The Sherman Morrison Iteration

Joseph T. Slagel

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Mathematics

Julianne Chung, Chair
Matthias Chung, Chair
Serkan Gugercin

May 6, 2015
Blacksburg, Virginia

Keywords: "Sherman Morrison, regularized least squares problem, extremely underdetermined linear system,"
Copyright 2015, Joseph T. Slagel

The Sherman Morrison Iteration

Joseph T. Slagel

(ABSTRACT)

The Sherman Morrison iteration method is developed to solve regularized least squares problems. Notions of pivoting and splitting are deliberated on to make the method more robust. The Sherman Morrison iteration method is shown to be effective when dealing with an extremely underdetermined least squares problem. The performance of the Sherman Morrison iteration is compared to classic direct methods, as well as iterative methods, in a number of experiments. Specific Matlab implementation of the Sherman Morrison iteration is discussed, with Matlab codes for the method available in the appendix.

Acknowledgments

I acknowledge Dr. Julianne Chung and Dr. Matthias Chung for their masterful patience and sage-like guidance during the composition of this thesis. I also acknowledge Dr. Serkan Gugercin for serving on my committee and being a transcendent teacher.

Contents

List of Figures

Chapter 1

Introduction

Consider the following regularized least squares problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + \lambda^2\|\mathbf{L}\mathbf{x}\|_2^2, \tag{1.1}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}, \mathbf{b} \in \mathbb{R}^m, \lambda \in \mathbb{R}$, and $\mathbf{L} \in \mathbb{R}^{\ell\times n}$ for some suitable $m, n, \ell \in \mathbb{N}$. Let us also assume that $n \gg m$, meaning that this least squares problem is extremely underdetermined. Least squares problems of this nature arise frequently in signal processing ([3, 19]), tomography ([8, 30, 33, 21]), geophysics ([26, 32]), and genetics ([20, 23]).

The primary goal of this thesis is to further develop the Sherman Morrison Iteration as described by P. Maponi ([22, 10]), and extend it as an efficient direct solver of (1.1). In special cases, the Sherman Morrison iteration becomes a preferable approach over traditional direct methods and even sophisticated iterative methods such as LSQR and LSMR.

First we must review some background topics. Chapter 2 reviews the basics of ill-posed inverse problems, least squares problems, and Tikhonov regularization. In Chapter 3, the Sherman Morrison iteration in its most general sense will be developed with methods such as pivoting and splitting. We will then look at the specific application of the Sherman Morrison iteration to regularized least squares problem and investigate when it is beneficial to use. Numerical results are provided in Chapter 4. The Matlab code for the developed Sherman Morrison Iteration is in the appendix.

Chapter 2

Preliminaries

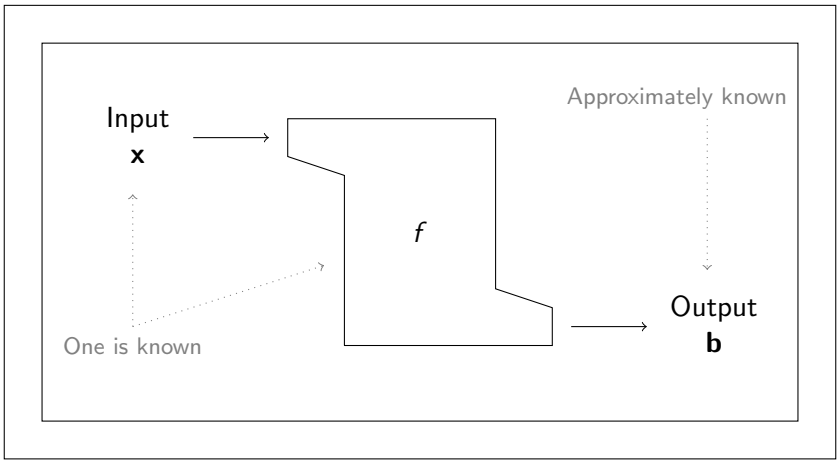
2.1 Ill-Posed Inverse Problems

Suppose $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ is a function with an input $\mathbf{x} \in \mathbb{R}^n$ and an output $\mathbf{b} \in \mathbb{R}^m$. In other words we have

$$\mathbf{f}(\mathbf{x}) = \mathbf{b}. \tag{2.1}$$

If $\mathbf{x}$ and $\mathbf{f}$ are known, finding a $\mathbf{b} \in \mathbb{R}^m$ that satisfies (2.1) is referred to as the direct problem. If we know $\mathbf{b}$ and $\mathbf{f}$ or $\mathbf{x}$, then the process of finding a $\mathbf{x} \in \mathbb{R}^n$ or $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ that satisfies (2.1) is known as the inverse problem [2].

Figure 2.1: The inverse problem set up. We have output $\mathbf{b}$ (perhaps contaminated by noise) of a function $\mathbf{f}$ with input $\mathbf{x}$. Either $\mathbf{x}$ or $\mathbf{f}$ is known, and the inverse problem is to find the other [15].



Jacques Hadamard was first to define the notion of a well posed problem (*parfaitement bien posé* [24]). A well-posed problem is a problem in which

1. there is a solution,
2. the solution is unique, and
3. the solution depends continuously on the data.

When a problem is not well-posed it is called ill-posed. A number of inverse problems are ill-posed. For problem (2.1) ill-posedness means

1. there is not an $\mathbf{x} \in \mathbb{R}^n$ such that (2.1) is satisfied,
2. there are many $\mathbf{x} \in \mathbb{R}^n$ such that (2.1) is satisfied, or
3. the solutions $\mathbf{x}$ and $\hat{\mathbf{x}}$ to $\mathbf{f}(\mathbf{x}) = \mathbf{b}$ and $\mathbf{f}(\hat{\mathbf{x}}) = \mathbf{b} + \boldsymbol{\eta}$, respectively, are not necessarily close to each other for small $\boldsymbol{\eta}$. This means that little perturbations in $\mathbf{b}$ result in potentially large perturbations of the solution to (2.1).

For detailed discussion about inverse problems in a general sense, refer to [1, 2, 25, 35]. We now explore the ill-posedness of a specific inverse problem, namely the solving of linear systems.

2.2 Least Squares Problems

Consider the problem

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{2.2}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$ for some suitable $m, n \in \mathbb{N}$. This is the same as (2.1) where $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}$. If we know $\mathbf{A}$ and $\mathbf{b}$, then finding $\mathbf{x}$ is indeed an inverse problem. This inverse problem can be ill-posed. Below are some simple examples of how ill-posedness can appear in such problems. First an example of a problem in which there is no solution:

Example 2.2.1. *Consider the system*

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

This system clearly has no solution $\mathbf{x}$ *because if it did the first entry of* $\mathbf{x}$ *would have to simultaneously be* 1 *and* 2.

Now an example where there is no unique solution

Example 2.2.2. *Consider the system*

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

This system is satisfied by any vector of the form

$$\mathbf{x} = \begin{pmatrix} 1 \\ c \end{pmatrix},$$

where $c \in \mathbb{R}$.

Now an example of an unstable system.

Example 2.2.3. *Consider the system*

$$\begin{pmatrix} 1 & 1 \\ 1 & 1+\epsilon \end{pmatrix} \mathbf{x} = \begin{pmatrix} 2 \\ 2 \end{pmatrix},$$

where $\epsilon > 0$. *The solution to this is clearly*

$$\mathbf{x} = \begin{pmatrix} 2 \\ 0 \end{pmatrix},$$

however, if we perturb $\mathbf{b}$ *slightly to obtain the linear system*

$$\begin{pmatrix} 1 & 1 \\ 1 & 1+\epsilon \end{pmatrix} \mathbf{x} = \begin{pmatrix} 2 \\ 2+\epsilon \end{pmatrix},$$

we see the solution to the system is

$$\mathbf{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

We see that a small perturbations to $\mathbf{b}$ *could impact the value of* $\mathbf{x}$ *drastically. We call such a system* un-stable. *When small perturbations of* $\mathbf{b}$ *result in small perturbations of* $\mathbf{x}$, *we call the system* stable.

A unique solution exists for (2.2) if and only if $\mathbf{A}$ is invertible. In that case, (2.2) can be solved by setting $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, but stability is still in an concern. Suppose we solve the system

$$\mathbf{A}\hat{\mathbf{x}} = \mathbf{b} + \boldsymbol{\eta}, \tag{2.3}$$

where $\boldsymbol{\eta}$ represents the noise in the measured $\mathbf{b}$. Solving the system we get

$$\hat{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{b} + \mathbf{A}^{-1}\boldsymbol{\eta}. \tag{2.4}$$

For ill-posed problems, it is possible that $\hat{\mathbf{x}}$ is not near $\mathbf{x}$, the solution to (2.2). To see this consider the bound on the relative error between $\mathbf{x}$ and $\hat{\mathbf{x}}$

$$\frac{\|\mathbf{x}-\hat{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2} \leq \text{cond}(\mathbf{A})\frac{\|\boldsymbol{\eta}\|_2}{\|\mathbf{b}\|_2}. \tag{2.5}$$

Here $\|\cdot\|_2$ represents the 2-norm (the euclidian norm) and $\text{cond}(\mathbf{A})$ represents the condition number of $\mathbf{A}$ in the 2-norm. In particular

$$\text{cond}(\mathbf{A}) = \frac{\sigma_1}{\sigma_r}, \tag{2.6}$$

where σ_1 is the largest singular value of $\mathbf{A}$ and σ_r is the smallest singular value of $\mathbf{A}$. For information on norms and condition numbers see [9].

The bound in (2.5) shows that the computed solution could be far from the real solution if $\mathbf{A}$ is not well conditioned and there some noise introduced in the system.

For systems with no unique solution, either too many solutions or none at all, we seek an $\mathbf{x}$ that is "close" to a solution, meaning that it minimizes

$$\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2.$$

We arrive at the least squares problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2. \tag{2.7}$$

Assuming that $\mathbf{A} \in \mathbb{R}^{m\times n}$ has full column rank, $\mathbf{x}, \in \mathbb{R}^n$, and $\mathbf{b} \in \mathbb{R}^m$. Solving (2.7) is the equivalent to minimizing the function $\mathbf{g}$ where

$$\begin{aligned}\mathbf{g}(\mathbf{x}) = \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 &= (\mathbf{A}\mathbf{x}-\mathbf{b})^\top(\mathbf{A}\mathbf{x}-\mathbf{b}) \\ &= (\mathbf{x}^\top\mathbf{A}^\top-\mathbf{b}^\top)(\mathbf{A}\mathbf{x}-\mathbf{b}) \\ &= \mathbf{x}^\top\mathbf{A}^\top\mathbf{A}\mathbf{x}-\mathbf{b}^\top\mathbf{A}\mathbf{x}-\mathbf{x}^\top\mathbf{A}^\top\mathbf{b}+\mathbf{b}^\top\mathbf{b}.\end{aligned}$$

To minimize this function we find its critical points. Taking the derivative of $\mathbf{g}$ we get

$$\nabla\mathbf{g}(\mathbf{x}) = \mathbf{A}^\top\mathbf{A}\mathbf{x}-\mathbf{A}^\top\mathbf{b},$$

setting this equal to zero we arrive at the system

$$\mathbf{A}^\top \mathbf{A}\mathbf{x} = \mathbf{A}^\top \mathbf{b}. \tag{2.8}$$

The system (3.14) is what is known as the *normal equations*. If $\mathbf{A}$ has full column rank then $\mathbf{A}^\top \mathbf{A}$ is positive definite, and therefore invertible. Hence, the solution of the least squares system is

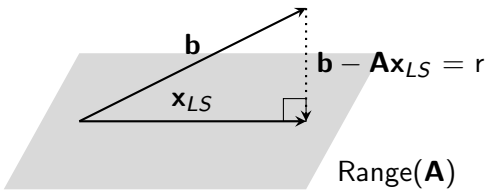
$$\mathbf{x}_{\text{LS}} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}.$$

We define the residual as

$$\mathbf{r} = \mathbf{A}\mathbf{x}_{\text{LS}} - \mathbf{b}$$

Upon further investigation, $\mathbf{x}_{\text{LS}}$ is found by projecting $\mathbf{b}$ orthogonally on the range of $\mathbf{A}$. A visualization of this is available in the Figure 2.2

Figure 2.2: The solution $\mathbf{x}_{\text{LS}}$ to the least squares problem (2.7) is the orthogonal projection of $\mathbf{b}$ onto the range of $\mathbf{A}$ [9].

