

Distribution Agreement

In presenting this thesis as a partial fulfillment of the requirements for a degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis in whole or in part in all forms of media, now or hereafter know, including display on the World Wide Web. I understand that I may select some access restrictions as part of the online submission of this thesis. I retain all ownership rights to the copyright of the thesis. I also retain the right to use in future works (such as articles or books) all or part of this thesis.

Hanyong Wu

April 05, 2017

Generalized Cross Validation for Ill-Posed Inverse Problems

by

Hanyong Wu

James G. Nagy
Advisor

Department of Mathematics and Computer Science

James G. Nagy
Advisor

Vicki S. Hertzberg
Committee Member

Christopher Curan
Committee Member

2017

Generalized Cross Validation for Ill-Posed Inverse Problems

by

Hanyong Wu

James G. Nagy

Advisor

Abstract of

A thesis submitted to the Faculty of Emory College
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Sciences with Honors

Department of Mathematics and Computer Science

2017

Abstract

Generalized Cross Validation for Ill-Posed Inverse Problems
By Hanyong Wu

In this thesis, we will introduce two popular regularization tools for ill-posed linear inverse problem, truncated singular value decomposition and Tikhonov regularization. After that we will implement them with the generalized cross validation (GCV) method to choose regularization parameters. We consider in particular problems that have noise in the measured data, noise in the matrix, and noise in both the measured data and the matrix. Numerical experiments are used to test the GCV method for each of these noise models.

Generalized Cross Validation for Ill-Posed Inverse Problems

by

Hanyong Wu

James G. Nagy

Advisor

A thesis submitted to the Faculty of Emory College
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelors of Science with Honors

Department of Mathematics and Computer Science

2017

Contents

1	Introduction	5
1.1	Background	6
1.2	Filtering	9
2	Regularization Methods	11
2.1	Truncated Singular Value Decomposition	11
2.2	Tikhonov Regularization	13
3	Choosing Regularization Parameters	16
3.1	Generalized cross validation for TSVD	18
3.2	Generalized cross validation for Tikhonov	19
3.3	Optimal Regularization Parameter	21
3.4	Optimal Regularization for TSVD	22
3.5	Optimal Regularization for Tikhonov	23
4	Numerical Experiments	24
4.1	Test problem experiments	25

4.1.1	With error in only b	25
4.1.2	With error in only A	31
4.1.3	With errors in both of A and b	37
4.2	Application to Image Deblurring	42
4.2.1	AtmosphericBlur10	46
4.2.2	AtmosphericBlur30	48
4.2.3	AtmosphericBlur50	52
5	Conclusions	55

List of Figures

1.1	Satellite image	6
2.1	Inverse singular values	14
3.1	Solutions for directly doing inverse function	17
3.2	Solutions for knowing the error norm	17
3.3	Flatness of GCV plot	20
4.1	Results for <code>deriv2</code>	26
4.2	Results for <code>phillips</code>	28
4.3	Results for <code>shaw</code>	29
4.4	Results for <code>heat</code>	30
4.5	Results for <code>deriv2</code>	32
4.6	Results for <code>phillips</code>	34
4.7	Results for <code>shaw</code>	35
4.8	Results for <code>heat</code>	36
4.9	Results for <code>deriv2</code>	38
4.10	Results for <code>phillips</code>	39

4.11 Results for shaw	40
4.12 Results for heat	41
4.13 The blurred graph and the true graph of satellite	46
4.14 Solutions using GCV regularization parameter	47
4.15 Relative residual with iteration	47
4.16 Solutions using optimal regularization parameter	48
4.17 The blurred graph and the true graph of satellite	48
4.18 Solutions using GCV regularization parameter	50
4.19 Relative residual with iteration	50
4.20 Solutions using optimal regularization parameter	51
4.21 The blurred graph and the true graph of satellite	52
4.22 Solutions using GCV regularization parameter	53
4.23 Relative residual with iteration	53
4.24 Solutions using optimal regularization parameter	54

Chapter 1

Introduction

An inverse problem in science is the process of calculating from a set of observations the causal factors that produced them: for example, calculating an image in computer tomography, source reconstructing in acoustics, or calculating the density of the Earth from measurements of its gravity field. This thesis will mainly focus on the inverse problem on image restoration.

Image restoration is the process of removing blur and noise from degraded images to recover an approximation of the original image. This field of imaging technology is becoming increasingly significant in many scientific applications such as military [6, 7], astronomy [6, 8, 9], surveillance [6, 7], microwave imaging [3] and so on.

For example, scientists use long-range telescopes to obtain pictures of distant stars. However, due to the distortion caused by the Earth's atmosphere and the random interfering light rays coming from different sources,

astronomers receive blurred images, such as that shown in Figure 1.1.

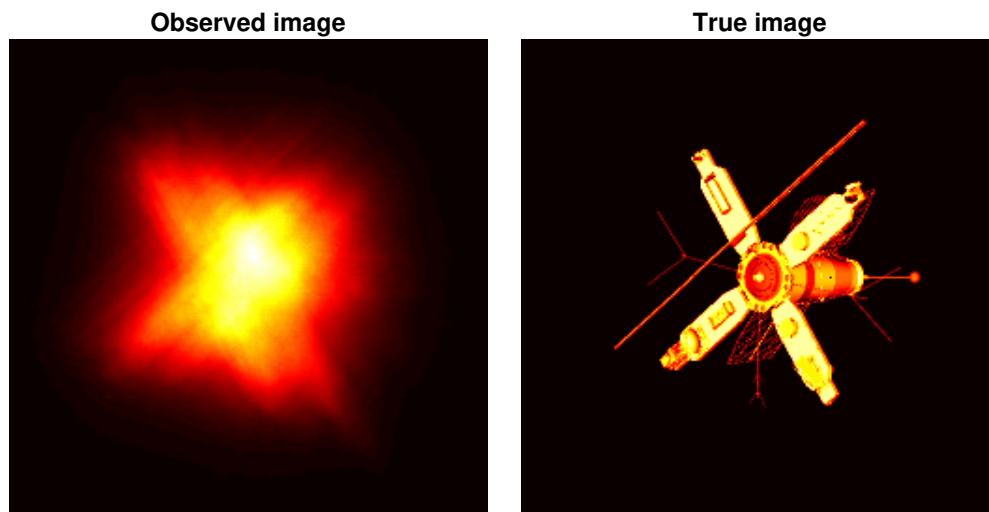


Figure 1.1: Satellite image

In each of these applications, obtaining clearer images can be accomplished by using computer programs to perform image enhancement techniques.

1.1 Background

In order to solve image restoration problems, the first step is to model the problem as a mathematical equation. This is often done using the linear problem

$$b = Ax_{true} + e \tag{1.1}$$

where b represents the blurred image, A is a blurring matrix, x_{true} represents the true image and e is the error (noise).

With the mathematical model in equation (1.1) in mind, the next step is to find a method that solves this equation. If we just simply assume e is small enough compared with b and A^{-1} exists, we can solve this problem by using $x_{true} = A^{-1}b$. However, this will lead to a wrong solution, because in fact A will usually be an $n \times n$ ill-conditioned and very large matrix. It will not only increase the difficulty to compute A^{-1} , but also influence the accuracy of computation. Additionally, in real applications, the error needs to be considered.

A filtering method can be utilized to compute an approximation of x , but before starting the approximation, let us talk about the singular value decomposition (SVD). The SVD of a matrix is its unitary reduction to diagonal form, which can be represented as

$$A = U\Sigma V^T = \sum_{i=1}^n u_i \sigma_i v_i^T \quad (1.2)$$

A is an $m \times n$ matrix, U and V are orthogonal matrices, and $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$.

The numbers σ_i (usually in descending order) are called the singular values of A , and the vectors u_i (columns of U) and v_i (columns of V) are referred to as the left and right singular vectors of A .

The SVD is an important tool for analyzing properties of numerical linear algebra problems. Here are some basic facts:

- A set of vectors $\{u_1, u_2, \dots, u_n\}$ is called orthonormal if:

$$u_i^T u_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

- An $n \times n$ matrix U is called orthogonal if $U^T U = I$; that is, the columns of U form an orthonormal set.
- It can be shown that

$$\|A\|_2 = \sigma_1 \quad (\text{i.e., the largest singular value})$$

and if A is nonsingular, then

$$\|A^{-1}\|_2 = \frac{1}{\sigma_n} \quad (\text{i.e., the reciprocal of the smallest singular value})$$

Thus, using the 2-norm, the condition number of the matrix A is:

$$k_2(A) = \frac{\sigma_1}{\sigma_n}$$

This is an important property: If the ratio of $\frac{\sigma_1}{\sigma_n}$ is large, then the matrix A is ill-conditioned. In the real-world problem, the matrix A is usually ill-conditioned.

By using the SVD of A , the solution can be written as,

$$x_{true} = A^{-1}b = V\Sigma^{-1}U^Tb \quad (1.3)$$

In the thesis, we assume the SVD of a matrix can be obtained and utilized in computation. Since image restoration problem is often ill-posed, which means small perturbations in the data result in large errors in the solution. The matrix, A , which is severely ill-conditioned, may have the following properties:

- The singular values, σ_i , decrease to and cluster at zero.
- There might be no gap to separate the large and small singular values.

One common way to stabilize the solution is to discard, or damp, using a regularization technique called filtering.

1.2 Filtering

In order to solve an ill-posed problem, a filtering or regularization method is needed. Regularization is a matter of finding out which erroneous SVD components to filter out and how to filter them out [1].

Our goal is to filter out the effects of the small σ_i , and this requires choosing a corresponding regularization parameter, α .

In the next Chapter, we discuss three filtering methods that will be considered in this thesis: Truncated Singular Value Decomposition (TSVD) and

Tikhonov regularization. Chapter 3 addresses generalized cross validation (GCV) as methods to determine good regularization parameters for the filter factors, as well as an optimal tolerance or parameter to do filtering, given the true image. Finally, Chapter 4 presents some numerical experiments and results, comparing filtering methods as well as parameter choice methods and will also test two methods in different cases. For example, error on b , error on A and errors on both. Chapter 5 provides conclusions.

Chapter 2

Regularization Methods

This chapter includes a description of two regularization methods: Truncated Singular Value Decomposition (TSVD) and Tikhonov regularization.

2.1 Truncated Singular Value Decomposition

Our goal is to avoid the magnification of errors in the problem due to small singular values, the most straightforward way to filter them out is to directly truncate those small singular values. This approach is called TSVD. As we talked before, we need to introduce a tolerance α equal to σ_k , where $k \leq n$, and all singular values less than the tolerance will get truncated. We defined

the TSVD filter factors to be

$$\phi_i = \begin{cases} 1 & i < k \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

and compute a TSVD filtered solution

$$\hat{x} = V \Sigma_F^\dagger U^T b, \quad (2.2)$$

\hat{x} is our approximation of x_{true} , $\Sigma_F = \text{diag}(\phi_i) \Sigma$ is the matrix of singular value after truncating. That is

$$\Sigma_F = \text{diag}(\sigma_1, \sigma_2 \cdots \sigma_k, 0 \cdots 0)$$

and

$$\Sigma_F^\dagger = \text{diag} \left(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \frac{1}{\sigma_3} \cdots \frac{1}{\sigma_k}, 0 \cdots 0 \right)$$

is the pseudo inverse of Σ_F .

We can see that those small singular values are truncated, which also causes the corresponding singular vectors to be truncated as well.

The main point is determining a good value for the tolerance deciding how many singular values to be truncated. Choosing a tolerance too big would result in dropping too many singular values, leading to a too smooth plot solution. On the contrary, choosing a tolerance too small allows too many small singular values to corrupt the solution, leading to a solution that

magnifies high frequency components of the noise. Methods to compute good values for the filtering tolerance will be briefly discussed in Chapter 3. More details and applications of TSVD can be found in [4].

