A matrix-free method for regularisation with unrestricted variables

Bjørn Harald Fotland

June 2008

Thesis for the degree of Master of Science

Department of Informatics
University of Bergen
Norway

Notation

- Matrices are denoted by capital Latin letters such as A, B.
- Diagonal matrices are given in capital Greek letters such as Λ and Σ.
- Vectors and positive integers are represented by lower case letters such as b and j. The type will be clear from the context.
- Lowercase Greek letters are reserved for scalar values such as δ, λ.
- a_i denotes the i-th column vector of a matrix A.
- A_{ij} denotes the element in the i-th row and the j-th column vector of a matrix A.

An exception from the convention is Δ, which denotes a positive scalar. Some exceptions also occur in the description of applications, although their meaning should be clear from the context.

Acknowledgements

First and foremost I would like to thank my supervisor Professor Trond Steihaug for always having time to answer questions and discuss the thesis. His insight and comments have been indispensable.

I visited the Institute for Informatics and Mathematical Modelling at the Danish Technical University and I would like to thank Professor Per Christian Hansen and others at the institute for making my stay a pleasant experience.

Last, but not least, I would like to thank my mother Bjørg Reime Fotland for her support and Ida Gudjonsson for her support and patience.

Contents

Chapter 1

Introduction

This thesis will be concerned with discrete ill-posed problems. Such a problem can be posed as an ill-conditioned linear system. The ill-conditioning occurs naturally, often from an underlying continuous ill-posed problem, and additional a priori information needs to be incorporated into the problem to stabilise the solution. Discrete inverse problems are a group of problems where, given a model and a set of measured data, the intent is to obtain a set of model parameters. These problems occur in many fields of science. For instance, medical imaging and geophysics.

The thesis is devoted to two related problems. The first being the norm constrained least squares problem

$$\begin{aligned} \min_x & \frac{1}{2}\|Ax - b\|^2 \\ \text{subject to } & \|x\| \leq \Delta, \end{aligned}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and m and n are positive numbers. The second problem is the partially norm constrained problem

$$\begin{aligned} \min_{x,y} & \frac{1}{2}\| \begin{bmatrix} A & B \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} - b\|^2 \\ \text{subject to } & \|x\| \leq \Delta, \end{aligned}$$

where $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{m\times p}, b \in \mathbb{R}^m$ and m, n and p are positive numbers.

While the first problem has been closely examined, the second problem is new.

Applications in model predictive control and image misalignment can be modelled using the second problem. Model predictive control aims to control systems in process industries by constructing a prediction model. In image misalignment two or more images are misaligned with respect to each other and the task is to align the images.

Motivated by these applications, a method for the second problem will be developed and applied to small and large-scale problems. The method will use a solver for the first problem as an inner iteration. Due to the iterative nature of the solver, its accuracy and sensitivity will be explored.

Common properties of the method and the solver are that only matrix-vector products are required and that the methods have a low storage requirement.

The thesis consists of eight chapters.

Chapter 2 first introduces concepts from numerical linear algebra. The regularisation problem and the trust-region subproblem are subsequently presented, before showing that the problems are equivalent for a specific choice of known values.

In Chapter 3 matrix-free methods are introduced and a recent large-scale method for the trust-region subproblem is presented. Some recent extensions are also discussed.

Chapter 4 explains the motivations behind the remainder of the thesis and gives an overview the problem and applications that will be discussed.

Chapter 5 describes numerical tests carried out to investigate the accuracy and sensitivity of the large-scale trust-region method presented in Chapter 3.

In Chapter 6 the partially norm constrained problem is explored. A reformulation with concurrent first order optimality condations is presented and finally an algorithm is suggested.

Chapter 7 presents applications for the partially norm constrained problem and how the application fits into the problem. Some preliminary results are also shown.

Chapter 8 summarises the results, concludes and gives suggestions to further work.

Chapter 2

Regularisation and trust-region optimisation

This chapter first presents the common approach for solving linear systems. Subsequently the least squares problem is explained and a few properties of a group of problems called discrete ill-posed problems are discussed. Section 2.1 presents regularisation due to Tikhonov and Section 2.2 presents a norm constrained quadratic problem. Finally the connection between the first two sections is made in Section 2.3.

In numerical linear algebra one of the main goals is to solve a system of equations

$$Ax = b \tag{2.1}$$

with respect to x. Here $A \in \mathbb{R}^{m\times m}$, while $x \in \mathbb{R}^m$ and $b \in \mathbb{R}^m$. For the moment, let A have full rank (see Definition 2 in Appendix A).

Given A and b the aim is to obtain x by implicitly inverting A. The implicit inversion is done by solving a set of linear systems. There are many methods for this, each with its own properties with respect to computational cost, stability of the solution and the structure of A. Golub and van Loan [7] cover most of these methods. Here we restrict ourselves to the *singular value decomposition* (SVD), which provides us with a good analytical tool for explaining properties and differences between problems.

The SVD method decomposes A into three matrices $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{m\times m}$, both orthogonal (see Definition 3 in Appendix A) and $\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_m)$, such that

$$A = U\Sigma V^T.$$

The singular values σ_i along the diagonal of Σ are sorted such that

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m.$$

Given a SVD of A, the solution of the linear system is

$$x = A^{-1}b = V\Sigma^{-1}U^T b$$

where Σ^{-1} has the reciprocal of each element of Σ along the diagonal.

Often the right hand side b originates from measurements. This signifies that the components in b are perturbed with measurement errors. The order of the errors depends on the accuracy of the measuring equipment and the precision used to store the data, called measurement errors and discretisation errors respectively. To model the errors, b is split into two parts, $b = b_{exact} + s$, where b_{exact} consists of the true or exact, but unknown, data and the right part s represents the errors or noise introduced by the data acquisition process. In some types of problems, ignoring the errors in b may have severe effects on the solution. This will be discussed further in Section 2.1.

As A often comes from a discretisation of an integral equation, discretisation errors occur here as well, although we assume that these errors are negligible throughout the thesis.

Matrices are not always square and of full rank. If $\operatorname{rank} A < m$ the inverse of A, A^{-1}, is not defined and if we let $A \in \mathbb{R}^{m\times n}$, with $m > n$ the system may have an infinite set of solutions. Nonetheless, we seek a solution in some sense. This can be done by relaxing the formulation of the problem. The *least squares* method does precisely this, by only requiring that the difference between both sides of (2.1), the residual, is minimised. If there is no exact solution x to the linear system, that is, there does not exists a vector x in the $\mathcal{R}(A)$, the least squares method selects the solution closest in Euclidian distance to the range of A. On the other hand, if an infinite set of solutions exists, the method picks the solution with minimum norm. Mathematically the problem is written as

$$\min_x \|Ax - b\|,$$

where $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$. Here $\|\cdot\|$ denotes the l_2 norm (see Definition 1 in Appendix A). This norm will be the default throughout the thesis, $\|\cdot\| \equiv \|\cdot\|_2$. Other norms will be specified with appropriate subscripts.

It turns out that when using the SVD, the solution of the least squares method is quite similar to the inverse, namely

$$x_{LS} = A^\dagger b = V\Sigma^\dagger U^T b = \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i} v_i, \tag{2.2}$$

where x_{LS} denotes the least squares solution. Assuming $p = \operatorname{rank} A < m$, $\Sigma^\dagger$ has now non-zero elements along the first p elements of the diagonal and

zero elements elsewhere. $A^\dagger$ is called the Moore-Penrose generalised inverse or the *pseudoinverse*.

It is appropriate to introduce a few properties of matrices and characterisations of problems before proceeding to Section 2.1.

The condition number of $A \in \mathbb{R}^{m\times m}$ in the Euclidian norm is defined as

$$\kappa(A) = \|A\|\|A^{-1}\| = \frac{\sigma_1}{\sigma_m},$$

where the last expression is the largest singular value divided by that smallest. Note that $\kappa(A) \geq 1$ for all A. If $\kappa(A)$ is small, A is said to be *well-conditioned* and on the other hand if $\kappa(A)$ is large, A is termed *ill-conditioned*. If A does not have full rank, $\kappa(A) = \infty$. The accuracy of the solution largely depends on the condition number of the matrix. In fact, as many as $\log_{10}\kappa(A)$ digits of accuracy can be lost after solving a linear system, see [27, Theorem 12.2].

A problem is said to be *well-posed* (due to Hadamard) if a solution exists, the solution is unique and the solution depends continuously on the data, b. When a problem does not satisfy one or more of these requirements it is termed *ill-posed*. From this definition we see that the least squares method solves problems where the first two requirements are not satisfied. The third requirement can also be stated differently; a small change in the data, b, should cause a small change in the solution.

If the discretised problem does not satisfy the third requirement, it typically has a high condition number. Discrete inverse problems are a group of problems having this property. The condition number of such problems is often extremely high and if the least squares method is applied to them, it does not produce meaningful solutions.

A classical example of an ill-posed problem is a Fredholm integral equation of the first kind, which can in general be written on the form

$$\int_0^1 K(s,t)f(t)dt = g(s), \qquad 0 \leq s \leq 1.$$

where g and K are known functions and f is an unknown function. Many inverse problems can be stated on this form. Discrete inverse problems often consist of a discretisation of the kernel K. The right hand side $g(s)$ is usually a set of measurements contained in a vector b.

Discrete inverse problems are characterised by having a gradual decay in the singular values, with no apparent jump in order of magnitude. If such a jump exists the problem is called *rank deficient* and can be solved by ignoring smaller singular values. A method called *truncated SVD* (TSVD) only utilises the k largest singular values for the solution. In this way a solution with lower resolution can be obtained by computing

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

where u_i, v_i are the i-th column vectors of U and V respectively and σ_i is the i-th diagonal element of Σ. If a decomposition of A is not feasible, we cannot easily know where to cut off the singular values and the problem should be treated as a discrete inverse problem.

2.1 Tikhonov regularisation

The least squares method does not take the errors in the right hand side b into account. When dealing with discrete ill-posed problems the noise must be considered. Due to the severe ill-conditioning of A, a small perturbation in b may change the solution completely. This leaves little confidence in the solution.

Regularisation methods impose additional constraints on the problem in hope of improving the accuracy and stability of the solution. The most widely used regularisation method is Tikhonov regularisation [26].

Given a δ, the method solves the problem

$$\min_x \|Ax - b\|^2 + \delta^2 \|Lx\|^2,$$

where δ is known as the *regularisation parameter* and L is often a discrete approximation to the first or second derivative. From here on we only consider Tikhonov regularisation on standard form,

$$\min_x \|Ax - b\|^2 + \delta^2 \|x\|^2. \tag{2.3}$$

In literature this form is also described as zeroth-order Tikhonov regularisation [2, Chapter 5], since no derivative information is used directly in the formulation. As long as L has full rank, the standard form can be obtained by a change of variables. If $L \neq I$ the regularisation imposes a smoothing constraint on the solution, and if $L = I$ it imposes a constraint on the size of the solution.

The least squares method minimises the residual. The goal of regularisation is twofold, minimising the norm of the residual while keeping the norm of the solution small. Notice that if $\delta = 0$, problem (2.3) reduces to the least squares problem. With this in mind Tikhonov regularisation can be seen as a way of further relaxing the original linear system (2.1). The solution closest

to the range of A is no longer the goal, but the solution should not stray too far from it either.

Choosing δ is a difficult problem in itself. There are three main classical strategies for determining the parameter, *the discrepancy principle*, *generalised cross-validation* (GCV) and the *L-curve criterion*. The first requires an estimate of the noise level, while the others are post priori estimates, requiring the solution of several regularisation problems for different values of δ. More recently another method due to Hansen et al. [10] has been suggested to decide the regularisation parameter. This method is based on statistical tools and fast Fourier transforms and is thereby computationally feasible even for large-scale problems.

There are two other ways of posing the Tikhonov regularisation problem, namely

$$(A^T A + \delta^2 I)x = A^T b$$

and

$$\min_x \left\| \begin{bmatrix} A \\ \delta I \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|$$

Note that latter is a least squares problem. By applying the SVD, the solution to the regularisation problem can be written as

$$x_\delta = A_\delta^\# b = \sum_{i=1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \delta^2} \frac{u_i^T b}{\sigma_i} v_i$$

where x_δ denotes the regularised solution using a given value of δ and $A_\delta^\#$ denotes the generalised inverse also dependent on δ. The vectors u_i and v_i represent the i-th vector of the orthogonal matrices in the SVD. Comparing the solution with (2.2), only the fraction $\sigma_i^2/(\sigma_i^2 + \delta^2)$ differ. As δ is constant for each term in the sum, we see that as the singular values σ_i decrease the weighting supplied by δ decrease. The small σ_i that correspond to high frequencies are thereby dampened and the solution becomes less sensitive to changes.

To check if regularisation is needed, the *discrete Picard condition* can be used. The discrete Picard condition is said to be satisfies if $|u_i^T b|$ decay faster than σ_i. Observe that if this condition holds the norm of the solution will be small. See Figure 2.1 for an example of the typical behaviour of a discrete ill-posed problem.

To emphasise the need for regularisation, let us look at an example.

Example 1. *Given $A \in \mathbb{R}^{20\times 20}$ and $b \in \mathbb{R}^{20}$, where A is a discretisation of the Shaw problem from [9]. A more detailed presentation of the Shaw problem is given in Section 5.1.1.*

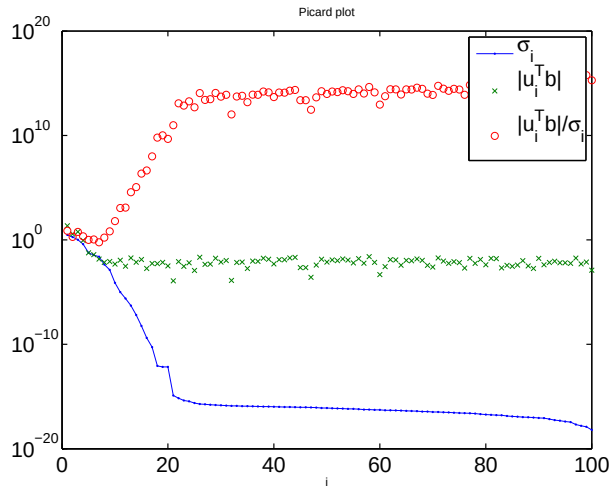


Figure 2.1: *A Picard plot of the Shaw test problem*

The aim is to find a solution to $Ax = b$*. A least squares approach using the* exact b *gives* $\|x_{LS}\| \approx 4.87$*, while a SVD-based Tikhonov approach with the exact* b *and* $\delta = 1.93 \times 10^{-5}$ *results in the solution* $\|x_\delta\| \approx 4.46$*. The norm of the exact solution is* $\|x_{exact}\| \approx 4.46$*. The relative accuracy of* x_{LS} *is* 0.4485 *and for* x_δ *it is* 0.015*. Figure 2.2 compares the elements of the solutions. Observe that the least squares solution behaves erratic in the middle of the solution, while the Tikhonov solution follows the curve of the exact solution.*

2.2 The trust-region subproblem

Optimisation methods can be divided into two main areas, line search and trust-region methods. In this section we look closer at the latter group of methods. The main task at each iteration is to solve a problem on the form,

$$\begin{aligned} &\min_x \frac{1}{2} x^T H x + g^T x \\ &\text{subject to } \|x\| \leq \Delta, \end{aligned} \tag{2.4}$$

where $H \in \mathbb{R}^{n \times n}$ and symmetric and $x, g \in \mathbb{R}^n$. The problem only needs to be solved to a desired accuracy when applied to optimisation problems.

