Distribution Agreement

In presenting this thesis as a partial fulfillment of the requirements for a degree with honors from Emory College, I agree that the Library of the University shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish, this thesis may be granted by the professor under whose direction it was written, or, in his absence, by the chairperson of my major department when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from, or publication of, this thesis which involves potential financial gain will not be allowed without written permission.

Signature:

James L. Herring	April 14,2010

The Effects of Projection on Iterative Methods in Image Deblurring

By

James L. Herring

James Nagy

Department of Mathematics and Computer Science

James Nagy
Advisor

Michele Benzi
Committee Member

Connie Roth
Committee Member

April 14, 2010

The Effects of Projection on Iterative Methods in Image Deblurring

by

James L. Herring

Advisor : James Nagy

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

Department of Mathematics and Computer Science

2010

Abstract

The Effects of Projection on Iterative Methods in Image Deblurring
By James L. Herring

Applications of iterative regularization methods encompass a broad spectrum, including image deblurring. One unique feature in image deblurring problems is the non-negativity of the solution. In light of this knowledge, this thesis explores the effectiveness of three projected iterative methods for image deconvolution: projected successive over-relaxation method (SOR), projected Landweber method, and an interior point gradient method. Specifically, this thesis compares the effectiveness of these methods to the standard un-projected SOR method, comparing quality of image reconstruction and the cost of each method. The thesis begins with an introduction to the field of image deblurring problems, iterative regularization, and the methods tested, and follows with experiments and analysis designed to determine the usefulness of these methods for image deblurring.

The Effects of Projection on Iterative Methods in Image Deblurring

by

James Herring

Advisor : James Nagy

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

Department of Mathematics and Computer Science

2010

Contents

List of Figures

Chapter 1

Introduction

Large-scale inverse problems are prevalent in numerous scientific fields, and as such research and exploration into accurately solving these problems has burgeoned in recent years. Frequently these problems cannot be solved analytically, and as a result the bulk of the efforts for solving inverse problems focuses on the implementation of reliable, cost-effective numerical methods for computing solutions [20]. Image deblurring represents one field where the need to efficiently solve large-scale inverse problems consistently arises.

Image deblurring encompasses mathematical effort to recover an approximation of an original, true image via the removal of the effect of blur and noise on a corrupted image. This process plays an important role in numerous scientific applications including biomedical imaging [4] [10] [13] [17] [20], seismic imaging [20], astronomy [1] [4] [16] [19], and various other fields.

One example in biomedical imaging is the the technique of X-ray computed tomography (CT). First introduced into hospitals during the early 1970s, CT images derive from solving ill-posed inverse problems [4]. Since then, the understanding and treatment of such ill-posed-problems has progressed significantly, resulting in major improvements in medical diagnosis and treatment. Today, improvements in computed tomography continue.

Another field where image restoration frequently arises is astronomy, where scien-

tists often encounter blurred images. For instance, an astronomer using a telescope on earth to photograph an object in space must account for the impact of atmospheric blurring and light randomly reflected off particles skewing the image. Thus, astronomers must utilize methods in image restoration to obtain a more accurate approximation of the true image.

The methods available are numerous for computing image deblurring solutions. Due to the complexity and costliness of many image deblurring problems (often encompassing over one million equations), methods for computing a solution can be specialized to focus on desired aspects of a solution particular to a given problem. As such, the constant advent of new methods and applications necessitates large amounts of time and exploration into the best choice for a given situation. However, despite the breadth of methods and problems available, the end goal remains an accurate and efficiently restored image.

With this goal in mind, this thesis explores the utility of projected iterative methods in the field of image deblurring. The rest of this chapter is devoted to establishing the mathematical background and structure of image deblurring problems, as well as the concept of iterative regularization. Chapter 2 deals with the specific iterative methods tested in this thesis. Chapter 3 provides some analysis on the filtering characteristics of iterative methods. Chapter 4 details the numerical experiments used to determine regularization parameters, stopping criteria, and the effect of noise level on the effectiveness of each method. Lastly, chapter 5 tests the effectiveness of each method on simulated real world problems and draws conclusions about the usefulness of each method in image deblurring.

1.1 Mathematical Background

When solving a particular problem, the first step is the discretization of the problem from real life into a mathematical problem. Many imaging problems are modeled as the linear system

$$\mathbf{b} = \mathbf{A}\mathbf{x}_{\mathbf{true}} + \mathbf{n} \tag{1.1}$$

where $\mathbf{b}$ is the blurred image, $\mathbf{A}$ is the blurring matrix, $\mathbf{x}_{\mathbf{true}}$ is true image, and $\mathbf{n}$ represents random noise affecting the problem. Note that although the image is originally given as a two dimensional array, it can be strung out column-wise as a vector for computational purposes. That is, if $\mathbf{x}$ is an image of $n \times n$ pixels,

$$\mathbf{x} = [\underline{\mathbf{x}}_{\mathbf{1}}\underline{\mathbf{x}}_{\mathbf{2}} \cdots \underline{\mathbf{x}}_{\mathbf{n}}]$$

where $\underline{\mathbf{x}}_{\mathbf{i}}$ is the i^{th} column of $\mathbf{x}$, then for mathematical and computational purposes, we can vectorize $\mathbf{x}$ as:

$$\mathbf{x} = \begin{bmatrix} \underline{\mathbf{x}}_{\mathbf{1}} \\ \underline{\mathbf{x}}_{\mathbf{2}} \\ \vdots \\ \underline{\mathbf{x}}_{\mathbf{n}} \end{bmatrix}$$

where $\mathbf{x}$ is now an $n^2 \times 1$ vector.

After successfully modeling a given problem, invariably the next step is to acquire a solution. In this case, that requires finding an accurate way to solve for the unknown true image $\mathbf{x}_{\mathbf{true}}$ given $\mathbf{b}$ and $\mathbf{A}$, and accounting for the noise $\mathbf{n}$. At first glance, solving system (1.1) seems a simple task, particularly if $\mathbf{A}^{-\mathbf{1}}$ exists and the noise $\mathbf{n}$ is small. However, this apparent solution,

$$\begin{aligned} \mathbf{x} &= \mathbf{A^{-1}b} \\ &= \mathbf{A^{-1}(Ax_{true} + n)} \\ &= \mathbf{x_{true} + A^{-1}n} \end{aligned} \tag{1.2}$$

does not account for several characteristics of the system, namely that $\mathbf{A}$ is typically ill-conditioned and very large. Due to these characteristics, the straightforward naive approach to solving the original system fails to produce any sort of useful reconstruction of the blurred image. This is illustrated in Figure 1.1.

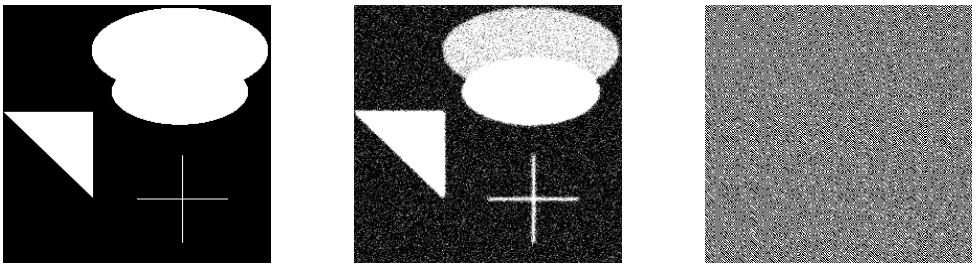

Figure 1.1: Example of the algorithm's instability in the presence of noise: Even in the case of seemingly simple images with very little blur, the naive reconstruction proves useless.

A deeper perspective on the specific challenges posed by $\mathbf{A}$ can be seen using the singular value decomposition. Singular value decomposition is the factoring of the

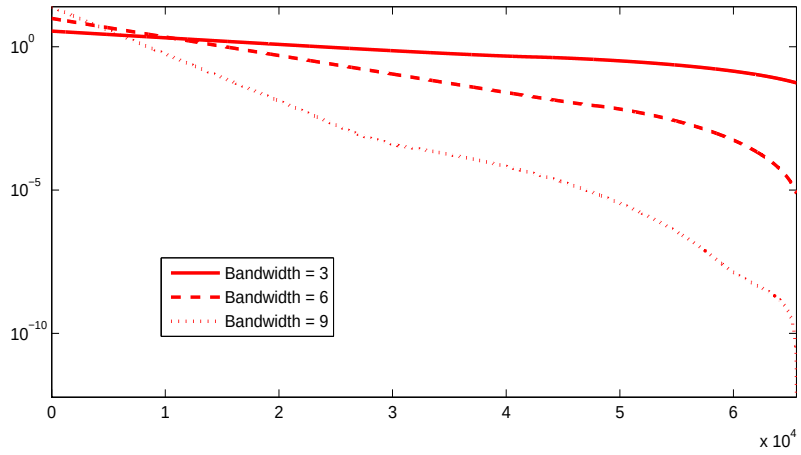

Figure 1.2: Example of the decay of singular values: The graphs above plot the singular values of $\mathbf{A}$ for three increasingly blurred problems. Noting that $\mathbf{cond}(\mathbf{A}) = \sigma_{\mathbf{max}}/\sigma_{\mathbf{min}}$, it can be observed that $\mathbf{A}$ quickly becomes ill-conditioned for problems with increased blur.

matrix $\mathbf{A}$ into a product of three matrices

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathbf{T}} \tag{1.3}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\mathbf{\Sigma}$= $diag(\sigma_1, \sigma_2, \cdots, \sigma_n)$ is a diagonal matrix with the singular values of $\mathbf{A}$ on the diagonal. The singular value decomposition provides a wealth of useful information about the matrix $\mathbf{A}$. In the case of the image deblurring problem, two facts present themselves as particularly important: first, that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ with the smaller σ_i clustering around zero (see Figure 1.2); and second, that the orthogonal basis vectors $\mathbf{v_i}$ corresponding to the singular values σ_i become increasingly oscillatory for larger i.