2.2 Tikhonov Regularization

Tikhonov regularization method seeks to incorporate assumptions about the size and smoothness of the desired solution. Rather than solving the ill-conditioned least squares problem $\|b - Ax\|^2$, this method imposes a penalty for making the norm of the residual too large, and this in turn reduces the effect of small singular values [5]. The formula for Tikhonov is,

$$\min_x (\|b - Ax\|_2^2 + \alpha^2 \|x\|_2^2)$$

where α is the regularization parameter (filter) controlling the solution's degree of smoothness. After solving the least square equation by using the SVD of A , we can get Tikhonov filter factors $\phi_i = \frac{\sigma_i^2}{\sigma_i^2 + \alpha^2}$. Accordingly, the Tikhonov solution is,

$$\hat{x} = V \Sigma_F^{-1} U^T b \tag{2.3}$$

where $\Sigma_F = \text{diag}(\phi_i) \Sigma$, and

$$\Sigma_F^{-1} = \text{diag} \left(\frac{\sigma_1}{\sigma_1^2 + \alpha^2}, \frac{\sigma_2}{\sigma_2^2 + \alpha^2}, \dots, \frac{\sigma_n}{\sigma_n^2 + \alpha^2} \right)$$

When σ_k is much larger than α there will not be too much influence from filtering, but when σ_k decreases and becomes equal or smaller than α , the filtering term will take the weight in order to keep the solution smooth. Instead of truncating all small singular values, Tikhonov damps them, which makes the solution less corrupted by high frequency components of the noise, see Figure 2.1.

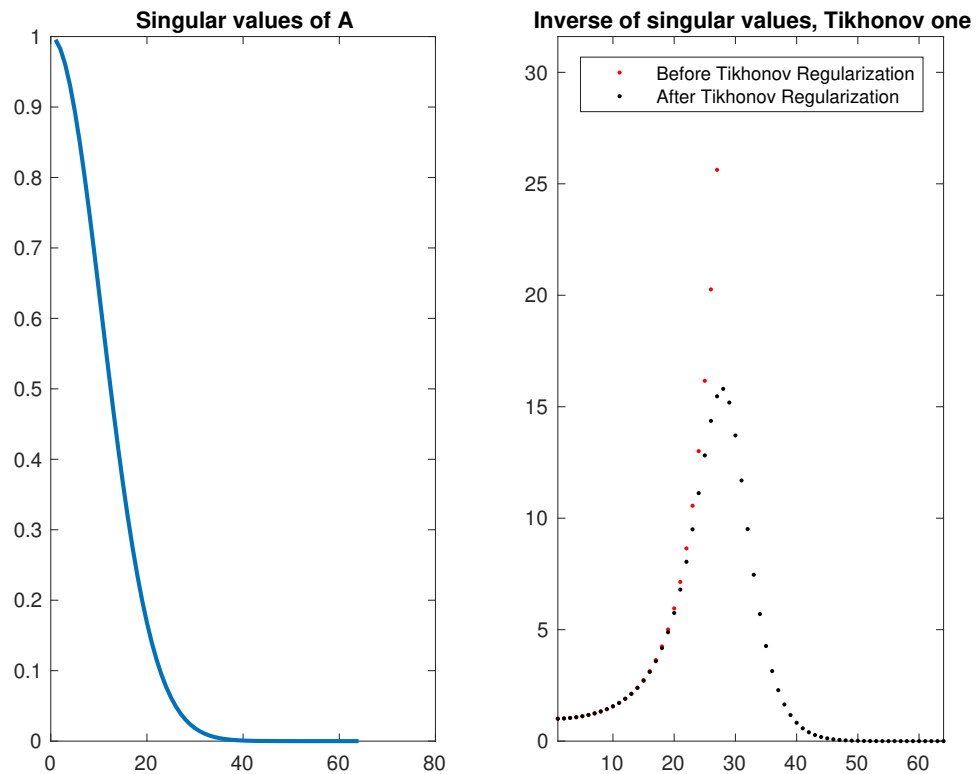


Figure 2.1: This figure shows the inverse of singular values before and after Tikhonov regularization in the test problem *spectra2* from Regularization Tools, see [2].

Similar to the TSVD case, choosing a tolerance too large would result in dropping too many singular values, leading to a too smooth solution. On the contrary, choosing a tolerance too small allows too many small singular values to corrupt the solution, leading to a too oscillating solution. Therefore, ways to choose an appropriate α value is important.

Recall that the filter factors for TSVD are:

$$\phi_i = \begin{cases} 1 & i < k \\ 0 & \text{otherwise} \end{cases}$$

Thus the TSVD corresponds to a sharp filter that simply cuts off the last $n - k$ components. In Tikhonov regularization, the filter factors are

$$\phi_i = \frac{\sigma_i^2}{\sigma_i^2 + \alpha^2}$$

correspond to a smooth filter that damps the components corresponding to $\sigma_i < \alpha$. When k is chosen such that $\sigma_k = \alpha$, the sharp filter TSVD can be seen as an approximation to the smooth filter of Tikhonov regularization. Thus, it seems possible that the regularized solutions in each scheme may be similar.

Moreover, in the Chapter 4 we will also see some experiments to show the performance comparison between TSVD and Tikhonov Regularization.

Chapter 3

Choosing Regularization Parameters

A good filtering parameter for the regularization is always important when solving inverse ill-posed problems.

If we do not choose any parameter and compute the inverse solution $x = A^{-1}b$, then we get a result that is corrupted by high frequency components of the noise. This is illustrated in the plot on the right in Figure 3.1.

There are two classes of parameter choice methods [1]:

- Those based on a priori knowledge and assumptions about the error norm. These requires a good estimate of the error, $\|e\|_2$
- Methods that do not require the error norm, but instead seek to extract this information from the given data.

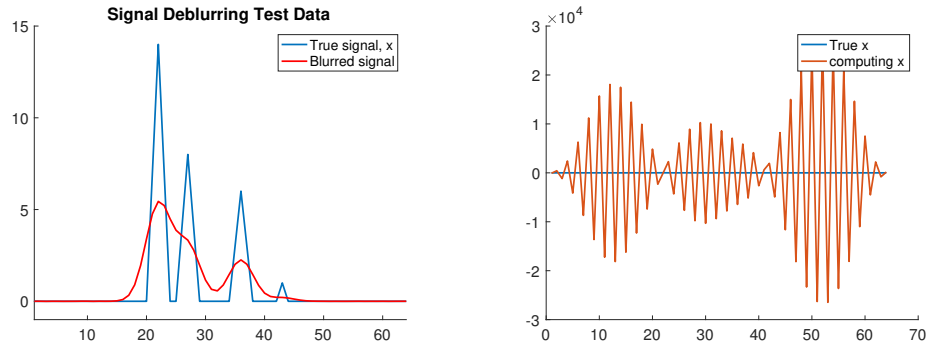


Figure 3.1: This figure shows the computed result for the test problem.

When we know about the error norm, we can simply use it in the process of deciding the parameter for either TSVD or GCV and directly solve it, for the example above we can get the approximation shown in Figure 3.2

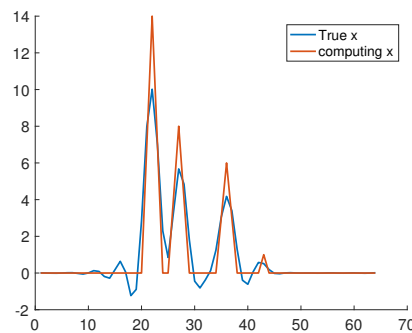


Figure 3.2: This figure shows the computed result for the test problem.

However, in this chapter we will focus on situation we have no priori knowledge and assumptions about the error norm. Generalized cross validation (GCV) will be mainly discussed in this chapter, which requires no information about the real error norm, $\|e\|_2$.

In addition, this chapter addresses the computation of an optimal regularization parameter, when the true solution is known.

GCV is a widely used and very successful predictive method for choosing the smoothing parameter. The basic idea is that, if one datum point is dropped, then a good value of the regularization parameter should predict the missing datum value fairly well [1].

Our goal is to minimize the absolute value sum of error between the original points, b_i , and the computed points, $A\hat{x}$. The GCV function is,

$$G = \frac{n\|A\hat{x} - b\|_2^2}{\text{trace}(I - AA_F^{-1})} \quad (3.1)$$

where the trace of a matrix is the sum of its diagonal entries. The goal here is to minimize the GCV function in order to control the error.

3.1 Generalized cross validation for TSVD

Before simplifying the GCV function for the specific cases of TSVD and Tikhonov regularization, we need the following formula:

$$\text{trace}(I - AA_F^{-1}) = \text{trace}(I - \Sigma\Sigma_F^{-1})$$

which can make the GCV equation transform to

$$G = \frac{n\|A\hat{x} - b\|_2^2}{\text{trace}(I - \Sigma\Sigma_F^{-1})} \quad (3.2)$$

When using the GCV method with TSVD, the function to minimize (3.1) can be simplified to:

$$G(k) = \frac{n \sum_{i=k+1}^n \hat{b}_i^2}{(n-k)^2} \quad (3.3)$$

where $\hat{b} = U^T b$, and by using the property:

$$\text{trace}(I - \Sigma \Sigma_F^{-1}) = n - k$$

By computing $G(k)$ for all $k = 1, 2, \dots, n$ and finding the minimum value of G , a value for the truncation tolerance can be computed.

Note: in the case of equation (3.3), instead of computing exact value of tolerance, we choose to compute out the index of truncation (after which index all singular values will be tuncated).

3.2 Generalized cross validation for Tikhonov

As we talked above, the GCV function when used with Tikhonov regularization can also be simplified to:

$$G(\alpha) = \frac{n \sum_{i=1}^n \left(\frac{\hat{b}_i}{\sigma_i^2 + \alpha^2} \right)^2}{\left(\sum_{i=1}^n \frac{1}{\sigma_i^2 + \alpha^2} \right)^2} \quad (3.4)$$

where $\hat{b} = U^T b$, α is the filtering parameter and using the following property:

$$\text{trace}(I - \Sigma \Sigma_F^{-1}) = \sum_{i=1}^n \frac{1}{\sigma_i^2 + \alpha^2}$$

Because $G(\alpha)$ here is a continuous function, as opposed to the discrete case in TSVD, we use Matlab's `fminbnd` function, which is based on a combination of Golden Section Search and Quadratic Interpolation Search, to find the value of α at which $G(\alpha)$ is minimized.

The disadvantage of GCV: Oftentimes the GCV function is very flat near the minimum, like Figure 3.3. It is sometimes difficult to find the minimum in such situations, and thus it can be difficult to determine a good regularization parameter.

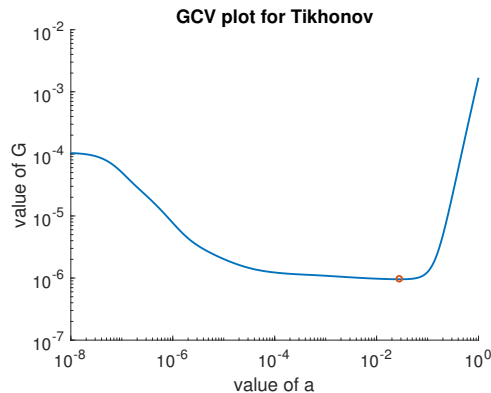


Figure 3.3: This figure shows the computed GCV plot for the test problem *phillips*.

Due to the flatness of the GCV function, using Matlab's `fminbnd` function to find the minimum can be very time consuming.

3.3 Optimal Regularization Parameter

Computing a guaranteed optimal regularization parameter works only when the true solution is given. Therefore, in real-life situations when the true solution is not known, this method cannot be used. However, the purpose of introducing this method is to provide a basis for comparison for the other parameter choice methods in tests and experiments.