This problem represents, from an optimisation point of view, a quadratic approximation to the function we want to find a local minimum of. The quadratic approximation is based on the Taylor series expansion. At each iteration of a trust-region method, we try to find a direction of decrease

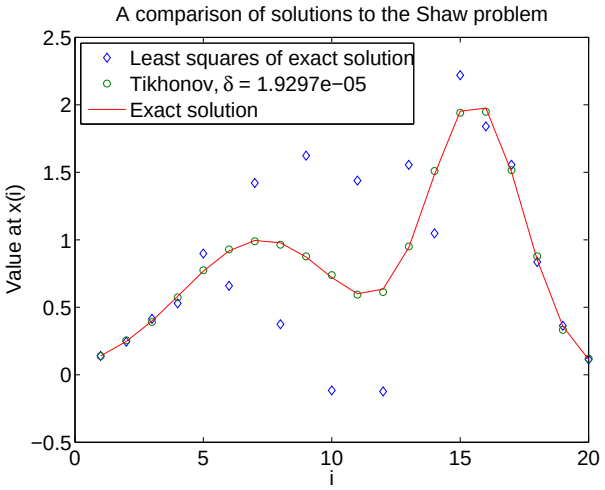


Figure 2.2: *A comparison of the least squares solution and the Tikhonov regularised solution for the Shaw problem*

(minimise the model) subject to a restriction on the distance Δ from the current iterate. Δ is called the *trust-region radius*, implying that we only trust our model around the current iterate.

The solution of the problem can be divided into two classes. Either the solution lies inside the trust-region, and the constraint is inactive or the solution lies on the boundary.

Observe that from a regularisation point of view, we are restricting the size of the solution. If the solution lies strictly inside the trust-region, no regularisation is achieved.

A special property of (2.4) is that the solution has to be on a special form. The following theorem discovered independently by Gay [6] and Sorensen [24] explain the structure of the solution.

The result is stated in non-standard form, with a non-positive Lagrange multiplier, which is in accordance with papers describing a large-scale TRS method (explained in Section 3.1) that will be central later in the thesis.

Theorem 2.2.1. *A vector $x^* \in \mathbb{R}^n$ is a global solution of the TRS*

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T H x + g^T x \text{ subject to } \|x\| \leq \Delta$$

if and only if x^* is feasible and there exists a scalar $\lambda \leq 0$ such that

$$\begin{aligned} (H - \lambda I)x^* &= -g, \\ \lambda(\Delta - \|x^*\|) &= 0, \\ (H - \lambda I) &\text{ is positive semi-definite} \end{aligned}$$

Proof. See for instance [24] or [17]. □

When it comes to solving the TRS we can divide the methods available into two distinct groups. One group of methods that solve the TRS approximately and another group of methods that provide exact solutions. The former group consists of methods mainly applicable in the context of optimisation, where a trade-off between the efficiency and accuracy is relevant since several TRS problems need to be solved.

The most prominent among these are Powell's dogleg method [17, Chapter 4, p. 73], which improves on the Cauchy point by a two step approximation, and Steihaug's approach, see [25] or [17, Chapter 7, p. 171], a conjugate gradients (CG) based method with a trust-region modification providing a solution lying in a Krylov subspace [27, p. 245].

Providing an exact solution on the other hand is a more cumbersome process, which will be discussed further in Section 3.1. The method of choice for small and medium scale problems (where a matrix factorisation is affordable) is the method due to Moré and Sorensen [15]. A more recent large-scale method, LSTRS [23], will be the main topic of Chapter 3.

2.3 Bridging the gap

It turns out that the relationship between Tikhonov regularisation and the trust-region subproblem is quite close. Let us start out with the situation where $H = A^TA$ and $g = -A^Tb$, and write the TRS slightly differently,

$$\begin{aligned} &\min_x \frac{1}{2}x^TA^TAx - (Ax)^Tb \\ &\text{subject to } \|x\|^2 \leq \Delta^2 \end{aligned}$$

By adding the constant $\frac{1}{2}b^Tb$ to the objective function and writing the Lagrange function of this, we get

$$\mathfrak{L}(x, \lambda) = \frac{1}{2}\|Ax - b\|^2 - \lambda(\|x\|^2 - \Delta^2),$$

with $\lambda \leq 0$ as the Lagrange parameter. Minimising with respect to x results in

$$\min_x \frac{1}{2}\|Ax - b\|^2 - \lambda\|x\|^2.$$

By selecting $\lambda = -\delta^2$, Tikhonov regularisation and the boundary solution of the TRS is shown to be equivalent with the given choices of H and g.

Although equivalent, this is under the assumption of the right values of the regularisation parameter and the trust-region radius. This is an important difference between trust-region based regularisation and Tikhonov regularisation. An interesting point is that there does not seem to be many direct relations to δ in applications. However, applications do exist for which an estimate of Δ is available. One such application is image processing, where Δ is the energy of the image signal [4, Chapter 5, p. 98].

A method for solving the TRS is presented in Chapter 3.

Chapter 3

Matrix-free methods

There are situations where the sheer size of the problem limits the use of full matrix decompositions such as the SVD. One such situation is image deblurring, where the right hand side, b is the measured image and A is a point spread function (PSF) which models the blurring effect.

An image of dimensions 256×256 has 65536 pixels (considering only the grayscale case). To fit into a linear model the image is vectorised by stacking each column in the image, creating a $b \in \mathbb{R}^m$ with $m = 256^2$. Assume A is square, then its size must be $256^2 \times 256^2$. If A was to be stored explicitly as double precision floats it would require $256^2 \times 256^2 \times 8$ bytes $\approx 32GB$ of storage.

Clearly an image of this size is quite small by today's standards. A full matrix decomposition is certainly out of the question, due to both storage and computational cost. Luckily PSFs and many other large A can be generated analytically and/or stored by sparse matrix schemes which exploit the large number of zero elements and other structure such as symmetry in A. A matrix decomposition such as QR or SVD would introduce dense matrices.

Methods for sparse matrices and large A can only work with the actions of A and A^T and possibly a partial decomposition. These methods are called *matrix-free*. They are iterative in nature, which means that they start out by an initial estimate and then converge towards a solution. The amount of work needed to converge is not known in advance, as opposed to direct methods, where the computational work can be estimated precisely.

In Section 3.1 a recent matrix-free algorithm for TRS is described. An important result about the potential hard case being the common case in regularisation is presented. The methods presented in Section 3.2 and 3.3 show extensions to handle non-negatively constrained and linear inequality constrained regularisation.

3.1 The large-Scale TRS

Large-scale or matrix-free methods are needed when the dimension of a problem is so large that either storage of A directly is not feasible or when direct methods become too costly to compute. In this section the main topic will be the large-scale TRS (LSTRS) method developed by Rojas et al. [23]. The method handles all cases of the TRS and was developed with regularisation of discrete ill-posed problems in mind.

First the idea behind the algorithm will be presented and afterwards a small digression will be made to introduce a useful function related to the characterisation of eigenvalues. Afterwards this function will be related to the method. The final part covers the so-called *hard case* and the potential hard case of the TRS and shows that the latter is the common case for discrete ill-posed problems.

Recall from Section 2.2 and Theorem 2.2.1 that the global solution of the TRS requires that

$$(H - \lambda I)x = -g, \tag{3.1}$$

with $(H - \lambda I)$ positive semidefinite and $\lambda \leq 0$. This can be written as

$$\begin{pmatrix} g & H \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} = \lambda x,$$

which looks like an eigenvalue problem, apart from the "missing" one in the first component in the rightmost vector. Adding this component, along with a modified matrix, results in

$$\begin{pmatrix} \alpha & g^T \\ g & H \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} = \lambda \begin{pmatrix} 1 \\ x \end{pmatrix}, \tag{3.2}$$

where α is a scalar. This is a symmetric eigenvalue problem, solvable by methods such as the QR algorithm [27, Lecture 28-29] and the Lanzcos iteration [27, Lecture 36]. Since the emphasis is on large-scale a variant of the latter is used; the implicitly restarted Lanzcos method (IRLM) [14]. For future reference, let

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

be the bordered matrix parametrised by α.

The main idea is to solve a sequence of parametrised eigenvalue problems, adjusting α such that $H - \lambda I$ is positive semidefinite and the remaining requirements of a global solution are satisfied.

Before further elaboration the *secular function* and its derivative will be introduced. The secular function is related to (3.1) and the characterisation of eigenvalues.

In the following, an analysis of the function is carried out by using the eigenvalue decomposition, which factorises a symmetric matrix into a set of orthonormal eigenvectors $Q = [q_1 \ q_2 \ \cdots \ q_n]$ and a diagonal matrix $\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_n)$, where $\lambda_i, i = 1, 2, \ldots, n$ are the eigenvalues of, such that $H = Q\Lambda Q^T$. By convention eigenvalues are ordered non-decreasing

$$\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n,$$

which is the opposite of singular values.

The secular function is defined as

$$\phi(\lambda) = g^T(H - \lambda I)^{-1}g, \tag{3.3}$$

and the derivative of ϕ at λ is

$$\phi'(\lambda) = g^T(H - \lambda I)^{-2}g. \tag{3.4}$$

with the assumption that $g \neq 0$. Applying the eigenvalue decomposition to (3.3) results in

$$\begin{aligned}\phi(\lambda) &= y^T(\Lambda - \lambda I)^{-1}y \\ &= \sum_{i=1}^{n} \frac{y_i^2}{\lambda_i - \lambda}\end{aligned}$$

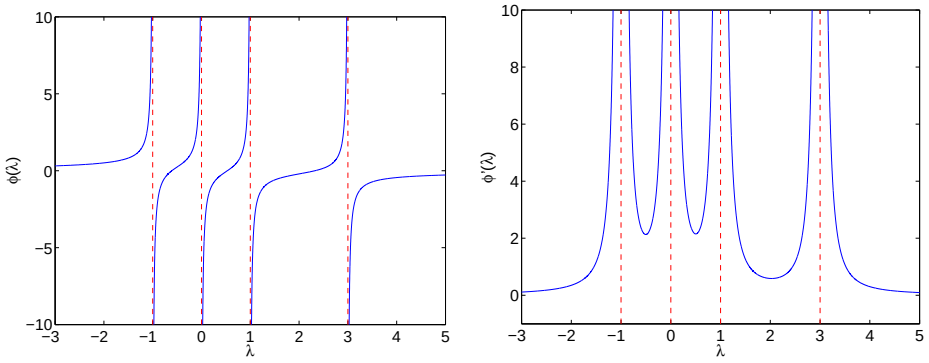
where $y = Q^T g$. Using a similar approach with the derivative of $\phi(\lambda)$ results in

$$\phi'(\lambda) = \sum_{i=1}^{n} \frac{y_i^2}{(\lambda_i - \lambda)^2}.$$

Observe that the poles of $\phi(\lambda)$ and $\phi'(\lambda)$ are the eigenvalues of H; when $\lambda = \lambda_i$. This behaviour is illustrated in Figure 3.1.

An important observation is that when $g^T x \neq 0$, with $x = -(H - \lambda I)^{-1}g$, the secular function and its derivative are monotonically increasing on the interval $(-\infty, \lambda_1)$. Also the eigenvalues of H interlace the eigenvalues of B_α. This is known as Cauchy's interlace theorem or the *interlacing property* [7, Theorem 8.1.7] and applies to any symmetric matrix.

For (3.1) to hold, we need to ensure that $H - \lambda I$ is positive semidefinite. If we can find the smallest eigenvalue λ of B_α the interlacing property ensures that $\lambda \leq \lambda_1$. $H - \lambda_1 I$ is thus positive semidefinite.



(a) $\phi(\lambda)$ **(b)** $\phi'(\lambda)$

Figure 3.1: *The secular function $\phi(\lambda)$ and its derivative for a matrix with eigenvalues $-1, 0, 1$ and 3.*

Given an initial α, the Hessian H and the gradient g, the strategy is to solve the eigenvalue problem obtaining the eigenpair $(\lambda_1, (1, x^T)^T)$ corresponding to the smallest eigenvalue of B_α. From this point the $H - \lambda I$ will be positive semidefinite. The next step is satisfying the remaining requirements from Theorem 2.2.1, namely

$$\lambda(\|x\| - \Delta) = 0 \quad \text{and} \quad \lambda \leq 0. \tag{3.5}$$

This is known as a complementarity condition, because either $\lambda = 0$ and the solution is $x = -H^{-1}g$, the unconstrained minimiser, or $\lambda \leq 0$ and $\|x\| = \Delta$, which corresponds a solution on the boundary of the trust-region.

It is time to look closer at the symmetric eigenvalue problem. Multiplying out both sides of (3.2) and rearranging result in

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

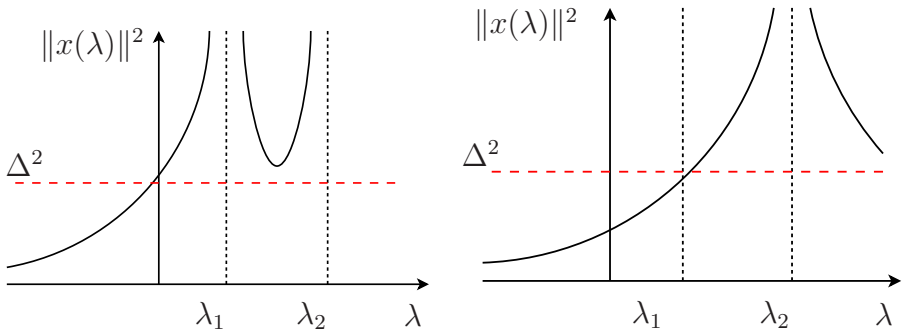
Assuming the matrix $H - \lambda I$ is nonsingular and substituting for x in the right expression yields

$$\alpha - \lambda = g^T (H - \lambda I)^{-1} g = \phi(\lambda)$$

where the right hand side is precisely the secular function (3.3). Its derivative can be written

$$x^T x = g^T (H - \lambda I)^{-2} g = \phi'(\lambda).$$

Observe that under the assumption that the eigenvector can be normalised to have the first component equal to one, $\phi(\lambda) = -g^T x$ and $\phi(\lambda) = x^T x$ can easily be evaluated.



