Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced 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 or dissertation in whole or in part in all forms of media, now or hereafter known, 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 or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

_____ _____

Julianne Chung Date

Numerical Approaches for Large-Scale Ill-Posed Inverse Problems

By

Julianne Chung
Doctor of Philosophy

Mathematics and Computer Science

James G. Nagy, Ph.D.
Advisor

Michele Benzi, Ph.D.
Committee Member

Eldad Haber, Ph.D.
Committee Member

Accepted:

Lisa A. Tedesco, Ph.D.
Dean of the Graduate School

Date

Numerical Approaches for Large-Scale Ill-Posed Inverse Problems

By

Julianne Chung
B.A. with Highest Honors in Mathematics, Emory University, 2004

Advisor: James G. Nagy, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the Graduate School
of Emory University in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
in Mathematics and Computer Science

2009

Abstract

Numerical Approaches for Large-Scale Ill-Posed Inverse Problems

By Julianne Chung

Ill-posed inverse problems arise in a variety of scientific applications. Regularization methods exist for computing stable solution approximations, but many of these methods are inadequate or insufficient for solving large-scale problems. This work addresses these limitations by developing advanced numerical methods to solve ill-posed inverse problems and by implementing high-performance parallel code for large-scale applications. Three mathematical models that frequently arise in imaging applications are considered: linear least squares, nonlinear least squares, and nonlinear Poisson maximum likelihood. Hybrid methods are developed for regularization of linear least squares problems, variable projection algorithms are used for nonlinear least squares problems, and reconstruction algorithms are investigated for nonlinear Poisson-based models. Furthermore, an efficient parallel implementation based on the Message Passing Interface (MPI) library is described for use on state-of-the-art computer architectures. Numerical experiments illustrate the effectiveness and efficiency of the proposed methods on problems from image reconstruction, super-resolution imaging, cryo-electron microscopy reconstruction, and digital tomosynthesis.

Numerical Approaches for Large-Scale Ill-Posed Inverse Problems

By

Julianne Chung
B.A. with Highest Honors in Mathematics, Emory University, 2004

Advisor: James G. Nagy, Ph.D.

A dissertation submitted to the Faculty of the Graduate School
of Emory University in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
in Mathematics and Computer Science

2009

Acknowledgments

Words cannot begin to express the gratitude and heartfelt thanks that I owe to my advisor, James Nagy. Jim, you are a truly phenomenal person, and I am so fortunate to have you as my academic father! Thank you for sharing your passions with me and helping me to find mine. You will forever be an exemplary role model and mentor in my life. Thank you for all the knowledge and passion you have shared with me, the compassion and kindness you have extended to me, and the generosity and support you have provided for me over the years. Nine long years ago, you recognized my potential, and, since then, you have never stopped believing in me. I owe my successes to you.

I would also like to express my sincerest gratitude to Michele Benzi and Eldad Haber. I feel very privileged to have learned from your wealth of knowledge and expertise. Thank you for challenging me to grow and for all the support you have provided me during my time at Emory. Thank you for serving on my committee and for your helpful comments to improve this manuscript.

Thank you to the faculty, staff, postdoctoral researchers and graduate students in the Department of Mathematics and Computer Science at Emory University for providing a stimulating environment for me to learn. In addition, I'd like to thank the Emory Graduate School, in particular, thank you to Rosemary Hynes and Geri Thomas for all your support in administering my fellowship.

I am grateful to the Department of Energy's Office of Science and National Nuclear Security Administration for supporting my research goals and dreams through the DOE Computational Science Graduate Fellowship program. To the staff at the Krell Institute, many thanks for all your support and encouragement, and to my fellow CSGFers, your stories and your commitment to science are inspiring. Thank you for sharing your experiences with me.

I have had the privilege to know and work with some amazing researchers who have contributed directly to the completion of my dissertation. Thank

you to Dianne O'Leary for the valuable insight and support you provided to improve the weighted-GCV approach. I am ecstatic about the opportunity to work with you next year! Also, thank you to Ioannis Sechopoulos for introducing me to the polyenergetic tomosynthesis problem and for assisting in the development of the work presented in Chapter 4. I owe many thanks to my practicum mentor Chao Yang at the Lawrence Berkeley National Laboratory for allowing me to get my hands dirty by experiencing large-scale high-performance computing on a real-life application. Thank you for a wonderful summer research experience and for your contributions to my dissertation.

In addition, I have befriended many passionate scientists and researchers during my journey, and I would like to thank them for their professional advice and valuable insights.

To my closest friends, I thank you for always being there for me, for making me laugh even in difficult times, and for helping me to see all the beauty there is in life. Graduate school would not have been nearly as much fun or as memorable without you guys.

Last but not least, I am indebted to my family. To my parents, Andrew and Mary, for all the sacrifices you have made and all the hardships you have endured, I am forever grateful for your unconditional love and support. And to my siblings, Marianne, Karianne, and Matthew, we share so many precious memories and you make my life more enjoyable and fulfilling every day. I love you always.

for my family, friends, and mentors

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Many scientific and engineering applications require numerical methods to compute efficient and reliable solutions to inverse problems. Inverse problems arise in important applications, including biomedical imaging, geophysics, astrophysics, inverse scattering and molecular biology; see for example, [40, 68, 69, 129] and the references therein. Oftentimes, real-life applications require the computer to process extremely large amounts of data, and previously proposed methods for solving inverse problems are not adequate for these large-scale problems. Thus, numerical methods that can efficiently and accurately solve large-scale inverse problems and novel implementation approaches that can take advantage of state-of-the-art parallel computing architectures must be developed. This is the focus of our work.

The basic goal of an inverse problem is to compute an approximation of the original model, given observed data and knowledge about the forward model. Physical systems that require reconstruction of an unknown input signal from the measured output signal are natural examples of inverse problems. One particular application is image deblurring, where the goal is to reconstruct a clearer image, given a blurred image and a point spread function that defines the blur. In other systems, the internal structure of an object is desired, but only measured output data is provided. For example, tomographic reconstruction is the process of reconstructing the inner structures of a three-dimensional (3D) volume, given a collection of two-dimensional (2D)

projection images and knowledge about the tomographic process. Inverse problems can take many forms. Developing a reliable mathematical model and incorporating appropriate regularization are key components for computing accurate solutions. In the next section we consider three mathematical frameworks that are common to many scientific applications.

1.1 Mathematical Models

A solid understanding of the underlying mathematical model is necessary to guide the development of solution methods. In this dissertation three mathematical frameworks are considered.

1.1.1 Linear Least Squares

Linear systems that arise from large-scale inverse problems are typically written as

$$\mathbf{b} = \mathbf{A}\mathbf{x}_{\text{true}} + \boldsymbol{\varepsilon}\,, \tag{1.1}$$

where $\mathbf{b} \in \mathcal{R}^m$ is a known (measured data) vector, $\mathbf{A} \in \mathcal{R}^{m\times n}$ is a matrix describing the forward model, and $\mathbf{x}_{\text{true}} \in \mathcal{R}^n$ represents the true solution. The vector $\boldsymbol{\varepsilon} \in \mathcal{R}^m$ represents unknown perturbations in the data (such as noise). Given $\mathbf{A}$ and $\mathbf{b}$, the aim is to compute an approximation of $\mathbf{x}_{\text{true}}$. We assume that the perturbations are independent and identically distributed with zero mean. With a Gaussian noise model, it is appropriate to consider the least squares formulation:

$$\min_{\mathbf{x}} ||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2\,. \tag{1.2}$$

Inverse problems of this form arise in many important applications, including image reconstruction, image deblurring, geophysics, parameter identification and inverse scattering. Note that we have not included regularization in

(1.2); we discuss this in detail in Section 1.2. Furthermore, for any norms not specified in this dissertation, we assume the 2-norm.

1.1.2 Separable Nonlinear Least Squares

Large-scale inverse problems may also come in nonlinear form:

$$\mathbf{b} = \mathbf{A}(\mathbf{y}_{\text{true}})\mathbf{x}_{\text{true}} + \boldsymbol{\varepsilon}\,, \tag{1.3}$$

where $\mathbf{A}(\mathbf{y}_{\text{true}}) \in \mathcal{R}^{m\times n}$ is a matrix defined by parameter vector $\mathbf{y}_{\text{true}}$, and $\boldsymbol{\varepsilon} \in \mathcal{R}^m$ is unknown additive noise. If matrix $\mathbf{A}(\mathbf{y}_{\text{true}})$ is known exactly, the problem follows the linear model, and the goal is to compute an approximation of $\mathbf{x}_{\text{true}}$. However, in realistic applications, we may only know the parametric form of $\mathbf{A}(\mathbf{y})$, and $\mathbf{y}_{\text{true}}$ must be approximated through additional measurements or device calibration. Thus, the goal of the nonlinear problem is to compute an approximation of $\mathbf{x}_{\text{true}}$, while simultaneously correcting the parameters in $\mathbf{y}$. Similar to the linear model, we assume a Gaussian noise distribution, thus resulting in the following nonlinear least squares system:

$$\min_{\mathbf{x},\mathbf{y}} ||\mathbf{A}(\mathbf{y})\mathbf{x} - \mathbf{b}||_2^2\,. \tag{1.4}$$

This type of problem can be found in applications such as super-resolution imaging, where registration parameters are imprecise, or blind deconvolution, where the point spread function is not known exactly. Furthermore, applications where the acquisition process includes errors may benefit from this work.

1.1.3 Nonlinear Poisson Maximum Likelihood

Another mathematical framework common to image processing applications is a Poisson-based model, in which the data is assumed to be a realization of a Poisson random variable. Tomographic imaging is a classic example of

this model. The observed data is based on counts, e.g. photon counts, and can be represented as

$$\mathbf{b} = \Upsilon[\mathbf{A}\mathbf{x}_{\text{true}}] + \boldsymbol{\varepsilon}\,, \tag{1.5}$$

where $\mathbf{x}_{\text{true}}$ contains voxel values for the true 3D volume, $\mathbf{A} \in \mathcal{R}^{m\times n}$ is a discrete representation of a ray trace operation, $\Upsilon[\cdot]$ models a nonlinear transmission tomography process, and $\boldsymbol{\varepsilon}$ is additive noise (assumed to have a Poisson distribution). Given the observed data and information regarding the tomographic process, the goal is to compute an approximation of $\mathbf{x}_{\text{true}}$ that likely produced the observed data $\mathbf{b}$. To solve this problem, the Poisson distribution is used to formulate the likelihood function, $p(\mathbf{b}, \mathbf{x})$, and standard optimization methods are implemented to maximize the likelihood function. That is, we consider the following problem:

$$\max_{\mathbf{x}} p(\mathbf{b}, \mathbf{x})\,. \tag{1.6}$$

This particular model arises in many imaging applications, but the application that we are interested in is digital tomosynthesis reconstruction for breast imaging. More details of this problem are presented in a later chapter.

1.2 Ill-Posed Problems

In this section a brief overview of ill-posed inverse problems is presented, with particular emphasis on the difficulties one typically encounters when computing solutions for ill-posed inverse problems. The linear least squares model from Section 1.1.1 is used to derive and illustrate some of the characteristics of ill-posed problems, but it is important to remark here that these properties are shared by all ill-posed problems, regardless of the mathematical model.

In the early 1920s, Hadamard first coined the term "ill-posed" [62]. He defined a problem to be ill-posed if the solution does not exist, is not unique, or is not a continuous function of the data. That is, small noise in the data

may give rise to significant errors in the computed approximations. In the linear problem (1.1), the ill-posed nature is revealed by the singular values of $\mathbf{A}$, which decay to and cluster at 0. Thus, $\mathbf{A}$ is severely ill-conditioned, and regularization is used to compute stable approximations of $\mathbf{x}_{\text{true}}$ [40, 58, 68, 129]. Regularization can take many forms; probably the most popular choice is Tikhonov regularization [58], which is equivalent to solving the augmented problem:

$$\min_{\mathbf{x}}\{\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2+\lambda^2\|\mathbf{L}\mathbf{x}\|_2^2\}\,, \tag{1.7}$$

where $\mathbf{L}$ is a regularization operator, often chosen to be the identity matrix or a discretization of a differentiation operator. The regularization parameter λ is a scalar that determines the smoothness of the desired solution. Various techniques can be used to select the regularization parameter, such as the discrepancy principle, the L-curve, or the generalized cross-validation (GCV) method [40, 68, 53, 129]. However, selecting a good parameter is difficult, and there are disadvantages to each of the above approaches [88].

For large-scale problems iterative regularization is a good alternative to direct regularization methods such as Tikhonov regularization. In this case, an iterative method such as LSQR [110] is applied to the least squares problem (1.2). When applied to ill-posed problems, iterative methods exhibit an interesting "semi-convergence" behavior. Specifically, the early iterations reconstruct information about the solution, while later iterations reconstruct information about the noise. If we terminate the iteration when the error is minimized, we obtain a regularized solution. However, the difficulty with using iterative methods for ill-posed inverse problems is that a good stopping point can be hard to know. Approaches used for well-posed problems, such as those based on the residual, generally do not work for ill-posed problems. Furthermore, an imprecise estimate of the termination point can result in a solution whose relative error is significantly higher than the optimal.

Although many regularization methods exist for ill-posed inverse problems,

major difficulties arise when dealing with large-scale problems. This dissertation investigates a variety of approaches for incorporating regularization in large-scale problems. More specifically, efficient hybrid regularization approaches that overcome the limitations of current methods are developed for linear least squares problems, and methods for incorporating regularization in separable nonlinear least squares problems are explored. Regularization for nonlinear Poisson-based models is significantly more challenging but can be achieved using less sophisticated regularization techniques.

1.3 Outline of Work

Developing numerical methods for large-scale ill-posed inverse problems requires a mathematical understanding of the underlying problem, regularization methods for the numerical treatment of the problem, and efficient high-performance implementations. This dissertation addresses all of these issues.

An outline for the work is as follows. Efficient hybrid regularization approaches for linear least squares problems are discussed in Chapter 2. Then in Chapter 3, a variable projection approach is considered for solving the separable nonlinear least squares problem. To efficiently incorporate regularization, connections are made with the hybrid method described in Chapter 2. A nonlinear Poisson-based model is elaborated upon in Chapter 4, in the context of a digital breast tomosynthesis application, and optimization approaches for a statistical model are developed. For large-scale problems, preconditioners can be used to accelerate the convergence rates for previously mentioned approaches. However, assuming that no such effective preconditioners are available, the significant computational burden in all of the proposed methods is the matrix-vector and matrix-transpose-vector multiplications. Chapter 5 presents a high-performance implementation scheme that

allows researchers to perform large-scale computations on state-of-the-art supercomputers. Numerical results, specific imaging applications, and future research directions are presented throughout the dissertation. Concluding remarks can be found in Chapter 6.

1.4 Contributions

The significant contributions of this dissertation research include developing effective approaches for regularizing large-scale linear and nonlinear least squares problems, deriving a novel mathematical framework so that statistical optimization techniques can be used for nonlinear Poisson-based inverse problems, and producing high-performance parallel implementations for execution on massive distributed computing architectures. More specifically, this dissertation presents key contributions in each of the following areas:

- **Linear Least Squares Problems**
  - Hybrid regularization methods are considered for large-scale linear least squares problems. Difficulties arise when using standard numerical techniques. A novel adaptive approach is developed for use in the weighted-generalized cross-validation method for selecting regularization parameters [24].
  - Software contributions for this project include an efficient and reliable MATLAB implementation for hybrid regularization. The codes are publicly available and have been used in a variety of imaging applications including blob-based super-resolution [74], multi-aperture imaging [113], cryo-electron microscopy (Cryo-EM) reconstruction [26], and motion blur removal for positron emission tomography.

- **Separable Nonlinear Least Squares Problems**
  - A variable projection algorithm is used for joint estimation of the model parameters and the desired image. Our contribution is to make it work for large-scale problems and to propose a slick way to incorporate regularization.
  - This work has been successfully applied to imaging applications such as super-resolution imaging and blind deconvolution [21, 22, 23]. Also, it has been cited in a variety of imaging publications [5, 27, 32, 59, 102, 121, 131].
- **Polyenergetic Digital Tomosynthesis**
  - Current algorithms for digital tomosynthesis reconstruction ignore the polyenergetic nature of the incident x-ray spectrum. This may result in artifacts or nonuniformities in the reconstructed images. This dissertation considers a challenging nonlinear inverse problem based on the polyenergetic tomosynthesis model.
  - We formulate a new mathematical framework for polyenergetic tomosynthesis that can take advantage of standard numerical optimization techniques. Some theoretical analysis of the new formulation is provided, and numerical methods are developed for computing quality 3D volume reconstructions [25].
- **High-Performance Implementation**
  - Problems from realistic applications often involve extremely large amounts of data and contain a large number of unknowns. During a research practicum at Lawrence Berkeley National Laboratory, I implemented a two-dimensional data distribution scheme to perform large-volume reconstructions from Cryo-EM data.

– My codes have run successfully on up to 15,344 processors using state-of-the-art supercomputers and are publicly available in the SPARX (single particle analysis for resolution extension) software package [26, 75].

Chapter 2

Linear Least Squares Problems

Many problems follow the linear mathematical model presented in Section 1.1.1. The focus of this chapter is to develop numerical methods for computing a solution to the linear least squares problem:

$$\min_{\mathbf{x}} ||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2 , \tag{2.1}$$

where the underlying problem is ill-posed. Developing efficient solvers for linear least squares systems can also be useful within nonlinear optimization schemes. For example, each iteration of a Gauss-Newton iterative scheme requires an efficient solution approximation for a linear system. In this chapter we investigate and develop efficient methods for linear least squares problems. More specifically, some test examples are presented in Section 2.1, and an overview of regularization approaches is provided in Section 2.2. Section 2.3 provides some background on hybrid methods and describes an adaptive approach for selecting regularization parameters. Numerical results and extensions of this work can be found in Sections 2.5 and 2.6 respectively.

2.1 Test Problems

To illustrate the behavior of our proposed methods, we use six test problems. The first problem comes from the iterative image deblurring package, 'RestoreTools' [96]. Image deblurring has the form (1.1), where the vector $\mathbf{x}_{\text{true}}$

represents the true image scene, $\mathbf{A}$ is a matrix representing a blurring operation, and $\mathbf{b}$ is a vector representing the observed, blurred and noisy image. Given the blurred image and information regarding the blur, the aim is to reconstruct an approximation of the true image. The RestoreTools package has several data sets and tools (such as matrix construction and multiplication routines) that can be used with iterative methods. The data set we use consists of a true image of a ***Satellite*** and a so-called point spread function (PSF) that defines the blurring operation. The matrix $\mathbf{A}$ is constructed from the PSF, using a matrix construction routine in RestoreTools. We then form the noise-free blurred image as $\mathbf{b}_{\text{true}} = \mathbf{A}\mathbf{x}_{\text{true}}$. The MATLAB instructions are the following:

```
>> load satellite
>> A = psfMatrix(PSF);
>> b_true = A*x_true;
```

The images have 256×256 pixels, so the vectors $\mathbf{b}_{\text{true}}$ and $\mathbf{x}_{\text{true}}$ have length $256^2 = 65,536$. The function `psfMatrix` uses an efficient data structure scheme to represent the $65,536 \times 65,536$ matrix $\mathbf{A}$, and the multiplication operator, *, is overloaded to allow for efficient computation of matrix-vector multiplications. For more details, see [96].

The other five test problems are taken from the 'Regularization Tools' package [67]. In each case we generate an $n \times n$ matrix $\mathbf{A}$, true solution vector $\mathbf{x}_{\text{true}}$, and (noise-free) observation vector $\mathbf{b}_{\text{true}}$, setting $n = 256$.

- ***Phillips*** is Phillips' "famous" test problem. $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{x}_{\text{true}}$ are obtained by discretizing the first kind Fredholm integral equation $b(s) =$

$\int_{-6}^{6} a(s,t)x(t)dt$, where

$$\begin{aligned} a(s,t) &= \begin{cases} 1+\cos(\frac{\pi(s-t)}{3}) & , \quad |s-t|<3 \\ 0 & , \quad |s-t|\geq 3 \end{cases} \\ x(t) &= \begin{cases} 1+\cos(\frac{\pi t}{3}) & , \quad |t|<3 \\ 0 & , \quad |t|\geq 3 \end{cases} \\ b(s) &= (6-|s|)\left(1+\frac{1}{2}\cos(\frac{\pi s}{3})\right)+\frac{9}{2\pi}\sin(\frac{\pi|s|}{3})\,. \end{aligned}$$

In MATLAB, the problem can be constructed with the simple statement:

```
>> [A, b_true, x_true] = phillips(n);
```

where `n` is the dimension of the problem.

- ***Shaw*** is a one-dimensional image restoration problem. $\mathbf{A}$ and $\mathbf{x}_{\text{true}}$ are obtained by discretizing, on the interval $-\frac{\pi}{2} \leq s,t \leq \frac{\pi}{2}$, the functions

$$\begin{aligned} a(s,t) &= (\cos(s)+\cos(t))\left(\frac{\sin(u)}{u}\right)^2\,,\ u=\pi(\sin(s)+\sin(t))\,, \\ x(t) &= 2\exp(-6(t-0.8)^2)+\exp(-2(t+0.5)^2)\,. \end{aligned}$$

  Then $\mathbf{b}_{\text{true}} = \mathbf{A}\mathbf{x}_{\text{true}}$. The data can be constructed with the simple MATLAB statement:

```
>> [A, b_true, x_true] = shaw(n);
```

  where `n` is the dimension of the problem.

- ***Deriv2*** constructs $\mathbf{A}$, $\mathbf{b}$ and $\mathbf{x}_{\text{true}}$ by discretizing a first kind Fredholm integral equation, $b(s) = \int_0^1 a(s,t)x(t)dt$, $0 \leq s \leq 1$, where the kernel $a(s,t)$ is given by the Green's function for the second derivative:

$$a(s,t) = \begin{cases} s(t-1) & , \quad s<t \\ t(s-1) & , \quad s\geq t \end{cases}\,.$$

There are several choices for x and b; in this chapter we use $x(t) = t$ and $b(s) = (s^3 - s)/6$. The data can be constructed with the simple MATLAB statement:

```
>> [A, b_true, x_true] = deriv2(n);
```

where `n` is the dimension of the problem.

- ***Baart*** constructs $\mathbf{A}$, $\mathbf{b}$ and $\mathbf{x}_{\text{true}}$ by discretizing the first kind Fredholm integral equation $b(s) = \int_0^\pi a(s,t)x(t)dt$, $0 \le s \le \frac{\pi}{2}$, where

$$\begin{aligned} a(s,t) &= \exp(s\cos t) \\ x(t) &= \sin t \\ b(s) &= \frac{2\sinh s}{s}. \end{aligned}$$

  The data can be constructed with the simple MATLAB statement:

```
>> [A, b_true, x_true] = baart(n);
```

  where `n` is the dimension of the problem.

- ***Heat*** is an inverse heat equation using the Volterra integral equation of the first kind on $[0, 1]$ with kernel $a(s,t) = k(s-t)$, where

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

  The vector $\mathbf{x}_{\text{true}}$ does not have a simple functional representation, but rather is constructed directly as a discrete vector; see [67] for details. The right-hand side $\mathbf{b}$ is produced as $\mathbf{b}_{\text{true}} = \mathbf{A}\mathbf{x}_{\text{true}}$. The data can be constructed with the simple MATLAB statement:

```
>> [A, b_true, x_true] = heat(n);
```

where `n` is the dimension of the problem.

In order to simulate noisy data, as modeled by equation (1.1), we generate a noise vector $\boldsymbol{\varepsilon}$ for each test problem. The entries of $\boldsymbol{\varepsilon}$ are chosen from a normal distribution with mean zero and variance one, and $\boldsymbol{\varepsilon}$ is scaled so that

$$\frac{\|\boldsymbol{\varepsilon}\|_2}{\|\mathbf{A}\mathbf{x}_{\text{true}}\|_2} = 0.1 \quad \text{(i.e., noise level} = 10\%)\,.$$

All of the above examples are ill-posed inverse problems that can be modeled as a linear least squares problem. Thus, appropriate regularization is required for computing meaningful solutions. This is the topic of the next section.

2.2 Regularization

Regularization is a tool for the numerical treatment of ill-posed inverse problems. There are two main approaches for regularization: direct and iterative regularization. As mentioned in the introduction, iterative regularization approaches are generally preferred for large-scale problems, but they suffer from semi-convergence limitations.

To better understand the need for regularization, we first present a theoretical analysis based on the singular value decomposition, or SVD. Let $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ denote the SVD of $\mathbf{A}$, where the columns $\mathbf{u}_i$ of $\mathbf{U}$ and $\mathbf{v}_i$ of $\mathbf{V}$ contain, respectively, the left and right singular vectors of $\mathbf{A}$ and $\boldsymbol{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_n)$ is a diagonal matrix containing the singular values of $\mathbf{A}$, with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$. Using the singular value expansion of $\mathbf{A}$, an inverse solution can be written as

$$\mathbf{x}_{\text{inv}} = \mathbf{A}^{-1}\mathbf{b} = \sum_{i=1}^{n} \frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}\mathbf{v}_i = \underbrace{\sum_{i=1}^{n} \frac{\mathbf{u}_i^T\mathbf{b}_{\text{true}}}{\sigma_i}\mathbf{v}_i}_{\mathbf{x}_{\text{true}}} + \underbrace{\sum_{i=1}^{n} \frac{\mathbf{u}_i^T\boldsymbol{\varepsilon}}{\sigma_i}\mathbf{v}_i}_{\text{error}}\,. \tag{2.2}$$

As indicated above, the inverse solution is comprised of two components: $\mathbf{x}_{\text{true}}$, which is the desired solution, and an error term. Before discussing

algorithms to compute approximations of $\mathbf{x}_{\text{true}}$, it is useful to study the error term.

Matrices arising from ill-posed inverse problems have the following properties.

P1. The matrix $\mathbf{A}$ is severely ill-conditioned, with the singular values σ_i decaying to zero without a significant gap to indicate numerical rank.

P2. The singular vectors corresponding to the small singular values tend to oscillate more (i.e., have higher frequency) than singular vectors corresponding to large singular values.

P3. The components $|\mathbf{u}_i^T \mathbf{b}_{\text{true}}|$ decay on average faster than the singular values σ_i. This is referred to as the discrete Picard condition [68].

From the first two properties, we see that the high frequency components of the error term are highly magnified by division of small singular values. The computed inverse solution (2.2) is dominated by these high frequency components and is, in general, a very poor approximation of $\mathbf{x}_{\text{true}}$. However, the third property suggests that there is hope of reconstructing some information about $\mathbf{x}_{\text{true}}$; that is, an approximate solution can be obtained by reconstructing components corresponding to the large singular values and filtering out components corresponding to small singular values. A filtered, or regularized, solution can be computed as

$$\mathbf{x}_{\text{filt}} = \sum_{i=1}^{n} \phi_i \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i \,, \tag{2.3}$$

where the filter factors, ϕ_i, satisfy $\phi_i \approx 1$ for large σ_i, and $\phi_i \approx 0$ for small σ_i. That is, the large singular value components of the solution are reconstructed, while the components corresponding to the small singular values are filtered out. Different choices of filter factors lead to different methods. For example,

the truncated SVD (TSVD) approach uses filter factors

$$\phi_i = \begin{cases} 1 & \text{if } i \leq \ell \\ 0 & \text{otherwise} \end{cases},$$

where $\ell \leq n$ is a prescribed truncation index that serves as a regularization parameter.

A well-known and widely-used approach called Tikhonov regularization can also be interpreted as a filtering method and is the focus of Section 2.2.1. In addition, the GCV approach for selecting regularization parameters is discussed. Then Section 2.2.2 describes the LSQR method for iterative regularization.

2.2.1 Tikhonov Regularization and GCV

Tikhonov regularization requires solving the minimization problem given in (1.7), where the problem is said to be in standard form if the matrix $\mathbf{L}$ is taken to be the identity matrix $\mathbf{I}$. That is, standard form Tikhonov regularization has the following equivalent formulations:

$$\min_{\mathbf{x}}\{\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2+\lambda^2\|\mathbf{x}\|_2^2\} \quad \Leftrightarrow \quad \min_{\mathbf{x}}\left\|\begin{bmatrix}\mathbf{A}\\ \lambda\mathbf{I}\end{bmatrix}\mathbf{x}-\begin{bmatrix}\mathbf{b}\\ \mathbf{0}\end{bmatrix}\right\|_2^2. \tag{2.4}$$

We remark that other regularization methods, such as generalized Tikhonov regularization (which damps the norm of an appropriate derivative of $\mathbf{x}$), total variation, ℓ_p-norm constraints, or even bound constraints [40, 68, 116, 129], may be preferable in some applications. In the case where $\mathbf{L} \neq \mathbf{I}$, it is common to use a standard form transformation, thereby making the problem simpler from a numerical point of view. That is, if $\mathbf{L}$ is invertible, then we can use the substitution $\mathbf{y} = \mathbf{L}\mathbf{x}$ and get the following form:

$$\min_{\mathbf{y}}\{\|\mathbf{A}\mathbf{L}^{-1}\mathbf{y}-\mathbf{b}\|_2^2+\lambda^2\|\mathbf{y}\|_2^2\}. \tag{2.5}$$

Notice that this formulation is equivalent to right preconditioning of the original system with preconditioner $\mathbf{L}$. Another common approach if $\mathbf{L}$ is not invertible is to use the A-weighted pseudoinverse of $\mathbf{L}$, defined as $\mathbf{L}_A^\dagger = (\mathbf{I} - (\mathbf{A}(\mathbf{I} - \mathbf{L}^\dagger\mathbf{L}))^\dagger\mathbf{A})\mathbf{L}^\dagger$ [39, 68]. The matrix of interest here is $\mathbf{A}\mathbf{L}_A^\dagger$, and the additional computational cost for including the regularization operator in iterative methods includes matrix-vector and matrix-transpose-vector multiplications with $\mathbf{L}_A^\dagger$. However, it may be difficult in practice to work with $\mathbf{L}_A^\dagger$ in this way, so a joint bidiagonalization algorithm has been proposed in [87] that only requires multiplications with $\mathbf{L}$ and $\mathbf{L}^T$. In this dissertation we assume $\mathbf{L} = \mathbf{I}$ and focus on the standard Tikhonov formulation (2.4).

Using the SVD of $\mathbf{A}$, Tikhonov regularization can be written in filtered form as [68]

$$\mathbf{x}_\lambda = \sum_{i=1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}\mathbf{v}_i \,. \tag{2.6}$$

Selecting an appropriate regularization parameter λ is crucial. If λ is too large, the filter factors damp (or, equivalently, filter out) too many of the components in the SVD expansion (2.6), and the corresponding solution is over-smoothed. On the other hand, if λ is too small, the filter factors damp too few components, and the corresponding solution is under-smoothed. In the extreme case, note that choosing $\lambda = 0$ corresponds to $\phi_i = 1$ for all i in (2.3), thereby giving the inverse solution (2.2).

We use a parameter estimation method called generalized cross-validation, or GCV, which is a predictive statistics-based method that does not require *a priori* estimates of the error norm [53, 68]. The basic idea of GCV is that a good choice of λ should predict missing values of the data. That is, if an arbitrary element of the observed data is left out, then the corresponding regularized solution should be able to predict the missing observation fairly well [68]. We leave out each data value in vector $\mathbf{b}$ in turn and seek the value

of λ that minimizes the prediction errors, measured by the GCV function

$$G_{A,\,b}(\lambda) = \frac{n\|(\mathbf{I} - \mathbf{A}\mathbf{A}_\lambda^\dagger)\mathbf{b}\|_2^2}{\Big(\text{trace}(\mathbf{I} - \mathbf{A}\mathbf{A}_\lambda^\dagger)\Big)^2}\,, \tag{2.7}$$

where $\mathbf{A}_\lambda^\dagger = (\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{I})^{-1}\mathbf{A}^T$ represents the pseudo-inverse of $\begin{bmatrix} \mathbf{A} \\ \lambda\mathbf{I} \end{bmatrix}$ and gives the regularized solution, $\mathbf{x}_\lambda = \mathbf{A}_\lambda^\dagger\mathbf{b}$. The subscripts on $G(\lambda)$ are used to emphasize the dependence of the GCV function on the matrix $\mathbf{A}$ and right hand side vector $\mathbf{b}$. By replacing $\mathbf{A}$ with its SVD, (2.7) can be rewritten, in the case $m \geq n$, as

$$G_{A,\,b}(\lambda) = \frac{n\left(\displaystyle\sum_{i=1}^{n}\left(\frac{\lambda^2\mathbf{u}_i^T\mathbf{b}}{\sigma_i^2 + \lambda^2}\right)^2 + \sum_{i=n+1}^{m}(\mathbf{u}_i^T\mathbf{b})^2\right)}{\left((m-n) + \displaystyle\sum_{i=1}^{n}\frac{\lambda^2}{\sigma_i^2 + \lambda^2}\right)^2}\,, \tag{2.8}$$

which is a computationally convenient form to evaluate. Thus, GCV can be easily used with standard minimization algorithms. However, these approaches require computing the singular value decomposition of the matrix $\mathbf{A}$ [55], which may be computationally impractical for large-scale problems.

2.2.2 Iterative Regularization: LSQR

A more favorable approach for regularizing large-scale problems of the form (2.1) is to use a conjugate gradient-type solver such as LSQR [110, 111]. This algorithm exhibits faster convergence than gradient descent methods, especially on ill-conditioned problems. However, the disadvantage is that noise contaminates the solution faster, so a good stopping criterion is crucial.

LSQR is based on the Lanczos bidiagonalization process (LBD)[1] [50]. With $\beta = \|\mathbf{b}\|$ and starting vector $\mathbf{w}_1 = \mathbf{b}/\beta$, LBD uses products of the form $\mathbf{A}^T\mathbf{w}$

[1] Also referred to as Golub-Kahan bidiagonalization [73].

and $\mathbf{Ay}$ to generate matrices

$$\mathbf{W}_k = [\ \mathbf{w}_1 \ \cdots \ \mathbf{w}_k\ ], \quad \mathbf{Y}_k = [\ \mathbf{y}_1 \ \cdots \ \mathbf{y}_k\ ], \quad \mathbf{B}_k = \begin{bmatrix} \alpha_1 & & \\ \beta_2 & \ddots & \\ & \ddots & \alpha_k \\ & & \beta_{k+1} \end{bmatrix}$$

for $k = 1, 2, ...$, where $||\mathbf{w}_k|| = ||\mathbf{y}_k|| = 1$ and $\alpha_k, \beta_k > 0$. Assuming exact arithmetic, we would have $\mathbf{W}_k^T\mathbf{W}_k = \mathbf{Y}_k^T\mathbf{Y}_k = \mathbf{I}$; however, the following relations hold to machine precision:

$$\mathbf{W}_{k+1}(\beta\mathbf{e}_1) = \mathbf{b} \tag{2.9}$$

$$\mathbf{A}\mathbf{Y}_k = \mathbf{W}_{k+1}\mathbf{B}_k \tag{2.10}$$

$$\mathbf{A}^T\mathbf{W}_{k+1} = \mathbf{Y}_k\mathbf{B}_k^T + \alpha_{k+1}\mathbf{y}_{k+1}\mathbf{e}_{k+1}^T\,, \tag{2.11}$$

where $\mathbf{e}_{k+1}$ denotes the last column of the identity matrix of dimension $(k+1)$. Given these relations, LSQR [110] solves the sequence of subproblems

$$\mathbf{f}_k = \underset{\mathbf{f}}{\operatorname{argmin}} ||\mathbf{B}_k\mathbf{f} - \beta\mathbf{e}_1||_2^2\,, \quad \mathbf{x}_k = \mathbf{Y}_k\mathbf{f}_k\,. \tag{2.12}$$

From (2.9) and (2.10) we see that $\mathbf{Ax} - \mathbf{b} = \mathbf{W}_{k+1}(\mathbf{B}_k\mathbf{f}_k - \beta\mathbf{e}_1)$. Since $||\mathbf{W}_{k+1}|| \approx 1$, we see that the vectors $\mathbf{x}_k$ should converge to a solution of (2.1), even if the columns of $\mathbf{W}_{k+1}$ lose orthogonality. As k increases to $k+1$, LSQR uses cheap recursions to update QR factors of $\mathbf{B}_k$ and to obtain $\mathbf{x}_{k+1}$, without having to compute $\mathbf{f}_{k+1}$. The main storage required is for the most recent vectors $\mathbf{w}_k$ and $\mathbf{y}_k$, which are used to generate the next vectors and are then overwritten.

An important property of the LBD process is that for small values of k the singular values of the matrix $\mathbf{B}_k$ approximate very well certain singular values of $\mathbf{A}$, with the quality of the approximation depending on the relative spread of the singular values; specifically, the larger the relative spread, the better

the approximation [8, 54, 118]. For ill-posed inverse problems the singular values decay to and cluster at zero, such as $\sigma_i = O(i^{-c})$ where $c > 1$, or $\sigma_i = O(c^i)$ where $0 < c < 1$ and $i = 1, 2, \ldots, n$ [126, 128]. Thus, the relative gap between large singular values is generally much larger than the relative gap between small singular values. We therefore expect that if we apply the LBD iteration to a linear system arising from discretization of an ill-posed inverse problem, then the singular values of $\mathbf{B}_k$ converge very quickly to the largest singular values of $\mathbf{A}$. The following example illustrates this situation.

Example 2.1 *Consider the inverse heat equation described in Section 2.1 that was generated by the function* `heat` *in Regularization Tools. We are interested in the nonzero singular values of* $\mathbf{A}$ *and their approximations computed from the LBD algorithm. In Figure 2.1 we show a plot of the singular values of* $\mathbf{A}$ *and their relative spread; that is,*

$$\frac{\sigma_i(\mathbf{A}) - \sigma_{i+1}(\mathbf{A})}{\sigma_i(\mathbf{A})},$$

where we use the notation $\sigma_i(\mathbf{A})$ *to denote the* i^{th} *largest singular value of* $\mathbf{A}$.

Figure 2.1 clearly illustrates the properties of ill-posed inverse problems; the singular values of $\mathbf{A}$ *decay to and cluster at 0. Moreover, we clearly see that in general the relative gap of the singular values is larger for the large singular values and smaller for the small singular values. Thus, for small values of* k*, we expect to observe that the singular values of* $\mathbf{B}_k$ *converge quickly to the large singular values of* $\mathbf{A}$*. This can be seen in Figure 2.2, which compares the singular values of* $\mathbf{A}$ *with those of the bidiagonal matrix* $\mathbf{B}_k$ *for* $k = 10, 20, 50$.