In order to find the optimal regularization parameter, we should minimize the error between computed x and x_{true} , which can be displayed as,

$$\min_x \|\hat{x} - x_{true}\|_2^2$$

We observe that V is an orthogonal matrix, and from equation (2.3), $\hat{x} = V\Sigma_F^{-1}U^Tb$. Then we use the fact that the 2-norm is invariant under orthogonal transformation, and we can get:

$$\|\hat{x} - x_{true}\|_2^2 = \|V^T\hat{x} - V^Tx_{true}\|_2^2 = \|\Sigma_F^{-1}U^Tb - V^Tx_{true}\|_2^2 \quad (3.5)$$

To minimize this value for each filtering method, we can find the optimal regularization parameter.

3.4 Optimal Regularization for TSVD

Recall that for the TSVD case, $\Sigma_F^\dagger = \text{diag}(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \frac{1}{\sigma_3}, \dots, \frac{1}{\sigma_k}, 0, \dots, 0)$ where $k \leq n$, which means the singular values with indexes larger than k all get truncated.

After knowing this, we start our minimization of equation (3.5),

$$\begin{aligned} \|\Sigma_F^{-1}U^Tb - V^T x_{true}\|_2^2 &= \sum_{i=1}^k \left(\frac{(U^Tb)_i}{\sigma_i} - (V^T x_{true})_i \right)^2 + \sum_{i=k+1}^n (V^T x_{true})_i^2 \\ &= \sum_{i=1}^k y_i^2 + \sum_{i=k+1}^n X_{truei}^2 \end{aligned}$$

where $y_i = \frac{(U^Tb)_i}{\sigma_i}$ and $X_{truei} = (V^T x_{true})_i$.

Let e_k be the error corresponding to truncating after k singular values.

We can get,

$$e_k = e_{k-1} - X_{truek}^2 + y_k^2$$

To find the optimal tolerance, we must compute e_k for $k = 0, 1, 2, \dots, n$ and find the value of k at which e_k is smallest. This will give the truncation index, and the optimal choice for the tolerance will be equal to σ_k . One thing we need to pay attention here is all computation is done in the form of index, which means we get the index of truncation, not exact tolerance.

3.5 Optimal Regularization for Tikhonov

Similarly, we can use the way above to find the optimal regularization parameter for Tikhonov filtering. As we prove before the singular values for Tikhonov regularization is,

$$\Sigma_F^{-1} = \text{diag} \left(\frac{\sigma_1}{\sigma_1^2 + \alpha^2}, \frac{\sigma_2}{\sigma_2^2 + \alpha^2}, \dots, \frac{\sigma_n}{\sigma_n^2 + \alpha^2} \right)$$

By using this in the minimization of equation (3.5), we can get,

$$\|\Sigma_F^{-1}U^Tb - V^T x_{true}\|_2^2 = \sum_{i=1}^n \left(\frac{\sigma_i(U^Tb)_i}{\sigma_i^2 + \alpha^2} - (V^T x_{true})_i \right)^2$$

Our goal is to find the α that will minimize this error. Similar to GCV, we can use Matlab's build-in `fminbnd` function to find the optimal regularization parameter.

In this case, one specific tolerance value can be found. An important note for finding optimal parameters is that the function to be minimized may have multiple minimums or local minimums if the graph is not unimodal. In these situations, we take the global minimum as the optimal regularization parameter. By comparing solutions that use the optimal parameter with those that use the GCV chosen parameter, we can better address the performance of GCV.

Chapter 4

Numerical Experiments

This chapter presents a culmination of all the numerical tests and experiments we performed, including in both small-scale matrix test problem and a large-scale real world application. Each section provides a brief summary of the goals of the experiment, a presentation of the results and observations or comments concerning the analysis of the results. We provide a numerical comparison of the two filtering methods (Tikhonov and TSVD) in different cases of error. A comparison of parameter choice methods for each filtering method will also be conducted. Lastly, we will show the performance of GCV for Tikhonov on a real-world problem.

Some important things need to be addressed before we present our experiments:

- We assume all SVD decompositions can be computed.
- For all test problems, the noise level will be set to be 0.001, which

means a 0.1% of random noise will be on the matrices.

- Singular values will be in the interval of $[0,1]$.
- For the small scale problems, all matrices will be of size 256×256 .
- We assume we have no idea about the noise level and use parameter choice methods to find the filtering parameter.

4.1 Test problem experiments

We start by doing some small-scale test problems to warmup, including *deriv2*, *phillips*, *shaw* and *heat* in MATLAB's *regulationtools* package. The basic idea is to solve the equation we talked in the beginning of thesis, (1.2).

$$b = Ax + e$$

In the experiments below, three cases, error in only b , error in only A and errors in both of them, will be tested in all of four problems.

4.1.1 With error in only b

In this section we consider the test problem with error only in b . The first test problem is *deriv2*, where the results are shown in Figure 4.1.

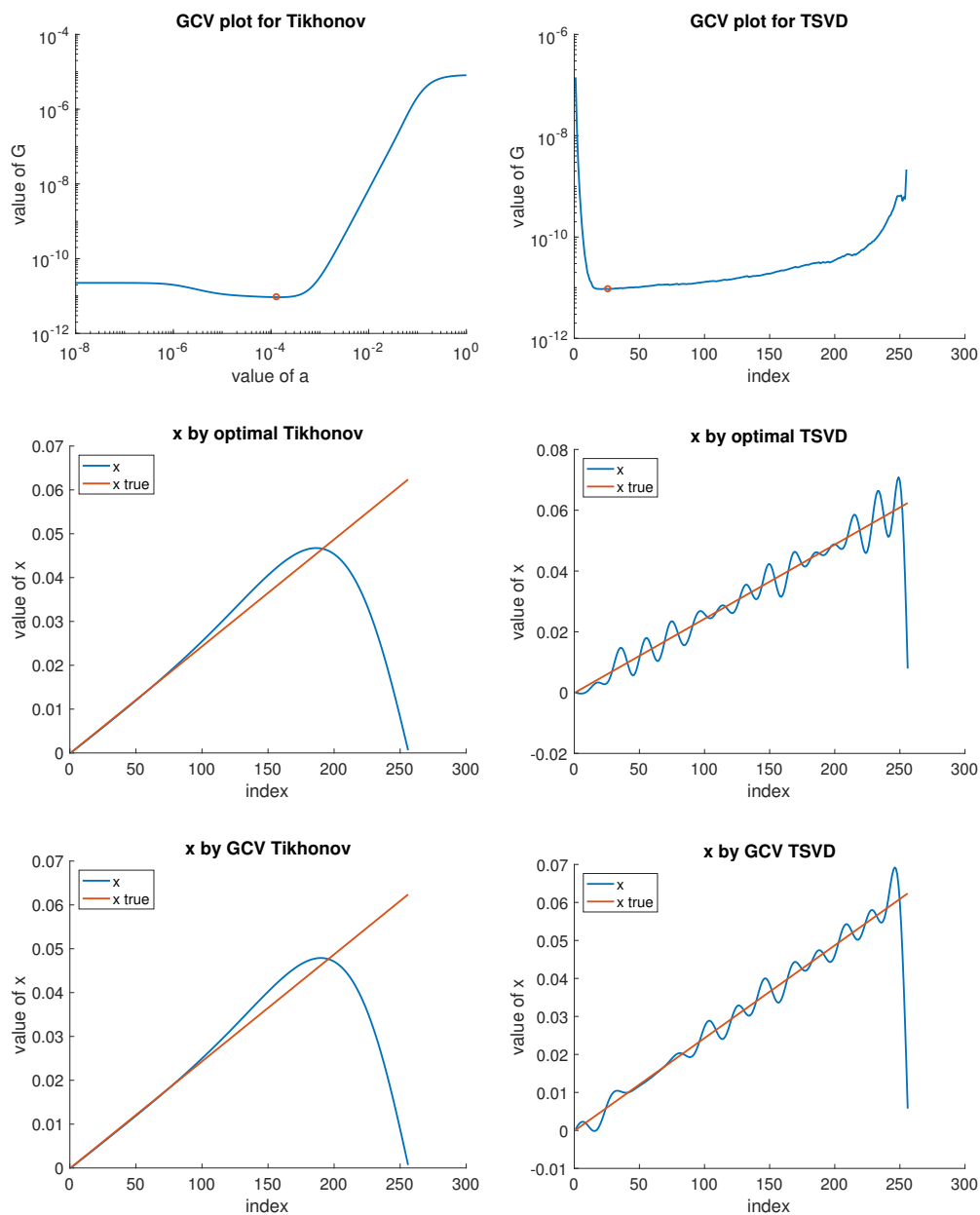


Figure 4.1: This figure shows the GCV curves (top row) and the the computed reconstructions (rows 2 and 3) for the test problem `deriv2`.

For the solutions shown in the bottom row of Figure 4.1, the solution was computed using the GCV chosen regularization parameter. The solutions in the middle row were computed using a regularization parameter that minimizes the relative error between the computed solution and the true solution. We refer to this as the “optimal” solution. The left shows results corresponding to Tikhonov regularization, and the right shows results for TSVD.

Similar plots are shown for the rest of three test problems in Figures 4.2 through 4.4.

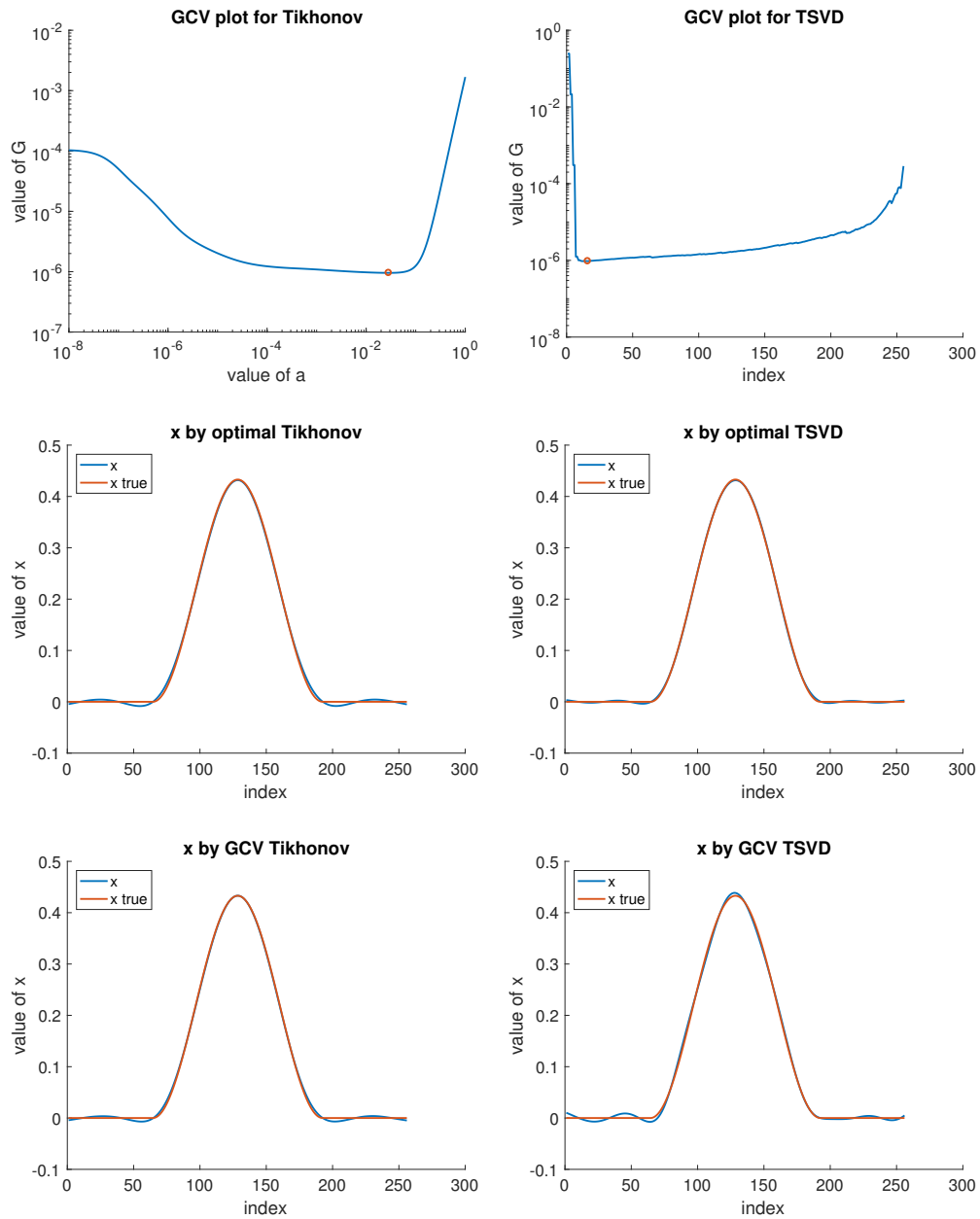


Figure 4.2: This figure shows the GCV curves (top row) and the the computed reconstructions (rows 2 and 3) for the test problem phillips.