(a) *Regular case* **(b)** *Hard case*

Figure 3.2: *Cases of the trust-region subproblem*

The aim is now to find an α such that (3.5) holds. The secular equation is

$$\|x\|^2 - \Delta^2 = 0. \tag{3.6}$$

This corresponds to where Δ^2 intersects $\phi'(\lambda)$, see Figure 3.2a. Since interest lies only in the interval $(-\infty, \lambda_1)$ we know that the function is monotonically increasing and the intersection is unique. As long as a decomposition of H is affordable, Newton's method can be applied to solve the equation. Large-scale options are the secant method or rational approximations. LSTRS uses the latter, but not directly on the highly nonlinear secular equation. Instead the following equation

$$\frac{1}{\phi'(\lambda)} - \frac{1}{\Delta^2} = 0,$$

is used. This equation is nonlinear, but typically has linear behaviour near the poles.

At each iteration a new α is chosen using rational interpolation on the available points of $\phi(\lambda)$ and $\phi'(\lambda)$, with safeguarding to guarantee convergence.

Up until now only the case where $g^T x \neq 0$ has been discussed. If $g^T x = 0$ the method lined out will break down. The two cases can be seen in Figure 3.2. The hard case only occurs when $H - \lambda I$ is positive semidefinite and singular, or indefinite.

As a symmetric eigenvalue problem this corresponds to the case where the first component of the eigenvector cannot be normalised,

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

When the eigenvector cannot be normalised it can be shown [22, Theorem 3.1] that any other eigenvector can be normalised to have the first component equal to one. Due to this, LSTRS computes the two eigenpairs, corresponding to the smallest eigenvalue and another from a cluster of the smallest eigenvalues.

Up to now the case, known as the *hard case* has been ignored. Recall that Q is the set of eigenvectors of H. The exact hard case occurs when $g \perp Q_1$, where Q_1 is the submatrix spanning the subspace corresponding to λ_1 and when $\|x\|^2 = \Delta^2$. In this case the secular equation has no pole corresponding to λ_1, as shown in Figure 3.2b.

For discrete ill-posed problems $H = A^T A$ and $g = -A^T b$, which implies that the requirement of $H - \lambda I$ positive semidefinite is automatically satisfied. An important result for discrete ill-posed problems is that the potential hard case or the near potential hard case is the common case [20, Lemma 3.2]. By potential we mean that $g \perp Q_1$, but $\Delta^2 = \|x\|^2$ is not necessarily satisfied and by near we mean that g is numerically orthogonal to Q_1, this makes the secular function extremely non-linear near the pole.

To show that the potential hard case occurs frequently for discrete ill-posed problems we need a few assumptions. First note that discrete ill-posed problems have a cluster of small singular values. This can be observed by making a sufficient discretisation of an underlying continuous inverse problem. Secondly, we assume that the discrete Picard condition holds, which means that $u_j^T b/\sigma_j \to 0$ on average, as the singular values decay. Hence $u_j^T b \to 0$ faster than σ_j.

Recall the singular value decomposition $A = U\Sigma V^T$ and that $b = b_{exact} + s$, where s is a vector of random noise. We now have $H = A^T A = V\Sigma^2 V^T$ and

$$\begin{aligned} g &= -V\Sigma U^T b \\ &= -V\Sigma U^T (b_{exact} + s). \end{aligned}$$

Notice that the columns of V corresponds to eigenvector of H.

Let u_j and v_j denote the left and right singular vectors respectively and σ_k be the k-th singular value with multiplicity m_k and $1 \leq j \leq m_k$. With V orthogonal, $V_k = \{v_k, v_{k+1}, \ldots, v_n\}$ is a basis for the subspace corresponding σ_k. The relation $\sigma_k^2 = \lambda_{n-k+1}$ is clear from the expression for H and by the opposite ordering of eigenvalues and singular values. Choose k such that it corresponds to the smallest eigenvalue, then V_k is equal to the eigenspace Q_1 corresponding to λ_1. Hence we can set $x = v_j$. Taking the inner product with v_j results in

$$v_j^T g = -v_j^T V\Sigma U^T (b_{exact} + s)$$

and since $v_j^T v_i = 0, i \neq j$ and $v_j^T v_j = 1$ by orthogonality

$$v_j^T g = -\sigma_k (u_j^T b_{exact} + u_j^T s).$$

The discrete Picard condition implies that $\sigma_k u_j^T b_{exact}$ will be effectively zero showing that the exact hard case occurs when there is no noise present. As long as σ_k is small, the noise must be large to avoid being numerically orthogonal, which means the potential hard case will occur.

Since there is a cluster of small singular values in discrete ill-posed problems, the potential hard case is likely to occur in multiple instances.

An interesting observation is that when σ_k is not very small, the noise in s may improve the problem in the sense that only a near potential hard case may occur. This suggests that the TRS approach to regularisation is easier to solve for high levels of noise in b, [20, Section 3.2].

3.2 Adding non-negativity constraint

Regularisation is about imposing properties on the solution in hope of achieving more stable and accurate solutions. Ensuring non-negativity of the solution is an additional requirement that can be imposed. The extension to non-negativity has applications in for instance image deblurring, where all pixel values must satisfy this property.

The non-negativity constrained regularisation problem is stated as

$$\begin{aligned} \min_x & \frac{1}{2}\|Ax - b\|^2 \\ \text{subject to } & \|x\| \leq \Delta \\ & x \geq 0, \end{aligned}$$

where $x \geq 0$ and the inequality is meant componentwise.

In [21] Rojas and Steihaug presents an interior-point trust-region method for large scale non-negative regularisation. The trust-region part is solved by the LSTRS algorithm. The objective function used is a logarithmic barrier function which eliminates the non-negativity constraint by including an addition term in the objective function. The aim is to

$$\begin{aligned} \min_x & \ \frac{1}{2}\|Ax - b\|^2 - \mu \sum_{i=1}^{n} \log \ x_i \\ \text{subject to } & \|x\| \leq \Delta \end{aligned}$$

The second term in the objective function goes to infinity as x_i goes to zero illustrated in Figure 3.3. Observe that as μ gets smaller the components are allowed to approach zero. The barrier term imposes non-negativity by forcing positivity of the solution.

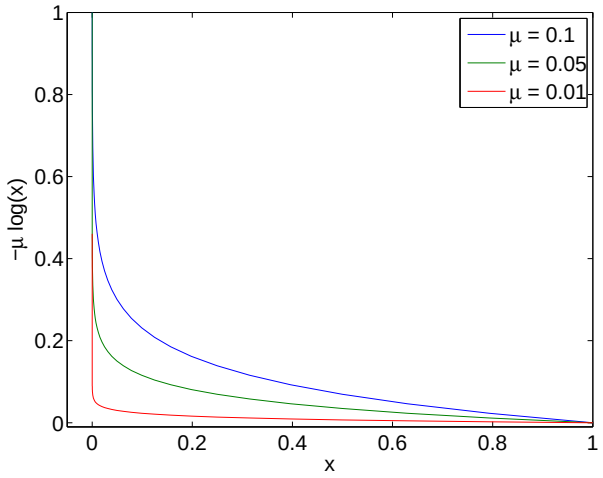


Figure 3.3: *The barrier term with only one component x. The functions for $\mu = \{0.1, 0.05, 0.01\}$ are shown in blue, red and green respectively.*

At each iteration of LSTRS, a trust region problem is solved while driving the μ parameter in the barrier term to zero. The parameter is often called the *barrier parameter* or the penalty-parameter. The hope is that as μ approaches zero the non-negativity constraints are satisfied and the objective function minimised.

Note that an extension to problems with linear inequality constraints on the form $G^T x \geq c$, $G \in \mathbb{R}^{n \times p}$ and $c \in \mathbb{R}^p$ is also discussed in the same paper.

Morigi et al. [16] suggest using a modified barrier function for the non-negativity extension. Namely,

$$\begin{aligned} \min_x \ & \frac{1}{2}\|Ax - b\|^2 + \frac{\phi(\mu)}{2}\|x\|^2 - \mu \sum_{i=1}^{n} \log\ x_i \\ \text{subject to } & \|x\| \leq \Delta \end{aligned}$$

where $\phi(\mu)$ is a positive increasing function $\mu \geq 0$ and $\phi(0) = 0$. The authors use $\phi(\mu) = \mu$ in numerical results. The method relies on the LSQR method [18] to solve the TRS. LSQR is mathematically equivalent to the conjugate gradient method. More information about LSQR with regards to regularisation can be found in [8, Chapter 6].

Iterates not satisfying the non-negativity constraint are projected into the first orthant. The extra term added, compared to the method of Rojas and Steihaug, is introduced to secure that the modified barrier function is strictly convex.

3.3 A generalisation to inequality constrained regularisation

By adding linear equality constraints to the regularisation, more properties of the solution can be included. The problem is then

$$\begin{aligned} &\min_x \frac{1}{2}\|Ax-b\|^2 \\ \text{subject to } &Cx \leq d \\ &\|x\| \leq \Delta. \end{aligned}$$

In Paper B in [3] Bergmann and Steihaug presents an interior-point method for this problem. The method applies the barrier approach to the inequality constraints, resulting in the modified problem

$$\begin{aligned} &\min_x \frac{1}{2}\|Ax-b\|^2 - \mu \sum_{i=1}^{m} \log(d_i - \sum_{j=1}^{n} C_{ij}x_j) \\ \text{subject to } &\|x\| \leq \Delta, \end{aligned}$$

where μ is the barrier parameter. At each iteration the modified problem is solved. Then a line search approach is used to assure that the step remains feasible with respect to the inequality constraints. The barrier parameter is chosen by an update rule based on an estimate of the dual variables, also known as the Lagrange multipliers.

By setting $C = -I$ and $d = 0$, the problem models the same situation as in Section 3.2, which means that imaging applications are also relevant to this problem. In [3] several imaging applications are mentioned. Among these are the possibility to impose upper and lower bounds on the solution and impose constraints on the mean in parts of an image or the whole image.

Chapter 4

The problems to be discussed

The previous chapters have not introduced any new results. They give an introduction to concepts and methods used for trust-region based regularisation, with emphasis on large-scale methods.

This chapter introduces the material which is treated in the upcoming chapters. Section 4.1 describes the main motivations for Chapter 5, while Section 4.2 introduces the problem of regularisation with unrestricted variables that will be elaborated in Chapters 6 and 7.

4.1 Accuracy and sensitivity of LSTRS

The result relating Tikhonov regularisation and the TRS is a mathematical one. Because of the discrete nature of computers, two methods that are mathematically equivalent may produce different results. An example of this is the QR decomposition by the Gram-Schmidt method. The method aims to decompose A into an orthogonal matrix Q and a tridiagonal matrix R. The classic version of the method [27, p.51] is unstable in a sense that it does not always produce a numerically orthogonal columns in Q. A modified version [27, p.58] results in a more stable method. Still, the methods are mathematically equivalent. It is therefore of interest to compare Tikhonov regularisation with LSTRS numerically. We want to investigate to what degree the equivalence carries over to the computational case.

As mentioned in Section 3.1 the LSTRS method relies on an iterative eigensolver when applied to large-scale problems. Iterative methods usually only provide approximate solutions, in this case a couple of eigenpairs (the second reserved for the exact hard case). With this in mind a further investigation of the sensitivity of the smallest eigenvalue and its corresponding eigenvector will be made.

4.2 The problem

The main goal in this thesis is to develop a large-scale method for regularisation of unrestricted variables by using LSTRS as the TRS solver. By unrestricted variables, we mean that some components of the solution are not restricted by the trust-region constraint. We write the problem as

$$\begin{aligned} &\min_{x,y} \frac{1}{2}\| \begin{bmatrix} A & B \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} - b\|^2 \\ &\text{subject to } \|x\| \leq \Delta. \end{aligned} \tag{4.1}$$

Some immediate observations are in order. Since only the components x_i of x are restricted to be within Δ, the corresponding matrix A may be ill-conditioned and originate for instance from an inverse problem. The unrestricted y_i components of y imply that B must be a well-conditioned matrix to some degree. We are therefore looking at a problem for applications where one part of the problem need restrictions, while the other part needs no further restrictions.

The motivation behind the development are applications in model predictive control (MPC) and imaging. In model predictive control the aim is to adjust system parameters based on a prediction model of a system. Some parts of a system may cause unstable predictions and need constraints on the change allowed. Other parts may be stable and can be predicted in an unrestricted manner.

In imaging, deblurring is a typical example of a discrete inverse problem. Deblurring may also occur in connection with misalignment of layers in colour images or between subsequent frames in a video. This may for instance happen due to camera intrinsics.

In Chapter 7 the two applications will be explained in further detail, and available test results presented.

Chapter 5

Accuracy and stability of LSTRS

This chapter is devoted to numerical testing of the LSTRS algorithm with comparisons to a conventional algorithm for Tikhonov regularisation based on the SVD. Several inverse test problems from Hansen's *Regularization Tools* [9] are used in the tests. The LSTRS implementation used is from the Matlab package due to Rojas et al. [23].

The overall aim is to use the LSTRS as a TRS solver for the development of regularisation algorithms. Therefore it is assuring to verify that the mathematical properties carry over to the computational case. Especially so since the use of TRS based algorithms has yet to see wide use in practice [2, Chapter 6, Notes].

Section 5.1 explains the test set-up, choices of variables and presents selected test problems. In section 5.2 the boundary solution of LSTRS is compared to Tikhonov regularisation, and finally in Section 5.3 the sensitivity of an eigenpair used by LSTRS is explored. Concluding remarks are given in Section 5.4.

5.1 Tolerances and test problems

To ensure convergence and to allow flexibility in the LSTRS algorithm with respect to the type of TRS to be solved, several parameters need to be set. Recall that regularisation problems account for the special case where $H = A^T A$ and $g = -A^T b$. Only the most relevant tolerance parameters for the testing will be described here, but for completeness a description and default values of tolerances are given in Appendix B . See [23] for further details.

Since only problems in need of regularisation will be considered here, **lopts.correction** and **lopts.interior** are disabled as suggested in [23]. The former was disabled because it can introduce high-frequency components into the solution and the latter was disabled because we are only interested in the boundary solution, for which the equivalence is proven.

A relevant tolerances is ϵ_Δ, which requires that a boundary solution x satisfies

$$\frac{(\|x\| - \Delta)}{\Delta} \leq \epsilon_\Delta.$$

Another is ϵ_{Int}, which declares the algebraically smallest eigenvalue of B_α. The eigenvalue λ_1 is considered positive if $\lambda_1 > -\epsilon_{Int}$. The parameter $\epsilon_{Int} = 0$ in all tests, while $\epsilon_\Delta = 10^{-4}$ by default, but will be changed in some cases.

Since the Shaw problem and the inverse heat problem are recurring test problems throughout the thesis a more detailed mathematical characterisation is given in Section 5.1.1 and Section 5.1.2.

5.1.1 The Shaw problem

The Shaw problem is a one-dimensional image restoration problem. A is the discretisation of $K(s,t)$ using quadrature, and $f(t)$ is the analytical solution. We want to solve $\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} K(s,t)f(t)dt = g(t)$ for $f(t)$.

$$\begin{aligned} K(s,t) &= (\cos(s) + \cos(t))^2 \left(\frac{sin(u)}{u} \right)^2 \\ u &= \pi(\sin(s) + \sin(t)) \\ f(t) &= a_1 e^{-c_1(t-t_1)^2} + a_2 e^{-c_2(t-t_2)^2} \end{aligned}$$

The standard values given in [9] of the constants are used, namely $a_1 = 2$, $a_2 = 1$, $c_1 = 6$, $c_2 = 2$, $t_1 = 0.8$ and $t_2 = -0.5$. Figure 5.1a shows the analytic solution.