This information provides an explanation as to why the inverse solution provides a poor reconstruction of the true image. Assuming $\mathbf{A}$ to be invertible, it is apparent that the naive approach used in equation (1.2) gives a solution which contains the

true solution $\mathbf{x_{true}}$, but also contains the inverted noise $\mathbf{A^{-1}n}$. Using the singular value decomposition, we can more closely observe the characteristics of the inverted noise portion of the solution,

$$\begin{aligned} \mathbf{x_{naive}} &= \mathbf{x_{true}} + \mathbf{A^{-1}n} \\ &= \mathbf{x_{true}} + \mathbf{V\Sigma^{-1}U^{T}n} \\ &= \mathbf{x_{true}} + \sum_{\mathbf{i=1}}^{\mathbf{n}} \frac{\mathbf{u_i^T n}}{\sigma_\mathbf{i}} \mathbf{v_i} \end{aligned} \tag{1.4}$$

Since $\sigma_\mathbf{i}$ tends to zero for large i, it follows that $\frac{\mathbf{u_i^T n}}{\sigma_\mathbf{i}}$ becomes very large as i increases. Combined with the increasingly oscillatory $\mathbf{v_i}$ vectors, the inverted noise overwhelms the desired $\mathbf{x_{true}}$ portion of the solution, leading to an incorrect and undesirable solution (Figure 1.1).

Having established that a simple inverse solution is of no use, finding alternative methods to produce an accurate solution is the next step. An ideal solution would simply extract some information about $\mathbf{x_{true}}$ while omitting the inverted noise. For this, it is useful to introduce two ideas. The first is to reformulate equation (1.1) into the least-squares problem:

$$\mathbf{min}\|\mathbf{b - Ax}\|_\mathbf{2}^\mathbf{2} \tag{1.5}$$

This presentation is common in the case of large, ill-posed problems, particularly in cases when equation (1.1) is inconsistent. Throughout our discussion of solving the initial problem, particularly in determining stopping criteria, the least-squares form of the problem will frequently be referenced.

The second useful idea for heading towards a good solution once again builds on knowledge of the singular value decomposition of $\mathbf{A}$. While a solution completely devoid of inverted noise is unattainable, a good solution attempts an accurate recon-

struction of $\mathbf{x_{true}}$ while limiting the negative effect of inverted noise on the image. From the singular value decomposition, we know inverted noise is amplified for smaller singular values σ_i and their corresponding $\mathbf{v_i}$, thus a good solution would exclude or limit the influence of these portions of the inverted noise from the solution. This process of filtering out the unwanted portion of the solution is known as regularization, and can be written as

$$\mathbf{x_{filt}} = \sum_{\mathbf{i=1}}^{\mathbf{n}} \phi_\mathbf{i} \frac{\mathbf{u_i^T b}}{\sigma_\mathbf{i}} \mathbf{v_i} \tag{1.6}$$

where the filter factors ϕ_i are chosen such that $\phi_i \approx 1$ for large singular values and $\phi_i \approx 0$ for small singular values. Some iterative methods have an inherent filtering behavior, which will be discussed further in chapter 3.

1.2 Structure and Sparsity

In choosing a method for regularization, one must first consider the discretization of a problem. In the case of image deblurring, the discretization of the problem into form (1.1) commonly results in characteristics which often influence the selection and efficient implementation of regularization methods. In particular, the blurring matrix $\mathbf{A}$ is often structured, sparse, or both.

Structure in the composition of $\mathbf{A}$ arises from the implementation of the two dimensional point spread function (PSF) to model the spatially invariant convolution of each point in the initial image $\mathbf{x_{true}}$. The PSF stems from the idea that a point in an image is blurred according to the points in its vicinity. Obviously, the composition of the PSF varies with the type of blur introduced in the problem. Three PSFs commonly seen in image deblurring are Gaussian blur, Moffat blur, and atmospheric turbulence blur (common in astronomical imaging problems). After the PSF for the

blur is identified, the blurring matrix $\mathbf{A}$ can be constructed accordingly. In the case of our problems, the modeling of the convolution using the PSF results in the matrix $\mathbf{A}$ being a Toeplitz matrix (constant on each diagonal). Specifically, a Toeplitz matrix arises in the case of a one dimensional problem. For 2-D imaging, $\mathbf{A}$ is a block Toeplitz matrix with Toeplitz blocks. Furthermore, some problems involve separability, e.g the horizontal and vertical blur of the image are separable from one another. Separability implies further structure, and $\mathbf{A}$ can be represented as a Kronecker product of two matrices in addition to the block structure associated with its boundary conditions [14].

The selection of boundary conditions further complicates things. Boundary conditions make assumptions about the pixels outside of the image $\mathbf{x}$. These assumptions are necessary due to the logical assumption that points near the edge of an image will be affected by others near the edge but not included in the image. The most frequently implemented boundary conditions are periodic, reflexive, and zero. Each of these assigns different quantities to the points outside the image, and each results in a different type of structured matrix $\mathbf{A}$. Zero boundary conditions assume all elements outside the given image to be zero, resulting in the general block Toeplitz with Toeplitz blocks (BTTB) structure mentioned previously. Periodic boundary conditions assume the given image repeats itself endlessly in all directions, and the resultant matrix structure is block circulant with circulant blocks (BCCB) where circulant matrices represent a specific class of Toeplitz matrices. Reflexive boundary conditions assume the image to reflect itself over the edge of the given image and result in $\mathbf{A}$ being modeled as a sum of block Toeplitz with Toeplitz blocks (BTTB), block Toeplitz with Hankel blocks (BTHB), block Hankel with Toeplitz blocks (BHTB), and block Hankel with Hankel blocks (BHHB) matrices, where Hankel matrices represent matrices which are constant on the anti-diagonal. Further information on structure due

to boundary conditions can be found in [14].

The PSF and boundary conditions give the matrix **A** a unique structure which can frequently be manipulated and utilized to facilitate more efficient methods for regularizing and solving a given problem. In addition to this useful structuring, **A** frequently is a sparse matrix. Sparse matrices are matrices where the number zero entries is significantly large or the zero entries are structured in such a way that they can be exploited by efficient implimentation, thus making various computations far more efficient than is the case for general, dense matrices. In particular, sparse matrices are particularly efficient for matrix-vector multiplications. This advantage can also be lost, as factorizations of sparse matrices often lead to a loss of sparsity and increased computational cost in addition to the already costly factorization. Thus, for large problems, the choice of a regularization method should take advantage of the increased efficiency of matrix-vector multiplication of sparse matrices while avoiding the potential loss of sparsity and increased cost associated with matrix factorization. Based on this observation, we introduce iterative regularization as an efficient method to effectively utilize the matrix **A**'s structure and sparsity.

1.3 Iterative Regularization

Numerous regularization methods exist, and different applications require varying types of regularization methods. Common techniques often include truncated singular value decomposition (TSVD), selective singular value decomposition, and Tikhonov regularization [9]. However, these methods rely on factorization of the matrix **A** via singular value decomposition. As discussed in the previous section, factorizations typically result in a loss of sparsity and structure of **A** common in image deblurring problems, making calculations for a solution inefficient due to increased cost and

storage [9]. Instead, regularization techniques which exploit matrix sparsity should be utilized to reduce cost and storage while obtaining a solution. One way to exploit matrix sparsity uses iterative regularization.

It should be noted that some of the regularization methods above can be also be put in terms of iterative regularization. In particular, Tikhonov regularization can be implemented in the form $\mathbf{min}(\|\mathbf{b} - \mathbf{Ax}\|_{\mathbf{2}}^{\mathbf{2}} + \alpha^{\mathbf{2}}\|\mathbf{x}\|_{\mathbf{2}}^{\mathbf{2}})$ and solved using an iterative method. This allows for the exploitation of matrix sparsity and structure. Unfortunately, in the case of Tikhonov regularization, two problems arise: firstly, implementing the non-negativity constraint is non-trivial, and secondly, a good estimate of α must be known. For these reasons, iterative solutions using Tikhonov regularization are not explored in this thesis.

Iterative regularization uses iterative methods for solving systems of equations to regularize the solution of ill-posed linear problems [4]. One essential characteristics of iterative regularization is that it is dependent on the number of iterations performed. That is, given an initial guess for a solution $\mathbf{x}^{[\mathbf{0}]}$, each iteration will produce better approximation of $\mathbf{x_{true}}$ up to some iteration $\mathbf{k}^{[\mathbf{opt}]}$, at which point each subsequent iteration will produce solutions increasingly close to the naive solution $\hat{\mathbf{x}}$ given earlier [9]. This phenomenon is called semi-convergence, and is based on the fact that iterative regularization tends to reproduce the parts of the solution corresponding to large singular values in the earlier iterations, with the undesirable small singular values and their corresponding highly oscillatory vectors overwhelming the solution later in the iterative process. An excellent diagram illustrating the principle of semi-convergence can be found in [9], and a discussion on the criteria for choosing $\mathbf{k}^{[\mathbf{opt}]}$ in a subsequent section 1.5.

It is important to make note of the difference in cost associated with iterative regularization and matrix decomposition based methods. Matrix factorizations such

as the singular value decomposition incur a computational cost of order n^3 multiplicative and additive operations for an $n \times n$ matrix $\mathbf{A}$ [7]. In the case of very large problems, such as imaging, such a cost is deemed too expensive for practical use. For this reason, iterative methods are generally preferred for any sort of efficient image deblurring. Because they are composed entirely of matrix-vector multiplications and additions, each iteration has a cost of order n^2. Thus, if the total number of iterations k for a given method to converge is significantly less than the size of system n, iterative methods provide a substantially cheaper, more efficient alternative to direct methods via factorizations. It can be observed that all of the methods presented in this thesis have a cost per iteration of order n^2 at most. However, this associated cost benefit will only become useful provided the methods converge to a useful solution in relatively few iterations. Subsection 1.5 will establish criteria for the regularization parameter (stopping the method to obtain a desirable solution), while the details of the iterative methods investigated in this thesis will be discussed in Chapter 2.

1.4 Projection: Additional Regularization

One characteristic associated with the discretization of imaging problems is non-negativity of the solutions. The entries in the vector $\mathbf{x}$ represent the color or grayscale intensities of the pixels in the image. Since these should not be negative, we know *a priori* that a desirable solution $\mathbf{x}^{[\mathbf{opt}]}$ will be non-negative ($\mathbf{x}_\mathbf{i}^{[\mathbf{opt}]} \geq 0$ for $i = 1 : n$). Utilizing this information, the projection of solutions to ensure non-negativity provides a valuable additional source of regularization for imaging problems. Fortunately, as will be discussed later, for the methods presented in this thesis projection can be implemented relatively easily and provide additional regularization without compromising rate of convergence or quality of the solution. For these reasons, projection

stands out as a viable option for image deblurring problems.