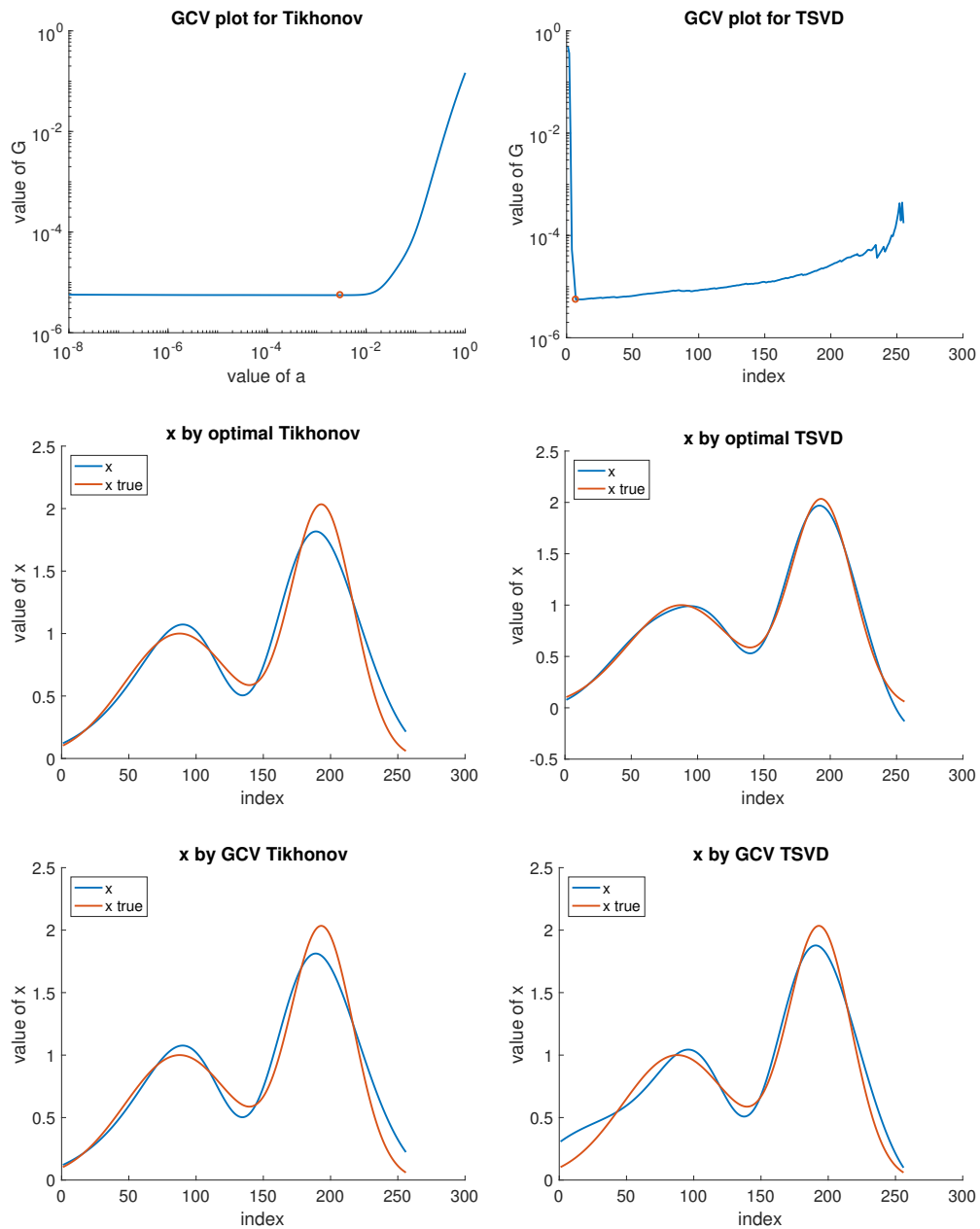


Figure 4.3: This figure shows the GCV curves (top row) and the the computed reconstructions (rows 2 and 3) for the test problem `shaw`.

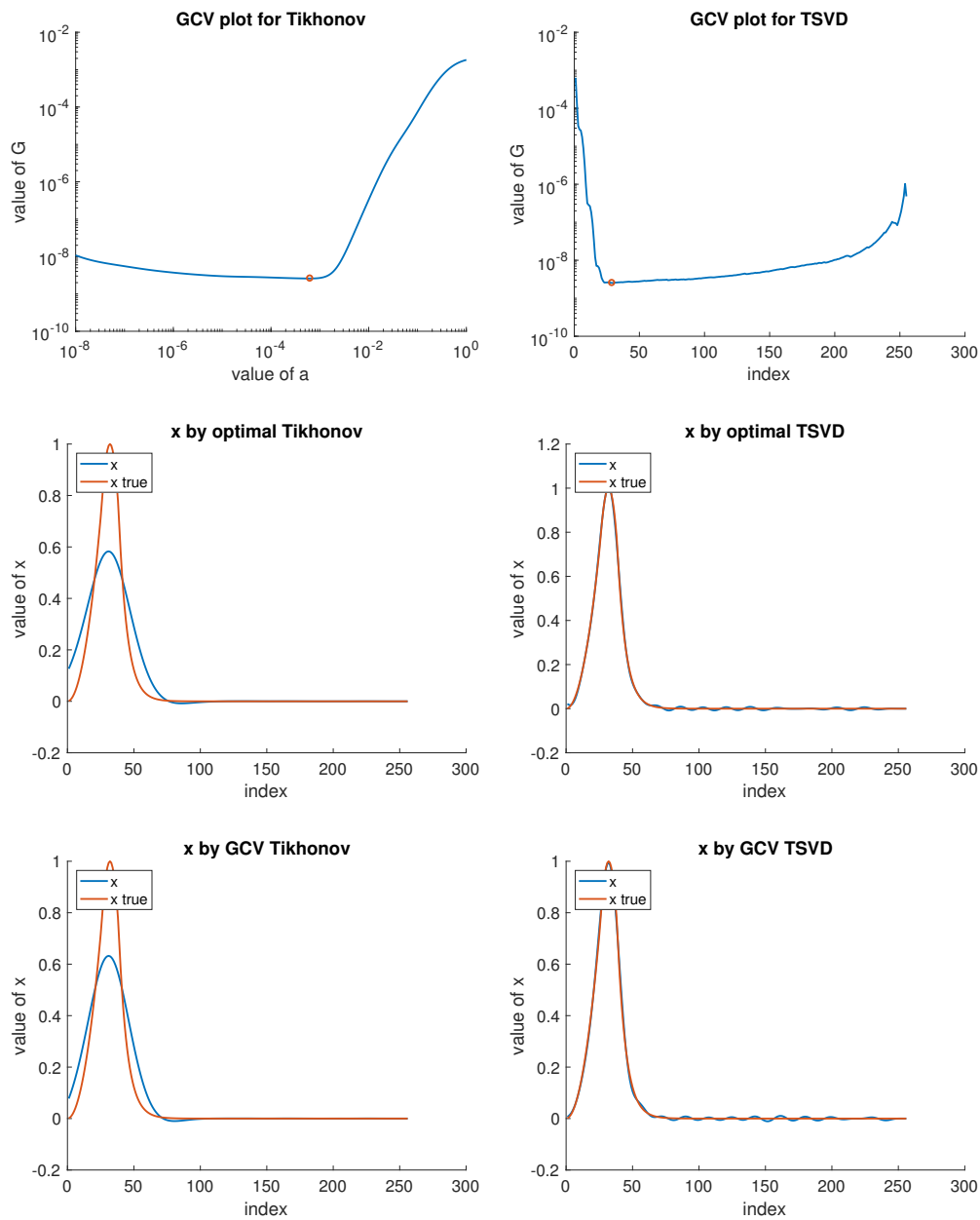


Figure 4.4: This figure shows the GCV curves (top row) and the the computed reconstructions (rows 2 and 3) for the test problem `heat`.

We can see in the case of error in only b , the performance of TSVD is pretty good and sometimes better than the Tikhonov.

4.1.2 With error in only A

In this section we consider the test problem with error only in A . The first test problem is `deriv2`, where the results are shown in Figure 4.5. We choose the regularization parameter by GCV.

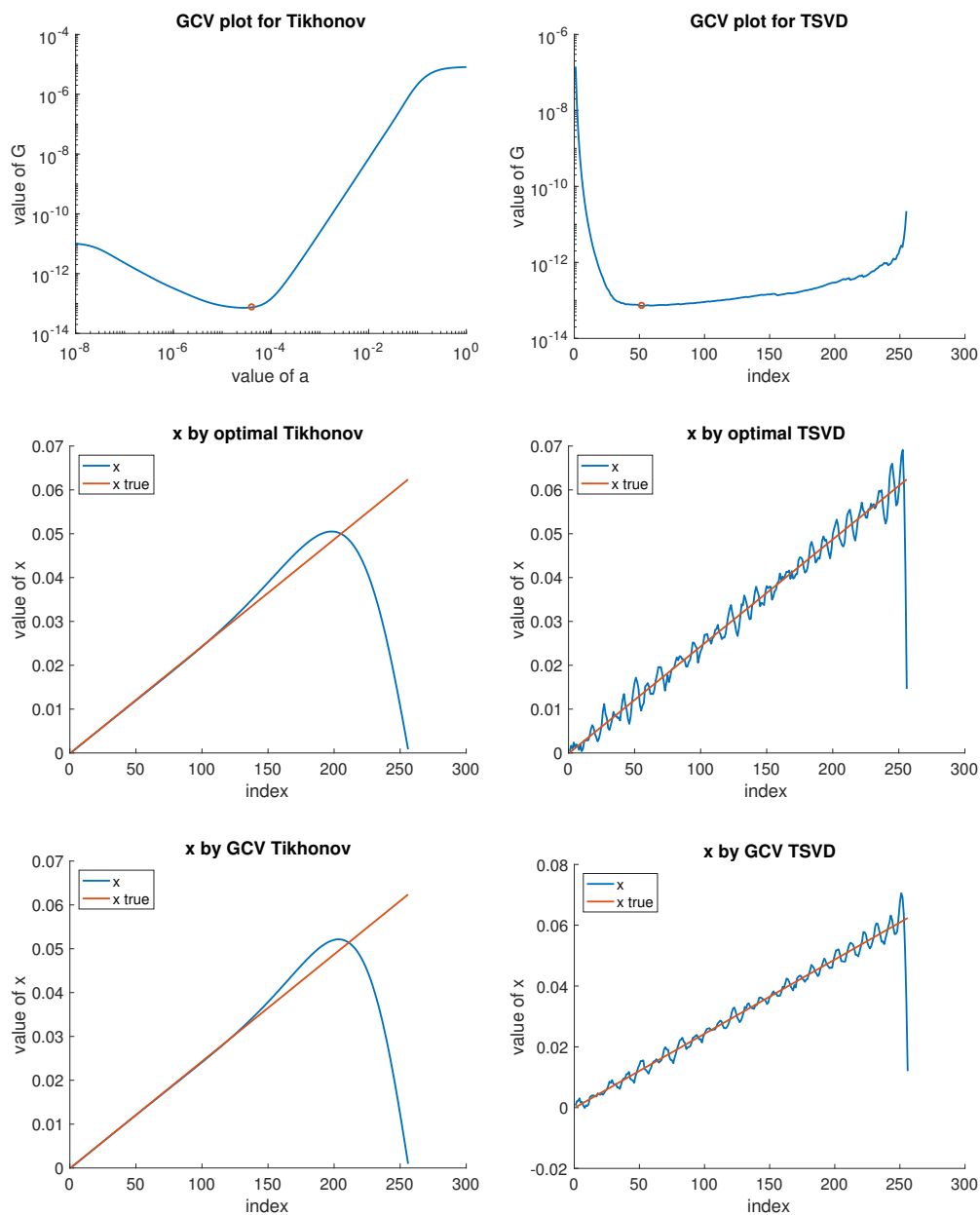


Figure 4.5: This figure shows the GCV curves (top row) and the the computed reconstructions (rows 2 and 3) for the test problem `deriv2`.

