if curresid < err_est,
        oldlb = lb;
        %lb = ub - beta*(err_est^2/curresid^2)*(ub-lb);
        % Store current norm and residual anyway - Free points!
        normx = [norm(xval) normx];
        resid = [curresid resid];
        xvals = [xval xvals];
        tvals = [lb tvals];
        lb = ub - beta*(err_est/curresid)*(err_est^2 - ...
            lambda*(norm(xval)^2+1));
```

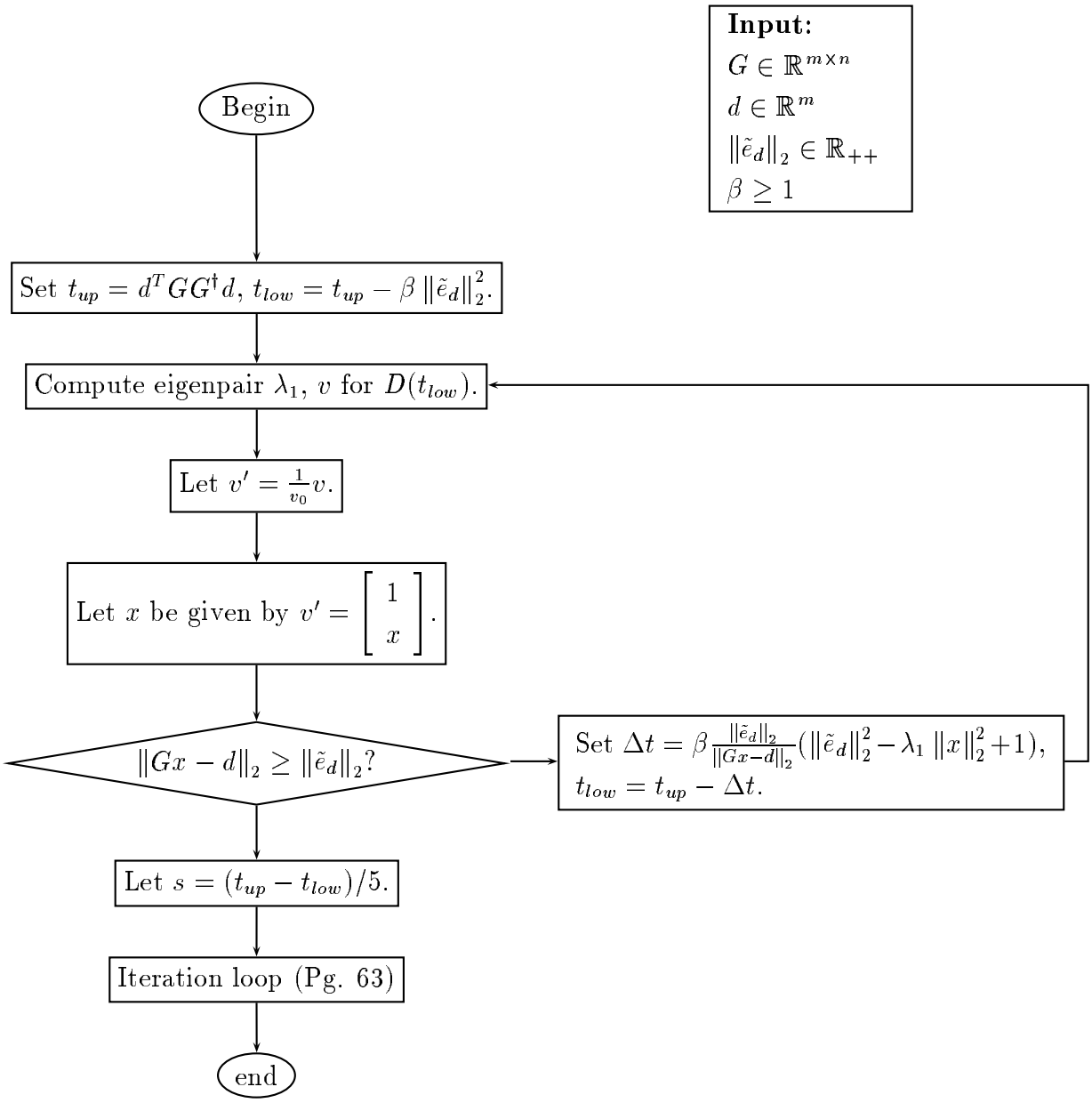
```
        disp(['Reestimated lower bound from ', ...
