A Guide to GIDE 5.1: A GUI for Graphical Image Deblurring Exploration

Brianna R. Cash and Dianne P. O'Leary
http://www.cs.umd.edu/users/oleary/software

February 2, 2015

Abstract

We describe the use of a MATLAB tool called GIDE that allows user-aided deblurring of images. GIDE helps practitioners restore a blurred grayscale image using their knowledge or intuition about the true image, but safeguarding from possible bias by validation using statistical diagnostics based on an assumption of Gaussian added noise. GIDE allows practitioners (or students) to visually explore the range of statistically likely solutions resulting from any of three regularization methods: Tikhonov, truncated SVD, and total variation.

Contents

1 Introduction **3**

2 A Brief Overview of GIDE **3**

3 Mathematical Model of Imaging and Regularization **5**
3.1 Continuous Representation 7
3.2 Discrete Representation 7
3.3 Constructing the Blurring Matrix from the Point Spread Function 8

4 Discrete Regularization Methods **8**
4.1 SVD-Based Regularization Methods 8
4.2 TV Regularization for the Continuous Problem 9
4.2.1 Formulation of Newton's Method 10
4.2.2 Linearization Based on Introducing a New Variable 10
4.3 Discretization of the TV Regularization Method 10
4.3.1 Discrete Formulation of Newton's Method 11
4.3.2 Formulation of Primal-Dual Newton's Method 11

5 Initial Parameter Selection **13**
5.1 Generalized Cross-Validation (GCV) 13
5.2 Discrepancy Principle 14

6 Statistical Diagnostics **14**
6.1 Choice of Diagnostics 14
6.1.1 Residual Diagnostic 1 15
6.1.2 Residual Diagnostic 2 15
6.1.3 Residual Diagnostic 3 15
6.2 Validation of Residual Diagnostics 15

7 Using the Software **16**
7.1 Installation 17
7.2 Choose Blur 17
7.3 Choose Image 17
7.4 Choose Method 17
7.5 Compute 18
7.6 Limitations 18

8 Results and Testing **18**
8.1 Test Images 18
8.2 Signal-to-Noise Ratio Effect on Diagnostics 19
8.3 Effects of γ on Computation Time 19
8.4 Results on Larger Images and Varied PSF 21

9 Summary and Conclusions **21**

1 Introduction

Restoring a blurred image requires choice of a regularization method and associated parameter. Different choices can lead to a very wide variety of reconstructed images. Practitioners faced with these choices might favor results biased by what they expect to see and thereby introduce image artifacts or miss true image features.

This is demonstrated in Figure 1. The practitioner might not expect the moon to have train tracks and may favor a reconstruction like the reconstructed image in the center of the figure, where that information is lost.

To avoid bias, we developed a methodology for method choice and parameter selection that uses three statistical diagnostics to validate solutions, under the assumption of Gaussian additive noise in a blurred grayscale image.

We packaged the methodology into MATLAB software called GIDE, Graphical Image Deblurring Exploration, including a user-friendly graphical user interface (GUI). The software was built upon James Nagy's RESTORETOOLS package [6]. It allows practitioners (or students) to visually explore the range of statistically likely solutions resulting from any of three regularization methods: Tikhonov, truncated SVD, and total variation.

2 A Brief Overview of GIDE

Medical and scientific imaging takes raw noisy data from a scientific instrument (MRI, CT, astronomical camera, etc.) and processes ("deblurs") this data to produce images that are useful to practitioners. These rather expensive images are often critical in making medical or scientific decisions, so it is important that the deblurring is performed well. Unfortunately, image deblurring is an example of an ill-posed inverse problem: small changes (e.g., noise) in the data can make arbitrarily large changes in the deblurred image.

To overcome this limitation, we typically use our knowledge about the particular problem to formulate constraints [3]. For example, we might choose to

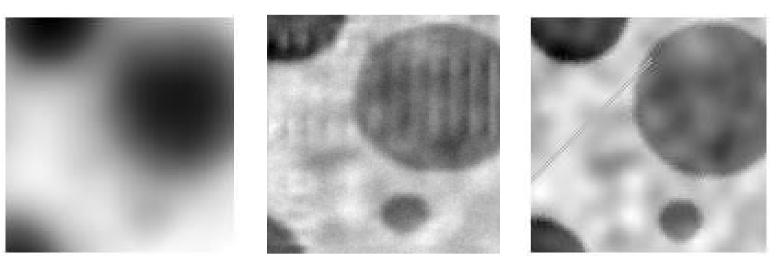

Figure 1: Left: Blurred Image, Center: Deblurred Image, Right: True Image which has "train tracks". Without knowledge that the true image has "train tracks" one might accept the deblurred result without realizing that important information has been lost.

limit the norm of the solution image (Tikhonov regularization), approximate the blurring operator by a low-rank matrix (Truncated Singular Value Decomposition (TSVD) regularization), or limit the Total Variation (TV) of the solution. Including these constraints makes the problem well-posed, and thus the new problem has a well-determined solution that we hope is near the true solution of the original problem. Once we choose a regularization method, we also need to choose a parameter that we call γ. These choices are not easy.

In this work we take the viewpoint of a practitioner looking for a good reconstruction of a single image, or of a student gaining experience with various methods. Figure 2 shows that different regularization methods can yield very different reconstructions. But choosing an appropriate method and parameter for a given problem is difficult, relying on properties of the particular problem and knowledge of the application area. Practitioners often have invaluable experience that is crucial in finding a good approximate solution, but too much reliance on intuition can lead them to see what they expect to see, rather than the true solution. Any candidate reconstruction should be validated using statistical analysis.

We present a methodology and software with a graphical user interface (GUI) that can be used by practitioners to choose an appropriate regularization method and associated parameter while reducing the bias that can be introduced by choosing based on seeing a visually appealing reconstruction. This methodology gives practitioners the ability to compare regularization methods by presenting a plausible range for γ and by presenting results of statistical tests of plausibility of each candidate image as a solution to the original ill-posed problem.

Our proof-of-concept software package GIDE (Graphical Image Deblurring Exploration) was built in MATLAB using the RESTORETOOLS package [6]. Figure 3 shows a screen shot of the interface. The GUI is easy to use. After typing `startGIDE`:

- A user can either provide a blurred image or choose from samples that we provide.