For the solutions shown in the bottom row of Figure 4.5, the solution was computed using the GCV chosen regularization parameter. The solutions in the middle row were computed using a regularization parameter that minimizes the relative error between the computed solution and the true solution. We refer to this as the “optimal” solution. The left shows results corresponding to Tikhonov regularization, and the right shows results for TSVD.

Similar plots are shown for the rest of three test problems in Figures 4.6 through 4.8.

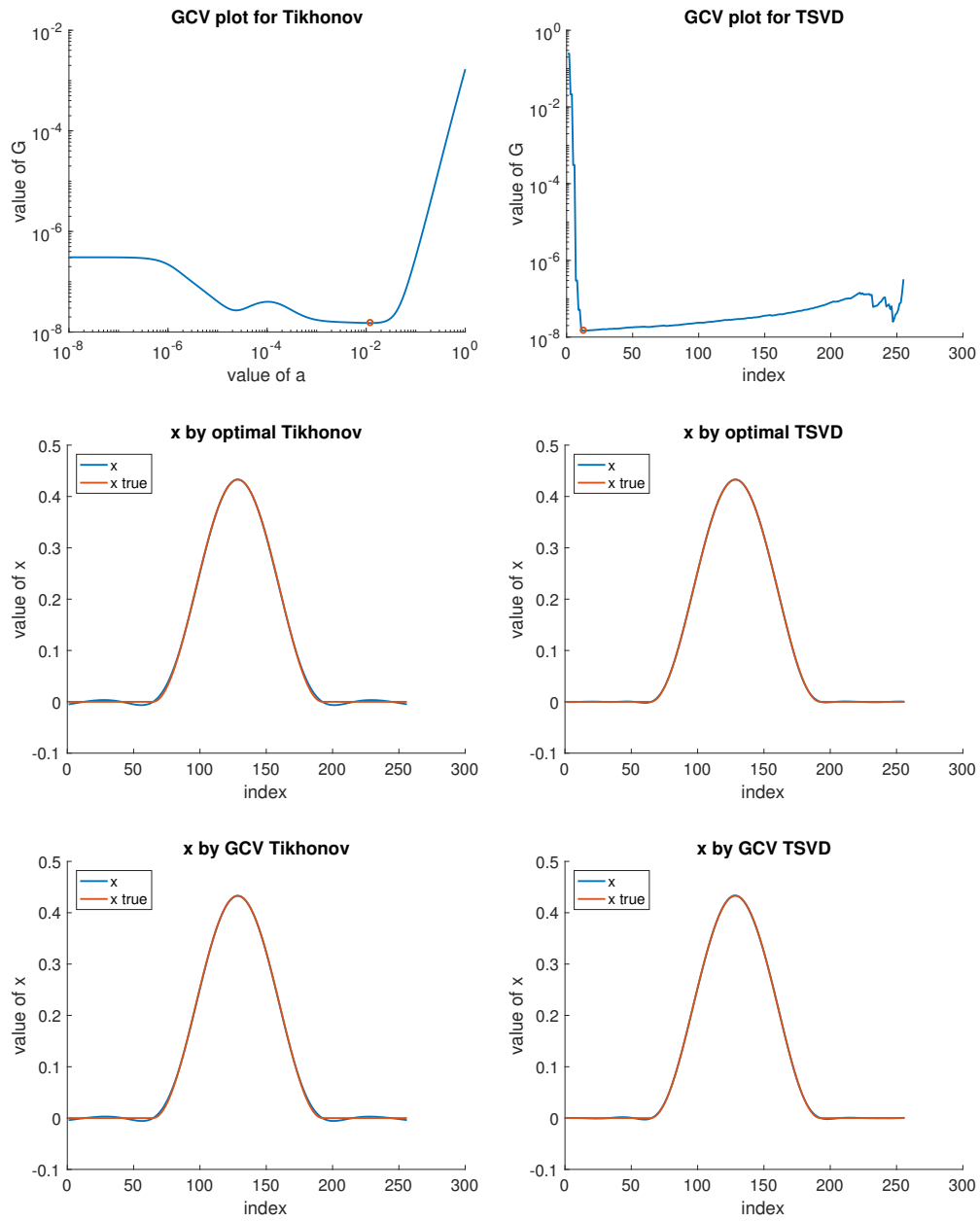


Figure 4.6: This figure shows the GCV curves (top row) and the the computed reconstructions (rows 2 and 3) for the test problem phillips.

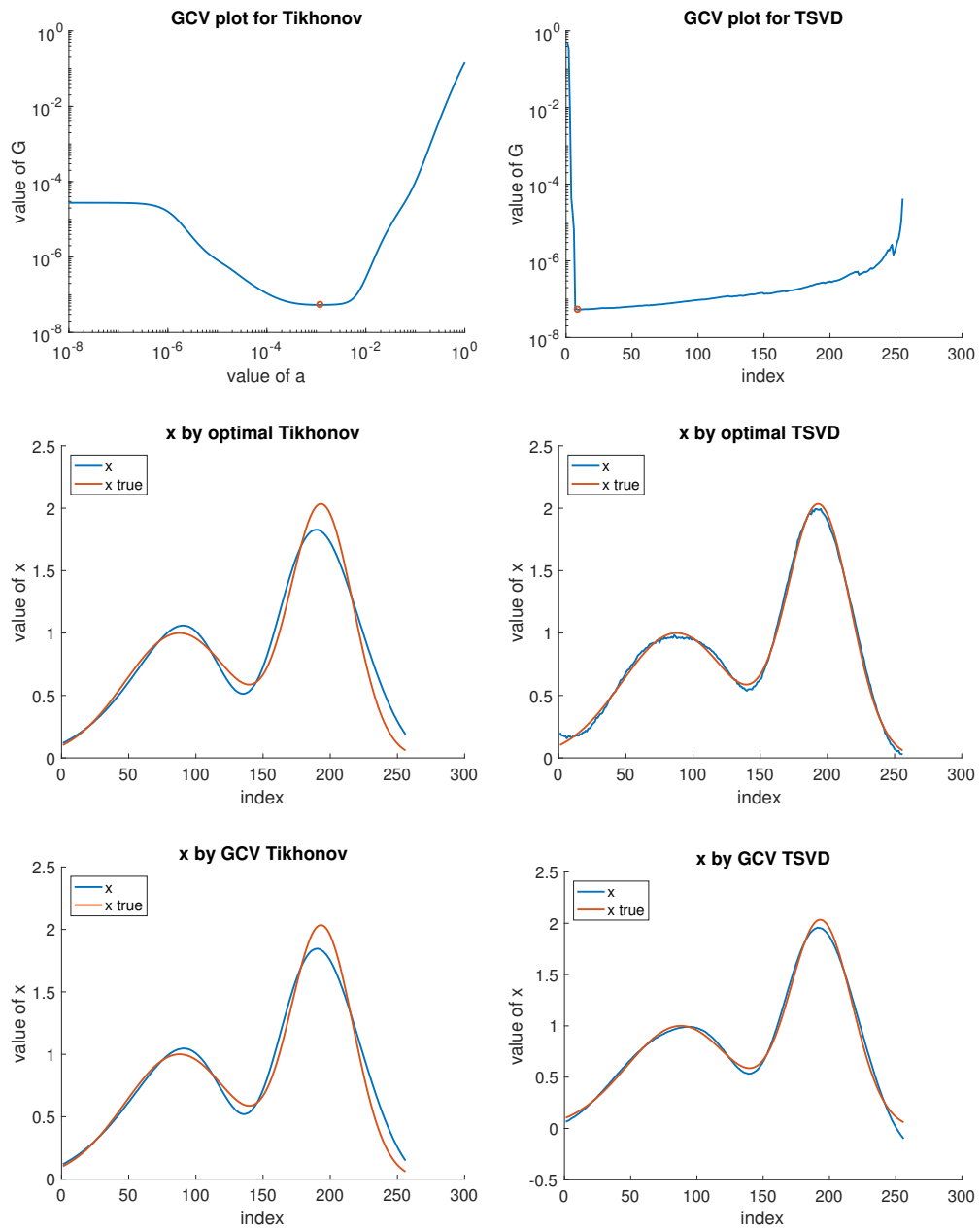


Figure 4.7: This figure shows the GCV curves (top row) and the the computed reconstructions (rows 2 and 3) for the test problem `shaw`.