The above example implies that if LSQR is applied to the least squares problem $\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$, then at early iterations the approximate solutions $\mathbf{x}_k$ will be in a subspace that approximates a subspace spanned by the large

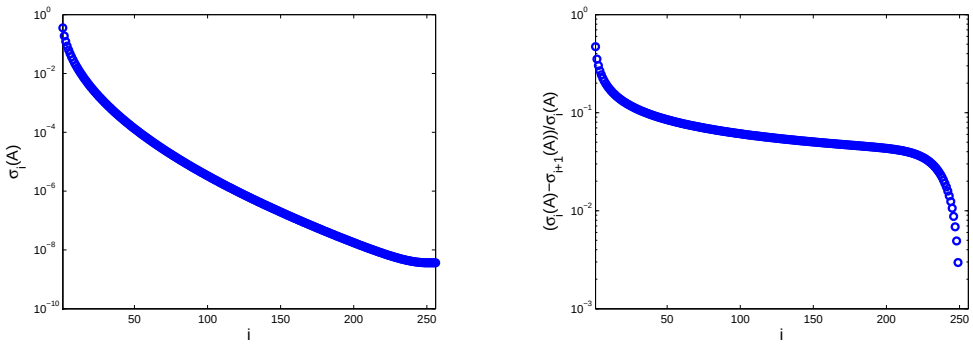


Figure 2.1: Plot of singular values and their relative spread. The left plot contains the singular values of $\mathbf{A}$, denoted as $\sigma_i(\mathbf{A})$, and the right plot contains the relative spread of $\mathbf{A}$'s singular values.

singular components of $\mathbf{A}$. Thus, for $k \ll n$, $\mathbf{x}_k$ is a regularized solution. However, eventually $\mathbf{x}_k$ should converge to the inverse solution, which is corrupted with noise. This means that the iteration index k plays the role of a regularization parameter; if k is too small, then the computed approximation $\mathbf{x}_k$ is an over-smoothed solution, while if k is too large, $\mathbf{x}_k$ is corrupted with noise. More extensive theoretical arguments of this semi-convergence behavior of conjugate gradient methods can be found elsewhere; see [64] and the references therein.

2.3 Hybrid Methods

As described in the previous section, direct methods such as Tikhonov regularization can be computationally impractical for large problems. Furthermore, iterative methods like LSQR when applied to ill-posed inverse problems exhibit inherent semi-convergence, where early iterations tend to approximate spectral components corresponding to signal, while later iterations be-

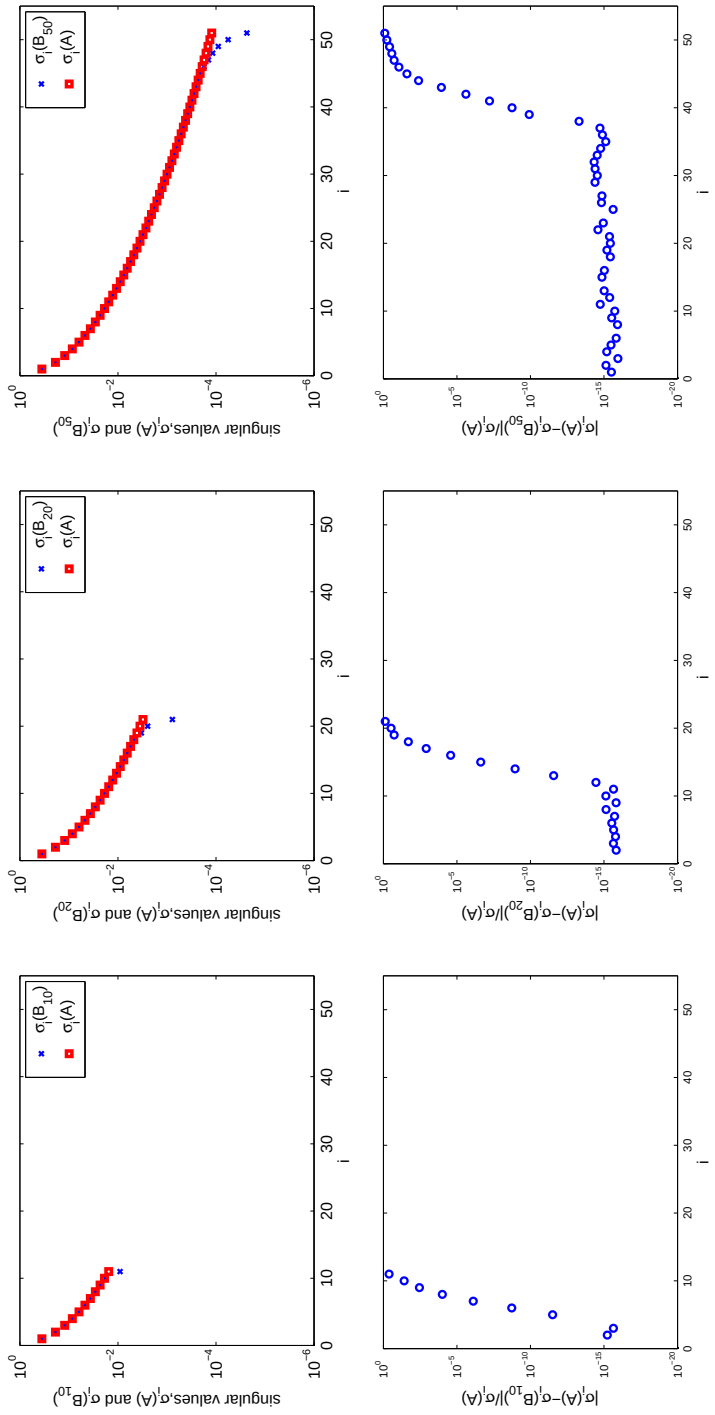


Figure 2.2: Convergence of singular values. The plots in the top row of this figure show the singular values of $\mathbf{A}$, denoted as $\sigma_i(\mathbf{A})$, along with the singular values of $\mathbf{B}_k$, denoted as $\sigma_i(\mathbf{B}_k)$, for $k = 10, 20, 50$. The plots in the bottom row show the relative difference, $\dfrac{|\sigma_i(\mathbf{A}) - \sigma_i(\mathbf{B}_k)|}{\sigma_i(\mathbf{A})}$.

come contaminated with noise [68]. A good stopping criterion is required for computing reliable solutions. To overcome these limitations, we consider hybrid methods for large-scale ill-posed inverse problems.

Previous work on hybrid methods can be divided broadly into two categories: those that use iterative methods to solve the regularized problem and those that embed regularization within an iterative scheme. In this work we focus on the latter case, but first we make some remarks on the former.

LSQR with Regularization

If a suitable regularization parameter, λ, is known in advance, the LSQR algorithm can incorporate Tikhonov regularization, and stopping rules can be implemented that are generally reliable [111]. That is, it has been proposed to use the LSQR algorithm to solve the Tikhonov problem (2.4), where λ is a fixed regularization parameter. That is, for $\lambda \geq 0$, LSQR(λ) [111] solves the subproblems

$$\mathbf{f}_k = \underset{\mathbf{f}}{\operatorname{argmin}} \left\| \begin{bmatrix} \mathbf{B}_k \\ \lambda \mathbf{I} \end{bmatrix} \mathbf{f} - \begin{bmatrix} \beta \mathbf{e}_1 \\ 0 \end{bmatrix} \right\|^2, \quad \mathbf{x}_k = \mathbf{Y}_k \mathbf{f}_k \,, \tag{2.13}$$

using similar recursions and slightly more elaborate QR factors. Essentially no more storage or work is required. Since $||\mathbf{Y}_k|| \approx 1$, we find from (2.9)-(2.11) that the vectors $\mathbf{f}_k$ should converge reliably to a solution of (2.4), even if $\mathbf{W}_{k+1}$ and $\mathbf{Y}_k$ lose orthogonality.

This is an efficient approach if a good value of λ is known. However, if λ is too small and the problem is ill-posed, LSQR(λ) also exhibits semi-convergence behavior. This would be useful as long as LSQR's stopping rules result in termination at the "right time."

As mentioned in the previous section, obtaining a good value of λ for large-scale problems can be very difficult. Standard regularization parameter selection methods typically require a good estimate of the noise level or the

SVD of matrix $\mathbf{A}$. Other methods such as L-curve require the solution of (2.4) for several regularization parameters. This limitation can be partially alleviated by exploiting redundancies and additional information available in certain iterative methods [15, 47].

Another option that has been proposed for large-scale problems is to formulate Tikhonov regularization as a quadratically constrained least squares problem, where the Lagrange multiplier serves as a regularization parameter [38, 56]. However, the computational cost can still be prohibitive for very large matrices, and the method proposed in [56] would need to be implemented carefully to avoid failure when a trial choice of parameter in the iteration is poor [17]. We propose to use an alternate approach that can automatically select regularization parameters and a stopping iteration.

Embedded Regularization

Another approach for stabilizing the semi-convergence behavior of LSQR and regularizing large-scale problems is to embed a direct regularization scheme, such as Tikhonov or TSVD, within an iterative Lanczos bidiagonalization algorithm [7, 9, 16, 66, 87, 88, 92, 107]. The basic idea of this approach is to project the large-scale problem onto Krylov subspaces of small (but increasing) dimension. Then the projected problem can be solved cheaply by using any direct regularization method.

In particular, we consider the hybrid methods of [107, 7], which are based on the Lanczos bidiagonalization algorithm, and we are interested in projected problem (2.12). Recall from the previous example that the singular values of $\mathbf{B}_k$ converge quickly to the large singular values of $\mathbf{A}$. However, since the original problem is ill-posed, $\mathbf{B}_k$ will eventually approximate the small singular values of $\mathbf{A}$, thereby becoming very ill-conditioned also. Since $\mathbf{B}_k$ is much smaller than $\mathbf{A}$, a number of spectral filtering algorithms [68] can

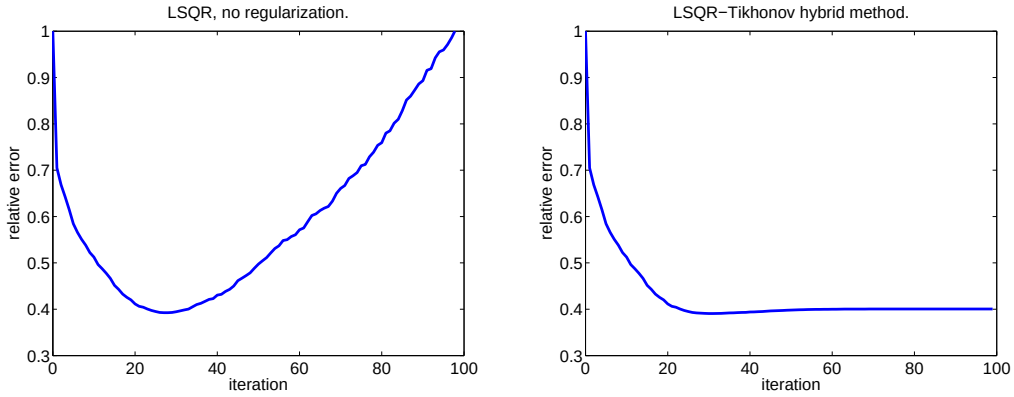


Figure 2.3: Semi-convergence behavior of LSQR and stabilization using a hybrid method. These plots represent relative errors, $\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|_2 / \|\mathbf{x}_{\text{true}}\|_2$, where $\mathbf{x}_{\text{true}}$ is the true solution and $\mathbf{x}_k$ is the solution at the k^{th} iteration. The left plot illustrates semi-convergence behavior of the iterative method LSQR for an ill-posed problem. The right plot illustrates how this semi-convergence behavior can be stabilized with an iterative LSQR-Tikhonov hybrid method.

be used to provide regularization at each iteration. O'Leary and Simmons [107] proposed using Tikhonov regularization to solve the projected problem, and Björck [7] suggested using TSVD with GCV to choose the regularization parameters. A variety of existing methods can be implemented. For a comparative study, see Kilmer and O'Leary [88].

The potential benefits of this approach are illustrated in the right plot of Figure 2.3. Notice that, in contrast to the behavior of the relative errors for LSQR, the hybrid approach can effectively stabilize the iteration so that an imprecise (over) estimate of the stopping iteration does not have a deleterious effect on the computed solution. Björck [7] has suggested using GCV as a way to determine an appropriate stopping iteration in the hybrid approach.

A disadvantage of this hybrid approach is that at each iteration we must choose a new regularization parameter for the projected problem. Although

this is not computationally expensive, in order for the approach to be viable for practical problems, we must choose good parameters. Optimal choices for the parameter at each iteration result in convergence behavior similar to that illustrated in the right plot of Figure 2.3. However, our computational experience indicates that such optimal behavior cannot be expected when using parameter selection methods such as the discrepancy principle, GCV, and the L-curve (see also [88]).

2.3.1 Tikhonov and GCV in Hybrid Methods

At the k^{th} iteration, we consider using Tikhonov regularization to regularize projected problem (2.12). Thus, we would like to solve

$$\mathbf{f}_k = \underset{\mathbf{f}}{\operatorname{argmin}} \left\| \begin{bmatrix} \mathbf{B}_k \\ \lambda_k \mathbf{I} \end{bmatrix} \mathbf{f} - \begin{bmatrix} \beta \mathbf{e}_1 \\ 0 \end{bmatrix} \right\|^2 , \quad \mathbf{x}_k = \mathbf{Y}_k \mathbf{f}_k \, . \tag{2.14}$$

Notice that the difference between (2.13) and (2.14) is that the regularization parameter, λ_k, is allowed to change per iteration here. We consider the GCV function to select these parameters:

$$G_{B_k,\, \beta e_1}(\lambda) = \frac{k \|(\mathbf{I} - \mathbf{B}_k \mathbf{B}_{k,\lambda}^{\dagger}) \beta \mathbf{e}_1\|_2^2}{\Big(\operatorname{trace}(\mathbf{I} - \mathbf{B}_k \mathbf{B}_{k,\lambda}^{\dagger})\Big)^2} \, .$$

Notice the dependence on matrix $\mathbf{B}_k$ and right hand side vector $\beta \mathbf{e}_1$. If we define the SVD of the $(k+1) \times k$ matrix $\mathbf{B}_k$ as

$$\mathbf{B}_k = \mathbf{P}_k \begin{bmatrix} \mathbf{\Delta}_k \\ \mathbf{0}^T \end{bmatrix} \mathbf{Q}_k^T \, , \tag{2.15}$$

then $G_{B_k,\, \beta e_1}(\lambda)$ can be written as

$$G_{B_k,\, \beta e_1(\lambda)} = \frac{k\beta^2 \left(\displaystyle\sum_{i=1}^{k} \left(\frac{\lambda^2}{\delta_i^2 + \lambda^2} \left[\mathbf{P}_k^T \mathbf{e}_1 \right]^{(i)} \right)^2 + \left(\left[\mathbf{P}_k^T \mathbf{e}_1 \right]^{(k+1)} \right)^2 \right)}{\left(1 + \displaystyle\sum_{i=1}^{k} \frac{\lambda^2}{\delta_i^2 + \lambda^2} \right)^2} , \tag{2.16}$$

where $\left[\mathbf{P}_k^T \mathbf{e}_1\right]^{(j)}$ denotes the j^{th} component of the vector $\mathbf{P}_k^T \mathbf{e}_1$, and δ_i is the i^{th} largest singular value of $\mathbf{B}_k$ (i.e., the i^{th} diagonal element of $\mathbf{\Delta}_k$).

2.3.2 Difficulty in using GCV in Hybrid Methods

In this section we use standard GCV to choose the Tikhonov regularization parameter λ_k at each iteration of the Lanczos-based hybrid methods for the test problems in Section 2.1. The results are shown in Figure 2.4. In all of our examples, LSQR, which is essentially LBD with no regularization, exhibits semi-convergent behavior, as we expect. If we use "optimal" regularization parameters at each iteration (determined using knowledge of $\mathbf{x}_{\text{true}}$ to make the relative error in the solution as small as possible), then Lanczos-hybrid methods would be excellent at stabilizing the regularized solution, as shown with the dashed lines. However, in realistic situations, we do not know the optimal solution, so this is impossible. On the *Phillips, Shaw* and *Deriv2* problems, the performance of standard GCV, though slightly worse than optimal, is acceptable. For the other three problems, the convergence behavior when using standard GCV is significantly worse than when the optimal parameters are used.

A major concern is the possibility that rounding errors in the computation of the matrices $\mathbf{W}_k$, $\mathbf{Y}_k$ and $\mathbf{B}_k$ are causing the poor behavior. Björck, Grimme and Van Dooren [9] showed that in some cases reorthogonalization may be necessary for better performance, and Larsen [92] considered partial reorthogonalization. However, in our tests GCV still had difficulty even after reorthogonalization. Another option is to use a different regularization method such as TSVD or exponential filtering, but we found little to no improvement in the solution. In addition, we delayed regularization until after $k > k_{min}$ to wait until $\mathbf{B}_k$ more fully captures the ill-conditioning of $\mathbf{A}$, but that attempt proved futile as well. These phenomena are illustrated in

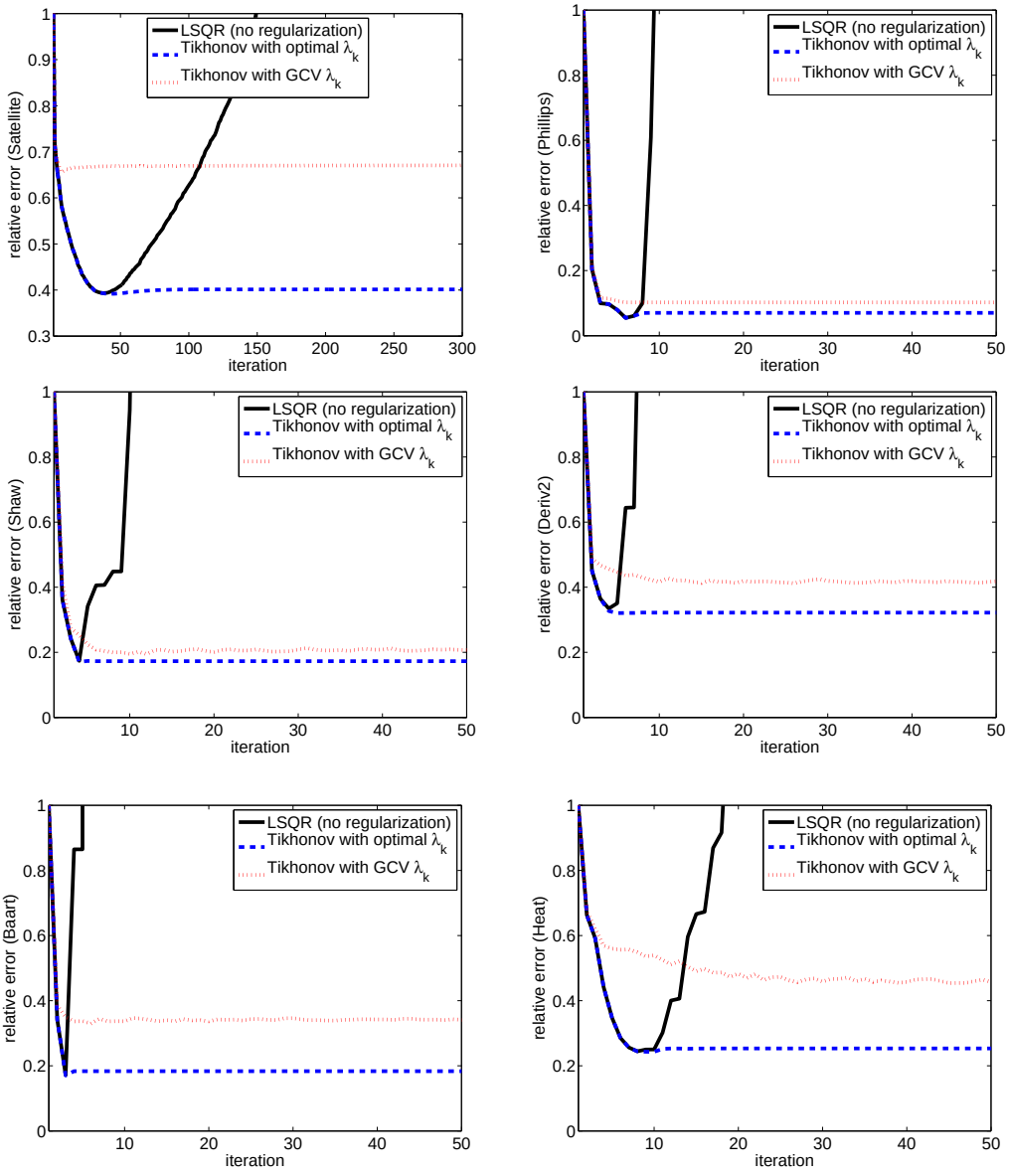


Figure 2.4: Relative errors with standard GCV. These plots show the relative error, $\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|_2/\|\mathbf{x}_{\text{true}}\|_2$, at each iteration of LSQR and the Lanczos-hybrid method. Upper left: Satellite. Upper right: Regtools-Phillips. Middle left: Regtools-Shaw. Middle right: Regtools-Deriv2. Bottom left: Regtools-Baart. Bottom right: Regtools-Heat.

the following example.

Example 2.2 *In Figure 2.5, we present observations corresponding to the Satellite example. In particular, various modifications were attempted to improve the results when using standard GCV to select regularization parameters in the Lanczos-hybrid method. Although the convergence results after using full reorthogonalization produced slightly smaller solution errors than those using no reorthogonalization (see the top left plot), the results are still not ideal. Furthermore, we found no improvement in using TSVD rather than Tikhonov filtering for this particular example (top right). The two bottom plots correspond to delaying regularization of the projected system until after 25 and 75 Lanczos iterations respectively. As we see, there is no benefit in delaying regularization for the projected problem since the errors immediately jump to the non-ideal curve from using Tikhonov with standard GCV.*

It is evident from these examples that there are good choices of the regularization parameters. However, the poor behavior is caused by the suboptimal parameter chosen by GCV. In particular, the standard GCV function is selecting regularization parameters at each iteration that are much too large. In the next section we propose replacing it by a weighted-GCV method that shows much better behavior.

2.4 Weighted GCV Method

In this section we describe a modification of the GCV function, called weighted-GCV (W-GCV), that improves our ability to choose regularization parameters for the projected problem. We first describe the approach for Tikhonov regularization for a general linear system of equations. Then in Section 2.4.3 we show how to apply it to the projected problem.

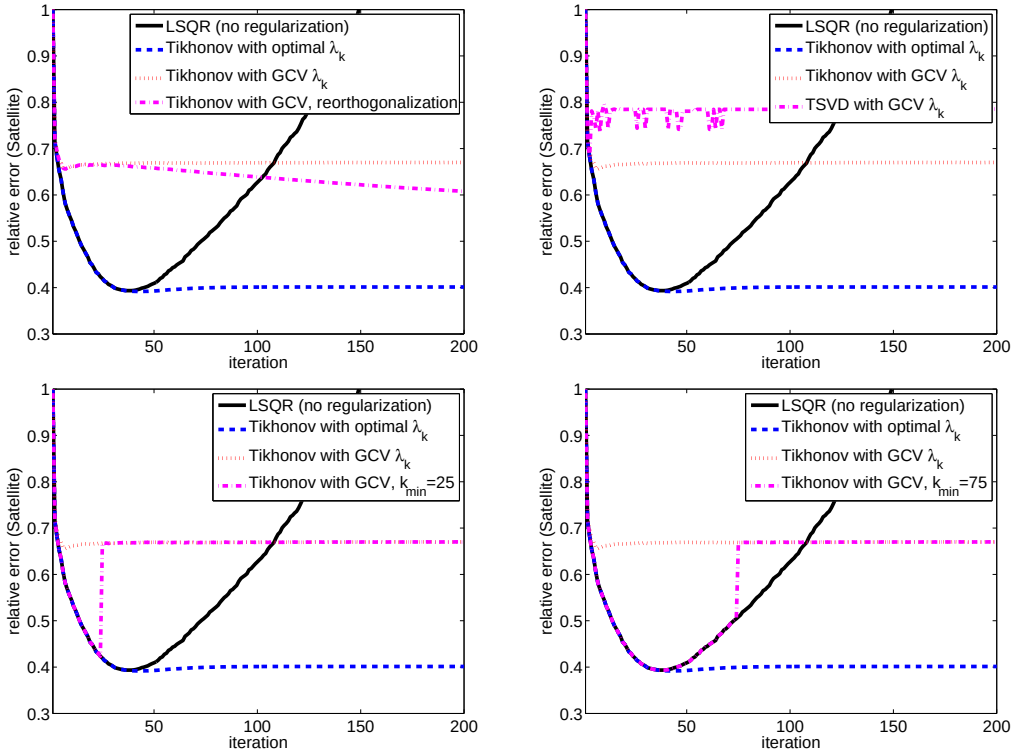


Figure 2.5: Relative errors for the Satellite example with standard GCV. These plots show the relative error, $\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|_2/\|\mathbf{x}_{\text{true}}\|_2$, at each iteration of LSQR and the Lanczos-hybrid method for the Satellite example. The top left plot provides results after full reorthogonalization of the Lanczos vectors, and the top right plot provides results after using TSVD filtering, rather than Tikhonov. The bottom plots show the results from delayed regularization of the projected problem.

2.4.1 W-GCV for Tikhonov Regularization

The standard GCV method is a popular parameter choice method used in a variety of applications; however, as illustrated in Section 2.3.2, the method may not perform well for certain classes of problems. Other studies in statistical nonparametric modeling and function approximation noted that in practical applications, GCV occasionally chose Tikhonov parameters too small, thereby under-smoothing the solution [28, 46, 89, 106, 127]. To circumvent this problem, these papers use a concept that we call weighted-GCV. In contrast, we observed over-smoothing difficulties when using GCV in Lanczos-hybrid methods, which motivated us to use a different range of weights in the W-GCV method.

Instead of the Tikhonov GCV function defined in (2.7), we consider the weighted-GCV function

$$G_{A,b}(\omega,\lambda) = \frac{n||(\mathbf{I}-\mathbf{A}\mathbf{A}_\lambda^\dagger)\mathbf{b}||^2}{\Big(\text{trace}(\mathbf{I}-\omega\mathbf{A}\mathbf{A}_\lambda^\dagger)\Big)^2}\,. \tag{2.17}$$

Notice the function's dependency on a new parameter ω in the denominator trace term. Choosing $\omega = 1$ gives the standard GCV function (2.7). If we choose $\omega > 1$, we obtain smoother solutions, while $\omega < 1$ results in less smooth solutions. The obvious question here is how to choose a good value for ω. To our knowledge, in all work using W-GCV, only experimental approaches are used to choose ω. For smoothing spline applications, Kim and Gu empirically found that standard GCV consistently produced regularization parameters that were too small, while choosing ω in the range of 1.2-1.4 worked well [89]. In our problems, though, the GCV regularization parameter is chosen too large; thus, we seek a parameter ω in the range $0 < \omega \leq 1$. In addition, rather than using a user-defined parameter choice for ω as in previous papers, we propose a more automated, adaptive approach that is also versatile and can be used on a variety of problems.

2.4.2 Interpretations of the W-GCV Method

In this section we consider the W-GCV method and look at various theoretical aspects of the method. By looking at different interpretations of the W-GCV method, we hope to shed some light on the role of the new parameter, ω.

As mentioned in Section 2.2.1, the standard GCV method is a "leave-one-out" prediction method [53]. In fact, in leaving out the j^{th} observation, the derivation seeks to minimize the prediction error, given by

$$\sum_{i=1, i\neq j}^{m} (b^{(i)} - [\mathbf{Ax}]^{(i)})^2 + \lambda^2 ||\mathbf{x}||_2^2 \,,$$

where $b^{(i)}$ and $[\mathbf{Ax}]^{(i)}$ are the i^{th} entries of vectors $\mathbf{b}$ and $\mathbf{Ax}$ respectively. If we define the $m \times m$ matrix

$$\mathbf{E}_j = \text{diag}(1, 1, ...1, 0, 1, ...1),$$

where 0 is the j^{th} entry, then the above minimization is equivalent to

$$\min_{\mathbf{x}} ||\mathbf{E}_j(\mathbf{b} - \mathbf{Ax})||_2^2 + \lambda^2 ||\mathbf{x}||_2^2.$$

We can derive the W-GCV method in a similar manner, but we instead use a weighted "leave-one-out" philosophy. More specifically, consider the case $0 < \omega < 1$. Then define the matrix

$$\mathbf{F}_j = \text{diag}(1, 1, ...1, \sqrt{1-\omega}, 1, ...1),$$

where $\sqrt{1-\omega}$ is the j^{th} diagonal entry of $\mathbf{F}_j$. By using the W-GCV method, we seek a solution to the following minimization problem:

$$\min_{\mathbf{x}} ||\mathbf{F}_j(\mathbf{b} - \mathbf{Ax})||_2^2 + \lambda^2 ||\mathbf{x}||_2^2.$$

In this problem, the j^{th} observation is still present but has been down-weighted by the factor $\sqrt{1-\omega}$; thus, it is completely left out when $\omega = 1$. A

more detailed derivation of the W-GCV method can be found in Appendix Section A.1.

By introducing a new parameter in the trace term of the GCV function, we not only introduce a new weighted prediction approach but also change the interpretation of the function we are minimizing, perhaps including bias in the estimator. We consider the special case of Tikhonov regularization and look at how the GCV function is altered algebraically and graphically with the new parameter. Using the SVD expansion of $\mathbf{A}$, it can be shown that the trace term in the standard GCV function is given by

$$\text{trace}(\mathbf{I} - \mathbf{A}\mathbf{A}_\lambda^\dagger) \;=\; \sum_{i=1}^{n} \frac{\lambda^2}{\sigma_i^2 + \lambda^2} + (m-n).$$

In contrast, the trace term for the W-GCV function is given by

$$\begin{aligned} \text{trace}(\mathbf{I} - \omega\mathbf{A}\mathbf{A}_\lambda^\dagger) &= \sum_{i=1}^{n} \frac{(1-\omega)\sigma_i^2 + \lambda^2}{\sigma_i^2 + \lambda^2} + (m-n) \\ &= \sum_{i=1}^{n} (1-\omega)\phi_i + \sum_{i=1}^{n} \frac{\lambda^2}{\sigma_i^2 + \lambda^2} + (m-n). \end{aligned}$$

Thus, if $\omega < 1$ then we are adding a multiple of the sum of the filter factors to the original trace term, and if $\omega > 1$ we are subtracting a multiple. The graph of the GCV function also undergoes changes as ω is changed from 1. The denominator becomes zero for some value of $\omega > 1$, so the W-GCV function has a pole. Fortunately, in our case, $0 < \omega \leq 1$. Note that larger values of ω result in larger computed regularization parameters, and smaller values of ω result in smaller values of λ.

2.4.3 W-GCV for the Bidiagonal System

In the previous section we discussed W-GCV in the context of Tikhonov regularization on the original (full) system of equations involving $\mathbf{A}$ and $\mathbf{b}$.

This allowed us to provide a general description, but our aim is to apply W-GCV to choosing regularization parameters for the projected problem, (2.12). In this case, the W-GCV function has the form:

$$\begin{aligned} G_{B_k,\,\beta e_1}(\omega,\lambda) &= \frac{k\|(\mathbf{I}-\mathbf{B}_k\mathbf{B}_{k,\lambda}^{\dagger})\beta\mathbf{e}_1\|_2^2}{\left(\text{trace}(\mathbf{I}-\omega\mathbf{B}_k\mathbf{B}_{k,\lambda}^{\dagger})\right)^2} \\ &= \frac{k\beta^2\left(\sum_{i=1}^{k}\left(\frac{\lambda^2}{\delta_i^2+\lambda^2}\left[\mathbf{P}_k^T\mathbf{e}_1\right]^{(i)}\right)^2+\left(\left[\mathbf{P}_k^T\mathbf{e}_1\right]^{(k+1)}\right)^2\right)}{\left(1+\sum_{i=1}^{k}\frac{(1-\omega)\delta_i^2+\lambda^2}{\delta_i^2+\lambda^2}\right)^2}, \end{aligned}$$

where, using the notation introduced in (2.16), $\mathbf{P}_k$ is an orthogonal matrix containing the left singular vectors of $\mathbf{B}_k$, δ_i is the i^{th} largest singular value of $\mathbf{B}_k$, and $\mathbf{W}_k^T\mathbf{b} = \beta\mathbf{e}_1$ with $\beta = \|\mathbf{b}\|$. Note that this reduces to the expression in (2.16) when $\omega = 1$.

2.4.4 Choosing ω

For many ill-posed problems, a good value of ω is crucial for the success of Lanczos-hybrid methods. In this section we consider how different values of ω may affect convergence behavior, and we present a heuristic, adaptive approach for finding a good value for ω.

Example 2.3 *Consider the test problem Heat, whose convergence graph with Tikhonov regularization and the standard GCV method is given in the bottom right corner of Figure 2.4. To illustrate the effects of using the W-GCV function with Lanczos-hybrid methods, we fix a value of ω for all iterations and present relative error plots. The results are shown in Figure 2.6.*

For this particular example, it is evident that $\omega = 0.2$ is a good value for the new parameter. However, finding a good ω in this way is not possible since the true solution is generally not available. Hence, we introduce

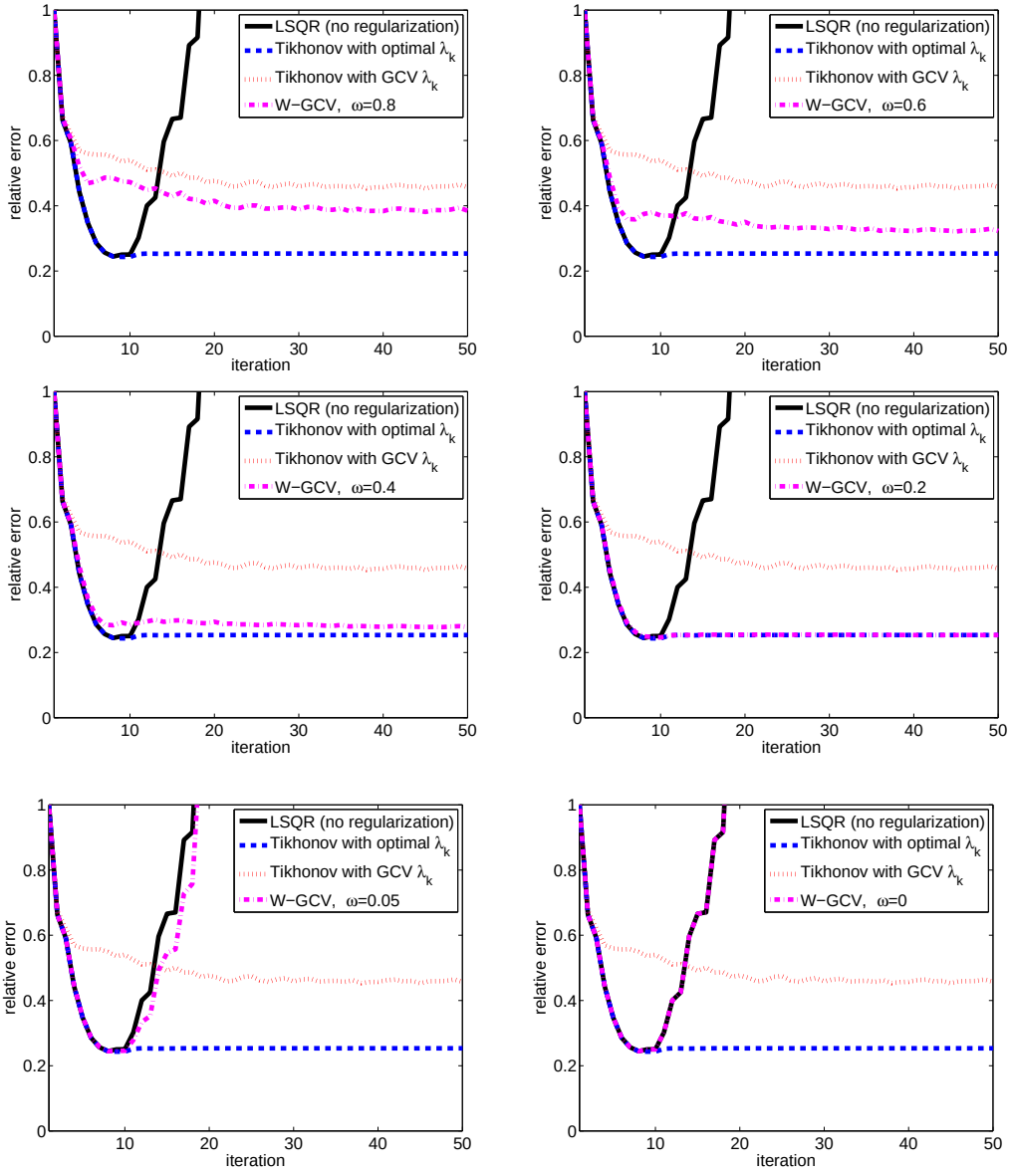


Figure 2.6: Relative errors for the Heat example with different values of ω. The various plots show how the convergence behavior changes when regularization parameters are chosen using the W-GCV method with different values of ω. Note that $\omega = 1$ is equivalent to using standard GCV, and $\omega = 0$ is equivalent to using no regularization.

an automated, adaptive approach that in our experience produces adequate results.

Recall from Section 2.2.2 that at each iteration of the Lanczos-hybrid method, we solve the projected least squares problem (2.12) using Tikhonov regularization. Since the early iterations of LBD do not capture the ill-conditioning of the problem, we expect that little or no regularization is needed to solve the projected least squares problem. Let $\lambda_{k,opt}$ denote the optimal regularization parameter at the k^{th} iteration. Then, we can assume that for small k, $\lambda_{k,opt}$ should satisfy

$$0 \leq \lambda_{k,opt} \leq \sigma_{min}\left(\mathbf{B}_k\right),$$

where σ_{min} denotes the smallest singular value of the matrix. If at iteration k, we assume that we know $\lambda_{k,opt}$, then we can find ω by minimizing the GCV function with respect to ω. That is, solving

$$\frac{\partial}{\partial\lambda}\left(G_{B_k,\,\beta e_1}(\omega,\lambda)\right)\Big|_{\lambda=\lambda_{k,opt}} = 0.$$

See Appendix Section A.2 for the details of computing the above derivative and solving for ω. Since we do not know $\lambda_{k,opt}$, we instead find $\hat{\omega}_k$ corresponding to $\lambda_{k,opt} = \sigma_{min}\left(\mathbf{B}_k\right)$. In later iterations, this approach fails because $\sigma_{min}\left(\mathbf{B}_k\right)$ becomes nearly zero due to ill-conditioning. For these iterations, a better approach is to adaptively take $\omega_k = \text{mean}\left\{\hat{\omega}_1, \hat{\omega}_2, \ldots, \hat{\omega}_k\right\}$. By averaging the previously computed ω values, we are essentially using the earlier well-conditioned components of our problem to help stabilize the harmful effects of the smaller singular values. There are two disadvantages to this approach. First, it over-smooths the solutions at early iterations, since it uses a rather large value of λ for a well-conditioned problem. Since these solutions are discarded, this is not a significant difficulty. Second, it under-smooths values for large k, so semi-convergence will eventually reappear. However, in practice we will also be using a method like GCV to choose a stopping

iteration, so k will not be allowed to grow too large; this is discussed in the next section.

2.4.5 Stopping Criteria for LBD

The next practical issue to consider is an approach to determine an appropriate point at which to stop the iteration. Björck [7] suggested using GCV for this purpose, when TSVD is used to solve the projected problem. However, Björck, Grimme and van Dooren [9] showed that modifications of the algorithm were needed to make the approach effective for practical problems. Specifically, they proposed a fairly complicated scheme based on implicitly restarting the iterations.