            num2str(oldlb),' to ',num2str(lb)]);
    else
        fixflag =0;
    end
    OPTIONS.v0 = va;
end
eigenvals = [lambda eigenvals];
eigenvecs = [va eigenvecs];
vatemp = va*(1/va(1));
xval = vatemp(2:n+1);
xvals = [xval xvals];
normx = [norm(xval) normx];
resid = [norm(G*xval-d) resid];
tvals = [lb tvals];
lambdas = [lambda lambdas];
stepsize = (ub-lb)/5;
end
i = 1;
Dt = lb + stepsize;
while i<10 & stepsize > 1000*eps,
    curcounter = counter;
    eigtime = cputime;
    [va,lambda,flageigs]=eigs(Dhandle,n+1,1,'SA',OPTIONS,G,d,Dt);
    eigenvals = [lambda eigenvals];
    eigenvecs = [va eigenvecs];
    vatemp = va*(1/va(1));
    xval = vatemp(2:n+1);
    if (flageigs==0) & lambda < -OPTIONS.tol & ...
        testnorm(norm(xval),normx,1.25),
        disp(['Got point ', num2str(i), ...
```

```
            ' - Eigs time:', num2str(cputime-eigtime), ...
            ' Dfun calls: ',num2str(counter-curcounter), ...
            ' Dt: ', num2str(Dt), ...
            ' Norm(x): ', num2str(norm(xval)), ...
            ' Resid: ', num2str(norm(G*xval-d))]);
          OPTIONS.v0 = va;
        xvals = [xval xvals];
        normx = [norm(xval) normx];
        resid = [norm(G*xval-d) resid];
        tvals = [Dt tvals];
        lambdas = [lambda lambdas];
        Dt = Dt + stepsize;
        if Dt < ub-.2*stepsize,
            i = i+1;
        else
            Dt = Dt - stepsize;
            stepsize = stepsize/5;
            disp(['Too close to boundary, reduced stepsize to ',...
                num2str(stepsize)]);
            Dt = Dt + stepsize;
            i = i+1;
        end
    else
        Dt = Dt-stepsize;
        stepsize = stepsize/5;
        disp(['Reduced stepsize to ',num2str(stepsize), ...
            '   Eig time: ', num2str(cputime-eigtime)]);
        Dt = Dt + stepsize;
    end
end
disp(['Running time: ', num2str(cputime - starttime)]);
```

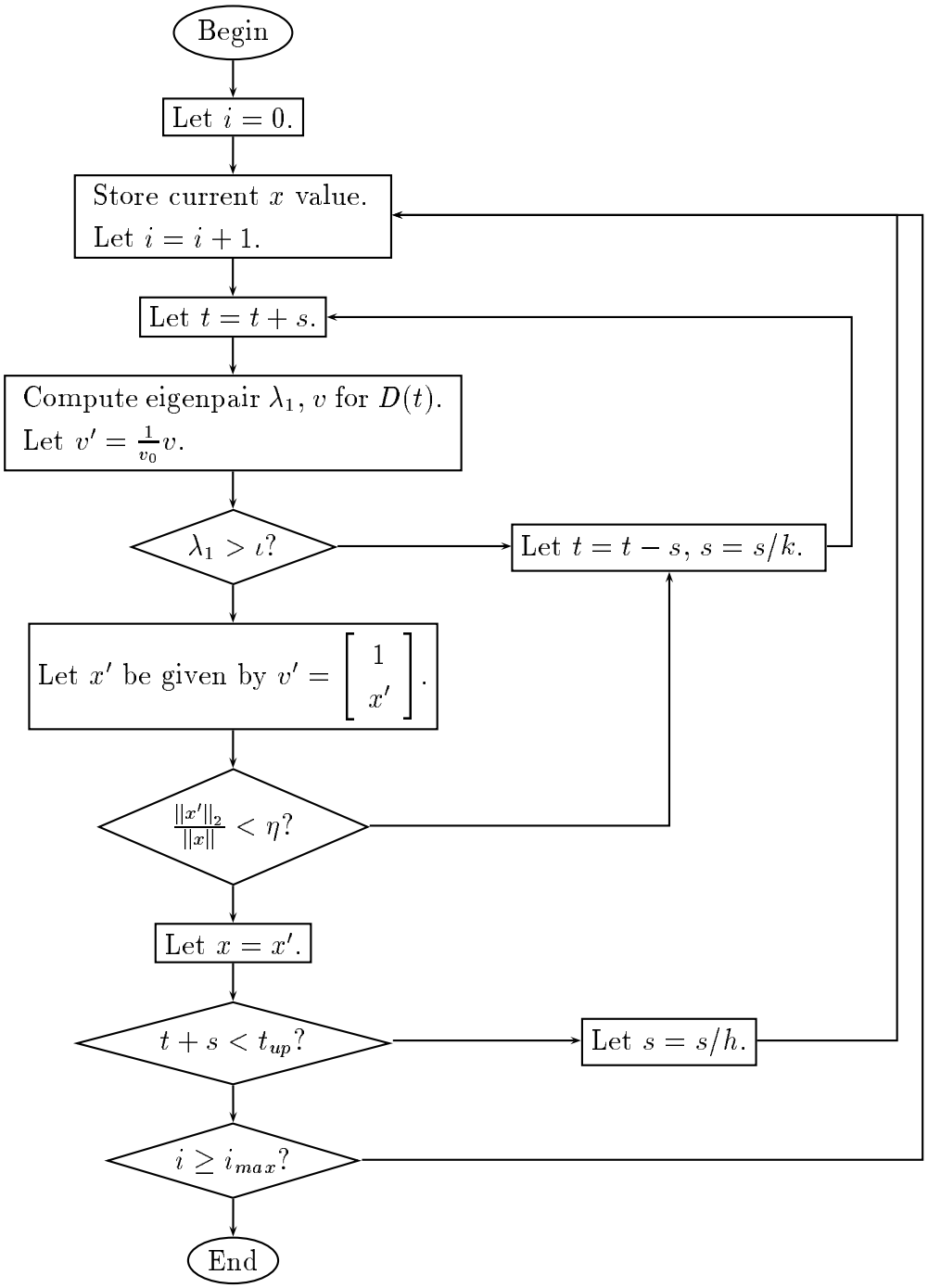
```
plot(log(normx),log(resid),'x');