- Similarly, a user either provides a blurring matrix or a separable Point Spread Function (PSF), or chooses the default boxcar or Gaussian blurs. (See Section 7.2 for the definition of these blurs.) Every new problem specification generates a different random noise sample, so final results will change.

- After selecting one of the regularization methods (TSVD regularization, Tikhonov regularization, or TV regularization), clicking COMPUTE produces an initial solution based on automatic selection of the regularization parameter γ.

- The resulting deblurred image appears, along with other information, including the results of the statistical diagnostics. For each diagnostic, in addition to detailed information, either YES or NO is displayed, indicating whether or not it is satisfied.

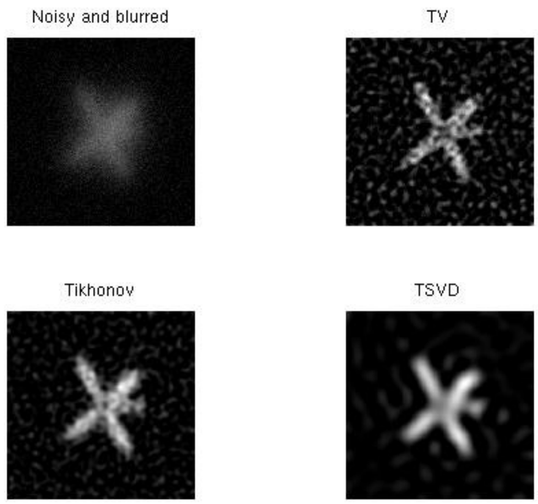

Figure 2: Restorations of the 256 × 256 blurred satellite image provided in RestoreTools with zero boundary conditions, signal-to-noise ratio SNR=9, and parameter γ chosen by the automatic parameter selection methods detailed in Section 5. This indicates the wide variety of results that can be produced by regularization methods.

- The user then uses the slide bar to adjust γ. This changes the resulting image and diagnostics in real time, allowing the user to explore the range of statistically plausible solutions. The blurred image, true image (if available), and deblurred image are also displayed, for convenience, in a separate figure.

In the remainder of this user's manual we briefly review the deblurring problem in Section 3, the regularization methods in Section 4, initial parameter choices in Section 5, statistical diagnostics in Section 6, and gide installation and use in Section 7. Finally we discuss testing and validation in Section 8 and present a summary and conclusion in Section 9.

3 Mathematical Model of Imaging and Regularization

An image can be thought of either as a real continuous function or as a collection of discrete square pixels.

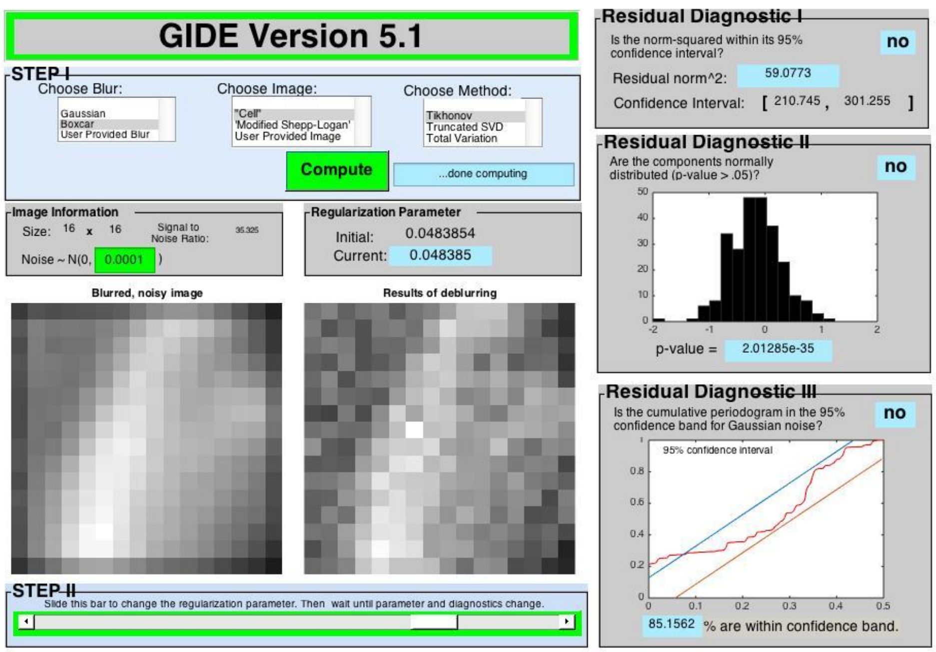

Figure 3: Screenshot of the GIDE GUI. Choices made at the top left result in the images displayed below and in the diagnostics on the right.

3.1 Continuous Representation

In the continuous case, the blurring model can be represented by

$$Ku + \nu = w, \tag{1}$$

where K is the known blurring operator, u is the true image, w is the observed image, and ν is noise that we assume to be generated by a Gaussian white noise process. Equation (1) is an example of an ill-posed problem since the operator K is an infinite-dimensional operator whose singular values converge to zero. One way to impose stability is to solve (1) in a least-squares sense while adding a regularization or penalty term $R(u)$ that incorporates *a priori* assumptions about the size and smoothness of the desired solution. In this case, our problem becomes

$$\min_u \frac{1}{2}\|Ku - z\|_2^2 + \gamma R(u), \tag{2}$$

where γ is a nonnegative parameter. The choice of space over which the minimization is performed can also provide regularization.

3.2 Discrete Representation

In Section 3.1, we modeled 2D deblurring problems as determination of a real function of two variables over the spatial domain of the image. Alternatively, we can model it by determination of a piecewise constant function, evaluated at $n_v \times n_h$ pixels, where v denotes vertical and h denotes horizontal. Let $n = n_v n_h$ be the number of pixels in the image we wish to reconstruct. We assume that the number of pixels in the $m_h \times m_v$ blurred image is $m = m_h m_v$ and that $m \geq n$. For definiteness, we form a vector of pixels by stacking columns of an image into a single column vector. We assume zero boundary conditions for pixels outside the border of our image.