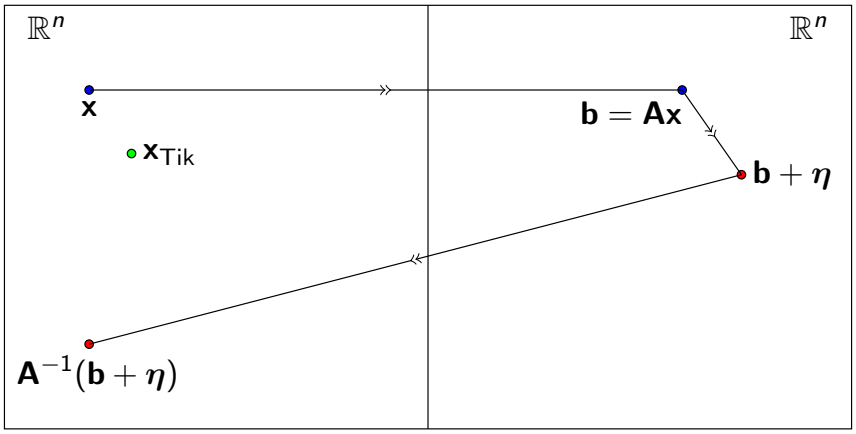


If we apply the bound (2.5) to the normal equations (2.9) we see the condition of the problem is on the order of $\text{cond}(\mathbf{A}^\top \mathbf{A}) = \text{cond}(\mathbf{A})^2$ [9]. When $\mathbf{A}$ is poorly conditioned, solving the normal equations becomes unstable. Recall from (2.6) that the condition number of $\mathbf{A}$ is the ratio between the largest and smallest singular value of $\mathbf{A}$. Small errors in $\mathbf{b}$ could result in error of the calculated solution when the smallest singular value of $\mathbf{A}$ is close to zero.

If $\mathbf{A}$ does not have full column rank there is not a unique least squares solution, since $\mathbf{A}^\top \mathbf{A}$ is not invertible. The solution set for this least squares problem can be characterized, with perturbation theory similar to that of the full column rank case [9].

To impose uniqueness and stability into an ill-posed least squares problem, we introduce Tikhonov regularization. For more information on least squares problems see [9, 12, 17, 25].

Figure 2.3: Motivation for Tikohnov regularization. The inverse solution $\mathbf{A}^{-1}(\mathbf{b}+\boldsymbol{\eta})$ is far from the true solution $\mathbf{x}$. The idea of Tikhonov regularization is to replace an ill-conditioned problem with a nearby well-conditioned problem, thus obtaining $\mathbf{x}_{\text{Tik}}$ which is close to the true solution [15].



2.3 Tikhonov Regularization

Tikhonov regularization, named after Andrey Tikhonov [34], replaces an ill-posed inverse problem with a nearby well-posed inverse problem.

Consider the minimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + \lambda^2\|\mathbf{L}\mathbf{x}\|_2^2, \tag{2.9}$$

where $\mathbf{A}\in\mathbb{R}^{m\times n}, \mathbf{L}\in\mathbb{R}^{\ell\times n}, \lambda\in\mathbb{R}$ and $\mathbf{b}\in\mathbb{R}^m$. This is called a regularized least squares problem. The matrix $\mathbf{L}$ is a regularization matrix. The regularization matrix forces structure or smoothness on the solution and λ is a constant that determines how much emphasis is put on the regularization term. Notice that equation (2.9) can be rewritten as

$$\min_{\mathbf{x}\in\mathbb{R}^n} \left\| \begin{pmatrix} \mathbf{A} \\ \lambda\mathbf{L} \end{pmatrix} \mathbf{x} - \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} \right\|_2^2. \tag{2.10}$$

The corresponding normal equations to (2.10) are

$$\begin{pmatrix} \mathbf{A} \\ \lambda\mathbf{L} \end{pmatrix}^\top \begin{pmatrix} \mathbf{A} \\ \lambda\mathbf{L} \end{pmatrix} \mathbf{x} = \begin{pmatrix} \mathbf{A} \\ \lambda\mathbf{L} \end{pmatrix}^\top \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix}, \tag{2.11}$$

which can be rewritten as

$$(\mathbf{A}^\top\mathbf{A} + \lambda^2\mathbf{L}^\top\mathbf{L})\mathbf{x} = \mathbf{A}^\top\mathbf{b}. \tag{2.12}$$

If $\lambda = 0$ then (2.12) is identical to (2.7). Given an underdetermined problem, the classical normal equations (3.14) will not have a unique solution. However, if $\mathbf{L}$ has full column rank, $(\mathbf{A}^\top\mathbf{A} + \mathbf{L}^\top\mathbf{L})$ is positive definite, and therefore invertible. Next we investigate the solution of (2.10) compared to solutions of the original least squares problem (2.7).

Let us start with a simple choice of the regularization matrix. Let $\mathbf{L} = \mathbf{I}$ where $\mathbf{I}$ is the $n \times n$ identity matrix, and let $\lambda > 0$. From (2.12) we arrive at the normal equations:

$$(\mathbf{A}^\top\mathbf{A} + \lambda^2\mathbf{I})\mathbf{x} = \mathbf{A}^\top\mathbf{b}$$

The matrix $\mathbf{A}^\top\mathbf{A} + \lambda^2\mathbf{I}$ is positive definite, and therefore invertible. We obtain the Tihkonov solution

$$\mathbf{x}_{\text{Tik}} = (\mathbf{A}^\top\mathbf{A} + \lambda^2\mathbf{I})^{-1}\mathbf{A}^\top\mathbf{b}.$$

Let $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ be the SVD decomposition of $\mathbf{A}$ [9]. Since $\mathbf{V}\mathbf{V}^\top = \mathbf{I}$ we see

$$\begin{aligned}
\mathbf{x}_{\text{Tik}} &= (\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^\top + \lambda^2\mathbf{V}\mathbf{V}^\top)^{-1}\mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top\mathbf{b} \\
&= (\mathbf{V}(\mathbf{\Sigma}^2 + \lambda^2\mathbf{I})\mathbf{V}^\top)^{-1}\mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top\mathbf{b} \\
&= \mathbf{V}(\mathbf{\Sigma}^2 + \lambda^2\mathbf{I})^{-1}\mathbf{\Sigma}\mathbf{U}^\top\mathbf{b} \\
&= \mathbf{V}\mathbf{D}\mathbf{\Sigma}^{-1}\mathbf{U}^\top\mathbf{b},
\end{aligned}$$

where $\mathbf{D}$ is a diagonal matrix with entries

$$d_{ii} = \frac{\sigma_i^2}{\sigma_i^2 + \lambda},$$

with σ_i being the ith singular value of $\mathbf{A}$.

When λ is close to zero, $\mathbf{D}$ will be close to the identity matrix, and $\mathbf{x}_{\text{Tik}}$ will become increasingly similar to the unregularized least squares solution, potentially contaminated by noise . The larger λ gets the more emphasis is on the regularization of $\mathbf{x}_{\text{Tik}}$, which for $\mathbf{L} = \mathbf{I}$ means reducing the overall magnitude of $\mathbf{x}_{\text{Tik}}$. As λ get larger, the smaller singular values in $\mathbf{A}$ are filtered out of the solution $\mathbf{x}_{\text{Tik}}$. Notice as $\lambda \to \infty$, $\mathbf{D} \to \mathbf{0}$, and so the solution $\mathbf{x}_{\text{Tik}}$ goes to $\mathbf{0}$. Methods for choosing λ are well developed [5, 4, 27].

There are more complicated choices of $\mathbf{L}$ than the identity matrix that impose various conditions on the solution to the system. For example, if $\mathbf{L}$ is the difference operator then regularization will impose smoothness on the solution $\mathbf{x}$, i.e. the entries of $\mathbf{x}$ will be close to one another. To look at the effect a regularization matrix $\mathbf{L}$ has in general on the least squares problem see [15].

It is important to think about what is necessary to solve the regularized least squares problem in equation (2.9) for a general $\mathbf{L}$. For canonical LSMR, LSQR, and CGLS implementations ([7, 11, 28, 29]) the standard Tikhonov problem (where $\mathbf{L} = \mathbf{I}$) can be solved with only one additional input parameter, λ. However, for general-form Tikhonov regularization (with general $\mathbf{L}$), additional care must be taken. If $\mathbf{L}$ is invertible we can make a change of variables. With $\mathbf{L}^{-1}\mathbf{y} = \mathbf{x}$, recast (2.9) as solving:

$$\min_{y \in \mathbb{R}^n} \|\mathbf{A}\mathbf{L}^{-1}\mathbf{y} - \mathbf{b}\|_2^2 + \lambda^2 \|\mathbf{y}\|_2^2, \tag{2.13}$$

where $\mathbf{x} = \mathbf{L}^{-1}\mathbf{y}$. Notice this process requires us to find $\mathbf{L}^{-1}$. If $\mathbf{L}$ is not invertible or $\mathbf{L}^{-1}$ is too costly to find, we could also apply LSQR, LSMR, or CGLS to (2.10).

There are also cases where we are given $(\mathbf{L}^\top \mathbf{L})^{-1}$ and must find $\mathbf{L}$ or $\mathbf{L}^{-1}$ in order to carry out traditional solvers of regularized least squares problems. Suppose we have the system

$$\mathbf{A}\mathbf{x} = \mathbf{b} + \boldsymbol{\eta},$$

where $\mathbf{x}$ and $\boldsymbol{\eta}$ are sampled from random variables $\mathbf{X}$ and $\mathbf{N}$, respectively, where $\mathbf{X} \sim \mathcal{N}(0, \gamma^2 \boldsymbol{\Gamma})$ and $\mathbf{N} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$. The maximum likelihood estimate of $\mathbf{x}$ is given by (2.9) where $\lambda = \sigma/\gamma$ and $\boldsymbol{\Gamma} = (\mathbf{L}^\top \mathbf{L})^{-1}$ [6]. To find the regularization matrix $\mathbf{L}$ a Cholesky decomposition must be preformed on $\boldsymbol{\Gamma}^{-1}$. In the next chapter we introduce a method that can avoid computing $\mathbf{L}$ and $\mathbf{L}^{-1}$ by working directly with $\boldsymbol{\Gamma}$.

There is a vast amount of literature regarding Tikhonov regularization. See [16, 5, 4, 14, 27, 31, 13].