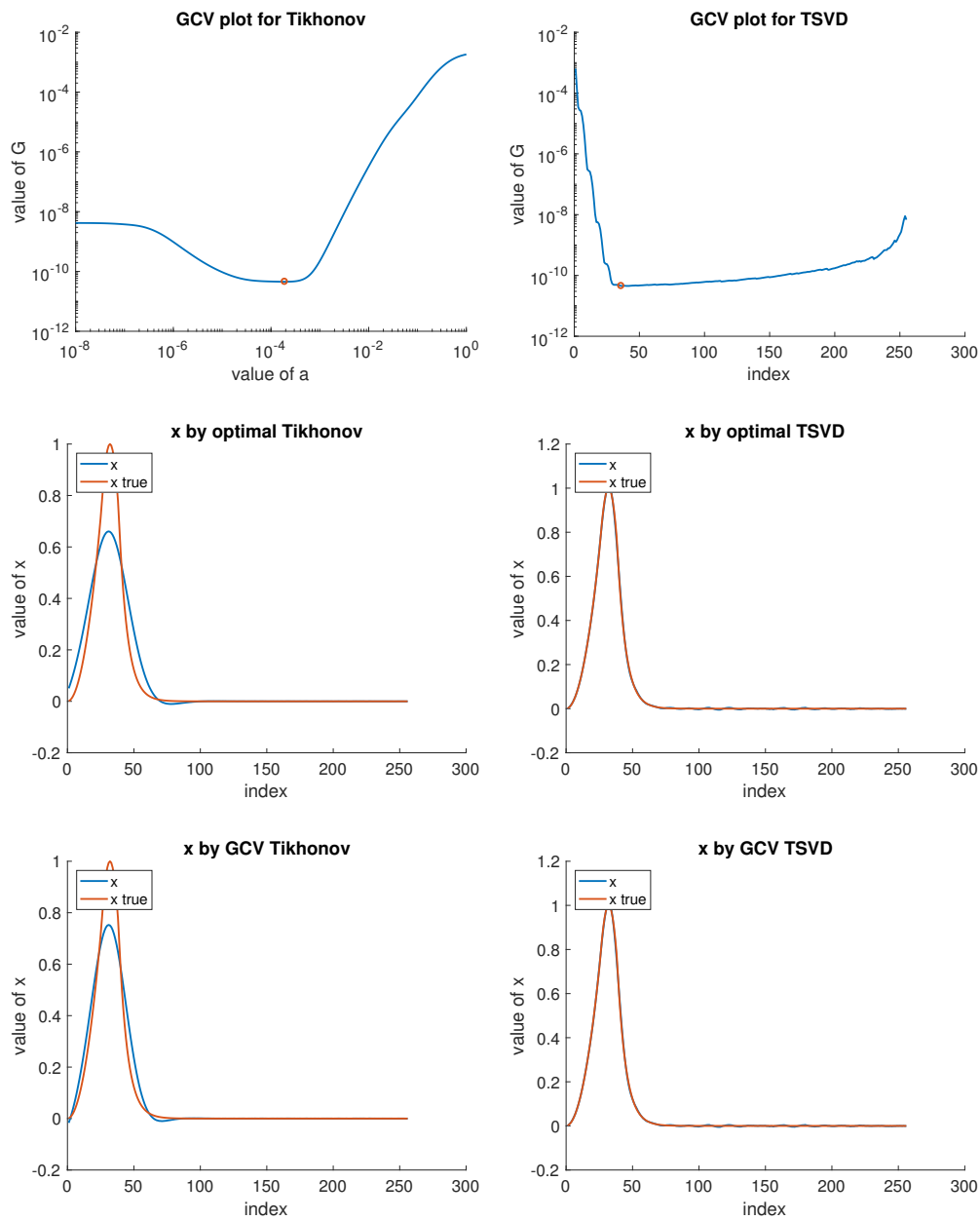


Figure 4.8: This figure shows the GCV curves (top row) and the the computed reconstructions (rows 2 and 3) for the test problem `heat`.

4.1.3 With errors in both of A and b

In this section we consider the test problem with errors both in b and A . The first test problem is `deriv2`, where the results are shown in Figure 4.5. We choose the regularization parameter by GCV.

For the solutions shown in the bottom row of Figure 4.9, the solution was computed using the GCV chosen regularization parameter. The solutions in the middle row were computed using a regularization parameter that minimizes the relative error between the computed solution and the true solution. We refer to this as the “optimal” solution. The left shows results corresponding to Tikhonov regularization, and the right shows results for TSVD.

Similar plots are shown for the rest of three test problems in Figures 4.10 through 4.12.

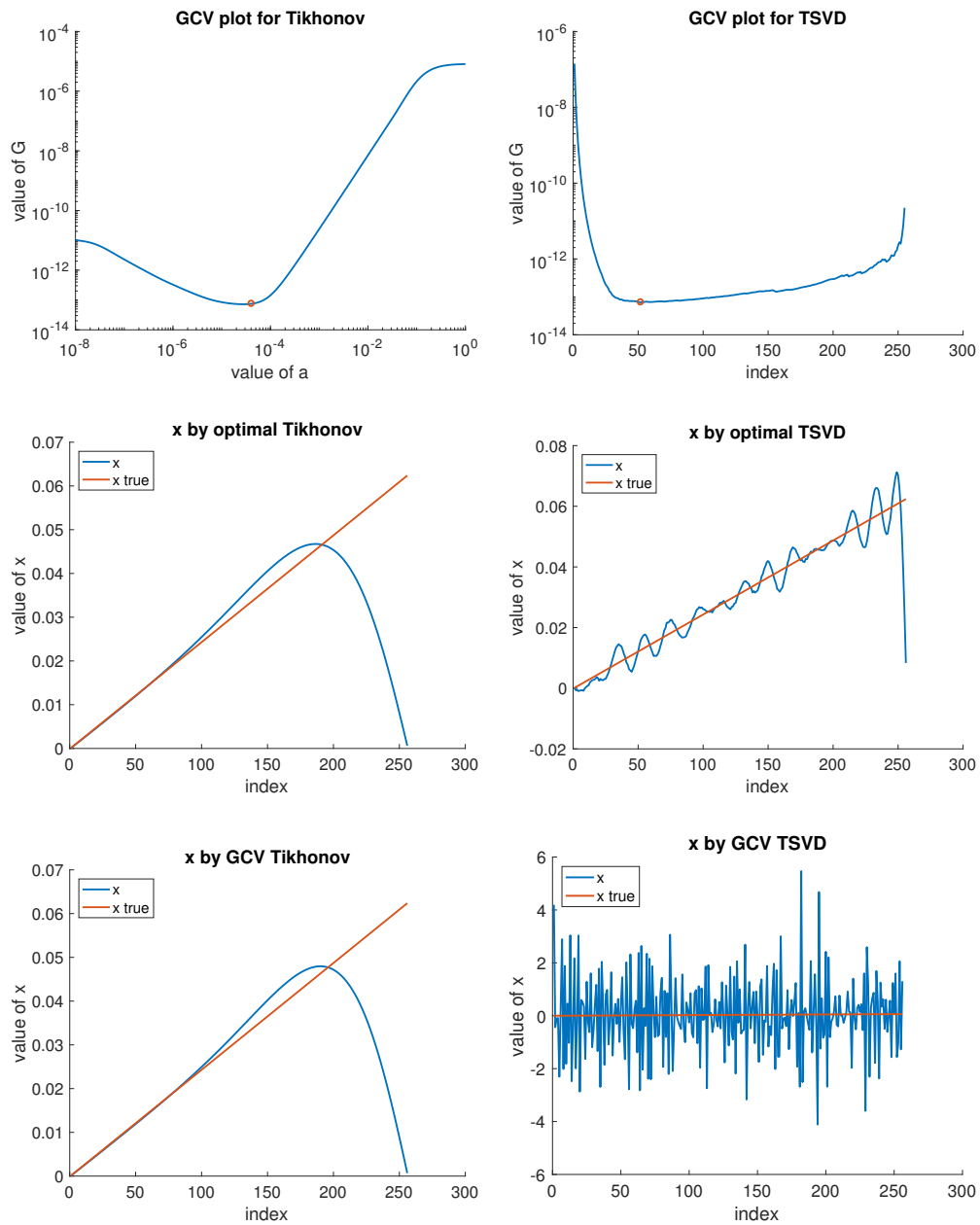


Figure 4.9: This figure shows the GCV curves (top row) and the the computed reconstructions (rows 2 and 3) for the test problem `deriv2`.

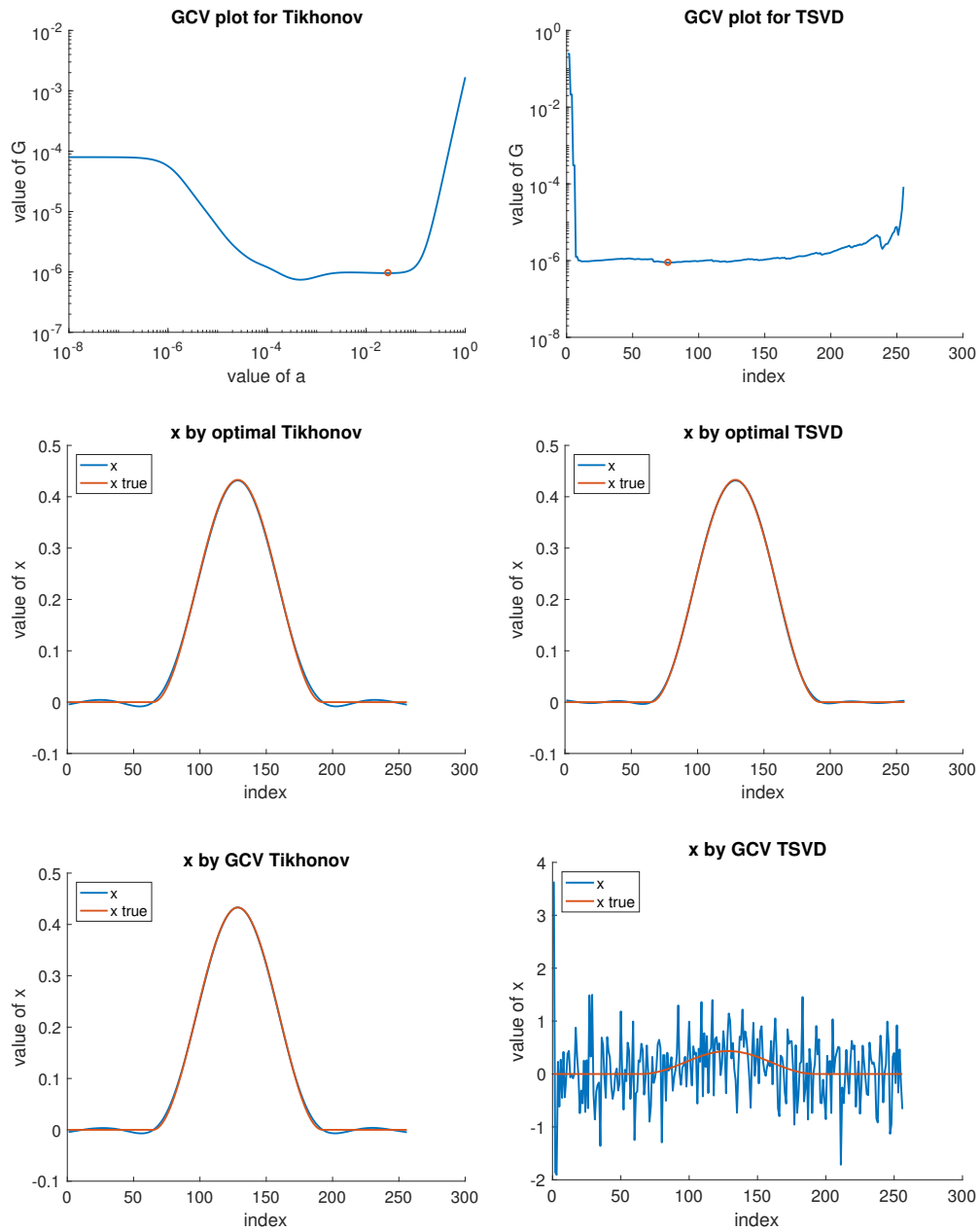


Figure 4.10: This figure shows the GCV curves (top row) and the the computed reconstructions (rows 2 and 3) for the test problem phillips.