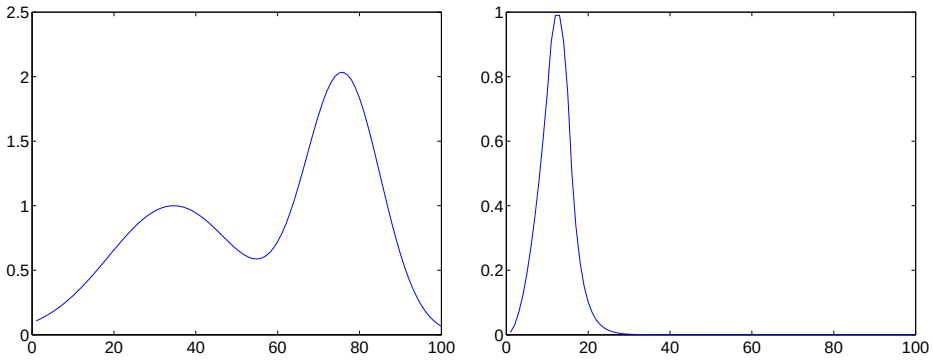
5.1.2 The Heat problem

The inverse heat equation is a Volterra integral of the first kind. Its kernel is given by $K(s,t) = k(s-t)$, where

$$k(t) = \frac{t^{-3/2}}{2\kappa\sqrt{\pi}} e^{-1/(4\kappa^2 t)}.$$

$K(s,t)$ is discretised by quadrature (see for instance [2, Chapter 3]). We want to solve $\int_0^1 K(s,t)f(t)dt = g(t)$ for $f(t)$. The variable κ can have a

value between 1 and 5, where 1 gives a severely ill-conditioned problem and 5 gives a well-conditioned problem. Unless stated otherwise, $\kappa = 1$.



(a) *The Shaw problem* **(b)** *The Heat problem*

Figure 5.1: *Analytical solutions*

5.2 Numerical equivalence

Mathematical equivalence between Tikhonov regularisation and the trust-region subproblem was shown in Section 2.3. Assuring as this is from a mathematical point of view, it does not in general imply that numerical algorithms will be well-behaved and produce accurate or even similar solutions.

Computers do not represent floating point numbers exactly. Instead a method similar to scientific notation is used to represent a real number as a combination of an exponent and a mantissa. In relation to this ϵ_{mach} is often used to denote the larger distance between two numbers in floating point format.

Other, more relevant errors, are discretisation errors and measurement errors. It is often the case that A is obtained from the discretisation of an integral equation. This introduces discretisation errors. If we look at the system $Ax = b$, b is nearly always obtained by measurements with some type of equipment. Recall from Chapter 2 that we model this by having $b = b_{exact} + s$, where b_{exact} is the unknown exact measurement, and s models the noise. Propagation errors are also present. Depending on which algorithm is used, the size of propagation errors may result in quite different solutions.

The L-curve criterion was mentioned in Section 2.1 as a way of deciding the optimal regularisation parameter. This section explains tests, where a sequence of points on the L-curve was generated for Tikhonov regularisa-

tion and the LSTRS method. The aim is to investigate the correspondence between the two methods in practice.

Since the two methods are parametrised differently, Tikhonov regularisation by δ and LSTRS by Δ, comparing solutions could be difficult. Fortunately the LSTRS Matlab routine returns the non-positive Lagrange multiplier λ of $(H + \lambda)x = -g$ as described in Section 3.1. The relationship between the regularisation parameter and Lagrange multiplier is $-\lambda = \delta^2$. The idea is then to solve a problem using LSTRS for different choices of Δ, resulting in corresponding values of λ. Subsequently Tikhonov regularisation based on the SVD is applied using the relation $\delta^2 = \lambda$. If the algorithms are numerically equivalent the results should be approximately the same for each of the two methods. By numerically equivalent we mean that the norm of the solutions and the norm of the residuals coincide to a large degree of precision.

The L-curve typically forms an L-shape when traced out (hence its name). Figure 5.2 shows the typical behaviour.

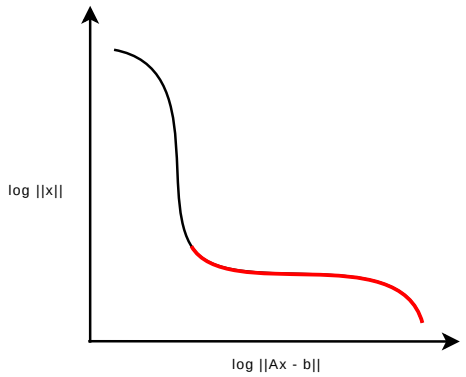


Figure 5.2: *L-curve behaviour. Upper left part corresponds to under-smoothing and lower right part corresponds to over-smoothing.*

Due to the parametrisation of the LSTRS method, the different λ returned will be unevenly distributed and this also applies to the solutions. Note that only the lower part of the L-curve is of interest here, which correspond to the boundary solution of the TRS. Although this boundary will be adjusted, it cannot be adjusted further than to where LSTRS starts to produce interior solutions.

Figure 5.3 shows a comparison of the two methods using the severely ill-posed Heat test problem for 100 point on the L-curve. Uniformly distributed noise on the order of 10^{-2} was added to the right hand side b and the external eigensolver routine was **eigs_lstrs_gateway**. Both methods produce approximately equal points. This can be seen on a smaller scale by examining the

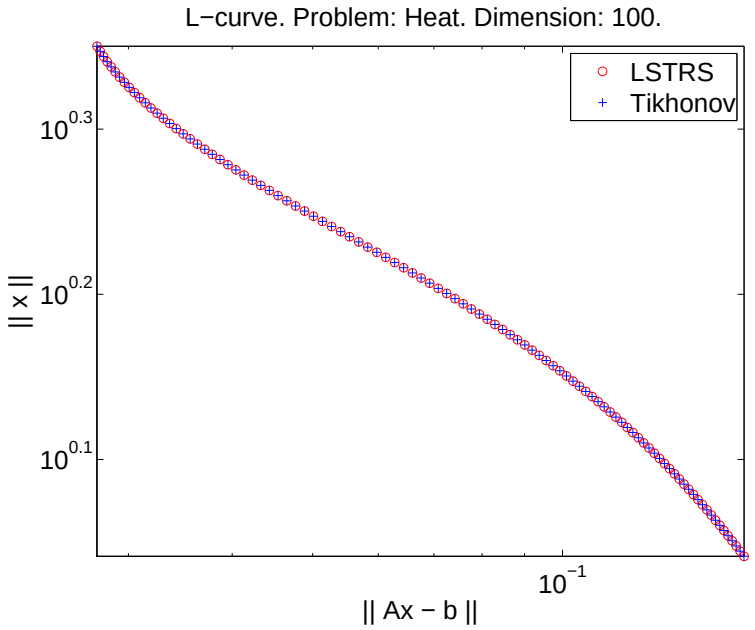


Figure 5.3: *A comparison between Tikhonov and LSTRS for the Heat problem. A blue cross denotes a point on L-curve for a Tikhonov solution, while a red circle corresponds to the LSTRS solution.*

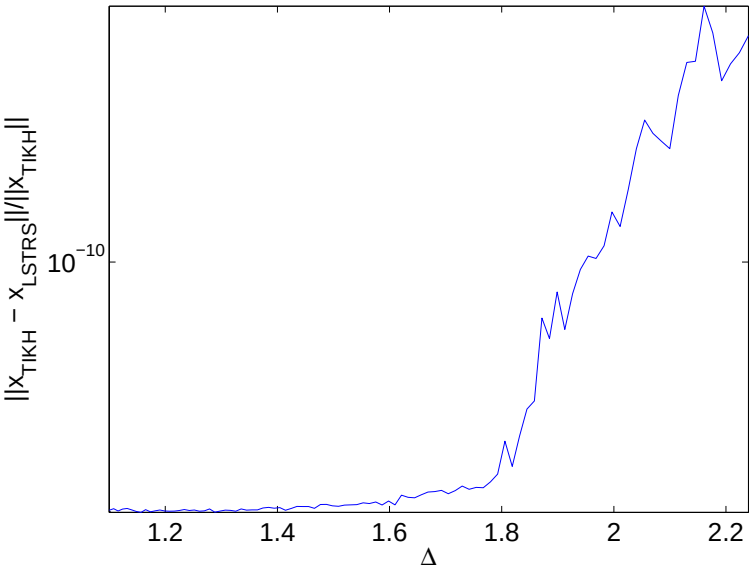


Figure 5.4: *Norm of the difference between the Tikhonov and the LSTRS solution for increasing* Δ.

norm of the difference in between the Tikhonov solutions and the LSTRS solutions. In Figure 5.4 this difference in solutions is compared against the increase in Δ, where Δ is approaching $\|x_{exact}\|$ from left to right. For small values of Δ the difference in solutions are close to machine accuracy (using double precision) and much more accurate than the required precision of the solution $\epsilon_\Delta = 10^{-4}$. As Δ increases, the difference between the Tikhonov solutions increase, but still stays one order below ϵ_Δ. By setting $\epsilon_\Delta = 10^{-8}$, the difference in solutions was decreased also for higher values of Δ.

5.3 Eigenpair sensitivity

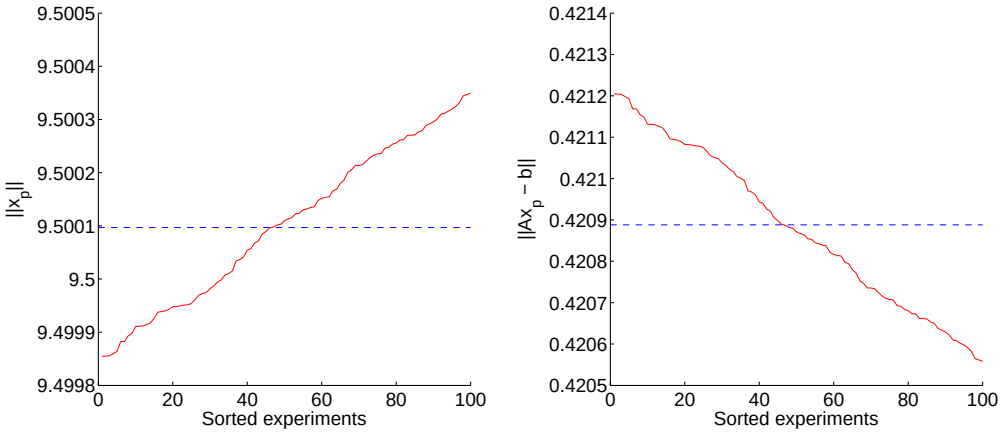
This section tests the sensitivity of LSTRS with respect to the eigenpair provided by the eigensolver. The LSTRS method needs two eigenpairs, where the first corresponds to the smallest eigenvalue of

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

and the second corresponds to an eigenvalue from a cluster of the same matrix. The second eigenpair is only needed when the hard case occurs. The eigenpairs are obtained from an external eigensolver. The interfaces provided by the Matlab code [23] are based on the Matlab functions **eig** and **eigs**, and an additional method using a Chebychev spectral transformation to improve convergence. In the following we restrict ourselves to the eigenpair corresponding to the smallest eigenvalue and to the eigensolvers **eig** and **eigs**.

The **eig** interface makes calls to LAPACK [1], while the **eigs** interface makes calls to ARPACK [14]. The difference between **eig** and **eigs** is that the former method gives nearly exact estimates of the eigenpair, while the latter applies the implicitly restarted Lanczos method (IRLM) and only gives estimates of the eigenpairs. Note that **eig** will not be feasible when the matrix is large, as it computes a complete eigenvalue decomposition of B_α.

The motivation behind the tests is that we want to assure that the solutions produced by the method are stable in the sense that the eigensolver used, need not produce very accurate eigenvalues. Given an inaccurate eigenvalue the LSTRS method should still be able to converge to a solution. The stability of the eigenvectors will also be tested. During tests the eigenvalue and corresponding eigenvector will be perturbed and the working hypothesis will be that if 90 percent of the perturbations do not exhibit dramatic change, the method will be concluded to be stable with respect to changes in the chosen eigenpair.



(a) *Perturbed solutions* **(b)** *Perturbed residuals*

Figure 5.5: *Perturbed solutions and residuals using* ***eig****. Horizontal lines denote* $\|x_u\|$ *and* $\|Ax_u - b\|$

Problems from Hansen's Regularization Tools [9] were used in the testing. We have chosen the problems; **shaw**, **heat** (ill-conditioned) and **phillips**. The right hand side, b, was perturbed by uniformly distributed random noise on the order of 10^{-2}, $b = b_{exact} + s$ and the problem matrix A is of dimensions 100×100. The choice of Δ was lowered from around $\|x_{exact}\|$ such that the method produced boundary solutions for all perturbations.

For the remainder of the section only results for the Shaw problem, where $\Delta = 9.5$ and $\|x_{exact}\| \approx 9.9820$, are presented. A hundred perturbations were made for each experiment.

5.3.1 Perturbing the smallest eigenvalue

In this experiment **eig_gateway.m** from the LSTRS package was modified to make it possible to perturb the eigenvalues returned from **eig**.

First a random vector of uniformly distributed numbers of order 10^{-2} was generated. The i-th component ξ_i of the random vector was added to the smallest eigenvalue value such that $\tilde{\lambda}_1 = \lambda_1 + \xi_i$, where $\tilde{\lambda}_1$ represents the perturbed eigenvalue. The problem was then solved for each perturbation.

Let x_u denote the unperturbed solution, while x_p denotes one of the perturbed solutions. In Figure 5.5 we take a look at the different $\|x_p\|$ and also the different $\|Ax_p - b\|$. The horizontal dashed blue line in each subplot marks the norm of the unperturbed solution and the unperturbed residual. To make the distribution of solutions and residuals clearer, the experiments have been sorted increasingly by $\|x_p\|$. Notice that this results in a decreasing

order for the residuals, which is in accordance with the behaviour of the L-curve. Observe that the changes in $\|x_p\|$, the computed norm of the solution is bounded by 10^{-4}.