In this section we describe a similar approach for Tikhonov regularization, but we do not need implicit restarts. We begin by defining the computed solution at each iteration of the Lanczos-hybrid method as

$$\mathbf{x}_k = \mathbf{Y}_k \mathbf{f}_{\lambda_k} = \mathbf{Y}_k (\mathbf{B}_k^T \mathbf{B}_k + \lambda_k^2 \mathbf{I})^{-1} \mathbf{B}_k^T \mathbf{W}_k^T \mathbf{b} \equiv \mathbf{A}_k^\dagger \mathbf{b} \,. \tag{2.18}$$

Using the basic idea of GCV, we would like to determine a stopping iteration, k, that minimizes

$$\widehat{G}(k) \;=\; \frac{n \|(\mathbf{I} - \mathbf{A}\mathbf{A}_k^\dagger)\mathbf{b}\|_2^2}{\Big(\mathrm{trace}(\mathbf{I} - \mathbf{A}\mathbf{A}_k^\dagger)\Big)^2} \,. \tag{2.19}$$

Using (2.18) and (2.10), the numerator of equation (2.19) can be written as

$$n \|(\mathbf{I} - \mathbf{A}\mathbf{A}_k^\dagger)\mathbf{b}\|_2^2 = n \|(\mathbf{I} - \mathbf{B}_k (\mathbf{B}_k^T \mathbf{B}_k + \lambda_k^2 \mathbf{I})^{-1} \mathbf{B}_k^T) \beta \mathbf{e}_1 \|_2^2 \,. \tag{2.20}$$

If we now replace $\mathbf{B}_k$ with its SVD (2.15), we obtain

$$\begin{aligned} n\|(\mathbf{I}-\mathbf{A}\mathbf{A}_k^{\dagger})\mathbf{b}\|_2^2 &= n\beta^2 \left\| \begin{bmatrix} \frac{\lambda_k^2}{\delta_1^2+\lambda_k^2} & & & \\ & \ddots & & \\ & & \frac{\lambda_k^2}{\delta_k^2+\lambda_k^2} & \\ & & & 1 \end{bmatrix} \mathbf{P}_k^T\mathbf{e}_1 \right\|_2^2 \\ &= n\beta^2 \left(\sum_{i=1}^{k} \left(\frac{\lambda_k^2}{\delta_i^2+\lambda_k^2} \left[\mathbf{P}_k^T\mathbf{e}_1\right]^{(i)} \right)^2 + \right. \\ &\qquad \left. \left(\left[\mathbf{P}_k^T\mathbf{e}_1\right]^{(k+1)} \right)^2 \right) . \quad (2.21) \end{aligned}$$

Similarly, the denominator of equation (2.19) can be written as

$$\left(\text{trace}\left(\mathbf{I}-\mathbf{A}\mathbf{A}_k^{\dagger}\right)\right)^2 = \left((m-k) + \sum_{i=1}^{k} \frac{\lambda_k^2}{\delta_i^2+\lambda_k^2} \right)^2 . \quad (2.22)$$

Thus, combining (2.21) and (2.22), equation (2.19) can be written as

$$\widehat{G}(k) = \frac{n\beta^2 \left(\sum_{i=1}^{k} \left(\frac{\lambda_k^2}{\delta_i^2+\lambda_k^2} \left[\mathbf{P}_k^T\mathbf{e}_1\right]^{(i)} \right)^2 + \left(\left[\mathbf{P}_k^T\mathbf{e}_1\right]^{(k+1)} \right)^2 \right)}{\left((m-k) + \sum_{i=1}^{k} \frac{\lambda_k^2}{\delta_i^2+\lambda_k^2} \right)^2} . \quad (2.23)$$

This is the form of $\widehat{G}(k)$ that we use to determine a stopping iteration in our implementations. The numerator is n/k times the numerator in (2.16) for $G_{A,b}(\lambda_k)$, and the denominator differs only in its first term.

In the ideal situation where the convergence behavior of the Lanczos-hybrid method is perfectly stabilized, we expect λ_k to converge to a fixed value corresponding to an appropriate regularization parameter for the original problem (2.4). In this case the values of $\widehat{G}(k)$ converge to a fixed value. Therefore, we choose to terminate the iterations when these values change

very little; for example,

$$\left| \frac{\widehat{G}(k+1) - \widehat{G}(k)}{\widehat{G}(1)} \right| < \text{tol}\,,$$

for some prescribed tolerance.

However, as remarked in the previous section, it may be impossible to completely stabilize the iterations for realistic problems, resulting in slight semi-convergent behavior of the iterations. In this case, the GCV values $\widehat{G}(k)$ will begin to increase. Thus, we implement a second stopping criterion to stop at iteration k_0 satisfying

$$k_0 = \underset{k}{\operatorname{argmin}}\, \widehat{G}(k)\,.$$

2.5 Numerical Results

In this section we illustrate the effectiveness of using the W-GCV method in Lanczos-hybrid methods with Tikhonov regularization.

2.5.1 Results on Various Test Problems

We implement the adaptive method presented in Section 2.4.4 for choosing ω and provide numerical results for each of the test problems in Section 2.1. The resulting convergence curves are displayed in Figure 2.7.

In all of the test problems, choosing ω adaptively provides nearly optimal convergence behavior. The results for the *Phillips* and *Shaw* problems are excellent with the adaptive W-GCV approach. The *Satellite, Baart* and *Heat* examples exhibit a slowed convergence compared to Tikhonov with the optimal regularization parameter but achieve much better results than with the standard GCV. This slowed convergence is due to the fact that at the early iterations the projected problem is well-conditioned and W-GCV

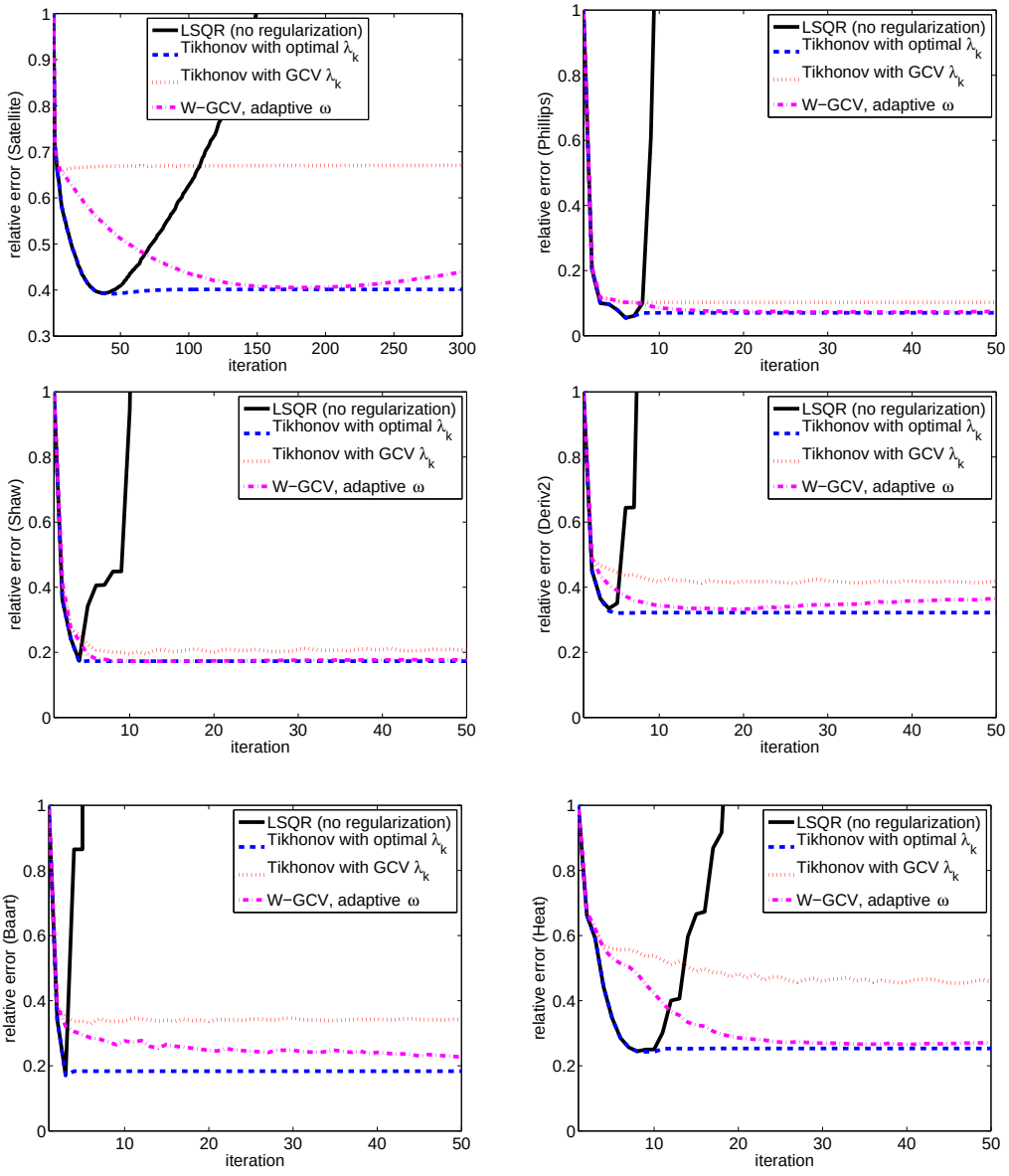


Figure 2.7: Relative errors for adaptive choice of ω. Upper left: Satellite. Upper right: Regtools-Phillips. Middle left: Regtools-Shaw. Middle right: Regtools-Deriv2. Bottom left: Regtools-Baart. Bottom right: Regtools-Heat. The W-GCV method, with our adaptive approach to choose ω, produces near optimal convergence behavior.

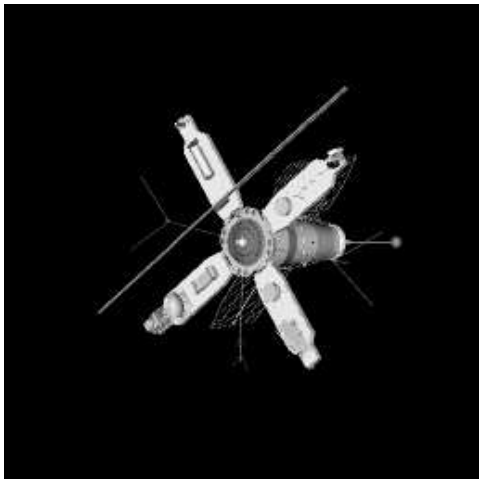
produces a solution that is too smooth. At later iterations, when more small singular value information is captured in the bidiagonalization process, better ω, and hence λ, parameters are found, and the W-GCV parameter choice is close to optimal. In addition, W-GCV avoids the early stagnation behavior that standard GCV exhibits.

It is interesting to note that the *Deriv2* example converges, but eventually exhibits small signs of semi-convergent behavior. Nevertheless, the results are still better than the standard GCV, and, moreover, if combined with the stopping criteria described in Section 2.4.5, the results are quite good. To illustrate, in Table 2.1 we report the iteration at which our code detected a minimum of $\widehat{G}(k)$.

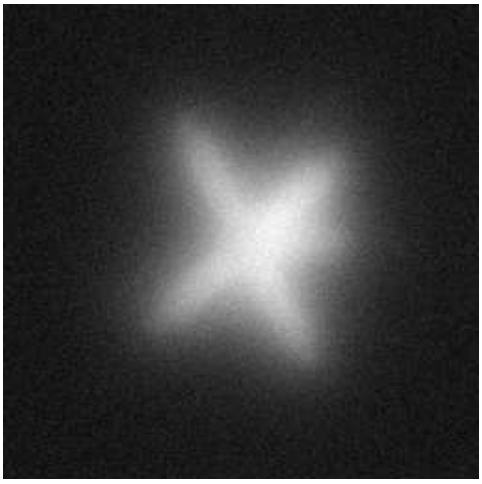
Table 2.1: Results of using $\widehat{G}(k)$ to determine a stopping iteration. The numbers reported in this table are the iteration index at which our Lanczos-hybrid code detected a minimum of $\widehat{G}(k)$.

Problem	*Satellite*	*Phillips*	*Shaw*	*Deriv2*	*Baart*	*Heat*
Stopping Iteration	197	18	23	20	9	21

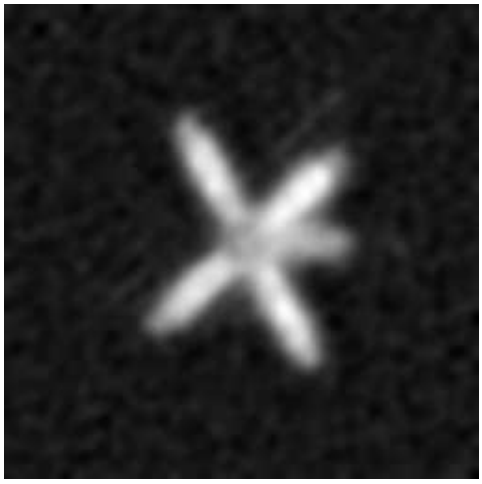
Comparing the results in Table 2.1 with the convergence history plots shown in Figure 2.7, we see that our approach to choosing a stopping iteration is very effective. For illustration purposes, we provide in Figure 2.8 the true and blurred satellite image, followed by the reconstructed image after 197 iterations of the Lanczos-hybrid method using Tikhonov and W-GCV. Although the scheme does not perform as well on the *Baart* example, the results are still quite good considering the difficulty of this problem. (Observe that with no regularization, semi-convergence happens very quickly, and we should therefore expect difficulties in stabilizing the iterations.) These results show that our adaptive W-GCV method performs better than standard GCV,



(a) True Image



(b) Observed Image



(c) Reconstructed Image

Figure 2.8: Satellite image deblurring example. The goal is to reconstruct an approximation of the true image (a), given the blurred and noisy observed image (b). Using the W-GCV approach with Tikhonov in the Lanczos-hybrid scheme, we obtain the reconstructed image (c) at 197 iterations.

and that we are able to determine an appropriate stopping iteration on a wide class of problems.

It should also be noted that when LBD takes many iterations, preconditioning could be used to accelerate convergence. For example, *Satellite* is a much larger problem than the other examples, so more iterations are needed. However, the Lanczos-hybrid method can easily incorporate standard preconditioning techniques to accelerate convergence. For the *Satellite* image deblurring example, we used a Kronecker product based preconditioner [80, 81, 97] implemented in RestoreTools. In this case, the Lanczos-hybrid method, with W-GCV, detects a minimum of $\widehat{G}(k)$ in only 54 iterations. The corresponding solution has relative error 0.4001, which is actually slightly lower than the relative error 0.4061 achieved at iteration 197 when using no preconditioning.

2.5.2 Effect of Noise on ω

We now consider how the choice of ω depends on the amount of noise in the data. In particular, we report on numerical results for the test problems described in Section 2.1 with three different noise levels:

$$\frac{\|\boldsymbol{\varepsilon}\|_2}{\|\mathbf{A}\mathbf{x}_{\text{true}}\|_2} = 0.1, 0.01, \text{ and } 0.001.$$

Thus, these problems have 10%, 1% and 0.1% noise levels respectively. Some of the results reported in previous sections for 10% noise are repeated here for comparison purposes.

Recall that because standard GCV computes regularization parameters that are too large, we should choose $0 < \omega \leq 1$ in W-GCV. Generally we observe that the over-smoothing caused by standard GCV is more pronounced for larger noise levels. Therefore large noise levels typically need smaller values of ω, while small noise levels need larger values of ω. Our next experiments were designed to see how far the "optimal" value of ω differs

from the standard GCV value $\omega = 1$. The results are shown in Table 2.2. We provide the ω values that allow W-GCV to compute near-optimal regularization parameters at each iteration of the Lanczos-hybrid method. For example, in Figure 2.6 we see that for 10% noise, $\omega = 0.2$ produces near optimal convergence behavior for the *Heat* problem; thus, this value appears in the first row, last column of Table 2.2.

Table 2.2: Values of ω for different noise levels. This table contains the values of ω (found experimentally) that produce optimal convergence behavior of the Lanczos-hybrid method for different noise levels. Figure 2.9 shows how these values perform on the *Baart* and *Heat* examples.

	Satellite	*Phillips*	*Shaw*	*Deriv2*	*Baart*	*Heat*
Noise Level	ω_{opt}	ω_{opt}	ω_{opt}	ω_{opt}	ω_{opt}	ω_{opt}
10 %	0.40	0.20	0.05	0.10	0.01	0.20
1%	0.50	0.40	0.05	0.20	0.05	0.40
0.1%	0.80	0.50	0.10	0.60	0.10	0.80

The results reported in Table 2.2 were found experimentally. We see clearly from this table that optimal values of ω depend on the noise level (increasing with decreasing noise level), as well as on the problem. However, more work is needed to better understand these relationships.

Figure 2.9 shows how our adaptive approach to choosing ω compares to the optimal values on two of the test problems (*Baart* and *Heat*) and for three different noise levels. These two test problems are representative of the convergence behavior we observed with the other test problems. We see that if a good choice of ω can be found, W-GCV is very effective (much more so than GCV) at choosing regularization parameters and thus at stabilizing the convergence behavior, especially for high noise levels. Moreover, although we

do not yet have a scheme that chooses the optimal value of ω, these results show that our adaptive approach produces good results on a wide class of problems, and for various noise levels.

2.6 Remarks and Future Directions

We have considered using a weighted-GCV method in Lanczos-hybrid methods for solving large-scale ill-posed problems. The W-GCV method requires choosing yet another parameter, so we proposed and implemented an adaptive, automatic approach for choosing this parameter. We have demonstrated through a variety of test problems that our approach is effective in stabilizing semi-convergence behavior.

Our MATLAB implementation used to generate the results presented in this chapter is named HyBR to represent Hybrid Bidiagonalization Regularization. We briefly describe its usage here. The main code is `HyBR.m` and can be called in the following way:

```
>> [x, output] = HyBR(A, b, P, options);
```

where the required inputs are `A` and `b`, as defined in Section 2.1. The optional input variables are `P`, which is a preconditioner, and `options`, which is a structure that includes input parameters. A parameter structure containing default parameters can be obtained using the `HyBRset.m` code. In particular,

```
>> [options] = HyBRset('HyBR');
```

An electronic copy of the MATLAB codes can be obtained from the following website: `http://www.mathcs.emory.edu/software/HyBR`.

Several open questions remain. With the ability to obtain near optimal solutions, Lanczos-hybrid methods should have a significant impact on many applications. Recently, Kilmer, Hansen and Español [87] suggested a projection-based algorithm that can be implemented for more general regularization operators. We can treat this iterative method as a hybrid method

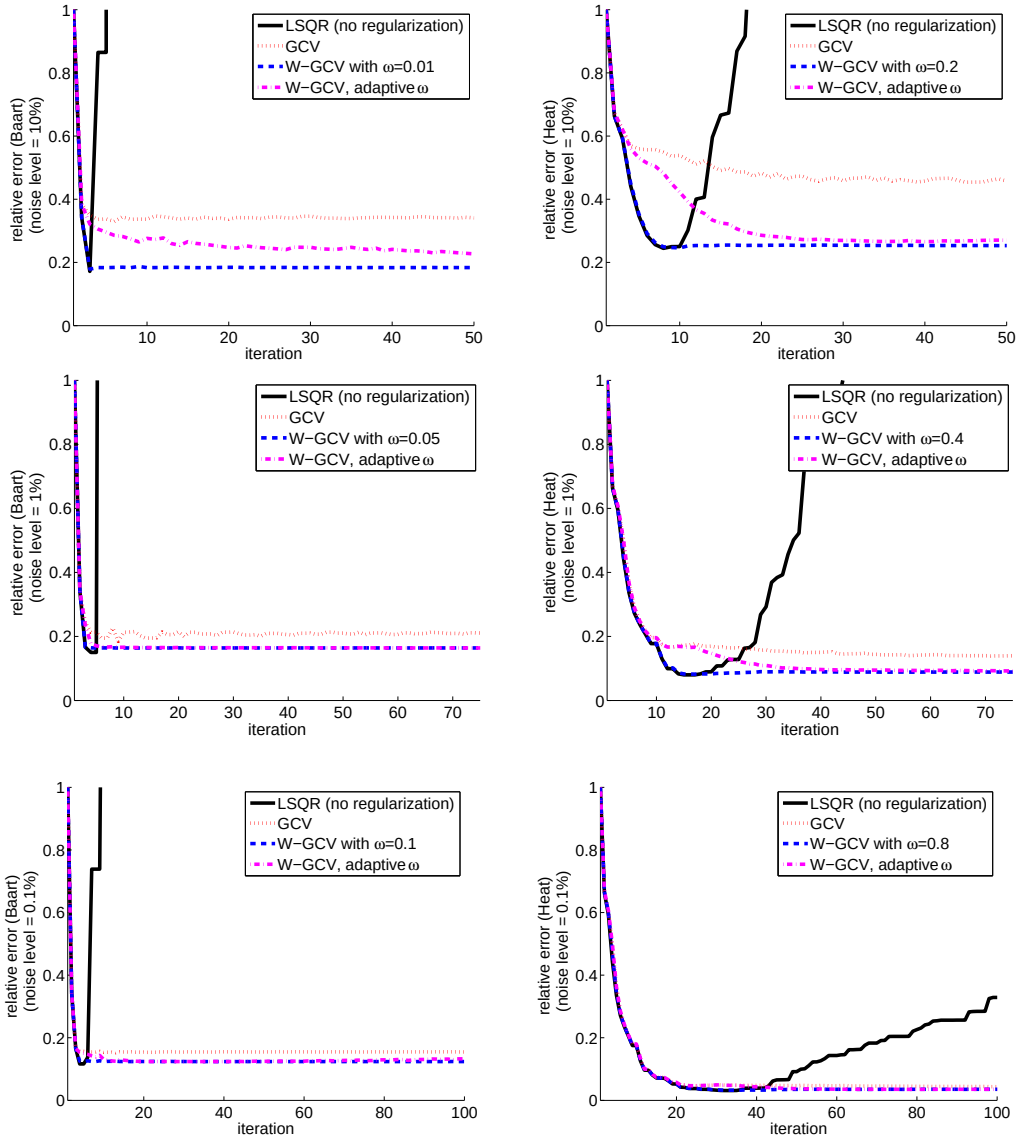


Figure 2.9: Relative errors for different noise levels. The plots on the left correspond to the Baart example, and the plots on the right correspond to the Heat example. Optimal choices of ω (found experimentally) produce optimal convergence behavior, and our adaptive approach to choose ω, produces near optimal convergence behavior. It can be observed that standard GCV is ineffective for moderate to high levels of noise.

and apply W-GCV. In addition, we would like to see how well W-GCV works in combination with other filtering methods and Lanczos-hybrid methods. Finally, work remains to be done on developing alternative ways to determine the new parameter in the W-GCV method, as well as providing a statistical justification for the new parameter [60, 105].

Chapter 3

Separable Nonlinear Least Squares Problems

Nonlinear least squares problems arise in a variety of applications and can be very difficult to solve. Oftentimes structural properties such as separability can be exploited to simplify the problem. In this chapter we consider problems that have the following form:

$$\min_{\mathbf{x},\mathbf{y}} ||\mathbf{A}(\mathbf{y})\mathbf{x} - \mathbf{b}||_2^2 , \tag{3.1}$$

and we investigate optimization approaches that can take advantage of separability to efficiently estimate the unknowns $\mathbf{x}$ and $\mathbf{y}$.

It is obvious to see that if the parameters in $\mathbf{y}$ are known, then we obtain the linear problem (2.1). However, we consider the case where the parameters in $\mathbf{y}$ are unknown; therefore, both sets of unknowns must be evaluated.

Since we are dealing with ill-posed problems, we consider the Tikhonov formulation for regularizing the unknowns in $\mathbf{x}$:

$$\min_{\mathbf{x},\mathbf{y}}\{\|\mathbf{A}(\mathbf{y})\mathbf{x} - \mathbf{b}\|_2^2 + \lambda^2\|\mathbf{x}\|_2^2\} \quad \Leftrightarrow \quad \min_{\mathbf{x},\mathbf{y}} \left\| \begin{bmatrix} \mathbf{A}(\mathbf{y}) \\ \lambda\mathbf{I} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2 . \tag{3.2}$$

To regularize the parameters in $\mathbf{y}$, we assume a reduced parameter space method. That is, we assume the parameters defining the forward operation can be spanned by a small set of known vectors. In the next section we present two imaging applications that fit this model.

3.1 Motivating Examples

Separable nonlinear least squares problems arise naturally in super-resolution imaging and blind deconvolution applications. In this section we provide a summary of previous work and a complete description of the mathematical model for each of these applications.

3.1.1 Super-Resolution Imaging

In many imaging applications, it is desirable to have images with high spatial resolution. One approach to obtain such images is to build sophisticated instrumentation having intrinsically high resolution capabilities. In addition to being costly, other limitations are difficult to overcome. For example, reducing the pixel sensor size decreases the signal to noise ratio and also results in a build up of shot noise [112]. An alternative, less expensive approach that has gained popularity in digital imaging and video applications is to use mathematical software tools to combine the information given by a set of low resolution images into one high resolution image. This is a process commonly referred to as super-resolution imaging [82].

In order for super-resolution techniques to work, the multiple low resolution images must contain different information of the same scene. This is typically accomplished by capturing low resolution images of slightly shifted versions of the (same) scene, with the shifts occurring at sub-pixel distances. For efficient implementation of the reconstruction algorithms, it is often assumed that the shifts are uniform and linear. Another approach that has been proposed is to use low resolution images of a stationary scene, but where each image has different amounts of defocus [35]. The idea of achieving super-resolution dates back to the early 1970s [3], but most substantial work on algorithms has been done more recently; see for example [13, 19, 35, 101, 103, 124]. Recent overview papers on super-resolution include [42, 112].

Algorithms for super-resolution involve two key steps: registration and reconstruction. That is, first the shifts, i.e. the relative displacement or deformation of each point in each image from each point of a reference image, must be estimated. Second, after the displacements have been evaluated, a linear inverse problem must be solved to obtain the high resolution image. Most approaches proposed in the literature decouple these steps. Decoupling makes sense if relative displacements are known *a priori* (possibly from a calibration process) or if they can be estimated from the low resolution images [13, 35, 42, 101, 103, 112]. For simple displacements, such as linear uniform translation, this procedure can work well. However, for more complex, nonlinear, nonuniform transformations, accurately estimating the displacements on the coarse image can be very difficult. In particular, because the estimation of displacements is done on the coarse image, fine-scale details of the high resolution image are ignored. This can lead to severe inaccuracies in the displacements, thus resulting in degradation of the reconstructed image.

Since the registration and reconstruction parts of the problem are not independent, it is possible to obtain better results by considering a coupled approach that jointly estimates the displacements and the reconstructed high resolution image. Although substantial work has been done in the development of algorithms for the uncoupled super-resolution problem, relatively little work has been done for the coupled problem, which requires solving a nonlinear optimization problem. This can be very expensive to implement unless one makes other simplifying assumptions; as pointed out in [42], several difficulties related to the joint estimation task still remain largely open. Tom and Katsaggelos [124] use a maximum likelihood formulation and an expectation maximization (EM) algorithm to solve the joint estimation problem. They implement the algorithm in the frequency domain, implying spatial invariance and, hence, linear uniform displacements. Cheeseman et al. [19] consider the maximum *a posteriori* (MAP) framework, using the simplex method to

compute registration parameters and the Jacobi algorithm to solve a system of linear equations. Hardie, Barnard, and Armstrong [70] also use a MAP framework. They consider only simple horizontal and vertical displacements and a numerical optimization approach that alternates between the two sets of variables.

In this work, we formulate the super-resolution problem as a separable nonlinear least squares problem and consider a new mathematical framework that enables us to couple the problem of estimating the displacements with the problem of estimating the high resolution image. First we develop a mathematical model for super-resolution imaging.

Mathematical Framework

Suppose we measure m low resolution images, $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m$, which could be obtained either simultaneously from multiple sensors targeted at the same object, or from a single sensor that captures images of the same object at multiple time instances. In either case, it is assumed that each low resolution image is shifted by subpixel displacements from a particular reference image. Assuming these low resolution images are undersampled, these subpixel displacements suggest that each low resolution image contains different information about the same object. Each low resolution image can be represented as

$$\mathbf{b}_i = \mathbf{R}\,\mathbf{S}_i\,\mathbf{x}_{\text{true}} + \boldsymbol{\varepsilon}_i\,, \tag{3.3}$$

where $\boldsymbol{\varepsilon}_i$ is additive noise, $\mathbf{R}$ is a restriction or decimation matrix that transforms a high resolution image into a low resolution image, and $\mathbf{S}_i$ is a sparse matrix that performs a geometric distortion (e.g., shift, rotation, etc.) of the high resolution image, $\mathbf{x}_{\text{true}}$. The geometric distortion, and hence $\mathbf{S}_i$, is defined by the parameter vector, $\mathbf{y}_i$. (To simplify this discussion we assume no blurring in the recorded images, but such effects can be easily incorporated into the model.)

Note that if we assume each low resolution image is only shifted horizontally and vertically, then each $\mathbf{y}_i$ contains only two values (the horizontal and vertical displacements). If we want to consider more complicated movement (such as rotation), then each $\mathbf{y}_i$ might contain up to six values that define, for example, general linear affine transformations [34]. This is the approach we follow.

Consider the deformed (shifted, rotated, scaled) image, $\mathbf{S}_i\mathbf{x}_{\text{true}}$, where $\mathbf{S}_i$ is a sparse $n\times n$ matrix and the nonzero elements of $\mathbf{S}_i$ are interpolation weights. To see how the weights used in the bilinear interpolation are calculated, define coordinate vectors for a discrete image with n pixels as

$$\mathbf{t}_1 = \begin{bmatrix} t_{1,1} \\ t_{2,1} \\ \vdots \\ t_{n,1} \end{bmatrix} \quad \text{and} \quad \mathbf{t}_2 = \begin{bmatrix} t_{1,2} \\ t_{2,2} \\ \vdots \\ t_{n,2} \end{bmatrix}.$$

That is, pixel j is centered at position $t_j = (t_{j,1}, t_{j,2})$ on a coordinate grid. Then, the deformation field at pixel j has the form:

$$u_i(t_j) = \begin{pmatrix} u_{i,1} \\ u_{i,2} \end{pmatrix} = \begin{pmatrix} y_i^{(1)} & y_i^{(2)} \\ y_i^{(4)} & y_i^{(5)} \end{pmatrix} \begin{pmatrix} t_{j,1} \\ t_{j,2} \end{pmatrix} + \begin{pmatrix} y_i^{(3)} \\ y_i^{(6)} \end{pmatrix}. \tag{3.4}$$

Since the parameters $y_i^{(1)}, y_i^{(2)}, \ldots, y_i^{(6)}$ are shared by all pixels in the i^{th} deformed image, the set of displacement vectors for image i can be written as

$$\mathbf{u}_i = \begin{bmatrix} \mathbf{u}_{i,1} \\ \mathbf{u}_{i,2} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_1 & \mathbf{t}_2 & \mathbf{1} & & & \\ & & & \mathbf{t}_1 & \mathbf{t}_2 & \mathbf{1} \end{bmatrix} \begin{bmatrix} y_i^{(1)} \\ \vdots \\ y_i^{(6)} \end{bmatrix} = \mathbf{D}\mathbf{y}_i\,, \tag{3.5}$$

where $\mathbf{1}$ is the vector of all ones. Since the displacements $\mathbf{u}_i$ are a function of the registration parameters $\mathbf{y}_i$, let $\mathbf{S}_i \equiv \mathbf{S}(\mathbf{u}_i) = \mathbf{S}(\mathbf{D}\mathbf{y}_i)$. This notation will aid in future derivations.

Furthermore, given $\mathbf{u}_i$, we can obtain the nonzero elements of $\mathbf{S}_i$ in the following way. That is, we would like to connect the displaced pixel, $x(t_j + u_i) = x(t_j + u_i(t_j))$, to four pixel values in the reference image that surround it. Because subpixel shifts allow us to recover more information about the object, it would be undesirable for a displaced pixel to fall precisely on one of the pixels of the reference image. To see how the weights used in the bilinear interpolation are calculated, suppose x^{NE}, x^{NW}, x^{SE}, x^{SW} are the four pixel values that surround $x(t_j + u_i)$, as illustrated in Figure 3.1. Then, assuming

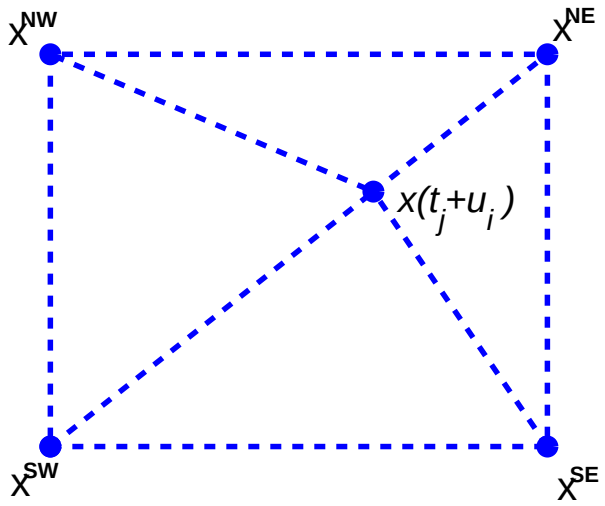


Figure 3.1: An illustration of bilinear interpolation. Here the corners x^{NW}, x^{NE}, x^{SW}, and x^{SE} represent given discrete pixel values, and $x(t_j + u_i)$ is a value that must be approximated.

without loss of generality a pixel size of 1×1, the interpolated point can be written as

$$\begin{aligned} x(t_j + u_i) &= (1 - u_{i,1})(1 - u_{i,2})x^{NW} + u_{i,1}(1 - u_{i,2})x^{NE} + \\ &\quad (1 - u_{i,1})u_{i,2}x^{SW} + u_{i,1}u_{i,2}x^{SE}. \end{aligned} \tag{3.6}$$

The bilinear products of $u_{i,1}$ and $u_{i,2}$ are precisely the interpolation weights found in matrix $\mathbf{S}_i$.

Recall that the aim of super-resolution image reconstruction is to fuse the different image information from the low resolution images in order to create one high resolution image. The inverse problem can then be modeled as the nonlinear least squares problem (3.1), where

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_m \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_m \end{bmatrix}, \quad \mathbf{A}(\mathbf{y}) = \begin{bmatrix} \mathbf{R}\,\mathbf{S}(\mathbf{D}\mathbf{y}_1) \\ \mathbf{R}\,\mathbf{S}(\mathbf{D}\mathbf{y}_2) \\ \vdots \\ \mathbf{R}\,\mathbf{S}(\mathbf{D}\mathbf{y}_m) \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \boldsymbol{\varepsilon}_1 \\ \boldsymbol{\varepsilon}_2 \\ \vdots \\ \boldsymbol{\varepsilon}_m \end{bmatrix}.$$

By only allowing rotation, translation, scaling and shear, we have assumed a parametric approach to registration, in that the displacements can be spanned by a small set of known vectors. This in turn regularizes the registration process. It is important to note that the number of parameters defining $\mathbf{y}$ is significantly fewer than the number of pixel values defining $\mathbf{x}$.

3.1.2 Blind Deconvolution

Another example of a separable nonlinear least squares problem (3.1) arises in image deblurring, when the blurring operator is not known exactly. This problem is often referred to as blind deconvolution in the image processing literature [18, 71, 76, 120, 130]. It is assumed that the observed image $\mathbf{b}$ is data measured by an imaging device (such as a camera, telescope, microscope, or medical imaging scanner), and $\mathbf{A}(\mathbf{y})$ is an operator that models how the image is captured. If the true parameters $\mathbf{y}_{\text{true}}$ were available, the problem reduces to the linear image deblurring problem discussed in Chapter 2. However, realistically the vector $\mathbf{y}$ is obtained through a calibration process, for example, by collecting images of known objects. Thus, it is only an approximation of the true parameters $\mathbf{y}_{\text{true}}$.

Matrix $\mathbf{A}(\mathbf{y})$ models the blurring operation and can be written as

$$\mathbf{A}(\mathbf{y}) = \mathbf{A}(\mathbf{P}(\mathbf{y}))\,,$$

where $\mathbf{P}(\mathbf{y})$ is a PSF. In many applications the blur is assumed to be spatially invariant, which means $\mathbf{P}(\mathbf{y})$ is an image of a point source object and $\mathbf{A}(\mathbf{P}(\mathbf{y}))$ is structured. The precise structure depends on the imposed boundary conditions, but it is usually a combination of Toeplitz and Hankel matrices; see [69] for more details.

In any blind deconvolution problem it is necessary to make some assumptions about the blur. For illustration purposes, we assume a general Gaussian blur, where the ij^{th} entry of the PSF, centered at (k,l), has the form:

$$\begin{aligned} p^{(ij)} &= \exp\left(-\frac{1}{2}\begin{bmatrix} i-k \\ j-l \end{bmatrix}^T \begin{bmatrix} s_1^2 & r^2 \\ r^2 & s_2^2 \end{bmatrix}^{-1} \begin{bmatrix} i-k \\ j-l \end{bmatrix}\right) \\ &= \exp\left(\frac{-(i-k)^2 s_2^2 - (j-l)^2 s_1^2 + 2(i-k)(j-l)r^2}{2s_1^2 s_2^2 - 2r^4}\right). \end{aligned} \tag{3.7}$$

Since general blind deconvolution is a highly underdetermined problem, there is no unique solution. To alleviate this concern, we assume that the blurring operation follows a certain parametric form. In particular, the parameters s_1, s_2 and r determine the spread and orientation of the Gaussian PSF, and oftentimes a scaling factor is introduced so that the PSF entries sum to 1. To match our previous notation, let

$$\mathbf{y} = \begin{bmatrix} s_1 \\ s_2 \\ r \end{bmatrix},$$

and let $\mathbf{A}(\mathbf{y})$ be the matrix defined by the Gaussian PSF with parameters $\mathbf{y}$. Notice that similar to the super-resolution imaging example, we have reduced the number of unknowns in $\mathbf{y}$ as a means of regularization. However, we still

need to regularize the unknowns in the image vector $\mathbf{x}$. Thus, the goal is to jointly estimate the image $\mathbf{x}$ and blur parameters $\mathbf{y}$ by solving (3.2). In the next section we describe some optimization approaches.

3.2 Solution through Optimization

Both of the applications described in the previous section require the solution of (3.2). In this section we consider a general nonlinear optimization scheme for simultaneous update of $\mathbf{x}$ and $\mathbf{y}$. A Newton or Gauss-Newton approach can be quite difficult since it requires computation or approximation of the Hessian matrix, which is typically very large. Furthermore, another concern is the selection or estimation of a good regularization parameter.