Then a discrete linear model of blur takes the form

$$\boldsymbol{Ax} + \boldsymbol{\epsilon} = \boldsymbol{b}, \tag{3}$$

where $\boldsymbol{A}$ is a (known) $m \times n$ blurring matrix, $\boldsymbol{x}$ is an (unknown) $n \times 1$ vector containing pixel values of the true image, $\boldsymbol{\epsilon}$ is an (unknown) $m \times 1$ vector of noise that we assume to be drawn from a normal distribution with zero mean, and $\boldsymbol{b}$ is the (known) $m \times 1$ blurred and noisy image data. Equation (3) is called a discrete ill-posed problem because $\boldsymbol{A}$ is an ill-conditioned matrix approximating the infinite-dimensional blurring operator.

To regularize, we replace (3) by

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{Ax} - \boldsymbol{b}\|_2^2 + \gamma Q(\boldsymbol{x}). \tag{4}$$

The first term ensures fidelity to the model (3), while the function Q is chosen to assure that the minimization problem is well-posed. The scalar parameter γ is chosen to balance these two objectives.

3.3 Constructing the Blurring Matrix from the Point Spread Function

Consider an image of a single white pixel surrounded by black pixels. The image resulting from blurring this image is called the *point spread function (PSF)* for that pixel. If the blur is identical at all pixels (i.e., spatially invariant), then the blurring matrix can be constructed from a single PSF; see, for example, [5]. In general, we can form a column of the blurring matrix by imaging a single white pixel and then stacking the columns of the resulting image into a single column.

For more details regarding finding the PSF and constructing the blurring matrix see the textbook [5].

We discuss two example PSFs, the boxcar and the Gaussian, in Section 7.2.

4 Discrete Regularization Methods

GIDE gives the user a choice of three different regularization methods. Two of the methods (Tikhonov and TSVD) can be easily applied once the Singular Value Decomposition (SVD) of $\boldsymbol{A}$ is computed. They were chosen because of their efffectiveness, popularity and ease of implementation. The third method, TV, was chosen because it favors solutions that include steep gradients (edges), typical of real images [10].

4.1 SVD-Based Regularization Methods

Define the SVD of $\boldsymbol{A}$ to be

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \sum_{i=1}^{n} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T \tag{5}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n]$ is an $m \times n$ matrix and $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n]$ is an $n \times n$ matrix, each with orthonormal columns, and the diagonal matrix $\boldsymbol{\Sigma}$ has entries $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$ [4].

The Tikhonov regularization function is

$$Q_{\text{tik}}(\boldsymbol{x}) = \|\boldsymbol{L}\boldsymbol{x}\|_2^2,$$

where $\boldsymbol{L}$ is the identity matrix, an approximation of the first derivative operator, a diagonal weighting matrix [3, p. 12], or a problem-specific operator. We choose $\boldsymbol{L}$ to be the identity matrix (otherwise the generalized SVD should be used), and the solution to problem (4) is then given [4] by

$$\boldsymbol{x}_\gamma = \sum_{i=1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \gamma} \frac{\boldsymbol{u}_i^T \boldsymbol{b}}{\sigma_i} \boldsymbol{v}_i. \tag{6}$$

Alternatively, in TSVD we regularize the problem by truncating $\boldsymbol{A}$. Effectively, Q_{tsvd} puts an infinite penalty on using components i of the SVD for which σ_i is too small. The TSVD solution is given by [4]

$$\boldsymbol{x}_\gamma = \sum_{i=1}^{n} \phi_i \frac{\boldsymbol{u}_i^T \boldsymbol{b}}{\sigma_i} \boldsymbol{v}_i \tag{7}$$

where $\phi_i = 1$ for $i \leq \gamma$ and $\phi_i = 0$ otherwise. In this case the regularization parameter γ is an integer chosen from the interval $[1, n]$.

Notice that for both Tikhonov regularization and TSVD, the coefficient $\frac{\boldsymbol{u}_i^T \boldsymbol{b}}{\sigma_i}$ is multiplied by a number close to one when σ_i is large and a number close to zero when σ_i is very small.

4.2 TV Regularization for the Continuous Problem

Before discussing the implementation of discrete TV regularization, we explain the concepts for the continuous problem.

In TV regularization, our regularization function is the l_1 norm of the gradient (∇) of the solution and thereby retains sharp edges in the image that may be obscured if, for example, the l_2 norm is used, as in Tikhonov regularization. Using a coordinate system $(s, t) \in \Omega$ for the domain of the image, the regularization function in (2) is taken to be

$$R_{TV}(x) = \int_\Omega |\nabla x| \, d\Omega, \tag{8}$$

where

$$|\nabla x| = \sqrt{x_s^2 + x_t^2}, \tag{9}$$

x_s denotes the partial derivative of x in the s direction, and x_t denotes the partial derivative in the t direction.

The continuous problem becomes the minimization of

$$f(u) = \frac{1}{2}\|Ku - z\|_2^2 + \gamma \int_\Omega |\nabla u| d\Omega. \tag{10}$$

The first-order condition of optimality (also known as the Euler-Lagrange equation) for the problem with homogeneous Neumann boundary conditions is

$$0 = K^*(Ku - z) - \gamma \nabla \cdot (\frac{\nabla u}{|\nabla u|}), \tag{11}$$

where K^* is the adjoint operator of K in the l_2 inner product space. This regularization problem is non-linear and the TV term is not differentiable at zero. Often this difficulty is avoided by adding a small positive constant $\beta > 0$ so that (11) becomes

$$g(u) = K^*(Ku - z) - \gamma \nabla \cdot (\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}}) = 0. \tag{12}$$

Unlike the Tikhonov and TSVD problems, which have the closed-form solutions (6) and (7), the TV problem must be solved iteratively by methods such as time marching schemes, Newton's method, lagged diffusivity fixed point iteration, or primal-dual Newton method. We used a primal-dual Newton method as presented in [1].