keyboard;

function val=testnorm(curnorm,oldnorms,threshold)
if size(oldnorms,2) == 0,
    val = 1;
else
    if curnorm/oldnorms(1) > threshold,
        disp(['Previous norm: ', num2str(oldnorms(1)), ...
            ', Current norm: ', num2str(curnorm), ...
            ' -- t-value probably near corner of L-curve']);
        val = 0;
    else
        val = 1;
    end
end
```

(* LaTeX generated by highlight 2.0-13, http://www.andre-simon.de/ *)

Regularization Algorithm



Iteration Loop



Bibliography

[1] R. ASTER, B. BORCHERS, and C. THURBER. *Parameter Estimation and Inverse Problems.* 2003.

[2] A. BEN-ISRAEL, T.N.E. GREVILLE *Generalized Inverses: Theory and Applications.* Wiley-Interscience, 1974.

[3] D. CALVETTI and L. REICHEL. *Tikhonov Regularization of Large Problems* J. Comput. Appl. Math., 123, pp. 217–240, 2002.

[4] A.R. CONN, N.I.M. GOULD, and P.L. TOINT. *Trust-Region Methods.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[5] P. CRAVEN and G. WAHBA. Smoothing noisy data with spline functions. *Num. Math.*, 31:377–403, 1979.

[6] J.W. DEMMEL. *Applied numerical linear algebra.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[7] A. EDELMAN, T. ARIAS, and S.T. SMITH. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353 (electronic), 1999.

[8] A. EDELMAN and S.T. SMITH. On conjugate gradient-like methods for eigen-like problems. *BIT*, 36(3):494–508, 1996. International Linear Algebra Year (Toulouse, 1995).

[9] A.R. FORMICONI, E. LOLI PICCOLOMINI, et al. Numerical methods and software for functional Magnetic Resonance Images reconstruction *Ann. Univ. Ferrara*, Sez. VII Sc. Mat., Supplemento al Vol. XLV, 1-0, 2000.

[10] C. FORTIN and H. WOLKOWICZ. A survey of the trust region subproblem within a semidefinite programming framework. Technical Report CORR 2002-22, University of Waterloo, Waterloo, Canada, 2002. URL:http://orion.math.uwaterloo.ca:80/~hwolkowi/henry/reports/ABSTRACTS.html#surveytrs.

