Automated Parameter Selection Tool for Solution to Ill-Posed Problems
Final Report

Brianna Cash (brcash@math.umd.edu)

Advisor: Dianne O'Leary (oleary@cs.umd.edu)

May 13, 2012

Abstract

In many applications of solving ill-posed problems there are a number of regularization methods to choose from as well as a free regularization parameter. The choice of the method and regularization can add bias if deblurred images are based on what the researcher expects. The presented project develops a tool for method choice and parameter selection by using a set of three statistical diagnostics to validate solutions. We include three regularization methods; Tikhonov, Truncated SVD, and Total Variation. The diagnostics are motivated by the idea that the residual after deblurring should be normally distributed if we assume that the added noise is normally distributed. The project develops a interface where users can freely choose a method, then find a range of regularization parameters that produce plausible solutions based on the diagnostics.

1 Background

In medical images such as MRI or CT scans (Figure 1), the images may be distorted and/or noisy due to the physics of the measurement and the structure of the material (human) being imaged. These images are expensive to produce and often are critical in making medical decisions. Image deblurring is an example of an ill-posed inverse problem. To find suitable approximate solutions to ill-posed inverse problems we use our knowledge about the particular problem to come up with constraints [4]. These constraints are used to determine the method and parameters to regularize the problem, replacing the ill-posed problem by one that is well-posed, and thus has an acceptable solution. Finding and selecting good regularization methods and parameters can be very expensive and subject to bias. Researchers often have invaluable information that is crucial in finding a good approximate solution, but without validation, there is risk of seeing what is expected and not the true solution or image (Figure 2). An effective tool that generates a plausible range of regularization parameters is needed to create a cost effective methodology and to control for bias when determining optimal solutions.

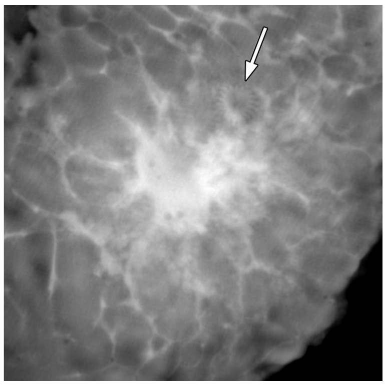


Figure 1: Tomography image of a mastectomy specimen. Stevens G M et al. Radiology 2003;228:569-575

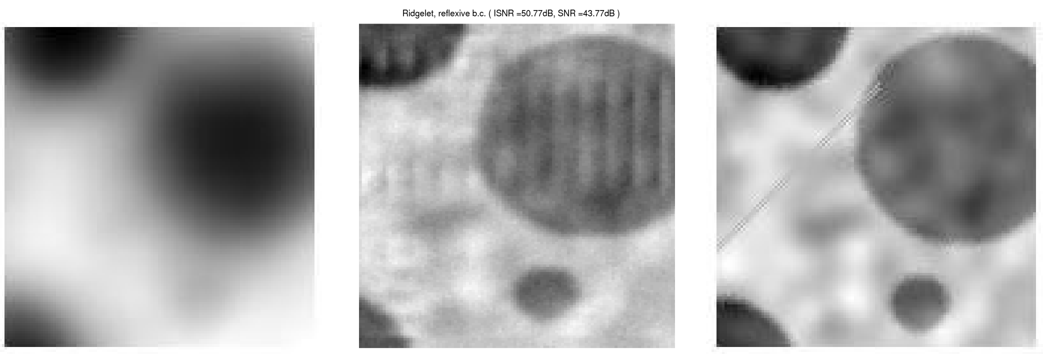


Figure 2: Left: Blurred Image, Center: Deblurred Image, Right: True Image which has “train tracks”. Without knowledge of the true image having “train tracks” one might accept the deblurred to be a good image without realizing that important information was lost in the process. Images courtesy of Dianne O’Leary

The goal of this project is to develop a software package with graphical user interface (GUI) for method and apply it to deblurring and denoising problems with Gaussian noise. The software uses an automated parameter selection tool for initial parameter selection and statistical diagnostics to validate candidate solutions based on user provided parameter. A number of regularization methods are used in hopes of determining optimal methods for a given image: Truncated SVD (TSVD) regularization, Tikhonov regularization, and Total Variation (TV) regularization.

2 Outline

3 Mathematical Model of Regularization Methods

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

3.1 Continuous Representation

In the continuous case, the model can be represented by

$$Ku + \nu = u_0, \tag{1}$$

where K is the known blurring operator, u is the true image, u_0 is the observed image, and ν arises from a Gaussian white noise process. Equation (1) is an example of an ill-posed problem. Stability is imposed by adding a regularization or penalty term $R(u)$ which incorporates *a priori* assumptions about the size and smoothness of the desired solution. Our problem becomes:

$$\min_u \frac{1}{2}\|Ku - u_0\|_2^2 + \gamma R(u), \tag{2}$$

where γ is a nonnegative parameter.

3.2 Discrete Representation

Consider the problem as a collection of discrete square pixels where we can write:

$$\mathbf{A}\mathbf{x} + \epsilon = \mathbf{b}, \tag{3}$$

where $\mathbf{A}$ is a known $m \times n$ matrix with $m \geq n$, $\mathbf{x}$ is an unknown $n \times \mathit{1}$ vector which represents a $n_h \times n_v$ rectangular image where $n = n_h \times n_v$ and v and h signify the vertical and horizontal directions, ϵ is an $m \times \mathit{1}$ vector of error drawn from a normal distribution with known variance and mean, and $\mathbf{b}$ is the $m \times \mathit{1}$ measured image where $m = m_h * m_v$. Because $\mathbf{A}$ is generally an ill-conditioned matrix (a property of discretizing the blurring operator), stability is imposed by adding a regularization term $Q(\mathbf{x})$ with parameter γ. The discrete problem becomes:

$$\min_x \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \gamma Q(\mathbf{x}). \tag{4}$$

4 SVD-Based Regularization Methods

There are two commonly used regularization methods based on the singular value decomposition (SVD) of $\mathbf{A}$, Tikhonov's regularization method and Truncated SVD (TSVD). The SVD of $\mathbf{A}$ is

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathbf{T}} = \sum_{i=1}^{n} \mathbf{u_i}\sigma_i\mathbf{v_i^T} \tag{5}$$

where $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_n)$ is a $m \times n$ matrix and $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_n)$ is a $n \times n$ matrix each with orthonormal columns, and $\mathbf{\Sigma} = diag(\sigma_1, \ldots, \sigma_n)$ is a matrix of the non-negative singular values that appear in decreasing order.

4.1 Tikhonov's Regularization Method

In this method our penalty function is

$$Q(x) = \|\mathbf{L}\mathbf{x}\|_{\mathbf{2}}^{\mathbf{2}}, \tag{6}$$

where $\mathbf{L}$ is the identity matrix, approximation of the first derivative operator, a diagonal weighting matrix [5], or another of operator based on the problem and the desired features.

This method has been implemented using a class in *RestoreTool* where we assume that $\mathbf{L}$ is the indentity matrix. The solution is

$$\mathbf{x}_{tik} = (\mathbf{A}^{\mathbf{T}}\mathbf{A} + \gamma\mathbf{I_n})^{-1}\mathbf{A}^{\mathbf{T}}\mathbf{b}. \tag{7}$$

When one takes the SVD of $\mathbf{A}$ it is easily shown that

$$\mathbf{x}_{tik} = \sum_{i=1}^{n} \frac{\sigma_i \mathbf{u_i} \mathbf{v_i^T} \mathbf{b}}{\sigma_i^2 + \gamma}. \tag{8}$$

4.2 Truncated SVD Method

In Truncated SVD (TSVD) we regularize the problem by truncating $\mathbf{A}$, ignoring the small singular values which cause $\mathbf{A}$ to be ill-conditioned. The regularization problem becomes

$$\min_{\mathbf{x}} \|\mathbf{A_k x} - \mathbf{b}\|_2^2, \tag{9}$$

where

$$\mathbf{A}_k = \sum_{i=1}^{k} \mathbf{u_i} \sigma_i \mathbf{v_i^T} \tag{10}$$

and the regularization parameter is k. Again using the SVD of $\mathbf{A}_k$ it is easily shown that the solution is

$$\mathbf{x}_{TSVD} = \sum_{i=1}^{n} \phi_i \frac{\mathbf{u_i} \mathbf{v_i^T} \mathbf{b}}{\sigma_i} \tag{11}$$

where $\phi_i = 1$ for $i = 1, \ldots, k$ and 0 for $i = k+1, \ldots, n$.

5 Total Variation Regularization Method

In Total Variation (TV) regularization, our penalty function is the l_1 norm of the gradient (∇) of the solution and thereby retains steep gradients that may be lost if the l_2 norm is used. The penalty function is

$$R(u) = TV(u) = \int_\Omega |\nabla u| d\Omega, \tag{12}$$

and

$$|\nabla u| = \sqrt{u_x^2 + u_y^2}, \tag{13}$$

where Ω is the domain (for images we can assume that it is rectangular). The continuous problem becomes the minimization of

$$f(u) = \frac{1}{2}\|Ku - u_0\|_2^2 + \gamma \int_\Omega |\nabla u| d\Omega. \tag{14}$$

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

$$0 = K^*(Ku - u_0) - \gamma \nabla \cdot (\frac{\nabla u}{|\nabla u|}), \tag{15}$$

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 everywhere differentiable. There have been many proposed iterative methods to approximate the solution, including time marching schemes, Newton's method, lagged diffusivity fixed point iteration, and primal-dual Newton method [3]. In many of these methods the difficulty of the $TV(u)$ term not being differentiable at zero is avoided by adding a small positive constant $\beta > 0$ so that (15) becomes

$$g(u) = K^*(Ku - u_0) - \gamma\nabla \cdot (\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}}) = 0. \tag{16}$$

For this project I solved the descretized optimization problem with Primal-Dual Newton Method with Conjugate Gradients (CG) based on the algorithm found in [3]. See section 9.2 for details about implementation.