Chapter 3

The Sherman Morrison Iteration

Suppose we have the regularized least squares problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + \lambda^2\|\mathbf{L}\mathbf{x}\|_2^2, \tag{3.1}$$

where $\mathbf{L}\in\mathbb{R}^{\ell\times n}$ is a Tikhonov regularization matrix with the property that $\mathbf{L}^\top\mathbf{L}$ is easily invertible ($\mathbf{L}$ has full column rank) and $\mathbf{A}\in\mathbb{R}^{m\times n}$ with $n \gg m$. As shown in Section 2.3, solving (3.1) is equivalent to solving the system

$$(\mathbf{A}^\top\mathbf{A}+\lambda^2\mathbf{L}^\top\mathbf{L})\mathbf{x} = \mathbf{A}^\top\mathbf{b}.$$

In this chapter we will investigate and develop Sherman Morrison iteration method for solving this system effectively. First we develop the general case, then we apply it to regularized least squares problem.

3.1 Developing the General Case

The goal of this section is to use the Sherman Morrison formula

$$(\mathbf{A}+\mathbf{u}\mathbf{v}^\top)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^\top\mathbf{A}^{-1}}{1+\mathbf{v}^\top\mathbf{A}^{-1}\mathbf{u}}, \tag{3.2}$$

to solve

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{3.3}$$

where $\mathbf{A}\in\mathbb{R}^{n\times n}$, $\mathbf{A}$ is non-singular, $\mathbf{x},\mathbf{b}\in\mathbb{R}^n$ and

$$\begin{aligned}\mathbf{A} &= \mathbf{Z}_0^{-1} + \mathbf{U}\mathbf{V}^\top \\ &= \mathbf{Z}_0^{-1} + \mathbf{u}_1\mathbf{v}_1^\top + \mathbf{u}_2\mathbf{v}_2^\top + \cdots + \mathbf{u}_k\mathbf{v}_k^\top.\end{aligned} \tag{3.4}$$

It should be noted that every matrix $\mathbf{A}$ has such a decomposition. For example a given matrix $\mathbf{A}$ can be written as (3.4) where

$$\mathbf{Z}_0^{-1} = \mathbf{I}_n, \qquad \mathbf{U} = \mathbf{A} - \mathbf{I}_n, \qquad \mathbf{V}^\top = \mathbf{I}_n.$$

P. Maponi in [22] and [10] developed the Sherman Morrison iteration without pivoting and with pivoting. In this chapter we extend the Sherman Morison iteration by developing notion of partial pivoting and splitting. We will start by describing the Sherman Morrison iteration without pivoting.

The goal is to solve (3.3). One approach is to invert $\mathbf{A}$ and then multiply it on the right by $\mathbf{b}$ to obtain $\mathbf{A}^{-1}\mathbf{b}$. To do this we define $\mathbf{Z}_i$ for $1 \leq i \leq k$ where

$$\mathbf{Z}_i = (\mathbf{Z}_{i-1}^{-1} + \mathbf{u}_i\mathbf{v}_i^\top)^{-1} \tag{3.5}$$

$$= \mathbf{Z}_{i-1} - \frac{\mathbf{Z}_{i-1}\mathbf{u}_i\mathbf{v}_i^\top\mathbf{Z}_{i-1}}{1 + \mathbf{v}_i^\top\mathbf{Z}_{i-1}\mathbf{u}_i}. \tag{3.6}$$

The Sherman Morrison identity in (3.2) was used to get from (3.5) to (3.6). Thus, there is a simple recursive relationship between $\mathbf{Z}_i$ and $\mathbf{Z}_{i-1}$. Using this recursion one could construct an algorithm that calculates

$$\mathbf{Z}_0 \longrightarrow \mathbf{Z}_1 \longrightarrow \ldots \longrightarrow \mathbf{Z}_k = \mathbf{A}^{-1}, \tag{3.7}$$

and then applies $\mathbf{A}^{-1}$ to $\mathbf{b}$ to find $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$. However, calculating the matrix $\mathbf{Z}_i$ for all i could be a costly task. Instead we consider an approach to find $\mathbf{x}$ directly instead.

Define

$$\mathbf{x}_i = \mathbf{Z}_i\mathbf{b},$$

for $1 \leq i \leq k$. It would be nice to have a simple recursive relationship for $\mathbf{x}_i$:

$$\mathbf{x}_0 \longrightarrow \mathbf{x}_1 \longrightarrow \ldots \longrightarrow \mathbf{x}_k = \mathbf{A}^{-1}\mathbf{b},$$

like the recursive relationship between the $\mathbf{Z}_i$ matrices in (3.7). Unfortunately, the recursion depends on multiplication with $\mathbf{Z}_{i-1}$. Using the Sherman Morrison identity we see

$$\begin{aligned}\mathbf{x}_i &= \mathbf{Z}_i\mathbf{b} \\ &= \mathbf{Z}_{i-1}\mathbf{b} - \frac{\mathbf{Z}_{i-1}\mathbf{u}_i\mathbf{v}_i^\top\mathbf{Z}_{i-1}}{1 + \mathbf{v}_i^\top\mathbf{Z}_{i-1}\mathbf{u}_i}\mathbf{b} \\ &= \mathbf{x}_{i-1} - \frac{\mathbf{Z}_{i-1}\mathbf{u}_i\mathbf{v}_i^\top}{1 + \mathbf{v}_i^\top\mathbf{Z}_{i-1}\mathbf{u}_i}\mathbf{x}_{i-1}.\end{aligned} \tag{3.8}$$

To avoid constructing the matrix $\mathbf{Z}_{i-1}$, define $\mathbf{z}_{i,j} = \mathbf{Z}_i \mathbf{u}_j$ for $1 \leq i, j \leq k$. Notice

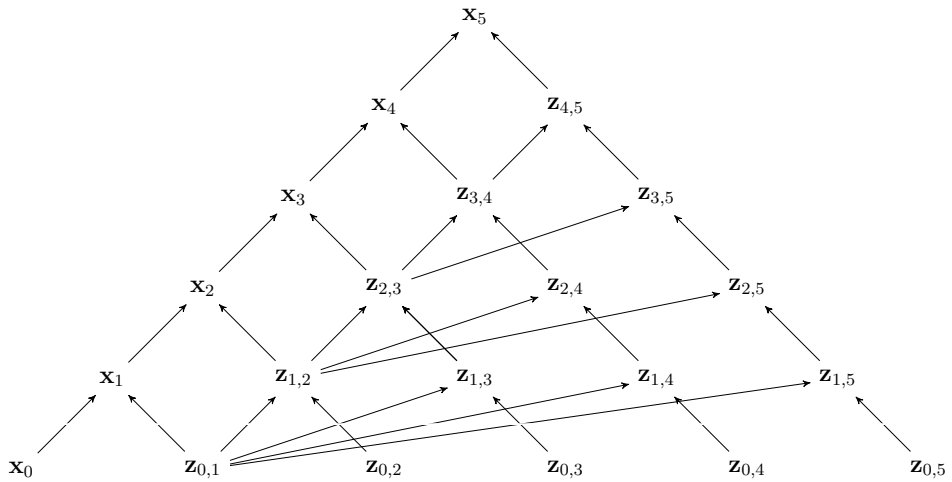
$$\begin{aligned} \mathbf{z}_{i,j} &= \mathbf{Z}_{i-1}\mathbf{u}_j - \frac{\mathbf{Z}_{i-1}\mathbf{u}_i\mathbf{v}_i^\top \mathbf{Z}_{i-1}}{1 + \mathbf{v}_i^\top \mathbf{Z}_{i-1}\mathbf{u}_i}\mathbf{u}_j \\ &= \mathbf{z}_{i-1,j} - \frac{\mathbf{z}_{i-1,i}\mathbf{v}_i^\top}{1 + \mathbf{v}_i^\top \mathbf{z}_{i-1,i}}\mathbf{z}_{i-1,j}, \end{aligned} \tag{3.9}$$

and that

$$\mathbf{x}_i = \mathbf{x}_{i-1} - \frac{\mathbf{z}_{i-1,i}\mathbf{v}_i^\top}{1 + \mathbf{v}_i^\top \mathbf{z}_{i-1,i}}\mathbf{x}_{i-1}. \tag{3.10}$$

To obtain $\mathbf{x}_i$ in (3.10) we need $\mathbf{x}_{i-1}$ and $\mathbf{z}_{i-1.i}$, which is dependent on the value of $\mathbf{z}_{i-2,i}$ and $\mathbf{z}_{i-2,i-1}$ in (3.9). For illustration of the relationship between the iterates $\{\mathbf{x}_i\}$ and the vectors $\{\mathbf{z}_{i,j}\}$ refer to Figure 3.1 for the case when $k = 5$.

Figure 3.1: An example of the recursive relationship between the iterates $\{\mathbf{x}_i\}$ and the vectors $\{\mathbf{z}_{i,j}\}$ when $k = 5$. Notice that $\mathbf{x}_i$ depends on $\mathbf{x}_{i-1}$ and $\mathbf{z}_{i-1,i}$. Note that $\mathbf{z}_{i,j}$ is dependent on $\mathbf{z}_{i-1,j-1}$ and $\mathbf{z}_{i-1,j}$.



Using this recursion we develop the standard Sherman Morrison iteration, detailed in 1.

Algorithm 1 (P. Maponi) Sherman Morrison Iteration. Solve $(\mathbf{Z}_0^{-1} + \mathbf{U}\mathbf{V}^\top)\mathbf{x} = \mathbf{b}$.

1: compute $\mathbf{x}_0 = \mathbf{Z}_0\mathbf{b}$ and $\mathbf{z}_{0,j} = \mathbf{Z}_0\mathbf{u}_j$ for $1 \le j \le k$
2: **for** $i = 1, 2, \ldots k$ **do**
3:

$$\mathbf{x}_i = \mathbf{x}_{i-1} - \frac{\mathbf{v}_i^\top \mathbf{x}_{i-1}}{1 + \mathbf{v}_i^\top \mathbf{Z}_{i-1,i}} \mathbf{z}_{i-1,i}$$

4: **for** $j = i + 1, \ldots, k$ **do**
5:

$$\mathbf{z}_{i,j} = \mathbf{z}_{i-1,j} - \frac{\mathbf{v}_i^\top \mathbf{z}_{i-1,j}}{1 + \mathbf{v}_i^\top \mathbf{z}_{i-1,i}} \mathbf{z}_{i-1,i}$$

6: **end for**
7: **end for**

There are a couple of problems with the basic Sherman Morrison iteration described in Algorithm 1 that need to be addressed. The Algorithm 1 can fail even when $\mathbf{A}$ is non-singular and Algorithm 1 is un-stable. Below is an example of when Algorithm 1 can break down.

Example 3.1.1 (Maponi)**.** *Let*

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & -1 \\ 1 & 1 & 0 \\ -1 & 0 & -1 \end{pmatrix},$$

with $\mathbf{Z}_0^{-1} = \mathrm{diag}(\mathbf{A}), \mathbf{V} = \mathbf{I}_3$, *and* $\mathbf{U} = \mathbf{A} - \mathrm{diag}(\mathbf{A})$. *Notice that* $\mathbf{A} = \mathbf{Z}_0^{-1} + \mathbf{U}\mathbf{V}^\top$ *and that* $\det(\mathbf{A}) = -1$. *Algorithm 1 fails when computing* $\mathbf{x}_2$

$$\mathbf{x}_2 = \mathbf{x}_1 - \frac{\mathbf{Z}_{1,2}\mathbf{v}_2^\top}{1 + \mathbf{v}_2^\top \mathbf{Z}_{1,2}} \mathbf{x}_1,$$

due to division by 0,

$$\begin{aligned} 1 + \mathbf{v}_2^\top \mathbf{Z}_{1,2} &= 1 + \mathbf{v}_2^\top \mathbf{Z}_1 \mathbf{u}_2 \\ &= 1 + \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 0 & -1 \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\ &= 1 + (-1) \\ &= 0. \end{aligned}$$

This is because

$$\mathbf{Z}_2 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ -1 & 0 & -1 \end{pmatrix}^{-1},$$

does not exist.