4.2.1 Formulation of Newton's Method

The first step of this method is to find a quadratic model that matches function value and the first two derivatives of f at the current iterate u, where $f(u)$ is given in (10). Our search direction δu points from the current iterate u to the minimizer of the quadratic function and satisfies the equation

$$H(u)\delta u = -g(u), \tag{13}$$

where

$$H(u) = K^*K - \gamma\nabla\cdot\left(\frac{1}{\sqrt{|\nabla u|^2+\beta}}(I - \frac{\nabla u\nabla u^T}{|\nabla u|^2+\beta})\nabla\right). \tag{14}$$

4.2.2 Linearization Based on Introducing a New Variable

In [1] the authors suggest improving the method by introducing a new variable

$$w = \frac{\nabla u}{\sqrt{|\nabla u|^2+\beta}}. \tag{15}$$

This technique is motivated by primal-dual optimization and gives better global convergence behavior than Newton's method [1]. From (12) we find the equivalent system of equations:

$$K^*(Ku - w) - \gamma\nabla\cdot w = 0, \tag{16}$$

$$w\sqrt{|\nabla u|^2+\beta} - \nabla u = 0. \tag{17}$$

We now linearize this system to find:

$$\begin{bmatrix} \sqrt{|\nabla u|^2+\beta} & -(I - \frac{w\nabla u^T}{\sqrt{|\nabla u|^2+\beta}})\nabla \\ -\gamma\nabla\cdot & K^*K \end{bmatrix}\begin{bmatrix} \delta w \\ \delta u \end{bmatrix} = -\begin{bmatrix} f(w,u) \\ g(w,u) \end{bmatrix}. \tag{18}$$

This method for computing search directions for w and u is called the primal-dual Newton's method [1].

4.3 Discretization of the TV Regularization Method

The method of Section 4.2 has a discrete analogue. After discretization, the regularization function becomes

$$Q_{\mathrm{tv}}(\boldsymbol{x}) = \sum_{i=1}^{n}\sqrt{\|\boldsymbol{D}_i^T\boldsymbol{x}\|_2^2+\beta}. \tag{19}$$

Assuming $\boldsymbol{x}$ is stacked by columns, $\boldsymbol{D}_i^T\boldsymbol{x} = [x_{i+n_v} - x_i, x_{i+1} - x_i]^T$ is the discretization of ∇x (with zero used when a subscript is out of range) [1]. A small $\beta > 0$ is added because the TV term is not differentiable at zero. The resulting discrete TV problem is

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \gamma \sum_{i=1}^{n} \sqrt{\|\boldsymbol{D}_i^T\boldsymbol{x}\|_2^2 + \beta}. \tag{20}$$

4.3.1 Discrete Formulation of Newton's Method

The discretization of (12) and (14) yields

$$\mathbf{g}(\mathbf{x}) = \mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) + \gamma\mathbf{D}\mathbf{E}^{-1}\mathbf{D}^T\mathbf{x}, \tag{21}$$

where

$$\nu_i = \sqrt{\|\mathbf{D}_i^T\mathbf{x}\|^2 + \beta}, \tag{22}$$

$$\mathbf{E} = \mathrm{diag}(\nu_i\mathbf{I}_2)_{i=1,..,m}, \tag{23}$$

$\mathbf{I}_2$ is a 2×2 identity matrix, and

$$\mathbf{H}(\mathbf{x}) = \mathbf{A}^T\mathbf{A} + \gamma\mathbf{D}\mathbf{E}^{-1}\mathbf{F}\mathbf{D}^T, \tag{24}$$

where

$$\mathbf{F} = \mathrm{diag}(\mathbf{I}_2 - \frac{\mathbf{D}_i^T\mathbf{x}\mathbf{x}^T\mathbf{D}_i}{\nu_i^2})_{i=1,..,m}. \tag{25}$$

4.3.2 Formulation of Primal-Dual Newton's Method

Analogous to the discretization in Section 4.3.1, we introduce the $2m \times 1$ vector $\mathbf{y}$ of dual variables. The discretization of the Primal-Dual Newton method becomes

$$\begin{bmatrix} \mathbf{E} & -\bar{\mathbf{F}}\mathbf{D}^T \\ \gamma\mathbf{D} & \mathbf{A}^T\mathbf{A} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta}\mathbf{y} \\ \boldsymbol{\Delta}\mathbf{x} \end{bmatrix} = -\begin{bmatrix} \mathbf{E}\mathbf{y} - \mathbf{D}^T\mathbf{x} \\ \gamma\mathbf{D}\mathbf{y} + \mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \end{bmatrix}, \tag{26}$$

where $\bar{\mathbf{F}} = \mathrm{diag}(\mathbf{I}_2 - \frac{\mathbf{y}_i^T\mathbf{x}^T\mathbf{D}_i}{\nu_i})_{i=1,...,m}$. This can be written as:

$$\begin{aligned} \mathbf{C}\boldsymbol{\Delta}\mathbf{x} &= \bar{\mathbf{g}}(\mathbf{x}), \\ \boldsymbol{\Delta}\mathbf{y} &= -\mathbf{y} + \mathbf{E}^{-1}\mathbf{D}^T\mathbf{x} + \mathbf{E}^{-1}\bar{\mathbf{F}}\mathbf{D}^T\boldsymbol{\Delta}\mathbf{x}, \end{aligned}$$

where

$$\mathbf{C} = \gamma\mathbf{D}\mathbf{E}^{-1}\bar{\mathbf{F}}\mathbf{D}^T + \mathbf{A}^T\mathbf{A} \tag{27}$$

and

$$\bar{\mathbf{g}}(\mathbf{x}) = -(\gamma\mathbf{D}\mathbf{E}^{-1}\mathbf{D}^T\mathbf{x} + \mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b})). \tag{28}$$

Note that $\mathbf{C}$ is not symmetric and as suggested by the authors of [1] should be replaced by symmetrization $\bar{\mathbf{C}} = \frac{1}{2}(\mathbf{C} + \mathbf{C}^T)$ of $\mathbf{C}$.

In summary, we solve the TV regularization problem using the Primal-Dual Newton Method (Algorithm 1) with conjugate gradients (Algorithm 2), as described in [1]. For the line search, we used *cvsrch.m*, translated into MATLAB from code developed by Jorge J. Moré and David J. Thuente as part of MINPACK.

Figure 4 shows the results of applying the TV method on two test images.