5.1 Formulation of Newton's Method

To solve our optimization problem we use Newton's method. The first step of this method is to find a quadratic model that fits the function $f(u + \delta u)$ where $f(u)$ is given in (14), and δu is the search direction. The search direction can be found by minimizing the quadratic model over δu. Newton's method becomes:

$$H(u)\delta u = -g(u), \tag{17}$$

where

$$H(u) = (K^*K - \gamma\nabla \cdot (\frac{1}{\sqrt{|\nabla u|^2 + \beta}}(I - \frac{\nabla u \nabla u^T}{|\nabla u|^2 + \beta})\nabla). \tag{18}$$

5.2 Linearization Based on Introducing a New Variable

In [3] the authors suggest an improved method to solving the above problem by introducing a new variable $w = \frac{\nabla u}{|\nabla u|}$ or $w = \frac{\nabla u}{\sqrt{|\nabla u|^2+\beta}}$. This technique is related to primal-dual optimization (details found in section 5.6) and gives better global convergence behavior than Newton's method [3].
From (16) we find the equivalent system of equations:

$$K^*(Ku - u_0) - \gamma\nabla \cdot w = 0, \tag{19}$$

$$w\sqrt{|\nabla u|^2 + \beta} - \nabla u = 0. \tag{20}$$

We now linearize this system to find:

$$\begin{bmatrix} \sqrt{|\nabla u|^2 + \beta} & -(I - \frac{w\nabla u^T}{\sqrt{|\nabla u|^2+\beta}})\nabla \\ -\gamma\nabla\cdot & K^*K \end{bmatrix} \begin{bmatrix} \delta w \\ \delta u \end{bmatrix} = -\begin{bmatrix} f(w,u) \\ g(w,u) \end{bmatrix}. \tag{21}$$

This method is called the primal-dual Newton's method [3].

5.3 Discretization and Computation of TV Regularization Method

The discretization of the TV term becomes

$$Q(\mathbf{x}) = \gamma \sum_{i=1}^{n} \sqrt{\|\mathbf{D}_i^T \mathbf{x}\|_2^2 + \beta}, \tag{22}$$

where ∇u is discretized using forward differencing with respect to the pixels in the horizontal direction and the vertical direction of the image:

$$\mathbf{D}_i^T \mathbf{x} = [x_{i+n_v} - x_i, x_{i+1} - x_i]^T, \tag{23}$$

and $\mathbf{D} = [\mathbf{D}_1, ...\mathbf{D}_m]$ is $n \times 2m$. The resulting problem is:

$$\min_x \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \gamma \sum_{i=1}^{m} \sqrt{\|\mathbf{D}_i^T \mathbf{x}\|^2 + \beta}. \tag{24}$$

5.4 Discrete Formulation of Newton's Method

The discretization of (16) and (18) yields

$$\mathbf{g}(\mathbf{x}) = \mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) + \gamma \mathbf{D}\mathbf{E}^{-1}\mathbf{D}^T\mathbf{x}, \tag{25}$$

where

$$\nu_i = \sqrt{\|\mathbf{D}_i^T \mathbf{x}\|^2 + \beta}, \tag{26}$$

$$\mathbf{E} = diag(\nu_i \mathbf{I}_2)_{i=1,..,m}, \tag{27}$$

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

$$\mathbf{H}(\mathbf{x}) = \mathbf{A}^T\mathbf{A} + \gamma \mathbf{D}\mathbf{E}^{-1}\mathbf{F}\mathbf{D}^T, \tag{28}$$

where

$$\mathbf{F} = diag(\mathbf{I}_2 - \frac{\mathbf{D}_i^T \mathbf{x}\mathbf{x}^T \mathbf{D}_i}{\nu_i^2})_{i=1,..,m}. \tag{29}$$

5.5 Formulation of Primal-Dual Newton's Method

Analogous to the discretization in section (5.4), we introduce the $2m \times 1$ vector $\mathbf{y}$ of dual variables, and the discretization of the Primal-Dual Newton method becomes

$$\begin{bmatrix} \mathbf{E} & -\bar{\mathbf{F}}\mathbf{D}^T \\ \gamma \mathbf{D} & \mathbf{A}^T\mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{\Delta y} \\ \mathbf{\Delta x} \end{bmatrix} = - \begin{bmatrix} \mathbf{E}\mathbf{y} - \mathbf{D}^T\mathbf{x} \\ \gamma \mathbf{D}\mathbf{y} + \mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \end{bmatrix}, \tag{30}$$

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

$$\mathbf{C}\mathbf{\Delta x} = \bar{\mathbf{g}(\mathbf{x})}, \tag{31}$$

$$\mathbf{\Delta y} = -\mathbf{y} + \mathbf{E}^{-1}\mathbf{D}^T\mathbf{x} + \mathbf{E}^{-1}\bar{\mathbf{F}}\mathbf{D}^T\mathbf{\Delta x}, \tag{32}$$

where

$$\mathbf{C} = \gamma \mathbf{D}\mathbf{E}^{-1}\bar{\mathbf{F}}\mathbf{D}^T + \mathbf{A}^T\mathbf{A} \tag{33}$$

and

$$\bar{\mathbf{g}(\mathbf{x})} = -(\gamma \mathbf{D}\mathbf{E}^{-1}\mathbf{D}^T\mathbf{x} + \mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b})). \tag{34}$$

Note that $\mathbf{C}$ is not symmetric and as suggested by the authors of [3] should be replaced by symmetrization $\bar{\mathbf{C}} = \frac{1}{2}(\mathbf{C} + \mathbf{C}^T)$ of $\mathbf{C}$.

5.6 Relationship to the Dual Problem

To understand the relationship of the above formulation to the primal-dual problem we first rewrite our minimization or primal problem (24) as a min-max problem by using the the following fact: $\|\mathbf{a}\| = \max_{\|\mathbf{b}\|\leq 1} \mathbf{a^T b}$ for any vector $\mathbf{a}$, as a result of the Cauchy-Schwartz inequality. Ignoring β we find that $\|\mathbf{D}^T\mathbf{x}\| = \max_{\|\mathbf{y}_i\|\leq 1} \mathbf{x^T D y}$. The resulting primal problem is the minimization with respect to $\mathbf{x}$ of:

$$\mathbf{P}(\mathbf{x}) = \frac{1}{2}\|\mathbf{Ax} - \mathbf{b}\|_2^2 + \gamma \max_{\|\mathbf{y}_i\|\leq 1} \mathbf{x^T D y}, \tag{35}$$

and from the concept of duality we know that the dual problem is simply the max-min problem [11]. Our dual problem becomes the maximization with respect to $\mathbf{y}$ of

$$\mathbf{N}(\mathbf{y}) = \min_{\mathbf{x}} \frac{1}{2}\|\mathbf{Ax} - \mathbf{b}\|_2^2 + \gamma \mathbf{x^T D y}. \tag{36}$$

Because the function $\frac{1}{2}\|\mathbf{Ax} - \mathbf{b}\|_2^2 + \gamma \mathbf{x^T D y}$ is convex in $\mathbf{x}$ and concave (actually linear) and bounded for $\|\mathbf{y}_i\| \leq 1$ we know there is a point that satisfies the saddle-point condition. This implies that there is strong duality so:

$$\min_{\mathbf{x}} \mathbf{P}(\mathbf{x}) = \max_{\|\mathbf{y}_i\|\leq 1} \mathbf{N}(\mathbf{y}). \tag{37}$$

The authors in [1] suggest considering the primal-dual problem, formulated by considering the conditions for which the difference between the primal and the dual objective function (duality gap) is zero, which holds when $\mathbf{x}$ and $\mathbf{y}$ are optimal:

$$\frac{1}{2}\|\mathbf{Ax} - \mathbf{b}\|_2^2 + \gamma \sum_{i=1}^{m} \mathbf{y}_i^T \mathbf{D}_i \mathbf{x} = \frac{1}{2}\|\mathbf{Ax} - \mathbf{b}\|_2^2 + \gamma \sum_{i=1}^{m} \|\mathbf{D}_i^T \mathbf{x}\|, \tag{38}$$

where equality holds when $\mathbf{y}_i^T \mathbf{D}_i^T \mathbf{x} = \|\mathbf{D}_i^T \mathbf{x}\|$ for every i where $\mathbf{D}_i^T \mathbf{x} \neq 0$ and $\|\mathbf{y}_i\| \leq 1$. The primal-dual formulation becomes

$$\mathbf{A}^T\mathbf{Ax} - \mathbf{b} \ + \gamma \mathbf{Dy} = 0, \tag{39}$$

$$\|\mathbf{D}_i^T \mathbf{x}\| \mathbf{y}_i - \mathbf{D}_i^T \mathbf{x} = 0, \tag{40}$$

and

$$\|\mathbf{y}_i\| \leq 1. \tag{41}$$

We know the solution of the above system yields the optimal $\mathbf{x}$ for the primal problem. Again we will add a small constant β in order to make the above well defined for all i. Therefore our primal-dual formulation yields the discretization of (19) - (20).

6 Initial Regularization Parameter Selection Method

Our software generates an initial regularization parameter using generalized cross-validation (GCV) or the discrepancy principle.

6.1 Generalized Cross-Validation

The method of generalized cross-validation (GCV) is to minimize the GCV function

$$G(\gamma) = \sum_{k=1}^{m} [\mathbf{b}_k - (\mathbf{A}\tilde{\mathbf{x}}_{\gamma}^{(k)})_k]^2, \tag{42}$$

where the $\tilde{\mathbf{x}}_{\gamma}^{(k)}$ is the estimate when the k^{th} measurement of $\mathbf{b}$ is omitted and the regularization parameter γ. For methods that are linear, the GCV expression can be greatly simplified [5]:

- *GCV for TSVD*: $G(k) = \frac{1}{(m-k)^2} \sum_{i=k+1}^{m} (\mathbf{u}_i^T \mathbf{b})^2$.
- *GCV for Tikhonov*: $G(\lambda) = \frac{\sum_{i=1}^{m} (\frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i^2 + \gamma^2})^2}{\sum_{i=1}^{m} (\frac{1}{\sigma_i^2 + \gamma^2})^2}$.