1.5 Regularization Parameters and the Discrepancy Principle

Having established the existence of a $\mathbf{k}^{[\mathbf{opt}]}$, the next step is to locate the iteration at which it occurs, i.e. where the calculated solution $\mathbf{x}^{[\mathbf{k}]}$ is closest to $\mathbf{x_{true}}$. This iteration index is known as a regularization parameter. Unfortunately, a flawless technique for choosing the ideal stopping parameters for general problems does not exist. Instead, there exist multiple methods which under certain assumptions tend to produce satisfactory solutions. These methods typically fall into two categories: methods based on knowledge about the error in the regularized solution and methods based on the analysis of the system [9]. For the experiments and tests in this paper, we will use an error-based regularization parameter known as the discrepancy principle. To introduce the discrepancy principle, it is first necessary to reference the least-squares form of the problem given in equation (1.5). First impulses would suggest iterating until the residual becomes tiny, that is until $\|\mathbf{b} - \mathbf{Ax}\|_\mathbf{2}^\mathbf{2} \approx \mathbf{0}$. But, as was the case with straightforward inverse solution, this represents a naive approach. Instead, notice that rearranging equation (1.1) gives $\mathbf{b} - \mathbf{Ax_{true}} = \mathbf{n}$. It follows that a desirable solution to the least squares problem above does not solve $\|\mathbf{b} - \mathbf{Ax}\|_\mathbf{2}^\mathbf{2} = \mathbf{0}$, but rather a desirable solution would give $\|\mathbf{b} - \mathbf{Ax}\|_\mathbf{2}^\mathbf{2} \approx \|\mathbf{n}\|_\mathbf{2}^\mathbf{2}$. This idea forms the basis of the discrepancy principle. The discrepancy principle determines the regularization parameter by the following criterion:

$$\text{Choose the largest } \mathbf{k} \text{ such that } \|\mathbf{b} - \mathbf{Ax}^{[\mathbf{k}]}\|_\mathbf{2}^\mathbf{2} \geq \delta \|\mathbf{n}\|_\mathbf{2}^\mathbf{2}$$

where $0 < \delta \leq 1$ is a weighting parameter used to counteract the discrepancy principle's tendency to stop after too few iterations. Like all methods that choose regularization parameters, the discrepancy principle has strengths and weaknesses. Often, it is favored due to its simplicity and relative ease in implementation. However, problems in imaging tend to be sensitive to the accuracy of $\|\mathbf{n}\|_{\mathbf{2}}^{\mathbf{2}}$, and the discrepancy principle can provide errant results in cases where a good estimate of the error is not known [9].

Chapter 2

Methods

The following section introduces the four methods that are implemented for the experiments in this thesis. First, the standard successive over-relaxation (SOR) method is presented as a baseline method for comparison. This method was chosen due to its relative simplicity as well as its frequent use as a commonly presented iterative method in multiple texts and papers [7],[6],[18]. Subsequently, three projected iterative methods are presented: projected SOR, projected Landweber method, and interior point gradient method. We note that this thesis does not consider conjugate gradient type methods often implemented for image deblurring due to the nontriviality of incorporating non-negativity constraints.

2.1 Successive Over-Relaxation

Successive over-relaxation is a commonly implemented stationary iterative method, derived as a weighted variation of the Gauss-Seidel method [7]. It is based on the matrix splitting $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$ where $\mathbf{L}$ is strictly lower triangular, $\mathbf{D}$ is diagonal, and $\mathbf{U}$ is strictly upper triangular. Frequently, SOR is applied to the normal equations $\mathbf{A^T A x} = \mathbf{A^T b}$ with the splitting $\mathbf{A^T A} = \mathbf{L} + \mathbf{D} + \mathbf{L^T}$. Using this splitting, we can

derive the following iteration:

$$\begin{aligned}
\mathbf{x^{[k+1]}} &= \mathbf{(D+\omega L)^{-1}[(1-\omega)D-\omega L^T]x^{[k]}+\omega(D+\omega L)^{-1}A^Tb} \\
\mathbf{(D+\omega L)x^{[k+1]}} &= \mathbf{[D-\omega D+\omega L-\omega L-\omega L^T]x^{[k]}+\omega A^Tb} \\
&= \mathbf{[(D+\omega L)-\omega(L+D+L^T)]x^{[k]}+\omega A^Tb} \\
&= \mathbf{[(D+\omega L)-\omega(A^TA)]x^{[k]}+\omega A^Tb} \\
&= \mathbf{(D+\omega L)x^{[k]}-\omega(A^TA)x^{[k]}+\omega A^Tb} \\
\mathbf{x^{[k+1]}} &= \mathbf{x^{[k]}+\omega(D+\omega L)^{-1}A^T(b-Ax^{[k]})} \qquad (2.1)
\end{aligned}$$

where ω is a weighting parameter. For general convergence to be possible, $0 < \omega < 2$ must hold. Additionally, it can be proven that ω in this interval is necessary and suffices for convergence in the case of a symmetric positive definite coefficient matrix [7]. This is particularly relevant in the case of the normal equations, in which if $\mathbf{A}$ is full rank, then $\mathbf{A^TA}$ is symmetric positive definite ($\mathbf{x^TA^TAx > 0}$ for any $\mathbf{x \neq \underline{0}}$). In reference to terminology, note that although the name SOR stands for successive over-relaxation, $\omega < 1$ is in fact under-relaxation while $1 < \omega$ is over-relaxation. This distinction will prove useful in later discussions of the experimental results and optimal ω selection.

It also often useful to view SOR applied to the normal equations in the following form, written component-wise:

$$\begin{aligned}
\mathbf{(D+\omega L)x_i^{[k+1]}} &= \mathbf{((1-\omega)D-\omega LT)x_i^{[k]}+\omega\tilde{b}_i} \\
\mathbf{\tilde{a}_{ii}x_i^{[k+1]}+\omega\sum_{j=1}^{i-1}\tilde{a}_{ij}x_j^{[k+1]}} &= \mathbf{(1-\omega)\tilde{a}_{ii}x_i^{[k]}-\omega\sum_{j=i+1}^{n}\tilde{a}_{ij}x_j^{[k]}+\omega\tilde{b}_i}
\end{aligned}$$

and thus,

$$\mathbf{x_i^{[k+1]}} = (\mathbf{1}-\omega)\mathbf{x_i^{[k]}} + \frac{\omega}{\tilde{\mathbf{a}}_{\mathbf{ii}}}[\tilde{\mathbf{b}}_{\mathbf{i}} - \sum_{\mathbf{j=1}}^{\mathbf{i-1}} \tilde{\mathbf{a}}_{\mathbf{ij}}\mathbf{x_j^{[k+1]}} - \sum_{\mathbf{j=i+1}}^{\mathbf{n}} \tilde{\mathbf{a}}_{\mathbf{ij}}\mathbf{x_j^{[k]}}] \tag{2.2}$$

We remark that the matrix form of the iteration is useful for analysis purposes, while the component-wise form is used for efficient implementation. It should be noted that for efficient implementations of SOR, $\mathbf{A^TA}$ and $\mathbf{A^Tb}$ need not be computed explicitly [5]. At this point, it is useful to make some pertinent remarks as to the cost of the SOR method. Note that by counting the multiplicative and additive operations in equation (2.2) it is easily observable that the cost of each SOR iteration is at most n^2. However, in image deblurring problems, matrix sparsity often makes the cost per iteration significantly less. This characteristic along with its relative simplicity, makes SOR a useful measuring stick by which other iterative methods for image deblurring can be judged.

2.2 Projected SOR

Projected successive over-relaxation (SOR+) follows simply from the unprojected method, and can be implemented with minimum additional work. Calculations for each step are identical to those in unprojected SOR, and the only addition involves checking each calculated $\mathbf{x_i}$ for its sign after the calculation of (2.2). If $\mathbf{x_i}$ is negative, it is set to zero; otherwise, all calculations are identical to the unprojected SOR method.

After implementation, the next logical concern about projected SOR pertains to its cost and convergence. Checking for negativity introduces no new operations, and consequently the cost per iteration is identical to that of SOR. Despite the relative

ease of implementing projected SOR, detailed analysis of the relaxation parameter ω and the convergence rate proves to be much more difficult than in the standard case. Fortunately, it is known that projected SOR, like unprojected SOR, converges for $0 < \omega < 2$ [6],[18].

Projected SOR can be used in a number of applications. Cryer proposed projected SOR as a means for solving the quadratic programming problem: maximize $\mathbf{f(x) = b^T x - (x^T Ax)/2}$ subject to the constraint $\mathbf{x \geq 0}$ [2],[6]. This equation can be used to approximate the solution in the case of solving free boundary problems for journal bearings using Christopherson's method and also in the linear complementarity problem. It should be noted in both of these cases that the matrix $\mathbf{A}$ is often large and sparse. This suggests that projected SOR may be useful in the case of image deblurring due to the known non-negativity of solutions and the size and frequent sparsity of the matrix $\mathbf{A}$ in these problems.

2.3 Projected Landweber Method

Projected Landweber method involves attaching non-negativity constraints to a general Landweber iteration. The Landweber method is a simple iterative scheme commonly applied to find the solutions to least squares problems [4]. It is presented in a number of texts as a fundamental example of a stationary iterative methods and is employed in various capacities including seismic topography and image deblurring [3] [4], [9], [15]. A general Landweber iteration takes the form:

$$\mathbf{x^{[k+1]} = x^{[k]} + \omega A^T (b - Ax^{[k]})} \tag{2.3}$$

where ω is a weighting parameter. Convergence analysis shows that standard Landweber iterations converge for $0 < \omega < 2\|\mathbf{A^T A}\|_{\mathbf{2}}^{-\mathbf{1}}$ when $\mathbf{A}$ is full rank [9]. We also remark that several iterative methods exist which are similar to the basic Landweber iteration, notably Cimmino's method [3].