The following theorem establishes conditions for which Algorithm 1 will run.

Theorem 3.1.2. *Algorithm 1 will converge to* $\mathbf{A}^{-1}\mathbf{b}$ *in a finite number of steps if and only if* $\mathbf{Z}_i$ *exists for all* $1 \leq i \leq k$.

Proof. Recall from (3.5) and (3.6) that $\mathbf{Z}_i$ depends on $\mathbf{Z}_{i-1}$ existing and $1 + \mathbf{v}_i^\top \mathbf{Z}_{i-1}\mathbf{u}_i \neq 0$. Similarly, we see in (3.8) that $\mathbf{x}_i$ is dependent on $\mathbf{Z}_{i-1}$ existing and $1 + \mathbf{v}_i^\top \mathbf{Z}_{i-1}\mathbf{u}_i \neq 0$. □

The previous example showed how Algorithm 1 will not run in general. Even when $\mathbf{Z}_i$ exists for all $1 \leq i \leq k$, Algorithm 1 will become unstable if $1 + \mathbf{v}_i^\top \mathbf{Z}_{i-1}\mathbf{u}_i$ is near zero for some i. In the next section the method of Pivoting is introduced to avoid such issues.

3.1.1 Pivoting

Now we consider a few methods to improve Algorithm 1. Our goal is a Sherman Morrison iteration method that works for the general case $\mathbf{A}$ nonsingular, with $\mathbf{Z}_0^{-1}, \mathbf{U}, \mathbf{V}$ chosen arbitrarily such that $\mathbf{A} = \mathbf{Z}_0^{-1} + \mathbf{U}\mathbf{V}^\top$.

P. Maponi proposed a pivoting approach where at the ith iteration the largest remaining $|1 + \mathbf{v}_j^\top \mathbf{z}_{i-1,j}|$ is selected to help avoid a breakdown and increase stability.

Algorithm 2 (Maponi) SMI with Pivoting. Solve $(\mathbf{Z}_0^{-1} + \mathbf{U}\mathbf{V}^\top)\mathbf{x} = \mathbf{b}$

compute $\mathbf{x}_0 = \mathbf{Z}_0\mathbf{b}$ and $\mathbf{z}_{0,j} = \mathbf{Z}_0\mathbf{u}_j$ for $1 \le j \le k$
$\mathbf{p} = (1, 2 \ldots, k)^\top$
for $i = 1, 2, \ldots k$ **do**
  $\hat{i} = \text{argmax}\{\left|1 + \mathbf{v}_{\mathbf{p}(j)}^\top \mathbf{z}_{i-1,\mathbf{p}(j)}\right|, j = i+1, \ldots, k\}$
  swap $\mathbf{p}(i), \mathbf{p}(\hat{i})$

$$\mathbf{x}_i = \mathbf{x}_{i-1} - \frac{\mathbf{v}_{\mathbf{p}(i)^\top \mathbf{x}_{i-1}}}{1 + \mathbf{v}_{\mathbf{p}(i)}^\top \mathbf{Z}_{i-1,\mathbf{p}(i)}} \mathbf{z}_{i-1,\mathbf{p}(i)}$$

  for $j = i+1, \ldots, k$ **do**

$$\mathbf{z}_{i,\mathbf{p}(j)} = \mathbf{z}_{i-1,\mathbf{p}(j)} - \frac{\mathbf{v}_{\mathbf{p}(i)}^\top \mathbf{z}_{i-1,\mathbf{p}(j)}}{1 + \mathbf{v}_{\mathbf{p}(i)}^\top \mathbf{z}_{i-1,\mathbf{p}(i)}} \mathbf{z}_{i-1,\mathbf{p}(i)}$$

  end for
end for

The Sherman Morrison iteration with pivoting may be computationally expensive, especially for large-scale problems. We propose to use partial pivoting, which saves computing time by using the first viable $|1 + \mathbf{v}_j^\top \mathbf{z}_{i-1,j}|$ instead of searching through all the remaining options. The Sherman Morrison iteration with partial pivoting is provided in Algorithm 3.

Algorithm 3 SMI with Partial Pivoting. Solve $(\mathbf{Z}_0^{-1} + \mathbf{U}\mathbf{V}^\top)\mathbf{x} = \mathbf{b}$

compute $\mathbf{x}_0 = \mathbf{Z}_0\mathbf{b}$ and $\mathbf{z}_{0,j} = \mathbf{Z}_0\mathbf{u}_j$ for $1 \le j \le k$
$\mathbf{p} = (1, 2 \ldots, k)^\top$
for $i = 1, 2, \ldots k$ **do**
  t=1
  while $1 + \mathbf{v}_{\mathbf{p}(j)}^\top \mathbf{z}_{i-1,\mathbf{p}(j)} = 0$ **do**
    swap $\mathbf{p}(i), \mathbf{p}(i+t)$
    $t = t + 1$
  end while

$$\mathbf{x}_i = \mathbf{x}_{i-1} - \frac{\mathbf{v}_{\mathbf{p}(i)^\top \mathbf{x}_{i-1}}}{1 + \mathbf{v}_{\mathbf{p}(i)}^\top \mathbf{Z}_{i-1,\mathbf{p}(i)}} \mathbf{z}_{i-1,\mathbf{p}(i)}$$

  for $j = i+1, \ldots, k$ **do**

$$\mathbf{z}_{i,\mathbf{p}(j)} = \mathbf{z}_{i-1,\mathbf{p}(j)} - \frac{\mathbf{v}_{\mathbf{p}(i)}^\top \mathbf{z}_{i-1,\mathbf{p}(j)}}{1 + \mathbf{v}_{\mathbf{p}(i)}^\top \mathbf{z}_{i-1,\mathbf{p}(i)}} \mathbf{z}_{i-1,\mathbf{p}(i)}$$

  end for
end for

Let us compare Algorithm 1 with Algorithm 2. When we do not use pivoting, updating $\mathbf{x}_i$ is dependent on the invertibility of $\mathbf{Z}_0^{-1} + \mathbf{u}_1\mathbf{v}_1^\top + \cdots + \mathbf{u}_i\mathbf{v}_i^\top$ or $\mathbf{Z}_{i-1}^{-1} + \mathbf{u}_i\mathbf{v}_i^\top$. Pivoting allows us to being able to update to $\mathbf{x}_i$ if there is a j between i and k inclusively such that $\mathbf{Z}_{i-1}^{-1} + \mathbf{u}_j\mathbf{v}_j^\top$ is invertible.

Note that example 3.1.1 works with pivoting. Pivoting also helps stability if it is enforced when $|1 + \mathbf{v}_i\mathbf{z}_{i-1,i}|$ is close to zero. This will avoid situations where we divide by a very small number and risk propagating error.

Unfortunately, Algorithm 2 does not work in the general case of $\mathbf{A}$ non-singular with arbitrary $\mathbf{Z}_0^{-1}, \mathbf{U}$, and $\mathbf{V}^\top$. Below is an example to illustrate when the Sherman Morrison iteration with pivoting fails.

Example 3.1.3. *Let*

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix},$$

with

$$\mathbf{U} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix},$$

and

$$\mathbf{V}^\top = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix},$$

and $\mathbf{Z}_0^{-1} = \mathbf{I}_3$. *Notice that* $\det(\mathbf{A}) = -2$. *However*

$$\mathbf{Z}_0^{-1} + \mathbf{u}_i\mathbf{v}_i^\top = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

is singular for all possible i. So Algorithm 2 will not be able to update x_0 to x_1.

Although pivoting does make the process of solving the linear system more robust, it still may fail.

3.1.2 Splitting

We now develop the idea of splitting.

Suppose that we are at the ith iteration in Algorithm 1. We know that updating $\mathbf{x}_i$ to $\mathbf{x}_{i+1}$ is dependent on the existence of $\mathbf{Z}_{i+1} = (\mathbf{Z}_i^{-1} + \mathbf{u}_{i+1}\mathbf{v}_{i+1}^\top)^{-1}$, but more importantly on the

fact that $1 + \mathbf{v}_{i+1}\mathbf{Z}_i\mathbf{u}_{i+1} \neq 0$. When $1 + \mathbf{v}_{i+1}\mathbf{Z}_i\mathbf{u}_{i+1} = 0$, we split

$$\mathbf{u}_{i+1}\mathbf{v}_{i+1}^\top = \frac{1}{2}\mathbf{u}_{i+1}\mathbf{v}_{i+1}^\top + \frac{1}{2}\mathbf{u}_{i+1}\mathbf{v}_{i+1}^\top, \tag{3.11}$$

then re-define $\mathbf{u}_{i+1}\mathbf{v}_{i+1}^\top = \frac{1}{2}\mathbf{u}_{i+1}\mathbf{v}_{i+1}^\top$ and send the other half of the outer product to the end of the line of outer products, making $\mathbf{u}_{k+1}\mathbf{v}_{k+1}^\top = \frac{1}{2}\mathbf{u}_{i+1}\mathbf{v}_{i+1}^\top$.

Since $1 + \mathbf{v}_{i+1}\mathbf{Z}_i\mathbf{u}_{i+1} = 0$ we see that $1 + \frac{1}{2}\mathbf{v}_{i+1}\mathbf{Z}_i\mathbf{u}_{i+1} = \frac{1}{2}$ and so we can update $\mathbf{x}_i$ to $\mathbf{x}_{i+1}$.

The Sherman Morrison iteration with splitting is described in Algorithm 4.

Algorithm 4 SMI with Splitting. Solve $(\mathbf{Z}_0^{-1} + \mathbf{U}\mathbf{V}^\top)\mathbf{x} = \mathbf{b}$

$i = 0$ compute $\mathbf{x}_0 = \mathbf{Z}_0\mathbf{b}$ and $\mathbf{z}_{0,j} = \mathbf{Z}_0\mathbf{u}_j$ for $1 \leq j \leq k$
while $i \leq k$ **do**
  $i = i + 1$
  if $1 + \mathbf{v}_i^\top\mathbf{z}_{i-1,i} = 0$ **then**
    $\mathbf{v}_i^\top = \frac{1}{2}\mathbf{v}_i^\top, \mathbf{z}_{i-1,k+1} = \mathbf{z}_{i-1}\mathbf{u}_i, k = k + 1$
  end if

$$\mathbf{x}_i = \mathbf{x}_{i-1} - \frac{\mathbf{v}_i^\top\mathbf{x}_{i-1}}{1 + \mathbf{v}_i^\top\mathbf{z}_{i-1,i}}\mathbf{z}_{i-1,i}$$

  for $j = i + 1, \ldots, k$ **do**

$$\mathbf{z}_{i,j} = \mathbf{z}_{i-1,j} - \frac{\mathbf{v}_i^\top\mathbf{z}_{i-1,j}}{1 + \mathbf{v}_i^\top\mathbf{z}_{i-1,i}}\mathbf{z}_{i-1,i}$$

  end for
end while

The benefit of this technique is that we can always update $\mathbf{x}_i$. The potential disadvantage is that an additional iteration is required every time a splitting occurs. Next we prove that the Sherman Morrison iteration with splitting is a direct method because it will converge to the solution in a finite number of steps.