For methods that are nonlinear like the total variation regularization method, direct implementation of GCV is generally too costly. There have been attempts to iteratively solve (42) [8]. This was found not to be a feasible approach for this project given the responsiveness constraints of the GUI.

6.2 Discrepancy Principle

Given that we know the distribution of the noise ϵ we see that an initial parameter γ can be found by solving:

$$\|\mathbf{A}\mathbf{x}_\gamma - \mathbf{b}\|_2 = \nu E(\|\epsilon\|_2), \tag{43}$$

where $\nu = 2$ is a safety factor [5] and E denotes the expected value.

Although GCV is often favored when applicable because it does not require any prior assumptions about the problem, the discrepancy principle provides a simple method that can be solved using an efficient root finding algorithm.

7 Statistical Based Diagnostics

The use of residual diagnostics for improving regularization parameters was demonstrated by Bert Rust and Dianne O'Leary [15] as an effective tool to determine plausible regularized solutions. In this project we will use three diagnostics to generate a range of plausible regularization parameters.

7.1 Motivation

Assuming that the errors in the data are independently identically normally distributed with mean zero and variance one, the discretized linear regression model is

$$\mathbf{b} = \mathbf{A}\mathbf{x}^* + \epsilon, \tag{44}$$

where

$$\epsilon \sim N(\mathbf{0}, \mathbf{I_m}). \tag{45}$$

Now if we have an estimate $\tilde{\mathbf{x}}$ of $\mathbf{x}^*$ then the residual vector

$$\tilde{\mathbf{r}} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}} \tag{46}$$

for a plausible $\tilde{\mathbf{x}}$ should be a sample from the distribution from which ϵ is drawn since our linear regression model can be written

$$\epsilon = \mathbf{b} - \mathbf{A}\mathbf{x}^*. \tag{47}$$

7.2 Diagnostic 1

The sum of the squares of m are independent and identically distributed (i.i.d.) standard normal random variables are Chi^2 distributed with expected value m and the variance $2m$ [16]. Therefore, our first diagnostic is that the residual norm squared $\tilde{\mathbf{r}}$ should be within two standard deviations (within the 95% condidence interval) of the expected value of $\|\epsilon\|_{\mathbf{2}}^{\mathbf{2}}$ or

$$\|\tilde{\mathbf{r}}\|_2^2 \in [m - 2\sqrt{2m}, m + 2\sqrt{2m}]. \tag{48}$$

7.3 Diagnostic 2

The graph of the elements of the vector of the residual $\tilde{\mathbf{r}}$ should look like samples from the distribution $\epsilon \sim N(0,1)$. Quantitatively this test is based on the goodness of fit of the normal curve to the histogram of the elements $\tilde{r}_i$. One way to test the goodness of fit is the Chi^2 test, where the null hypothesis for standard normal distributed with significance 0.05 [16] or in other words is our sample normal distributed with 95% confidence. Another significance test is the Fisher test which has been shown to be more accurate.

7.4 Diagnostic 3

If we consider the elements of ϵ and $\tilde{\mathbf{r}}$ as time series with index $i = 1, \ldots, m$ where $\epsilon_i \sim N(0,1)$ forming a white noise series then $\mathbf{r}$ should also be a white noise series. One way to measure this quantitatively is to find the cumulative periodogram of the residual time-series.

To find the the cumulative periodogram one first finds the spectral density of the series by taking the Fourier transform

$$x_t = \frac{a_o}{2} + \sum_{k=1}^{n} (a_k \cos(w_k t) + b_k \sin(w_k t)), \tag{49}$$

where if x_t has m observations then $n = \frac{m-1}{2}$ and the frequency is $w_k = \frac{2\pi k}{n}$ for $k = 1, \ldots, n$. The periodogram is defined as

$$I(w_k) = \frac{n}{2}(a_k^2 + b_k^2). \tag{50}$$

[6] which represents the sum of squares of the coefficients at each frequency w_k. The cumulative periodogram is given by

$$C_k = \frac{\sum_{(j=0)}^{k} I_m(w_j)}{\sum_{(j=0)}^{n} I_m(w_j)}. \tag{51}$$

If we look at the spectral density of white noise we would expect the magnitude, which is measured by (50), to be equally distributed over every frequency which implies that the cumulative periodogram is a straight line between 0 and 1. The diagnostic then becomes a test to see if the plot of the cumulative periodogram of the residual plot looks like the cummulative periodogram of white noise. The 95% confidence interval is giving by plus (upperbound) or minus (lower bound) $1.36/\sqrt{(n-1)}$[6] from the linear line from 0 to 1. This approximation is said to hold for $n - 1 > 30$ [6].

8 Software

Below is a description of the pieces that form the Matlab software package for this project.

8.1 Frontend Software

The idea of having a frontend or Graphical User Interface (GUI) for this project is to provide an interactive platform for users who may have invaluable information about the image (for example, the doctor looking at a CT image). The GUI allows the user to change the regularization method or parameter to see the effect and use the statistical diagnostics for validation. The user does not need to have knowledge about the implementation to effectively use the interface.

8.2 Backend Software

- Regularization method
  - Regularization methods Tikhonov and TSVD (see Sections 4.1 - 4.2)
  - Total Variation regularization method (see Section 5)
- Method for initial parameter selection
  - Generalized Cross-Validation (GCV) for TSVD and Tikhonov (see Section 6.1)
  - Discrepancy Principle for Total Variation (see Section 6.2)
- Validate candidate solutions using statistical diagnostics
  - Apply statistical diagnostics (see Section 7).

9 Hardware

The software is designed to run on a modern desktop PC or laptop with no special hardware requirements.

10 Implementation

In this section we present details about the implementation of each piece of the matlab software included.

10.1 Tikhonov and TSVD with GCV for Initial Parameter Selection

Tikhonov is implemented using the *RestoreTools* function *Tikhonov.m*. The initial parameter is found using *GCVfun.m*, and Matlab function *fminbnd* is used to find the minimum. Note that there was a small mistake in *RestoreTools* implementation using *fminbnd* within *Tikhonov.m* that I fixed, properly documented in the function and files.

TSVD is implemented using the *RestoreTools* function *TSVD.m* where the initial parameter is found using *GCVforSVD.m*.

As with all methods in *RestoreTools*, Tikhonov and TSVD regularization method utilizes the *psfMatrix* class to make the implementation more efficient by not storing the $\mathbf{A}$ explicitly but instead creating a *psfMatrix* object which stores the Point Spread Function (PSF), a function that describes how a single point is blurred or distorted, and the boundary conditions from which the blurring matrix can be formed. Note that the PSF function must be separable. Operations like matrix vector multiplication, SVD, as well as operations on $\mathbf{A^T}$ make up the *psfMatrix* class. This greatly reduces the amount of storage because the PSF is much smaller than the blurring matrix.

10.2 Total Variation (TV) Regularization Method

The TV regularization method was implemented using the Primal-Dual Newton Method with Conjugate Gradients (CG)[3].

10.2.1 Pseudo-Code for Primal Dual Newton Method with CG

Algorithm 1 Primal-Dual Newton Method

```
Initialize: $\mathbf{x}_0$, set $k = 0$
while $\mathbf{x}_k$ is not "good enough" do
    Solve $\bar{\mathbf{C}}\mathbf{\Delta x_k} = -\bar{\mathbf{g}}(\mathbf{x})$ using CG to find the Newton direction $\mathbf{\Delta x}_k$.
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{\Delta x}_k$ where $\alpha_k$ is determined by a linesearch (cvsrch.m provided by Dianne O'Leary).
    $\mathbf{\Delta y} = -\mathbf{y} + \mathbf{E}^{-1}\mathbf{D}^T\mathbf{x} + \mathbf{E}^{-1}\bar{\mathbf{F}}\mathbf{D}^T\mathbf{\Delta x}$.
    $\mathbf{y}_{k+1} = \mathbf{y}_k + s_k \mathbf{\Delta x}_k$ where $s_k$ is determined by $s_k = .9sup\{s_d : \|y_i + s_d\Delta y_i\| < 1, i = 1, \ldots, m\}$.
    Set $k = k + 1$
end while
"good enough": $\|\mathbf{g}(\mathbf{x}_k)\|/\|\mathbf{g}(\mathbf{x}_0)\| < 10^{-3}$.
```

Algorithm 2 CG

```
Initialize: $\mathbf{r} = -\bar{\mathbf{g}}(\mathbf{x}) - \bar{\mathbf{C}}(\mathbf{x})\mathbf{\Delta x}_k$, $\mathbf{q} = \mathbf{r}$, $\rho = \|\mathbf{r}\|$, $\gamma = \rho$, $k = 0$.
while $\|\mathbf{r}\|/\rho >= tol$ do
  $\alpha = \frac{\rho}{\mathbf{q^T \bar{C}(x) q}}$.
  $\mathbf{\Delta x} = \mathbf{\Delta x} + \alpha \mathbf{q}$.
  $\mathbf{r} = \mathbf{r} - \alpha \bar{\mathbf{C}}(\mathbf{x})\mathbf{q}$.
  $\hat{\gamma} = \|\mathbf{r}\|^2$.
  $\beta = \frac{\hat{\gamma}}{\gamma}$, $\gamma = \hat{\gamma}$.
  $\mathbf{q} = \mathbf{r} + \beta \mathbf{q}$.
  Set $k = k + 1$.
end while
Tolerance for CG: $tol = 0.1$ when $k = 0$ and $tol = \min(0.1, 0.9\|\mathbf{g}(\mathbf{x}_k)\|^2/\|\mathbf{g}(\mathbf{x}_{k-1})\|^2)$ are the suggested conditions in [3].
```

10.2.2 Discrepancy Principle for Initial Parameter Selection

To find the initial TV parameter γ,

$$\|\mathbf{A}\mathbf{x}_\gamma - \mathbf{b}\|_2 - 2\|\epsilon\|_2 = 0 \qquad (52)$$

was solved using Matlab's root finding function *fzero*. Since the parameter selection only has to be a rough estimate, the primal-dual TV method is only run for 5 iterations and the tolerance for *fzero* was set to $tol = 10^{-9}$.