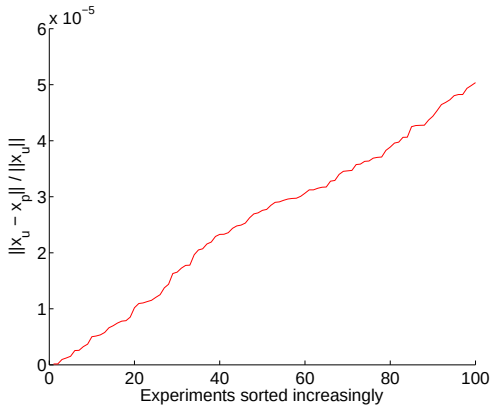


Figure 5.6: *Differences between the perturbed and the unperturbed solutions*

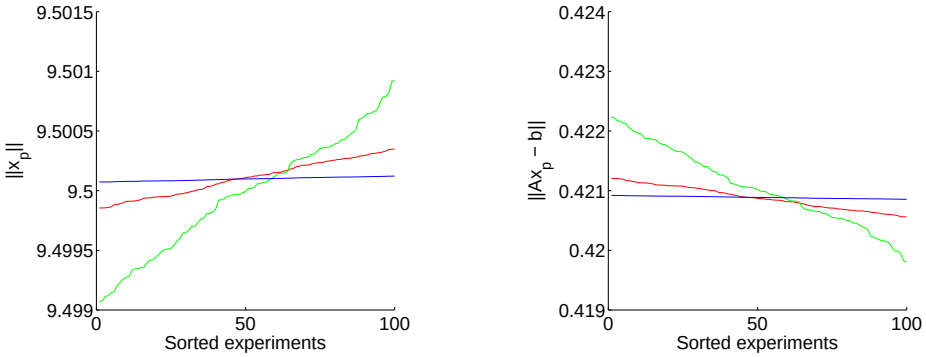
Next the relative errors between $\|x_u\|$ and $\|x_p\|$ of the different perturbations are compared in Figure 5.6. Observe that the relative error in all cases is on the order of 10^{-5}. In this figure and also in the previous figure the norm of the solutions and residuals are distributed linearly.

For the unperturbed solution 7 iterations of LSTRS were required, and also an average of 7 iterations for the perturbed values. For perturbations on the order of 10^{-1} of the smallest eigenvalue, the average was 8.68. The perturbations in the eigenvalues do not seem to have a major impact on the number of iterations.

As a second part of this experiment a graphical representation of the distribution of the perturbation was made. This can be seen in Figure 5.7, were perturbations on the order of 10^{-1}, 10^{-2} and 10^{-3} are coloured green, red and blue respectively. In this figure we see that in all the orders of perturbations the behaviour is similar. The distribution of solutions and residuals is nearly linear.

By modifying the file **eigs_lstrs_gateway.m**, test were carried out using **eigs**. Similar results were observed, with the exception of a few outliers. Still, over 90 percent of the perturbations turn out to have a linear behaviour. The results for the Shaw problem can be seen in Figure 5.8, which corresponds to Figure 5.5.

By using the iterative eigensolver **eigs** we are, in addition to the iteration count, able to get the total number of matrix vector products used. This number is used to measure the efficiency of iterative and matrix-free algorithms.



(a) *Perturbed solutions* **(b)** *Perturbed residuals*

Figure 5.7: *Changes in distribution of perturbations for several orders of magnitude.*

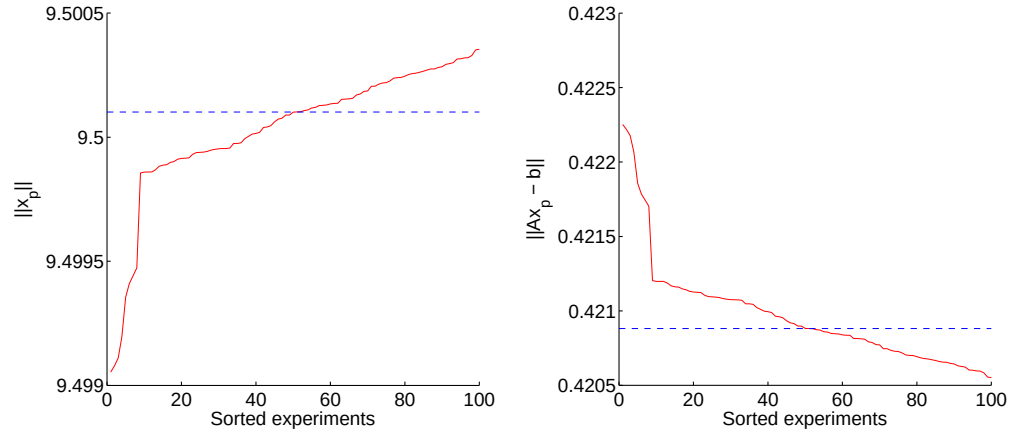
For x_u, 306 matrix vector products were needed, while on average for the different x_p, 267.15 matrix vector products were needed. Notice that the average is less than the number of matrix vector required by the unperturbed solution. This indicates that this particular x_u required more work than one of the perturbed ones. The LSTRS method does not seem to be overly affected by an imprecise eigenvalue from the eigensolver.

5.3.2 Perturbing the eigenvector of the smallest eigenvalue

In this section the smallest eigenvalue is left unperturbed, but the corresponding eigenvector is perturbed by a vector of uniformly distributed random noise. As LSTRS uses this vector in the final result (all but the first component makes up the solution), the last iteration directly perturbs the solution.

Note that due to a frequent occurrence of many small components in eigenvectors, only relative perturbation is considered. For the eigenvector q and the perturbation vector s we have that $\tilde{q} = q*(1+s)$, where the multiplication $(*)$ is meant componentwise and $\tilde{q}$ represents a perturbed eigenvector. Table 5.1 shows the growth in number of iterations for increasing order of perturbations. The number of iterations needed steadily increased as the order of perturbations increase.

We look closer at the norms of the different x_p and $Ax_p - b$ for a perturbation on the order of 10^{-3}, in Figure 5.9. Observe that the distribution of the solutions is similar to previous experiments, but the norms of the residuals



(a) *Perturbed solutions* **(b)** *Perturbed residuals*

Figure 5.8: *Perturbed solutions and residuals using* ***eigs****. Horizontal lines denote* $\|x_u\|$ *and* $\|Ax_u - b\|$

Order of perturbations	Avg. # of iterations
10^{-5}	7.0
10^{-4}	7.9
10^{-3}	11.5
10^{-2}	12.2

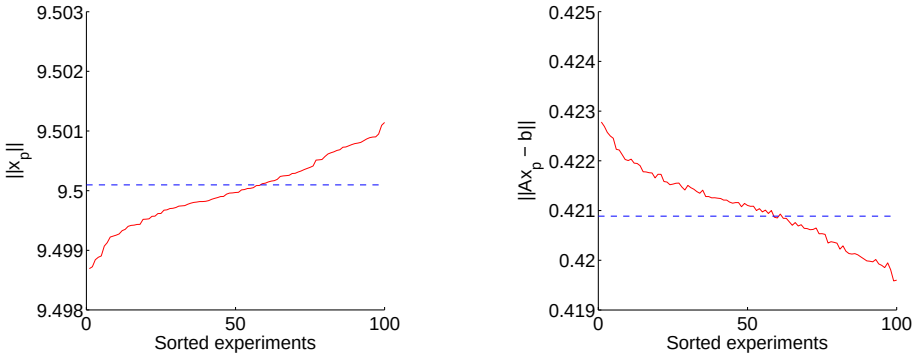
Table 5.1: *A comparison of order of perturbations and iteration count*

are starting to oscillate. The oscillations quickly get more frequent for larger perturbations.

When perturbing the smallest eigenvector using **eigs** similar observations as when using **eig** were made. Figures and further elaboration corresponding to **eigs** is therefore omitted.

5.4 Concluding remarks

In section 5.2 we saw an example of the correspondence between Tikhonov regularisation and the LSTRS method for boundary solutions. The methods can be said to be numerically equivalent for small values of Δ. The values of Δ when getting near to $\|x_{exact}\|$ corresponds to larger differences between the methods, although this can be improved by adjusting the tolerance ϵ_Δ. The larger Δ values correspond to where the TRS has potential hard cases, the increasing non-linearity of the secular equation (3.6) makes it harder to



(a) *Solutions* **(b)** *Residuals*

Figure 5.9: *Solutions and residuals for eigenvector perturbations using* ***eig***

solve. Still LSTRS is able to solve the problems to high degree of precision when compared with the SVD-based Tikhonov regularisation. During these tests we have looked into perturbations of the smallest eigenvalue and its corresponding eigenvector.

The smallest eigenvalue given by the eigensolver does not seem crucial to a good result of the LSTRS method. A good estimate will decrease the number of iterations, but not heavily so.

The eigenvector is more sensitive to perturbations, especially absolute perturbations as this will make zero or small components change drastically. Using relative perturbations, mostly large components of the eigenvector were perturbed. This test showed that as long as small components are kept small, perturbations of the large components do not need to be overly accurate. Perturbation of the eigenvector also affects the number of iterations needed by LSTRS to converge.

As mention, the test problems **heat** and **phillips** were also tested, giving similar results. Thus the figures presented show a trend found in several problems and is not due to special properties of the Shaw problem.

In [21] and Paper B in [3] and in Section 6.4, LSTRS is used in the innermost loop with a repeated number of solves. It is therefore important to know that the method is robust with respect to inaccurate solutions.

Chapter 6

Regularisation with unrestricted variables

This chapter describes a method which allows for regularisation with some variables left unrestricted. Section 6.1 formulates the problem and also presents a reformulation, while in Section 6.2 a more general problem is presented. Section 6.3 shortly describes a strategy for solving constrained problems. Finally a method for the general problem is presented in Section 6.4.

6.1 The problem

The problem of interest is a least-squares problem where some components of the solution are restricted to be within a closed ball Δ, while other components remain unrestricted. Specifically the aim is to

$$\begin{aligned} \min_{x,y} \ & \frac{1}{2}\| \begin{bmatrix} A & B \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} - b\|^2 \\ \text{subject to } & \|x\| \leq \Delta, \end{aligned} \tag{6.1}$$

where $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times p}$ and full rank, $b \in \mathbb{R}^m$ and $\Delta > 0$. The unknowns are $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^p$ and m, n and p are positive integers such that $m \geq p$.

By formulating the problem in this way, it can be interpreted statistically as a partially biased problem. Because Tikhonov regularisation imposes a priori knowledge on the solution it can be shown to be biased, while a weighted least squares method can be shown to be unbiased [2, p. 21]. Notice also that

if $p = 0$ the problem reduces to the one of Tikhonov regularisation, covered in Section 2.1.

To simplify notation, the objective function is restated as

$$f(x,y) = \frac{1}{2}\| \begin{bmatrix} A & B \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} - b\|^2 = \frac{1}{2}\|Ax + By - b\|^2. \tag{6.2}$$

It follows that

$$\begin{aligned} f(x,y) &= \frac{1}{2}(Ax + By - b)^T(Ax + By - b) \\ &= \frac{1}{2}x^TA^TAx + \frac{1}{2}x^TA^TBy - \frac{1}{2}x^TA^Tb \\ &+ \frac{1}{2}y^TB^TBy + \frac{1}{2}y^TB^TAx - \frac{1}{2}y^TB^Tb \\ &- \frac{1}{2}b^TAx - \frac{1}{2}b^TBy + \frac{1}{2}b^Tb. \end{aligned}$$

The terms x^TA^TBy, x^TA^Tb and y^TB^Tb can be combined with their respective transpose term, because $\beta = \beta^T$ for any scalar β. The objective function is therefore simplified, resulting in

$$\begin{aligned} f(x,y) = \frac{1}{2}x^TA^TAx - x^TA^Tb + \frac{1}{2}b^Tb \\ + y^TB^TAx - y^TB^Tb + \frac{1}{2}y^TB^TBy. \end{aligned}$$

Finally, we get

$$f(x,y) = \frac{1}{2}\|Ax - b\|^2 + y^T(B^TAx + \frac{1}{2}B^TBy - B^Tb).$$

Observe that given a solution (x^*, y^*) of (6.1), y^* must be a solution of

$$\min_y y^T(B^TAx^* + \frac{1}{2}B^TBy - B^Tb).$$

This is an unconstrained quadratic problem. In general the matrix B^TB is positive semi-definite and y^* must satisfy the linear system

$$B^TBy^* = B^Tb - B^TAx^*$$

Since only B with full rank is considered here, B^TB it is positive definite and the solution is unique.

If B has less than full rank, the problem can be reduced by eliminating the columns in B that are linear dependent on the others. In this case a

matrix decomposition is needed, which limits the size of B in a large-scale perspective. The result of such a reduction is a decrease in the number of unrestricted variables needed; the number of components in y^*.

The solution for a given x^* is $y^* = -(B^TB)^{-1}B^T(Ax^* - b)$ and for any x, the minimal y can be uniquely determined by this expression. By substituting for y in (6.2) we obtain

$$\frac{1}{2}\|Ax - B(B^TB)^{-1}B^T(Ax - b) - b\|^2.$$

Rearranging the terms and setting $\hat{P} = B(B^TB)^{-1}B^T$ results in

$$\frac{1}{2}\|(Ax - b) - \hat{P}(Ax - b)\|^2,$$

and finally by factorising we have

$$\frac{1}{2}\|P(Ax - b)\|^2, \tag{6.3}$$

where $P = (I - \hat{P})$. Note that P is a projector [27, Lecture 6], which means it has the property that $P = P^2$.

The reformulation of the objective function suggests a connection with the following problem,

$$\begin{aligned} \min_{x,y} \ & \frac{1}{2}\|\hat{A}x - \hat{b}\|^2 \\ \text{subject to } & Cx + Dy = d \\ & \|x\| \leq \Delta \end{aligned} \tag{6.4}$$

where $C \in \mathbb{R}^{\hat{m}\times n}$, $D \in \mathbb{R}^{\hat{m}\times p}$ and $d \in \mathbb{R}^{\hat{m}}$ and $\hat{m} \geq p$.

This is a linear equality constrained quadratic problem with a trust-region constraint. Note that the unknown y only appears in the equality constraints.

To compare a solution to (6.1) with a solution to (6.4) we need the first-order necessary conditions, also known as the KKT-conditions [17, Theorem 12.1]. Since the Lagrangian function is central to the KKT-conditions it is included for completeness. For consistency with earlier chapters, the non-standard nonpositive Lagrange multiplier is used for inequality constraints.

The Lagrangian of (6.1) is

$$\mathfrak{L}(x, y, \lambda) = \frac{1}{2}\|Ax - b\|^2 - y^T(B^TAx + \frac{1}{2}B^TBy - B^Tb) - \lambda(\|x\|^2 - \Delta^2),$$

and the KKT-conditions are

$$A^TAx^* + A^TBy^* - \lambda x^* = A^Tb \tag{6.5a}$$
$$B^TAx^* + B^TBy^* = B^Tb \tag{6.5b}$$
$$\|x^*\| \leq \Delta \tag{6.5c}$$
$$\lambda(\|x^*\|^2 - \Delta^2) = 0, \tag{6.5d}$$

where $\lambda \leq 0$.

Next we consider (6.4). Its Lagrangian is

$$\mathfrak{L}(x, y, z, \lambda) = \frac{1}{2}\|\hat{A}x - \hat{b}\|^2 - z^T(Cx + Dy - d) - \lambda(\|x\|^2 - \Delta^2)$$

and its KKT-conditions are

$$\hat{A}^T\hat{A}x^* - \lambda x^* - C^Tz = \hat{A}^T\hat{b} \tag{6.6a}$$
$$Cx^* + Dy^* = d \tag{6.6b}$$
$$\|x^*\| \leq \Delta \tag{6.6c}$$
$$\lambda(\|x^*\|^2 - \Delta^2) = 0 \tag{6.6d}$$
$$D^Tz = 0 \tag{6.6e}$$

where $z \in \mathbb{R}^{\hat{m}}$ and $\lambda \leq 0$.