Algorithm 1 The Primal-Dual Newton Method

Initialize $\mathbf{x}_0$ and set $k = 0$.
while $\mathbf{x}_k$ is not "good enough" **do**
  To compute the Newton direction $\mathbf{\Delta x}_k$, use CG to solve $\bar{\mathbf{C}}\mathbf{\Delta x}_k = -\bar{\mathbf{g}}(\boldsymbol{x}_k)$.
  Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{\Delta x}_k$ where α_k is determined by a linesearch.
  Compute $\mathbf{\Delta y}_k = -\mathbf{y}_k + \mathbf{E}^{-1}\mathbf{D}^T\mathbf{x}_k + \mathbf{E}^{-1}\bar{\mathbf{F}}\mathbf{D}^T\mathbf{\Delta x}_k$.
  Set $\mathbf{y}_{k+1} = \mathbf{y}_k + s_k \mathbf{\Delta y}_k$ where $s_k = .9 \max_{i=1,\ldots,m}\{s : \|\mathbf{y}_i + s\mathbf{\Delta y}_i\| < 1\}$.
  Set $k = k + 1$.
end while

"good enough": $\|\mathbf{g}(\mathbf{x}_k)\|/\|\mathbf{g}(\mathbf{x}_0)\| < 10^{-3}$.

Algorithm 2 The Conjugate Gradient Method for solving $\bar{\mathbf{C}}(\mathbf{x})\mathbf{\Delta x} = -\bar{\mathbf{g}}$

Initialize $\mathbf{r} = -\bar{\mathbf{g}}(\mathbf{x}_k) - \bar{\mathbf{C}}(\boldsymbol{x}_k)\mathbf{\Delta x}$, $\mathbf{q} = \mathbf{r}$, $\rho = \|\mathbf{r}\|$, and $\gamma = \rho^2$.
while $\|\mathbf{r}\|/\rho \geq tol$ **do**
  Compute $\alpha = \frac{\gamma}{\mathbf{q}^\mathbf{T}\bar{\mathbf{C}}(\mathbf{x})\mathbf{q}}$.
  Set $\mathbf{\Delta x} = \mathbf{\Delta x} + \alpha\mathbf{q}$.
  Set $\mathbf{r} = \mathbf{r} - \alpha\bar{\mathbf{C}}(\mathbf{x})\mathbf{q}$.
  Compute $\hat{\gamma} = \|\mathbf{r}\|^2$.
  Compute $\beta = \frac{\hat{\gamma}}{\gamma}$, $\gamma = \hat{\gamma}$.
  Set $\mathbf{q} = \mathbf{r} + \beta\mathbf{q}$.
end while

Suggested tolerance [1]: $tol = \min(0.1, 0.9\|\mathbf{g}(\mathbf{x}_k)\|^2/\|\mathbf{g}(\mathbf{x}_{k-1})\|^2)$ for $k > 0$, and $tol = 0.1$ when $k = 0$.

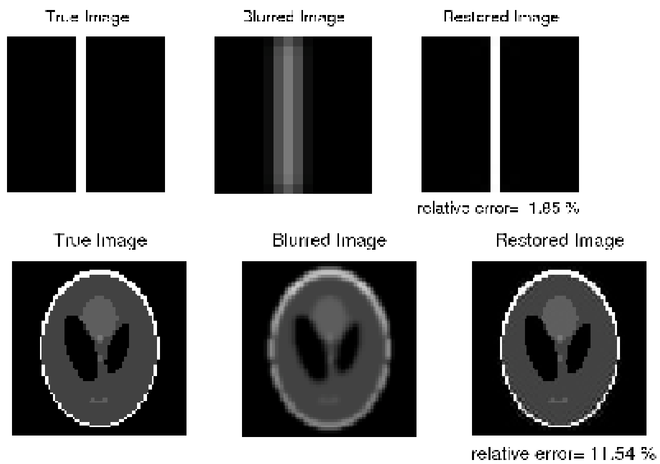

Figure 4: Results of Newton-CG for Total Variation regularization for two images, a 16×16 striped image (top) and a 64×64 Modified Shepp-Logan image(bottom).

5 Initial Parameter Selection

A number of automated parameter selection methods have been developed, some based on prior knowledge of the particular problem (distribution of noise or errors), others based on statistical criteria. The parameters chosen by these methods are often far from those that minimize the deviation of the computed solution from the true solution [3]. In this work, automated parameter selection methods are used only to find an initial parameter, to provide a starting point for the user. We choose to use generalized cross validation (GCV) for the SVD-based methods, since the computation can be performed efficiently. For TV, GCV is too costly, so we use the discrepancy principle.

5.1 Generalized Cross-Validation (GCV)

GCV is based on the popular leave-one-measurement-out model, checking the reasonableness of a parameter determined from $m-1$ measurements by seeing how well the resulting model predicts the mth measurement [4, p. 95]. The idea is to choose the parameter γ that minimizes the prediction errors. In GCV we formulate this as minimizing

$$G(\gamma) = \sum_{k=1}^{m} [b_k - (\boldsymbol{A}\boldsymbol{x}_\gamma^{(k)})_k]^2, \tag{29}$$

where $\boldsymbol{x}_\gamma^{(k)}$ is the estimate that results from using the regularization parameter γ but omitting the k^{th} measurement b_k. For SVD-based methods, the GCV expression can be greatly simplified [5]. For Tikhonov regularization, G becomes

$$G_{\text{tik}}(\gamma) = \frac{\sum_{i=1}^{m} (\frac{\boldsymbol{u}_i^T \boldsymbol{b}}{\sigma_i^2+\gamma^2})^2}{\sum_{i=1}^{m} (\frac{1}{\sigma_i^2+\gamma^2})^2} \tag{30}$$

while for TSVD,

$$G_{\text{tsvd}}(\gamma) = \frac{1}{(m-\gamma)^2} \sum_{i=\gamma+1}^{m} (\boldsymbol{u}_i^T \boldsymbol{b})^2. \tag{31}$$

5.2 Discrepancy Principle

If we know the distribution of the noise $\boldsymbol{\epsilon}$ then we can choose γ so that

$$\|\boldsymbol{A}\boldsymbol{x}_\gamma - \boldsymbol{b}\|_2 = \nu \mathcal{E}(\|\boldsymbol{\epsilon}\|_2), \tag{32}$$

where $\mathcal{E}$ denotes expected value and $\nu = 2$ is a safety factor [4, p. 90]. The appropriate value of γ is computed by solving (32) using MATLAB's `fzero`, an efficient root finding algorithm.