10.3 Efficient Implementation

To achieve efficient implementation, we make the algorithm low storage in order to work with larger images. We include the option to not store the Hessian explicitly which is an advantage of using CG. Instead we use a function to form

$$\mathbf{H}\mathbf{v} = \mathbf{H}(\mathbf{x})\mathbf{v}, \qquad (53)$$

where $\mathbf{v}$ is an arbitrary vector. In the implementation, the blurring matrix is also not stored explicitly. For smaller images (smaller than 32×32) a sparse representation is used. This was chosen because the RestoreTools class *psfMatrix* does not work with smaller images. For larger images *psfMatrix* is used which stores the PSF and information regarding the boundary conditions as a structure and performs the necessary computations without explicitly storing the matrix.

In addition to low-storage the method should be fast enough to work with the GUI interface. In order to cut down on the number of function evaluations, a structure was used to pass information between modular pieces. Additional modifications were made in the direct implementation to the GUI to make it work with the slidebar which requires very fast computation. I choose to work with 16×16 images because of the short computation time for a large range of parameters (see Figure **??**). In addition I used the solution found through the discrepancy principle as an initial guess.

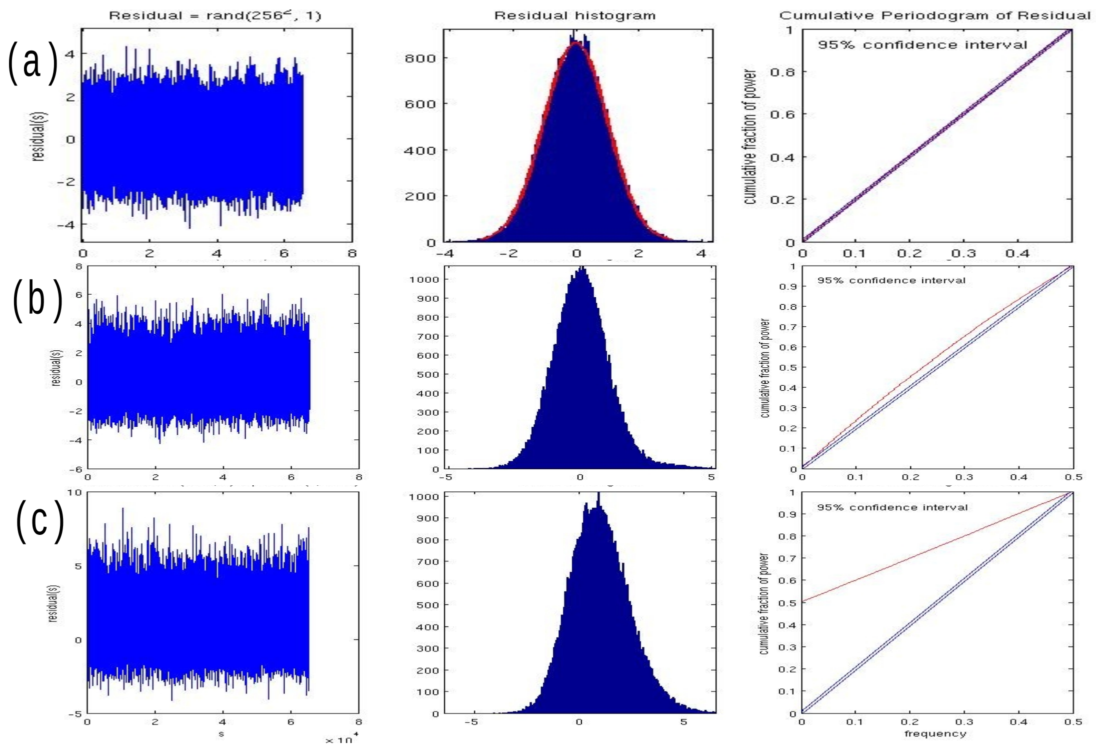


Figure 3: Plot of the residual, histogram of the residual and the cumulative for three different residuals. (a) Normal distributed residual (b) Normal plus Poisson, (c) Normal plus features

10.4 Statistical Diagnostics

- Diagnostic 1. This is directly implemented by checking whether $\|\tilde{\mathbf{r}}\|_2^2$ falls within the bounds $[m - 2\sqrt{2m}, m + 2\sqrt{2m}]$ for each solution.
- Diagnostic 2. This diagnostic is implemented visually using the Matlab plotting function *hist*. The null χ^2 hypothesis test is implemented using the Matlab function *chi2gof(resid,'cdf',@normcdf)* assuming 10 bins.
- Diagnostic 3. The cumulative periodogram was found by first using the Fast Fourier Transform (the Matlab function *fft*), then taking the square of each element, and finally finding the cumulative sum. The diagnostic is measured by finding the number of points that fell outside the 95% interval over each frequency. The implementation follows the implementation of *checkperiod.m* by Dianne O'Leary which follows a Fortran program by Bert Rust.