To reduce the computational effort, we discuss a variable projection approach to take advantage of certain properties of the problem [51, 52, 85, 109, 117]. The basic idea is to exploit the following properties: the variables $\mathbf{x}$ and $\mathbf{y}$ separate, the problem is linear in $\mathbf{x}$, and there are significantly fewer parameters defining $\mathbf{y}$ than there are defining $\mathbf{x}$. More specifically, we mathematically eliminate one set of parameters to obtain a reduced cost functional, and a Gauss-Newton method is used for the reduced problem. This approach is a slight modification of the methods proposed by Golub and Pereyra [52] for separable nonlinear least squares problems and is similar to the approach used by Vogel, Chan and Plemmons [130] for phase diversity blind deconvolution. We are particularly interested in making the methods feasible for large-scale problems and incorporating a slick implementation for selecting good regularization parameters.

3.2.1 General Gauss-Newton Approach

To simplify notation, we can write the nonlinear least squares problem given in equation (3.2) as

$$\min_{\mathbf{z}} \psi(\mathbf{z}) = \min_{\mathbf{z}} \|\mathbf{f}(\mathbf{z})\|_2^2 \,, \tag{3.8}$$

where $\mathbf{z}^T = [\ \mathbf{x}^T \ \ \mathbf{y}^T\ ]$ and

$$\mathbf{f}(\mathbf{z}) = \mathbf{f}(\mathbf{x},\mathbf{y}) = \begin{bmatrix} \mathbf{A}(\mathbf{y}) \\ \lambda \mathbf{I} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} .$$

The nonlinear least squares problem (3.8) can be solved using a Gauss-Newton method [31, 86, 104, 108], which is an iterative algorithm that computes

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \mathbf{d}_k \,, \quad k = 0, 1, 2, \ldots$$

where

$$\mathbf{d}_k = -(\widehat{\psi}''(\mathbf{z}_k))^{-1} \psi'(\mathbf{z}_k) \,,$$

$\mathbf{z}_0$ is an initial guess, and $\widehat{\psi}''$ is an approximation of the Hessian ψ''. It is not difficult to show that $\psi' = \mathbf{J}_\psi^T \mathbf{f}$, and an approximation of ψ'' is given by $\widehat{\psi}'' = \mathbf{J}_\psi^T \mathbf{J}_\psi$, where $\mathbf{J}_\psi$ is the Jacobian matrix; for our problem it can be written as

$$\mathbf{J}_\psi = \begin{bmatrix} \mathbf{f}_x & \mathbf{f}_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial \mathbf{f}(\mathbf{x},\mathbf{y})}{\partial \mathbf{x}} & \dfrac{\partial \mathbf{f}(\mathbf{x},\mathbf{y})}{\partial \mathbf{y}} \end{bmatrix} .$$

If we define $\mathbf{r} = -\mathbf{f} = \mathbf{b} - \mathbf{A}(\mathbf{y})\,\mathbf{x}$, then the computation to update the search direction $\mathbf{d}_k$ at each Gauss-Newton iteration is equivalent to solving a least squares problem of the form:

$$\min_{\mathbf{d}} \|\mathbf{J}_\psi \mathbf{d} - \mathbf{r}\|_2^2 \,. \tag{3.9}$$

To summarize, then, a Gauss-Newton method applied to (3.8) has the basic form:

General Gauss-Newton Algorithm

choose initial $\mathbf{z}_0 = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{y}_0 \end{bmatrix}$

for $k = 0, 1, 2, \ldots$

$\mathbf{r}_k = \mathbf{b} - \mathbf{A}(\mathbf{y}_k)\,\mathbf{x}_k$

$\mathbf{d}_k = \underset{\mathbf{d}}{\operatorname{argmin}} \|\mathbf{J}_\psi \mathbf{d} - \mathbf{r}_k\|_2$

$\mathbf{z}_{k+1} = \mathbf{z}_k + \mathbf{d}_k$

end

This general Gauss-Newton approach can work well, but constructing and solving linear systems with $\mathbf{J}_\psi$ can be very expensive. Effective preconditioners for (3.9) may be difficult to find. Moreover, it requires either specifying *a priori* the regularization parameter λ or estimating it within a nonlinear iterative scheme; see [61] and the references therein. A further difficulty when using a fully coupled approach is that we do not take algorithmic advantage of the fact that the problem is strongly convex in $\mathbf{x}$. Thus, we may take very small steps due to the nonlinearity induced by $\mathbf{y}$. Instead, we would like to adapt the variable projection method for separable nonlinear least squares problems to work for large-scale ill-posed inverse problems and develop an approach where the regularization parameter can be estimated by the algorithm.

3.2.2 Variable Projection Method

The variable projection method can be used to solve the nonlinear least squares problem (3.2). The approach exploits the fact that $\psi(\mathbf{x}, \mathbf{y})$ is linear

in $\mathbf{x}$ and takes advantage of the relatively fewer parameters contained in $\mathbf{y}$, compared to $\mathbf{x}$. However, rather than explicitly separating variables $\mathbf{x}$ and $\mathbf{y}$ as in coordinate descent, the variable projection approach mathematically eliminates the linear parameters $\mathbf{x}$, thus obtaining a reduced cost functional that depends only on $\mathbf{y}$. A Gauss-Newton method is then applied to the reduced cost functional. Specifically, consider

$$\varphi(\mathbf{y}) \equiv \psi(\mathbf{x}(\mathbf{y}), \mathbf{y}) \,, \tag{3.10}$$

where $\mathbf{x}(\mathbf{y})$ is a solution of

$$\min_{\mathbf{x}} \psi(\mathbf{x}, \mathbf{y}) = \min_{\mathbf{x}} \left\| \begin{bmatrix} \mathbf{A}(\mathbf{y}) \\ \lambda \mathbf{I} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2 . \tag{3.11}$$

Using notation from Chapter 2, the solution of (3.11) can be written as $\mathbf{x}(\mathbf{y}) = \mathbf{A}_\lambda^\dagger(\mathbf{y})\mathbf{b}$. Then the residual norm with linear parameters eliminated can be represented as

$$\begin{aligned} ||\mathbf{b} - \mathbf{A}(\mathbf{y})\mathbf{x}(\mathbf{y})|| &= ||\mathbf{b} - \mathbf{A}(\mathbf{y})\mathbf{A}_\lambda^\dagger(\mathbf{y})\mathbf{b}|| \\ &= ||(\mathbf{I} - \mathbf{A}(\mathbf{y})\mathbf{A}_\lambda^\dagger(\mathbf{y}))\mathbf{b}|| \,. \end{aligned}$$

The origin of the name "variable projection" comes from the fact that the matrix $\mathbf{I} - \mathbf{A}(\mathbf{y})\mathbf{A}_\lambda^\dagger(\mathbf{y})$ is the projector onto the orthogonal complement of the column space of $\mathbf{A}(\mathbf{y})$ [52]. We say that the residual is the variable projection of $\mathbf{b}$.

Now to use the Gauss-Newton algorithm to minimize the reduced cost functional $\varphi(\mathbf{y})$, we need to compute $\varphi'(\mathbf{y})$. Note that because $\mathbf{x}$ solves (2.4), it follows that $\psi_{\mathbf{x}} = 0$, and thus

$$\varphi'(\mathbf{y}) = \frac{d\mathbf{x}}{d\mathbf{y}} \psi_{\mathbf{x}} + \psi_{\mathbf{y}} = \psi_{\mathbf{y}} = \mathbf{f}_y^T \mathbf{f} \,.$$

Although $\frac{d\mathbf{x}}{d\mathbf{y}}$ does not need to be computed, the Jacobian of the reduced cost functional,

$$\mathbf{J}_\varphi = \mathbf{f}_y = \frac{\partial[\mathbf{A}(\mathbf{y})\mathbf{x}]}{\partial \mathbf{y}} \,, \tag{3.12}$$

must be evaluated analytically or approximated. Computing $\mathbf{J}_\varphi$ may be nontrivial, but it is often much more tractable than constructing $\mathbf{J}_\psi$.

For completeness, a Gauss-Newton method applied to the reduced cost functional has the basic form:

Reduced Gauss-Newton Algorithm

choose initial $\mathbf{y}_0$

for $k = 0, 1, 2, \ldots$

$$\mathbf{x}_k = \underset{\mathbf{x}}{\operatorname{argmin}} \left\| \begin{bmatrix} \mathbf{A}(\mathbf{y}_k) \\ \lambda \mathbf{I} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2$$

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A}(\mathbf{y}_k)\,\mathbf{x}_k$$

$$\mathbf{d}_k = \underset{\mathbf{d}}{\operatorname{argmin}} \|\mathbf{J}_\varphi \mathbf{d} - \mathbf{r}_k\|_2$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \mathbf{d}_k$$

end

A major advantage of using the variable projection method for large-scale inverse problems is that we can use the hybrid approach discussed in Section 2.3 to simultaneously estimate an appropriate regularization parameter and compute $\mathbf{x}_k$ at each iteration. That is, in the above algorithm we replace the statement for $\mathbf{x}_k$ with

$$\mathbf{x}_k = \text{HyBR}(\mathbf{A}(\mathbf{y}_k), \mathbf{b})$$

and notice that the regularization parameter may now change at each Gauss-Newton iteration. That is, we select λ_k based on the problem at hand.

We conclude this section with a few remarks on computational issues. First, it may be necessary to include a line search in the Gauss-Newton method [86], such as an Armijo rule to ensure sufficient decrease of the objective function.

However, care must be taken to maintain consistency when implementing a line search strategy because the regularization parameter, and hence the objective function, is changing at each Gauss-Newton iteration [61].

Issues in constructing the Jacobian matrix (3.12) are addressed in Section 3.2.3. In regards to computing the update step, $\mathbf{d}_k$, the least squares problem involving the Jacobian is generally not very difficult to solve, at least for the applications we have considered where the number of parameters in $\mathbf{y}$ is significantly less than the number of parameters in $\mathbf{x}$. In this case, $\mathbf{J}_\varphi$ has only a few columns (corresponding to the number of parameters in $\mathbf{y}$) and is generally well-conditioned.

3.2.3 Jacobian Construction

It is not necessary to have an analytical expression for $\mathbf{J}_\varphi$, since a finite difference approach can be used to numerically approximate the Jacobian. However, with a reduced parameter space formulation for both the super-resolution and blind deconvolution examples, deriving the Jacobian is not too difficult. Thus, for completeness, we provide a mathematical derivation in this section.

Super-Resolution

Computing the Jacobian for the super-resolution application described in Section 3.1.1 requires us to take derivatives with respect to the registration parameters, $\mathbf{y}$. For ease of derivation, consider the derivative associated with the parameters for the i^{th} low resolution image:

$$\frac{\partial[\mathbf{RS}(\mathbf{D}\mathbf{y}_i)\mathbf{x}]}{\partial \mathbf{y}_i} = \frac{\partial[\mathbf{RS}(\mathbf{u}_i)\mathbf{x}]}{\partial \mathbf{y}_i}.$$

First notice that using the interpolation formula (3.6), we have

$$\frac{\partial x(t_j + u_i)}{\partial u_{i,1}} = (1 - u_{i,2})(x^{NE} - x^{NW}) + u_{i,2}(x^{SE} - x^{SW}) \tag{3.13a}$$

and

$$\frac{\partial x(t_j + u_i)}{\partial u_{i,2}} = (1 - u_{i,1})(x^{SW} - x^{NW}) + u_{i,1}(x^{SE} - x^{NE})\,. \tag{3.13b}$$

Observe that the expressions given in (3.13) are equivalent to a simple discretization of the gradient of the image, assuming it is a piecewise linear function. We therefore define the Jacobian of the image with respect to the displacements $u_i(t)$ as

$$\mathbf{G}_i \equiv \mathbf{G}(u_i) \equiv \frac{\partial[\mathbf{S}_i\mathbf{x}]}{\partial u_i}\,. \tag{3.14}$$

Then using the chain rule, we can obtain an expression for the Jacobian with respect to $\mathbf{y}_i$,

$$\mathbf{J}_i(\mathbf{y}) = \frac{\partial[\mathbf{RS}(\mathbf{D}\mathbf{y}_i)\mathbf{x}]}{\partial \mathbf{y}_i} = \mathbf{R}\mathbf{G}_i\mathbf{D}\,, \tag{3.15}$$

and the partial derivative of φ with respect to $\mathbf{y}_i$,

$$\frac{\partial \varphi}{\partial \mathbf{y}_i} = \mathbf{D}^T\mathbf{G}_i^T\mathbf{R}^T(\mathbf{R}\mathbf{S}_i\mathbf{x} - \mathbf{b}_i)\,. \tag{3.16}$$

To conclude, the desired Jacobian for the reduced Gauss-Newton approach is given by

$$\mathbf{J}_\varphi = \begin{bmatrix} \mathbf{J}_1(\mathbf{y}) & & \\ & \ddots & \\ & & \mathbf{J}_m(\mathbf{y}) \end{bmatrix}, \tag{3.17}$$

where $\mathbf{J}_i(\mathbf{y})$ is defined in (3.15).

Blind Deconvolution

The Jacobian of the reduced cost functional for blind deconvolution can also be computed using the chain rule in the following way:

$$\begin{aligned} \mathbf{J}_\varphi &= \frac{\partial [\, \mathbf{A}(\, \mathbf{P}(\mathbf{y})\,)\,\mathbf{x}\,]}{\partial \mathbf{y}} \\ &= \frac{\partial [\, \mathbf{A}(\, \mathbf{P}(\mathbf{y})\,)\,\mathbf{x}\,]}{\partial \mathbf{P}} \cdot \frac{\partial [\, \mathbf{P}(\mathbf{y})\,]}{\partial \mathbf{y}} \\ &= \mathbf{A}(\mathbf{X}) \cdot \frac{\partial [\, \mathbf{P}(\mathbf{y})\,]}{\partial \mathbf{y}}\,, \end{aligned} \tag{3.18}$$

where $\mathbf{x} = \mathrm{vec}(\mathbf{X})$, i.e. $\mathbf{x}$ is the vector obtained by stacking columns of matrix $\mathbf{X}$. The last equality is due to the special structure of a spatially invariant blurring operator and the commutative property of convolution. For our particular example, where we assume Gaussian blur with PSF (3.7), it is not too difficult to get an analytical expression for $\frac{\partial [\, \mathbf{P}(\mathbf{y})\,]}{\partial \mathbf{y}}$. However, we remark that disregarding scaling factors that depend on the blur parameters can lead to erroneous derivative calculations, and careful calculation is needed to compute the correct values.

For efficient implementation, we use the function `psfMatrix` from RestoreTools (c.f. Section 2.1) to construct $\mathbf{A}(\mathbf{P}(\mathbf{y}))$ and $\mathbf{A}(\mathbf{X})$. The associated routines were then used for efficiently computing matrix-vector multiplications. The object oriented approach with operator overloading used in RestoreTools makes it very easy to perform these operations.

3.3 Numerical Results

In this section we demonstrate that using HyBR to solve the regularized least squares problem at each Gauss-Newton iteration of the variable projection method can be beneficial to super-resolution and blind deconvolution imaging applications. More specifically, we show that one can achieve sufficient

objective function and gradient norm decrease, as well as more accurate parameter estimation, by using the HyBR method in a reduced Gauss-Newton framework. Furthermore, we illustrate that sufficient reconstructions can be computed without requiring *a priori* selection of a regularization parameter.

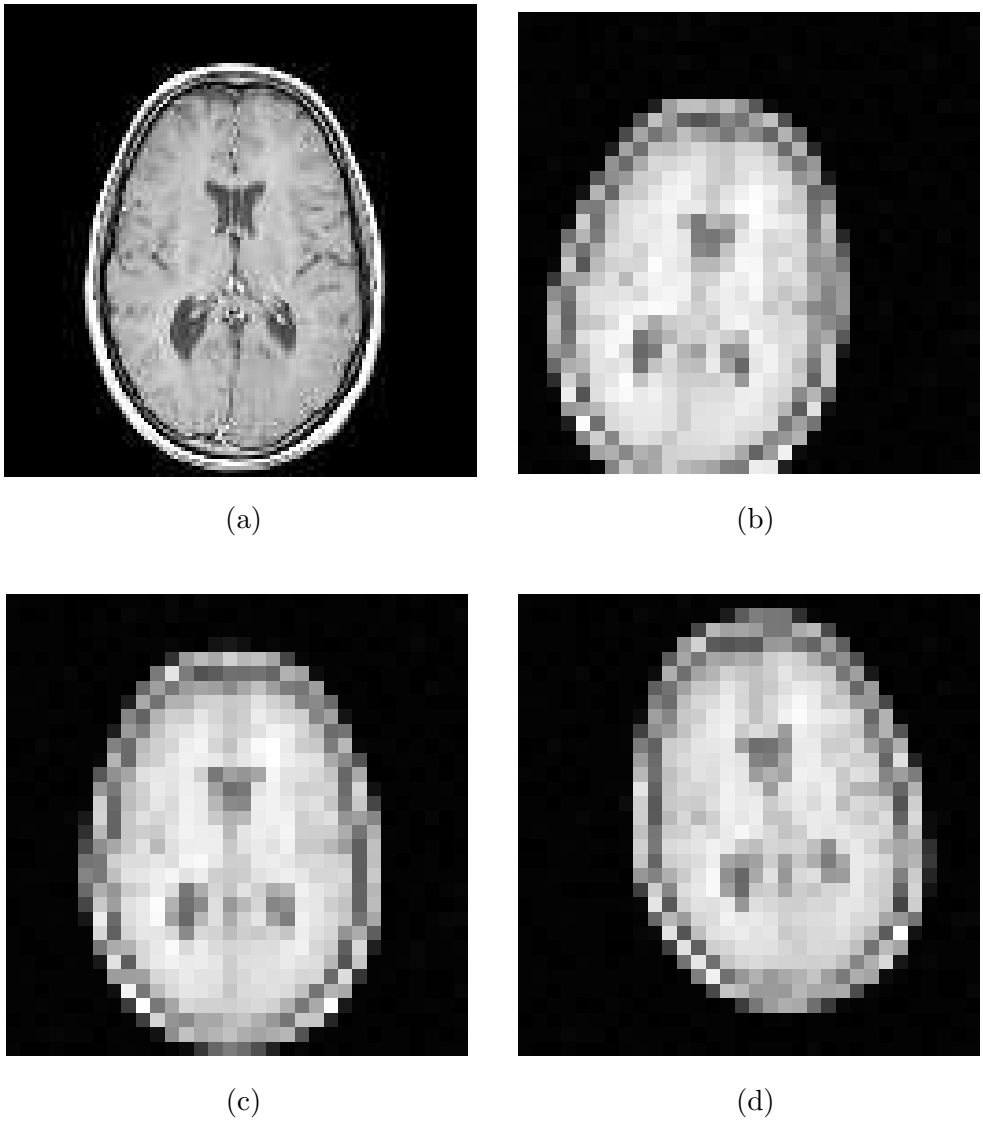
Super-Resolution

For numerical tests reported for the super-resolution imaging example, we use a magnetic resonance (MR) image, which is available in MATLAB. The original high resolution image with 128^2 pixels, together with three low resolution images of 32^2 pixels, is shown in Figure 3.2.

The low resolution images were generated using a sequence of rotations and translations of the original image, followed by a decimation with averaging operator. Then 1% random noise was added to each low resolution image. More specifically, each low resolution image was generated as in equation (3.3), where the noise vector $\boldsymbol{\varepsilon}_i$ consists of pseudo-random values drawn from a normal distribution with mean zero and standard deviation one, scaled such that $||\boldsymbol{\varepsilon}_i||_2/||\mathbf{R}\,\mathbf{S}_i\mathbf{x}_{\text{true}}||_2 = 0.01$.

We consider solving problem (3.2) using the reduced Gauss-Newton algorithm with regularization parameters of $\lambda = 0.01$, $\lambda = 0.1$, and $\lambda = 0.45$. The motivation for choosing $\lambda = 0.1$ was that it seemed to produce adequate results in numerical experiments. However, we also present results for regularization parameters of 0.01 and 0.45 to illustrate the potential disadvantages of an inaccurate regularization parameter. An initial guess, $\mathbf{y}_0$, was found by minimizing the displacements of each of the coarse images from a reference image.

To evaluate and compare algorithms, we examine a variety of measures at each iteration. These include the relative objective function value, the relative gradient value, and the relative error of the rotation and translation



(a) (b)

(c) (d)

Figure 3.2: Super-resolution example. The high resolution image is shown in (a), and three selected low resolution images are shown in (b-d).

parameters. More precisely, this last measure is computed as

$$\Delta \mathbf{y} = \frac{\|\mathbf{y}_k - \mathbf{y}_{\text{true}}\|}{\|\mathbf{y}_{\text{true}}\|}, \tag{3.19}$$

where $\mathbf{y}_{\text{true}}$ are the true parameters and $\mathbf{y}_k$ are the parameter approximations at the k^{th} iteration. The results for the reduced Gauss-Newton approach can be found in Table 3.1 for various fixed regularization parameters.

We then consider the reduced Gauss-Newton with HyBR approach described in Section 3.2.2 and present these results in Table 3.2. The additional column in this table reports the regularization parameter selected by HyBR at each Gauss-Newton iteration. For better visual inspection, only a specific region of the true and reconstructed images after 10 Gauss-Newton iterations is shown in Figure 3.3.

Notice that HyBR in the Gauss-Newton iteration gives similar or slightly better results than the reduced Gauss-Newton approach with fixed regularization parameter. However, the great advantage here is not having to make an *a priori* selection of a regularization parameter. An imprecise value of the regularization parameter may result in poor convergence behavior, as demonstrated by $\lambda = 0.01$ and $\lambda = 0.45$. A regularization parameter that is chosen too small results in a noisy image such as that in Figure 3.3(c), while one chosen too large results in a solution that is too smooth, like that in Figure 3.3(e).

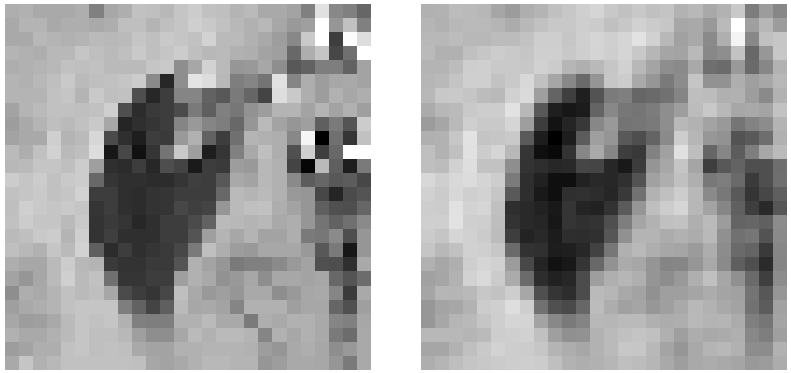
In addition, we include convergence results for the reduced Gauss-Newton approach with HyBR on data containing a higher noise level. Results for 10% noise are presented in Table 3.3. Notice that, as expected, HyBR computed larger regularization parameters than in the case of 1% noise, in order to incorporate more regularization. Furthermore, we observed that using Gauss-Newton with HyBR resulted in similar convergence behavior to that of the reduced Gauss-Newton method with a fixed, appropriately chosen regularization parameter. However, a good value of λ may not be known

Table 3.1: Super-resolution: Convergence of iterations for reduced Gauss-Newton approach with fixed regularization parameter. (1% noise)

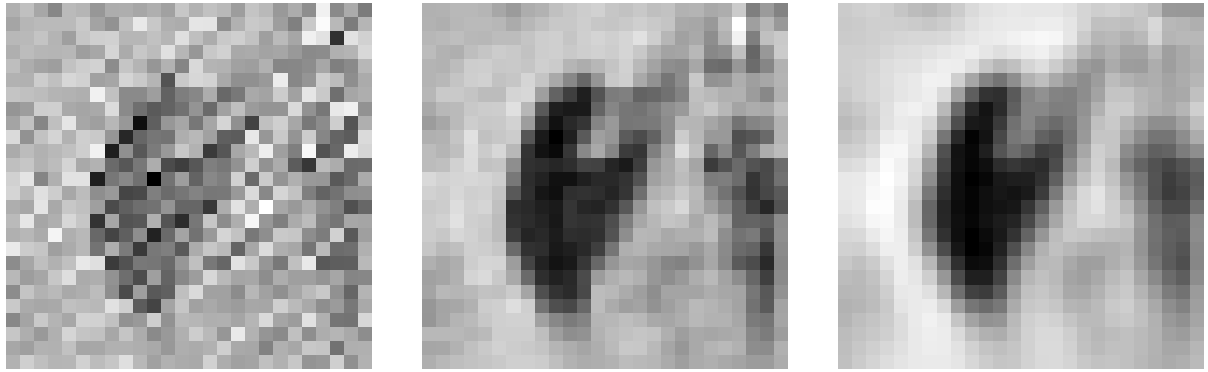
$\lambda = 0.01$				$\lambda = 0.1$				$\lambda = 0.45$			
iteration	rel obj	rel grad	$\Delta\mathbf{y}$	iteration	rel obj	rel grad	$\Delta\mathbf{y}$	iteration	rel obj	rel grad	$\Delta\mathbf{y}$
0	1.0000	1.0000	0.5717	0	1.0000	1.0000	0.5717	0	1.0000	1.0000	0.5717
1	0.7533	0.6668	0.5083	1	0.5067	0.5706	0.3896	1	0.9041	0.5659	0.3305
2	0.5566	0.5058	0.4417	2	0.2885	0.3558	0.2586	2	0.8680	0.4600	0.1791
3	0.4097	0.3661	0.3764	3	0.1850	0.2541	0.1687	3	0.8527	0.4240	0.1026
4	0.3154	0.2753	0.3213	4	0.1295	0.1957	0.1105	4	0.8463	0.4026	0.0899
5	0.2413	0.2102	0.2719	5	0.1041	0.1478	0.0778	5	0.8430	0.3878	0.1035
6	0.1925	0.1886	0.2261	6	0.0925	0.1130	0.0614	6	0.8409	0.3768	0.1223
7	0.1523	0.1839	0.1839	7	0.0870	0.0861	0.0540	7	0.8395	0.3675	0.1419
8	0.1161	0.1932	0.1473	8	0.0843	0.0672	0.0510	8	0.8386	0.3589	0.1616
9	0.0890	0.1467	0.1187	9	0.0829	0.0548	0.0495	9	0.8381	0.3509	0.1812
10	0.0740	0.1242	0.0975	10	0.0821	0.0472	0.0486	10	0.8380	0.3470	0.1909

Table 3.2: Super-resolution: Convergence of iterations for reduced Gauss-Newton approach with HyBR. (1% noise)

iteration	rel obj	rel grad	$\Delta\mathbf{y}$	HyBR computed λ
0	1.0000	1.0000	0.5717	0.2222
1	0.4218	0.5352	0.3410	0.1770
2	0.2132	0.3159	0.2045	0.1502
3	0.1215	0.2225	0.1242	0.1276
4	0.0818	0.1601	0.0817	0.1141
5	0.0654	0.1191	0.0611	0.1079
6	0.0579	0.0895	0.0522	0.1043
7	0.0545	0.0685	0.0488	0.1030
8	0.0528	0.0546	0.0474	0.1018
9	0.0519	0.0461	0.0466	0.1012
10	0.0514	0.0413	0.0461	0.1009



(a) True (b) G-N, HyBR



(c) G-N, $\lambda = .01$ (d) G-N, $\lambda = .1$ (e) G-N, $\lambda = .45$

Figure 3.3: Super-resolution: Comparison of reconstructed images. The images shown are only a specific region of the entire reconstructed image, and these results correspond to the 1% noise case.

Table 3.3: Super-resolution: Convergence of iterations for reduced Gauss-Newton approach with HyBR. (10% noise)

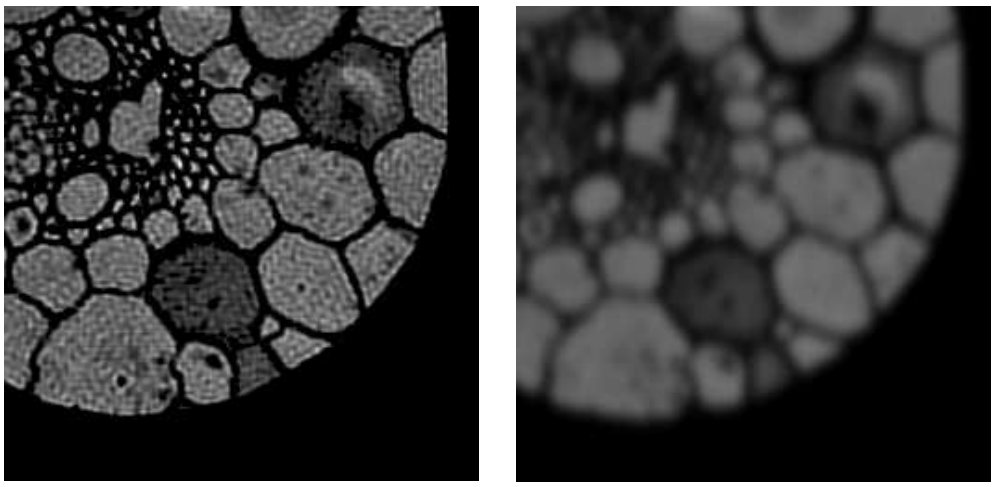
iteration	rel obj	rel grad	$\Delta\mathbf{y}$	HyBR computed λ
0	1.0000	1.0000	0.5941	0.2179
1	0.9021	0.5674	0.4844	0.2118
2	0.8401	0.3674	0.3889	0.2113
3	0.7996	0.2531	0.3207	0.2084
4	0.7787	0.2068	0.2746	0.2072
5	0.7636	0.1831	0.2390	0.2054
6	0.7506	0.1740	0.2058	0.2044
7	0.7424	0.1701	0.1733	0.2045
8	0.7324	0.1714	0.1450	0.2020
9	0.7258	0.1657	0.1212	0.2009
10	0.7209	0.1448	0.1027	0.2002

a priori, and HyBR can select that parameter automatically. Thus, the reduced Gauss-Newton with HyBR approach can be an effective scheme for jointly estimating the displacement parameters and the reconstructed high resolution image in super-resolution imaging applications.

We also remark that by exploiting sparsity of the matrix $\mathbf{A}(\mathbf{y})$, the method is very efficient. It is difficult to provide a precise cost analysis because the amount of work (e.g., number of HyBR iterations, number of Gauss-Newton iterations) is problem dependent, but we can report on wall clock timings for the computations in this dissertation. The joint estimation process for this problem only took 1-2 minutes for 10 iterations of reduced Gauss-Newton with HyBR. All computations were done in MATLAB, using IEEE double precision arithmetic, on a laptop with a 2.33 GHz dual core Intel CPU.

Blind Deconvolution

We now illustrate the use of the variable projection approach with HyBR on a blind deconvolution example. For the results presented in this dissertation, we assume that our goal is to reconstruct the image shown in Figure 3.4(a), given the blurred image in Figure 3.4(b).



(a) True Image (b) Observed Image

Figure 3.4: Blind deconvolution example. The goal is to reconstruct an approximation of the true image of a piece of grain, given the blurred and noisy observed image.

The images presented here contain 256×256 pixels; however, the blurred image was created by convolving a larger 512×512 image with a Gaussian PSF defined by parameters $\mathbf{y}_{\text{true}} = [3, 4, .5]$ and then cropping the center of the image. In this way, we construct a realistic example where boundary conditions and artifacts may play a role. We also added 1% Gaussian white noise.

For the proposed Gauss-Newton algorithm, we used the following initial

Table 3.4: Blind deconvolution: Convergence of iterations for reduced Gauss-Newton approach with HyBR. (1% noise)

iteration	rel obj	rel grad	$\Delta\mathbf{y}$	HyBR computed λ
1	1.0000	1.0000	0.5716	0.1684
2	0.3309	0.6411	0.3347	0.1224
3	0.1493	0.4342	0.2192	0.0988
4	0.0762	0.2953	0.1469	0.0807
5	0.0479	0.2262	0.0998	0.0718
6	0.0355	0.1862	0.0639	0.0678
7	0.0288	0.1614	0.0345	0.0661
8	0.0242	0.1419	0.0139	0.0651
9	0.0213	0.1265	0.0240	0.0645
10	0.0190	0.1151	0.0449	0.0645
11	0.0171	0.1042	0.0656	0.0641
12	0.0159	0.0947	0.0853	0.0638

guess for the blur parameters: $\mathbf{y}_0 = [5, 6, 1]$. The relative error for these parameters is defined in equation (3.19). The relative objective function value, the relative gradient value, the error in the blur parameters, and the computed regularization parameters are presented in Table 3.4. It is impressive to see the accuracy of the blur parameters improve by more than an order of magnitude after only 7 Gauss-Newton iterations.

We notice that with too many iterations, the error in the blur parameters eventually begins to increase. Based on the values of $\Delta\mathbf{y}$, a good stopping point is at 8 iterations. The improved reconstructed image is shown in Figure 3.5, along with the true image and the initial reconstruction using $\mathbf{y}_0$. Since $\mathbf{y}_{\text{true}}$ is not known in practice, alternate methods for selecting a stopping

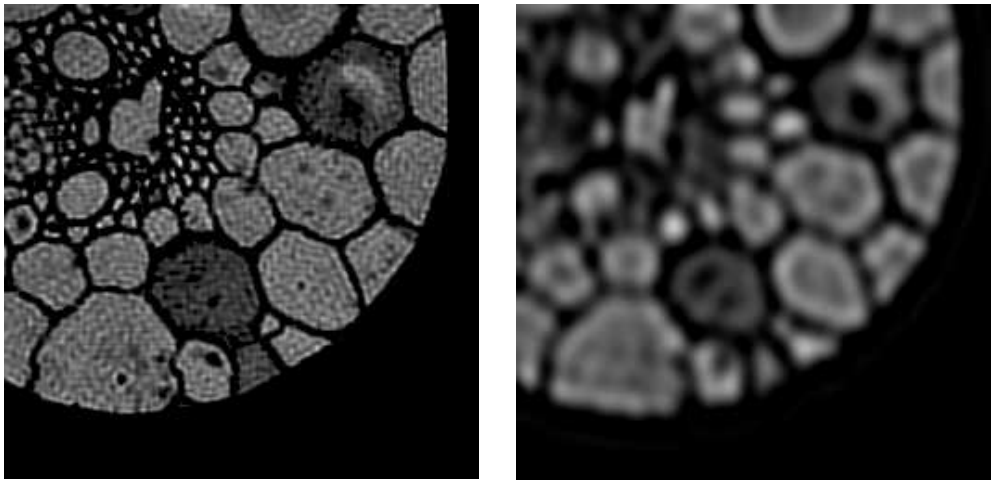
iteration still need to be investigated.

A well-known property of the blind deconvolution problem is the lack of a unique solution. Previously proposed blind deconvolution algorithms have addressed this issue by requiring a significant amount of additional constraints on the problem and by including many parameters for the user to tune and select [71, 130]. Although we have not directly addressed the non-uniqueness problem, we have reduced the number of user-defined parameters to one, specifically, the stopping iteration. Furthermore, from the reconstructed image after 12 Gauss-Newton iterations, shown in Figure 3.5(d), we can see that an advantage of this algorithm is that a slight overestimate of the stopping iteration does not severely affect the image.

In this section we have shown that a reduced Gauss-Newton approach combined with an efficient linear solver with regularization can improve blind deconvolution algorithms. We mention here that this approach can be easily extended to a related but more complicated multi-frame blind (MFB) deconvolution problem [120]. In MFB deconvolution, multiple blurred images are collected, each with a different point spread function and noise realization. Following the notation from equation (3.3), we have each blurred image represented as

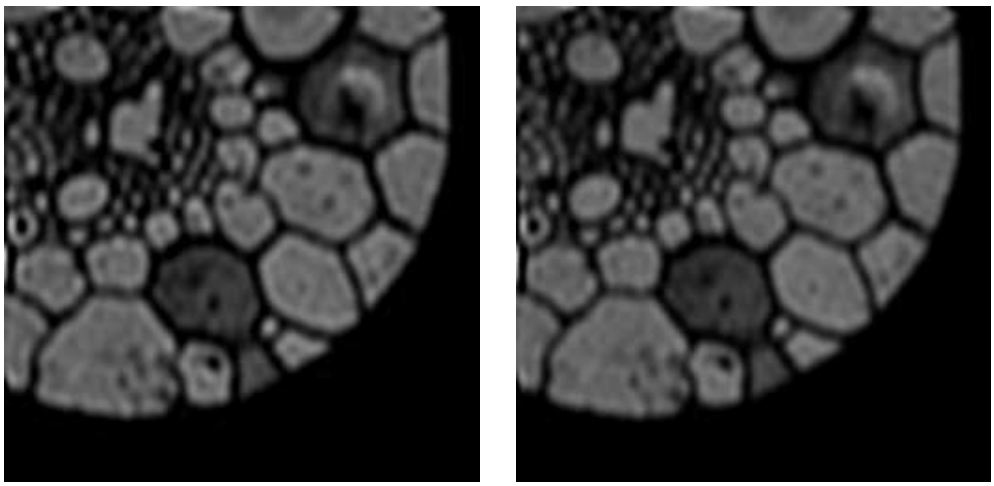
$$\mathbf{b}_i = \mathbf{A}(\mathbf{y}_i)\,\mathbf{x}_{\text{true}} + \boldsymbol{\varepsilon}_i\,,$$

for $i = 1, 2, ..., m$. The goal once again is to simultaneously update the blur parameters and reconstruct the image. The regularized variable projection framework from Section 3.2.2 can be used. We remark that the blurred images and blur parameters should be concatenated to form one long vector, thus allowing all of the blur parameters for all of the images to be updated at the same time. The key difference is that the Jacobian for the MFB deconvolution problem is a block diagonal matrix of the form (3.17) where $\mathbf{J}_i$ is the Jacobian defined in equation (3.18) corresponding to image i. In our experiments, we found that the Gauss-Newton algorithm with HyBR



(a) True Image

(b) Initial Reconstruction, $\mathbf{y}_0$



(c) 8 G-N iterations

(d) 12 G-N iterations

Figure 3.5: Blind deconvolution: Comparison of reconstructed images. Images correspond to the 1% noise case.

applied to the MFB deconvolution problem produced similar results to the blind deconvolution problem presented above.

3.4 Summary and Future Work

We have described an efficient iterative approach for solving separable nonlinear inverse problems. Many researchers have studied the separable nonlinear least squares problem, but few have specifically applied it to large-scale ill-posed inverse problems. We have addressed this problem and shown that by combining a Gauss-Newton approach for minimizing a reduced cost functional with a sophisticated iterative solver for computing Tikhonov regularized solutions for linear ill-posed inverse problems, one can efficiently solve large-scale nonlinear inverse problems, with relatively little user input required.

Nonlinear inverse problems of this form arise in many applications, and we have provided two examples from image deblurring in which the proposed algorithm can successfully update both the image and the imaging parameters simultaneously. Future work includes understanding how the additional regularization term affects the theoretical convergence properties of this algorithm when applied to ill-posed problems and developing an automated way to select the stopping iteration.

Chapter 4

A Nonlinear Poisson-based Inverse Problem

As mentioned in Chapter 1, nonlinear inverse problems are significantly more challenging to solve than linear inverse problems. If certain properties can be exploited, as in the case of the separable nonlinear least squares problem from Chapter 3, efficient methods can be implemented. However, severe nonlinearities arise in many multi-physics applications due to the complicated nature and coupling of physical processes. Furthermore, contrary to the assumption of Gaussian noise in the problems from Chapters 2 and 3, it is common in many medical imaging applications to assume that the imaging process and additive noise follow a Poisson probability distribution. In this chapter we seek iterative statistical techniques for a nonlinear Poisson-based inverse problem arising in a particular application from medical imaging: digital tomosynthesis.