Similarly to projected SOR, implementing a non-negativity constraint on the Landweber method proves relatively simple. However, unlike projected SOR, where non-negativity was enforced on each $\mathbf{x_i}$ within the iteration, the Landweber method completes each iteration then checks the entire vector $\mathbf{x^{[k+1]}}$, changing negative entries to zero. Fortunately, as was the case with SOR, the addition of projection to the Landweber method does not change the criteria for convergence and $0 < \omega < 2\|\mathbf{A^T A}\|_{\mathbf{2}}^{-\mathbf{1}}$ suffices once again [15]. Also, implementing the non-negativity constraint introduces no multiplicative or additive operations, keeping the costs of the projected Landweber iteration comparable with those of the regular iteration.

2.4 Interior-Point Gradient Method

The final method introduced for this thesis is the interior-point gradient method (IPGM), a scaled gradient descent method for non-negative problems such as occur in image deblurring [11]. Implementation of the IPGM algorithm is slightly more complicated than either SOR or Landweber, where given $\mathbf{x^{[0]}} > \mathbf{0}$, each iteration is calculated by the following algorithm [11]:

Step 1 Compute $\mathbf{p^{[k]}} = -\mathbf{d^{[k]}} \circ \nabla\mathbf{q^{[k]}}$ ($\circ$ denotes component-wise multiplication), where $\nabla\mathbf{q^{[k]}} = \mathbf{A^T A x^{[k]}} - \mathbf{A^T b}$ and $\mathbf{d_i^{[k]}} = \mathbf{x_i^{[k]}}/(\mathbf{A^T A x^{[k]}})_\mathbf{i}$ for $\mathbf{i} = \mathbf{1} : \mathbf{n}$.

Step 2 Choose $\tau_\mathbf{k} \in [\tau, \mathbf{1})$ and let $\alpha_\mathbf{k} = \mathbf{min}(\tau_\mathbf{k}\hat{\alpha}_\mathbf{k}, \alpha_\mathbf{k}^*)$, where $\hat{\alpha}_\mathbf{k} = \mathbf{max}(\alpha : \mathbf{x^k} + \alpha\mathbf{p^{[k]}} \geq \mathbf{0})$ and $\alpha_\mathbf{k}^* = -(\mathbf{p^{[k]}})^\mathbf{T}\nabla\mathbf{q^{[k]}}/\mathbf{p^{[k]}})^\mathbf{T}\mathbf{A^T A p^{[k]}}$.

Step 3 Set $\mathbf{x}^{[\mathbf{k+1}]} = \mathbf{x}^{[\mathbf{k}]} + \alpha_{\mathbf{k}}\mathbf{p}^{[\mathbf{k}]}$.

The τ parameter serves as a weighting that measures the maximum step size of a given iteration. It can be shown that IPGM converges for $0 < \tau < 1$, and a detailed proof of this convergence can be found in Merritt and Zhang's paper on the algorithm [11]. Like the projected Landweber and projected SOR methods, the IPGM method also utilizes projection. This adds slightly more regularization to the solution and suggests IPGM's applicability to image deblurring problems. The cost of the IPGM, while slightly higher than either of the SOR methods or projected Landweber, is still of order n^2 per iteration.

Chapter 3

Filtering Properties

We mentioned in the discussion of the mathematical background that several iterative methods display inherent filtering behavior. This characteristic enables successive approximate solutions obtained by these methods to automatically filter out portions of the solution corresponding to smaller singular values σ_i and the corresponding oscillatory $\mathbf{v_i}$ vectors which pollute the solution. For the basic Landweber iteration, this inherent filtering is easily observable and applies also in the case of the projected method. The analysis of the filtering behavior for the SOR method is not straightforward, but rather instead the possible filtering will be discussed in the form of the symmetric SOR method (SSOR). We remark that analysis of the behavior of IPGM is significantly more difficult and is not attempted in this thesis.

3.1 Filtering in Landweber

From equation (2.3), it can be seen that the Landweber method can be rearranged to the basic iteration,

$$\mathbf{x^{[k+1]}} = \omega \mathbf{A^T b} + (\mathbf{I} - \omega \mathbf{A^T A})\mathbf{x^{[k]}}$$

It then follows that if we use the initial solution $\mathbf{x^{[0]}} = \underline{\mathbf{0}}$, then each successive iteration yields,

$$\begin{aligned}
\mathbf{x^{[1]}} &= \omega\mathbf{A^T b} \\
\mathbf{x^{[2]}} &= \omega[\mathbf{I} + (\mathbf{I} - \omega\mathbf{A^T A})]\mathbf{A^T b} \\
\mathbf{x^{[3]}} &= \omega[\mathbf{I} + (\mathbf{I} - \omega\mathbf{A^T A}) + (\mathbf{I} - \omega\mathbf{A^T A})^{\mathbf{2}}]\mathbf{A^T b}
\end{aligned} \tag{3.1}$$

and in general,

$$\mathbf{x^{[k+1]}} = \mathbf{P_k}(\mathbf{A^T A})\mathbf{A^T b} \tag{3.2}$$

where $\mathbf{P_k}(\lambda)$ is a polynomial given by the expression:

$$\begin{aligned}
\mathbf{P_k}(\lambda) &= \omega[1 + (1 - \omega\lambda) + (1 - \omega\lambda)^2 + \cdots + (1 - \omega\lambda)^k] \\
&= \sum_{j=0}^{k} \omega(1 - \omega\lambda)^j
\end{aligned} \tag{3.3}$$

Using our knowledge of geometric series it can easily be shown that equation (3.3) simplifies to the expression:

$$\lambda\mathbf{P_k}(\lambda) = \mathbf{1} - (\mathbf{1} - \omega\lambda)^{\mathbf{k+1}} \tag{3.4}$$

Next, we use the singular value decomposition of $\mathbf{A^T A}$. Firstly, it should be noted that due to the orthogonality of $\mathbf{U}$ and $\mathbf{V}$,

$$\mathbf{A^T A} = \mathbf{V\Sigma^T U^T U\Sigma V^T} = \mathbf{V\Sigma^T\Sigma V^T}$$

and furthermore,

$$(\mathbf{A^T A})^{\mathbf{k}} = \mathbf{V}(\mathbf{\Sigma^T \Sigma})^{\mathbf{k}}\mathbf{V^T}$$

Using this knowledge, we can then write any k_{th} degree polynomial $\mathbf{P_k}(\mathbf{A^T A})$ in the following form:

$$\mathbf{P_k}(\mathbf{A^T A}) = \mathbf{V}[\mathbf{P_k}(\mathbf{\Sigma^T \Sigma})]\mathbf{V^T} \tag{3.5}$$

Combining equations (3.2),(3.4) and (3.5), we finally see that the standard Landweber iteration can be written as the following,

$$\begin{aligned}
\mathbf{x}^{[\mathbf{k+1}]} &= \mathbf{V}\mathbf{P_k}(\mathbf{\Sigma^T \Sigma})\mathbf{V^T}\mathbf{A^T}\mathbf{b} \\
&= \mathbf{V}\mathbf{P_k}(\mathbf{\Sigma^T \Sigma})\mathbf{\Sigma}\mathbf{U^T}\mathbf{b} \\
&= \sum_{\mathbf{i=1}}^{\mathbf{n}} \sigma_\mathbf{i}\mathbf{P_k}(\sigma_\mathbf{i}^\mathbf{2})(\mathbf{u_i^T b})\mathbf{v_i} \\
&= \sum_{\mathbf{i=1}}^{\mathbf{n}} \sigma_\mathbf{i}^\mathbf{2}\mathbf{P_k}(\sigma_\mathbf{i}^\mathbf{2})\frac{\mathbf{u_i^T b}}{\sigma_\mathbf{i}}\mathbf{v_i} \\
&= \sum_{\mathbf{i=1}}^{\mathbf{n}} [\mathbf{1} - (\mathbf{1} - \omega\sigma_\mathbf{i}^\mathbf{2})^{\mathbf{k+1}}]\frac{\mathbf{u_i^T b}}{\sigma_\mathbf{i}}\mathbf{v_i}
\end{aligned} \tag{3.6}$$

Comparing equation (3.6) with equation (1.6), we see that the Landweber method provides filtering within each iteration, with the iteration dependent filter factors,

$$\phi_\mathbf{i}^{[\mathbf{k}]} = \mathbf{1} - (\mathbf{1} - \omega\sigma_\mathbf{i}^\mathbf{2})^{\mathbf{k+1}} \tag{3.7}$$

Observe that, for instance, if we set $\omega = 1$ and scale the problem such that $1 = \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \approx 0$, then for large σ_i, $\phi_i^{[k]} \approx 1$, and for small σ_i, $\phi_i^{[k]} \approx 0$. Thus, the Landweber method filters out components of the solution corresponding to small singular values. This avoids the pollution of the solution by inverted noise and allows the method to converge quickly.

3.2 SOR Filtering

Next, we consider SOR. First, notice from equation (2.1) that the standard SOR iteration can be rewritten as:

$$\mathbf{x^{[k+1]}} = \mathbf{x^{[k]}} + \mathbf{M^{-1}A^T}(\mathbf{b} - \mathbf{Ax^{[k]}}) \tag{3.8}$$

where $\mathbf{M} = \frac{\mathbf{1}}{\omega}\mathbf{D} + \mathbf{L}$ and $\mathbf{A^TA} = \mathbf{L} + \mathbf{D} + \mathbf{L^T}$ as discussed in section 2.1. Unfortunately, it is difficult to analyze the filtering properties of the SOR iteration in this form. One possible solution would be to view SOR as a preconditioned form of the Landweber iteration, but this in still difficult because the matrix $\mathbf{M}$ is not symmetric positive definite. So, rather than the standard SOR iteration, we will analyze the filtering properties of the symmetric SOR method (SSOR). In this case the matrix $\mathbf{M}$ takes the following form:

$$\mathbf{M} = \frac{\omega}{\mathbf{2} - \omega}(\frac{\mathbf{1}}{\omega}\mathbf{D} + \mathbf{L})\mathbf{D^{-1}}(\frac{\mathbf{1}}{\omega}\mathbf{D} + \mathbf{L})^{\mathbf{T}}$$

If $\mathbf{A^TA}$ is symmetric positive definite and $0 < \omega < 2$, then $\mathbf{M}$ is symmetric positive definite [12]. This fact allows us to make the following observations:

- If $\mathbf{A^TA}$ is symmetric positive definite, then the diagonal entries of $\mathbf{D} = \mathbf{diag}(\mathbf{A^TA})$ are all positive.
- Let $\mathbf{R} = \sqrt{\frac{\omega}{\mathbf{2}-\omega}}\mathbf{D^{-1/2}}(\frac{\mathbf{1}}{\omega}\mathbf{D} + \mathbf{L})^{\mathbf{T}}$, where $\mathbf{D^{1/2}} = \mathbf{diag}(\sqrt{\mathbf{d_{11}}}, \sqrt{\mathbf{d_{22}}}, \cdots, \sqrt{\mathbf{d_{nn}}})$. Then $\mathbf{M} = \mathbf{R^TR}$ is the Cholesky factorization of $\mathbf{M}$.