Theorem 3.1.4. *Given a system* $\mathbf{A}\mathbf{x} = \mathbf{b}$ *where* $\mathbf{A}$ *is non-singular and* $\mathbf{A} = \mathbf{Z}_0^{-1} + \mathbf{U}\mathbf{V}^\top$, *Algorithm 4 will converge to* $\mathbf{A}^{-1}\mathbf{b}$ *in a finite number of steps.*

Proof. Recall that

$$\min\{\|\mathbf{E}\|_2 : \mathbf{A} + \mathbf{E} \text{ singular}\} = \frac{1}{\|\mathbf{A}^{-1}\|_2}. \tag{3.12}$$

As we run the splitting algorithm, the worst case scenario is that we keep splitting all the remaining $\mathbf{u}_j\mathbf{v}_j^\top$. Let us denote $\mathbf{u}_{j(old)}\mathbf{v}_{j(old)}^\top$ as the original columns of $\mathbf{U}$ and $\mathbf{V}^\top$. Recall that

k is the number of outer products in the decomposition of $\mathbf{A}$ given in (3.4). Choose $b \in \mathbb{N}$ such that

$$\frac{1}{2^b} \max\{\|\mathbf{u}_{j(old)}\mathbf{v}_{j(old)}^\top\|_2 : 1 \leq j \leq k\} < \frac{1}{k\|\mathbf{A}^{-1}\|_2}.$$

If Algorithm 4 doesn't repeatedly split all the remaining vectors, it will end after a finite number of steps. Let us assume that Algorithm 4 has split all $\mathbf{u}_{j(old)}\mathbf{v}_{j(old)}^\top$ b times. This is the only case that is necessary to prove the theorem.

We will show that all the remaining $\mathbf{Z}_i^{-1} + \mathbf{u}_{i+1}\mathbf{v}_{i+1} + \cdots + \mathbf{u}_k\mathbf{v}_k^\top$ are invertible. So each of the remaining $\mathbf{Z}_j$ exists for $i+1 \leq j \leq k$ and we can update $\mathbf{x}_i$ all the way to the solution of the system without having to split anymore.

Let j be inclusively between i and k. Notice

$$\mathbf{A} + (-\mathbf{u}_{j+1}\mathbf{v}_{j+1}^\top - \cdots - \mathbf{u}_k\mathbf{v}_k^\top) = \mathbf{Z}_i^{-1} + \mathbf{u}_{i+1}\mathbf{v}_{i+1}^\top + \cdots + \mathbf{u}_j\mathbf{v}_j^\top,$$

and

$$\begin{aligned}
\| - \mathbf{u}_{j+1}\mathbf{v}_{j+1}^\top - \cdots - \mathbf{u}_k\mathbf{v}_k^\top\|_2 &= \|\mathbf{u}_{j+1}\mathbf{v}_{j+1}^\top + \cdots + \mathbf{u}_k\mathbf{v}_k^\top\|_2 \\
&\leq \|\mathbf{u}_{j+1}\mathbf{v}_{j+1}^\top\|_2 + \cdots + \|\mathbf{u}_k\mathbf{v}_k^\top\|_2 \\
&\leq \frac{k-(j+1)}{2^b} \max\{\|\mathbf{u}_{j(old)}\mathbf{v}_{j(old)}^\top\|_2 : 1 \leq j \leq k\} \\
&\leq \frac{k-(j+1)}{k\|\mathbf{A}^{-1}\|_2} \\
&< \frac{1}{\|\mathbf{A}^{-1}\|_2}.
\end{aligned}$$

So, from (3.12), $\mathbf{Z}_i^{-1} + \mathbf{u}_1\mathbf{v}_1^\top \ldots \mathbf{u}_j\mathbf{v}_j^\top$ is non-singular therefore $\mathbf{Z}_j$ exists. We may update to $\mathbf{x}_j$ for the rest of algorithm with no splitting. We have our result. □

Algorithm 4 will work in the general case where $\mathbf{A}$ is nonsingular. Algorithm 4 will also work with any non-singular choice of $\mathbf{Z}_0^{-1}$ and any choice of $\mathbf{U}$ and $\mathbf{V}$.

Corollary. *Given a system* $\mathbf{Ax} = \mathbf{b}$, *where* $\mathbf{A}$ *is non-singular, there exists a* $\mathbf{Z}_0^{-1}, \mathbf{U}, \mathbf{V}^\top$ *such that* $\mathbf{A} = \mathbf{Z}_0^{-1} + \mathbf{UV}^\top$, *and Algorithm* 1 *and* 2 *will successfully solve the system.*

Proof. Start with any choice of $\mathbf{Z}_0^{-1}, \mathbf{U}$, and $\mathbf{V}^\top$, that satisfy (3.4). Run Algorithm 4. As splitting occurs add a column vector to $\mathbf{U}$ on right side and add row vector to bottom of $\mathbf{V}^\top$. At end of algorithm 3 we have $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}^\top$. Running Algorithm 1 with $\hat{\mathbf{U}}, \hat{\mathbf{V}}^\top$, and $\mathbf{Z}_0^{-1}$ will converge in a finite number of steps trivially. □

To conclude this section we offer a visualization of what are necessary and sufficient conditions for the Sherman Morrison iteration to run without pivoting, with pivoting, and with splitting:

Figure 3.2: This is a visualization of the necessary and sufficient conditions for the Sherman Morrison iteration to run without pivoting when $k = 3$. All the nodes of the graph that are black must be invertible.

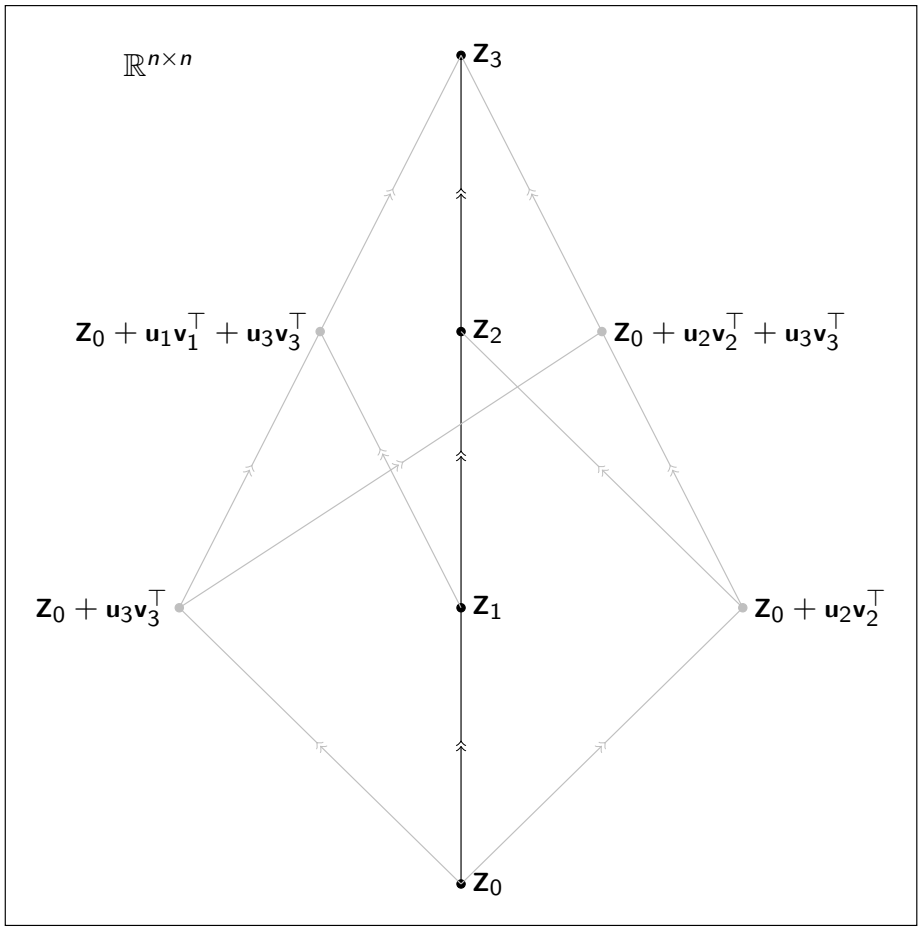


Figure 3.3: This is a visualization of the necessary and sufficient conditions for the Sherman Morrison iteration to run with pivoting when $k = 3$. We need one directed path from $\mathbf{Z}_0$ to $\mathbf{Z}_3$ to exist where all the matrices at the nodes we pass through are invertible.

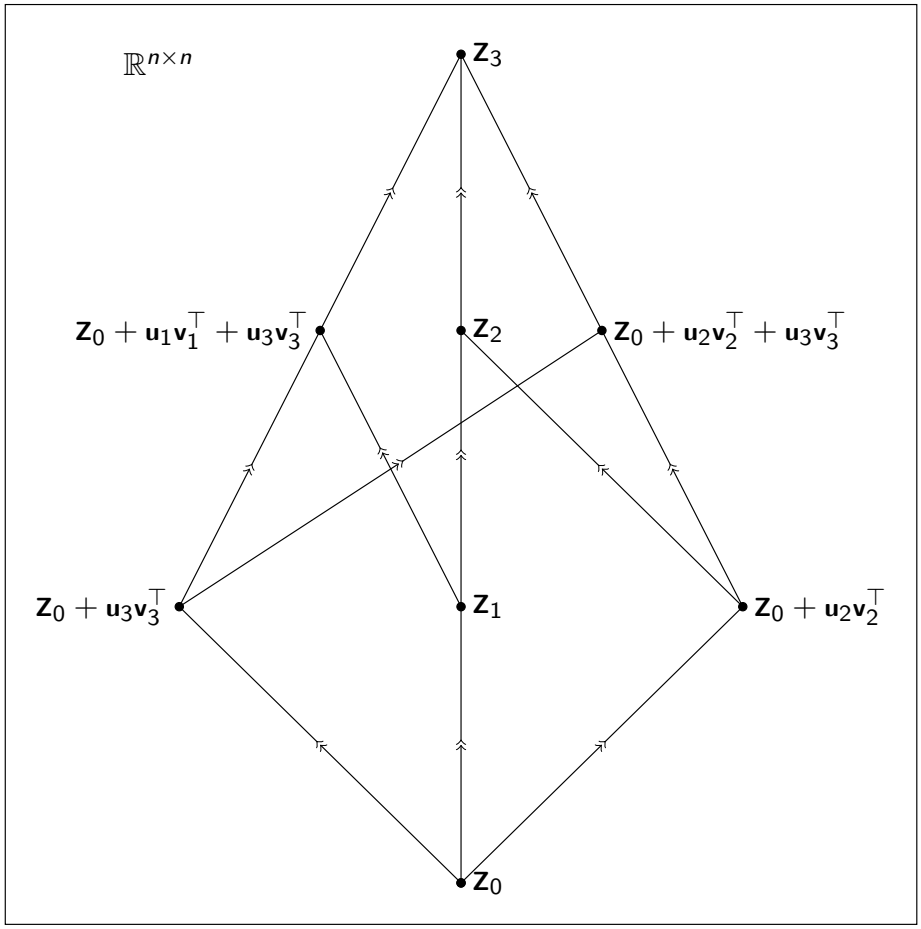
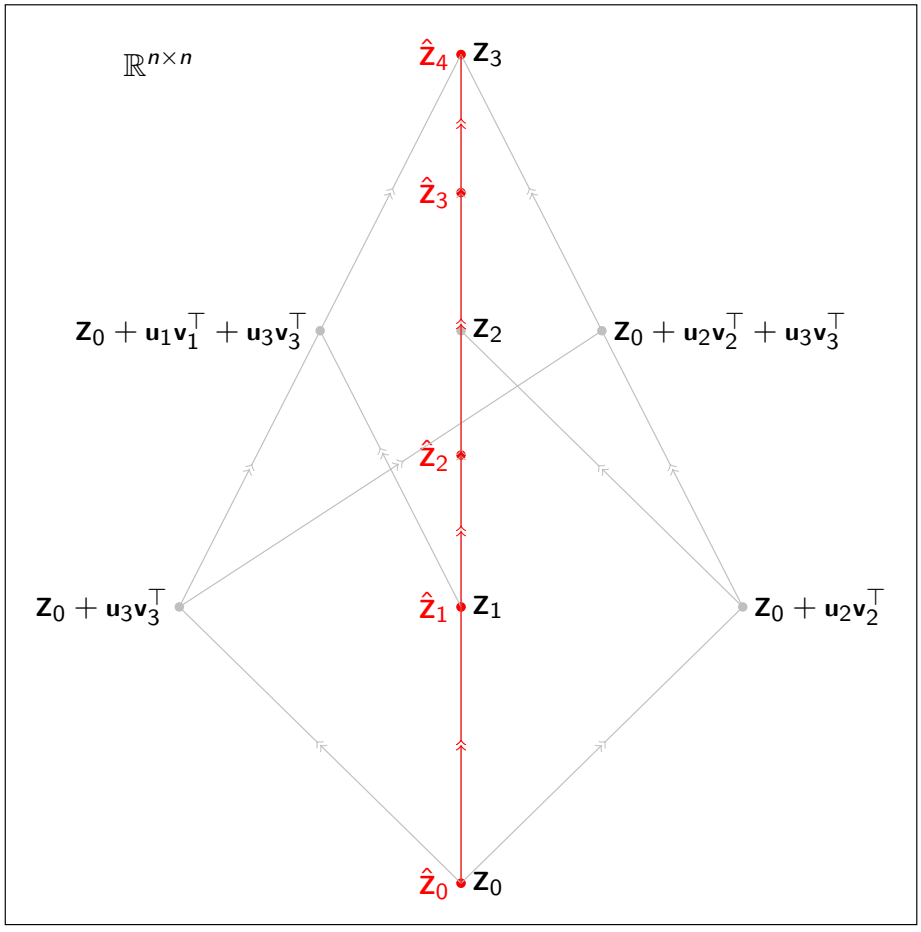