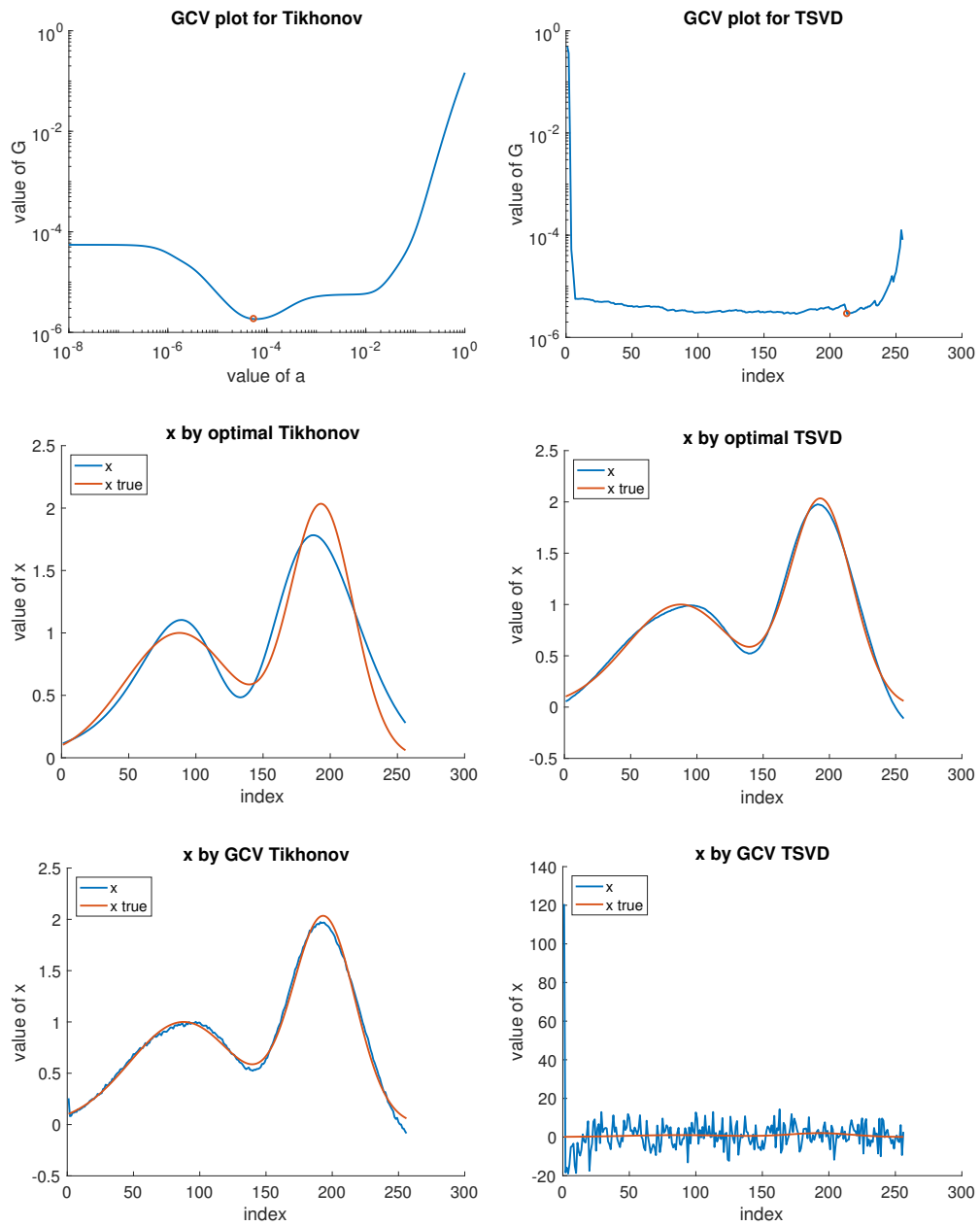


Figure 4.11: This figure shows the GCV curves (top row) and the the computed reconstructions (rows 2 and 3) for the test problem `shaw`.

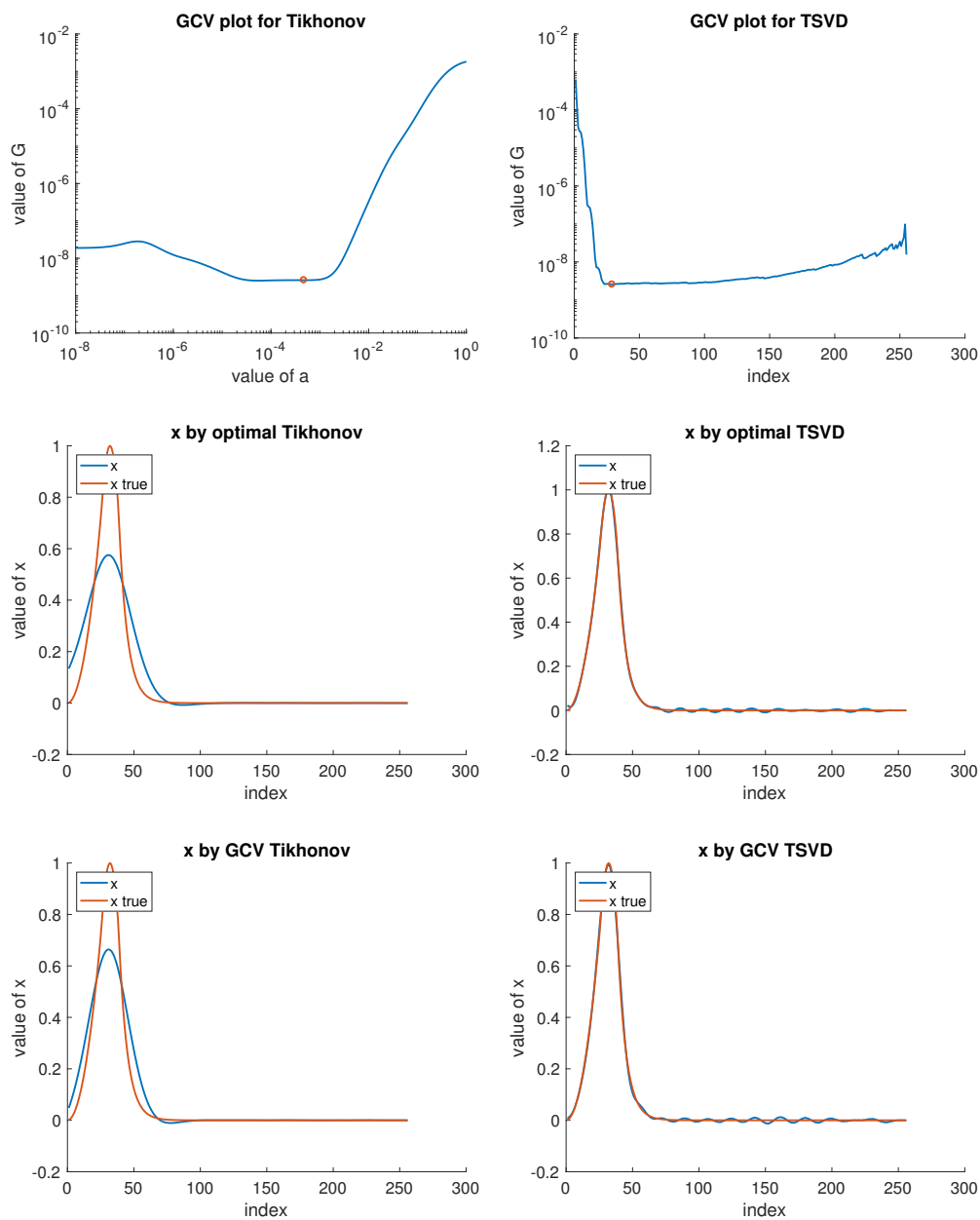


Figure 4.12: This figure shows the computed reconstructions for the test problem heat.

According to the results above, we can see that when there are errors in both of A and b , the TSVD function may not work well for all cases.

4.2 Application to Image Deblurring

After doing some small-scale matrix experiments, we'd like to see our GCV Tikhonov regularization's performance in real-world problem solving. In particular, we consider a problem in image deblurring, where b is an observed blurred image, A is a matrix that models the blurring operation, and x is the true image we want to recover. In this specific example, we use the satellite data [2], assuming zero boundary conditions, and we use different noise level to test. The basic idea is to transform the 2-dimensional images into vectors, and solve the equation,

$$b = Ax + e$$

where b is the observed blurred images, A is the blurring matrix and e is the random error. In this case, we know rarely about the noise level. Our goal is to compute an approximation of x , is the true (but unknown) image.

In the applications like this the matrix A is usually very ill-conditioned, so we need some regularization. We will use Tikhonov regularization. The other issue is that A is usually very large, and generally it is computationally infeasible (much too expensive) to compute an SVD of the matrix A . How-

ever, in certain situations we can compute approximations of A for which we can compute an SVD. Specifically, in image deblurring the matrix A can be written as

$$A = B_1 \otimes C_1 + B_2 \otimes C_2 + \cdots B_m \otimes C_m$$

where \otimes denotes Kronecker product. If A is an $m \times n$ matrix and B is a $p \times q$ matrix, then the Kronecker product $A \otimes B$ is the $mp \times nq$ block matrix [10]:

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}$$

It has been shown that in many cases that the first term in this summation, $B_1 \otimes C_1$ is a good approximation of A . Moreover, if we have a single Kronecker product like this, the SVD can be computed efficiently. Specifically, if we compute the SVDs of the smaller matrices B_1 and C_1 ,

$$B_1 = U_b \Sigma_b V_b^T \quad \text{and} \quad C_1 = U_c \Sigma_c V_c^T$$

then the SVD of $B_1 \otimes C_1$ is given by:

$$B_1 \otimes C_1 = (U_b \otimes U_c)(\Sigma_b \otimes \Sigma_c)(V_b \otimes V_c)^T.$$

By exploiting properties of Kronecker products we do not need to explicitly form the large matrices to do computations with them. For example, matrix-

vector multiplication with $U_b \otimes U_c$ can be done as:

$$\mathbf{y} = (U_b \otimes U_c)\mathbf{x} \quad \Leftrightarrow \quad Y = U_c X U_b$$

where $\mathbf{y} = \text{vec}(Y)$ and $\mathbf{x} = \text{vec}(X)$, and $\text{vec}(X)$ reshapes a matrix with columns $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ into a long vectors with the columns stacked one on top of the other. That is,

$$\text{vec}(X) = \text{vec}([\mathbf{x}_1 \ \mathbf{x}_2 \ \cdots \ \mathbf{x}_n]) = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}$$

Thus, with proper reshaping of the vector \mathbf{x} , the large matrix-vector multiplication can be done with small matrix-matrix multiplications.

In Tikhonov regularization, we try to solve the damped least squares problems

$$\min_{\mathbf{x}} \{ \|A\mathbf{x} - \mathbf{b}\|_2^2 + \alpha^2 \|\mathbf{x}\|_2^2 \}. \quad (4.1)$$

As discussed in previous chapters of this thesis, the regularization parameter α controls “smoothness” of the regularized solution. We will generally assume that a good value of λ is known, or can be estimated from other techniques.

The problem (4.1) is equivalent to solving the overdetermined least squares

(LS) problem

$$\min_{\mathbf{x}} \left\| \begin{bmatrix} A \\ \alpha I \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2. \quad (4.2)$$

We now consider the LS problem,

$$\min_{\mathbf{x}} \left\| \begin{bmatrix} A \\ \alpha I \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2.$$

To solve this problem, we will use the iterative method LSQR, which is usually recommended for large scale least squares problems where it is computationally expensive to compute a full decomposition of the given matrix, but where it is possible to efficiently compute matrix-vector multiplications with the given matrix.

Before we can solve this, we need a good estimate of the regularization parameter. To do this, we first use the approximation

$$A \approx B_1 \otimes C_1$$

and use the SVD of this approximation as input to the GCV function to compute an estimate of α . The results of our previous numerical experiments on small scale matrices leads us to believe that parameters computed using good matrix approximations should provide good values. We want to verify this with some large scale problems.

The large scale problems we use are available in the *RestoreTools* package,

<http://www.mathcs.emory.edu/~nagy/RestoreTools/index.html> and some introduction is refer to [11].

4.2.1 AtmosphericBlur10



Figure 4.13: This figures show the graph we observe and the true graph before being blurred

By using GCV function we can find α and then use LSQR to solve the Tikhonv regularized least squares problem.

And we also plot out the relative residual graph, where a larger number of iteration leads to a smaller relative residual.

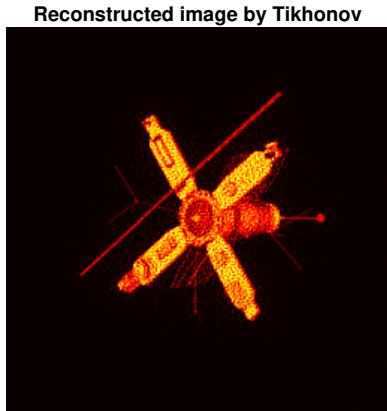


Figure 4.14: This figure shows the computed reconstructions for the satellite problem by using GCV regularization parameter.