[11] D.M. GAY. Computing optimal locally constrained steps. *SIAM J. Sci. Statist. Comput.*, 2:186–197, 1981.

[12] G.H. GOLUB, M. HEATH, and G. WAHBA. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21:215–223, 1979.

[13] N. GOULD, S. LUCIDI, M. ROMA, and Ph. L. TOINT. Solving the trust-region subproblem using the Lanczos method. *SIAM J. Optim.*, 9(2):504–525, 1999.

[14] P.C. HANSEN. Truncated singular value decomposition solutions to discrete ill-posed problems with ill-determined numerical rank. *SIAM J. Sci. Statist. Comput.*, 11(3):503–518, 1990.

[15] P.C. HANSEN. Analysis of discrete ill-posed problems by means of the L-curve. *SIAM Rev.*, 34(4):561–580, 1992.

[16] P.C. HANSEN and D.P. O'LEARY. The use of the L-curve in the regularization of discrete ill-posed problems. *SIAM Journal on Scientific Computing*, 14:1487–1503, 1993.

[17] P.C. HANSEN. Regularization tools: a Matlab package for analysis and solution of discrete ill-posed problems. *Numer. Algorithms*, 6(1-2):1–35, 1994.

[18] P.C. HANSEN. *Rank-deficient and discrete ill-posed problems.* SIAM Monographs on Mathematical Modeling and Computation. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998. Numerical aspects of linear inversion.

[19] N.J. HIGHAM. *Accuracy and Stability of Numerical Algorithms.* SIAM, Philadelphia, 1995.

[20] A.E. HOERL and R.W. KENNARD. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[21] A.N. MALYSHEV. A unified theory of conditioning for linear least squares and tikhonov regularization solutions. *SIAM J. Matrix Anal. Appl.*, 24(4):1186–1196, 2003.

[22] J. NOCEDAL and S.J. WRIGHT *Numerical Optimization.* Springer, 1999.

[23] D.P. O'LEARY Near-Optimal Parameters for Tikhonov and Other Regularization Methods. *SIAM Journal on Scientific Computing*, 23:4, pp. 1161–1171, 2002.

[24] C.C. PAIGE and M.A. SAUNDERS LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM TOMS*, 8(1), 43-71 (1982).

[25] R. PENROSE. On best approximate solutions of linear matrix equations. *Proc. Cambridge Philos. Soc.*, 52:17–19, 1956.

[26] F. RENDL and H. WOLKOWICZ. A semidefinite framework for trust region subproblems with applications to large scale minimization. *Math. Programming*, 77(2, Ser. B):273–299, 1997.

[27] M. ROJAS and D.C. SORENSEN. A trust-region approach to the regularization of large-scale discrete forms of ill-posed problems. *SIAM J. Sci. Comput.*, 23(6):1842–1860 (electronic), 2002.

[28] C.B. SHAW, Jr. Improvement of the resolution of an instrument by numerical solution of an integral equation. *J. Math. Anal. Appl.*, 37:83–112, 1972.

[29] D.C. SORENSEN. Newton's method with a model trust region modification *SIAM J. Numer. Anal.* 19(2):409–426, 1982.

[30] D.C. SORENSEN. Minimization of a large-scale quadratic function subject to a spherical constraint. *SIAM Journal on Optimization*, 7(1):141–161, 1997.

[31] R. STERN and H. WOLKOWICZ Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations. *SIAM J. Optimization*, 5(2):286-313, 1995.

[32] R.J. STERN and J.J. YE. Variational analysis of an extended eigenvalue problem. *Linear Algebra Appl.*, 220:391–417, 1995.

[33] J. STOER and R. BULIRSCH. *Introduction to Numerical Analysis.* Springer-Verlag, New York, NY, 1980.

[34] A.N. TIKHONOV and V.Y. ARSENIN. *Solutions of Ill-Posed Problems.* V.H. Winston & Sons, John Wiley & Sons, Washington D.C., 1977. Translation editor Fritz John.

[35] Z. ZHANG and Y. HUANG. A projection method for least squares problems with a quadratic equality constraint. *SIAM J. Matrix Anal. Appl.*, 25(1):188–212, 2003.