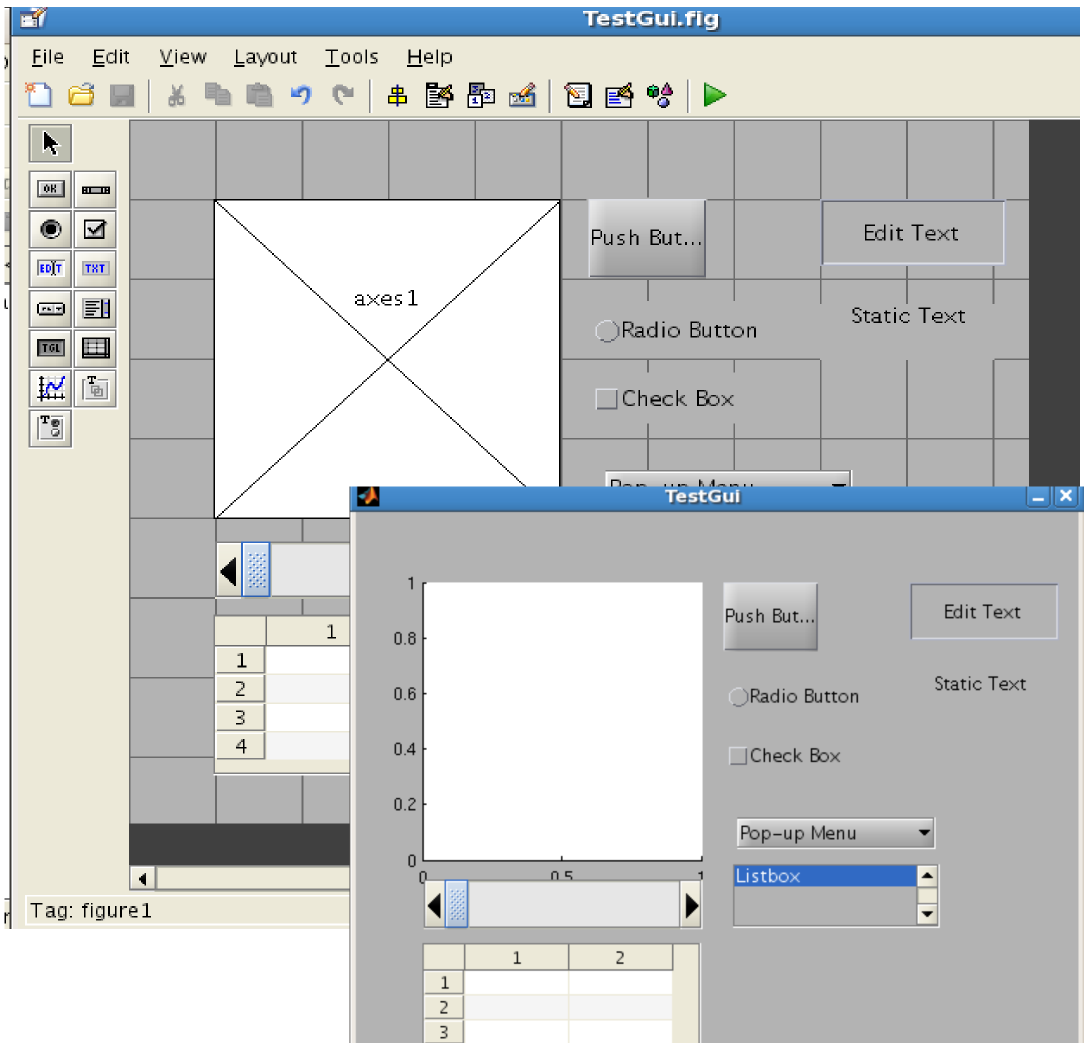
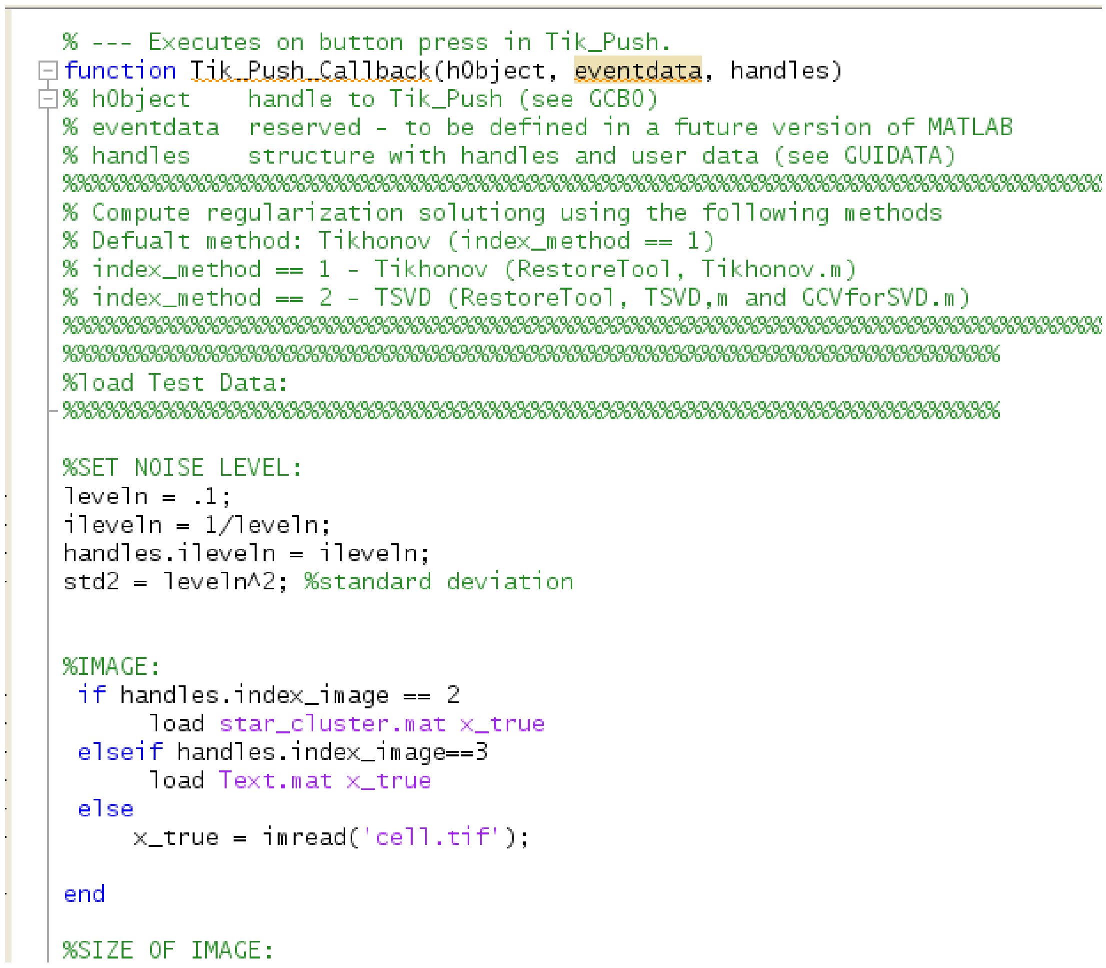
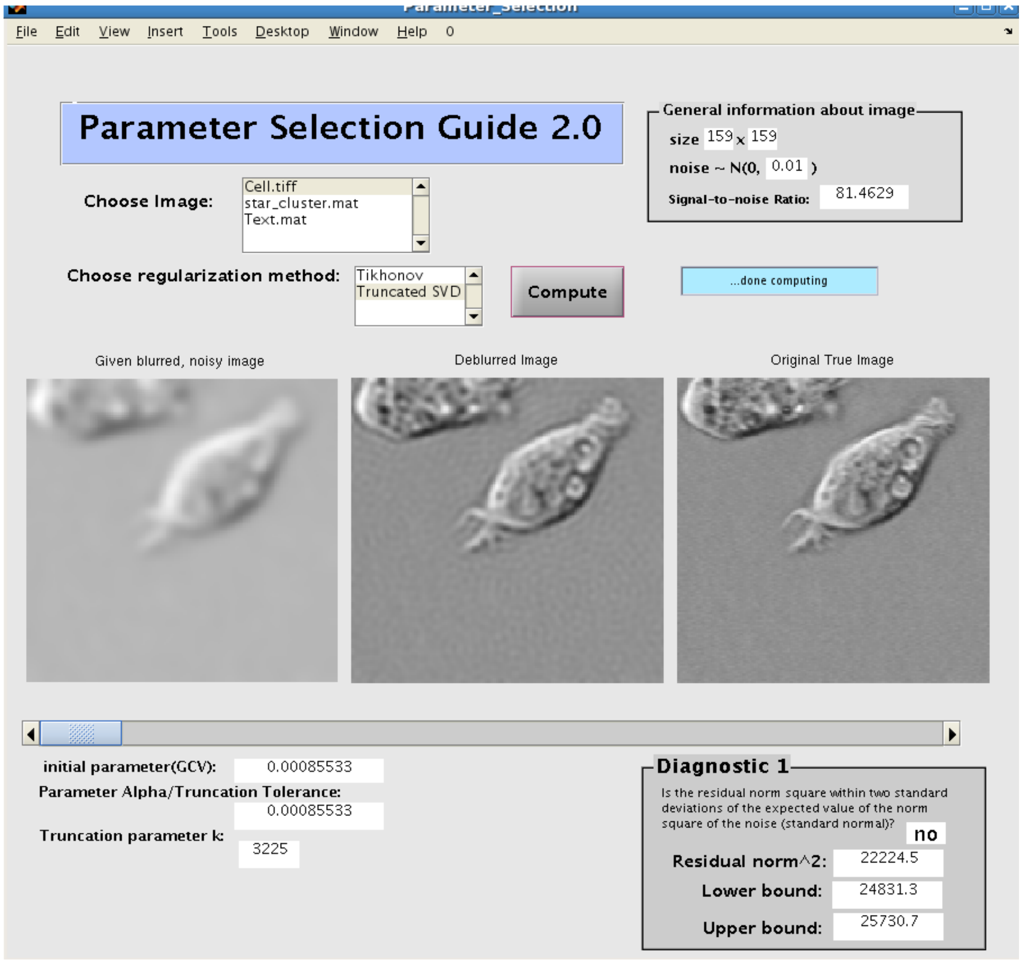


Figure 4: Back: GUI figure generation tool. Front: Resulting GUI.



```
% --- Executes on button press in Tik_Push.
function Tik_Push_Callback(hObject, eventdata, handles)
% hObject    handle to Tik_Push (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute regularization solutiong using the following methods
% Defualt method: Tikhonov (index_method == 1)
% index_method == 1 - Tikhonov (RestoreTool, Tikhonov.m)
% index_method == 2 - TSVD (RestoreTool, TSVD,m and GCVforSVD.m)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%load Test Data:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%SET NOISE LEVEL:
leveln = .1;
ileveln = 1/leveln;
handles.ileveln = ileveln;
std2 = leveln^2; %standard deviation


%IMAGE:
 if handles.index_image == 2
      load star_cluster.mat x_true
 elseif handles.index_image==3
      load Text.mat x_true
 else
     x_true = imread('cell.tif');

end

%SIZE OF IMAGE:
```

Figure 5: Snip-it of a *callback function* taken from the parameter selection GUI


Parameter_Selection
File Edit View Insert Tools Desktop Window Help
Compute
Given blurred, noisy image
Deblurred Image
Original True Image

Parameter Selection Guide 2.0

Choose Image: Cell.tiff, star_cluster.mat, Text.mat

Choose regularization method: Tikhonov, Truncated SVD

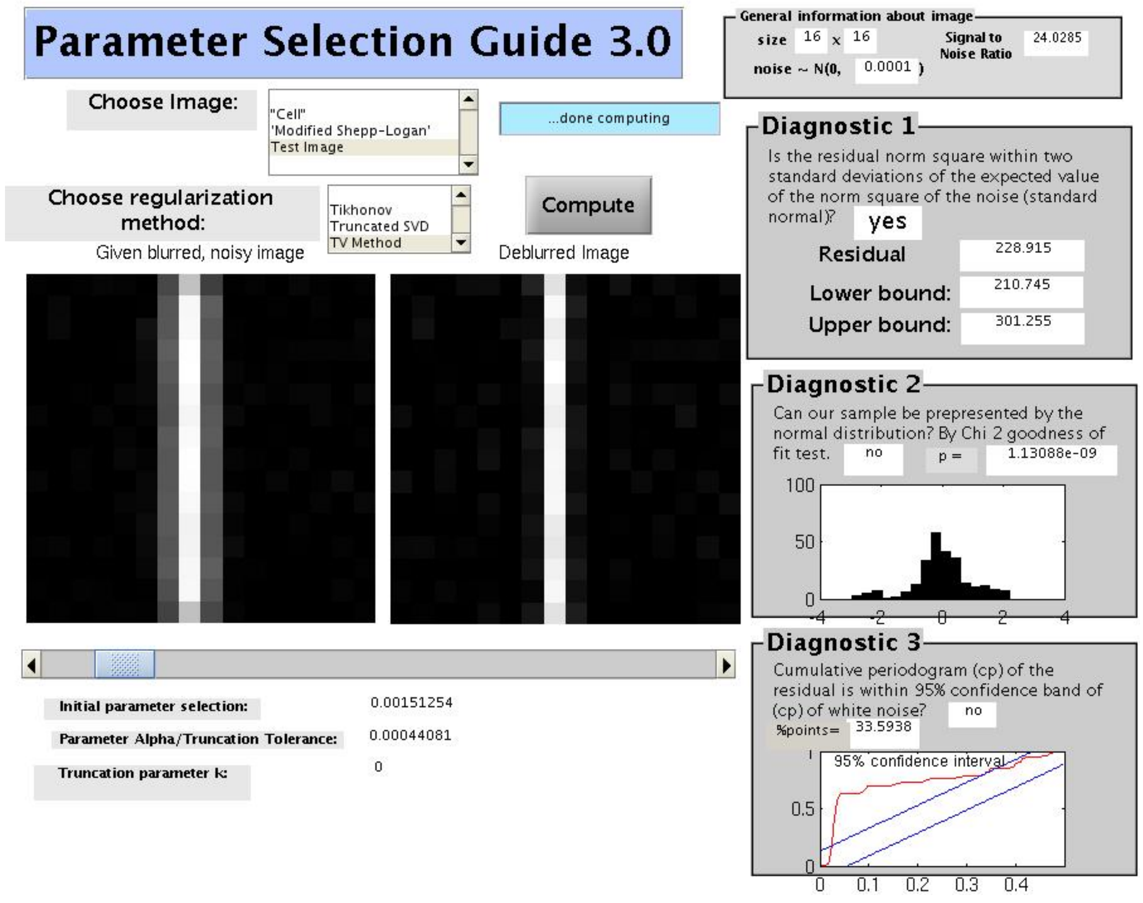
...done computing

General information about image
size 159 x 159
noise ~ N(0, 0.01)
Signal-to-noise Ratio: 81.4629

initial parameter(GCV): 0.00085533
Parameter Alpha/Truncation Tolerance: 0.00085533
Truncation parameter k: 3225