Tomosynthesis imaging involves the acquisition of a series of projection images over a limited angular range, that after reconstruction, results in a pseudo-3D representation of the imaged object. The partial separation of features in the third dimension improves the visibility of objects of interest by reducing the effect of the superimposition of tissues. More specifically, in breast cancer imaging, tomosynthesis is a viable alternative to standard mammography; however, current algorithms for image reconstruction do not

take into account the polyenergetic nature of the x-ray source beam entering the object. This results in inaccuracies in the reconstruction, making quantitative imaging analysis challenging and allowing for beam hardening artifacts.

In this chapter we propose a new mathematical formulation that takes into account the polyenergetic source spectrum and develop a statistical framework that results in a Poisson-based model cost function to minimize. Standard optimization algorithms are considered, and we derive the necessary tools for tomosynthesis reconstruction. Although the model we propose is specific to digital breast tomosynthesis, it can be easily extended to other nonlinear tomographic imaging applications.

We begin with some background information on tomosynthesis in Section 4.1. Then Section 4.2 derives the mathematical framework for modeling the effect of polyenergetic x-rays on the observed images. In Section 4.3 we discuss some of the properties of the problem and consider a variety of iterative optimization techniques for solving the inverse problem of reconstructing a 3D image from 2D projection images. Numerical results in Section 4.4 illustrate the success of our proposed algorithms, and conclusions and future directions can be found in Section 4.5.

4.1 Background Information

Ever since Röntgen produced the first medical x-ray image of his wife's hand in 1895, projection radiography, or x-ray imaging, has made a significant impact in the field of medical imaging. For example, mammography has played a key role in the early detection of breast cancer, allowing doctors to prevent metastatic spread and decrease the number of deaths related to breast cancer. However, a severe limitation of these conventional x-ray systems is that only one 2D projection image of a 3D object is available from each scan.

The projection images have a severe decrease in contrast of structures due to the superimposition of overlaying tissue. Specifically in breast imaging, a false negative diagnosis may be caused by breast cancer obscured by overlapping tissue, while superimposed normal tissues may appear to be a cancerous mass, resulting in a false positive diagnosis [137].

Tomosynthesis is a technique for inversely reconstructing slices of a 3D object from a set of 2D projection images. Though the idea of tomosynthesis is rooted in the theory of conventional geometric tomography, known since the 1930s, it was not until the late 1960s and early 1970s when researchers put these ideas into practice [48, 57]. However tomosynthesis suffered from issues of practical implementation, including insufficient imaging detectors and inadequate computing technology. The introduction of digital technology and electronic image acquisition in the mid to late 70s significantly improved the contrast and resolution capabilities, compared to classical screen-film conventional x-ray systems, revitalizing research in tomosynthesis. Unfortunately, there were still computational and algorithmic limitations, and by the late 70s, tomosynthesis took a "back-seat" to other imaging techniques such as computed tomography (CT) and magnetic resonance imaging (MRI) [33].

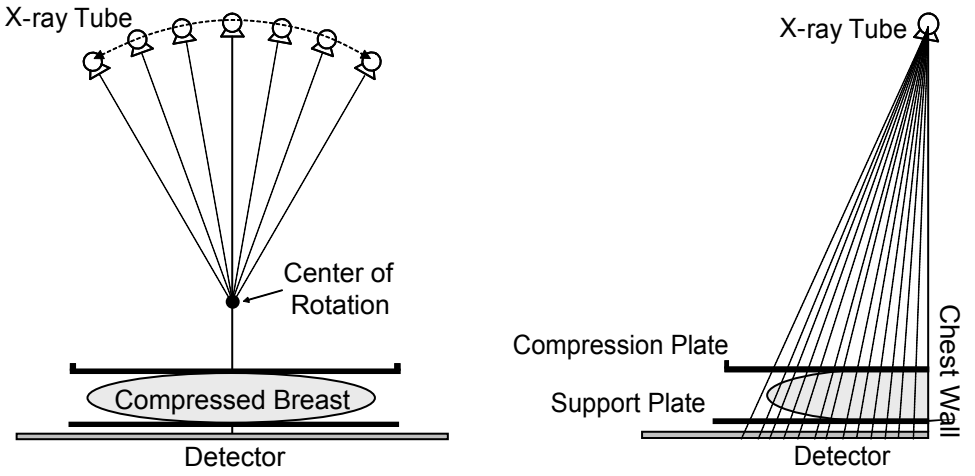
Computed tomography allows the three-dimensional reconstruction of objects by obtaining a complete 360 degree rotation of projection data around the object. Though impressive in many respects, CT has its limitations. The time to complete a CT scan and the radiation dosage requirements for CT can become prohibitively large compared to standard x-ray imaging. Furthermore, certain regions of the body such as breast tissue can be difficult to reconstruct with CT, due to the similar densities of breast tissue compared to that of water [37]. CT is particularly challenging for breast imaging because the patient must be in the prone position during the scan. In addition to being difficult for mobility challenged individuals, this positioning makes it difficult to effectively image the chest wall and axilla area [83]. These undesir-

able properties of CT, along with recent advancements in digital technology, post-processing reconstruction algorithms, and computational power, have motivated and reignited interest in tomosynthesis as a viable alternative to CT for breast imaging.

The basic idea underlying tomosynthesis is that multiple 2D image projections of the object are taken at varying incident angles, and each 2D image provides different information about the 3D object. See Figure 4.1 for an illustration of a typical geometry for breast tomosynthesis imaging. From the limited set of 2D projection images, reconstruction algorithms should be able to reconstruct any number of slices of the 3D object. Sophisticated approaches used for 3D CT reconstruction cannot be applied here because projections are only taken from a limited angular range, leaving entire regions of the frequency space un-sampled. The main challenge in tomosynthesis reconstruction is to remove the out-of-plane blur caused by the backprojection. Variants of the shift-and-add algorithm have been proposed for performing the deblurring operation, but another class of algorithms, which we follow in this chapter, includes iterative reconstruction algorithms that seek to minimize an appropriate cost function. See [33] and references therein for a survey of previous approaches.

Since the observed projection data from x-ray transmission tomography is known to follow a Poisson distribution, especially in the case of low-count transmission scans, many researchers seek to maximize the corresponding log likelihood function. A common approach to solve the optimization problem uses a convexity argument to simplify the problem, a technique described in [91] and used for tomosynthesis reconstruction in [134, 135, 133, 137]. More recently, Chen and Barner [20] have proposed a multi-resolution, maximum *a posteriori* reconstruction algorithm, and Sidky et. al. [123] have implemented a total variation minimization algorithm.

However, an inaccurate assumption in nearly all of the previously proposed



(a) Front view (b) Side view with x-ray tube at 0°

Figure 4.1: Breast tomosynthesis example. Typical geometry of the imaging device used in breast imaging.

reconstruction algorithms is that the x-ray source is monoenergetic; that is, all incident photons have the same energy level. X-ray photons emitted from an x-ray tube have a distribution of energies, and as the x-ray beam passes through any attenuating medium (in this case the breast), there is a preferential absorption of low energy photons that results in an increase in the mean energy of the x-ray beam. This phenomenon, called beam hardening, is of concern in reconstruction methods developed for quantitative imaging because of the attenuation coefficients' dependence on x-ray energy.

Ignoring this energy dependence in the mathematical model can lead to severe artifacts in the reconstructed image, apparent in "halo" effects around bones or streaking in the image. Specifically in breast imaging, "cupping" artifacts or background nonuniformities may appear and are evident in dark bands appearing behind dense objects or a reduction in overall contrast. Few researchers have studied methods for eliminating beam hardening arti-

facts in the case of x-ray computed tomography [6, 14, 29, 36, 37]. Previously proposed methods for eliminating beam hardening artifacts include pre-processing the projection data, post-processing images, or utilizing a dual-energy imaging modality. All of these approaches have some limitations.

4.2 Polyenergetic Tomosynthesis Model

In this section we describe the image acquisition process for breast tomosynthesis and develop a statistical model for image reconstruction. In particular, we develop a mathematical model based on a polyenergetic x-ray source spectrum and derive a statistical model for the problem.

Although most x-ray projection models are derived in terms of the density values for the voxels, it is common in breast imaging to interpret the voxels as a composition of adipose tissue, glandular tissue, or a combination of both [63]. Thus, each voxel of the object can be represented using the percentage glandular fraction, i.e. the percentage of glandular tissue present in that voxel. If density or attenuation coefficient values are desired, then these can be obtained from the glandular fraction through a simple algebraic transformation.

4.2.1 Polyenergetic Model Development

Assume that the 3D object is discretized into a regular grid of voxels and that each of the 2D projection images is discretized into a regular grid of pixels. Specifically, let N represent the number of voxels in the discretized 3D object and let M be the number of pixels in a discretized 2D projection image. In practice N is on the order of a few billion and M is the order of a few million, depending on the size of the imaging detector. The energy-

dependent linear attenuation coefficient for voxel $j = 1, 2, ..., N$ in the breast can be represented as

$$\mu(e)^{(j)} = s(e)x_{\text{true}}^{(j)} + z(e),$$

where $x_{\text{true}}^{(j)}$ represents the percentage glandular fraction in voxel j of the "true" object, and $s(e)$ and $z(e)$ are known energy-dependent linear fit coefficients. This type of decomposition to reduce the number of degrees of freedom is similar to an approach used by De Man et. al. [29] for CT, in which they express the energy dependent linear attenuation coefficient in terms of its photoelectric component and Compton scatter component. However, their model is not optimal for our particular application.

In tomosynthesis, a limited number of projections are taken from various angles in a predetermined angular range, and the photon energies are discretized into a fixed number of levels. Let there be n_θ angular projections and assume the incident x-ray has been discretized into n_e photon energy levels. In practice, a typical scan may have $n_\theta = 21$ and $n_e = 43$. We would like to formulate a mathematical representation for the θ^{th} projection image. For a particular projection angle, we first compute a monochromatic ray trace for one energy level and then sum over all energies. Let $a^{(ij)}$ represent the length of the ray that passes through voxel j, contributing to pixel i. Then the discrete monochromatic ray trace for pixel i can be represented by

$$\sum_{j=1}^{N} \mu(e)^{(j)} a^{(ij)} = s(e) \sum_{j=1}^{N} x_{\text{true}}^{(j)} a^{(ij)} + z(e) \sum_{j=1}^{N} a^{(ij)} \,. \tag{4.1}$$

Using the standard mathematical model for transmission radiography, the i^{th} pixel value for the θ^{th} noise-free projection image, incorporating all photon energies present in the incident x-ray spectrum, can be written as

$$b_\theta^{(i)} = \sum_{e=1}^{n_e} \varrho(e) \exp\left(- \sum_{j=1}^{N} \mu(e)^{(j)} a^{(ij)} \right), \tag{4.2}$$

where $\varrho(e)$ is a product of the current energy with the number of incident photons at that energy.

To simplify notation, let's define $\mathbf{A}_\theta$ to be an $M \times N$ matrix with entries $a^{(ij)}$. Then equation (4.1) gives the i^{th} entry of vector

$$s(e)\mathbf{A}_\theta \mathbf{x}_{\text{true}} + z(e)\mathbf{A}_\theta \mathbf{1}\,,$$

where $\mathbf{x}_{\text{true}}$ is a vector whose j^{th} entry is $x^{(j)}_{\text{true}}$ and $\mathbf{1}$ is a vector of all ones. Furthermore, the θ^{th} noise-free projection image in vector form can be written as

$$\mathbf{b}_\theta = \sum_{e=1}^{n_e} \varrho(e) \exp\left(-[s(e)\mathbf{A}_\theta \mathbf{x}_{\text{true}} + z(e)\mathbf{A}_\theta \mathbf{1}]\right), \tag{4.3}$$

where the exponential function is applied component-wise.

Tomosynthesis reconstruction is an inverse problem where the goal is to approximate the volume, $\mathbf{x}_{\text{true}}$, given the set of projection images from various angles, $\mathbf{b}_\theta$, $\theta = 1, 2, ...n_\theta$. In the next section we discuss a statistical model used for solving this nonlinear inverse problem.

4.2.2 Poisson-based Likelihood Function

It is widely accepted in the medical imaging community that measurements obtained by x-ray transmission imaging, i.e. photon counts, can be accurately modeled as independently distributed Poisson random variables, with additional background noise. Based on x-ray projection model (4.1) and (4.2), the expected value for the measured data at pixel i for angle θ given volume approximation $\mathbf{x}$ can be written as

$$\begin{aligned} E[b^{(i)}_\theta, \mathbf{x}] &= \sum_{e=1}^{n_e} \varrho(e) \exp\left(-\left[s(e)\sum_{j=1}^{N} x^{(j)}a^{(ij)} + z(e)\sum_{j=1}^{N} a^{(ij)}\right]\right) + \bar{\varepsilon}^{(i)} \\ &\equiv \bar{b}^{(i)}_\theta + \bar{\varepsilon}^{(i)}, \end{aligned}$$

where $\bar{\varepsilon}^{(i)}$ represents the mean of the errors due to electronic noise and scatter in the observed data. In tomography applications, it is assumed that

the additive noise is a realization of a Poisson random variable, where the statistical mean $\bar{\varepsilon}^{(i)}$ is known or can be approximated.

For each angle, the measured data can be statistically modeled as an independent Poisson random process [36]. That is, the i^{th} pixel of the observed projection image, $\mathbf{b}_\theta$, is a realization of a Poisson random variable with mean, $\bar{b}_\theta^{(i)} + \bar{\varepsilon}^{(i)}$:

$$b_\theta^{(i)} \sim \text{Poisson}(\bar{b}_\theta^{(i)} + \bar{\varepsilon}^{(i)}) \,.$$

Thus, we can say that the probability or likeliness of observing projection image $\mathbf{b}_\theta$, given volume $\mathbf{x}$, is described by the likelihood function [78, 129]

$$p(\mathbf{b}_\theta, \mathbf{x}) = \prod_{i=1}^{M} \frac{e^{-(\bar{b}_\theta^{(i)} + \bar{\varepsilon}^{(i)})} (\bar{b}_\theta^{(i)} + \bar{\varepsilon}^{(i)})^{b_\theta^{(i)}}}{b_\theta^{(i)}!} \,. \tag{4.4}$$

We would like to compute the glandular fractions, $\mathbf{x}$, that maximize this likelihood function. For ease of computation, a monotonic negative log operation is applied to the likelihood function (4.4), and the maximum likelihood estimator (MLE) can be found by minimizing the negative log likelihood function:

$$\begin{aligned} -L_\theta(\mathbf{x}) &= -\log p(\mathbf{b}_\theta, \mathbf{x}) \\ &= \sum_{i=1}^{M} (\bar{b}_\theta^{(i)} + \bar{\varepsilon}^{(i)}) - b_\theta^{(i)} \log(\bar{b}_\theta^{(i)} + \bar{\varepsilon}^{(i)}) \,, \end{aligned} \tag{4.5}$$

for all θ. In the next section we consider efficient algorithms for minimizing the above negative log likelihood function.

4.3 Iterative Reconstruction Algorithms

In this section we describe some numerical algorithms for estimating the MLE solution for the polyenergetic tomosynthesis reconstruction problem:

$$\mathbf{x}_{MLE} = \underset{\mathbf{x}}{\text{argmin}} \left\{ \sum_{\theta=1}^{n_\theta} -L_\theta(\mathbf{x}) \right\} . \tag{4.6}$$

To simplify notation in the derivation, we fix a particular angle and drop the subscript, θ, for the remainder of this section.

For a monoenergetic likelihood function, a variety of researchers have studied this optimization problem. In 1995, Lange and Fessler [91] presented a comparison of the EM algorithm, a scaled gradient descent algorithm, and a "convex" algorithm, in which properties of convexity were used to iteratively approximate the log likelihood function [90, 41]. Under simplifying assumptions that the solution exists, is unique and lies in the interior of the feasible region, they prove that all three methods converge locally to the MLE solution. Furthermore, they prove global convergence for the EM and convex algorithm when applied to the log posterior function:

$$\Phi(\mathbf{x}) = L(\mathbf{x}) - \lambda R(\mathbf{x}),$$

where $R(\mathbf{x})$ is a penalizing prior, or smoothing function, and λ is a regularization parameter controlling the accepted level of smoothness. By selecting a strictly convex prior function, strict concavity of the log posterior function can be established [91]. However, all of the derivations assume a monoenergetic x-ray source and a strictly convex cost function, meaning noise is set to zero in the model (c.f. [43] for derivation).

Our problem not only assumes a polyenergetic x-ray beam, but also takes into account the presence of noise in the data. Thus, the theories from these previous algorithms for maximizing the likelihood and posterior functions do not apply. More specifically, due to severe nonlinearities, the polyenergetic cost function may not be convex, and regularization must be incorporated to suppress the noise.

With respect to convexity, there are some reasonable assumptions under which the polyenergetic cost function is convex. With the new formulation, it can be shown that the cost function is convex with respect to the glandular fractions, under the following two conditions:

1. $\mathbf{A}$ is full rank, and

2. $b^{(i)} - (\bar{b}^{(i)} + \bar{\varepsilon}^{(i)}) \leq \dfrac{\min\limits_{e} s(e)}{\max\limits_{e} s(e)} \left(\dfrac{b^{(i)}}{\bar{b}^{(i)} + \bar{\varepsilon}^{(i)}} \right) \bar{b}^{(i)} \qquad \text{for all } i.$

Further details of the derivation of these conditions can be found in Appendix Section A.3. In our numerical experiments, we found that the first condition holds true, and a good initial guess that satisfies the second condition is not difficult to obtain.

In regards to the additive noise, selecting the optimal regularizing function $R(\mathbf{x})$ for breast tomosynthesis reconstruction is still an open question. It has been noted that for transmission image reconstruction, nonquadratic, edge-preserving penalty functions are more desirable for images with piecewise smooth regions [30, 43]. For the monoenergetic case, Bleuet et. al. [10] suggested an adaptive 3D regularization scheme, Sidky et. al. [123] and Kastanis et. al. [84] implemented a total variation optimization approach and Chen and Barner [20] use a Markov random fields regularization function. For the polyenergetic case, Elbakri and Fessler [37] used a convex, edge-preserving Huber penalty for its desirable properties. However, current research has not yet determined optimal regularization methods for breast image reconstruction, and this topic should be investigated in future studies. For this dissertation we incorporate regularization to deal with noise and errors in the data via early termination of the iterations. That is, we focus on the new polyenergetic formulation and investigate optimization algorithms to minimize the original negative log likelihood function (4.5).

We consider a gradient descent and a Newton algorithm. To do this, we need to compute the gradient and Hessian of the objective function with

respect to the 3D volume, $\mathbf{x}$. We first establish two important equalities:

$$\frac{\partial \bar{b}^{(i)}}{\partial x^{(j)}} = -a^{(ij)} \sum_{e=1}^{n_e} \varrho(e) s(e) \exp\left(- \left[s(e) \sum_{j=1}^{N} a^{(ij)} x^{(j)} + z(e) \sum_{j=1}^{N} a^{(ij)} \right] \right) \quad (4.7)$$

$$\frac{\partial}{\partial x^{(\ell)}} \left(\frac{\partial \bar{b}^{(i)}}{\partial x^{(j)}} \right) = a^{(i\ell)} a^{(ij)} \sum_{e=1}^{n_e} \varrho(e) s(e)^2 \exp\left(- \left[s(e) \sum_{j=1}^{N} a^{(ij)} x^{(j)} + z(e) \sum_{j=1}^{N} a^{(ij)} \right] \right) . \quad (4.8)$$

These equations will aid in the derivation of the following algorithms.

4.3.1 Gradient Descent Algorithm

The first approach we consider is a simple gradient descent algorithm for minimizing the function in equation (4.5), which takes the following form:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla L(\mathbf{x}_k) \,, \quad (4.9)$$

where α_k is an iteration dependent step length parameter and $\nabla L(\mathbf{x}_k) = \frac{\partial}{\partial x^{(j)}}(-L_\theta)$ for all θ.

The first derivative of the negative log likelihood function with respect to $\mathbf{x}$ is given by

$$\begin{aligned} \frac{\partial}{\partial x^{(j)}}(-L) &= \sum_{i=1}^{M} \left(1 - \frac{b^{(i)}}{\bar{b}^{(i)} + \bar{\varepsilon}^{(i)}} \right) \frac{\partial \bar{b}^{(i)}}{\partial x^{(j)}} \\ &= \sum_{i=1}^{M} a^{(ij)} \left(\frac{b^{(i)}}{\bar{b}^{(i)} + \bar{\varepsilon}^{(i)}} - 1 \right) \sum_{e=1}^{n_e} \varrho(e) s(e) \exp \\ &\qquad \left(- \left[s(e) \sum_{j=1}^{N} a^{(ij)} x^{(j)} + z(e) \sum_{j=1}^{N} a^{(ij)} \right] \right) , \end{aligned}$$

where the second equation follows from equation (4.7). Using matrix notation, the gradient can be written simply as

$$\nabla L(\mathbf{x}_k) = \mathbf{A}^T \mathbf{v}_k \,,$$

where $\mathbf{v}_k$ is a vector whose i^{th} entry is given by

$$v_k^{(i)} = \left(\frac{b^{(i)}}{\bar{b}^{(i)} + \bar{\varepsilon}^{(i)}} - 1 \right) \sum_{e=1}^{n_e} \varrho(e) s(e) \exp\left(- \left[s(e) \sum_{j=1}^{N} a^{(ij)} x_k^{(j)} + z(e) \sum_{j=1}^{N} a^{(ij)} \right] \right) .$$

The gradient descent approach is known to converge slowly, and the step length parameter α_k is chosen to ensure that the objective function decreases.

4.3.2 Newton Approach

Another approach for minimizing the negative log likelihood function is to employ a Newton-type method. Newton methods are well-known to have faster convergence properties; however, they are more sensitive than gradient descent methods to noise in the data and require the initial estimate to be a good enough approximation. A typical Newton iteration has the following form:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{H}_k^{-1} \nabla L(\mathbf{x}_k) \,. \tag{4.10}$$

However, the Hessian, $\mathbf{H}_k$, may be nontrivial or impossible to compute. One of our main contributions to this project is to make the Newton approach feasible. That is, with the new polyenergetic formulation, we are able to derive an analytical formula for the Hessian matrix by using some detailed calculus and matrix algebra.

We provide the mathematical details here. For $j = 1, 2, ...N$ and $\ell =$

$1, 2, ...N$, the $j\ell^{th}$ entry of the Hessian matrix, $\mathbf{H}$, can be written as

$$\begin{aligned} h^{(j\ell)} &= \frac{\partial}{\partial x^{(\ell)}}\left(\frac{\partial}{\partial x^{(j)}}(-L(\mathbf{x}))\right) \\ &= \frac{\partial}{\partial g^{(\ell)}}\left(\sum_{i=1}^{M}\left(1-\frac{b^{(i)}}{\bar{b}^{(i)}+\bar{\varepsilon}^{(i)}}\right)\frac{\partial \bar{b}^{(i)}}{\partial x^{(j)}}\right) \\ &= \sum_{i=1}^{M}\left\{\left(1-\frac{b^{(i)}}{\bar{b}^{(i)}+\bar{\eta}^{(i)}}\right)\frac{\partial}{\partial x^{(\ell)}}\left(\frac{\partial \bar{b}^{(i)}}{\partial x^{(j)}}\right)+ \right. \\ &\qquad \left. \frac{\partial \bar{b}^{(i)}}{\partial x^{(j)}}\frac{\partial}{\partial x^{(\ell)}}\left(1-\frac{b^{(i)}}{\bar{b}^{(i)}+\bar{\varepsilon}^{(i)}}\right)\right\}, \quad (4.11) \end{aligned}$$

where the last equality is just application of the product rule. The derivative in the first term can be evaluated by equation (4.8) and the derivative in the second term can be expanded to be

$$\frac{\partial}{\partial x^{(\ell)}}\left(1-\frac{b^{(i)}}{\bar{b}^{(i)}+\bar{\varepsilon}^{(i)}}\right)=\frac{b^{(i)}}{(\bar{b}^{(i)}+\bar{\varepsilon}^{(i)})^2}\frac{\partial \bar{b}^{(i)}}{\partial x^{(\ell)}}\,. \quad (4.12)$$

Now, plugging in (4.7), (4.8) and (4.12) into (4.11), we get the following expression for the $j\ell^{th}$ entry of the Hessian matrix:

$$\begin{aligned} h^{(j\ell)} &= \sum_{i=1}^{M} a^{(ij)}a^{(i\ell)}\left\{\left(1-\frac{b^{(i)}}{\bar{b}^{(i)}+\bar{\varepsilon}^{(i)}}\right)\sum_{e=1}^{n_e}\varrho(e)s(e)^2\exp \right. \\ &\qquad \left(-\left[s(e)\sum_{j=1}^{N}a^{(ij)}x^{(j)}+z(e)\sum_{j=1}^{N}a^{(ij)}\right]\right)+ \\ &\qquad \frac{b^{(i)}}{(\bar{b}^{(i)}+\bar{\varepsilon}^{(i)})^2}\left[\sum_{e=1}^{n_e}\varrho(e)s(e)\exp \right. \\ &\qquad \left.\left.\left(-\left[s(e)\sum_{j=1}^{N}a^{(ij)}x^{(j)}+z(e)\sum_{j=1}^{N}a^{(ij)}\right]\right)\right]^2\right\}. \quad (4.13) \end{aligned}$$

Since nothing in the curly brackets of equation (4.13) depends on j or ℓ, let's define vector $\mathbf{w}$ with entries

$$w^{(i)}=\{...\}\,, \quad (4.14)$$

then equation (4.13) simplifies to

$$h^{(j\ell)} = \sum_{i=1}^{M} a^{(ij)} a^{(il)} w^{(i)},$$

corresponding to the matrix

$$\mathbf{H} = \mathbf{A}^T \mathbf{W} \mathbf{A}\,,$$

where $\mathbf{W}$ is a diagonal matrix with vector $\mathbf{w}$ on the diagonal. Note that only matrix $\mathbf{W}$ is iteration dependent. Thus, we have

$$\mathbf{H}_k = \mathbf{A}^T \mathbf{W}_k \mathbf{A},$$

and the Newton step at iteration k can be found by solving the following system:

$$\mathbf{H}_k \mathbf{s}_k = -\nabla L(\mathbf{x}_k). \tag{4.15}$$

Note that equation (4.15) is the normal equations formulation of the least squares problem:

$$\min_{\mathbf{s}_k} \left\| \mathbf{W}_k^{\frac{1}{2}} \mathbf{A} \mathbf{s}_k - \mathbf{W}_k^{-\frac{1}{2}} \mathbf{v}_k \right\|_2 , \tag{4.16}$$

where $\mathbf{W}_k^{\frac{1}{2}} = \mathrm{diag}(\mathbf{w}^{\frac{1}{2}})$. A variety of methods can be used to solve (4.16). An important remark here is that for large-scale problems, it is recommended to use inexact Newton approaches [86, 104], where an iterative method with early termination is used to approximately solve the inner system. For our problem we use the conjugate gradient algorithm for least squares (CGLS) [8], so our approach fits a Newton-CG, or truncated Newton, optimization framework.

4.4 Numerical Results

In this section we illustrate the success of the proposed algorithms presented in Section 4.3 for solving the polyenergetic tomosynthesis reconstruction problem for a simulated breast imaging example.

Given a 3D volume with $128 \times 128 \times 128$ voxels [12], we normalized the values so that the voxel values range between 0 and 100, each value representing the percentage fraction of glandular tissue in that voxel. Then 21 projection images were taken from equally spaced angles, within an angular range from -30° to 30° at 3° intervals, using the typical geometry for breast tomosynthesis, as illustrated in Figure 4.1. Each 2D projection image was 192×256 pixels. The original object represented a medium-sized breast of size 12.8 cm $\times$ 12.8 cm $\times$ 6.4 cm, and the detector was 19.2 cm $\times$ 25.6 cm. The source to detector distance at 0° was set to 66 cm and the distance from the center of rotation to detector was 0 cm. The incident x-ray spectrum consisted of 43 different energy levels, ranging from 5.0 keV to 26 keV in 0.5 keV steps.

For each projection angle, the ray trace matrix $\mathbf{A}_\theta$ was computed using a cone beam model with Siddon's ray tracing algorithm [122]. For each of the reconstruction algorithms, the initial guess of the 3D volume was a uniform image with all voxel values set to 50, meaning half glandular and half adipose tissue. To simulate a more realistic example, we created the projection images using a $128 \times 128 \times 128$ volume, but reconstructed a $128 \times 128 \times 8$ volume. Furthermore, the projection images included enough additive Poisson noise so that the relative noise level was 0.1%. The slices of the volume that we would like to reconstruct can be found in Figure 4.2, and a few of the cropped, observed projection data can be found in Figure 4.3.

To evaluate the performance of each of the algorithms presented in Section 4.3, we present in Table 4.1 the relative objective function value, the relative gradient value and the relative error for the 3D image. We note here that for the Newton algorithm, the stopping criteria used for CGLS on the inner problem (4.16) was a scaled residual tolerance of .17 or a maximum number of 100 iterations. The number of CGLS iterations reported for the inner problem at each Newton iteration can be found in the last column of the

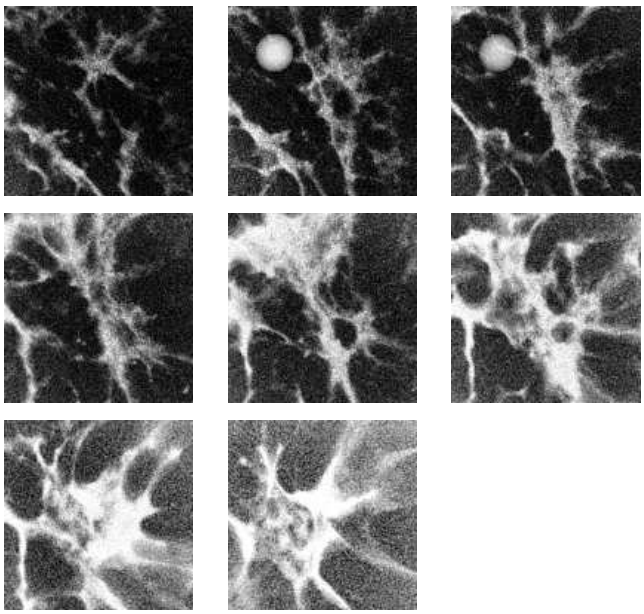


Figure 4.2: Breast tomosynthesis example. True volume slices.

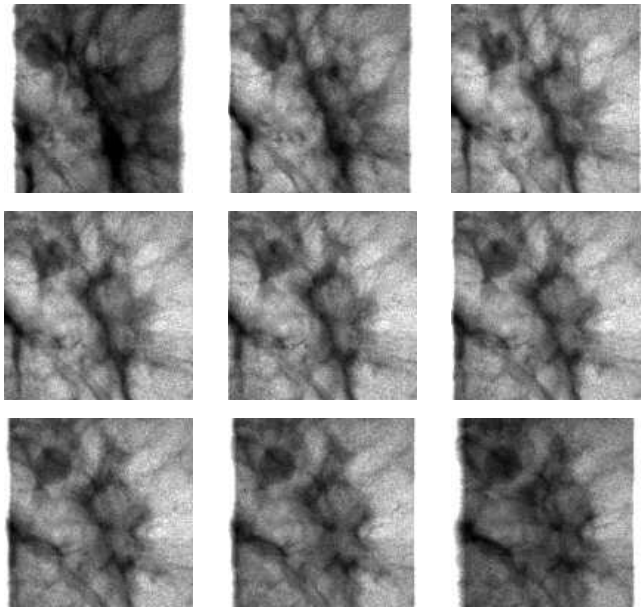


Figure 4.3: Breast tomosynthesis example. Sample projection images.

second table. Furthermore, we remark that in applications where the Hessian cannot be derived analytically, a quasi-Newton approach such as LBFGS can be a good alternative. However, in our experience, the LBFGS method was quite slow in converging, and a Newton approach worked much better.

It is evident from Table 4.1 that 4 iterations of the Newton algorithm produced a small relative error for the image. However, since each Newton iteration requires the solution of a linear system, it is difficult to present a fair comparison of the reconstruction algorithms. In terms of computational effort, we remark that the most computationally burdensome aspect of the reconstruction is the matrix-vector and matrix-transpose-vector multiplication with ray trace matrix, $\mathbf{A}$. Each function and gradient evaluation of the likelihood function requires a total of 3 "ray trace" multiplications (2 for the function evaluation and 1 more for the gradient), and a multiplication operation with the Hessian (or its transpose) only requires 2 "ray trace" multiplications. Furthermore, we use a backtracking line search strategy at each iteration of the optimization scheme that uses the Cauchy point as an initial guess, thus requiring another multiplication with the Hessian. It is therefore the case that the computational cost and timing for, say, one Newton iteration with 50 inner CG iterations with the Hessian is not equivalent to 50 gradient descent iterations.

In Figures 4.4 and 4.5 we present a visual comparison of images, with slices of the true volume in the first column. In the second column, we provide the "best" monoenergetic reconstruction of the linear attenuation coefficients using Lange and Fessler's "convex" algorithm. Recall that the monoenergetic algorithm reconstructs attenuation coefficients rather than glandular fractions, so by "best" we mean that this reconstruction provided the smallest computed image error between the reconstructed attenuation coefficients and the attenuation coefficients at the median energy level for the true volume. In the third and fourth columns, we present the images for the

Table 4.1: Breast tomosynthesis: Convergence of iterations for gradient descent and Newton with CGLS.

Gradient Descent

iteration	relative objective	relative gradient	Image Error
0	1.739e-4	1.0000	0.6377
1	1.583e-4	0.6370	0.5994
25	1.440e-4	0.0349	0.4457
50	1.438e-4	0.0194	0.4189
75	1.436e-4	0.0314	0.3899
100	1.434e-4	0.0048	0.3069

Newton with CGLS

Newton iteration	relative objective	relative gradient	Image Error	inner CGLS iterations
0	1.739e-4	1.0000	0.6377	-
1	1.497e-4	0.4501	0.4852	8
2	1.442e-4	0.0774	0.4503	5
3	1.433e-4	0.0136	0.2831	31
4	1.433e-4	0.0022	0.2724	72
5	1.433e-4	0.0008	0.2876	100

gradient descent and Newton algorithms after approximately 12 minutes of wall clock time. These correspond to reconstructed images after 20 iterations of gradient descent and 3 iterations of Newton, with the number of "ray trace" multiplications being 103 and 100 respectively. In terms of timing, the Newton algorithm for this problem is the clear winner because it computed a solution with better visual quality.

It is important to remark here that with more iterations of the monoenergetic algorithm, the images become significantly worse in terms of contrast resolution. This is expected because we are using an inaccurate model for reconstruction. However, with more iterations of the gradient descent algorithm with the accurate polyenergetic model, the image will eventually resemble the superior quality obtained from the Newton algorithm. Also, we remark that although the image errors in Table 4.1 decrease in early iterations, these errors will eventually increase. This is slightly evident in the later Newton iterations and is typical of ill-posed problems. Future work is needed to develop methods to suppress noise amplification. Our numerical results have successfully shown that reconstruction based on a polyenergetic model can produce significantly better results than the current reconstruction algorithms.

4.5 Significance and Future Directions

We have described a novel formulation for polyenergetic tomosynthesis reconstruction and shown that standard numerical optimization techniques can be used to reconstruct 3D volumes from limited angle 2D projection images. Many researchers have studied the monoenergetic tomosynthesis problem, but few have investigated the nonlinear problem that arises from a polyenergetic spectrum. We have addressed this problem and shown that by reformu-

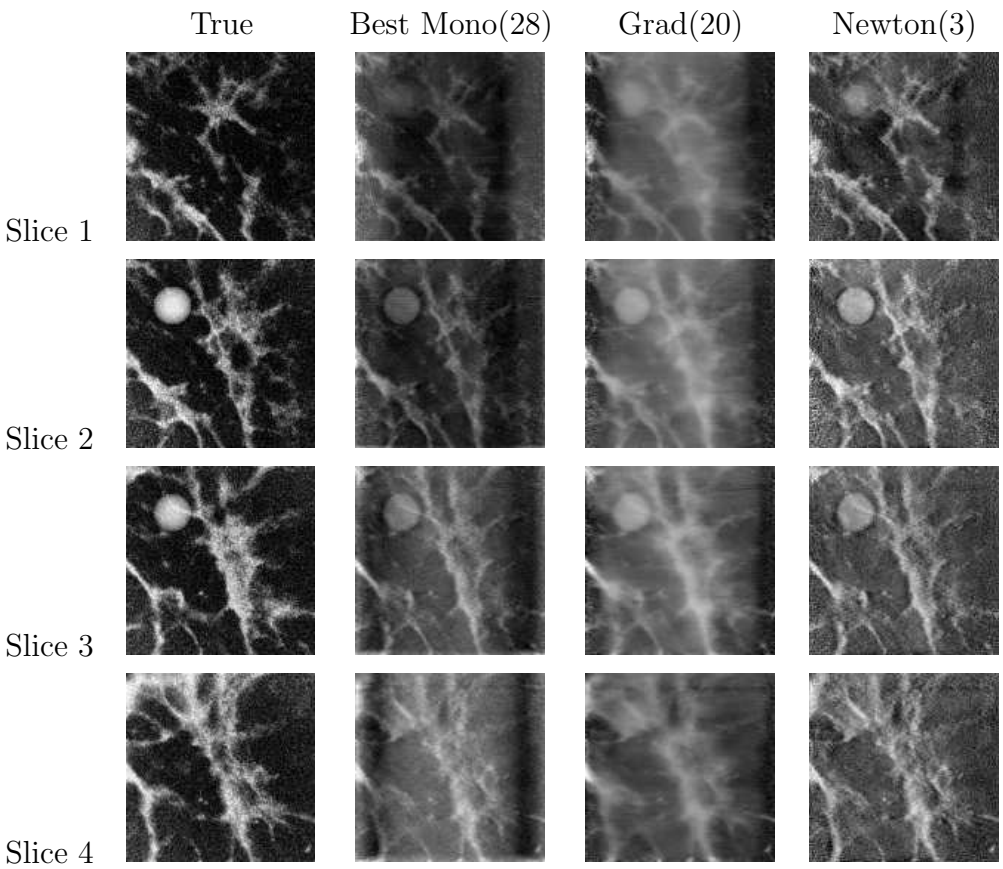


Figure 4.4: Breast tomosynthesis: Reconstructed slices 1-4.