When setting $C = B^TA$, $D = B^TB$ and $d = B^Tb$ into the conditions above, $z = 0$, since B^TB has a trivial null space. As a result of this the equation (6.6e) equals zero and the term involving z in (6.6a) vanishes. Comparing what is left with the KKT conditions of (6.1), we see that only (6.6a) appears to differ; it lacks the A^TBy^* term.

By choosing $\hat{A} = PA$ and $\hat{b} = Pb$ the KKT conditions for both problems are the same. This is clear by observing that these choices result in the objective function (6.3).

To summarise, we have that by proper choice of the known values in (6.4) the solutions of the two problems, (6.1) and (6.4) coincide. Thus a problem on the form (6.1) can be solved by an algorithm solving (6.4) and vice versa.

6.2 The general problem

In this section a problem incorporating both problems from the previous section, is introduced.

Consider the following problem,

$$\begin{aligned} \min_{x,y} \ & \frac{1}{2}\|Ax + By - b\|^2 \\ \text{subject to } & Cx + Dy = d \\ & \|x\| \leq \Delta. \end{aligned} \tag{6.7}$$

where $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times p}$, $C \in \mathbb{R}^{\hat{m}\times \hat{n}}$, $D \in \mathbb{R}^{\hat{m}\times \hat{p}}$ and $b \in \mathbb{R}^m$, $d \in \mathbb{R}^{\hat{m}}$. Given an algorithm for this problem we can solve problems on the forms (6.1) and (6.4) for the right choices of coefficient matrices. If C and D are zero matrices of appropriate dimensions, we get a problem on the same form as (6.1). If B is a zero matrix or B has full rank, we can get the problem on the form of (6.4). In addition, if both B and D are zero matrices or if all components of the solution are restricted by Δ, problems on the form

$$\begin{aligned} \min_{x} \ & \frac{1}{2}\|Ax - b\|^2 \\ \text{subject to } & Cx = d \\ & \|x\| \leq \Delta \end{aligned}$$

can also be solved. Note that to solve all these cases an algorithm must handle situations where B, C and D are zero matrices and the case where B has full rank.

The reformulated problem (6.4) is not applicable for large-scale problems as the matrices in its objective function are on a special form and the number of variables in the equality constraint and the objective function differ. On the other hand, since both problems from Section 6.1 can be solved by (6.7), a combination might solve the problem more efficiently.

Let us now look at the equality constrained quadratic problem

$$\begin{aligned} \min_{x} \ & \psi(x) = \frac{1}{2}x^T Hx + g^T x \\ \text{subject to } & Cx = d \end{aligned} \tag{6.8}$$

where $H \in \mathbb{R}^{n\times n}$ and symmetric and $g \in \mathbb{R}^n$. The problem has the following necessary conditions

$$\begin{aligned} Hx^* &= -g + C^T z^* \\ Cx^* &= d \end{aligned} \tag{6.9}$$

where $z^* \in \mathbb{R}^n$ denotes Lagrange multipliers. Problem (6.7) fits into this problem if the norm constraint is ignored. The Lemma 6.2.1 presents some properties of the equality constrained quadratic problem.

Lemma 6.2.1. *Let Z be a matrix of full rank such that $CZ = 0$.*

(i) Let x^* and z^* be a solution of (6.9) and x^* a local solution of (6.8), then x^* is also a global solution.

(ii) Let x^* and z^* be a solution of (6.9) and Z^THZ be positive semi-definite, then x^* is a global solution.

(iii) If x^* and z^* satisfy (6.9) and x^* is a local solution, then Z^THZ is positive semi-definite.

Proof. In the first statement (i) we have that x^* is a local solution. The second order necessary conditions requires that H must be positive semi-definite. The objective function is thus convex at x^*. The constraints are also convex, which implies that any local solution x^* must be a global solution.

For (ii), let x be any feasible point satisfying $Cx = d$ and let $p = x^* - x$. $Cx^* = Cx = d$ implies that $Cp = 0$. Substituting for x in the objective function gives

$$\begin{aligned}\psi(x) &= \frac{1}{2}(x^* - p)^T H(x^* - p) + g^T(x^* - p) \\ &= \psi(x^*) - p^T Hx^* + \frac{1}{2}p^T Hp - g^T p.\end{aligned}$$

Replacing for the first condition in (6.9) results in

$$\begin{aligned}\psi(x) &= \psi(x^*) - p^T(-g + C^T z^*) + \frac{1}{2}p^T Hp - g^T p \\ &= \psi(x^*) - (Cp)^T z + \frac{1}{2}p^T Hp.\end{aligned}$$

Since $Cp = 0$ we have

$$\psi(x) = \psi(x^*) + \frac{1}{2}p^T Hp$$

and p can be written $p = Zu$ because it lies in the null space of C, where Z is a basis for the null space.

$$\psi(x) = \psi(x^*) + \frac{1}{2}u^T Z^T HZu \tag{6.10}$$

Since Z^THZ is assumed to be positive semi-definite $\psi(x) \geq \psi(x^*)$ for all x, which proves (ii).

For the last part (iii), we can also arrive at (6.10). Now assume that Z^THZ is not positive semi-definite. Then $\frac{1}{2}u^T Z^T HZu < 0$, which implies that there exists an x with a smaller function value. This is a contradiction of the assumption that x^* is a local solution, thereby concluding the proof. □

If we were to add trust-region constraint $\|x\| \leq \Delta$ to (6.9) there is a possibility that the linear equality constraints are inconsistent with respect to the size of the solution. That is, there might not be a point x satisfying both the linear equality constraints and the trust-region constraint. When the trust-region is applied to a subset of the components of x we have (6.7).

The first order necessary conditions of (6.7) are

$$\begin{aligned}
\begin{bmatrix} A^TA & A^TB \\ B^TA & B^TB \end{bmatrix} \begin{bmatrix} x^* \\ y^* \end{bmatrix} - \lambda^* \begin{bmatrix} x^* \\ 0 \end{bmatrix} - \begin{bmatrix} C^T \\ D^T \end{bmatrix} z &= \begin{bmatrix} A^T \\ B^T \end{bmatrix} b \\
\begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} x^* \\ y^* \end{bmatrix} &= d \\
\|x^*\| &\leq \Delta \\
\lambda^*(\|x^*\| - \Delta^2) &= 0
\end{aligned}$$

where $z^* \in R^{\hat{m}}$ and $\lambda^* \leq 0$ are Lagrange multipliers (dual variables).

Due to the properties of the linear equality constrained quadratic problem given in Lemma 6.2.1 together with the properties of the TRS solution given in Theorem 2.2.1, it seems probable that the necessary conditions (6.7) also are sufficient.

Conjecture 6.2.2. *The necessary conditions for* (6.7) *are also sufficient.*

Recall that the aim is to use LSTRS as a subroutine in an algorithm, which means that a subproblem must be on the form of a TRS. Certainly, for (6.7) this is not the case, as it has an additional linear equality constraint. A strategy for avoiding constraints is to include additional terms in the function that penalise directions where the constraints are not satisfied. This strategy is explained in more detail in the next section.

6.3 Penalty functions

Penalty functions include constraints as terms in the objective function. In this section the general problem (6.7) is reformulated as a penalty problem. For simplicity the norm constraint is ignored.

Several choices are available when choosing a penalty function. The main three classes are: quadratic, augmented Lagrangian and nonsmooth. See [17, Chapter 17] or [5, Chapter 14] for an overview. Here the focus will be on quadratic and nonsmooth penalty functions.

Penalty functions introduce a penalty parameter $\nu > 0$, which is used to adjust the weighting of the penalty terms. Quadratic penalty functions are

smooth and are basically made up of an additional quadratic term which in the case of problem (6.7) results in,

$$Q(x, y; \nu) = \frac{1}{2}\|Ax + By - b\|^2 + \frac{\nu}{2}\|Cx + Dy - d\|^2.$$

Nonsmooth penalty functions have the favourable property of being *exact*. Here exact means that, for a particular choice of ν, a single minimisation of the penalty function can result in the exact solution to the non-linear programming problem [17, Section 17.2]. Typical choices for nonsmooth penalty functions include the l_1, l_2 (not squared) and l_∞-norms (see Definition 1 in Appendix A), where the l_1 is reported [17, p. 507] to be a common choice.

For the objective function in (6.7) the l_1 penalty function is

$$Q(x, y; \nu) = \frac{1}{2}\|Ax + By - b\|^2 + \nu\|Cx + Dy - d\|_1. \tag{6.11}$$

The exactness comes at the cost of l_1-norm not being differentiable (nonsmooth). It does however have a directional derivative along any direction. Constrained optimisation methods also use exact functions such as (6.11) as *merit functions*. Merit functions are used for step acceptance. Only a sufficient reduction in the merit function will lead to the acceptance of the step and thereby the progression of the algorithm. If the step is not accepted adjustment of the penalty parameter may be needed. By requiring a reduction in the merit function global convergence is assured.

6.4 A scaling algorithm

As noted in previous sections, problem (6.7) only has a norm restriction on x. Let us look at the problem where the norm of y is also constrained by Δ,

$$\begin{aligned} \min_{x,y} \ &= \frac{1}{2}\|Ax + By - b\|^2 \\ \text{subject to} \ & Cx + Dy = d \\ & \left\| \begin{bmatrix} x \\ y \end{bmatrix} \right\| \leq \Delta. \end{aligned} \tag{6.12}$$

Apart from the linear quality constraints this problem is on the form of a TRS. Observe that if $\|y\|$ can be kept small compared to $\|x\|$, the contribution of the components of y may be negligible.

The idea is to introduce a scaling parameter $\sigma > 0$ and apply this to y such that $\hat{y} = \sigma y$ and let $\sigma \searrow 0$. Note that to keep the same problem, B

and D need to be scaled by the reciprocal of σ. This results in the problem,

$$\begin{aligned} \min_{x,y} \ & \frac{1}{2}\|Ax + \frac{1}{\sigma}B\hat{y} - b\|^2 \\ \text{subject to } \ & Cx + \frac{1}{\sigma}D\hat{y} = d \\ & \| \begin{bmatrix} x \\ \hat{y} \end{bmatrix} \| \leq \Delta. \end{aligned}$$

Now the linear equality constraints need to be handled. Applying the quadratic penalty strategy from Section 6.3 results in,

$$\begin{aligned} \min_{x,y} \ & \frac{1}{2}\|Ax + \frac{1}{\sigma}B\hat{y} - b\|^2 + \frac{\nu}{2}\|Cx + \frac{1}{\sigma}D\hat{y} - d\|^2 \\ \text{subject to } \ & \| \begin{bmatrix} x \\ \hat{y} \end{bmatrix} \| \leq \Delta. \end{aligned} \tag{6.13}$$

where ν is the penalty parameter.

To develop an algorithm we estimate the objective function of (6.13) around a given iterate $(x^T, y^T)^T$ by using a quadratic model. First we simplify notation by introducing the following

$$\hat{A} = \begin{bmatrix} A & \frac{1}{\sigma}B \end{bmatrix} \quad \hat{C} = \begin{bmatrix} C & \frac{1}{\sigma}D \end{bmatrix} \quad \hat{x} = \begin{bmatrix} x \\ \hat{y} \end{bmatrix}$$

Furthermore we have that

$$\begin{aligned} f(\hat{x}) &= \frac{1}{2}\|\hat{A}\hat{x} - b\|^2 + \nu c(\hat{x}) \\ \nabla f(\hat{x}) &= \hat{A}^T\hat{A} - \hat{A}^T b + \nu \nabla c(\hat{x}) \\ \nabla^2 f(\hat{x}) &= \hat{A}^T\hat{A} + \nu \nabla^2 c(\hat{x}) \\ c(\hat{x}) &= \frac{1}{2}\|\hat{C}\hat{x} - d\|^2 \\ \nabla c(\hat{x}) &= \hat{C}^T\hat{C}\hat{x} - \hat{C}^T d \\ \nabla^2 c(\hat{x}) &= \hat{C}^T\hat{C}. \end{aligned}$$

The quadratic approximation is written as

$$q(\hat{x} + h) = f(\hat{x}) + \nabla f(\hat{x})^T h + \frac{1}{2}h^T \nabla^2 f(\hat{x}) h.$$

When minimising this function, constant terms $f(\hat{x})$ and $c(\hat{x})$ can be ignored and the problem becomes

$$\begin{aligned} \min_{\hat{x}+h} \ & p(\hat{x} + h) = \frac{1}{2}(\hat{x} + h)^T H(\hat{x} + h) + (\hat{x} + h)^T g \\ \text{subject to } \ & \|\hat{x} + h\| \leq \Delta. \end{aligned}$$

Algorithm 1: Backtracking subroutine

Data: x, h, ν
Result: α
$\alpha = 1$, $\rho \in (0,1)$, $\gamma_h \in (0,1)$
while $m(x + \alpha h; \nu) \leq m(x; \nu) + \gamma_h \alpha \nabla m(x; \nu)^T h$ **do**
    $\alpha = \rho\alpha$
end
$\alpha_k = \alpha$

where

$$H = \hat{A}^T\hat{A} + \nu\hat{C}^T\hat{C} \quad \text{and} \quad g = -\hat{A}^T b - \nu\hat{C}^T d$$

By setting $z = x + h$, we get the TRS problem

$$\begin{aligned} \min_z \frac{1}{2} z^T H z + z^T g \\ \text{subject to } \|z\| \leq \Delta. \end{aligned} \tag{6.14}$$

As mentioned in Section 6.3, l_1 penalty functions are also applicable as merit functions. This is precisely the aim for our algorithm. Let

$$m(\hat{x}; \nu) \ = \frac{1}{2}\|\hat{A}\hat{x} - b\|^2 + \nu\|\hat{C}\hat{x} - d\|_1 \tag{6.15}$$

be the merit function. We want to ensure that a sufficient decrease of (6.15) occurs at each iteration. It is important that the scaling of x_k and the matrices in the penalty function are equal when the function is evaluated.

To decide if the new point $\hat{x}_+$ results in sufficient decrease, a backtracking approach is used. As the l_1 norm is not differentiable at points $\hat{x}$ where $\hat{C}\hat{x} - d = 0$, this may lead to difficulties. Let us first ignore this situation and suggest a backtracking subroutine.

From line search theory [17, Section 3.1] we have the backtracking algorithm, see Algorithm 1. The backtracking approach begin by checking if the whole step $\hat{x} + h$ ensures sufficient decrease. If not, it cuts the step short to some fraction ρ of the previous step. We will allow $\rho \in [\rho_{min},\ \rho_{max}]$ where $0 < \rho_{min} \leq \rho_{max} < 1$. ρ_{min} and ρ_{max} are lower and upper limits of ρ that can be adjusted at each iteration.