Diagnostic 1
Is the residual norm square within two standard deviations of the expected value of the norm square of the noise (standard normal)? no
Residual norm^2: 22224.5
Lower bound: 24831.3
Upper bound: 25730.7

Figure 6: Initial parameter selection with Diagnostic 1 and the true image along with the blurred and deblurred image



Parameter Selection Guide 3.0

Choose Image: "Cell" / 'Modified Shepp-Logan' / Test Image

...done computing

Choose regularization method: Tikhonov / Truncated SVD / TV Method

Compute

Given blurred, noisy image

Deblurred Image

Initial parameter selection: 0.00151254

Parameter Alpha/Truncation Tolerance: 0.00044081

Truncation parameter k: 0

General information about image

size 16 x 16

noise ~ N(0, 0.0001)

Signal to Noise Ratio 24.0285

Diagnostic 1

Is the residual norm square within two standard deviations of the expected value of the norm square of the noise (standard normal)? yes

Residual 228.915

Lower bound: 210.745

Upper bound: 301.255

Diagnostic 2

Can our sample be prepresented by the normal distribution? By Chi 2 goodness of fit test. no p = 1.13088e-09

Diagnostic 3

Cumulative periodogram (cp) of the residual is within 95% confidence band of (cp) of white noise? no

%points= 33.5938

Figure 7: Final implementation of the GUI with the three Diagnostics

10.5 Frontend

The GUI interface was implemented using Matlab's GUI interface toolbox. The GUI is built by first creating a figure window using a figure editor. Within the editor the user has a number of components (fields, axes, sliders, dropbox, input fields, push buttons....) to choose from. User-operated controls, dynamic and static components can be added. Once the figure has the desired features Matlab generates the basic code the figure (Figure 4). The components that are user-operated or dynamic must have a *Callback function* to designate the response to the user-generated event. Each component has a tag and a handle structure (e.g., the slider has a handle for maximum value, minimum value and current value). Figure 5 is an example of the *Callback function* for the parameter selection tool.

In Figure 7 is the final GUI for this project. In this GUI the user first chooses a test image and regularization method (left corner) then presses the "compute" button. For the given method and blurred image, an initial parameter is computed using either GCV or discrepancy principle. In addition the GUI also displays information about the image (right corner), the blurred image, and deblurred image for the initial parameter. The user can then adjust the slide bar and in realtime the new solution (deblurred image) is computed and displayed. In the right column the user can verify that the parameter selected is plausible by checking to see if the three diagnostics are satisfied.

11 Results and Testing

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

11.1 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 by

$$SNR = 10\log_{10}(\frac{\|\mathbf{b}\|^2}{\|\epsilon\|^2}). \tag{54}$$

The range of plausible parameters that meet the diagnostics increases as the SNR goes to zero, and as the SNR increases the range of plausible solutions becomes smaller (Figure 8). Note that a user should be careful when they have 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 similar relation was found for the different methods, diagnostics and different test images included in the GUI.

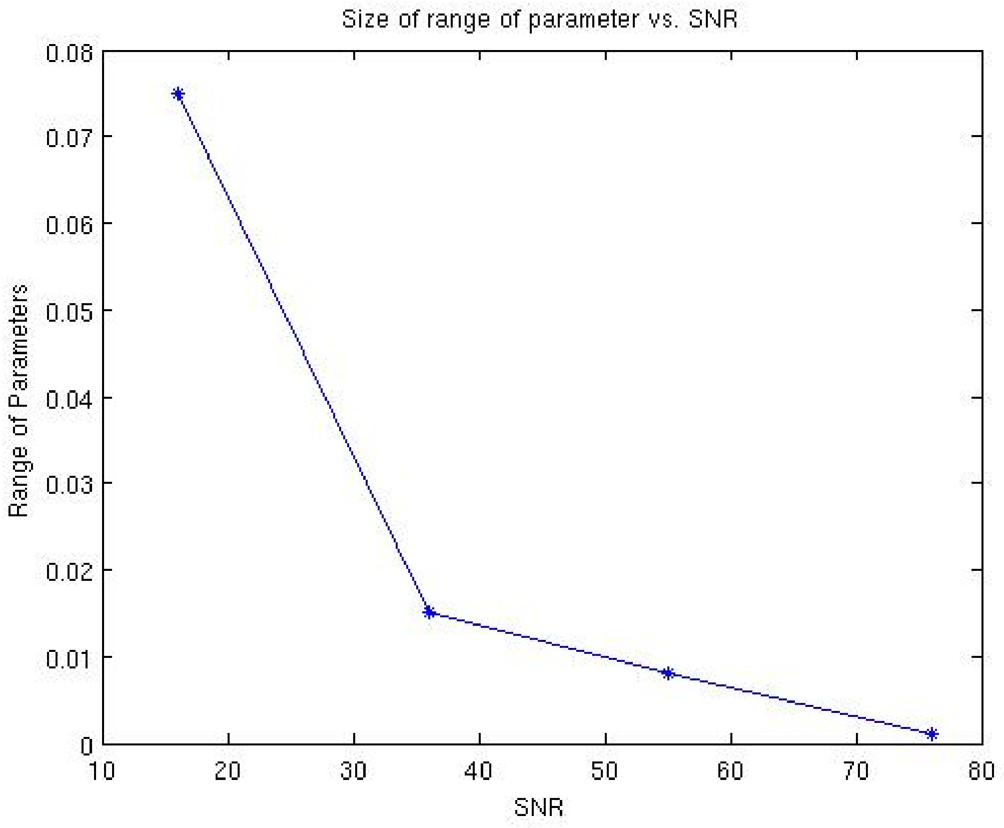


Figure 8: For the 16×16 segment of the image "cell.tif" the range (± 0.0025) was found for Diagnostic 1 using Tikhonov regularization. From the plot we can see that as the Signal-to-Noise Ratio decreases the range of parameters satisfying the diagnostic increases
.

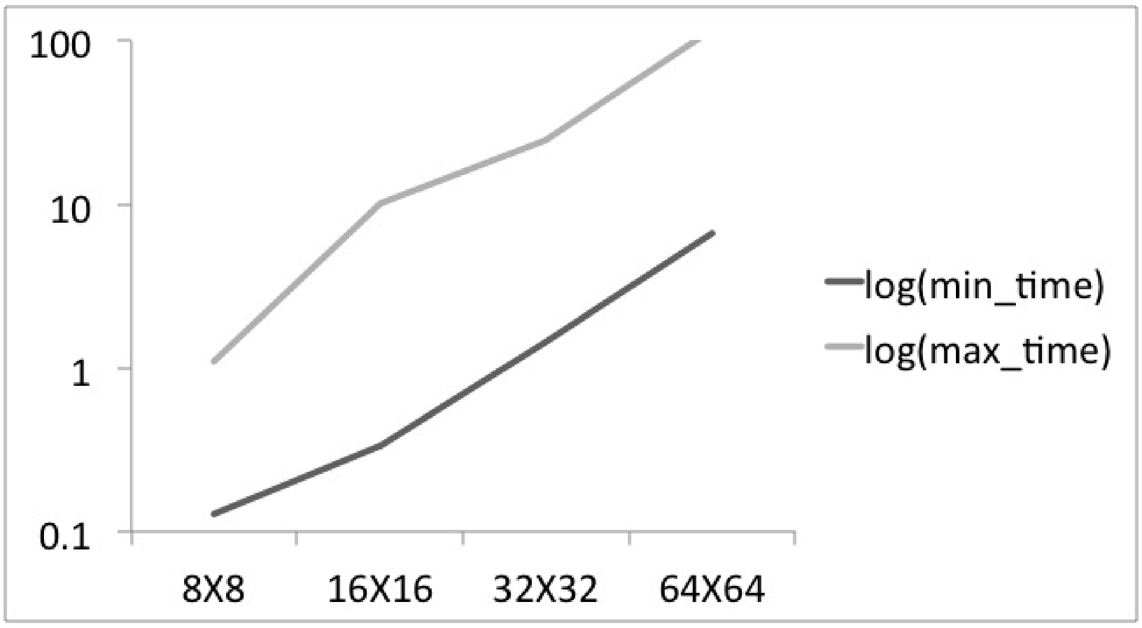


Figure 9: The difference in log_{10} of computational time (seconds) for the TV regularization method for parameters between $\gamma = 1$ and $\gamma = 10^{-9}$. For all image sizes the maximum time was for $\gamma = 1$.
.

11.2 Effects of γ on Computation Time

It was found that the computation time of the TV regularization method (dependent on the number of CG iterations) is dependent on the value of γ. Figure 9 shows the range. See Figure 10 for results for different values of γ.

As a result of this fact, I looked into using preconditioners to speed up the algorithm for large γ. ILU is suggested for the Primal-Dual Newton method [3]. I did not find the use of ILU to be robust enough to be included in the GUI as I had to adjust the drop tolerance depending on γ.

11.3 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 PSF. See Figures 11 - 12 comparing the results using Tikhonov and TSVD to the Primal Dual TV method used.

12 Validation

All the regularization methods have been initially tested on the same set of test images using the same PSF function and boundary conditions and noise levels. The code is written to work with any separable PSF (a restriction of *psfMatrix*).