Figure 3.4: This is a visualization of the necessary and sufficient conditions for the Sherman Morrison iteration to run with splitting when $k = 3$. The splitting algorithm makes its own directed path to $\mathbf{Z}_3$ where every node it lands on (by design) is invertible. This is guaranteed to work as long as $\mathbf{A}$ is invertible.



3.2 Application to Regularized Least Squares Problem

Recall we are interested in solving the regularized least squares problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + \lambda^2\|\mathbf{L}\mathbf{x}\|_2^2, \tag{3.13}$$

where $\mathbf{L}\in\mathbb{R}^{\ell\times n}, \mathbf{L}^\top\mathbf{L}$ is easily invertible (meaning $\mathbf{L}$ is full column rank), and $\mathbf{A}\in\mathbb{R}^{m\times n}$ with $n \gg m$, and $\lambda > 0$. The normal equations are given by

$$(\mathbf{A}^\top\mathbf{A} + \lambda^2\mathbf{L}^\top\mathbf{L})\mathbf{x} = \mathbf{A}^\top\mathbf{b}. \tag{3.14}$$

We can solve the system (3.14) using the Sherman Morrison iteration. Let $\mathbf{Z}_0^{-1} = \lambda^2\mathbf{L}^\top\mathbf{L}, \mathbf{U} = \mathbf{A}^\top$, and $\mathbf{V} = \mathbf{A}^\top$, then Algorithm 1 will solve the system, with no breakdowns. This is because each of the $\mathbf{Z}_i^{-1} + \mathbf{u}_{i+1}\mathbf{v}_{i+1}^\top$ are positive definite and therefore non-singular, which will be proven in the preceding sentences. Notice $\mathbf{L}^\top\mathbf{L}$ is positive definite. Let $\mathbf{w}\neq 0$ and let $\mathbf{a}_i$ denote the ith row in $\mathbf{A}$. We have:

$$\begin{aligned}
\mathbf{w}^\top(\mathbf{Z}_{i-1}^{-1} + \mathbf{u}_i\mathbf{v}_i^\top)\mathbf{w} &= \mathbf{w}^\top(\mathbf{Z}_0^{-1} + \mathbf{u}_1\mathbf{v}_1^\top + \mathbf{u}_2\mathbf{v}_2^\top + \cdots + \mathbf{u}_i\mathbf{v}_i^\top)\mathbf{w} \\
&= \mathbf{w}^\top(\lambda^2\mathbf{L}^\top\mathbf{L} + \mathbf{a}_1^\top\mathbf{a}_1 + \mathbf{a}_2^\top\mathbf{a}_2 + \cdots + \mathbf{a}_i^\top\mathbf{a}_i)\mathbf{w} \\
&= \mathbf{w}^\top\lambda^2\mathbf{L}^\top\mathbf{L}\mathbf{w} + \mathbf{w}^\top\mathbf{a}_1^\top\mathbf{a}_1\mathbf{w} + \cdots + \mathbf{w}^\top\mathbf{a}_i^\top\mathbf{a}_i\mathbf{w} \\
&= \lambda^2\mathbf{w}^\top\mathbf{L}^\top\mathbf{L}\mathbf{w} + (\mathbf{a}_1\mathbf{w})^2 + \cdots + (\mathbf{a}_i\mathbf{w})^2 > 0,
\end{aligned}$$

since $\mathbf{L}^\top\mathbf{L}$ is positive definite and so $\lambda^2\mathbf{w}^\top\mathbf{L}^\top\mathbf{L}\mathbf{w} > 0$.
So $\mathbf{Z}_i$ are all positive definite, and thus non-singular. So no pivoting or splitting is needed to solve (3.14). The Sherman Morrison Iteration for solving (3.14) is provided in Algorithm 5.

Algorithm 5 SMI to Solve $(\lambda^2\mathbf{L}^\top\mathbf{L} + \mathbf{A}^\top\mathbf{A})\mathbf{x} = \mathbf{A}^\top\mathbf{b}$

1: compute $\mathbf{x}_0 = \frac{1}{\lambda^2}(\mathbf{L}^\top\mathbf{L})^{-1}\mathbf{A}^\top\mathbf{b}$ and $\mathbf{z}_{0,j} = \frac{1}{\lambda^2}(\mathbf{L}^\top\mathbf{L})^{-1}\mathbf{a}_j^\top$ for $1\le j\le m$
2: **for** $i = 1, 2, \ldots m$ **do**
3:
$$\mathbf{x}_i = \mathbf{x}_{i-1} - \frac{\mathbf{a}_i\mathbf{x}_{i-1}}{1+\mathbf{a}_i\mathbf{z}_{i-1,i}}\mathbf{z}_{i-1,i}$$
4: **for** $j = i+1, \ldots, m$ **do**
5:
$$\mathbf{z}_{i,j} = \mathbf{z}_{i-1,j} - \frac{\mathbf{a}_i\mathbf{z}_{i-1,j}}{1+\mathbf{a}_i\mathbf{z}_{i-1,i}}\mathbf{z}_{i-1,i}$$
6: **end for**
7: **end for**

If m is small, then both "for" loops are very small in Algorithm 5. So when the number of rows of $\mathbf{A}$ are small, this method could potentially be very quick in solving the system.

Recall that Algorithm 1 had two main issues: the fact that it could breakdown when $\mathbf{A}$ was non-singular, and also a stability problem when $1+\mathbf{v}_i^\top \mathbf{z}_{i-1,i}$ was very close to zero. We have shown Algorithm 5 will solve the system (3.14), Algorithm 5 is also stable. Notice

$$\begin{aligned} 1+\mathbf{v}_i^\top \mathbf{z}_{i-1,i} &= 1+\mathbf{v}_i^\top \mathbf{Z}_{i-1}\mathbf{u}_i \\ &= 1+\mathbf{a}_i^\top \mathbf{Z}_{i-1}\mathbf{a}_i. \end{aligned}$$

Since $\mathbf{Z}_{i-1}$ is positive definite, for $\mathbf{a}_i \neq 0$, $\mathbf{a}_i^\top \mathbf{Z}_{i-1}\mathbf{a}_i > 0$ and so $1+\mathbf{v}_i^\top \mathbf{Z}_{i-1,i} > 1$ for any i. We don't have to worry about error propagation due to dividing by near-zero numbers in Algorithm 5.

3.3 Efficiencies

The goal of this section is to compare the implementation of the Sherman Morrison iteration in Algorithm 5 with implementations via CGLS, LSQR, and LSMR [7, 11, 28, 29]. This discussion will be focused on the regularization matrix $\mathbf{L}$.

Suppose we are using Algorithm 5 to solve the regularized least squares problem (3.13). Algorithm 5 needs $\mathbf{A}, \mathbf{b}$ and $\frac{1}{\lambda^2}(\mathbf{L}^\top \mathbf{L})^{-1}$ as inputs. If $\mathbf{L} = \mathbf{I}$ is the identity matrix, then Algorithm 5 can be fashioned to take the regularization parameter λ as an input instead of $\frac{1}{\lambda^2}(\mathbf{L}^\top \mathbf{L})^{-1}$. LSQR, CGLS, and LSMR handle the $\mathbf{L} = \mathbf{I}$ case in the same way.

When $\mathbf{L}$ is not the identity matrix, CGLS, LSQR, and LSMR can be used to solve the Tikhonov problem,

$$\min_{\mathbf{x}\in\mathbb{R}^n} \left\| \begin{pmatrix} \mathbf{A} \\ \lambda \mathbf{L} \end{pmatrix} \mathbf{x} - \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} \right\|_2^2. \tag{3.15}$$

When $\mathbf{L}$ is invertible, (3.13) is equivalent to

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{L}^{-1}\mathbf{y} - \mathbf{b}\|_2^2 + \lambda^2 \|\mathbf{y}\|_2^2, \tag{3.16}$$

where $\mathbf{y} = \mathbf{L}\mathbf{x}$, and any iterative method can be used to solve (3.16).

At the end of Chapter 2 we saw an example of when the matrix $(\mathbf{L}^\top \mathbf{L})^{-1}$ is given without knowledge of $\mathbf{L}$ or $\mathbf{L}^{-1}$. In order to use CGLS, LSQR, or LSMR on the system (3.16) a Cholesky decomposition of $(\mathbf{L}^\top \mathbf{L})^{-1}$ would be necessary to find $\mathbf{L}^{-1}$. The Sherman Morrison iteration would not require such a decomposition. The Sherman Morrison iteration is a particularly advantageous approach in this situation, since a Cholesky Decomposition on $(\mathbf{L}^\top \mathbf{L})^{-1}$ could be costly.

The next chapter contains experiments comparing the Sherman Morrison iteration and CGLS, LSQR, and LSMR.

Chapter 4

Experiments

This chapter is split up into two parts. In the first part the Sherman Morrison Iteration solver for a regularized least squares problem will be compared to solving a regularized least squares problem via the Cholesky factorization and with the QR decomposition. In the second part the Sherman Morrison Iteration solver for a regularized least squares problem will be compared to more sophisticated solvers including LSQR, CGLS and LMSR.

4.1 Comparison with Classical Methods

Suppose we are given the regularized least squares problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + \lambda^2\|\mathbf{L}\mathbf{x}\|_2^2, \tag{4.1}$$

where $\mathbf{A}\in\mathbb{R}^{m\times n}, \mathbf{L}\in\mathbb{R}^{\ell\times n}, \lambda\in\mathbb{R}$ and $\mathbf{b}\in\mathbb{R}^m$. for suitable $m, n, \ell\in\mathbb{N}$. Let us also assume that $\mathbf{L}$ has full column rank. We could solve the normal equations (3.14) via Cholesky decomposition, since $\mathbf{A}^\top\mathbf{A}+\lambda^2\mathbf{L}^\top\mathbf{L}$ is symmetric positive definite. To carry out the Cholesky decomposition the matrix $\mathbf{A}^\top\mathbf{A}+\lambda^2\mathbf{L}^\top\mathbf{L}$ must be constructed. We could also rearrange the original minimization problem as in (3.15) and solve the system via a QR factorization. For more information on the Cholesky decomposition or QR factorization see [9].