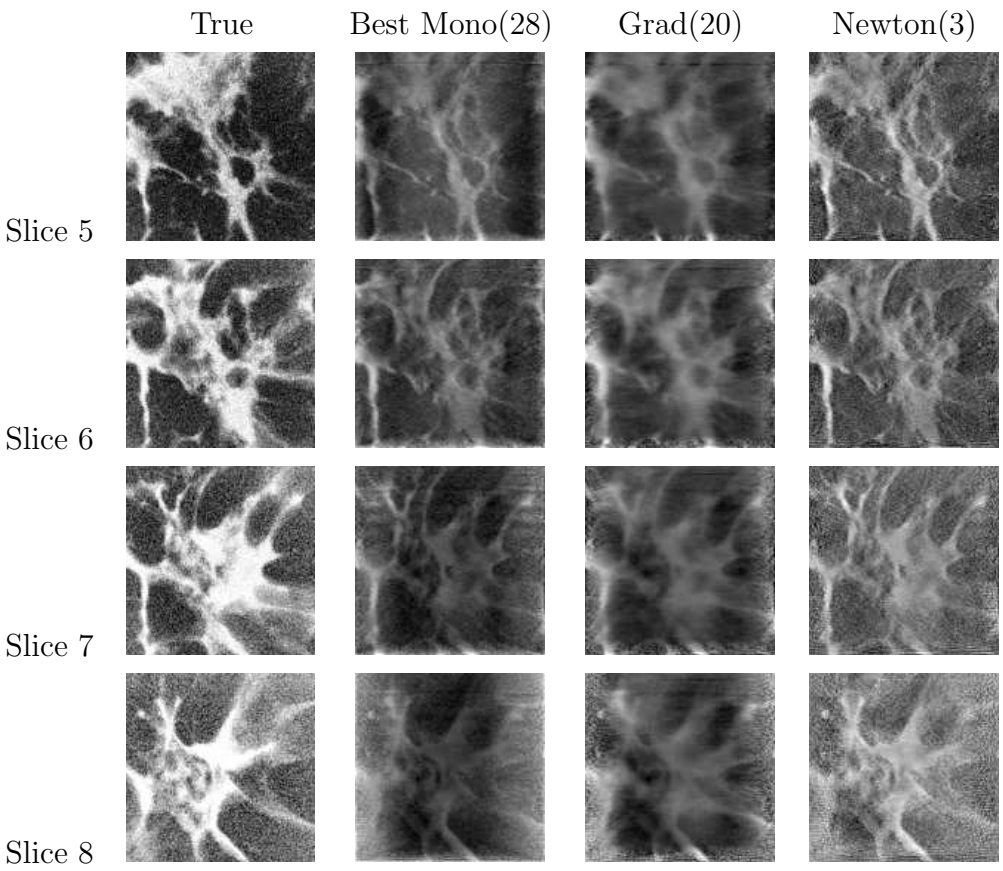


Figure 4.5: Breast tomosynthesis: Reconstructed slices 5-8.

lating the problem in terms of the glandular fractions, one can analytically compute the necessary gradients and Hessians for efficiently solving the nonlinear inverse problem.

Nonlinear inverse problems of this form arise in many applications, and we have focused on one particular application from breast imaging. Our numerical results illustrate the potential for successful application of sophisticated mathematical techniques and approaches to solve this problem. However, many open problems still remain, especially in regards to computing the inner Newton iterations. Future work includes developing efficient preconditioners for the inner Newton system. For example, limited-memory quasi-Newton methods, such as LBFGS, have been proposed as good preconditioners for use within inexact Newton methods [1, 4, 98]. Another potential cause for concern is non-convexity. In our experiments, we were able to choose an initial guess that ensured convexity of the objective function. However, if the Hessian is not positive definite or is close to being singular, i.e. if the diagonal entries in $\mathbf{W}$ become negative or zero, then using a modified Newton method that uses a positive definite approximation to the Hessian may be beneficial [104]. In either case, knowing how many inner Newton iterations to run can be difficult, and standard approaches that stop the inner iterations when the residual is small may not be appropriate, especially for ill-posed problems.

Nonlinear optimization approaches, such as the Newton-type methods mentioned above, become significantly more difficult to analyze for ill-posed problems [40, 65, 79]. In addition to better understanding the convergence of nonlinear schemes, a comprehensive evaluation of regularization methods for breast imaging needs to be conducted, and accurate methods for selecting regularization parameters need to be investigated. Furthermore, there may be potential benefits in implementing bound constrained algorithms that restrict the solution to a feasible region.

A direction of particular interest from the medical community is the quan-

tification of physical uncertainties from the system geometry. Due to the massive size and constant movement of the x-ray source, errors from misalignment of the x-ray tube with the image detector are inevitably introduced in the mathematical model. Efficient methods for estimating and correcting for these errors should be investigated. In addition, evaluating the performance of these methods in the presence of materials that do not conform to this model will be pursued in our future work. Also, we briefly mention here that another approach to tomosynthesis that may require future mathematical contributions is 3D volume reconstruction from a continuous x-ray scan. In this case, we believe a motion blur technique may be used to model the acquisition process.

Chapter 5

Large-Scale Implementation

Large systems inevitably arise when dealing with real-life imaging applications. Thus, it is important for us to investigate efficient computer implementations for large-scale problems. By exploiting structure or sparsity in the problem, or if good approximations of the Hessian can be found, high computational costs and storage requirements may be alleviated. For example, in the image deblurring and blind deconvolution examples from Chapters 2 and 3 respectively, we were able to exploit structure in the matrix $\mathbf{A}$ and efficiently compute matrix-vector products using an object oriented approach from RestoreTools. However, larger images would require more data to manage and more unknowns to compute. In this case the blurring matrices may become prohibitively large, especially for 3D images. In other situations exploiting nice properties may not be obvious or even possible. If high-performance capabilities such as shared memory or distributed computing clusters are available, then parallel processing is a powerful tool that can be used. In this chapter we describe some efficient techniques for large-scale parallel implementations.

First we describe an imaging application from molecular biology where it is essential to have efficient large-scale implementations. We discuss iterative methods for estimating a 3D density map of a macromolecular complex from a large number of 2D Cryo-EM projection images. The problem is computationally demanding because not only does the problem reconstruct large

3D images, but also the low signal to noise ratio requires us to use a large number of projection images for achieving atomic resolution. After describing the problem in Section 5.1, we describe an efficient implementation of the matrix-vector and matrix-transpose-vector multiplications using the Message Passing Interface (MPI) library on distributed memory parallel computers in Section 5.2. In particular, we present a parallelization strategy in which both the 2D images and the 3D electron density are distributed on a 2D processor grid. We estimate the theoretical performance of our parallelization scheme in terms of the floating point operation and communication volume ratio and report the measured performance results of our implementation for a few datasets.

We emphasize here that all of the previously described applications from earlier chapters can also benefit from the implementation techniques developed here. For example, in super-resolution imaging, having more low resolution images may allow the reconstruction of a higher resolution image. Also, high-performance computing will become increasingly important for extension of these methods to work for 3D images. Furthermore, in digital tomosynthesis a finer scale or discretization for the 3D volume would provide doctors with more detailed images for better detection and diagnosis of cancers. Since no preconditioning of the problem is assumed, the main computational bottleneck in all of the previously discussed iterative algorithms is the matrix-vector multiplications. Thus, efficient implementations for these operations would be helpful for future extensions and developments.

5.1 Motivating Application: Cryo-EM

In the post-genomic era, high resolution determination of protein structures becomes extremely important for accurate interpretations of biological functions at the molecular level. The reconstruction of these 3D macromolecular

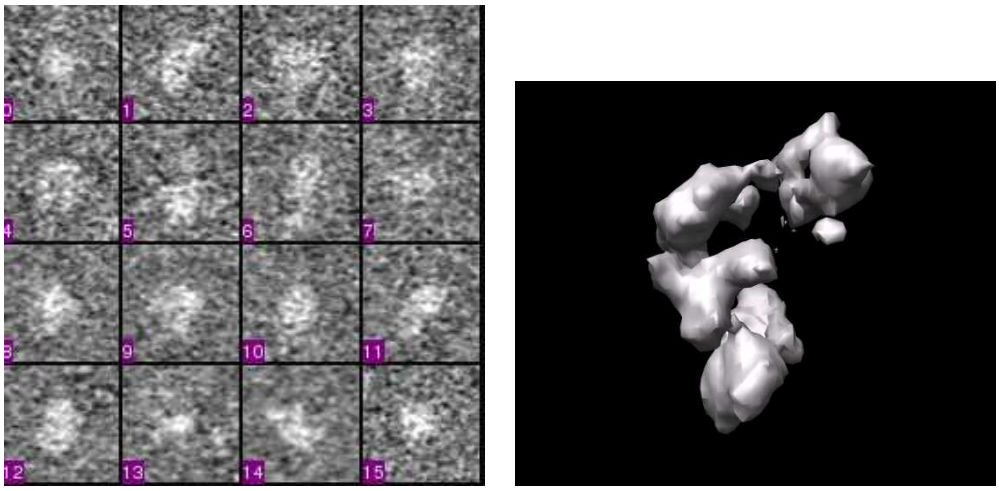
structures from 2D electron microscopy images of frozen hydrated samples (a technique often referred to as single-particle Cryo-EM [44]) has many advantages over other imaging techniques such as x-ray crystallography [132] and nuclear magnetic resonance (NMR) imaging [11]. In particular, it is suitable for large macromolecular complexes that are difficult to crystallize, and it allows molecular biologists to capture the structure of macromolecules in their native states. However, in order to perform a successful reconstruction, we must solve a nonlinear inverse problem described below.

The image formation theory in single-particle Cryo-EM asserts that each experimentally collected 2D image $\mathbf{b}_i$ $(i = 1, 2, \ldots, m)$ represents a projected view of the 3D electron density $\mathbf{x}$ along an unknown projection direction. The lack of projection direction information is a direct consequence of the way biological samples are prepared and how their images are taken. In single-particle Cryo-EM, an aqueous solution that contains purified macromolecule samples is spread over a carbon support grid. The grid is plunged into ethane at liquid nitrogen temperature. At such a low temperature, the solution substrate and the randomly oriented macromolecule samples it contains freeze rapidly. The frozen specimen is then placed in a transmission electron microscope (TEM) where it is bombarded by a low dose electron beam that ultimately produces a set of low-contrast and noisy 2D images on either a film or a CCD camera.

To reconstruct the 3D electron density map of the macromolecule from these 2D images, we must solve a nonlinear optimization problem in which the discrepancy between the collected images and the image formation model is minimized [136]. This problem can be viewed as a nonlinear least squares problem in which both the 3D density map and the unknown orientation parameters associated with each image are treated as decision variables, essentially the problem discussed in Chapter 3. Currently, the most widely used algorithm for solving this type of problem is the projection matching

algorithm [114]. The algorithm uses a generalized coordinate descent approach to identify the optimal orientation parameters and the 3D density in two alternating steps. In the first step, approximate orientation parameters are obtained through an exhaustive search in a 5D space of orientation and translational parameters using an initial guess of the 3D structure as a reference. This step is computationally costly, but can be easily parallelized. In the second step, a new 3D structure is reconstructed using the most recent orientation parameters. Although this step is generally faster than the first step on a single processor, it is more difficult to parallelize due to the collective communication required to merge 2D data in 3D. Overall, ten to twenty cycles of these two steps are typically required to obtain a satisfactory solution to the nonlinear least squares problem.

Because the signal-to-noise ratio (SNR) associated with each image $\mathbf{b}_i$ is extremely low (due to the low electron dose in the TEM to minimize radiation damage to the sample), a large number of images are required to boost the SNR of the reconstructed 3D density and obtain the required resolution. It has been estimated that to achieve atomic resolution, the number of 2D images required to perform a 3D reconstruction may be as many as one million. Furthermore, when each image is sampled with a small pixel size to enhance resolution, the dimension of each image can be quite large (e.g., hundreds to thousands of pixels in each direction). Consequently, both steps described above can be computationally demanding. Although the nonlinear variable projection algorithms described in Chapter 3 can be used for this application, this chapter focuses on the design and high-performance implementation of efficient and robust algorithms for the linear least squares problem discussed in Chapter 2. Equivalently, we focus on the second step of the projection matching algorithm and the linear subproblem in the variable projection algorithm. See Figure 5.1 for sample images from a Cryo-EM example.



(a) EM images (b) A 3D density map

Figure 5.1: Cryo-EM example. Provided with a set of projection images (a) that often have very low signal-to-noise ratio, the goal of single-particle Cryo-EM reconstruction is to recover a 3D density map such as the one shown in (b).

The 3D reconstruction problem to be solved in the second step of the nonlinear optimization procedure is a well studied problem in computerized tomography (CT) [99]. The two factors that distinguish single-particle Cryo-EM reconstruction from CT are the large volume of data involved and the fact that the orientation parameters associated with each image are not uniformly distributed. The weighted backprojection algorithm [99] often used in CT calculates optimal weights based on the assumption of uniformly distributed projection angles, making it inappropriate for Cryo-EM reconstruction.

When the dimension of each 2D image is relatively small, a rapid reconstruction can be obtained by using a direct Fourier inversion algorithm [115], a technique that is based on the central section theorem [100] and makes use of 2D and 3D fast Fourier transforms (FFTs). For 2D images that contain a larger number of pixels, the memory requirement of a direct Fourier inversion is quite substantial, and the inversion must be parallelized on a distributed memory parallel computer. However, because it is not easy to parallelize a 3D FFT efficiently on a distributed memory parallel computer due to its data access pattern and because the interpolation and weighting procedures used in Fourier inversion tend to introduce undesirable artifacts in the reconstructed density, it is not the preferred choice of method to use for high resolution reconstructions of large macromolecular structures.

In the next section we develop the mathematical framework for this problem and focus on iterative algorithms that construct and refine approximations to 3D macromolecular structures in real space. Cryo-EM is an inverse problem, so as discussed in Section 2.2, premature termination of standard iterative methods such as LSQR results in an over-smoothed solution and late termination can result in significant noise amplification in the computed solution. Choosing an optimal stopping criterion is therefore equivalent to choosing an optimal regularization parameter that allows as much information to be recovered as possible while minimizing the effect of noise amplification. The

hybrid methods that we described in Section 2.3 make such a task easier. One of our key contributions to this project is to apply the hybrid regularization algorithm to large-scale Cryo-EM reconstruction.

5.1.1 Mathematical Framework

Assume that each EM projection image $\mathbf{b}_i$ contains $n \times n$ pixels. Similarly, the sampled 3D density volume can be represented by $n \times n \times n$ voxels and can be denoted as $\mathbf{x}$. The finite-dimensional version of the projection operation along the i^{th} projection direction can be expressed as

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

where $\mathbf{A}_i$ is an $n^2 \times n^3$ matrix describing the projection operation. The matrix $\mathbf{A}_i$ depends on an Euler angle triplet $(\phi_i, \theta_i, \psi_i)$ that determines the direction of the projection as well as any in-plane rotation. Figure 5.2 gives a geometric view of the Euler angle convention used in our definition of the projector operation.

For ease of notation, let

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_m \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_m \end{bmatrix}. \tag{5.1}$$

Then, we can express the reconstruction problem as

$$\min_{\mathbf{x}} \rho(\mathbf{x}) \equiv ||\mathbf{A}\mathbf{x} - \mathbf{b}||^2\,. \tag{5.2}$$

Because of the large dimension of $\mathbf{A}$, iterative methods that do not require an explicit construction of $\mathbf{A}$ are desirable for solving (5.2). We briefly examine some of these methods in this section. All of these methods access $\mathbf{A}$ through matrix-vector multiplication subroutines that compute $\mathbf{w} \leftarrow \mathbf{A}\mathbf{x}$ and $\mathbf{v} \leftarrow \mathbf{A}^T\mathbf{b}$. We describe efficient implementations of these subroutines in Section 5.2.

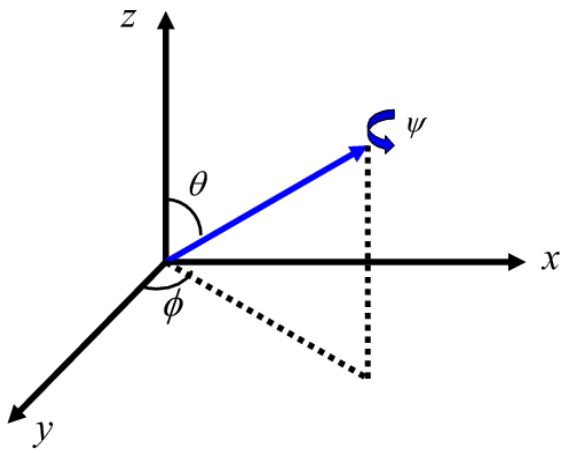


Figure 5.2: Euler angle convention. The blue arrow indicates the direction of the projection, which can be completely specified by a pair of angles (ϕ, θ).

5.1.2 Iterative Reconstruction Methods

In this section we consider algorithms for Cryo-EM reconstruction. A commonly used iterative reconstruction algorithm in the structural biology community is the simultaneous iterative reconstruction technique (SIRT) [49, 115]. The algorithm is essentially a steepest descent algorithm with a fixed line search parameter (step length).

Given an initial guess $\mathbf{x}_0$ (which could be zero), the 3D density of the macromolecule is updated iteratively as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda \nabla \rho(\mathbf{x}_k)\,, \tag{5.3}$$

where $\nabla \rho(\mathbf{x}_k) = \mathbf{A}^T(\mathbf{A}\mathbf{x}_k - \mathbf{b})$ and λ is a line search parameter chosen in advance by the user. It is well known that when $\mathbf{A}$ is ill-conditioned, the steepest descent search direction $-\nabla \rho(\mathbf{x}_k)$ results in slow convergence [104]. Furthermore, to delay the effect of semi-convergence, the step length

λ, which can also be considered as a regularization parameter, is often chosen to be a very small number. This further slows down the convergence of the reconstruction algorithm. In the existing software [45, 75], hundreds of SIRT iterations are typically required before high resolution features of the macromolecule emerge.

Since the problem is a linear least squares problem, we propose to use the Lanczos-hybrid methods developed in Chapter 2, more specifically, we use the HyBR implementation. Using these methods makes the regularization task easier by mitigating the semi-convergence behavior of standard iterative reconstruction algorithms. Consequently, an imprecise estimate of the number of iterations required in hybrid methods does not have a deleterious effect on the computed solution.

5.2 Large-Scale Implementation

Regardless of which iterative reconstruction algorithm is used, the key to achieving high performance in these algorithms is to develop an efficient implementation of the matrix-vector multiplications (MATVEC) $\mathbf{w} \leftarrow \mathbf{A}\mathbf{x}$ and $\mathbf{v} \leftarrow \mathbf{A}^T\mathbf{b}$. These operations correspond to finite dimensional approximations to the projection and backprojection calculations. They dominate the computational cost per iteration for all of the iterative methods considered here. Because image data is sampled on a finite number of grid points, projection and backprojection operations are approximated by finite sums of sampled data. Because different coordinate systems are used to sample 2D images and 3D density map, we must interpolate as we move from one coordinate system to another. We use linear interpolation in our implementation of the projection and backprojection operations (c.f. Section 3.1.1, [26]). As a result, the matrix $\mathbf{A}$ is extremely sparse. However, since the dimension of $\mathbf{A}$ is extremely large, we do not store the non-zero elements of $\mathbf{A}$ explicitly.

Instead, we generate these non-zeros on the fly from the orientation parameters $(\phi_i, \theta_i, \psi_i)$ associated with each image as they are used in the MATVEC subroutines.

Many proteins and viruses have a globular shape (i.e., their electron density fills up a globular spatial domain). Therefore, it is often convenient and efficient to enclose the 3D density map to be reconstructed within a spherical mask, and store and operate only on the density values associated with voxels within the mask. In the following section we discuss how to store a 3D density map $\mathbf{x}$ using a compact data structure that improves the data locality of the matrix-vector multiplications. We also discuss strategies for distributing 2D images and 3D density maps on multi-processor machines to achieve optimal parallel performance.

5.2.1 Compact Volume Representation

In many cases (for example, viruses and large macromolecular complexes), it is known in advance that the molecular structure to be determined has a globular shape. Efficiency can be gained by working with voxels that lie within a spherical mask with a radius $r < n/2$.

A straightforward implementation of the projection calculation would simply go through each voxel (i, j, k) and check whether it satisfies the condition:

$$(i - c_x)^2 + (j - c_y)^2 + (k - c_z)^2 \leq r^2, \tag{5.4}$$

where $c = (c_x, c_y, c_z)$ is the predefined center of the volume. Then only those that satisfy the above condition are projected onto a 2D image. The projected density values are accumulated only on pixels that are within a circular mask with the same radius r.

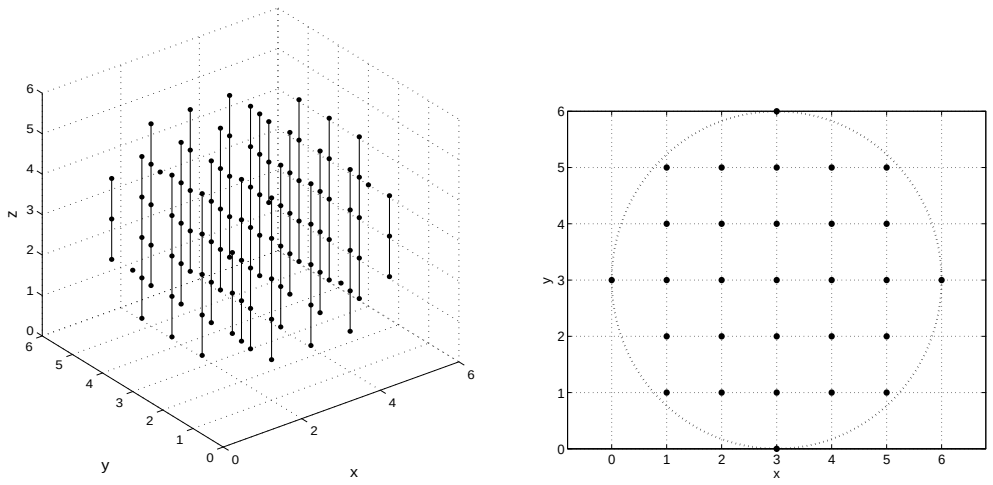
Because the projection and backprojection calculations are performed repeatedly in an iterative reconstruction algorithm, the extra conditional statement required to test (5.4) introduces significant overhead. As illustrated in

Section 5.3, branch mispredictions incurred by the condition statement and cache misses caused by fetching density values associated with non-adjacent voxels can lead to suboptimal performance. Furthermore, storing density values outside of the spherical mask increases the communication volume in the parallel implementation of the iterative reconstruction algorithm.

To achieve high performance, we use a compact data structure to store and work with only those density values defined at voxels within a spherical mask. To be specific, the density values defined at voxels that satisfy (5.4) are stored in a contiguous floating point array (`x`). The density values are arranged column by column, where each column is defined to be a set of sampling grid points (i, j, k) that share the same k value. As we can see from Figure 5.3(a), each column has a different number of sampling grid points. In our compact data structure, the starting location of each column within the array `x` is kept in a separate integer array (`colptr`) of size $n_{col} + 1$ where n_{col} is the total number of columns required to represent all voxels within the spherical mask. We use an additional array (`cord`) of size $n_{col} \times 3$ to keep track of the coordinates of the the first voxel in each column whose z-coordinate is minimal. Figure 5.3(b) shows the coordinates of the columns displayed in Figure 5.3(a). The pseudocode listed in Figure 5.4 illustrates how the desired density values and their 3D positions are retrieved in a contiguous fashion in our implementation of the projection and backprojection calculations.

5.2.2 Parallelization using 1D Data Distribution

The iterative reconstruction algorithms described in Section 5.1.2 can be easily parallelized on distributed memory parallel computers such as a Linux cluster if each processor has sufficient memory to store the 3D density map and other 3D data, such as $\nabla\rho$ and the i^{th} column of $\mathbf{Y}_k$ from the Lanczos bidiagonalization. Since the major computational cost of an iterative recon-



(a) The grid points within a spherical mask with radius $r = 3$.

(b) Top-down view of the sampling grid points

Figure 5.3: Compact volume representation of the 3D data.

```
for jcol = 1:ncol
   ix = cord(jcol,1);
   iy = cord(jcol,2);
   iz = cord(jcol,3);
   for i = colptr(jcol):colptr(jcol+1)-1
      dval = x(i);
      iz = iz + 1;
   endfor
endfor;
```

Figure 5.4: Retrieving density values and their coordinates using the compact data structure.

struction procedure is the matrix-vector multiplications associated with the projection and backprojection operation, parallelization of the reconstruction procedure essentially amounts to parallelizing these operations.

When 3D data such as $\mathbf{x}$, $\nabla\rho$ and the i^{th} column of $\mathbf{Y}_k$ can be replicated on each processor, the parallelization of iterative reconstruction algorithms can be achieved by simply distributing 2D images and their corresponding orientation and translation parameters evenly among different processors. Such a data distribution scheme allows us to partition the forward projection calculation with no inter-processor communication. For example, the residual vector $\mathbf{r}_i = \mathbf{A}_i\mathbf{x} - \mathbf{b}_i$ can be computed independently on the processor to which $\mathbf{b}_i$ is assigned. In order to compute the gradient $\nabla\rho$, we must perform the following backprojection operation:

$$\nabla\rho = \sum_{i=1}^{m} \mathbf{A}_i^T \mathbf{r}_i \,. \tag{5.5}$$

Because the residual images $\mathbf{r}_i$ for $i = 1, 2, \ldots, m$ are distributed on different processors, each processor can perform a partial sum of the right hand side of (5.5). In the 1D distribution scheme, these partial sums are collected, accumulated and broadcast back to all processors using the `MPI_Allreduce` function.

To assess the efficiency of the parallelization schemes, we measure the ratio between the number of floating point operations executed and the total amount of data transferred among different processors. When images are distributed among processors, the total number of floating point operations required to compute $\nabla\rho$ is $O(mn^3)$, when linear interpolation is used for forward and backward projection calculations. The total amount of data transfer required is $O(n^3 n_p)$, where n_p is the number of processors. Thus, the ratio of floating point operations (flop) to the volume of data communication is

$$\tau = \frac{O(mn^3)}{O(n^3 n_p)} = O\left(\frac{m}{n_p}\right) \,. \tag{5.6}$$

As clearly seen from (5.6), τ is independent of the image size n. However, if m remains unchanged, τ becomes smaller as we increase n_p. This suggests that the parallel performance will eventually deteriorate due to the increased amount of communication overhead as we add more processors in the gradient calculation.

5.2.3 Parallelization using 2D Data Distribution

Digitizing electron microscopy images with a small pixel size (which is required for high resolution reconstruction) can lead to images with large dimensions, especially for large macromolecular complexes and viruses. It is not uncommon these days to collect virus images with 512×512 pixels. In some cases, it has been estimated that an image size of 1024×1024 pixels may be required to determine 3D structures of viruses (sampled with 1Å per pixel or even smaller pixel sizes) at higher than 4Å resolution [119].

For extremely large structures, it is impossible to replicate the entire 3D volume on each processor due to the memory limitations of most distributed memory parallel computers. In this case, a parallelization mechanism that allows 3D volume data to be distributed among different processors is desirable. Furthermore, distributing 3D volume data enables us to exploit parallelism more effectively in the projection and backprojection calculation on state-of-the-art high-performance computers equipped with tens of thousands of processors.

In this section we describe a parallel implementation of the iterative reconstruction algorithms that can overcome the existing memory limitation problem. We discuss how the projection and backprojection operations with the new data distribution scheme map naturally onto a Cartesian topology, allowing us to take advantage of easy-to-use, built-in MPI virtual topologies. We show that this implementation not only allows reconstruction of large-

volume structures by addressing the memory limitation problem, but also yields better parallel scalability by improving the floating point operations to communication ratio.

The new data distribution scheme requires the processors to be grouped according to a particular 2D layout (see Figure 5.5). Assume that the total

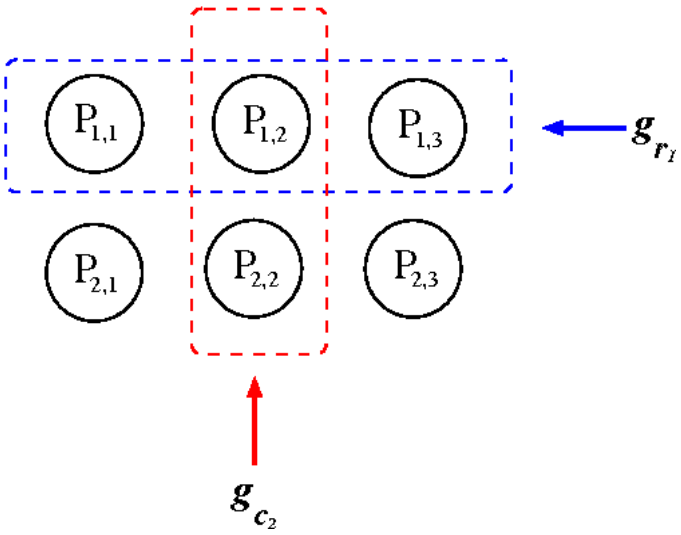


Figure 5.5: Processor layout for volume data distribution. The six processors $p_{i,j}$, for $i = 1, 2, 3$ and $j = 1, 2, 3$, shown here are grouped by row and column. Each processor is identified by its row and column group numbers.

number of processors n_p can be factored as $n_p = n_r \times n_c$, where n_r and n_c correspond to the number of rows and columns in the 2D processor grid, respectively. Let g_{r_i} denote the processor group that consists of processors $p_{i,1}, p_{i,2}, \ldots, p_{i,n_c}$, and let g_{c_j} denote the processor group that consists of processors $p_{1,j}, p_{2,j}, \ldots, p_{n_r,j}$. We remark here that the above notation is purely for convenience. In actual implementations, we use the MPI virtual Cartesian topology functionality, which simplifies coding and allows the MPI library to assign Cartesian coordinates to processors in such a way that automatically

takes best advantage of the underlying network.

Two-dimensional Cartesian decomposition can be easily created using the function `MPI_Cart_create`, and corresponding row and column communicators can be created using the `MPI_Cart_sub` function. To compute the residual images using this topology, we distribute the calculation over the processor row groups $g_{r_1}, g_{r_2}, \ldots, g_{r_{n_r}}$. Thus, all processors within the same row group receive replicates of roughly m/n_r 2D images and the orientation and translation parameters that define their corresponding projection operators. The 3D volume data $\mathbf{x}$ and gradient $\nabla\rho$, in their compact spherical representation, are divided as evenly as possible into n_c sub-vectors in the following manner:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \vdots \\ \mathbf{x}^{(n_c)} \end{bmatrix}, \quad \nabla\rho = \begin{bmatrix} \nabla\rho_{\mathbf{x}^{(1)}} \\ \nabla\rho_{\mathbf{x}^{(2)}} \\ \vdots \\ \nabla\rho_{\mathbf{x}^{(n_c)}} \end{bmatrix},$$

and these sub-vectors are distributed over the column groups $g_{c_1}, g_{c_2}, \ldots, g_{c_{n_c}}$ and replicated within each group. Thus, each processor only stores a partial volume in its memory, and the sub-vector $\mathbf{x}^{(j)}$ is replicated only on processors within the column group g_{c_j}. It also implies that each projection operator $\mathbf{A}_i$, which can be represented as a sparse matrix (with the same number of non-zeros in each column), is now partitioned as

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{A}_i^{(1)} & \mathbf{A}_i^{(2)} & \cdots & \mathbf{A}_i^{(n_c)} \end{bmatrix}.$$

Using this data distribution scheme, we can express each projection calculation as

$$\mathbf{A}_i\mathbf{x} = \sum_{j=1}^{n_c} \mathbf{A}_i^{(j)}\mathbf{x}^{(j)}. \tag{5.7}$$

Unlike the 1D partition, in which no communication is required in the projection calculation, (5.7) requires partial sums from different column groups

to be collected and summed within the row group g_{r_k} to which the i^{th} image has been assigned. The image sum must then be broadcast to all processors within g_{r_k}. By the same token, the backprojection calculation required in computing $\nabla\rho$ must be modified as well. Because $\mathbf{x}$ and $\nabla\rho$ are distributed among different processor column groups, each column group only needs to compute a sub-volume $\nabla\rho_{\mathbf{x}^{(j)}}$, which can be expressed as

$$\nabla\rho_{\mathbf{x}^{(j)}} = \sum_{i=1}^{m} (\mathbf{A}_i^{(j)})^T \mathbf{r}_i \,. \tag{5.8}$$

Because $\mathbf{r}_i$ and $\mathbf{A}_i^{(j)}$ are distributed among different processor row groups, each processor within the j^{th} column group can only perform a partial sum. These partial sums must be collected and added by a master processor within that column group. Then the result must be broadcast back to all processors within that column group.

Using the newly proposed data distribution scheme, the ratio of flops to communication data volume for the forward projection operation is

$$\tau_c = \frac{O(n^3)}{O(n^2 n_c)} = O\left(\frac{n}{n_c}\right). \tag{5.9}$$

Similarly, the ratio of flops over communication data volume for the back-projection operation is

$$\tau_r = \frac{O(mn^3/n_c)}{O(n_r n^3/n_c)} = O\left(\frac{m}{n_r}\right). \tag{5.10}$$

It follows from (5.9) and (5.10) that the overall scalability of the gradient calculation in the new parallel distribution scheme is determined by the minimum of $\frac{n}{n_c}$ and $\frac{m}{n_r}$. Because n is typically much smaller than m, we should choose $n_c < n_r$ in most cases. When n_r is sufficiently large (but still less than n), the 2D data distribution scheme may have a more favorable flop count to communication volume ratio.

For a given number of processors n_p, there can be several choices for n_c and n_r. However, it is easy to see from (5.9) and (5.10) that there is a trade-off between a larger value of n_c and n_r. If the memory available on each node for storing the volume and image data is M bytes and the total number of processors available is n_p, then n_c and n_r should be chosen such that the following constraints are satisfied:

$$\frac{4n^2m}{n_r} + \frac{4kn^3}{n_c} \leq M \tag{5.11}$$

$$n_c \cdot n_r \leq n_p\,, \tag{5.12}$$

where k is the number of 3D volumes that must be maintained in the iterative reconstruction algorithm.

The optimal choice of n_c and n_r (among all feasible choices) that leads to the minimum execution time of a 3D reconstruction depends on the values of n, m, and k, as well as on the communication bandwidth and latency associated with the user's computer cluster.

We should point out that the 2D data distribution scheme has a higher latency cost. In addition to performing a collective communication required by the backprojection operation among processors belonging to the same column group at each iteration, we must now also perform a global sum in the projection calculation among processors belonging to the same row group. Such a global sum operation must be performed for each projection direction. To reduce the latency cost, we can allocate more memory to hold the partially projected images until all projections assigned to a processor group are completed. In this case, only one extra collective communication needs to be performed at each iteration. The drawback is that the memory requirement for the image data increases by a factor of two, potentially increasing the number of processors needed to carry out the calculation.

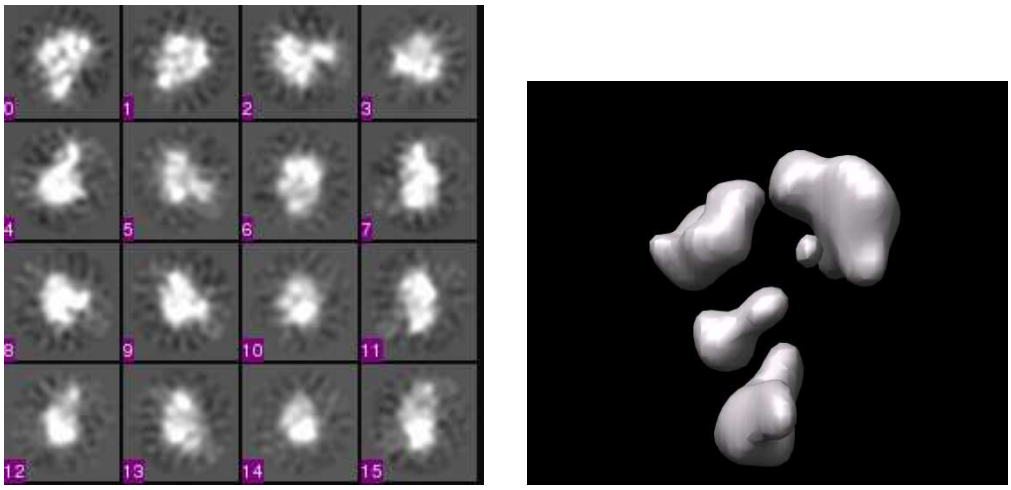
5.3 Numerical Results

In this section we present numerical results that illustrate the quality and performance of the iterative 3D reconstruction procedures described in Sections 5.1.2 and 2.3. In particular, we demonstrate that HyBR, our implementation of a Lanczos-hybrid with Tikhonov regularization method where the regularization parameter is chosen using the W-GCV method, allows us to obtain high quality 3D structures with a minimal number of iterations. We also show that using a compact volume representation reduces cache misses and branch mispredictions in the projection and backprojection calculations. We report the scalability of parallel reconstructions measured for both 1D and 2D data distribution schemes on a variety of datasets. We also show that distributing our data on a 2D Cartesian grid allows us to obtain 3D reconstructions of large virus structures in several minutes on 15,344 CPUs.

5.3.1 Quality of Iterative Reconstruction Algorithms

In this section we compare the numerical quality of the iterative reconstruction algorithms presented in Sections 5.1.2, 2.2 and 2.3, namely, SIRT, LSQR, and HyBR on a relatively small dataset. Two numerical experiments were performed. In the first experiment, synthetic image data was produced by projecting a previously reconstructed and low-pass filtered 3D density map of the TFIID protein [2] along 84 evenly distributed projection directions. Each image in our 2D image set $\{\mathbf{b}_i\}$, $i = 1, 2, \ldots, 84$, contains 64×64 pixels. We generated normally distributed ($\mathcal{N}(0, 1)$) random noise images $\boldsymbol{\varepsilon}_i$ to add to each image after scaling the noise level so that $\|\boldsymbol{\varepsilon}_i\|/\|\mathbf{b}_i\| = 0.1$ for all i. Given the set of noise-perturbed 2D projection images, some of which are shown in Figure 5.6(a), the goal is to reconstruct an approximation of the true volume shown in Figure 5.6(b).

The advantage of using synthetic data for testing is that we can examine



(a) Simulated 2D projection images (b) The true solution

Figure 5.6: Cryo-EM example. Sample simulated 2D projection images and the 3D density map used to generate the 2D data.

the accuracy of the reconstructed 3D density map by evaluating the relative error.

Figure 5.7 shows the relative error measured at each iteration of SIRT, LSQR, and HyBR. The dash-dotted line illustrates the slow convergence of the SIRT algorithm, compared to the Lanczos-based iterative methods. The regularization parameter λ in (5.3) was tuned by hand through trial and error. The optimal choice, which we used in the run that produced the convergence history curve shown in Figure 5.7, appears to be $\lambda = 10^{-3}$.