6 Statistical Diagnostics

We use statistical diagnostics to test the plausibility of a candidate regularization solution as a solution to the original ill-posed problem. We use the three diagnostics from [9] to generate a range of plausible regularization parameters.

These diagnostics, proposed by Bert Rust [8], are based on the simple observation that since

$$\boldsymbol{\epsilon} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x} \tag{33}$$

is noise drawn from some statistical distribution, then

$$\boldsymbol{r}_\gamma = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_\gamma \tag{34}$$

should ideally equal $\boldsymbol{\epsilon}$ and therefore be a sample from the same distribution. We use standard statistical tests to evaluate how typical $\boldsymbol{r}_\gamma$ is as a sample from the distribution, which we assume to be normal with known variance.

6.1 Choice of Diagnostics

To use the diagnostics, we normalize our problem so that $\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \boldsymbol{I}_m)$. If the error is distributed as $N(\boldsymbol{0}, \boldsymbol{S}^2)$, this can be done by multiplying the blurring matrix $\boldsymbol{A}$ and the observed image $\boldsymbol{b}$ by $\boldsymbol{S}^{-1}$.

We now discuss the three diagnostics shown on the right side of the GUI in Figure 3.

6.1.1 Residual Diagnostic 1

Since $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \boldsymbol{I}_m)$, we know the distributionof $\|\boldsymbol{\epsilon}\|_2^2$. The sum of squares of a set of m independent identically distributed (i.i.d.) standard normal ($\boldsymbol{\epsilon} \sim N(\mathbf{0}, \boldsymbol{I}_m)$) random samples is a random variable with a χ^2 distribution. It has expected value m and variance $2m$ [7]. Therefore, our first diagnostic tests whether the residual norm squared, $\|\boldsymbol{r}_\gamma\|_2^2$, is within two standard deviations (i.e., within the 95% confidence interval) of the expected value of $\|\boldsymbol{\epsilon}\|_2^2$. Therefore, we want

$$\|\boldsymbol{r}_\gamma\|_2^2 \in [m - 2\sqrt{2m}, m + 2\sqrt{2m}]. \tag{35}$$

GIDE displays the residual norm-squared, the endpoints of the confidence interval and a yes/no answer to whether we are within the confidence interval.

6.1.2 Residual Diagnostic 2

The histogram of the elements of $\boldsymbol{r}_\gamma$ should look like a bell-shaped curve. We use a χ^2 goodness-of-fit test [7] which tests whether the residual is drawn from an i.i.d standard normal distribution (null hypothesis) by comparing it to the theoretical distribution. GIDE displays the histogram of the residual and a yes/no answer to whether the *p-value* displayed satisfies $p > 0.05$. If yes, then one should accept the null hypothesis with 95% confidence.

6.1.3 Residual Diagnostic 3

If we view the elements of $\boldsymbol{\epsilon}$ and $\boldsymbol{r}$ as time series with index $i = 1, \ldots, m$ then $\{\epsilon_i\}$ forms a white noise series. We expect $\{r_i\}$ to also be a white noise series. One way to assess this is to compute the *cumulative periodogram* of the residual [2, Chapter 7].

First we compute the discrete Fourier transform of $\boldsymbol{r}$ to form a vector $\hat{\boldsymbol{r}}$. The periodogram is defined as the vector of squared absolute values $|\hat{\boldsymbol{r}_k}^2|$. The cumulative periodogram (with components numbered from 0 to $p = \lceil (m-1)/2 \rceil$) is given by

$$c_k = \frac{\sum_{j=0}^{k} |\hat{\boldsymbol{r}_k}^2|}{\sum_{j=0}^{p} |\hat{\boldsymbol{r}_k}^2|} \quad j = 0, \ldots, p \tag{36}$$

If we look at the periodogram of white noise, the expected value of $|\hat{\boldsymbol{r}_k}^2|$ at each frequency k is the same, so a plot of the expected values of c_k vs. k is a straight line from $(0, 0)$ to $(p, 1)$. The 95% confidence interval for the plot of the cumulative periodogram of white noise is displayed by GIDE, along with the plot of the computed $\mathbf{c}$ for the residual. The endpoints of the confidence interval are at approximately at plus or minus $1.36/\sqrt{(p-1)}$ relative to the straight line whenever $p > 31$ [2].

6.2 Validation of Residual Diagnostics

The three diagnostics all test the hypothesis that the residual components are drawn from a normal distribution, but the diagnostics are sensitive to somewhat

Table 1: For 1000 runs, number of times the Diagnostics fail to be satisfied. $I(i) = 1$ if $(i-1) \mod (100) = 0$ or $(i-2) \mod (100) = 0$ and $I(i) = 0$ otherwise.

Residual	Diag. 1	Diag. 2	Diag. 3
$r_i \sim N(0,1)$	51	46	14
$r_i + I(i)$	950	999	539
$r_i + .05 * s_p \ s_p \sim pois(1)$	156	1000	149

different perturbations, as we illustrate here.

Each of the diagnostics was applied first to sequences of 1000 standard normal i.i.d. samples to confirm that it was satisfied approximately 95% of the time; see Table 6.2. From the first line in the table, for 1000 different sequences, this is true.

Then we applied the diagnostics to sequences of standard normal i.i.d. samples, but with one added to every 100th sample. The second line of the table shows that Diagnostics 1 and 2 accurately detect that the result is not normal, but Diagnostic 3 is only correct in 54% of the trials.

Finally, we add a small amount of Poisson-distributed noise to sequences of standard normal i.i.d. samples. The third line of the table shows that only Diagnostic 2 reports non-normality reliably.

Diagnostic 2 is reliable in all three of these simple tests, but the others provide useful information as well. Diagnostic 1 gives the user a range of residual norms that satisfy that diagnostic; this helps the user easily find a range of feasible values for γ. Diagnostic 3 gives a useful visualization of how similar the residual is to the Gaussian distribution, which is sometimes difficult to see from Diagnostic 2.

Note that for a given noise sample, there may be no parameter that satisfies all three diagnostics. The diagnostics should just be considered as indicators that the solution computed is more or less reasonable, based upon the hypothesis of the distribution of errors in the data.