Applying these observations to the iteration in equation (3.8), we get

$$\begin{aligned}
\mathbf{x^{[k+1]}} &= \mathbf{x^{[k]}} + (\mathbf{R^T R})^{-1}(\mathbf{A^T b} - \mathbf{A^T A x^{[k]}}) \\
&= \mathbf{x^{[k]}} + \mathbf{R^{-1} R^{-T}}(\mathbf{A^T b} - \mathbf{A^T A x^{[k]}}) \\
\mathbf{R x^{[k+1]}} &= \mathbf{R x^{[k]}} + (\mathbf{R^{-T} A^T b} - \mathbf{R^{-T} A^T A x^{[k]}}) \\
&= \mathbf{R x^{[k]}} + (\mathbf{R^{-T} A^T b} - \mathbf{R^{-T} A^T A R^{-1} R x^{[k]}}) \qquad (3.9)
\end{aligned}$$

If we let $\hat{\mathbf{A}} = \mathbf{A R^{-1}}$ and $\hat{\mathbf{x}} = \mathbf{R x^{[k]}}$, then equation (3.9) is in the form of the basic Landweber iteration:

$$\hat{\mathbf{x}}^{[\mathbf{k+1}]} = \hat{\mathbf{x}}^{[\mathbf{k}]} + (\hat{\mathbf{A}}^{\mathbf{T}}\mathbf{b} - \hat{\mathbf{A}}^{\mathbf{T}}\hat{\mathbf{A}}\hat{\mathbf{x}}^{[\mathbf{k}]}) \qquad (3.10)$$

Thus, the convergence and filtering associated with the SSOR iteration depends on the singular values of the matrix $\hat{\mathbf{A}} = \mathbf{A R^{-1}}$, and follows the same analysis applied in the previous subsection to the Landweber method. It should be noted that each SSOR iteration costs twice as much as a regular SOR or projected SOR iteration, and although convergence is nearly twice as fast this is significant enough to deter us from implementing SSOR for this thesis [12]. Also, whereas there has been previous analysis on the behavior of the SOR and projected SOR methods in the case of ill-posed problems, we do not know of any such analysis for SSOR. Instead, keeping in mind the filtering properties of SSOR analyzed in this section, we will observe from the semi-convergent behavior of the results for SOR and projected SOR in the subsequent chapters that SOR and projected SOR also exhibit filtering in their solutions.

Chapter 4

Numerical Experiments

In order to evaluate the four presented methods in the context of image deblurring, we perform selected numerical experiments. Specific topics for testing include the determination of optimal weighting parameter, the effect of random noise level on the effectiveness of each method, number of iterations required for an optimal solution, and general remarks as to the resultant usefulness of each method.

Experiments used the `blur` function from the Regularization Tools package for Matlab [8]. The `blur` function is a test problem which simulates the deblurring of images by atmospheric turbulence, modeled by a Gaussian point-spread function. `[A,b,x] = blur(n,band,sigma)` results in an $\mathbf{A}$ matrix which is a sparse, symmetric $(n^2 \times n^2)$ block Toeplitz matrix with Toeplitz blocks. The `band` parameter controls the bandwidth while `sigma` controls the shape of the Gaussian point-spread function. Typically, experiments were run for `blur(256,3,0.7)`, where the chosen `band` and `sigma` values were the default values of the function.

In addition to the system created by the `blur`, random noise was added to the system to create a realistic image deblurring problem. The noise added to the system takes the form

$$\mathbf{noise} = \frac{\mathbf{nlevel} * \|\mathbf{b_{true}}\|_\mathbf{2} * \mathbf{n}}{\|\mathbf{n}\|_\mathbf{2}}$$

where $\mathbf{n}/\|\mathbf{n}\|_\mathbf{2}$ is a normalized vector of random noise values (chosen from a normal distribution with $mean = 0$ and $variance = 1$), and **nlevel** amounts to a percentage

value of $\mathbf{b_{true}}$ used to weight the noise for a given problem. Once the selected amount of noise has been selected and added to $\mathbf{b_{true}}$, we are ready to run experiments with the blurred image.

4.1 Parameters

The first set of experiments were designed to determine good weighting parameters for each method within the context of the test problem. For SOR and projected SOR this constituted testing various $0 < \omega < 2$. For simplicity and comparison projected Landweber was also run for the same ω parameter as SOR and projected SOR rather than explicitly calculating $2\|\mathbf{A^T A}\|_{\mathbf{2}}^{\mathbf{-1}}$. The τ for IPGM was chosen and tested within the dictated range of convergence and held constant throughout each full run of the method rather than using the optional reselection of τ detailed in step 2 of the algorithm. Figure 4.1 shows the results for each method. Note that for the graphs in figure 4.1 and all subsequent graphs, the x axis represents the iteration of each method, while the y axis measures the relative error of the solution given by $\frac{\|\mathbf{x_k}-\mathbf{x_{true}}\|}{\|\mathbf{x_{true}}\|}$.

It is easily observed that the best results for SOR and projected SOR came for $\omega \ll 1$, or severe under-relaxation of the system. Although this result seems counterintuitive on the surface, other tests have found similar results in the application of both projected and unprojected SOR methods to imaging problems [18]. On the other hand, the projected Landweber method does not exhibit this trend, instead tending to obtain the best solutions within the fewest iterations for $\omega > 1$. Similarly, IPGM achieves its most rapid convergence and minimum error for τ values nearer to the upper bound required for convergence. Using the results obtained, preferable parameters were chosen for each method, and will be used throughout the rest of the

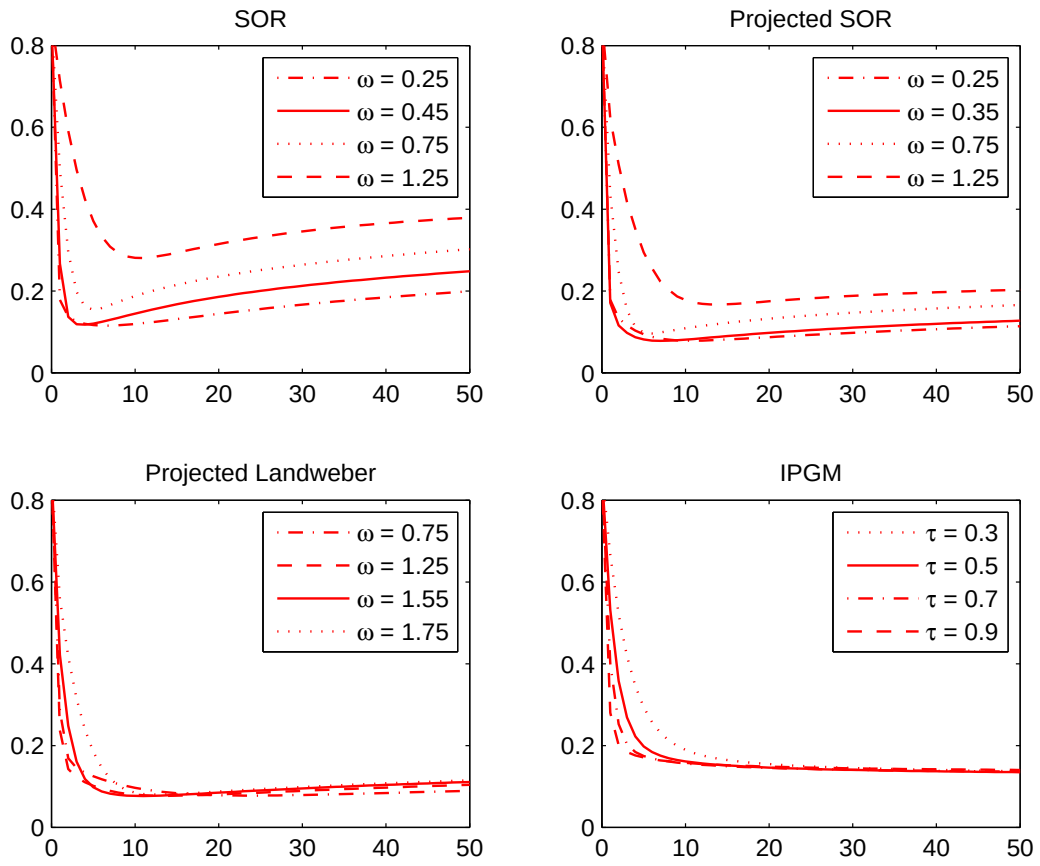

Figure 4.1: Relative error versus iteration for different parameter values. The relative error of the chosen optimum parameter is plotted with the solid line, displayed along with a few other tested parameters for the sake of comparison.

thesis when running each method.

4.2 Noise Level

After testing each method to determine an optimal weighting parameter, the next area of concern is the effect of noise on the quality of solution. For this, the methods were first compared for the effect of different concentrations of noise on the solution using a single method (Figure 4.2), and second to test each method versus the others for different levels of noise (Figure 4.3).

As would be expected, each method obtained poorer results for higher noise values. Despite this seemingly intuitive result, useful observations can be drawn. Firstly, increased noise value tended to make the semi-convergence more severe. For 10% percent noise, SOR, projected SOR, and projected Landweber reached a minimum relative error after a few iterations, after which the error steeply rose, polluting the solution. However, it was noticed that the projected methods, particularly projected SOR and projected Landweber, seem less vulnerable to increased noise than regular SOR. While regular SOR reaches its minimum relative error after only a few iterations, it then begins to increasingly worsen at a quicker rate than the projected methods. This makes the institution of stopping criteria and the location of the iteration $\mathbf{k}^{[\mathbf{opt}]}$ much more sensitive, a potentially detrimental characteristic.