The sufficient decrease condition is

$$m(\hat{x}_+; \nu_k) \leq m(\hat{x}_k; \nu_k) + \gamma_h \alpha \nabla m(\hat{x}_k; \nu_k)$$

where $\hat{x}_+ = \hat{x}_k + \alpha h$ and $\gamma_h \in (0,1)$. This implies that the gradient of $m(\hat{x}_k; \nu_k)$ is needed.

The first part of the gradient is straight forward,

$$\nabla m(\hat{x};\nu) = \hat{A}^T\hat{A} - \hat{A}^T b + \nu\nabla l(\hat{x})$$

but the second part, here denoted by $l(\hat{x}) = \|\hat{C}\hat{x}-d\|_1$ requires more thought.

In the following the subscript k denotes the k-th component of a vector. Since the aim is to evaluate the function, we can write the k-th partial derivative of $\hat{x}$ as

$$\frac{\partial l(\hat{x})}{\partial \hat{x}_k} = \sum_{i=1}^{\hat{m}} \frac{\partial |c_i(\hat{x})|}{\partial \hat{x}_k} = \sum_{i=1}^{\hat{m}} \hat{C}_{ik}\frac{c_i(\hat{x})}{|c_i(\hat{x})|}$$

where $c_i(\hat{x}) = \sum_{j=1}^{n+p} C_{ij}\hat{x}_j - d_i$.

The components for which (6.15) is not differentiable correspond to the case when the constraint is satisfied and they should therefore not contribute to the direction and may be set equal to zero.

To express the gradient on matrix form we can write

$$\nabla l(\hat{x}) = \hat{C}^T\Xi(\hat{C}\hat{x} - d)$$

where $\Xi = diag(1/|c_1(\hat{x})|, 1/|c_2(\hat{x})|, \ldots, 1/|c_m(\hat{x})|)$.

Recall that if $c_i(x) = 0$ the derivative does not exist and this corresponds to the case where get a division by zero in Ξ. From a computational point of view this can be avoided by first evaluating $C\hat{x}-d$, and when constructing Ξ a zero can be introduced if $c_i(\hat{x}) = 0$ occurs. This way of constructing $\nabla l(\hat{x})$ comes from the iteratively restarted least squares (IRLS) method described in [2, Chapter 2, Section 4].

In summary, $\nabla m(\hat{x};\nu)$ can be constructed by the procedure outlined above and a sufficient decrease condition can be used ensure decrease in the merit function.

Algorithm 2 gives an overview of the scaling approach. Note that $\{\nu_k\}, \nu_k \to \infty$ and $\{\sigma_k\}, \sigma_k \to 0$.

6.4.1 Choice of initial values

An initial x_0 can be obtained by solving (6.12). In other words, solving the problem without scaling and penalty function. To get a reasonably feasible y_0, we solve

$$\min_y \|Dy + Cx_0 - d\|. \tag{6.16}$$

For regularisation problems with unrestricted variables B is assumed to have full rank, and $D = B^TB$ is thereby symmetric positive definite and (6.16)

Algorithm 2: Scaling algorithm

Data: A, B, C, D, b, d, $\Delta > 0$, ϵ_f, ϵ_x, ϵ_ν
Result: x^*, y^*
Choose x_0, y_0, $\nu_0 > 0$, $\sigma_0 \in (0, 1]$
$\hat{x}_0 = (x_0^T, y_0^T)^T$
$k \leftarrow 0$
while *not converged* **do**
    Scale B and D by $1/\sigma_k$
    Solve (6.14) for z
    Adjust scaling of x_k to match z
    $h \leftarrow z - \hat{x}_k$
    Compute α using Algorithm 1
    $\hat{x}_{k+1} \leftarrow \hat{x}_k + \alpha h$
    if $\|\hat{C}\hat{x}_k - d\| \geq \epsilon_\nu$ **then**
        Compute ν_{k+1}
    else
        $\nu_{k+1} \leftarrow \nu_k$
    end
    Compute σ_{k+1}
    $k \leftarrow k + 1$
end
$x^* \leftarrow \hat{x}_k(1 : n)$
$y^* \leftarrow \hat{x}_k(n + 1 : n + p)/\sigma_k$

can be solved by a CG method if the number of variables is large. Otherwise a direct method will suffice.

Initial values for σ and ν will depend on the problem, and are chosen in an ad-hoc manner.

6.4.2 Updating strategy

The penalty parameter ν_k is updated a constant multiple as long as the feasibility of $\|\hat{C}\hat{x} - d\|_1 \geq \epsilon_\nu$, where ϵ_ν is a given tolerance. The update constant is chosen such that $\gamma_\nu > 0$, resulting in the update rule

$$\nu_{k+1} = \gamma_\nu \nu_k.$$

The scaling parameter σ_k need to be decreased at each iteration and the update is chosen to be constant multiple $\gamma_\sigma \in (0, 1)$. This results in the following update rule,

$$\sigma_{k+1} = \gamma_\sigma \sigma_k.$$

6.4.3 Stopping conditions

The conditions for convergence are based on the change in (6.15) and the change between the previous iterate and the current.

$$|m(\hat{x}_k; \nu_k) - m(\hat{x}_{k-1}; \nu_{k-1})| \leq \epsilon_f |m(\hat{x}_k; \nu_k)|$$
and
$$\|\hat{x}_k - \hat{x}_{k-1}\| \leq \epsilon_x \|\hat{x}_k\|$$

The tolerances ϵ_f and ϵ_x need to be supplied by the user.

Chapter 7

Numerical results

In this chapter two applications for (6.2) are presented. Numerical results for a prototype implementation are also given. The tests were carried out in Matlab version 7.5. In addition routines from the Matlab packages of Hansen [9] and Rojas et al [23] were used.

In Section 7.1 an application in model predictive control is presented, while Section 7.2 covers misalignment and image deblurring.

7.1 Model predictive control: The four-tank system

Model predictive control (MPC) is a part of the larger field of process control. Given a model of a facility and measurements of its current state, the aim is to predict how changes in input, for instance the effect of a pump, will affect the state of the system and use this prediction to drive the system towards a *steady state*. At given intervals a so-called *finite horizon* control problem is solved. Finite horizon means that the prediction model is sampled a finite number of time steps. The current measured state is used as starting point. The control problem is commonly solved by formulating it as a quadratic programming (QP) problem, see [19] for an overview of the design process. Only the first time step is used for adjusting the actual system. At the following interval the system state is checked again and the control problem is solved using the current state as the initial state.

The four-tank system, also known as the quadruple-tank process, is due to Johannsson [13] and is used in several courses in process control. Figure 7.1 gives a schematic overview. The test setup consists of four interconnected tanks, each with a given level of liquid, two valves and two pumps. When voltage is applied to a pump, liquid is distributed to the connected tanks

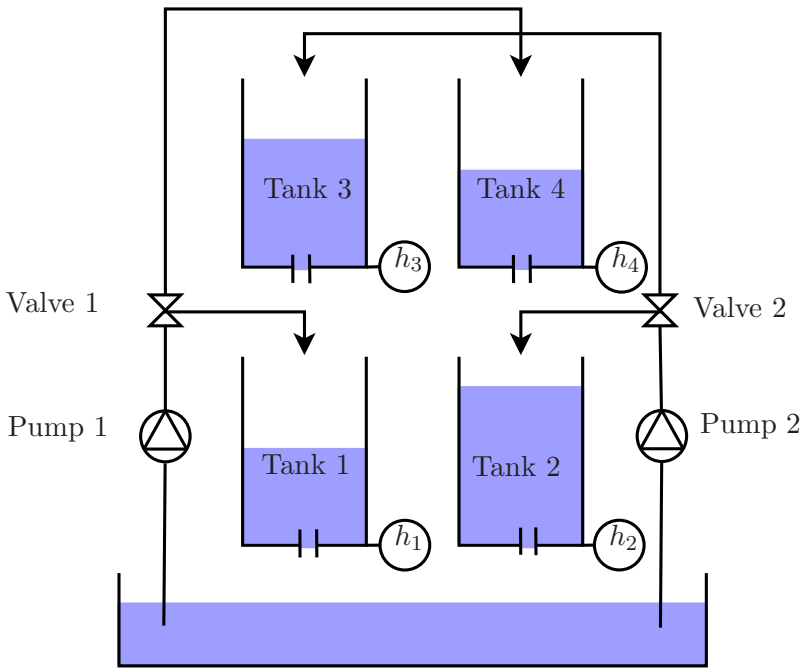


Figure 7.1: *The four-tank system consists of four tanks, two pumps and two valves. The circles denoted h_1, h_2, h_3 and h_4 represent measured values.*

via a valve. Pump 1 distributes liquid to Tank 1 and Tank 4, while Pump 2 distributes liquid to Tank 2 and Tank 3.

The objective is to stabilise the levels in Tank 1 and Tank 2 by adjusting the pump voltages. In our case one pump is uncertain and may give inaccurate predictions. To improve the certainty of predictions a norm constraint is imposed, penalising the size of the predictions. The second pump is precise and is therefore not in need of additional restrictions. The relevance to our problem (6.1) should now be clear. One part of the problem needs regularisation, while the other is well-behaved.

As mentioned these problems are commonly modelled as a QP. In such programs the norm constraint is usually in the l_∞ norm. The choice of this norm is believed to be mainly due to be its ease of fit into the QP framework. As the l_2 norm fits into our problem, this can be used instead. Note that this may also make an interesting comparison between algorithms.

7.1.1 Description of model

The four-tank system can be described by the nonlinear model

$$\begin{aligned}
\frac{dh_1}{dt} &= \frac{\gamma_1}{A_1}F_1 + \frac{a_3}{A_1}\sqrt{2gh_3} - \frac{a_1}{A_1}\sqrt{2gh_1} \\
\frac{dh_2}{dt} &= \frac{\gamma_2}{A_2}F_2 + \frac{a_4}{A_2}\sqrt{2gh_4} - \frac{a_2}{A_2}\sqrt{2gh_2} \\
\frac{dh_3}{dt} &= \frac{1-\gamma_2}{A_3}F_2 - \frac{a_3}{A_3}\sqrt{2gh_3} \\
\frac{dh_4}{dt} &= \frac{1-\gamma_1}{A_4}F_1 - \frac{a_4}{A_4}\sqrt{2gh_4}
\end{aligned} \tag{7.1}$$

where a_i, $i = 1, 2, 3, 4$ is the area of each of the four pipes and A_i, $i = 1, 2, 3, 4$ is the cross sectional area of each of the four tanks. The flow distribution constants, for each of the two valves, are represented by γ_1 and γ_2 and g is the acceleration of gravity. F_1 and F_2 corresponds to the voltage applied to each of the two pumps. Given F_1^s and F_2^s, the steady state or the stationary point, is found by solving the equations in (7.1) set to zero, giving $x^s = (h_1^s, h_2^s, h_3^s, h_4^s)^T$, where the superscript s denotes the steady state. Also the following steady state variables are needed,

$$u^s = \begin{bmatrix} F_1^s \\ F_2^s \end{bmatrix} \quad \text{and} \quad z^s = \begin{bmatrix} h_1^s \\ h_2^s \end{bmatrix}.$$

The linearised version of the system is given by

$$\begin{aligned}
\frac{dx}{dt} &= A_c x(t) + B_c u(t) \\
z(t) &= Cx(t)
\end{aligned}$$

where

$$A_c = \begin{bmatrix} -\frac{1}{T_1} & 0 & \frac{A_3}{A_1 T_3} & 0 \\ 0 & -\frac{1}{T_2} & 0 & \frac{A_4}{A_2 T_4} \\ 0 & 0 & -\frac{1}{T_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T_4} \end{bmatrix} \qquad B_c = \begin{bmatrix} \frac{\gamma_1}{A_1} & 0 \\ 0 & \frac{\gamma_2}{A_2} \\ 0 & \frac{1-\gamma_2}{A_3} \\ \frac{1-\gamma_1}{A_4} & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

and $T_i = \frac{A_i}{a_i}\sqrt{\frac{2h_i^s}{g}}$ for $i = 1, \ldots, 4$. $x(t)$ is the previous tank levels and $u(t)$ denotes the previous voltage levels. Note that these matrices do *not*

correspond to the ones for problem (6.1). A discrete-time linear system

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ z_k &= Cx_k \end{aligned}$$

can be constructed by computing

$$\begin{aligned} A &= e^{A_c T_s} \\ B &= \int_0^{T_s} e^{A_{cs}} B_c ds. \end{aligned}$$

The MPC design process formulates the problem such that a method (usually a QP method) can be applied to it, by defining the matrices

$$\Phi = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix} \quad \Gamma = \begin{bmatrix} H_1 & 0 & \cdots & 0 \\ H_2 & H_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ H_N & H_{N-1} & \cdots & H_1 \end{bmatrix} \quad H_i = CA^{i-1}B \qquad 1 \leq i \leq N$$

All odd numbered columns of Γ correspond to Pump 1 and all even numbered columns correspond to Pump 2.

We now have enough information to get the four tank system on the form of problem (6.1). To get the problem on this form Φ is separated with respect to each pump.

$$A = \begin{bmatrix} \phi_1 & \phi_3 & \cdots & \phi_{N-1} \end{bmatrix} \quad B = \begin{bmatrix} \phi_2 & \phi_4 & \cdots & \phi_N \end{bmatrix} \quad b = -\Phi x_0 \quad (7.2)$$

where x_0 denotes the initial state and ϕ_i is the i-th vector of Φ. Furthermore the linear equality constraints are constructed by setting

$$C = B^T A \quad D = B^T B \quad d = B^T b. \quad (7.3)$$

This was suggested in Section 6.1. What remains is the choice of Δ, the restriction on the size of the solution. Once Δ is chosen we have all the information needed and Algorithm 2 can be applied to solve the system.

It is important to note that when solving the control problem, it is the deviation from the steady state that is found.

7.1.2 Testing

This section describes the testing of the four-tank system by applying Algorithm 2. An overview of the parameters used in testing is given in Table

Value	Description
1.2272	a_i,i, $i = 1, 2, 3, 4$, area of outlet pipe in cm^2
380.1327	A_i, $i = 1, 2, 3, 4$, cross sectional area of Tank i in cm^2
981	g, acceleration of gravity in cm/s^2
0.45	γ_1, flow distribution constant, Valve 1
0.40	γ_2, flow distribution constant, Valve 2

Table 7.1: *Parameters used during testing*

7.1. The values are based on a test system used in labs at the Technical University of Denmark.

The steady state was obtained by choosing $F_1^s = F_2^s = 300$ and solving the system (7.1) set equal to zero.

Initial values related to the algorithm were chosen to be $\nu_0 = 0.1$ and $\sigma_0 = 0.1$. Update strategy used is $\gamma_\nu = 10$ and $\gamma_\sigma = 0.1$. Figure 7.2 shows a result.