7 Using the Software

GIDE was built using MATLAB's GUI toolbox. It enables a user with limited or no experience with the methods discussed above to find suitable solutions to a problem. As Figure 3 shows, a user can select the type of blur, an example test problem, and the regularization method. Clicking COMPUTE finds an initial solution based on the approximate parameter (found using GCV or the discrepancy principle). The user can then use the slide bar below the images to vary the regularization parameter in order to explore the results and look for solutions that satisfy each of the three diagnostics on the right.

7.1 Installation

GIDE can be downloaded from `http://www.cs.umd.edu/users/oleary/software`. GIDE Version 5.1 has been tested under MATLAB version R2014b and RESTORETOOLS version April 2012.

You need to also download RESTORETOOLS from `http://www.mathcs.emory.edu/~nagy/RestoreTools/` and follow the installation instructions.

Then edit the GIDE file `startGIDE.m` to set `path_to_RestoreTools` and `path_to_GIDE` to the complete directory names where you have stored these two packages.

Typing `startGIDE` into MATLAB should bring up the GUI.

7.2 Choose Blur

Either of the preloaded blurs (Gaussian or boxcar) can be used with either of the preloaded images to generate a test problem. Alternatively, upload a blurring matrix for your own test problem.

The blurring matrix titled "Boxcar" models a boxcar blur of band size three where the nonzero components of the PSF are given by

$$\frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

This means that each given true pixel value has been averaged with its eight neighboring pixels to obtain a blurred pixel value.

The blurring matrix titled "Gaussian" models a Gaussian blur of band size three and $\sigma = .7$ where the nonzero components are

$$\frac{1}{2\pi\sigma^2}\begin{pmatrix} \exp(-\frac{(-1)^2+1^2}{2\sigma^2}) & \exp(-\frac{0^2+1^2}{2\sigma^2}) & \exp(-\frac{1^2+1^2}{2\sigma^2}) \\ \exp(-\frac{(-1)^2+0^2}{2\sigma^2}) & \exp(-\frac{0^2+0^2}{2\sigma^2}) & \exp(-\frac{1^2+0^2}{2\sigma^2}) \\ \exp(-\frac{(-1)^2+(-1)^2}{2\sigma^2}) & \exp(-\frac{0^2+(-1)^2}{2\sigma^2}) & \exp(-\frac{(-1)^2+(-1)^2}{2\sigma^2}) \end{pmatrix}.$$

Similarly, this means that the blurred pixel values are weighted averages of true pixel values in a 3×3 neighborhood.

The SVD-based methods, the matrix structures, and the implementations of GCV and the discrepancy principle were taken from RESTORETOOLS [6]. We implemented the primal-dual TV algorithm and the statistical diagnostics.

7.3 Choose Image

The `cell.tif` and a 16×16 version of MATLAB's Shepp-Logan are available for testing. You can also specify your own "Test Image".

7.4 Choose Method

You have your choice of Tikhonov, truncated SVD, or TV regularization.

7.5 Compute

After choosing the blur, the image, and the method, the COMPUTE button will produce an initial deblurred image using an automatic choice of the regularization parameter γ. The box next to COMPUTE will indicate when the computation is complete. The three diagnostics will be displayed. The parameter γ can be adjusted using the slide bar that appears under the original and deblurred images. Again, the box next to COMPUTE will indicate when the computation is complete.

Adjusting the parameter so that the diagnostics move into their "yes" ranges produces a reconstruction that is statistically plausible. There is no guarantee, however, that a parameter exists that satisfies all three diagnostics.

Sliding to the right increases the residual-norm-squared (Diagnostic 1) and tends to move the red line (the cumulative periodogram) in Diagnostic 3 to the left.

7.6 Limitations

The speed of today's computers limits the size of images for which real-time response is reasonable in the GUI.

GIDE is meant to be a tool for exploration. If you have a large image that you need to deblur, we suggest extracting a small piece of it. Using GIDE, you can determine find an appropriate regularization method and a statistically-validated candidate parameter that can then be used for the full image. Using this regularization parameter, the computation for the full image can be done using RESTORETOOLS or the TV program `TVPrimDual.m`.

GIDE is a working proof-of-concept that could be scaled to a faster computational tool by using a compiled computer language and high-performance computing.

8 Results and Testing

In this section we discuss some testing we performed on different components of GIDE on a variety of test images.

8.1 Test Images

We used artificially generated images and PSF functions for development and initial testing of the software. These data sets were created to be of any size. For testing and validation we used the images found in *RestoreTools* as well as a variety of PSFs also found in *RestoreTools*.

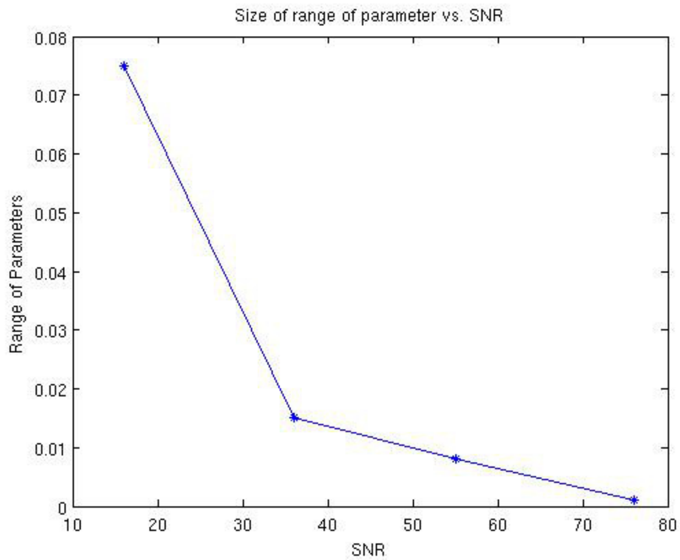

Figure 5: For Tikhonov regularization for deblurring the 16×16 segment of the image `cell.tif` with SNR $= 75$, parameters in a range (± 0.0025) were found to satisfy Diagnostic 1. As the SNR decreases, the range of parameters satisfying the diagnostic increases.