Also of interest, while the minimum relative error was poorer for higher noise levels, IPGM did not exhibit semi-convergence. This suggests that the method has some extra regularization, a characteristic which may be of interest if the stopping criteria are volatile or difficult to determine.

The second set of tests compared each method's minimum error for multiple selected noise values. The results in Figure 4.2 show that the projected SOR and

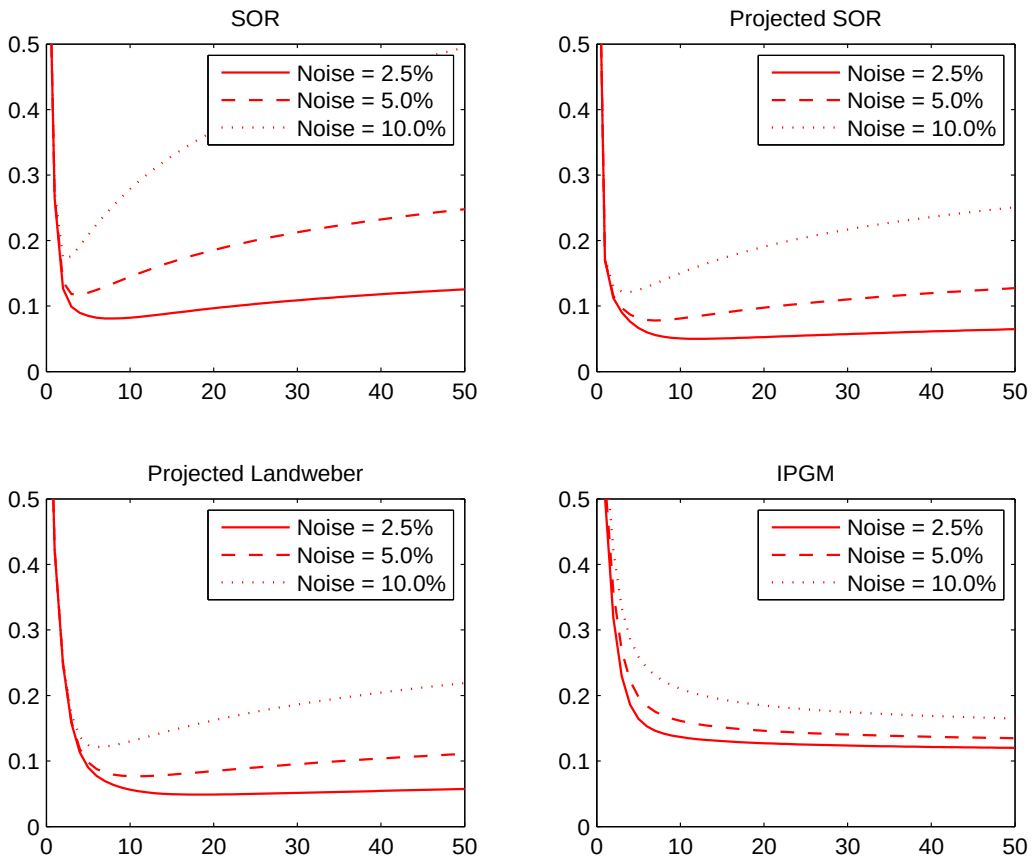

Figure 4.2: Relative error and semi-convergence as affected by noise levels.

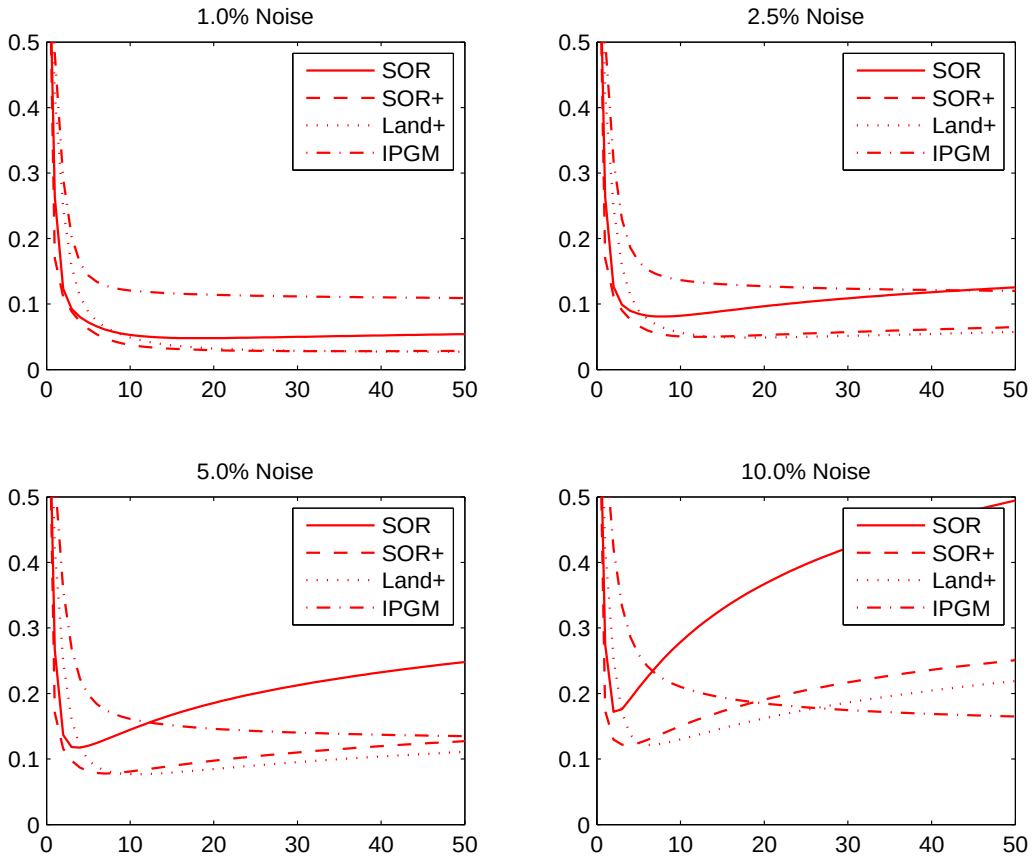

Figure 4.3: Relative error for each method for varying noise levels.

Landweber methods tend to have the minimum errors and quickest convergence for each noise value. However, projected SOR tends to achieve this minimum in fewer iterations, making it preferable in terms of cost. This tends to be a common characteristic of both projected and un-projected Landweber method, and other literature has suggested the use of pre-conditioning to accelerate convergence and increase projected Landweber method's usefulness in the field of image deblurring [15]. We remark that projected SOR can be viewed as a preconditioned version of projected Landweber.

4.3 Stopping Criterion

The final experiment run was to determine suitable weighting parameter δ for the discrepancy principle to establish effective stopping criteria. Specifically, the experiments were designed to test the tendency of each method to over-smooth (stop too early) or under-smooth (stop too late) the reconstructed image. To accomplish this, each method was tested on the `blur` function using ten percent random noise and various levels of image convolution, produced by increasing the bandwidth and distribution of the PSF.

In general, using $\delta = 1$ provided encouraging results for all methods. Interestingly, SOR produced slightly under-smoothed solutions while projected SOR displayed a tendency to over-smooth. However, it is notable that in most cases, particularly for simulations with increased blur, the projected SOR solution method stopped at a much lower relative error in fewer iterations. This suggests that for very blurry images, projected SOR presents itself as a more viable alternative to SOR in terms of both quality and cost. Similarly, projected Landweber produced slightly over-smoothed solutions, but notably required fewer iterations to achieve its minimum relative error than either SOR or projected SOR. This characteristic could prove useful to minimize

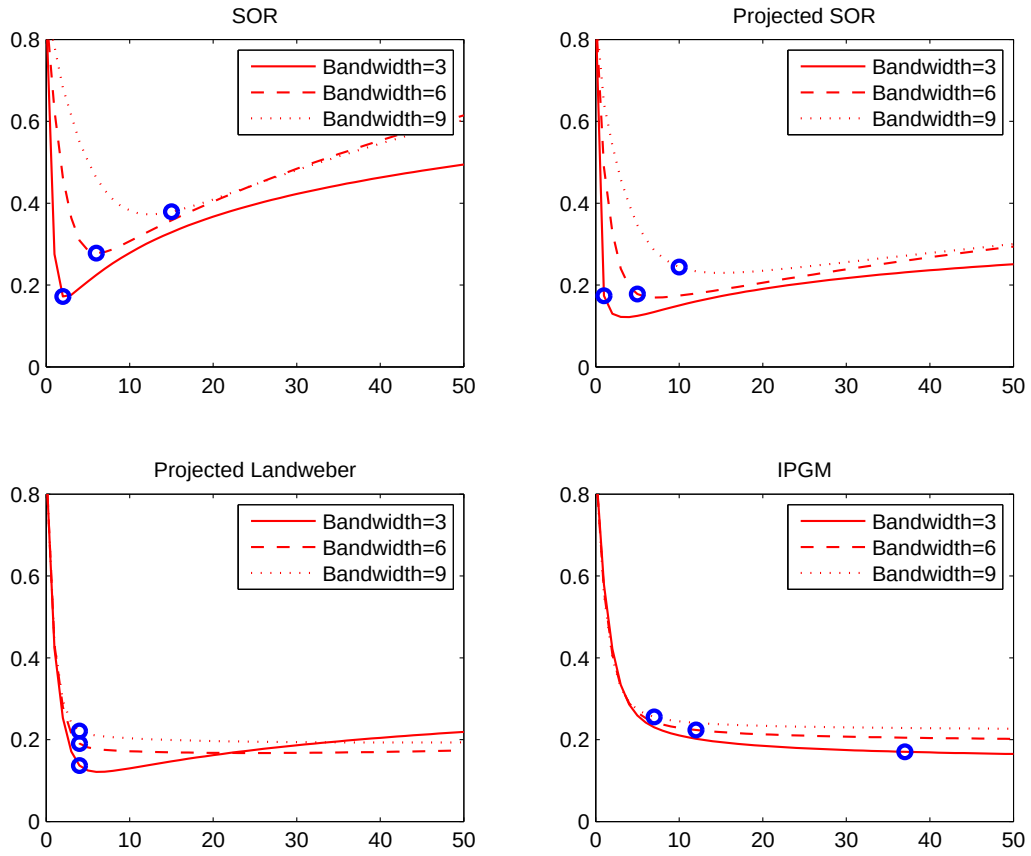

Figure 4.4: These graphs display the effectiveness of the discrepancy principle as the stopping criterion for each method. Each graph displays the results for the individual method for $\delta = 1$, 10% noise, and varying levels of blur.