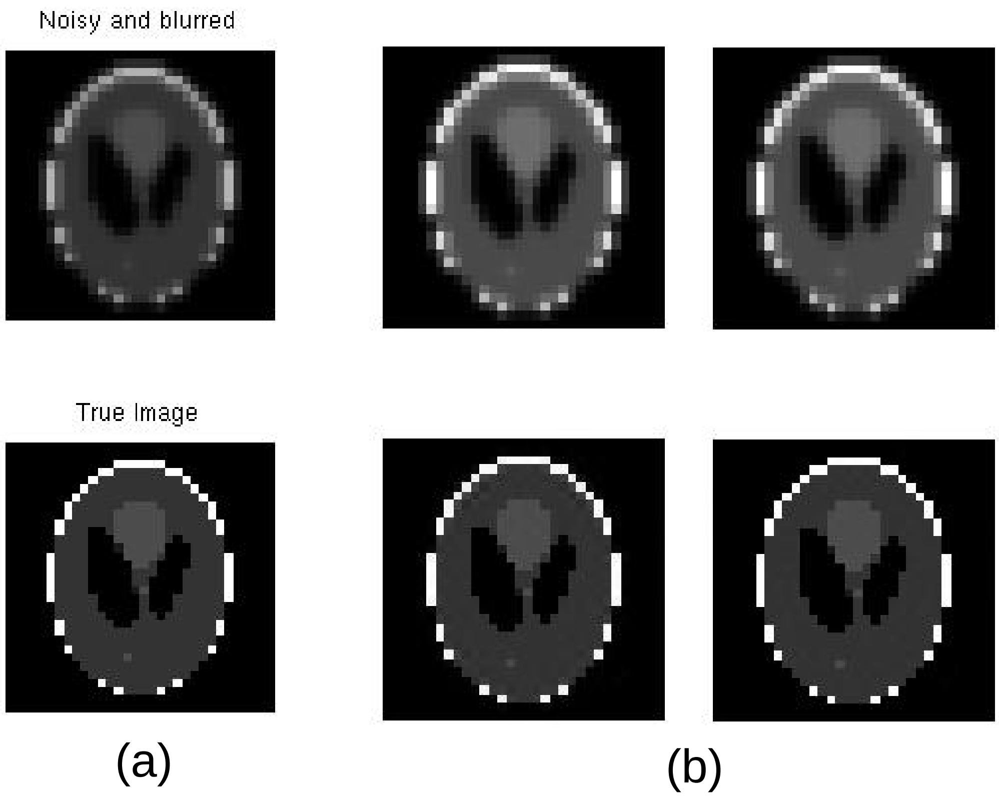


Figure 10: (a) True image and Blurred/Noisy image (b)Top: Left to right: $\gamma = 1$, $\gamma = 10^{-2}$ Bottom: Left to right: $\gamma = 10^{-4}$, $\gamma = 10^{-6}$
.

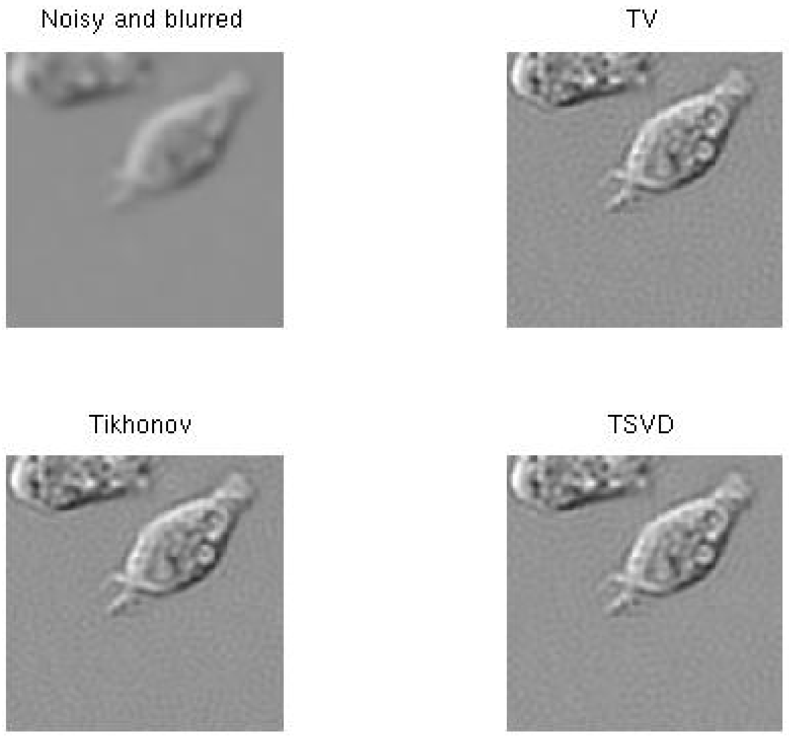


Figure 11: 129×129 image of "cell.tiff" with Gaussian blur and zero boundary conditions with SNR of 60
.

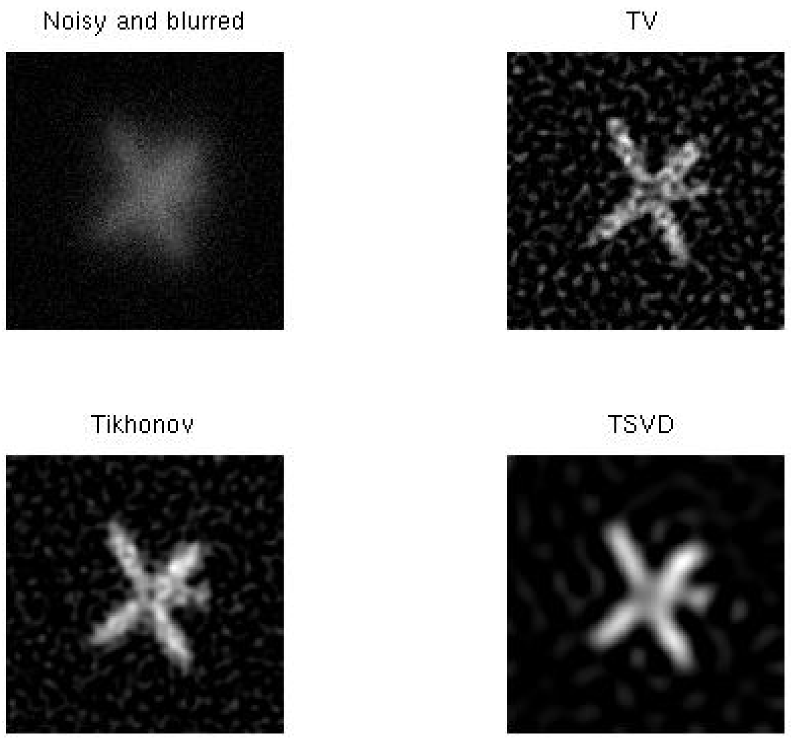


Figure 12: 256×256 image of Satellite with PSF provided in *RestoreTools* with zero boundary conditions and SNR=9
.

Residual	Diag. 1	Diag. 2	Diag. 3	Fisher
$r_n \sim N(0,1)$	51	46	14	48
$r_n + I(i)$	950	999	539	1000
$r_n + .05 * r_p \; r_p \sim pois(1)$	156	1000	149	1000

Table 1: For 1000 runs, number of times the Diagnostics are **NOT** satisfied. $I(i) = 1$ if $(i-1)mod(100) = 0$ or $(i-2)mod(100) = 0$ and $I(i) = 0$ otherwise.

For testing purposes a Gaussian PSF with zero boundary conditions was used unless otherwise noted.

12.1 Validation of Diagnostics

The validation of the diagnostics was done experimentally based on expected statistical results. Each of the diagnostics were tested first with standard normal i.i.d. samples to confirm that that each of the diagnostics is satisfied approximately 95% of the time (see Table 1).

From the table it is seen for 1000 different standard normal i.i.d. samples the three diagnostics and the Fisher normality test (not described) tell us for these 1000 samples, 95% satisfy the diagnostics. Then if we perturb the standard normal samples by periodically adding artifacts or adding small faction of poisson distributed samples one sees that Diagnostic 2 and the Fisher Test (which should be equivalent) do a very good job of determining that the sample is no longer stardard normally distributed which is what we expected given the pertubation.

12.2 Validation of Total Variation Method

The implementation was programmed modularly so that each piece (Newton Step, CG, function evaluations, linesearches) can be validated individually.

- CG was validated for small $\mathbf{Ax} = \mathbf{b}$ test problems where the results could be verified.
- The minimization function $f(\mathbf{x})$, gradient $\mathbf{g}(\mathbf{x})$, and Hessian times a vector $\mathbf{Hv}(\mathbf{x}, \mathbf{v})$ were run separately and were validated for a given input $\mathbf{x}$.

Basic results for blurred images without noise are shown in Figure 13. TV method is seen to have small relative error.

In addition a modular piece of the code was also compared to results found using an implementation of the Primal-Dual Newton's method by Curtis Vogel [14]. To run the comparison I used the "cell.tif" and a Gaussian PSF (Figure 14). Shown are results for a 64×64 image with $SNR = 81$ the relative error of my implementation was 0.9% compared to 1.02% for Vogel (Figure 15). But when tested for smaller SNR values (more noise) the Vogel algorithm produced results with much better relative error compared to my implementation, this is most likely accounted because of differences in internal convergence tolerations.