8.2 Signal-to-Noise Ratio Effect on Diagnostics

As one may expect, the diagnostics are affected by the Signal-to-Noise Ratio (SNR) which is defined as

$$SNR = 10 \log_{10}\left(\frac{\|\mathbf{b}\|^2}{\|\epsilon\|^2}\right). \tag{37}$$

The range of plausible parameters that meet the diagnostics increases as the SNR approaches to zero and as the SNR increases the range of plausible solutions becomes smaller (Figure 5). Note that users should be careful when they have a very small SNR as the diagnostics used may not be the best measure for plausible solutions. This test was performed on a 16×16 piece of Matlab test image `cell.tif` where the range of SNR was varied from 20 to 75. Ranges plotted are for Diagnostic 1 and the Tikhonov method, although a similar relationship was found for the different methods, diagnostics, and different test images included in the GUI.

8.3 Effects of γ on Computation Time

We found that the computation time of the TV regularization method (dependent on the number of CG iterations) is dependent on the value of γ. See Figure

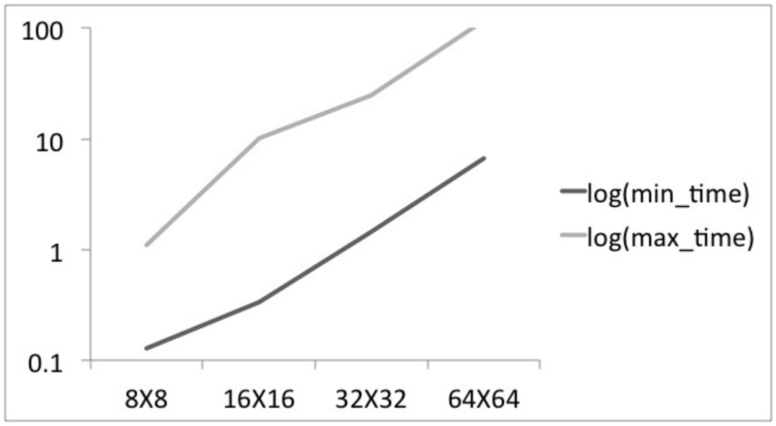

Figure 6: The range in log_{10} of computational time (seconds) for the TV regularization method for parameters between $\gamma = 1$ and $\gamma = 10^{-9}$. For all image sizes the maximum time occurred for $\gamma = 1$.

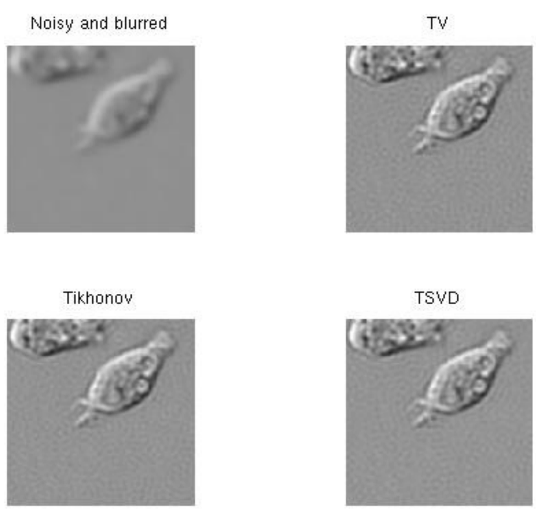

Figure 7: 129×129 image of `cell.tif` with Gaussian blur and zero boundary conditions with SNR $= 60$.

6 for results for different values of γ.

As a result of this finding, we looked into using preconditioners to speed up the algorithm for large γ. An incomplete LU factorization (ILU) is suggested as a preconditioner for the Primal-Dual Newton method [1]. We did not find the use of ILU to be robust enough to be included in the GUI as we had to adjust the drop tolerance depending on γ.

8.4 Results on Larger Images and Varied PSF

Although the GUI is only able to handle very small images, additional tests of the Primal Dual TV method were done on larger images as well as images with a variety of PSFs. See Figure 2 and Figure 7 comparing the results using Tikhonov and TSVD to the TV method.

9 Summary and Conclusions

When decisions are based on images, it is important to use a regularization method and parameter that can be justified on statistical grounds. GIDE helps practitioners do this. The software takes advantage of the practitioner's trained eyes while limiting bias by using statistical diagnostics. Even without detailed knowledge of the numerical method, the user can explore different solutions with real-time diagnostics determining whether the solution is statistically plausible.

There has been work in automatic parameter selection but these methods have shown not to be reliable over a variety of problems. Given this challenge and the lack of automatic approaches for choosing a regularization methods, our methodology is a straightforward approach for finding an appropriate method and parameter.

To effectively be used in real time, our methodology is currently limited to relatively small images. That being said the software has been proved useful in an undergraduate course on image restoration, giving the students immediate feedback about the effects of different regularization methods and parameter choices.

Acknowledgements

This work, including the development of GIDE was supported by the National Science Foundation under grant DMS 1016266.

References

[1] Tony F. Chan, Gene H. Golub, and Pep Mulet. A nonlinear primal-dual method for total variation-based image restoration. *Lecture Notes in Control and Information Sciences*, 219:241–251, 1996.

[2] Wayne A. Fuller. *Introduction to Statistical Time Series.* Wiley-Interscience, New York, N.Y., 1996.

[3] Per Christian Hansen. *Rank-Deficient and Discrete Ill-Posed Problems.* Numerical Aspects of Linear Inversion. SIAM, Philadelphia, PA, 1998.

[4] Per Christian Hansen. *Discrete Inverse Problems: Insight and Algorithms.* SIAM, Philadelphia, PA, 2010.

[5] Per Christian Hansen, James G. Nagy, and Dianne P. O'Leary. *Deblurring Images: Matrices, Spectra, and Filtering.* SIAM, Philadelphia, PA, 2006.

[6] James G. Nagy. RestoreTools: An object oriented Matlab package for image restoration, 2004. `http://www.mathcs.emory.edu/~nagy/RestoreTools/`.

[7] Sheldon Ross. *A First Course in Probability, Sixth Edition.* Prentice Hall, Upper Saddle River, N.J., 2002.

[8] Bert W. Rust. Parameter selection for constrained solutions to ill-posed problems. *Computing Science and Statistics*, 32:333–347, 2000.

[9] Bert W. Rust and Dianne P. O'Leary. Residual periodograms for choosing regularization parameters for ill-posed problems. *Inverse Problems*, 24(034005):30, 2008.

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