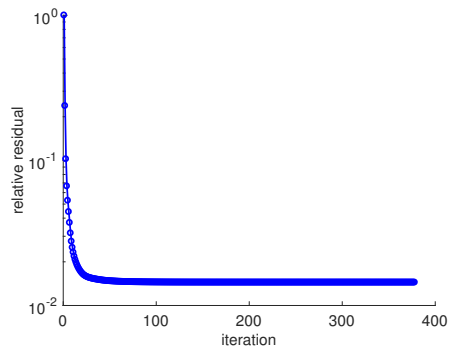


Figure 4.15: This figure shows the level of relative residual

And then we will use optimal regularization parameter to show the best solution we can get.

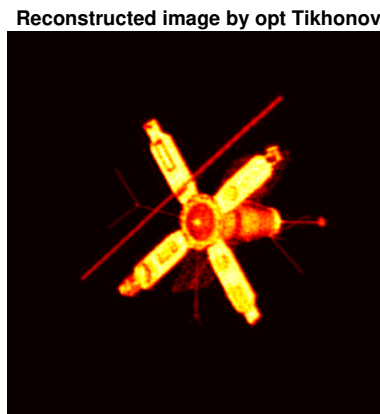


Figure 4.16: Best possible solutions using optimal regularization parameter.

We can obviously see that for the AtmosphericBlur10, parameters computed using good matrix approximations provide good values.

4.2.2 AtmosphericBlur30

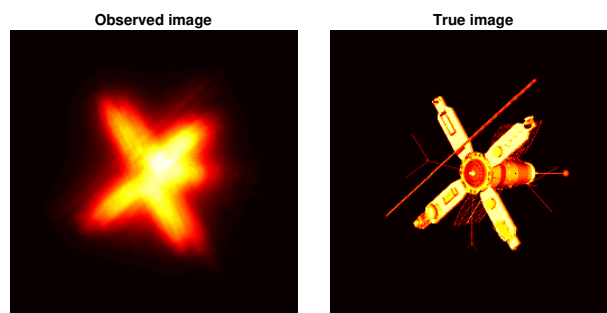


Figure 4.17: This figures show the graph we observe and the true graph before being blurred

By using GCV function we can find α and then use LSQR to solve the Tikhonv regularized least squares problem.

And we also plot out the relative residual graph, where a larger number of iteration leads to a smaller relative residual.

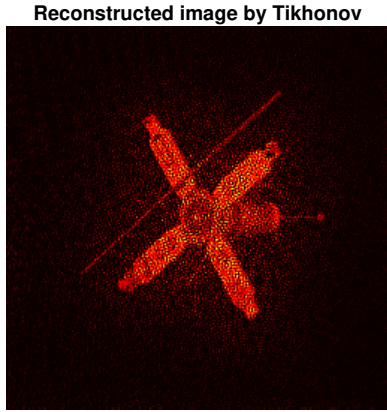


Figure 4.18: This figure shows the computed reconstructions for the satellite problem by using GCV regularization parameter.

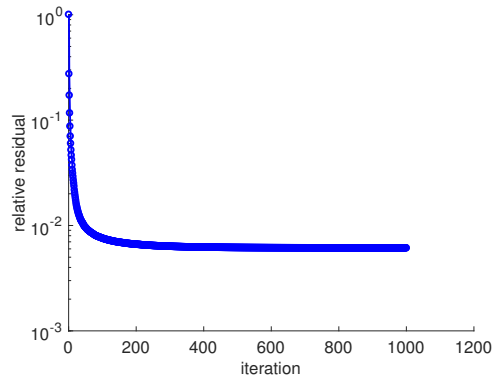


Figure 4.19: This figure shows the level of relative residual

And then we will use optimal regularization parameter to show the best solution we can get.

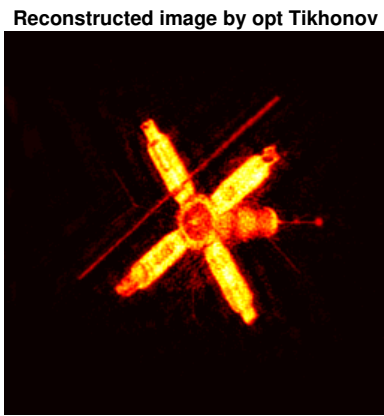


Figure 4.20: Best possible solutions using optimal regularization parameter.

4.2.3 AtmosphericBlur50

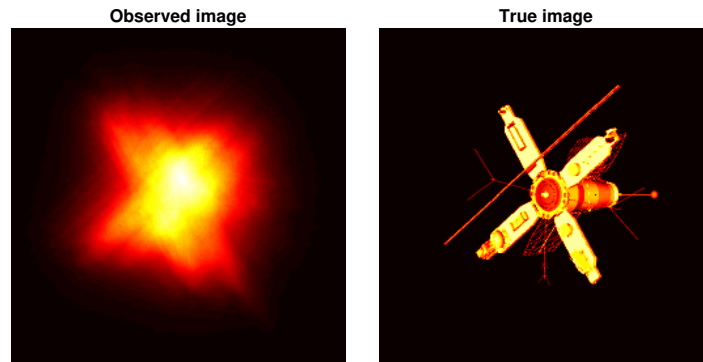


Figure 4.21: This figures show the graph we observe and the true graph before being blurred

By using GCV function we can find α and then use LSQR to solve the Tikhonv regularized least squares problem.

And we also plot out the relative residual graph, where a larger number of iteration leads to a smaller relative residual.

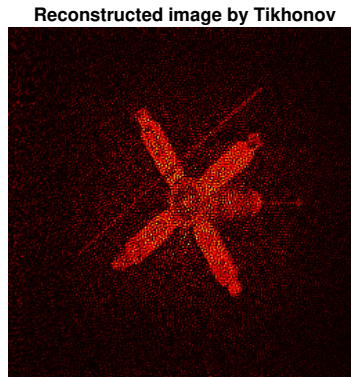


Figure 4.22: This figure shows the computed reconstructions for the satellite problem by using GCV regularization parameter.

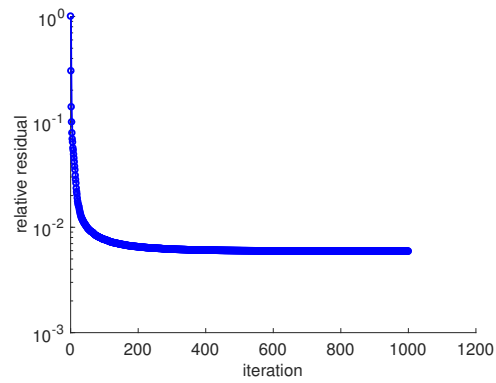


Figure 4.23: This figure shows the level of relative residual

And then we will use optimal regularization parameter to show the best solution we can get.

Reconstructed image by opt Tikhonov

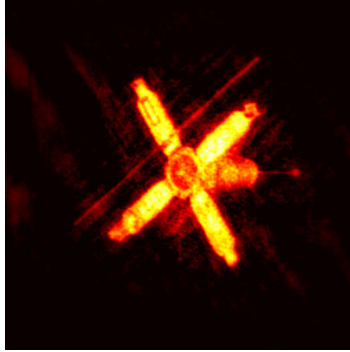


Figure 4.24: Best possible solutions using optimal regularization parameter.

We can see that even for AtmosphericBlur50 case, our GCV Tikhonov still works quite well, which confirm that Tikhonov with GCV parameter choosing function can work to solve some real-world problems.

Chapter 5

Conclusions

We have considered two filtering methods for image restoration and two different methods for choosing the regularization parameter. Specifically, we utilized the singular value decomposition (SVD), as well as tools from numerical linear algebra and MATLAB built-in functions, to efficiently implement each method. Numerical experiments were conducted to demonstrate and compare the performance of two filtering methods in different situations of noise (noise only in b , noise in A , and both). Moreover, we use satellite image restoration to check the application of Tikhonov regularization with parameter chosen by GCV function on a real-world problem.

- We first conclude that by implementing with GCV function both of TSVD and Tikhonov, which are well-known and widely-utilized filtering methods, generally work well for small-scale test problems. This is evident in the comparison of computed solutions in Chapter 4. In addi-

tion, we see that when using the optimal parameters, TSVD works better than Tikhonov in the cases that the true graph is linear or smooth (i.e., very few oscillations).

- When choosing a parameter by GCV function, both TSVD and Tikhonov filtering yield better results in general. However, when we test two regularization tools in different situation of noise, we found that in some cases that with noise in both of the matrix A and the data b , TSVD might not work that good, because noise makes GCV function unable to address a good parameter. In addition, we can also see that in our experiments when using the optimal parameters, TSVD works better than Tikhonov, it might be concluded as TSVD works better than Tikhonov when the true graph is close to linear or has very few oscillation.
- Lastly, we test the performance of Tikhonov with GCV on a real-world problem, solutions restoration of satellite image. By the comparison between solutions processed by Tikhonov with GCV and optimal Tikhonov regularizations, we can obviously see that Tikhonov with GCV can provide a quite good approximation when we having no information about the actual noise level.

The efficient implementation of these methods illustrate that our techniques can serve as a foundation for further studies. According to a deeper level of study about GCV function, people also come up with a method

called weighted GCV, which does improve the performance of GCV function in choosing filtering parameter. Moreover, further investigation of other parameter choice methods such as L-Curve analysis may potentially generate better image restorations [14].

Bibliography

- [1] P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1998.
- [2] P. C. Hansen. *Regularization tools: A Matlab package for the analysis and solution of discrete ill-posed problems*. Numerical Algorithms, 6, 1994, pp.1–35.
- [3] M. Pastorino. *Microwave Imaging*. Wiley Series in Microwave and Optical Engineering. Hoboken: Wiley, 2010.
- [4] J. Mueller and S. Siltanen. *Linear and Nonlinear Inverse Problems with Practical Applications*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2012.
- [5] J. G. Nagy and D. P. O’Leary. *Image Deblurring: I Can See Clearly Now*. IEEE Computing in Science and Engineering. May/June 2003.
- [6] M. R. Banham and A. K. Katsaggelos. *Digital Image Restoration*. IEEE Signal Processing Magazine. March 1997: pp. 24-41.

- [7] N. Nguyen, P. Milanfar and G. Golub. *Efficient Generalized Cross Validation with Applications to Parametric Image Restoration and Resolution Enhancement*. IEEE Transactions on Image Processing. Vol. 10, No. 9, 2001.
- [8] J. Van. *Bringing Fuzzy Field into Focus: Image Help for the Hubble*. Chicago Tribune. December 15, 1991.
- [9] R. Berry and J. Burnell. *The Handbook of Astronomical Image Processing*. Willmann-Bell Inc., Richmond, VA, 2000.
- [10] W. H. Steeb, and H. Yorick. *Matrix Calculus and Kronecker Product : A Practical Approach to Linear and Multilinear Algebra*. 2nd ed. Singapore; Hackensack, N.J.: World Scientific, 2011.
- [11] H. J. Trussell, pages, 193-243. *Academic Press Library in Signal Processing. Vol. 4, Image, Video Processing and Analysis, Hardware, Audio, Acoustic and Speech Processing*. Safari Books Online. LCNAMES. Kidlington, Oxford, UK; Waltham, MA: Academic Press, 2014.
- [12] *Mathematics and Physics of Emerging Biomedical Imaging*. National Research Council Institute of Medicine, National Academy Press, Washington, D.C., 1996.
- [13] P. Linz and R. Wang. *Exploring Numerical Methods: An Introduction to Scientific Computing using MATLAB*. Jones and Bartlett Publishers, Sudbury, MA, 2003.

- [14] C. R. Vogel. *Computational Methods for Inverse Problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2002.