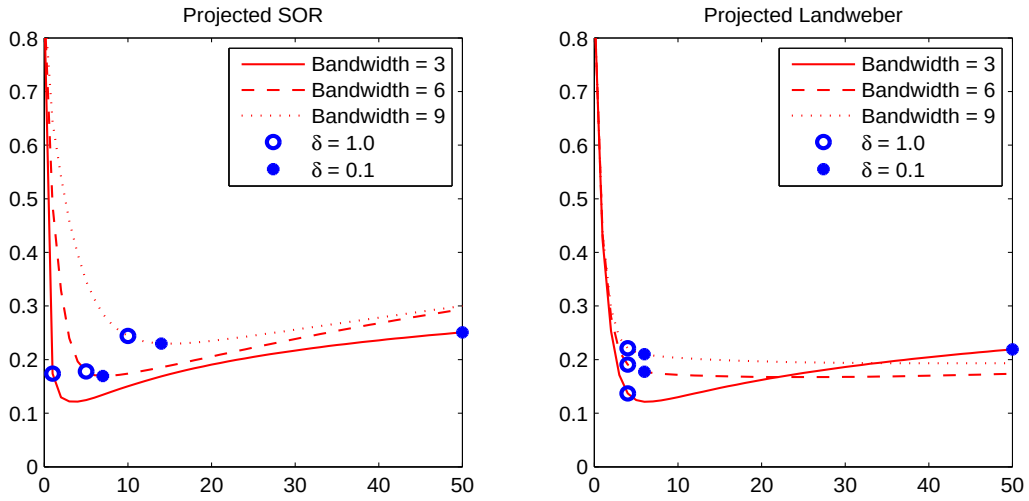

Figure 4.5: For problems with smaller bandwidth, $\delta = 0.1$ proved ineffective and led to significant under-smoothing of the solution. However, as bandwidth increased, $\delta = 0.1$ proved useful, helping to correct over-smoothing while avoiding under-smoothing the solution. This is particularly observable for the projected SOR method.

costs or in choosing a stopping iteration in cases when a good estimate of the noise is unknown. IPGM also over-smoothed, and provided comparatively poor results in the case of slightly blurred images, but displayed some potentially beneficial traits for the tests with increased blur. In cases of more blurred problems, IPGM tended to stop after fewer iterations while the relative error of the solutions increased only moderately. This suggests that for highly blurred images, IPGM may be a viable option in terms of cost and solution quality.

In order to alleviate the effects of over-smoothing on projected SOR and projected Landweber methods, further tests were run for differing values of δ. Despite IPGM solutions also being over-smoothed, no tests to limit the over-smoothing were made due to the already high number of iterations required in many cases and the failure of the relative error to exhibit semi-convergence. The results of the additional tests for projected SOR and projected Landweber indicate that $\delta = 0.1$ helped to alleviate

the over-smoothing of solutions for both methods (see Figure 4.5). Although this improvement did not hold for problems with less blur, $\delta = 0.1$ proved effective weighting for more blurred images and may be useful in many cases.

Chapter 5

Experiments on Simulated Problems

Having established good values for the weighting parameters and discrepancy principle via experiment as well as experimenting with the effect of noise on each of the methods, we will test the effectiveness of the methods on simulated problems using real world images. The two images used are a satellite and an x-ray of a hand such as might be found in astronomical or medical imaging respectively (see Figures 5.1,5.2). For the tests, both the satellite and x-ray image were tested for the same type of blur modeled by the `blur` function. However, to test the effectiveness of each method, the tests were run on more severely blurred images than used in experiments for determining parameters, with bandwidth= 12, $\sigma = 2.0$ for the PSF, and 10 percent noise.

5.1 Tests on Real Images

The tests run on the satellite image produced interesting results (see Figure 5.1). It is easily observed that projected SOR represents an improvement over the regular SOR method, both in terms of minimum error and reconstruction quality. While some of this may be due to the under-smoothing of the regular SOR solution, the

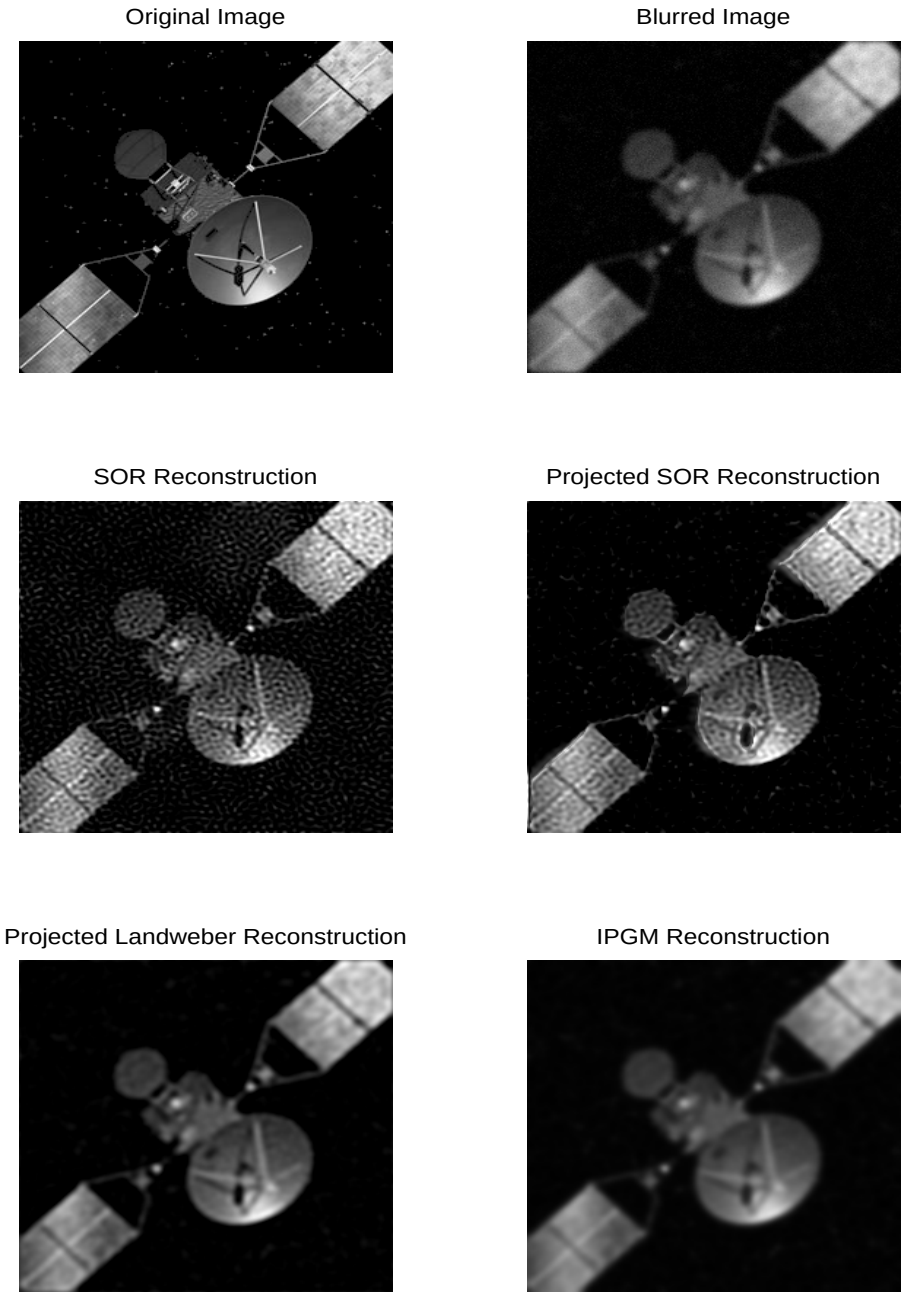

Figure 5.1: The reconstructions of the satellite image after blurring.