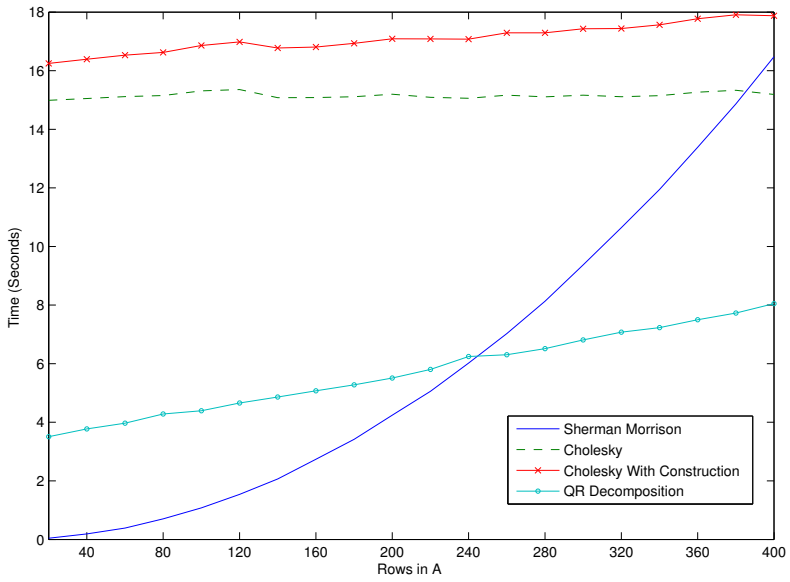
4.1.1 Experiment 1

least In the first experiment we constructed the regularized least squares problem in (4.1) where

- $\mathbf{A}\in\mathbb{R}^{m\times 10{,}000}$ with $m = 20\ldots 400$ increasing at intervals of 20.

- Entries of $\mathbf{A}$, $a_{i,j}$, are sampled from $\mathcal{N}(0, 1)$.
- Regularization paramater $\lambda = 1$. Regularization matrix $\mathbf{L} = \mathbf{I}_n$.
- The true solution $\mathbf{x} = \mathbf{1}$, with $\mathbf{b} = \mathbf{A}\mathbf{x}$ defined accordingly with Gaussian noise added (with variance of .01).
- We compared timings of Sherman Morrison iteration, QR factorization (solving (3.15)), and the Cholesky decomposition (solving (3.14)), including timings with and without construction of $\mathbf{A}^\top\mathbf{A} + \lambda^2\mathbf{L}^\top\mathbf{L}$).
- The experiment was repeated 30 times, averaged the times at each m value.
- See Figure 4.1 for results. The Sherman Morison iteration is beneficial over the other methods when the number of rows in $\mathbf{A}$ is sufficiently low.

Figure 4.1: Results of experiment 1, see section for details.



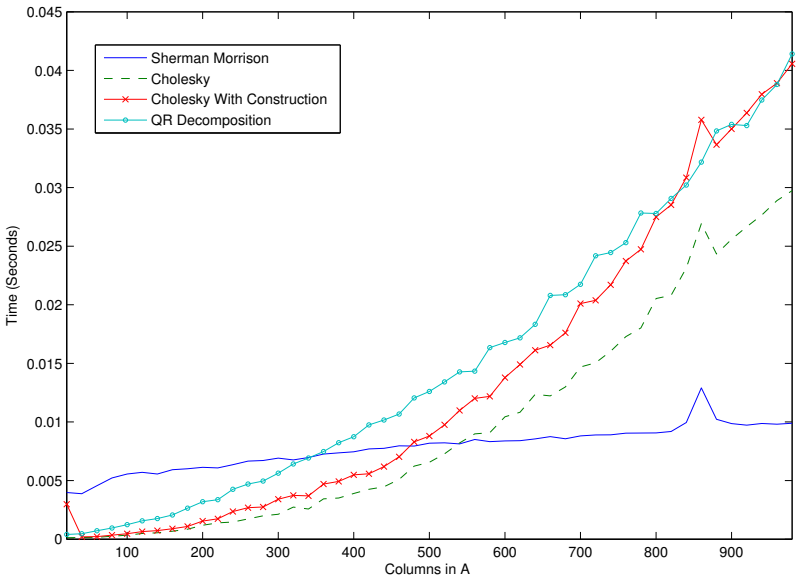
4.1.2 Experiment 2

In the second experiment we constructed the regularized least squares problem in (4.1) where

- $\mathbf{A} \in \mathbb{R}^{25\times n}$ with $n = 20 \ldots 1000$ increasing at intervals of 20.

- Entries of $\mathbf{A}$, $a_{i,j}$, are sampled from $\mathcal{N}(0,1)$.
- Regularization paramater $\lambda = 1$. Regularization matrix $\mathbf{L} = \mathbf{I}_n$.
- The true solution $\mathbf{x} = \mathbf{1}$, with $\mathbf{b} = \mathbf{A}\mathbf{x}$ defined accordingly with Gaussian noise added (with variance of .01).
- We compared timings of Sherman Morrison iteration, QR factorization (solving (3.15)), and the Cholesky decomposition (solving (3.14)), including timings with and without construction of $\mathbf{A}^\top\mathbf{A} + \lambda^2\mathbf{L}^\top\mathbf{L}$).
- The experiment was repeated 30 times, averaged the times at each n value.
- See Figure 4.2 for results. When the number of rows in $\mathbf{A}$ is low enough compared to the number of columns in $\mathbf{A}$ we see that the Sherman Morrison iteration becomes a preferable option compared to the other methods.

Figure 4.2: Results of experiment 2, see section for details.



We have shown that there are situations where the Sherman Morrison iteration will out perform the QR iteration and the Cholesky factorization methods for solving an underdetermined regularized least square problem.

4.2 Comparison with Modern Methods

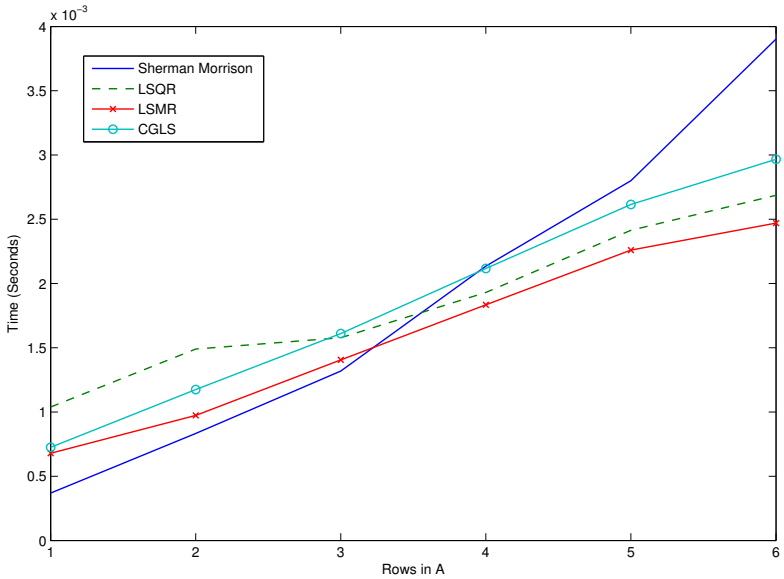
In this section we compare the Sherman Morrison iteration with three iterative solvers for the regularized least squares problems LSQR, CGLS, and LSMR [7, 11, 28, 29].

4.2.1 Experiment 3

In the third experiment we constructed the regularized least squares problem in (4.1) where

- $\mathbf{A} \in \mathbb{R}^{m \times 20{,}000}$ with $m = 1 \ldots 6$ increasing at intervals of 1.
- Entries of $\mathbf{A}$, $a_{i,j}$, are sampled from $\mathcal{N}(0, 1)$.
- Regularization paramater $\lambda = 1$. Regularization matrix $\mathbf{L} = \mathbf{I}_n$.
- The true solution $\mathbf{x} = \mathbf{1}$, with $\mathbf{b} = \mathbf{A}\mathbf{x}$ defined accordingly with Gaussian noise added (with variance of .01).
- We compared timings of Sherman Morrison iteration, LSQR, CGLS, and LSMR (All three solving (4.1))
- The experiment was repeated 30 times, averaged the times at each m value.
- See Figure 4.3 for results. The Sherman Morison is comparable to the other methods.

Figure 4.3: Results of experiment 3, see section for details.

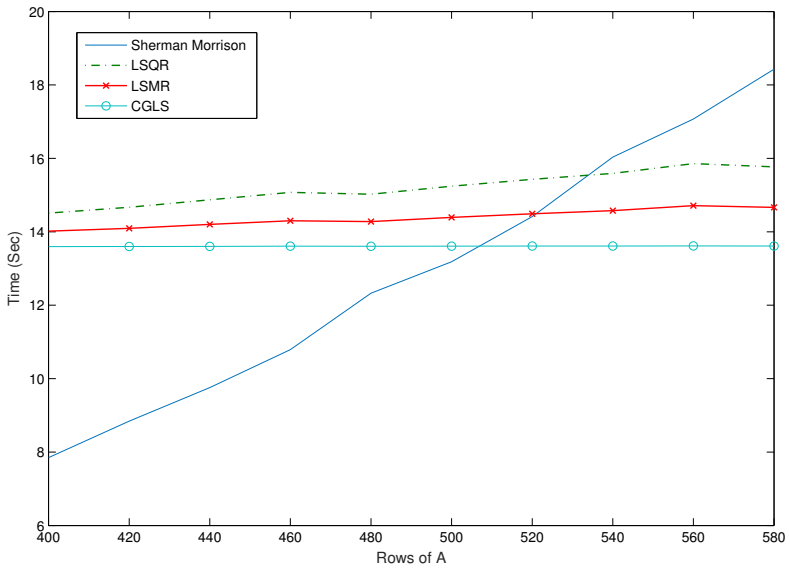


4.2.2 Experiment 4

In the forth experiment we constructed the regularized least squares problem in (4.1) where

- $\mathbf{A} \in \mathbb{R}^{m \times 10{,}000}$ with $m = 400 \ldots 600$ increasing at intervals of 20.
- Entries of $\mathbf{A}$, $a_{i,j}$, are sampled from $\mathcal{N}(0, 1)$.
- Regularization paramater $\lambda = 1$. Regularization matrix $\mathbf{L}$ is the difference operator, given as $(\mathbf{L}^\top \mathbf{L})^{-1}$. The regularization matrix $\mathbf{L}$ or $\mathbf{L}^{-1}$ is not made available at the beginning of the experiment, it must be found via a Cholesky decomposition of $\mathbf{L}^\top \mathbf{L}$ or $(\mathbf{L}^\top \mathbf{L})^{-1}$.
- The true solution $\mathbf{x} = \mathbf{1}$, with $\mathbf{b} = \mathbf{A}\mathbf{x}$ defined accordingly with Gaussian noise added (with variance of .01).
- We compared timings of Sherman Morrison iteration with LSQR, CGLS, and LSMR, all three solving (3.16) after a Cholesky decomposition of $(\mathbf{L}^\top \mathbf{L})^{-1}$ (timings of the Cholesky decomposition included in timings for LSQR, CGLS, and LSMR).
- The experiment was repeated 30 times, averaged the times at each m value.
- See Figure 4.4 for results. The Sherman Morison iteration is beneficial over the other methods when the number of rows in $\mathbf{A}$ is sufficiently low.