As described in Section 2.2.2, the convergence of LSQR exhibits typical semi-convergence behavior, where the approximate solution improves at early iterations, but eventually noise begins to enter and contaminate the computed reconstructions. For this problem, we would ideally like to terminate the process after 6 iterations, thereby giving us a regularized solution and avoiding error amplification. However, it is difficult to determine the iteration number that minimizes the relative error without knowing the true

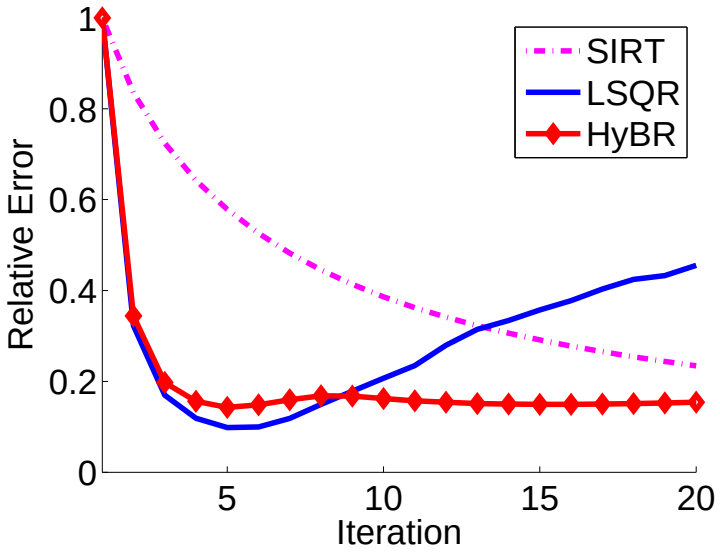


Figure 5.7: Cryo-EM: Relative error plot for synthetic data.

solution.

For the HyBR reconstruction, we use the W-GCV method to select regularization parameters. Then a suitable stopping iteration is found by monitoring a GCV function based on the original problem (see Section 2.4.5). Similar to LSQR, we see in Figure 5.7 that the relative errors for HyBR reconstruction decrease quickly within the first few iterations. However, since the noise is suppressed with regularization at every iteration, the hybrid method converges to a regularized solution rather than the inverse solution. This explains the absence of semi-convergence behavior, which is a benefit of using hybrid methods for computing Cryo-EM reconstructions.

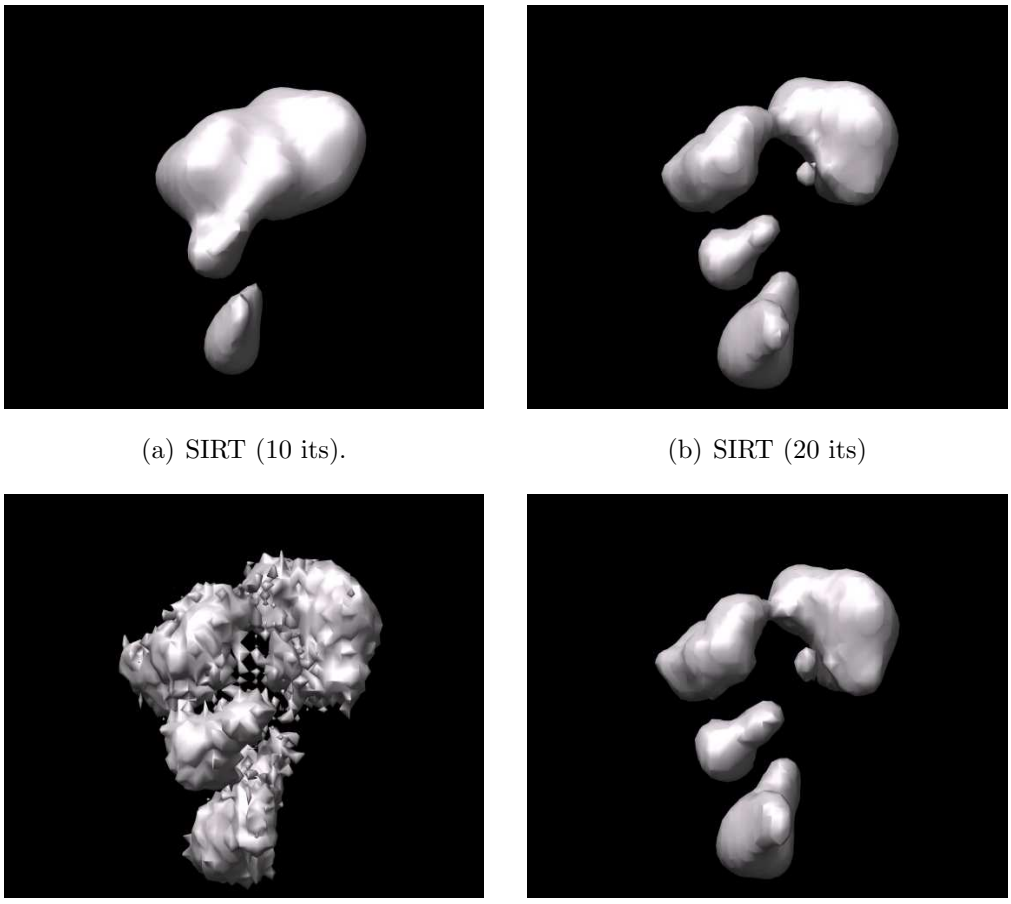
In Figure 5.8 we show the isosurface renderings of four reconstructed volumes produced by the three different iterative methods. The same threshold is used in all of the rendering plots. Figure 5.8(a) shows a reconstructed density map obtained after running 10 SIRT iterations. It is clear from this figure

that the method has not fully converged, resulting in an overly smooth solution. A much better reconstruction, which is obtained after running SIRT for 20 iterations, is shown in Figure 5.8(b). Figure 5.8(d) shows that an equally good reconstruction that is virtually indistinguishable from the true solution shown in Figure 5.6(b) is obtained by running only 9 iterations of HyBR. The noise amplification produced by running 20 iterations of LSQR can be clearly seen from Figure 5.8(c). All of these observations are consistent with the relative error plot shown in Figure 5.7. HyBR is clearly more efficient than SIRT in this case due to the smaller number of iterations required to obtain a good reconstruction. (Note that the cost per iteration is the same for all iterative methods considered here.)

A comparison of SIRT and CG-type methods such as LSQR can be found in [125]; however, it is important to note that both SIRT and LSQR have the disadvantage of requiring the user to select a stopping iteration. Previously proposed methods for terminating these iterations should be investigated for this problem. For HyBR, we use the stopping criteria developed in Section 2.4.5, which automatically terminated the process after 9 iterations for this particular example.

In the second experiment, we applied HyBR to real Cryo-EM data produced in [2] to determine the 3D density map of the TFIID protein shown in Figure 5.1(b) and to compare the quality of the reconstruction with those obtained from SIRT, LSQR and the direct Fourier reconstruction method described in [115].

The dataset contains 4418 2D images collected from a TEM. Some of these images are shown in Figure 5.1(a). The initial Euler angles we used in the reconstruction were obtained from the 10^{th} iteration of a previous projection matching run that was used to produce the 3D density map shown in Figure 5.1(b). In that projection matching run, the 3D reconstruction problem (5.2) was solved by running 100 iterations of SIRT.



(a) SIRT (10 its).

(b) SIRT (20 its)

(c) LSQR (20 its)

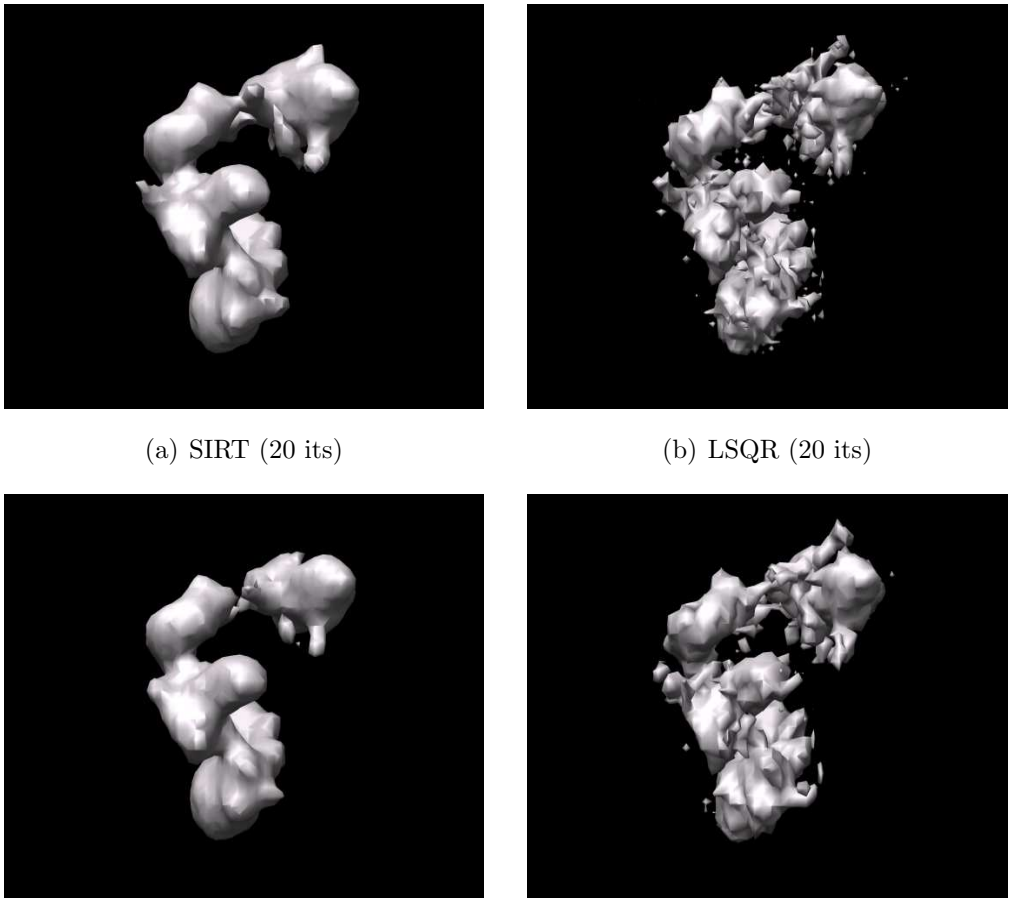
(d) HyBR (9 its)

Figure 5.8: Cryo-EM: Reconstructed 3D structures from synthetic data. The true solution is shown in Figure 5.6(b).

Without knowing the true solution to the inverse problem, it is difficult to assess the true quality of a reconstructed 3D density map. In Figure 5.9, we show the isosurface renderings of reconstructed density maps produced by the three different iterative methods as well as a 3D density map produced by the direct Fourier inversion algorithm described in [115]. It is clear from this figure that both SIRT and HyBR produced 3D density maps that contain all visible features shown in Figure 5.1(b), whereas both LSQR (20 iterations) and direct Fourier inversion produced 3D maps that contain a significant amount of noise and artifact. The HyBR run was terminated at the 10^{th} iteration, and hence is more efficient than SIRT. We should comment that similar convergence behavior was observed for other datasets including the ones listed in Table 5.2. That is, HyBR typically terminates automatically in at most 10 iterations. To achieve the same reduction in the objective function (5.2), we often need to run more than 20 SIRT iterations. Terminating SIRT sooner often produces an over-smoothed reconstruction. The exact number of iterations depends on the SNR of the image and the choice of line search parameter used in SIRT. Although the objective function (5.2) decreases rapidly when using LSQR, noise contamination occurs quickly also. Earlier termination would have produced nice solutions, but selecting a good stopping point is not obvious.

5.3.2 Single Processor Performance

In this section we report the performance of the LBD algorithm when it is executed on a single processor. The single processor performance is measured in terms of the total CPU time used to run 5 LBD iterations, the number of floating point operations performed per second, as well as the number of level-2 (L2) cache misses and branch mispredictions measured in the run. These measurements are collected by the PAPI [95] performance measurement tool.



(a) SIRT (20 its)

(b) LSQR (20 its)

(c) HyBR (10 its)

(d) Fourier reconstruction

Figure 5.9: Cryo-EM: Reconstructed 3D structures from real data. Figure 5.1(b) shows a reconstruction obtained from running 100 iterations of SIRT.

The TFIID dataset that was used in the second experiment in Section 5.3.1 was used for this performance study, which was carried out on an AMD Opteron processor that runs at a clock speed of 2.2 GHz, has a 2 MB L2 cache, and has peak performance of 4.4 gigaflops/sec (Gflops). We ran two different implementations of the LBD algorithm. In the first implementation, the compact volume representation (CVR) was used to store the 3D data ($\mathbf{x}$ and $\mathbf{Y}_k$). In the second implementation, $\mathbf{x}$ and columns of $\mathbf{Y}_k$ are stored as standard 3D arrays. In both implementations, projection and backprojection operations only make use of voxels that lie within a sphere of predefined radius r; thus, they perform the same number of floating point operations.

Table 5.1: Performance characteristics of two different implementations of the LBD algorithm.

Performance characteristics	CVR	3D array
CPU time (seconds)	281	565
flops/second	$1,402 \times 10^6$	804×10^6
L2 cache misses	15,601,052	2,293,781,072
branch misprediction	130,801,673	1,215,150,812

Table 5.1 shows that the use of CVR improves the performance of LBD by a factor of two compared to a simple implementation in which 3D data objects are stored as 3D arrays. It allows the projection and backprojection calculations to run at 31% of the peak performance. The superior performance exhibited by the CVR version of the LBD run is due to the much smaller number of L2 cache misses and branch mispredictions shown in the third and fourth rows of the table respectively.

5.3.3 Parallel Performance

In this section we report the performance of the parallel implementations of the reconstruction algorithms discussed earlier. All of these algorithms perform the same projection and backprojection calculations in each iteration. Since these calculations constitute a significant portion of the computational cost per iteration, the parallel performance characteristics should be similar.

Three different datasets are used in the following experiments. The characteristics of these datasets are described in Table 5.2. Note that the virus data, which is a subset of the data used in [119], contains only 959 images. However, since the structure of the virus is known to have icosahedral symmetry, each image is backprojected 60 times from distinct but symmetrically related projection directions. That is, each image provides information equivalent to 60 images of a particle with no symmetry, making the effective image count 57,540.

Our parallel scalability analysis is performed on the Franklin cluster maintained at NERSC. Franklin is a distributed-memory parallel system with 9,660 compute nodes. Each compute node consists of a 2.6 GHz dual-core AMD Opteron processor with a theoretical peak performance of 5.2 Gflops. Each compute node has 4 GBytes of memory. Each compute node is connected to a dedicated SeaStar2 router through Hypertransport with a 3D torus topology that ensures high performance, low-latency communication for MPI.

When 1D data distribution is used in the parallel implementation of the reconstruction algorithm, a collective communication involving all processors is performed at each iteration. The analysis we showed in Section 5.2 indicates that the parallel scalability of the reconstruction, which we measure by

$$s = \frac{T_1}{T_{n_p}}\,, \tag{5.13}$$

Table 5.2: Cryo-EM: Datasets used in the performance analysis

Target structure	image size	number of images
TFIID	64×64	4418
Ribosome	150×150	150,000
Adenovirus	500×500	959 ($\times 60$)

where T_{n_p} is the wall clock time required to run 10 iterations of the LBD procedure in parallel on n_p processors, will eventually degrade as n_p increases. This phenomenon can be clearly observed from Table 5.3 in which we show the speedup factor s for both the TFIID and the Ribosome datasets. For the TFIID dataset, which was used in [2], linear speedup is observed for parallel runs with up to 256 processors. A slight departure from linear speedup can be observed when the total number of processors used reaches 512.

For the Ribosome dataset, which contains 150,000 images, a minimum of 128 processors are used in the performance evaluation. Therefore, our definition of the speedup factor is modified to become $s = T_{128}/T_{n_p}$. The parallel LBD calculation scales linearly up to 4096 processors. A slight departure from linear scalability is observed when 8192 processors are used. Compared to the TFIID dataset, the increased number of images (m) in the Ribosome dataset leads to a higher flop-to-communication volume ratio τ (defined in equation (5.6)) for a fixed number of processors. As a result, linear scalability extends to a larger processor count.

To measure how parallel reconstruction performs when both the 2D images and the 3D data are distributed on a 2D processor grid, we run 10 iterations of HyBR with several choices of n_r and n_c such that $n_r \times n_c = n_p \leq 128$. Table 5.4 shows the wall clock time used in the reconstruction of the TFIID structure for each (n_r, n_c) pair. We clearly observe from this table that the parallel reconstruction scales linearly with respect to the number of column

(a) Speedup of the TFIID reconstruction.

n_p	T_1/T_{n_p}
2	2
4	4
8	8.2
16	16.4
32	33.1
64	66.3
128	130.7
256	255.8
512	489.9
1024	636.4

(b) Speedup of the Ribosome reconstruction.

n_p	$n_p/128$	T_{128}/T_{n_p}
256	2	2.0
512	4	3.9
1024	8	7.7
2048	16	15.6
4096	32	30.9
8192	64	61.4

Table 5.3: Speedup of the reconstruction for both the TFIID and Ribosome datasets when 2D images are distributed on a 1D processor grid. The speedup of the TFIID reconstruction is measured by T_1/T_{n_p}, whereas the speedup of the Ribosome reconstruction is measured by T_{128}/T_{n_p}.

groups n_c, when the number of row groups n_r is fixed. A similar observation can be made for fixing n_c and increasing n_r. However, for a fixed number of processors, different combinations of n_r and n_c result in slightly different performance, as we can see along the diagonals of the table.

Table 5.4: Wallclock time (seconds) used to reconstruct the 3D density of the TFIID molecule on a processor grid with $n_r \times n_c$ processors

		n_c							
		1	2	4	8	16	32	64	128
n_r	128	4.652	*	*	*	*	*	*	*
	64	9.261	**4.635**	*	*	*	*	*	*
	32	18.07	**9.255**	4.743	*	*	*	*	*
	16	35.91	18.18	9.582	5.003	*	*	*	*
	8	71.95	36.22	18.77	9.997	5.588	*	*	*
	4	143.0	72.48	37.42	19.93	11.44	6.824	*	*
	2	286.0	144.9	74.65	39.38	22.05	13.41	9.160	*
	1	570.8	289.9	149.9	78.76	43.55	26.73	18.07	13.89

To explain this observation, we recall from Section 5.2.3 that the ratio of flops to communication data volume for parallel reconstructions performed on a 2D processor grid with $n_r \times n_c$ processors is determined by the minimum of n/n_c and m/n_r, where n is the image size and m is the total number of images. Since $m/n_r \geq m/(n_r n_c)$, the performance of the 2D data distributed parallel calculation can exceed that of the 1D distributed calculation (i.e., $n_c = 1$) only when $m/n_r < n/n_c$, or equivalently, when $n_r/n_c > m/n$. Because the number of images in the TFIID dataset ($m = 4418$) is almost two orders of magnitude larger than the image size $n = 64$, we do not expect the 2D data distribution scheme to outperform the 1D data distribution scheme unless n_r is roughly two orders of magnitude larger than n_c. This is indeed the case as

we can see that parallel reconstructions carried out on the 64×2 and 32×2 grids slightly outperform those carried out on a 1D processor grid with 128 and 64 processors respectively.

Similarly, because $m/n = 1000$ for the the Ribosome data set, we do not expect to see the benefits of the 2D data distribution scheme when n_r/n_c is significantly less than 1000. This is confirmed by the timing results reported in Table 5.5.

Table 5.5: Wallclock time (seconds) used to reconstruct the 3D density of the Ribosome molecule on a processor grid with $n_r \times n_c$ processors

		n_c			
		1	2	4	8
n_r	8192	68.2	*	*	*
	4096	138.6	69	*	*
	2048	271.7	**133.8**	**68.1**	*
	1024	526.7	265.6	**133.8**	69
	512	1050	532.7	267.8	136
	256	1998	1060	535.2	270.5
	128	3905	2112	1069	547.4

However, for the Adenovirus dataset, which contains 959 images, each with 500×500 pixels, there is a clear advantage to using a 2D data distribution. Storing 500^3 voxels of single precision density values takes roughly 500 MB of memory. Even if we use a compact volume representation scheme, the memory requirement for a single 3D structure is still larger than 250 MB. Therefore, replicating $\mathbf{x}$ and $\mathbf{Y}_k$ on all processors in a parallel HyBR run that distributes only 2D images along a 1D processor grid is not feasible for moderately large values of k (e.g., $k = 10$).

Even if there is enough memory to hold $\mathbf{x}$ and $\mathbf{Y}_k$ on each node, the maxi-

mum level of parallelism one can exploit from the 1D data distribution scheme is limited by the number of images, since at least one image must be assigned to each processor. The 2D data distribution scheme enables us to utilize more processors when they are available.

In Table 5.6, we report how the parallel LBD computation scales as we add more processors to the 2D processor grid. We use the wall clock time required to complete 20 LBD iterations on a 137×7 processor grid as T_1 in (5.13) for the speedup factor calculation. Table 5.6 shows that we were able to run 10 iterations of the parallel LBD reconstruction on as many as 15,344 (959×16) processors. The total amount of wall clock time required to perform such a calculation is roughly 10 minutes. Because Table 5.6 shows that the speedup of the parallel LBD reconstruction is almost linear, we estimate that running the same calculation on a single processor with sufficient memory would take 106 days.

Table 5.6: Wallclock time (seconds) used to reconstruct the 3D density of the Adenovirus on a processor grid with $n_r \times n_c$ processors

n_r	n_c	wall clock (sec)	speedup
137	7	9635	1
959	2	4841	2
959	4	2406	4
959	8	1335	7.2
959	16	609	15.8

5.4 Research Impact

We have developed a high-performance iterative reconstruction method for estimating the 3D electron density map of a macromolecule from a large num-

ber of 2D Cryo-EM images. Our approach uses the Lanczos-hybrid bidiagonalization regularization algorithm, HyBR, from Section 2.3 that allows us to stabilize the reconstruction process and accurately compute a reconstruction. Our parallel implementation of the iterative reconstruction algorithm utilizes a 2D data distribution scheme. It allows both the 2D image data and the 3D data such as the density map and the right vectors from the Lanczos bidiagonalization to be distributed among different processors, thereby overcoming the potential memory limitation on a cluster of commodity processors. We demonstrated that our parallel implementation of the iterative reconstruction algorithm scales up to tens of thousands of processors.

We should point out that the 2D parallelization strategy presented here can also be utilized in other reconstruction algorithms. For example, a quasi-Newton method referred to as the unified approach in [136] that simultaneously seeks the optimal 3D density and orientation parameters by minimizing a nonlinear least squares objective function can benefit from this implementation. The variable projection method from Chapter 3 can also be used. These methods have been shown to work well when a good initial guess is available. We should also mention that another popular iterative method for solving the linear reconstruction problem (5.2) is the Algebraic Reconstruction Technique (ART) [72, 77] and its block variants [94]. However, contrary to SIRT and the Lanczos-hybrid algorithms where the projection and backprojection operations can be computed for all 2D images simultaneously, ART and Block ART require a sequential use of only one 2D image per iteration. It has been shown that the parallel performance of SIRT is generally superior to that of Block ART on distributed architectures [93]. Furthermore, it may be difficult to achieve scalability when more than one hundred processors are used. More comparisons between parallel implementations of ART and Lanczos bidiagonalization based iterative methods will be pursued in our future work.

Chapter 6

Concluding Remarks

In this dissertation we presented some significant mathematical results, utilized high-performance computing capabilities for large-scale implementation, and contributed to scientific advancement in a variety of applications. We considered three different mathematical models that frequently arise in imaging applications. The common thread in all of the examples was the need for efficient regularization and robust implementation for large-scale ill-posed inverse problems.

The *mathematical contributions* from this work include developing regularization approaches for linear and nonlinear least squares problems and deriving numerical methods for nonlinear Poisson-based maximum likelihood problems. For the linear least squares problem, we developed an adaptive approach for selecting parameters in a standard Tikhonov framework, showed how it can be effectively used in an iterative hybrid bidiagonalization regularization (HyBR) method, and produced a user-friendly set of MATLAB codes for software distribution. For the nonlinear least squares problem, we adapted a variable projection approach to use HyBR to solve the linear problem at each nonlinear iteration. This in turn allowed us to efficiently incorporate robust methods for regularization. A different mathematical model that assumed a nonlinear Poisson distribution was also considered in this dissertation. A new mathematical framework was developed in the context of

breast tomosynthesis, and standard optimization methods were made feasible for the maximum likelihood formulation. To summarize, a variety of mathematical problems were considered in this work, and progress was made on many fronts.

Oftentimes it takes advanced computational capabilities to make the mathematical contributions significant in real-life applications. That is, high-performance computing is important for efficient large-scale implementations. The significant *computational contributions* from this research include a massively parallel code for large-scale image reconstruction. Using the MPI library, we implemented a 2D data distribution for use on multi-processor computers. The codes have been included in the publicly available software package called SPARX and have allowed reconstructions that were previously not possible.

The *scientific contributions* are evident in the variety of imaging applications that have benefitted from our work. In particular, the separable nonlinear least squares framework arises naturally in super-resolution imaging, blind deconvolution and Cryo-EM reconstruction. All of these applications, in addition to standard image deblurring, have benefitted from the development of HyBR and the large-scale implementations. With new algorithms for polyenergetic tomosynthesis reconstruction, significant advancements in breast imaging will hopefully lead to better detection of breast abnormalities. Although this dissertation focuses on imaging applications, ill-posed inverse problems arise in many other scientific applications, and the numerical methods developed here can promote progress and development in other fields as well.

Appendix

Here we provide further details and derivations for some of the problems discussed in this dissertation. In particular, we provide the detailed derivation of the weighted-GCV function as a "weighted leave-one-out" approach in Section A.1. Then in Section A.2 we provide details on how we select the ω parameter for the W-GCV function. Finally, the derivation of the conditions under which the polyenergetic tomosynthesis cost function is convex is presented in Section A.3.

A.1 Weighted-GCV

As mentioned in Section 2.4.2, W-GCV can be interpreted as a "weighted leave-one-out" approach. In this section we provide more of the mathematical details for deriving the proposed weighted-GCV function. Our derivation of the weighted-GCV function (2.17) follows a similar derivation for the cross-validation and generalized cross-validation function found in Golub, Heath and Wahba [53].

Preliminary Results from Cross-Validation

We begin by defining the matrix

$$\mathbf{E}_j = \text{diag}(1, 1, ...1, 0, 1, ...1)\,,$$

where 0 is the j^{th} entry. Then let $\mathbf{x}_{\lambda,j}$ be the solution of the following minimization problem:

$$\min_{\mathbf{x}} ||\mathbf{b}^{(j)} - \mathbf{A}^{(j)}\mathbf{x}||_2^2 + \lambda^2||\mathbf{x}||_2^2\,,$$

where vector $\mathbf{b}^{(j)}$ is $\mathbf{b}$ with the j^{th} entry missing and matrix $\mathbf{A}^{(j)}$ is $\mathbf{A}$ with the j^{th} row missing. Hence, the above minimization problem is equivalent to the following problem:

$$\min_{\mathbf{x}} ||\mathbf{E}_j(\mathbf{b} - \mathbf{A}\mathbf{x})||_2^2 + \lambda^2 ||\mathbf{x}||_2^2 \,. \tag{A-1}$$

The cross-validation method is based on the concept of prediction error. In the derivation of the cross-validation approach, consider $\mathbf{x}_{\lambda,j}$ as defined above. Then consider the j^{th} entry of the residual vector:

$$b^{(j)} - [\mathbf{A}\mathbf{x}_{\lambda,j}]^{(j)} \,,$$

where $b^{(j)}$ and $[\mathbf{A}\mathbf{x}_{\lambda,j}]^{(j)}$ are the j^{th} entries of vectors $\mathbf{b}$ and $\mathbf{A}\mathbf{x}_{\lambda,j}$ respectively. Thus, we can define the average error as

$$V(\lambda) = \frac{1}{m}\sum_{j=1}^{m}(b^{(j)} - [\mathbf{A}\mathbf{x}_{\lambda,j}]^{(j)})^2 \,.$$

The value of λ that minimizes $V(\lambda)$ is called the "cross-validation" estimate for λ. The derivation of the cross-validation method, and hence the generalized cross-validation method, is based on the following theorem, which is proved in [53].

Theorem A.1 *We can write* $V(\lambda) = \frac{1}{m}||\mathbf{D}(\lambda)(\mathbf{I} - \mathbf{T}(\lambda))\mathbf{b}||_2^2$, *where*

$$\mathbf{T} = \mathbf{A}(\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{I})^{-1}\mathbf{A}^T$$

and

$$\mathbf{D} = \text{diag}\left(\frac{1}{1 - t^{(jj)}}\right) \,,$$

where $t^{(jj)}$ *is the* jj^{th} *entry of matrix* $\mathbf{T}$.

Next we follow a similar derivation for weighted-cross-validation.

Derivation of Weighted-Cross-Validation

We produce an analogous average error function for weighted-cross-validation. To proceed, we prove the following theorem:

Theorem A.2 *We can write the average error* $V_\omega(\lambda) = \frac{1}{m}||\widehat{\mathbf{D}}(\lambda)(\mathbf{I}-\mathbf{T}(\lambda))\mathbf{b}||_2^2$, *where*

$$\mathbf{T} = \mathbf{A}(\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{I})^{-1}\mathbf{A}^T$$

and

$$\widehat{\mathbf{D}} = \text{diag}\left(\frac{1}{1-\omega t^{(jj)}}\right).$$

Notice that the only difference between the results in Theorem A.1 and A.2 is the matrix $\widehat{\mathbf{D}}$, where $\widehat{\mathbf{D}}$ incorporates the new weighting parameter ω. Furthermore, if $\omega = 1$ in Theorem A.2, then we get the result in Theorem A.1.

To prove Theorem A.2, we proceed in four steps:

1. First we find an expression for $\mathbf{x}_{\lambda,j}$.
2. Next we compute the vector $\mathbf{A}\mathbf{x}_{\lambda,j}$.
3. Then we evaluate $[\mathbf{A}\mathbf{x}_{\lambda,j}]^{(j)}$.
4. Finally we get an expression for $b^{(j)} - [\mathbf{A}\mathbf{x}_{\lambda,j}]^{(j)}$ and evaluate

$$V_\omega(\lambda) = \frac{1}{m}\sum_{j=1}^{m}(b^{(j)} - [\mathbf{A}\mathbf{x}_{\lambda,j}]^{(j)})^2.$$

Proof. **Step 1:** Without loss of generality, let's assume $0 < \omega < 1$. Then define the matrix

$$\mathbf{F}_j = \text{diag}(1, 1, ...1, \sqrt{1-\omega}, 1, ...1),$$

where $\sqrt{1-\omega}$ is the j^{th} entry. Our goal is to find $\mathbf{x}_{\lambda,j}$ that solves the following minimization problem:

$$\min_{\mathbf{x}} ||\mathbf{F}_j(\mathbf{b}-\mathbf{A}\mathbf{x})||_2^2 + \lambda^2||\mathbf{x}||_2^2 \,. \tag{A-2}$$

By equivalent formulations of Tikhonov regularization (2.4), $\mathbf{x}_{\lambda,j}$ also solves the following problem:

$$\min_{\mathbf{x}} \left|\left|\left[\begin{array}{c} \mathbf{F}_j\mathbf{A} \\ \lambda\mathbf{I} \end{array}\right]\mathbf{x} - \left[\begin{array}{c} \mathbf{F}_j\mathbf{b} \\ 0 \end{array}\right]\right|\right|_2 .$$

Then the normal equations can be written as

$$(\mathbf{A}^T\mathbf{F}_j^T\mathbf{F}_j\mathbf{A} + \lambda^2\mathbf{I})\mathbf{x}_{\lambda,j} = \mathbf{A}^T\mathbf{F}_j^T\mathbf{F}_j\mathbf{b} \,, \tag{A-3}$$

and an explicit expression for $\mathbf{x}_{\lambda,j}$ can be written as

$$\mathbf{x}_{\lambda,j} = (\mathbf{A}^T\mathbf{F}_j^T\mathbf{F}_j\mathbf{A} + \lambda^2\mathbf{I})^{-1}\mathbf{A}^T\mathbf{F}_j^T\mathbf{F}_j\mathbf{b} \,. \tag{A-4}$$

The following two relations will be important for future use:

1. $\mathbf{F}_j = \mathbf{I} - (1-\sqrt{1-\omega})\mathbf{e}_j\mathbf{e}_j^T$
2. $\mathbf{F}_j^T\mathbf{F}_j = \mathbf{I} - \omega\mathbf{e}_j\mathbf{e}_j^T$

where $\mathbf{e}_j$ is the j^{th} column of the identity matrix.

Using the second relationship, we can rewrite the coefficient matrix in equation (A-3) as

$$\begin{aligned} \mathbf{A}^T\mathbf{F}_j^T\mathbf{F}_j\mathbf{A} + \lambda^2\mathbf{I} &= \mathbf{A}^T(\mathbf{I} - \omega\mathbf{e}_j\mathbf{e}_j^T)\mathbf{A} + \lambda^2\mathbf{I} \\ &= (\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{I}) - (\sqrt{\omega}\mathbf{A}^T\mathbf{e}_j)(\sqrt{\omega}\mathbf{e}_j^T\mathbf{A}) \,. \end{aligned}$$

Now let $\mathbf{a}_j^T = \mathbf{e}_j^T\mathbf{A}$ be the j^{th} row of $\mathbf{A}$.

Then

$$(\mathbf{A}^T\mathbf{F}_j^T\mathbf{F}_j\mathbf{A} + \lambda^2\mathbf{I})^{-1} = (\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{I} - (\sqrt{\omega}\mathbf{a}_j)(\sqrt{\omega}\mathbf{a}_j)^T)^{-1} \,.$$

Let's define

$$\mathbf{T}(\lambda) = \mathbf{A}(\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{I})^{-1}\mathbf{A}^T\,.$$

Then by the Sherman-Morrison-Woodbury formula and with some algebra, we can see that $\mathbf{x}_{\lambda,j}$ from equation (A-4) can be written as

$$\mathbf{x}_{\lambda,j} = \frac{1}{1-\omega t^{(jj)}}[(1-\omega t^{(jj)})\mathbf{A}_\lambda^\dagger + \omega\mathbf{A}_\lambda^\dagger\mathbf{e}_j\mathbf{e}_j^T\mathbf{T}]\mathbf{F}_j^T\mathbf{F}_j\mathbf{b}\,, \tag{A-5}$$

where $\mathbf{A}_\lambda^\dagger$ is defined in Chapter 2 as $\mathbf{A}_\lambda^\dagger = (\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{I})^{-1}\mathbf{A}^T$.

Step 2: From equation (A-5), we can get an expression for $\mathbf{A}\mathbf{x}_{\lambda,j}$:

$$\begin{aligned}
\mathbf{A}\mathbf{x}_{\lambda,j} &= \frac{1}{1-\omega t^{(jj)}}[(1-\omega t^{(jj)})\mathbf{T} + \omega\mathbf{T}\mathbf{e}_j\mathbf{e}_j^T\mathbf{T}]\mathbf{F}_j^T\mathbf{F}_j\mathbf{b} && \text{(A-6)}\\
&= \frac{1}{1-\omega t^{(jj)}}[(1-\omega t^{(jj)})\mathbf{I} + \omega\mathbf{T}\mathbf{e}_j\mathbf{e}_j^T]\mathbf{T}\mathbf{F}_j^T\mathbf{F}_j\mathbf{b}\,. && \text{(A-7)}
\end{aligned}$$

Step 3: Furthermore, we can evaluate $[\mathbf{A}\mathbf{x}_{\lambda,j}]^{(j)}$:

$$\begin{aligned}
[\mathbf{A}\mathbf{x}_{\lambda,j}]^{(j)} &= \mathbf{e}_j^T\mathbf{A}\mathbf{x}_{\lambda,j}\\
&= \frac{1}{1-\omega t^{(jj)}}[(1-\omega t^{(jj)})\mathbf{e}_j^T + \omega\mathbf{e}_j^T\mathbf{T}\mathbf{e}_j\mathbf{e}_j^T]\mathbf{T}\mathbf{F}_j^T\mathbf{F}_j\mathbf{b}\\
&= \frac{1}{1-\omega t^{(jj)}}[\mathbf{e}_j^T - \omega t^{(jj)}\mathbf{e}_j^T + \omega t^{(jj)}\mathbf{e}_j^T]\mathbf{T}\mathbf{F}_j^T\mathbf{F}_j\mathbf{b}\\
&= \frac{1}{1-\omega t^{(jj)}}\mathbf{e}_j^T\mathbf{T}\mathbf{F}_j^T\mathbf{F}_j\mathbf{b}\,.
\end{aligned}$$

Step 4: Finally, notice that

$$b^{(j)} - [\mathbf{A}\mathbf{x}_{\lambda,j}]^{(j)} = \mathbf{e}_j^T\mathbf{b} - [\mathbf{A}\mathbf{x}_{\lambda,j}]^{(j)}\,,$$

and recall that

$$\mathbf{F}_j^T\mathbf{F}_j = \mathbf{I} - \omega\mathbf{e}_j\mathbf{e}_j^T\,.$$

Then with some algebra, we get

$$b^{(j)} - [\mathbf{A}\mathbf{x}_{\lambda,j}]^{(j)} = \mathbf{e}_j^T\widehat{\mathbf{D}}[\mathbf{I} - \mathbf{T}]\mathbf{b}\,, \tag{A-8}$$

where

$$\widehat{\mathbf{D}} = \text{diag}\left(\frac{1}{1-\omega t^{(jj)}}\right).$$

In conclusion,

$$\begin{aligned} V_\omega(\lambda) &= \frac{1}{m}\sum_{j=1}^{m}(b^{(j)} - [A\mathbf{x}_{\lambda,j}]^{(j)})^2 \\ &= \frac{1}{m}||\widehat{\mathbf{D}}(\lambda)(\mathbf{I} - \mathbf{T}(\lambda))\mathbf{b}||_2^2. \end{aligned} \tag{A-9}$$

Thus, we have obtained our result. □

Now, extension from weighted-cross-validation to the weighted-GCV function is analogous to the generalization process from cross-validation to GCV provided in [53].