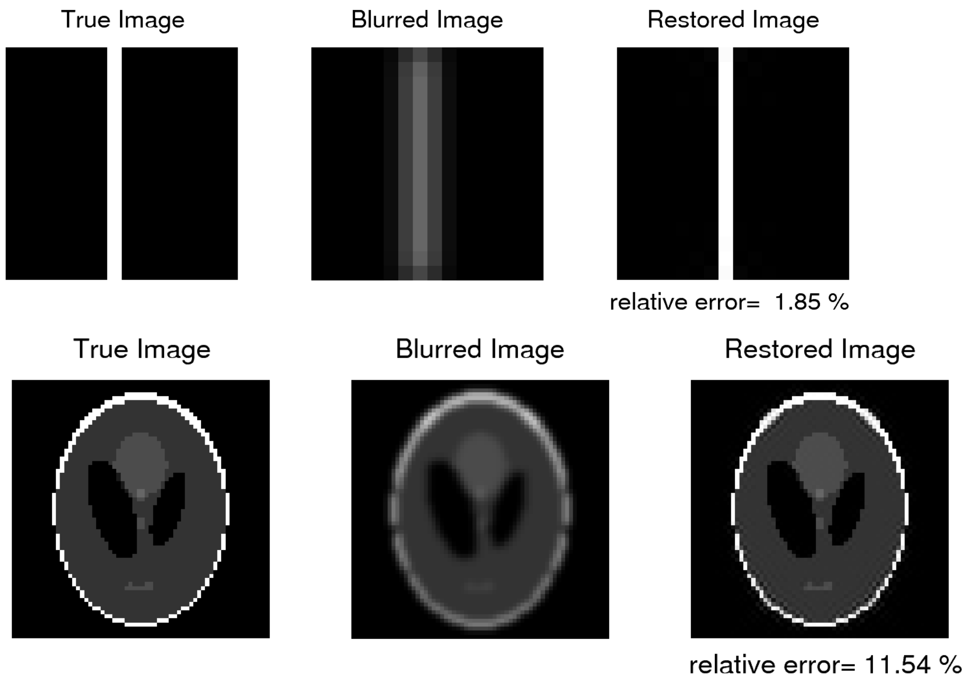


Figure 13: Results of Newton-CG for Total Variation regularization for two images, top is a 16×16 generated in Matlab and the bottom image is 64×64 Modified Shepp-Logan generated in Matlab

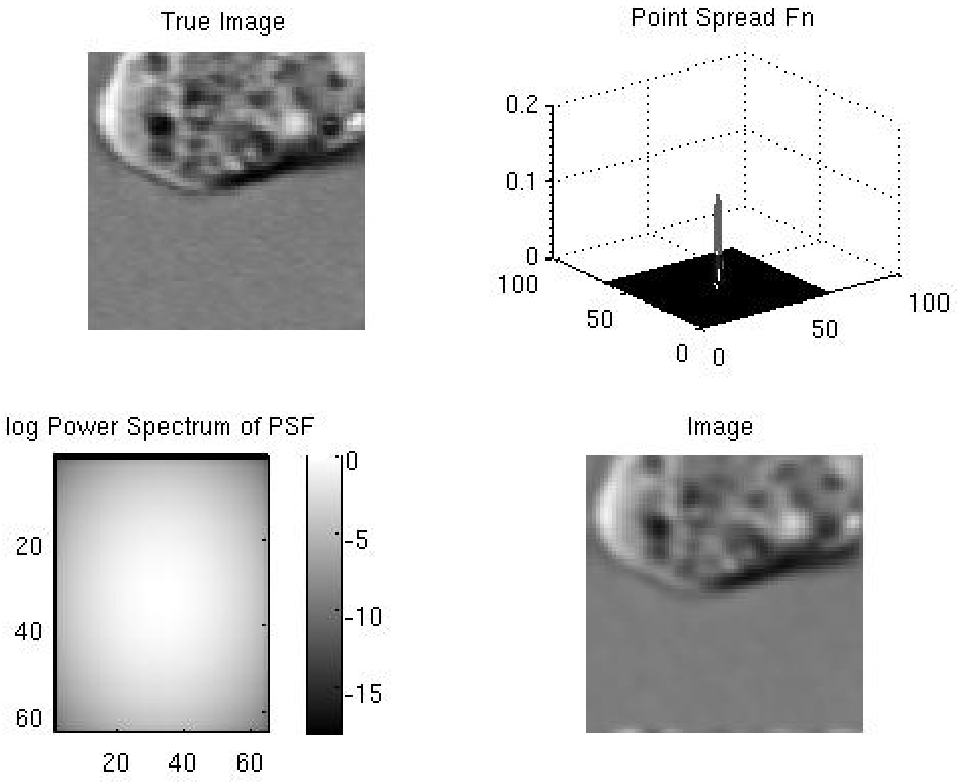


Figure 14: Clockwise: 64×64 image, PSF, Power Spectrum of PSF, Blurred and Noise image ($SNR = 81$)
.

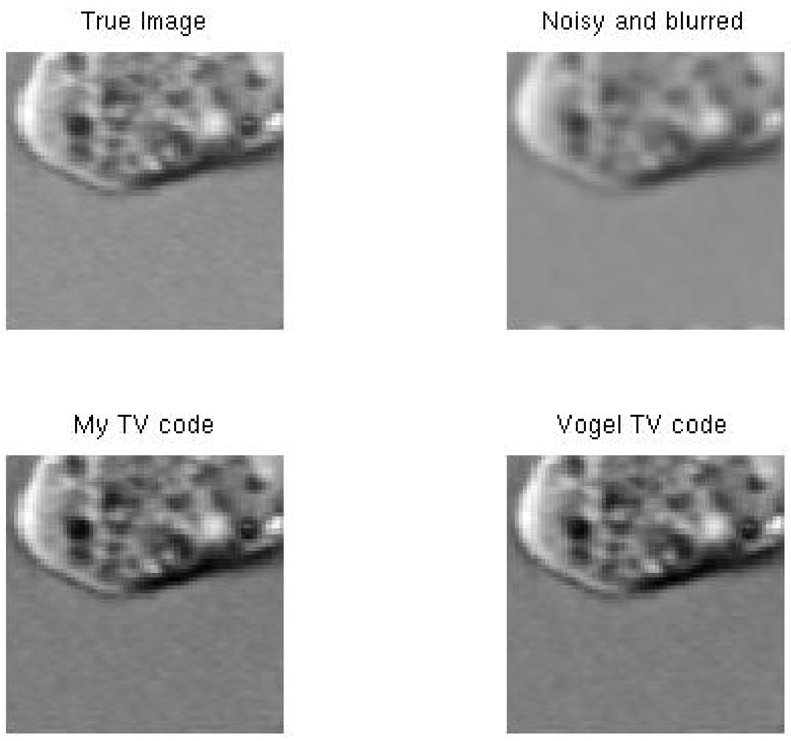


Figure 15: Results of the TV regularization method for my implementation and code found in [14].
.

12.3 Validation of Usefulness of Software

In addition to the demonstration of the GUI done as part of the AMSC 663/664 course I also demonstrated the software outside of class. The GUI was presented and distributed to Dr. Dianne O'Leary's undergraduate class on Deblurring Digital Images (CMSC/AMSC 498D) as a educational too. Guided by this, a number of students included GUI's in their final project. The software was also presented to the AMSC student seminar where I got feedback from two students on the potential usefulness of the GUI for their current work. Picking appropriate methods and regularization parameters remains a challenge in application. This software is a proof of concept that such a tool can help address these issues.

13 Project Milestones

- A basic GUI to use existing tools in RestoreTools [5]
- Validated periodogram diagnostics to RestoreTools
- GUI that help determines a range of plausible regions using the periodogram diagnostics
- Outline of Total Variation regularization algorithm and basic implementation in Matlab
- Deliver mid-year report and presentation
- Total Variation (TV) regularization in RestoreTools framework
- Initial parameter selection tool (discrepancy principle) for Total Variation regularization method
- Validated TV to with initial parameter selection tool included in the GUI
- Software package finished and optimized
- Deliver final presentation and report

14 References

1. R. Acar and C.R. Vogel. Analysis of Bounded Variation Penalty Methods for Ill-Posed Problems. *Inverse Problems.* 10: 1217-1229, 1994.

2. Tony F. Chan and Jianhong Shen. Image Processing and Analysis. *SIAM*, Philadelphia, PA, 2005.

3. 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*, Vol.219, pp.241-251, 1996.

4. Per Christian Hansen. Regularization Toolbox, http://www2.imm.dtu.dk/ pch/Regutools/regutools.html

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

6. Wayne A. Fuller. Introduction to Statistical Time Series, *Wiley-Interscience*, New York, NY, 1996.

7. C.T. Kelly. Iterative Methods for Linear and Nonlinear Equations. *SIAM*, Philadelphia, PA, 1995.

8. P. Liu, D. Liu. Selection of regularization parameter based on generalized cross-validation in total variation remote sensing image restoration. *Image and Signal Processing for Remote Sensing*, Vol.7830, 2010.

9. James G. Nagy, RestoreTools, http://www.mathcs.emory.edu/ nagy/RestoreTools/

10. James G. Nagy, K. Palmer and L. Perrone. Iterative Methods for Image Deblurring: A Matlab Object Oriented Approach. *Numerical Algorithms.* 36: 73-93, 2004.

11. Stephen G. Nash. Linear and Nonlinear Programming. *McGraw-Hill*, 1996.

12. Dianne P. 0'Leary. Scientific Computing with Case Studies. *SIAM*, Philadelphia, PA, 2009.

13. Curtis R. Vogel. Computational Methods for Inverse Problems. *SIAM*, Philadelphia, PA, 2002.

14. Curtis R. Vogel. Chap 8 Code. www.math.montana.edu/ vogel/Book/Codes/Ch8, 2001.

15. C. Vogel and M. Oman. Iterative methods for total variation denoising, *SIAM Journal on Scientific Computing*, Vol.17, pp.227 238, Jan.1996.

16. Sheldon Ross. A First Course in Probability, Sixth Edition. *Prentice Hall*, Upper Saddle River, NJ, 2002.

17. Bert W. Rust and Dianne P. O'Leary. Residual programs for choosing regularization parameters for ill-posed problems. *Inverse Problems*, 24:034005 (30 pages), 2008. Invited Paper

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