Figure 4.4: Results of experiment 4, see section for details.



In the case where $\mathbf{L} = \mathbf{I}$ the Sherman Morrison iteration is comparable to LSQR, LSMR, and CGLS in a narrow set of circumstances. When $\mathbf{L}$ is non-trivial, there are cases where the Sherman Morrison iteration is beneficial to use, particularly in the case where the matrix $(\mathbf{L}^\top\mathbf{L})^{-1}$ is given and not $\mathbf{L}$ or $\mathbf{L}^{-1}$.

Chapter 5

Conclusion

The Sherman Morrison iteration has been developed as an effective direct solver of an extremely underdetermined regularized least square problem, which appear in a medley of real world problems. The general Sherman Morrison iteration was discussed as developed by P. Maponi [22, 10], methods of partial pivoting and splitting were introduced, and then the Sherman Morrison iteration was applied as a solver of the regularized least squares problem. Chapter 4 showcased some situations were the Sherman Morrison iteration could be preferable to use as a solver to the regularized least squares problem over sophisticated methods like CGLS, LSQR, and LSMR.

There is future work regarding the Sherman Morrison iteration that still needs to be done. It would be beneficial to have some formal error analysis to the Sherman Morrison iteration. Finding a specific application where the Sherman Morrison Iteration can be successfully applied on non-synthetic data would be advantageous.

The Woodbury matrix identity [18] could be used to develop a quicker Sherman Morrison iteration algorithm . A method of this nature might be able to incorporate more level three basic linear algebra subprograms, which could potentially speed up run times. It could also potentially prevent breakdowns that appeared in the general Sherman Morrison iteration with and without pivoting.

Matlab code for Algorithm 5 is available in the appendix.

Appendix A

Matlab Codes

This chapter will contain Matlab codes for

1. the regularized least squares Sherman Morrison iteration (scaler regularization), and
2. the regularized least squares Sherman Morrison iteration (arbitrary regularization matrix).

Both codes are based on Algorithm 5. This is the Sherman Morrison iteration for the regularized least squares problem for a non-identity regularization matrix.

```
function [x, info] = shermanMorrisonIteration(g,A,Z0);
%
% function [s, info] = shermanMorrisonIteration(g,A,Z0)
%
% Authors:
%   (c)  Julianne Chung (e-mail: jmchung@vt.edu)
%   and  Matthias Chung (e-mail: mcchung@vt.edu)
%   and  Joseph Slagel (email:slagelj@vt.edu) in December 2014
%
% Description:
%   Given a least squares problem
%               min_x ||Ax - b||_2^2 + lambda^2||L x||_2^2
%    with a invertible matrix Z_0=1/lambda^2*inv(L'*L), this methods solves
%    the normal equations
%                       (A'A + inv(Z0))x =  g
%   Where g=A'b. This method computes the vector
%                        x = inv((A'A + inv(Z0)))g
%   by using the Sherman-Morrison formula and avoiding to build any matrix.
%
% Input arguments:
%   g    - A'b
```

```
%   A    - the A matrix in system Ax-b
%   Z0   - Z0=1/lambda^2*inv(L'L) where L is regularization matrix
%
% Output arguments:
%   x    - Solution to regularized least squares problem
%   info - #
%     .X - step approximations
[l,n] = size(A); % get size of A
x = Z0*g;        % initialize solution vector
U=Z0*A';         % initialize Z_{i,j} vectors
if nargout > 1, info.X = zeros(n,l); end
for i = 1:l           % loop over all rows of A
    U(:,i+1:l) = U(:,i+1:l) - bsxfun(@times,A(i,:)...
        *U(:,i+1:l)/(1+A(i,:)*U(:,i)),U(:,i));
    %Update Z_{i,j} vectors
    x = x - (A(i,:)*x)/(1+A(i,:)*U(:,i))*U(:,i)
    %Update x via Sherman Morisson formula
    if nargout > 1, info.X(:,i) = x; end
end
```

This is the Sherman Morrison iteration for the regularized least squares problem when the regularization matrix is the identity matrix.

```
function [x, info] = shermanMorrisonIterationScaler(g,A,lambda);
%
% function [s, info] = shermanMorrisonIteration(g,A,Z0)
%
% Authors:
%   (c)  Julianne Chung (e-mail: jmchung@vt.edu)
%   and  Matthias Chung (e-mail: mcchung@vt.edu)
%   and  Joseph Slagel (email:slagelj@vt.edu) in December 2014
%
% Description:
%   Given a least squares problem
%               min_x ||Ax - b||_2^2 + lambda^2||x||_2^2
%    This methods solves the normal equations
%                       (A'A + 1/lambda^2*I)x =  g
%   Where I is the identity matrix and g=A'*b This method computes the vector
%                        x = inv(A'A + 1/lambda^2*I)g
%   by using the Sherman-Morrison formula and avoiding to build any matrix.
%
% Input arguments:
%   g    - A'b
%   A    - the A matrix in system Ax-b
%   lambda   - regularization paramater
% Output arguments:
%   x    - Solution to regularized least squares problem
%   info - #
%     .X - step approximations
lambda=1/lambda^2;
```

```
[l,n] = size(A); % get size of A
x = lambda*g;         % initialize solution vector
U=lambda*A';          % initialize Z_{i,j} vectors
if nargout > 1, info.X = zeros(n,l); end
for i = 1:l           % loop over all rows of A
    U(:,i+1:l) = U(:,i+1:l) - bsxfun(@times,A(i,:)...
        *U(:,i+1:l)/(1+A(i,:)*U(:,i)),U(:,i));
    %Update Z_{i,j} vectors
    x = x - (A(i,:)*x)/(1+A(i,:)*U(:,i))*U(:,i);
    %Update x via Sherman Morisson formula
    if nargout > 1, info.X(:,i) = x; end
end
```

Bibliography

[1] A Bakushinsky and A Goncharsky. *Ill-posed problems: theory and applications*. Springer Netherlands, 1994.

[2] Mario Bertero and Patrizia Boccacci. *Introduction to inverse problems in imaging*. CRC press, 2010.

[3] Alfred M. Bruckstein, David L. Donoho, and Michael Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, 2009.

[4] D. Calvetti and L. Reichel. Tikhonov regularization of large scale problems. *BIT*, 43:263–283, 2003.

[5] D. Calvetti and L. Reichel. Tikhonov regularization with a solution constraint. *SIAM Journal on Scientific Computing*, 26:224–239, 2004.

[6] Daniela Calvetti and Erkki Somersalo. *Introduction to Bayesian Scientific Computing: Ten Lectures on Subjective Computing (Surveys and Tutorials in the Applied Mathematical Sciences)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[7] Sou-Cheng T. Choi. *Iterative methods for singular linear equations and least-squares problems*. PhD thesis, Stanford University, 2006.

[8] Mark E. Davison. The ill-conditioned nature of the limited angle tomography problem. *SIAM Journal on Applied Mathematics*, 43(2):428–448, 1983.

[9] J.W. Demmel. *Applied Numerical Linear Algebra*. Miscellaneous Bks. Society for Industrial and Applied Mathematics, 1997.

[10] N Egidi and P. Maponi. A sherman-morrison approach to the solution of linear systems. *Journal of Computational and Applied Mathematics*, 189(1-2):703–718, May 2006.

[11] David Chin-Lung Fong and Michael Saunders. Lsmr: An iterative algorithm for sparse least-squares problems. *SIAM Journal on Scientific Computing*, 33(5):2950–2971, 2011.

[12] W. Gautschi. *Numerical Analysis*. SpringerLink : Bücher. Birkhäuser Boston, 2011.

[13] G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

[14] C. W. Groetsch. *The Theory of Tikhonov Regularization for Fredholm Integral Equations of the First Kind*. Pitman, Boston, 1984.

[15] P. Hansen. *Discrete Inverse Problems*. Society for Industrial and Applied Mathematics, 2010.

[16] P. C. Hansen. *Rank-deficient and Discrete Ill-posed Problems*. SIAM, Philadelphia, PA, 1997.

[17] P.C. Hansen, V. Pereyra, and G. Scherer. *Least Squares Data Fitting with Applications*. Least Squares Data Fitting with Applications. Johns Hopkins University Press, 2012.

[18] N. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, second edition, 2002.

[19] A. Scottedward Hodel. Solution of underdetermined sylvester equations in sensor array signal processing. *Linear Algebra and its Applications*, 249(1-3):1–14, December 1996.

[20] Timothy R. Lezon, Jayanth R. Banavar, Marek Cieplak, Amos Maritan, and Nina V. Fedoroff. Using the principle of entropy maximization to infer genetic interaction networks from gene expression patterns. *Proceedings of the National Academy of Sciences*, 103(50):19033–19038, 2006.

[21] Y. Q. Li and S. Liu. Image reconstruction algorithm for ultrasound tomography based on transmission mode. *Key Engineering Materials*, 474-476:754–759, 2011.

[22] P. Maponi. The solution of linear systems by using the sherman-morrison formula. *Linear Algebra and its Applications*, 420(2-3):276–294, January 2007.

[23] Adam Margolin and Andrea Califano. Theory and limitations of genetic network inference from microarray data. *Annals of the New York Academy of Sciences*, 1115(1):51–72, 2007.

[24] V.G. Maz'ya and T.O. Shaposhnikova. *Jacques Hadamard: A Universal Mathematician*. History of mathematics. American Mathematical Society, 1998.

[25] J. Mueller and S. Siltanen. *Linear and Nonlinear Inverse Problems with Practical Applications*. SIAM, Philadelphia, 2012.

[26] S.M. Oganyesyan, V.I.Starostenko, and M.G. Oganyesyan. The dual method of solving linear incorrect under-determined geophysical problems. *Izvestiya of the Academy of Sciences, USSR.: Physics of the solid earth*, 20:450, 1984.

[27] D.P. O'Leary. Near-optimal parameters for Tikhonov and other regularization methods. *SIAM Journal on Scientific Computing*, 23(4):1161–1171, 2001.

[28] Christopher C. Paige and Michael A. Saunders. Algorithm 583: Lsqr: Sparse linear equations and least squares problems. *ACM Trans. Math. Softw.*, 8(2):195–209, June 1982.

[29] Christopher C. Paige and Michael A. Saunders. Lsqr: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Softw.*, 8(1):43–71, 1982.

[30] Asher Peres. Tomographic reconstruction from limited angular data. *Journal of Computer Assisted Tomography*, 3(6), December 1979.

[31] Lothar Reichel, Fiorella Sgallari, and Qiang Ye. Tikhonov regularization based on generalized Krylov subspace methods. *Applied Numerical Mathematics*, 62(9):1215–1228, 2012.

[32] R. Snieder and J. Trampert. Inverse problems in geophysics. In Armand Wirgin, editor, *Wavefield Inversion*, volume 398 of *International Centre for Mechanical Sciences*, pages 119–190. Springer Vienna, 1999.

[33] Kwok-Cheong Tam, Victor Mendez, and Burns Macdonald. 3-d object reconstruction in emission and transmission tomography with limited angular input. *Nuclear Science, IEEE Transactions on*, 26(2):2797–2805, April 1979.

[34] Andre˘i Nikolaevich Tikhonov, Vasili˘i Arsenin, and Fritz John. *Solutions of ill-posed problems*. Winston Washington, DC, 1977.

[35] C. R. Vogel. *Computational Methods for Inverse Problems*. SIAM, Philadelphia, PA, 2002.