A.2 Choosing ω in W-GCV

This section provides more details about how we select ω in the weighted-GCV approach described in Section 2.4. In particular, recall from Section 2.4.4 that we find ω by minimizing the GCV function with respect to λ. That is,

$$\frac{\partial}{\partial\lambda}\left[G(\omega,\lambda)\right]\Big|_{\lambda=\lambda_{k,opt}} = 0.$$

Although in HyBR ω is used to help select regularization parameters for the projected bidiagonal system (2.12), here we derive the selection of ω for the generic weighted-GCV function found in equation (2.17). That is, we use the GCV function that depends on $\mathbf{A}$ and $\mathbf{b}$, and for convenience introduce H and L notation to represent the numerator and square root of the denominator of the GCV function respectively. Let $\hat{b}_i = \mathbf{u}_i^T\mathbf{b}$, where $\mathbf{u}_i$ is the i^{th} left singular vector of $\mathbf{A}$, then the weighted-GCV function can be

written as

$$G(\omega,\lambda)=\frac{n\left[\sum_{i=1}^{n}\left(\frac{\lambda^2\hat{b}_i}{\sigma_i^2+\lambda^2}\right)^2+\sum_{i=n+1}^{m}\hat{b}_i^2\right]}{\left(\sum_{i=1}^{n}\frac{(1-\omega)\sigma_i^2+\lambda^2}{\sigma_i^2+\lambda_i^2}+m-n\right)^2}\equiv\frac{H}{L^2}\,. \tag{A-10}$$

We would like to take the derivative of $G(\omega,\lambda)$ with respect to λ, so by the quotient rule, we obtain

$$\frac{\partial G(\omega,\lambda)}{\partial\lambda}=\frac{L^2\partial H-H\partial(L^2)}{L^4}\,.$$

Notice that

$$\begin{aligned}\frac{\partial H}{\partial\lambda} &= n\sum_{i=1}^{n}\frac{\partial}{\partial\lambda}\left[\left(\frac{\lambda^2\hat{b}_i}{\sigma_i^2+\lambda^2}\right)^2\right]\\ &= n\sum_{i=1}^{n}\left(2\left(\frac{\lambda^2\hat{b}_i}{\sigma_i^2+\lambda^2}\right)\left(\frac{(\sigma_i^2+\lambda^2)2\lambda\hat{b}_i-\lambda^2\hat{b}_i(2\lambda)}{(\sigma_i^2+\lambda^2)^2}\right)\right)\\ &= 4n\sum_{i=1}^{n}\frac{\sigma_i^2\lambda^3\hat{b}_i^2}{(\sigma_i^2+\lambda^2)^3}\end{aligned}$$

and

$$\begin{aligned}\frac{\partial L^2}{\partial\lambda} &= 2L\frac{\partial L}{\partial\lambda}\\ &= 4\lambda\left(\sum_{i=1}^{n}\frac{(1-\omega)\sigma_i^2+\lambda^2}{\sigma_i^2+\lambda_i^2}+m-n\right)\left[\sum_{i=1}^{n}\frac{\omega\sigma_i^2}{(\sigma_i^2+\lambda^2)^2}\right]\\ &= 4\lambda(L)\left[\sum_{i=1}^{n}\frac{\omega\sigma_i^2}{(\sigma_i^2+\lambda^2)^2}\right].\end{aligned}$$

Thus, the derivative for the GCV function with respect to λ can be written as in equation (A-11). Now to find ω, we set the numerator of (A-11) to 0 (see equation (A-12)) and obtain an explicit formula for ω provided in equation (A-13).

$$\frac{\partial G(\omega,\lambda)}{\partial \lambda} = \frac{4n\lambda\left[\left(\sum_{i=1}^{n}\frac{(1-\omega)\sigma_i^2+\lambda^2}{\sigma_i^2+\lambda^2}+m-n\right)\left(\sum_{i=1}^{n}\frac{\sigma_i^2\lambda^2\hat{b}_i^2}{(\sigma_i^2+\lambda^2)^3}\right)-\omega\left(\sum_{i=1}^{n}\left(\frac{\lambda^2\hat{b}_i}{\sigma_i^2+\lambda^2}\right)^2+\sum_{i=n+1}^{m}\hat{b}_i^2\right)\sum_{i=1}^{n}\frac{\sigma_i^2}{(\sigma_i^2+\lambda^2)^2}\right]}{\left(\sum_{i=1}^{n}\frac{(1-\omega)\sigma_i^2+\lambda^2}{\sigma_i^2+\lambda_i^2}+m-n\right)^3} \tag{A-11}$$

$$\left(\sum_{i=1}^{n}\frac{(1-\omega)\sigma_i^2+\lambda^2}{\sigma_i^2+\lambda^2}+m-n\right)\left(\sum_{i=1}^{n}\frac{\sigma_i^2\lambda^2\hat{b}_i^2}{(\sigma_i^2+\lambda^2)^3}\right)-\omega\left(\sum_{i=1}^{n}\left(\frac{\lambda^2\hat{b}_i}{\sigma_i^2+\lambda^2}\right)^2+\sum_{i=n+1}^{m}\hat{b}_i^2\right)\sum_{i=1}^{n}\frac{\sigma_i^2}{(\sigma_i^2+\lambda^2)^2}=0 \tag{A-12}$$

$$\omega = \frac{m\sum_{i=1}^{n}\frac{\sigma_i^2\lambda^2\hat{b}_i^2}{(\sigma_i^2+\lambda^2)^3}}{\left[\sum_{i=1}^{n}\frac{\sigma_i^2}{\sigma_i^2+\lambda^2}\left(\sum_{i=1}^{n}\frac{\sigma_i^2\lambda^2\hat{b}_i^2}{(\sigma_i^2+\lambda^2)^3}\right)+\sum_{i=1}^{n}\frac{\sigma_i^2}{(\sigma_i^2+\lambda^2)^2}\left(\sum_{i=1}^{n}\left(\frac{\lambda^2\hat{b}_i}{\sigma_i^2+\lambda^2}\right)^2+\sum_{i=n+1}^{m}\hat{b}_i^2\right)\right]} \tag{A-13}$$

A.3 Convexity for Tomosynthesis

In Section 2.4.2 we remarked that the polyenergetic cost function is convex with respect to the glandular fractions, under the following two conditions:

1. $\mathbf{A}$ is full rank, and
2. $b^{(i)} - (\bar{b}^{(i)} + \bar{\varepsilon}^{(i)}) \leq \dfrac{\min\limits_{e} s(e)}{\max\limits_{e} s(e)} \left(\dfrac{b^{(i)}}{\bar{b}^{(i)} + \bar{\varepsilon}^{(i)}} \right) \bar{b}^{(i)} \qquad \text{for all } i.$

In this section we derive these conditions.

Recall that the Hessian matrix has the following form: $\mathbf{H} = \mathbf{A}^T\mathbf{W}\mathbf{A}$, where $\mathbf{A}$ is the ray trace matrix and $\mathbf{W}$ is a diagonal matrix with entries $w^{(i)}$ on the diagonal. We would like to show that the Hessian matrix is positive definite or positive semi-definite under the above assumptions. If we assume matrix $\mathbf{A}$ is full rank, we would like to determine the conditions under which $w^{(i)} \geq 0$ for all i. That is, our goal is to prove the following:

$$\left(1 - \frac{b^{(i)}}{\bar{b}^{(i)} + \bar{\varepsilon}^{(i)}}\right) \sum_{e=1}^{n_e} \varrho(e) s(e)^2 \exp\left(- \left[s(e) \sum_{j=1}^{N} a^{(ij)} x^{(j)} + z(e) \sum_{j=1}^{N} a^{(ij)} \right] \right) + \frac{b^{(i)}}{(\bar{b}^{(i)} + \bar{\varepsilon}^{(i)})^2} \left[\sum_{e=1}^{n_e} \varrho(e) s(e) \exp\left(- \left[s(e) \sum_{j=1}^{N} a^{(ij)} x^{(j)} + z(e) \sum_{j=1}^{N} a^{(ij)} \right] \right) \right]^2 \geq 0 \,. \tag{A-14}$$

Assuming $\bar{b}^{(i)} + \bar{\varepsilon}^{(i)} \geq 0$, then (A-14) is equivalent to the following:

$$\left(\bar{b}^{(i)} + \bar{\varepsilon}^{(i)} - b^{(i)}\right) \sum_{e=1}^{n_e} \varrho(e) s(e)^2 \exp\left(- \left[s(e) \sum_{j=1}^{N} a^{(ij)} x^{(j)} + z(e) \sum_{j=1}^{N} a^{(ij)} \right] \right) + \frac{b^{(i)}}{(\bar{b}^{(i)} + \bar{\varepsilon}^{(i)})} \left[\sum_{e=1}^{n_e} \varrho(e) s(e) \exp\left(- \left[s(e) \sum_{j=1}^{N} a^{(ij)} x^{(j)} + z(e) \sum_{j=1}^{N} a^{(ij)} \right] \right) \right]^2 \geq 0 \,. \tag{A-15}$$

We begin by mentioning that due to the physical interpretation of these formulas, the two sums (over e) are positive. This follows since the exponential function is positive, the linear fit coefficients $s(e)$ for all e are positive, and values $\varrho(e)$ for all e, which represent the incident x-rays at different energy levels, are positive. Furthermore, the observed datum $b^{(i)}$ and projected datum $\bar{b}^{(i)} + \bar{\varepsilon}^{(i)}$ are positive values.

Now we consider the following two cases.

1. Assume $0 < b^{(i)} \leq \bar{b}^{(i)} + \bar{\varepsilon}^{(i)}$, then $\bar{b}^{(i)} + \bar{\varepsilon}^{(i)} - b^{(i)} \geq 0$. Since $\frac{b^{(i)}}{(\bar{b}^{(i)} + \bar{\varepsilon}^{(i)})} > 0$, (A-15) is satisfied and we are done.

2. Assume $0 < \bar{b}^{(i)} + \bar{\varepsilon}^{(i)} < b^{(i)}$. This is a more complicated situation, and first we need the following lemma.

Lemma A.3 *Assume* $0 < \bar{b}^{(i)} + \bar{\varepsilon}^{(i)} < b^{(i)}$, *and assume*

$$b^{(i)} - (\bar{b}^{(i)} + \bar{\varepsilon}^{(i)}) \leq \frac{\min_e s(e)}{\max_e s(e)} \left(\frac{b^{(i)}}{\bar{b}^{(i)} + \bar{\varepsilon}^{(i)}} \right) \bar{b}^{(i)} \qquad \textit{for all } i. \tag{A-16}$$

Then we have the following relationship:

$$\frac{b^{(i)}}{(\bar{b}^{(i)} + \bar{\varepsilon}^{(i)})} \left[\sum_{e=1}^{n_e} \varrho(e) s(e) \exp\left(- \left[s(e) \sum_{j=1}^{N} a^{(ij)} x^{(j)} + z(e) \sum_{j=1}^{N} a^{(ij)} \right] \right) \right]^2 \geq$$
$$\left(b^{(i)} - (\bar{b}^{(i)} + \bar{\varepsilon}^{(i)}) \right) \sum_{e=1}^{n_e} \varrho(e) s(e)^2 \exp\left(- \left[s(e) \sum_{j=1}^{N} a^{(ij)} x^{(j)} + z(e) \sum_{j=1}^{N} a^{(ij)} \right] \right). \tag{A-17}$$

Proof. By assumption, $0 < \bar{b}^{(i)} + \bar{\varepsilon}^{(i)} < b^{(i)}$, so $0 < \frac{\bar{b}^{(i)} + \bar{\varepsilon}^{(i)}}{b^{(i)}} < 1$. Thus, condition (A-16) is equivalent to

$$\frac{\bar{b}^{(i)} + \bar{\varepsilon}^{(i)}}{b^{(i)}} \left(b^{(i)} - (\bar{b}^{(i)} + \bar{\varepsilon}^{(i)}) \right) \leq \frac{\min_e s(e)}{\max_e s(e)} \bar{b}^{(i)}. \tag{A-18}$$

Also notice that the following two inequalities hold:

$$\sum_{e=1}^{n_e} \varrho(e)s(e)^2 \exp\left(-\left[s(e)\sum_{j=1}^{N} a^{(ij)}x^{(j)} + z(e)\sum_{j=1}^{N} a^{(ij)}\right]\right) \leq \max_e s(e) \sum_{e=1}^{n_e} \varrho(e)s(e)\exp\left(-\left[s(e)\sum_{j=1}^{N} a^{(ij)}x^{(j)} + z(e)\sum_{j=1}^{N} a^{(ij)}\right]\right), \quad \text{(A-19)}$$

and

$$\sum_{e=1}^{n_e} \varrho(e)s(e)\exp\left(-\left[s(e)\sum_{j=1}^{N} a^{(ij)}x^{(j)} + z(e)\sum_{j=1}^{N} a^{(ij)}\right]\right) \geq \min_e s(e) \sum_{e=1}^{n_e} \varrho(e)\exp\left(-\left[s(e)\sum_{j=1}^{N} a^{(ij)}x^{(j)} + z(e)\sum_{j=1}^{N} a^{(ij)}\right]\right) = \min_e s(e)\, \bar{b}^{(i)}, \quad \text{(A-20)}$$

where the last equality is by definition of $\bar{b}^{(i)}$.

Then consider the following term:

$$\frac{\left[\sum_{e=1}^{n_e} \varrho(e)s(e)\exp\left(-\left[s(e)\sum_{j=1}^{N} a^{(ij)}x^{(j)} + z(e)\sum_{j=1}^{N} a^{(ij)}\right]\right)\right]^2}{\sum_{e=1}^{n_e} \varrho(e)s(e)^2\exp\left(-\left[s(e)\sum_{j=1}^{N} a^{(ij)}x^{(j)} + z(e)\sum_{j=1}^{N} a^{(ij)}\right]\right)} \geq \frac{\left[\sum_{e=1}^{n_e} \varrho(e)s(e)\exp\left(-\left[s(e)\sum_{j=1}^{N} a^{(ij)}x^{(j)} + z(e)\sum_{j=1}^{N} a^{(ij)}\right]\right)\right]^2}{\max_e s(e) \sum_{e=1}^{n_e} \varrho(e)s(e)\exp\left(-\left[s(e)\sum_{j=1}^{N} a^{(ij)}x^{(j)} + z(e)\sum_{j=1}^{N} a^{(ij)}\right]\right)} \geq \frac{\min_e s(e)}{\max_e s(e)}\, \bar{b}^{(i)}, \quad \text{(A-21)}$$

where the first inequality uses equation (A-19) and the second uses equation (A-20). Thus, combining equations (A-18) and (A-21), we get

$$\frac{\left[\sum_{e=1}^{n_e} \varrho(e)s(e)\exp\left(-\left[s(e)\sum_{j=1}^{N} a^{(ij)}x^{(j)} + z(e)\sum_{j=1}^{N} a^{(ij)}\right]\right)\right]^2}{\sum_{e=1}^{n_e} \varrho(e)s(e)^2\exp\left(-\left[s(e)\sum_{j=1}^{N} a^{(ij)}x^{(j)} + z(e)\sum_{j=1}^{N} a^{(ij)}\right]\right)} \geq \frac{\bar{b}^{(i)} + \bar{\varepsilon}^{(i)}}{b^{(i)}}\left(b^{(i)} - (\bar{b}^{(i)} + \bar{\varepsilon}^{(i)})\right),$$

which is equivalent to (A-17). □

In conclusion, with this lemma, (A-15) follows easily. That is,

$$\left(\bar{b}^{(i)} + \bar{\varepsilon}^{(i)} - b^{(i)}\right)\sum_{e=1}^{n_e} \varrho(e)s(e)^2\exp\left(-\left[s(e)\sum_{j=1}^{N} a^{(ij)}x^{(j)} + z(e)\sum_{j=1}^{N} a^{(ij)}\right]\right) +$$
$$\frac{b^{(i)}}{(\bar{b}^{(i)} + \bar{\varepsilon}^{(i)})}\left[\sum_{e=1}^{n_e} \varrho(e)s(e)\exp\left(-\left[s(e)\sum_{j=1}^{N} a^{(ij)}x^{(j)} + z(e)\sum_{j=1}^{N} a^{(ij)}\right]\right)\right]^2 \geq$$
$$\left(\bar{b}^{(i)} + \bar{\varepsilon}^{(i)} - b^{(i)}\right)\sum_{e=1}^{n_e} \varrho(e)s(e)^2\exp\left(-\left[s(e)\sum_{j=1}^{N} a^{(ij)}x^{(j)} + z(e)\sum_{j=1}^{N} a^{(ij)}\right]\right) +$$
$$\left(b^{(i)} - (\bar{b}^{(i)} + \bar{\varepsilon}^{(i)})\right)\sum_{e=1}^{n_e} \varrho(e)s(e)^2\exp\left(-\left[s(e)\sum_{j=1}^{N} a^{(ij)}x^{(j)} + z(e)\sum_{j=1}^{N} a^{(ij)}\right]\right)$$
$$= 0\,.$$

Bibliography

[1] V. Akcelik, G. Biros, and O. Ghattas. Parallel multiscale Gauss-Newton-Krylov methods for inverse wave propogation. In *Proc. IEEE/ACM SC 2002 Conference*, 2002.

[2] F. Andel, A. G. Ladurner, C. Inouye, R. Tjian, and E. Nogales. Three-dimensional structure of the human TFIID-IIA-IIB complex. *Science*, 286:2153–2156, 1999.

[3] H. C. Andrews. *N* Topics in search of an editorial: Heuristics, super-resolution, and bibliography. *Proc. IEEE*, 60(7):340–343, 1972.

[4] J. M. Bardsley. A limited-memory, quasi-Newton preconditioner for nonnegatively constrained image reconstruction. *J. Opt. Soc. Am. A*, 21:724–731, 2004.

[5] R. Barnard, V. P. Pauca, T. C. Torgersen, R. J. Plemmons, S. Prasad, J. van der Gracht, J. Nagy, J. Chung, G. Behrmann, S. Matthews, and M. Mirotznik. High-resolution iris image reconstruction from low-resolution imagery. *Proc. SPIE, Advanced Signal Processing Algorithms, Architectures and Implementations*, 6313:D1–D13, 2006.

[6] M. Bazalova, J. Carrier, L. Beaulieu, and F. Verhaegen. Dual-energy CT-based material extraction for tissue segmentation in Monte Carlo dose calculations. *Phys. Med. Biol.*, 53:2439–2456, 2008.

[7] Å. Björck. A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations. *BIT*, 28:659–670, 1988.

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

[9] Å. Björck, E. Grimme, and P. van Dooren. An implicit shift bidiagonalization algorithm for ill-posed systems of linear equations. *BIT*, 34:510–534, 1994.

[10] P. Bleuet, R. Guillemaud, and I. E. Magnin. Resolution improvement in linear tomosynthesis with an adapted 3D regularization scheme. *Proc. SPIE*, 4682:117–125, 2002.

[11] B. Bluemich. *NMR Imaging of Materials*. Oxford University Press, Oxford, UK, 2003.

[12] J. M. Boone, T. R. Nelson, K. K. Lindfors, and J. A. Seibert. Dedicated breast CT: Radiation dose and image quality evaluation. *Radiology*, 221:657–667, 2001.

[13] N. K. Bose and K. J. Boo. High-resolution image reconstruction with multisensors. *Int. J. Imaging Syst. Technol.*, 9:294–304, 1998.

[14] R. Brooks and G. Di Chiro. Beam hardening in x-ray reconstructive tomography. *Phys. Med. Biol.*, 21:390–398, 1976.

[15] D. Calvetti, G. H. Golub, and L. Reichel. Estimation of the L-curve via Lanczos bidiagonalization. *BIT*, 39:603–619, 1999.

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

[17] D. Calvetti and L. Reichel. Tikhonov regularization with a solution constraint. *SIAM J. Sci. Comput.*, 26:224–239, 2004.

[18] A. S. Carasso. Direct blind deconvolution. *SIAM J. Appl. Math.*, 61:1980–2007, 2001.

[19] P. Cheeseman, B. Kanefsky, R. Kraft, J. Stutz, and R. Hanson. Super-resolved surface reconstruction from multiple images. Technical Report Tech. Rep. FIA-94-12, NASA Ames Research Center, Moffett Field, CA, Dec., 1994.

[20] P. Chen and K. Barner. A multi-resolution statistical reconstruction for digital tomosynthesis. Available online at http://www.ece.udel.edu/∼pchen/My Publications/Tomo.pdf.

[21] J. Chung, E. Haber, and J. G. Nagy. Numerical methods for coupled super-resolution. *Inverse Problems*, 22:1261–1272, 2006.

[22] J. Chung and J. G. Nagy. Nonlinear least squares and super resolution. *Journal of Physics Conference Series*, 124:012019, 2008.

[23] J. Chung and J. G. Nagy. Separable nonlinear least squares for large scale problems. Technical Report TR-2008-014, Emory University Math/CS, Submitted 2008.

[24] J. Chung, J. G. Nagy, and D. P. O'Leary. A weighted GCV method for Lanczos hybrid regularization. *Elec. Trans. Numer. Anal.*, 28:149–167, 2008.

[25] J. Chung, J. G. Nagy, and I. Sechopoulos. Numerical algorithms for polyenergetic digital breast tomosynthesis reconstruction. Technical Report TR-2009-006, Emory University Math/CS, Submitted 2009.

[26] J. Chung, P. Sternberg, and C. Yang. High performance 3-D image reconstruction for molecular structure determination. Technical Report LBNL-874E, Lawrence Berkeley National Laboratory, Submitted 2008.

[27] G. Cristóbal, E. Gil, F. Šroubek, J. Flusser, C. Miravet, and F. B. Rodríguez. Superresolution imaging: a survey of current techniques. *Proc. SPIE, Advanced Signal Processing Algorithms, Architectures and Implementations*, 7074, 2008.

[28] D. Cummins, T. Filloon, and D. Nychka. Confidence intervals for nonparametric curve estimates: Toward more uniform pointwise coverage. *J. Am. Stat. Assoc.*, 96:233–246, 2001.

[29] B. De Man, J. Nuyts, P. Dupont, G. Marchal, and P. Suetens. An iterative maximum-likelihood polychromatic algorithm for CT. *IEEE Trans. Med. Imaging*, 20:999–1008, 2001.

[30] A. H. Delaney and Y. Bresler. Globally convergent edge-preserving regularized reconstruction: An application to limited-angle tomography. *IEEE Trans. Image Process.*, 7(2):204–221, 1998.

[31] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia, 1996.

[32] F. Di Benedetto, C. Estatico, J. G. Nagy, and M. Pastorino. Numerical linear algebra for nonlinear microwave imaging. Technical Report TR-2008-020, Emory University Math/CS, Submitted 2008.

[33] J. T. Dobbins III and D. J. Godfrey. Digital x-ray tomosynthesis: current state of the art and clinical potential. *Phys. Med. Biol.*, 48:R65–R106, 2003.

[34] N. Efford. *Digital Image Processing: A Practical Introduction using Java*. Addison-Wesley, Harlow, England, 2000.

[35] M. Elad and A. Feuer. Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images. *IEEE Trans. Image Proc.*, 6(12):1646–1658, 1997.

[36] I. A. Elbakri and J. A. Fessler. Statistical image reconstruction for polyenergetic x-ray computed tomography. *IEEE Trans. Med. Imaging*, 21:89–99, 2002.

[37] I. A. Elbakri and J. A. Fessler. Segmentation-free statistical image reconstruction for polyenergetic x-ray computed tomography with experimental validation. *Phys. Med. Biol.*, 48:2453–2477, 2003.

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

[39] L. Eldén. A weighted pseudoinverse, generalized singular values, and constrained least squares problems. *BIT*, 22:487–502, 1982.

[40] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Kluwer Academic Publishers, Dordrecht, 2000.

[41] H. Erdoğan and J. A. Fessler. Monotonic algorithms for transmission tomography. *IEEE Trans. Med. Imaging*, 18(9):801–814, 1999.

[42] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar. Advances and challenges in super-resolution. *Int. J. Imaging Syst. Technol.*, 14(2):47–57, 2004.

[43] J. A. Fessler. Statistical image reconstruction methods for transmission tomography. In M. Sonka and J. M. Fitzpatrick, editors, *Handbook of Medical Imaging, Medical Image Processing and Analysis*, volume 2. SPIE, Bellingham, WA, 2000.

[44] J. Frank. *Three-Dimensional Electron Microscopy of Macromolecular Assemblies*. Oxford University Press, New York, 2006.

[45] J. Frank, M. Radermacher, P. A. Penczek, J. Zhu, Y. Li, M. Ladjadj, and A. Leith. SPIDER and WEB: Processing and visualization of images in 3D electron microscopy and related fields. *J. Struct. Biol.*, 116:190–199, 1995.

[46] J. Friedman and B. Silverman. Flexible parsimonious smoothing and additive modeling. *Technometrics*, 31(1):3–21, 1989.

[47] A. Frommer and P. Maass. Fast CG-based methods for Tikhonov-Phillips regularization. *SIAM J. Sci. Comput.*, 20:1831–1850, 1999.

[48] J. B. Garrison, D. G. Grant, W. H. Guier, and R. J. Johns. Three dimensional roentgenography. *Am. J. Roentgenol.*, 105:903–908, 1969.

[49] P. Gilbert. Iterative methods for the three-dimensional reconstruction of an object from projections. *J Theor Biol*, 36:105–17, 1972.

[50] G. Golub and W. Kahan. Calculating the singular values and pseudoinverse of a matrix. *SIAM J. Numer. Anal.*, 2:205–224, 1965.

[51] G. Golub and V. Pereyra. The differentiation of pseudo-inverses and nonlinear least squares whose variables separate. *SIAM J. Numer. Anal.*, 10:413–432, 1973.

[52] G. Golub and V. Pereyra. Separable nonlinear least squares: the variable projection method and its applications. *Inverse Problems*, 19:R1–R26, 2003.

[53] 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.

[54] G. H. Golub, F. T. Luk, and M. L. Overton. A block Lanczos method for computing the singular values and corresponding singular vectors of a matrix. *ACM Trans. Math Soft.*, 7:149–169, 1981.

[55] G. H. Golub and C. F. Van Loan. *Matrix Computations, third edition.* Johns Hopkins University Press, 1996.

[56] G. H. Golub and U. von Matt. Quadratically constrained least squares and quadratic problems. *Numer. Math.*, 59:561–580, 1991.

[57] D. G. Grant. Tomosynthesis: A three-dimensional radiographic imaging technique. *IEEE Trans. Biomed. Eng.*, 19:20–28, 1972.

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

[59] W. Gropp, E. Haber, S. Heldmann, D. Keyes, N. Miller, J. Schopf, and T. Yang. Grid-based image registration. In P. W. Gaffney and J. C. T. Poll, editors, *Grid-Based Problem Solving Environments*, volume 239, pages 435–448. Springer, Boston, MA, 2007.

[60] C. Gu. Smoothing noisy data via regularization: statistical perspectives. *Inverse Problems*, 24:034002, 2008.

[61] E. Haber and D. Oldenburg. A GCV based method for nonlinear inverse problems. *Computational Geoscience*, 4:41–63, 2000.

[62] J. Hadamard. *Lectures on Cauchy's Problem in Linear Differential Equations.* Yale University Press, New Haven, 1923.

[63] G. R. Hammerstein, D. W. Miller, D. R. White, M. E. Masterson, H. Q. Woodard, and J. S. Laughlin. Absorbed radiation dose in mammography. *Radiology*, 130:485–491, 1979.

[64] M. Hanke. *Conjugate Gradient Type Methods for Ill-Posed Problems.* Pitman Research Notes in Mathematics, Longman Scientific & Technical, Harlow, Essex, 1995.

[65] M. Hanke. Regularizing properties of a truncated Newton-CG algorithm for nonlinear inverse problems. *Numer. Funct. Anal. Optim.*, 18:971–993, 1997.

[66] M. Hanke. On Lanczos based methods for the regularization of discrete ill-posed problems. *BIT*, 41:1008–1018, 2001.

[67] P. C. Hansen. Regularization tools: A MATLAB package for analysis and solution of discrete ill-posed problems. *Numerical Algorithms*, 6:1–35, 1994.

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

[69] P. C. Hansen, J. G. Nagy, and D. P. O'Leary. *Deblurring Images: Matrices, Spectra and Filtering.* SIAM, Philadelphia, PA, 2006.

[70] R. Hardie, K. J. Barnard, and E. E. Armstrong. Joint MAP registration and high-resolution image estimation using a sequence of undersampled images. *IEEE Trans. Image Proc.*, 6:1621–1633, 1997.

[71] L. He, A. Marquina, and S. Osher. Blind deconvolution using TV regularization and Bregman iteration. *International Journal of Imaging Systems and Technology*, 15:74–83, 2005.

[72] G. T. Herman. *Image Reconstruction from Projection: The Fundamentals of Computerized Tomography.* Academic Press, New York, 1980.

[73] I. Hnětynková, M. Plešinger, and Z. Strakoš. Golub-Kahan iterative bidiagonalization and revealing the size of the noise in a data. Available

online at http://www.cs.cas.cz/strakos/download/2008_HnPlSt.pdf, Preprint 2008.

[74] E. Y. T. Ho and A. E. Todd-Pokropek. Blob-based super-resolution reconstruction using iterative Lanczos-hybrid regularization. *IEEE Nuclear Science Symposium Conference Record*, 4:2754–2759, 2007.

[75] M. Hohn, G. Tang, G. Goodyear, P. R. Baldwin, Z. Huang, P. A. Penczek, C. Yang, R. M. Glaeser, P. D. Adams, and S. J. Ludtke. SPARX, a new environment for Cryo-EM image processing. *J. Structural Biology*, 157:47–55, 2007.

[76] S. M. Jefferies, K. J. Schulze, C. L. Matson, K. Stoltenberg, and E. K. Hege. Blind deconvolution in optical diffusion tomography. *Optics Express*, 10:46–53, 2002.

[77] S. Kaczmarz. Angenäherte Auglösung von Systemen linearer Gleichungen. *Bulletin de l'Academie Polonaise des Sciences et Lettres*, A35:355–357, 1937.

[78] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*. Springer, New York, 2005.

[79] B. Kaltenbacher. Some Newton-type methods for the regularization of nonlinear ill-posed problems. *Inverse Problems*, 13:729–753, 1997.

[80] J. Kamm and J. G. Nagy. Kronecker product and SVD approximations in image restoration. *Linear Algebra Appl.*, 284:177–192, 1998.

[81] J. Kamm and J. G. Nagy. Optimal Kronecker product approximation of block Toeplitz matrices. *SIAM J. Matrix Anal. Appl.*, 22:155–172, 2000.

[82] M. G. Kang and S. Chaudhuri. Super-resolution image reconstruction. *IEEE Signal Processing Magazine*, 20(3):19–20, 2003.

[83] A. Karellas, J. Lo, and C. Orton. Point/counterpoint: Cone beam x-ray CT will be superior to digital x-ray tomosynthesis in imaging the breast and delineating cancer. *Med. Phys.*, 35(2):409–411, 2008.

[84] I. Kastanis, S. Arridge, A. Stewart, S. Gunn, C. Ullberg, and T. Francke. 3D digital breast tomosynthesis using total variation regularization. In *Lecture Notes in Computer Science*, volume 5116. Springer, 2008.

[85] L. Kaufman. A variable projection method for solving separable nonlinear least squares problems. *BIT*, 15:49–57, 1975.

[86] C. T. Kelley. *Iterative Methods for Optimization*. SIAM, Philadelphia, 1999.

[87] M. E. Kilmer, P. C. Hansen, and M. I. Español. A projection-based approach to general-form Tikhonov regularization. *SIAM J. Sci. Comput.*, 29:315–330, 2007.

[88] M. E. Kilmer and D. P. O'Leary. Choosing regularization parameters in iterative methods for ill-posed problems. *SIAM J. Matrix Anal. Appl.*, 22:1204–1221, 2001.

[89] Y. Kim and C. Gu. Smoothing spline Gaussian regression: More scalable computation via efficient approximation. *J. Roy. Stat. Soc.*, 66:337–356, 2004.

[90] K. Lange. An overview of Bayesian methods in image reconstruction. *SPIE*, 1351:270–287, 1990.

[91] K. Lange and J. A. Fessler. Globally convergent algorithms for maximum a posteriori transmission tomography. *IEEE Trans. Image Process.*, 4:1430–1438, 1995.

[92] R. M. Larsen. Lanczos bidiagonalization with partial reorthogonalization. (DAIMI PB-357), 1998.

[93] C. Laurent, F. Payrin, J. M. Chassery, and M. Amiel. Parallel image reconstruction on MIMD computers for three-dimensional cone-beam tomography. *Parallel Computing*, 24:1461–1479, 1998.

[94] R. Marabini, G. T. Herman, and J.M. Carazo. 3D reconstruction in electron microscopy using ART with smooth spherically symmetric volume elements (blobs). *Ultramicroscopy*, 72:53–65, 1998.

[95] S. Moore, P. Mucci, J. Dongarra, S. Shende, and A. Malony. Performance instrumentation and measurement for terascale systems. In *Lecture Notes in Computer Science*, volume 2723, pages 53–62, Heidelberg, 2003. Springer-Verlag.

[96] J. Nagy, K. Palmer, and L. Perrone. Iterative methods for image deblurring: A MATLAB object oriented approach. *Numerical Algorithms*, 36:73–93, 2004.

[97] J. G. Nagy, M. K. Ng, and L. Perrone. Kronecker product approximation for image restoration with reflexive boundary conditions. *SIAM J. Matrix Anal. Appl.*, 25:829–841, 2004.

[98] S. G. Nash. Preconditioning of truncated-Newton methods. *SIAM J. Sci. Stat. Comp.*, 6:599–616, 1985.

[99] F. Natterer. *The Mathematics of Computerized Tomography*. SIAM, Philadelphia,PA, 2001.

[100] F. Natterer and F. Wübbeling. *Methematical Methods in Image Reconstruction*. SIAM, Philadelphia, PA, 2001.

[101] M. Ng, R. Chan, T. Chan, and A. Yip. Cosine transform preconditioners for high resolution image reconstruction. *Linear Algebra Appl.*, 316:89–104, 2000.

[102] M. K. Ng, H. Shen, E. Y. Lam, and L. Zhang. A total variation regularization based super-resolution reconstruction algorithm for digital video. *EURASIP J. on Advances in Signal Processing*, page 74585, 2007.

[103] N. Nguyen, P. Milanfar, and G. Golub. Efficient generalized cross-validation with applications to parametric image restoration and resolution enhancement. *IEEE Trans. Image Proc.*, 10(9):1299–1308, 2001.

[104] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, New York, 1999.

[105] R. D. Nowak. Optimal signal estimation using cross-validation. *IEEE Signal Processing Letters*, 4:23–25, 1997.

[106] D. Nychka, B. Bailey, S. Ellner, P. Haaland, and M. O'Connel. FUNFITS: Data analysis and statistical tools for estimating functions. In *Case Studies in Environmental Statistics*, pages 159–179. Springer, New York, 1998.

[107] D. P. O'Leary and J. A. Simmons. A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems. *SIAM J. Sci. Stat. Comp.*, 2:474–489, 1981.

[108] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables (2nd edition)*. SIAM, Philadelphia, 2003.

[109] M. R. Osborne. Separable least squares, variable projection, and the Gauss-Newton algorithm. *Elec. Trans. Numer. Anal.*, 28:1–15, 2007.

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

[111] C. C. Paige and M. A. Saunders. Algorithm 583, LSQR: Sparse linear equations and least-squares problems. *ACM Trans. Math. Soft.*, 8(2):195–209, 1982.

[112] S. C. Park, M. K. Park, and M. G. Kang. Super-resolution image reconstruction: A technical overview. *IEEE Signal Processing Magazine*, 20(3):21–36, 2003.

[113] V. P. Pauca, D. Chen, J. van der Gracht, R. J. Plemmons, S. Prasad, and T. C. Torgersen. Pupil phase encoding for multi-aperture imaging. *Proc. Annual SPIE Meeting*, 2008. Available online at http://www.wfu.edu/∼plemmons/papers/SPIE08.pdf.

[114] P. A. Penczek, M. Radermacher, and J. Frank. Three-dimensional reconstruction of single particles embedded in ice. *Ultramicroscopy*, 40:33–53, 1992.

[115] P. A. Penczek, R. Renka, and H. Schomberg. Gridding-based direct Fourier inversion of the three-dimensional ray transform. *J. Opt. Soc. Am. A*, 21:499–509, 2004.

[116] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[117] A. Ruhe and P. Wedin. Algorithms for separable nonlinear least squares problems. *SIAM Review*, 22:318–337, 1980.

[118] Y. Saad. On the rates of convergence of the Lanczos and the block-Lanczos methods. *SIAM J. Numer. Anal.*, 17:687–706, 1980.

[119] S. D. Saban, M. Silvestry, G. Nemerow, and P. L. Stewart. Visualization of alpha-helices in a 6 Angstrom resolution cryoEM structure of adenovirus allows refinement of capsid protein assignments. *J. Virol*, 80:12049–59, 2006.

[120] T. Schulz. Multiframe blind deconvolution of astronomical images. *J. Opt. Soc. Am. A*, 10:1064–1073, 1993.

[121] H. Shen, L. Zhang, B. Huang, and P. Li. A MAP approach for joint motion estimation, segmentation, and super resolution. *IEEE Trans. Image Process.*, 16:479–490, 2007.

[122] R. L. Siddon. Fast calculation of the exact radiological path for a three-dimensional CT array. *Med. Phys.*, 12(2):252–255, 1985.

[123] E. Y. Sidky, C. Kao, and X. Pan. Accurate image reconstruction from few-views and limited-angle data in divergent-beam CT. *Journal of X-Ray Science and Technology*, 14:119–139, 2006.

[124] B. C. Tom and A. K. Katsaggelos. Reconstruction of a high-resolution image by simultaneous registration, restoration, and interpolation of low-resolution images. In *Proc. IEEE Int. Conf. Image Processing*, Washington, DC, 1995.

[125] A. van der Sluis and H. A. van der Vorst. SIRT- and CG- type methods for the iterative solution of sparse linear least-squares problems. *Linear Algebra Appl.*, 130:257–302, 1990.

[126] J. M. Varah. Pitfalls in the numerical solution of linear ill-posed problems. *SIAM J. Sci. Stat. Comp.*, 4:164–176, 1983.

[127] R. Vio, P. Ma, W. Zhong, J. Nagy, L. Tenorio, and W. Wamsteker. Estimation of regularization parameters in multiple-image deblurring. *Astronomy and Astrophysics*, 423:1179–1186, 2004.

[128] C. R. Vogel. Optimal choice of a truncation level for the truncated SVD solution of linear first kind integral equations when data are noisy. *SIAM J. Numer. Anal.*, 23:109–117, 1986.

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

[130] C. R. Vogel, T. Chan, and R. J. Plemmons. Fast algorithms for phase diversity-based blind deconvolution. In *Adaptive Optical System Technologies*, volume 3353. SPIE, 1998.

[131] F. Šroubek, G. Cristóbal, and J. Flusser. Simultaneous super-resolution and blind deconvolution. *Journal of Physics Conference Series*, 124:012048, 2008.

[132] M. M. Woolfson. *An Introduction to X-ray Crystallography*. Cambridge University Press, Cambridge, UK, second edition, 1997.

[133] T. Wu, R. H. Moore, E. A. Rafferty, and D. B. Kopans. A comparison of reconstruction algorithms for breast tomosynthesis. *Med. Phys.*, 31:2636–2647, 2004.

[134] T. Wu, A. Stewart, M. Stanton, T. McCauley, W. Phillips, D. B. Kopans, R. H. Moore, J. W. Eberhard, B. Opsahl-Ong, L. Niklason, and M. B. Williams. Tomographic mammography using a limited number of low-dose cone-beam projection images. *Med. Phys.*, 30:365–380, 2003.

[135] T. Wu, J. Zhang, R. Moore, E. Rafferty, D. Kopans, W. Meleis, and D. Kaeli. Digital tomosynthesis mammography using a parallel maximum likelihood reconstruction model. *Proc. SPIE*, 1:5368, 2004.

[136] C. Yang, E. Ng, and P. Penczek. Unified 3-D structure and projection orientation refinement using quasi-Newton algorithm. *J. Structural Biology*, 149:53–64, 2005.

[137] Y. Zhang, H. Chan, B. Sahiner, J. Wei, M. M. Goodsitt, L. M. Hadjiiski, J. Ge, and C. Zhou. A comparative study of limited-angle cone-beam reconstruction methods for breast tomosynthesis. *Med. Phys.*, 33:3781–3795, 2006.