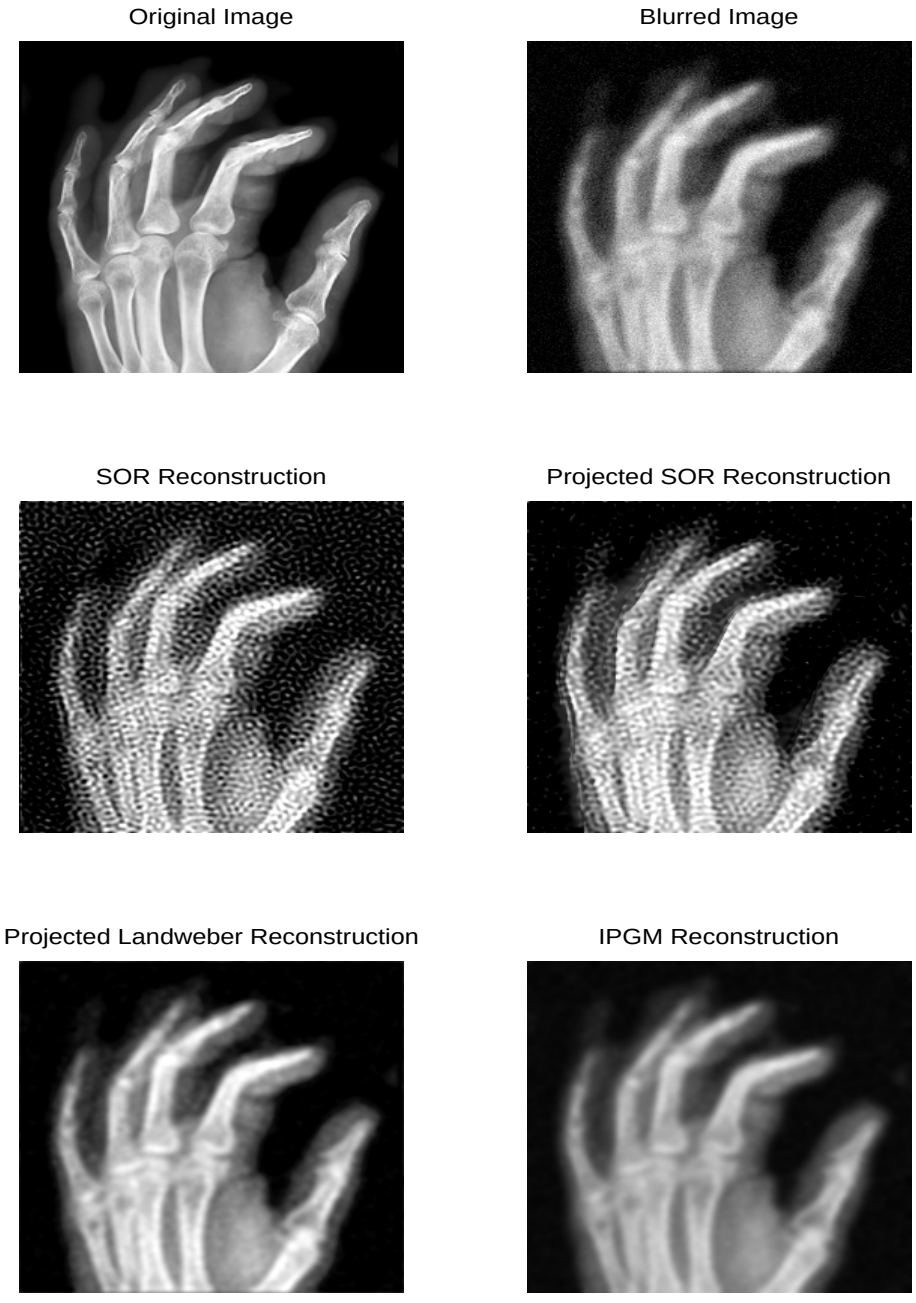

Figure 5.2: The reconstructions of the xray image after blurring.

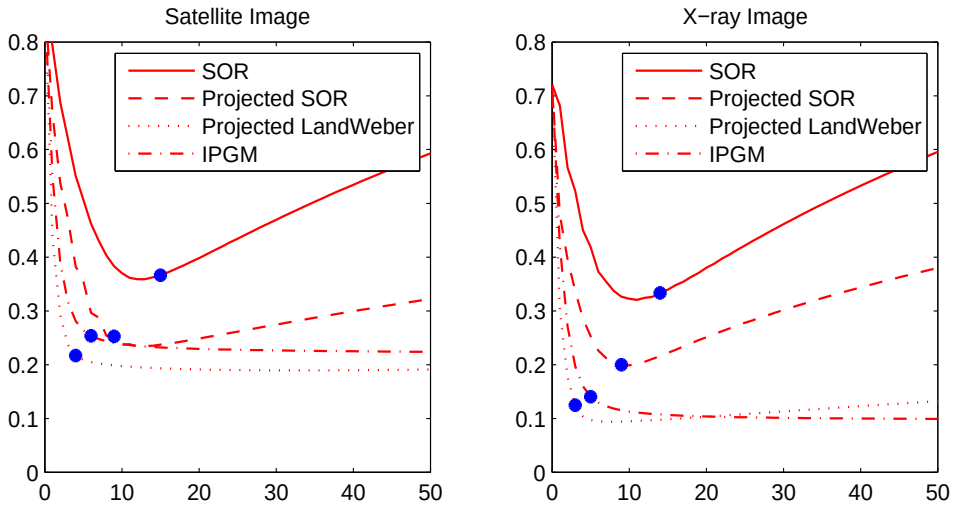

Figure 5.3: Comparing the relative error and stopping iteration for both the satellite and the x-ray image, projected SOR represents a marked improvement on regular SOR, while projected Landweber and IPGM provide competitive alternatives in terms of minimizing solution error.

graph of the relative error shows that projected SOR represents a better choice than its unprojected alternative. Comparing projected SOR to projected Landweber and IPGM, it is noticeable that both projected Landweber and IPGM produce over-smoothed solutions while projected SOR produces a more grainy solution. However, from Figure 5.3 we see that projected SOR stops near to its minimum relative error and that all three methods stop at comparable relative error values. This suggests that while all three may provide solutions that are equal in terms of mathematical accuracy, each may be preferable for different applications depending on the characteristics of a desirable reconstruction.

Looking at the graph of the relative error for the hand x-ray image, we see an across the board improvement in the minimum relative error and relative error at the stopping iteration (see Fig. 5.3). Comparisons between the methods themselves yield similar results to those for the satellite image, with projected Landweber and IPGM

attaining the lowest relative errors and taking the fewest iterations, while still producing over-smoothed solutions. Projected SOR once again represented a significant improvement over the regular SOR iteration in terms of cost and quality, but despite stopping close to its minimum relative error, the projected SOR reconstruction of the hand x-ray appears over-smoothed and grainy in comparison with the projected SOR reconstruction of the satellite image. In contrast, the over-smoothed reconstructions using projected Landweber and IPGM seem to show the joints and finer details of the image more clearly. This once again suggests one method may be preferable based on the particular types of images or desired characteristics of the reconstruction.

5.2 Conclusions

The results of the image restoration experiments enable us to make several conclusions about the usefulness of projection for image deblurring. Firstly, both in terms of the quality of image reconstruction and the relative error of the solution, the projected methods produced far better, more accurate solutions than the unprojected SOR method. The projected methods were also preferable in terms of cost, taking fewer iterations for convergence. This suggests that despite the non-triviality of implementing non-negativity constraints for many methods, projection is a useful and worthwhile constraint for image deblurring solutions.

With respect to individual methods, projected SOR, projected Landweber, and IPGM each displayed useful characteristics. In comparison with the other projected methods tested for this thesis, projected SOR was unique in that it did not over-smooth solutions. This benefit is slightly counteracted by the increased cost and higher relative error of the projected SOR solutions, but could prove useful in applications where over-smoothing needs to be avoided. Also of note, projected SOR's re-

construction of both the satellite and hand x-ray images show a lot of high frequency oscillations. This suggests pollution by the inverted noise, a negative consequence which may possibly be improved with the implementation of additional regularization or filtering.

Projected Landweber produced the best solutions in terms of relative error and cost, but tended to over-smooth solutions. In order to avoid this over-smoothing, significantly higher cost would be incurred, decreasing the viability of the method for practical purposes. One suggested way to avoid this additional cost and increase the efficiency of the method is pre-conditioning, which would give quicker convergence, allowing greater regularization within the same number of iterations [15]. It would be interesting to investigate the SSOR method further as a pre-conditioner for the projected Landweber method. Projected Landweber also displayed less severe semi-convergence than projected SOR, suggesting that it may be preferable in instances with higher noise values or when a good approximation of the noise is unknown.

The IPGM algorithm proved the most robust of the methods implemented for this thesis, not exhibiting the semi-convergent behavior of the other methods. In the case of the simulated image deblurring problems, IPGM produced comparable solutions in terms of relative error, but like projected Landweber tended to over-smooth solutions. This suggests that, if combined with pre-conditioning or some alternative method for decreasing the iterations required for better solution, IPGM could prove useful for problems with large or unknown amounts of noise.

In conclusion, while the implementation of non-negativity can prove non-trivial in many cases, it represents a logical and useful constraint in image deblurring problems. In addition, each of the three projected methods in this thesis displayed potential utility for image deblurring, but the choice of a particular method may be determined by the characteristics of the image, noise, and other characteristics of a given prob-

lem. Finally, for each of the methods, further work may improve the efficiency and regularization to improve upon the solutions produced by the projected methods.

Bibliography

[1] J. M. Bardsley and J. G. Nagy. Convariance-preconditioned iterative methods for non-negatively constrained astronomical imaging. *SIAM J. Matrix Anal. Appl.*, 27:1184–1197, 2006.

[2] M. Benzi. Remarks on the numerical solution of certain numerical complementarity problems. *Journal of Computational and Applied Mathematics*, 83:137–143, 1997.

[3] M. Benzi. Gianfranco Cimmino's contributions to numerical mathematics. In *Seminariodi Analisi Matematica, Dipartimento di Matematica dellUniversit di Bologna, Ciclo di Conferenze in Ricordo di Gianfranco Cimmino, Marzo-Maggio 2004*, pages 87–109. Technoprint, Bologna, Italy, 2005.

[4] M. Bertero and P. Boccacci. *Introduction to Inverse Problems in Imaging*. Institute of Physics Publishing, Bristol, UK, 1998.

[5] A. Björck. *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1996.

[6] C. W. Cryer. The solution of a quadratic programming problem using systematic overrelaxation. *SIAM J. Control*, 9:385–392, 1971.

[7] J. W. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997.

[8] P. C. Hansen. Regularization tools. *Numerical Algorithms*, 46:189–194, 2007.

[9] P. C. Hansen. *Discrete Inverse Problems: Insight and Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2009.

[10] W. Wein M. Blume, D. Zikic and N. Navab. *Medical Image Computing and Computer-Assisted Intervention*. Springer Berlin, Heidelberg, Germany, 2007.

[11] M. Merritt and Y. Zhang. Interior-point gradient method for large-scale totally nonnegative least squares problems. *Inverse Problems*, 13:441–463, 1997.

[12] G. Meurant. *Computer Solution of Large Linear Systems*. Elsevier Science B.V., Amsterdam, Netherlands, 1999.

[13] S. Suryanarayanan N. Raghunath, T.L. Faber and J.R. Votaw. Motion correction of pet brain images through deconvolution: II. practical implementations and algorithm optimization. *Phys. Med. Biol*, 54.3:813–829, 2009.

[14] J. G. Nagy P. C. Hansen and D. P. O'Leary. *Deblurring Images: Matrices, Spectra, and Filtering*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2006.

[15] M. Piana and M. Bertero. Projected landweber method and preconditioning. *Inverse Problems*, 13:441–463, 1997.

[16] T.J. Schulz. Multiframe blind deconvolution of astronomical images. *J. Opt. Soc. Am. A*, 10:1064–1073, 1993.

[17] J.A. Seibert and J.M. Boone. X-ray scatter removal by deconvolution. *Med. Phys.*, 15.4:567–575, 1988.

[18] V. N. Strakhov and S. V. Vorontsov. Digital image deblurring with SOR. *Inverse Problems*, 24:1–17, 2008.

[19] B.E. Stribling T.J. Schulz and J.J. Miller. Multiblind deconvolution with real data: Imagery of the hubble space telescope. *Optics Express*, 11:355–362, 1997.

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