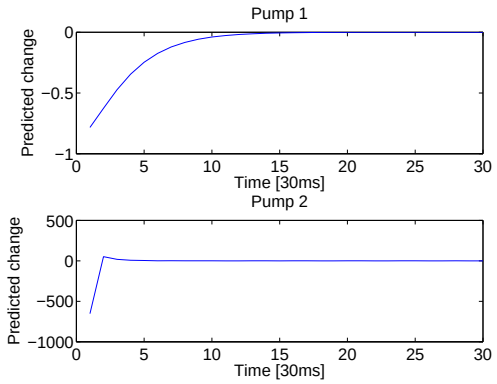


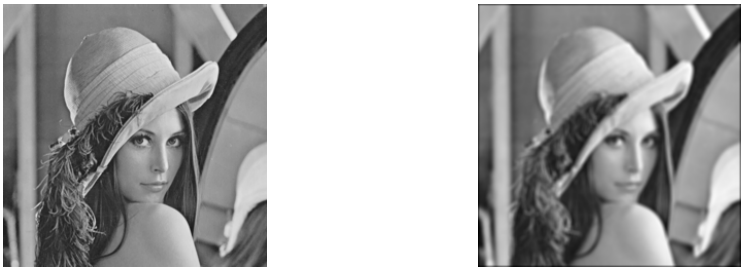
Figure 7.2: *A result for the four-tank system*

One iteration for the Algorithm 2 was needed. An accuracy of only $\epsilon_\Delta = 10^{-1}$ was achieved, implying that the algorithm did not perform well on the problem.

7.2 Image deblurring and misalignment

When images are captured by an imaging system, for example by a camera or a microscope, *blurring* occurs. Blurring processes typically result in images with less contrast and less distinct edges. During capture not all of the high-frequency components, the parts containing details, are retained.

There are several types of blurring, among which the simplest is perhaps linear motion blur, where the camera is moved in a certain direction while capturing the image at low shutter speed. This results in a smeared unclear image. Another form of blur is when the focal lens of the camera is wrongly adjusted, resulting in out-of-focus blur. Yet another form of blur originates from atmospheric turbulence. This kind of blur is often simulated by a applying a Gaussian kernel to an image. This is illustrated with the standard test image of Lena [28] in Figure 7.3.



(a) *Original image* **(b)** *Blurred image*

Figure 7.3: *(a) original image, (b) image blurred by a Gaussian kernel*

When digitalising an image, noise is typically introduced after the blurring, adding to the complexity of deblurring. A natural starting point for a deblurring method is to have a blurred image with noise. In the process of trying out methods, we simulate noise by stochastic models. Typically pseudorandom numbers distributed uniformly or by a Gaussian distribution is added to the blurred image.

Colour images are divided into channels, where each channel represents a distinct part of the image. The RGB (Red-Green-Blue) colour model is one way this representation can be carried out. Each channel consists of an array of intensities related to each of the three colours, which means that the whole image is represented as a multidimensional array.

When capturing an image, misalignment may occur between the colour channels. This is due to the optics and intrinsics of the camera. A source of misalignment is digital cameras with one CCD (Charge Coupled Device). In such cameras the incoming light is split into three parts representing each colour in the model and stored in a filter called a Bayer pattern. If this pattern is misaligned with respect to the incoming light, some light representing another colour may be falsely interpreted, causing an incorrect distribution of colours, [11, Chapter 7].

Perhaps a more common place to observe image misalignment is when reading a newspaper. Sometimes the printing presses are misaligned causing a displacement between the colour channels (typically using a different colour model known as CMY (Cyan-Magenta-Yellow)), resulting in an unappealing image.

Misalignment is also termed in a more positive way, namely alignment. A related problem is image registration where the goal is to automatically align related images. Registration is used in fields such as medical imaging and computer vision.

In Section 7.2.1 a model for deblurring and misalignment is presented and in Section 7.2.2 a test using Algorithm 2 is shown. Note that both deblurring and misalignment are solved simultaneously in one problem.

7.2.1 A model for misalignment

To test out the algorithm we restrict ourselves to only two colour channels and model the problem as follows. The blurring process is modelled using atmospheric turbulence blur. The point spread function (PSF) is generated by the **blur** routine from [9]. The blur was first applied to the whole test image then two overlapping parts of the image were used to represent each colour channel. The blurred channels, B_1 and B_2, were subjected to additive noise on the order of 10^{-2}. The mathematical model for the blurring is

$$A = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}$$

where $A_1, A_2 \in \mathbb{R}^{128^2 \times 128^2}$ represent identical PSFs. The channels are represented as matrices, to fit the problem they first needed to be vectorised $b_1 = vec(B_1)$ and similarly, $b_2 = vec(B_2)$. The operation $x = vec(X)$ performs stacking of the columns of X, in the order $x_1, x_2, \ldots x_n$ from top to bottom.

The misalignment was modelled using homogeneous coordinates, which is commonly used in computer graphics. See for example [12, Chapter 5]. Homogeneous coordinates allows affine translation, or misalignment, of points to be described by a matrix operation. We get the following model for the misalignment,

$$T = \begin{pmatrix} 1 & 0 & m_x \\ 0 & 1 & m_y \\ 0 & 0 & h \end{pmatrix}$$

The numbers m_x and m_y denote the misalignment in each direction and h is a number different from zero. To obtain the misalignment in Euclidean coordinates, a division by h will suffice. In the tests, $h = 1$ initially.

The misalignment is now almost on the form of (6.1). By using the PSF matrix A, constructing a zero-padded translation matrix

$$B = \begin{bmatrix} T \\ 0 \end{bmatrix}$$

and the misaligned and blurred channels as $b = \begin{bmatrix} b_1^T & b_2^T \end{bmatrix}^T$, we have the parameters needed for the objective function. The matrices and vector corresponding to the linear equality constraint can be constructed as suggested in Section 6.1, namely $C = B^T A$, $D = B^T B$ and $d = B^T b$. The unknown x represents the deblurred image channels, $x = \begin{bmatrix} x_1^T & x_2^T \end{bmatrix}^T$, where $x_1 = vec(X_1)$ and $x_2 = vec(X_2)$. The unknown $y \in \mathbb{R}^3$ represents the inverse misalignment in homogeneous coordinates. Finally we have to choose Δ. As mention in Section 2.3, for imaging a natural choice for Δ is energy of the image signal.

7.2.2 Testing

To test the algorithm we use the image from Figure 7.3a. Two subsections of the image around the hat were extracted and the algorithm was applied as outlined in the previous section.

Figure 7.4: *Deblurring and misalignment result of two* 128×128 *images. The green and red colour channel overlaps to one pixel accuracy in each direction.*

For $\sigma_0 = 0.01$, $\nu_0 = 0.1$ and update constants $\gamma_\nu = 10$ and $\gamma_\sigma = 0.01$. The result for the test images is shown in Figure 7.4. A misalignment of $(m_x, m_y) = (-30, -20)$ was added before running the algorithm. The result gives nearly the inverse, namely $(x_s, y_s) \approx (30.95, 21.03)$. Since only two colour channels were used the third channel (blue) was filled with zeros, resulting in the black background.

Two iterations were needed in addition to computing x_0 and y_0, at the cost of 614 matrix-vector products.

Chapter 8

Conclusions and further work

The trust-region subproblem solver was intended to be used as an inner iteration of a method for the partially norm constrained least squares problem. The solver's accuracy and sensitivity has been explored.

First boundary solutions of the solver were compared with Tikhonov regularisation solved by SVD. The correspondence was found to be close to machine precision for small Δ, and sufficiently adjustable for solutions near to or in the potential hard case.

The smallest eigenvalue of the Hessian matrix in the trust-region subproblem is crucial for finding solutions. The sensitivity of the smallest eigenvalue along with its corresponding eigenvector was investigated by perturbing the eigenvalue and eigenvector separately. Results show that the solver gives solutions within the given accuracy requirement. Also the distribution of the norm of the perturbed solutions shows an overall linear tendency.

The problem of regularisation with unrestricted variables has been presented together with a reformulation. Based on these formulations a method applicable to large-scale problems was developed. The method seems promising in one of the two applications presented.

Some areas require further work. Among these are the need for more strategic update choices of the penalty and scaling parameters for the scaling algorithm. Also argumentation for that a decrease in the quadratic penalty function guarantees a decrease in the merit function needs to be given.

The comparison with Tikhonov regularisation suggests that LSTRS solves the boundary case of the TRS cost effective when not in a potential hard case. As the hard case is not common in optimisation, it would be interesting to compare the computational costs of LSTRS with other large-scale methods which only approximate the solution of the TRS.

Appendix A

Definitions

Definition 1. The most common vector norms are the 1-norm (l_1), 2-norm (l_2) and ∞-norm (l_∞) defined below on a vector $x \in \mathbb{R}^n$.

$$\begin{aligned} \text{1-norm} \qquad & \|x\|_1 = \sum_{i=1}^{n} |x_i| \\ \text{2-norm} \qquad & \|x\|_2 = \left(\sum_{i=1}^{n} x_i^2 \right)^{1/2} = \sqrt{x^T x} \\ \infty\text{-norm} \qquad & \|x\|_\infty = \max_{i=1,\ldots,n} |x_i|. \end{aligned}$$

The second of these is often referred to as the Euclidian norm.

Definition 2. The rank of a matrix A is the dimension of the column space of A and is denoted $\operatorname{rank} A$.

A matrix that with the maximum possible rank is said to have *full rank*. If not the matrix is *rank deficient*.

Definition 3. An *orthogonal matrix* is a square invertible matrix Q with the property that $Q^{-1} = Q^T$.

Note that an orthogonal matrix has orthonormal columns, that is, each column q_i is normalised to have $\|q_i\| = 1$.

Appendix B

LSTRS Tolerances

A description of tolerances supplied to the LSTRS method and their default values. More information on input parameters is available in the software manual for the LSTRS package at the LSTRS homepage[1]. See [23] for elaboration on the tolerances and other aspects of the LSTRS method.

Tolerance	Default value	Description
ϵ_Δ	10^{-4}	The accuracy in the norm of the solution with respect to a boundary solution $\frac{\lvert\lVert x\rVert-\Delta\rvert}{\Delta} \leq \epsilon_\Delta$.
ϵ_{HC}	10^{-4}	The accuracy of a quasi-optimal solution. Let $\psi(x) = \frac{1}{2}x^T Hx + g^T x$ and x^* be the optimal solution and $\tilde{x}$ the quasi-optimal solution, then $\psi(x^*) \leq \psi(\tilde{x}) \leq (1 - \epsilon_{HC})\psi(x^*)$ must hold.
ϵ_{Int}	10^{-10}	Declares the algebraically smallest eigenvalue of B_α in the test for an interior solution. λ_1 is considered positive if $\lambda_1 > -\epsilon_{Int}$.
ϵ_α	10^{-8}	The safeguarding interval for α is considered too small when $\lvert\alpha_U - \alpha_L\rvert \leq \epsilon_\alpha \max\{\lvert\alpha_L\rvert, \lvert\alpha_H\rvert\}$.
ϵ_ν	10^{-2}	The relative size of an eigenvector component ξ is small when $\lvert\xi\rvert \leq \epsilon_\xi \frac{\lVert q\rVert}{\lVert g\rVert}$.

Table B.1: *Tolerances of LSTRS*

[1] http://www2.imm.dtu.dk/~mr/lstrs.html

Bibliography

[1] E. Anderson, A. McKenney, D. Sorensen, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. J. Dongarra, J. Du Croz, S. Hammarling, et al. *LAPACK Users' guide*. SIAM, 1999. ISBN 0-89871-447-8.

[2] R. C. Aster, B. Borchers, and C. H. Thurber. *Parameter Estimation and Inverse Problems*. Elsevier Academic Press, 2005. ISBN 0-12-065604-3.

[3] Ø. Bergmann. *Optimization issues in medical imaging and fiber-tracking*. PhD thesis, University of Bergen, 2008. ISBN 978-82-308-0520-6.

[4] M. Bertero and P. Boccacci. *Introduction to Inverse Problems in Imaging*. IOP Publishing, Bristol, UK, 1998. ISBN 0-7503-0435-9.

[5] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust Region Methods*. SIAM, Philadelphia, PA, 2000. ISBN 0-89871-460-5.

[6] D. M. Gay. Computing optimal locally constrained steps. *SIAM Journal on Scientific and Statistical Computing*, 2(2):186–197, 1981.

[7] G. H. Golub and C. van Loan. *Matrix Computations*. Johns Hopkins University Press Baltimore, MD, USA, London, third edition, 1996. ISBN 0-8018-5414-8.

[8] P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems*. SIAM, Philadelphia, PA, 1998. ISBN 0-89871-403-6.

[9] P. C. Hansen. Regularization Tools version 4.0 for Matlab 7.3. *Numerical Algorithms*, 46(2):189–194, 2007.

[10] P. C. Hansen, M. Kilmer, and R. H. Kjeldsen. Exploiting residual information in the parameter choice for discrete ill-posed problems. *BIT*, 46:41–59, 2006.

[11] P. C. Hansen, J. G. Nagy, and D. P. O'leary. *Deburring Images: Matrices, Spectra, and Filtering*. SIAM, 2006. ISBN 0-89871-618-7.

[12] D. Hearn and M. Baker. *Computer Graphics with OpenGL*. Prentice-Hall PTR, third edition, 2004. ISBN 0-13-120238-3.

[13] K. H. Johansson. The quadruple-tank process: a multivariable laboratory process with an adjustable zero. *Control Systems Technology, IEEE Transactions on*, 8(3):456–465, 2000.

[14] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users' Guide: Solution of Large-scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. SIAM, 1998. ISBN 0-89871-407-9.

[15] J. J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.

[16] S. Morigi, L. Reichel, and F. Sgallari. An interior-point method for large constrained discrete ill-posed problems. *J. Comput. Appl. Math., in press*, 2008.

[17] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer series in operations research. Springer-Verlag, second edition, 2006. ISBN 0-387-30303-0.

[18] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Software*, 8(1): 43–71, 1982.

[19] J. B. Rawlings. Tutorial overview of model predictive control. *Control Systems Magazine, IEEE*, 20(3):38–52, 2000.

[20] 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):1843–1861, 2002.

[21] M. Rojas and T. Steihaug. An interior-point trust-region-based method for large-scale non-negative regularization. *Inverse Problems*, 18(5): 1291–1307, 2002.

[22] M. Rojas, S. A. Santos, and D. C. Sorensen. A new matrix-free algorithm for the large-scale trust-region subproblem. *SIAM J. Optim.*, 11(3):611–646, 2000.

[23] M. Rojas, S. A. Santos, and D. C. Sorensen. LSTRS: MATLAB software for large-scale trust-region subproblems and regularization. *ACM Trans. Math. Software*, 34(2):11, 2008.

[24] D. C. Sorensen. Newton's method with a model trust region modification. *SIAM Journal on Numerical Analysis*, 19(2):409–426, April 1982.

[25] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983. ISSN 00361429.

[26] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 4:1035–1038, 1963.

[27] L. N. Trefethen and D. Bau, III, editors. *Numerical Linear Algebra*. SIAM, Philadelphia, PA, 1997. ISBN 0-89871-361-7.

[28] A. G. Weber. The USC-SIPI image database version 5. *USC-SIPI Report*, 315:1–24, 1997.