Variable Projection Method for Semi-Blind Deconvolution with Mixed Gaussian Kernels

by

Jordan Dworaczyk

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Arts

Approved June 2023 by the
Graduate Supervisory Committee:

Malena Español, Chair
Bruno Welfert
Rodrigo Platte

ARIZONA STATE UNIVERSITY

August 2023

ABSTRACT

The variable projection method has been developed as a powerful tool for solving separable nonlinear least squares problems. It has proven effective in cases where the underlying model consists of a linear combination of nonlinear functions, such as exponential functions. In this thesis, a modified version of the variable projection method to address a challenging semi-blind deconvolution problem involving mixed Gaussian kernels is employed. The aim is to recover the original signal accurately while estimating the mixed Gaussian kernel utilized during the convolution process. The numerical results obtained through the implementation of the proposed algorithm are presented. These results highlight the method's ability to approximate the true signal successfully. However, accurately estimating the mixed Gaussian kernel remains a challenging task. The implementation details, specifically focusing on constructing a simplified Jacobian for the Gauss-Newton method, are explored. This contribution enhances the understanding and practicality of the approach.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 The Semi-Blind Deconvolution Problem

Consider a one-dimensional convolution problem of the form

$$d(s) = \int_a^b g(s - s')x(s')\,ds'. \tag{1.1}$$

This is a one-dimensional version of a model that occurs in two-dimensional imaging problems [1]. In imaging applications, the function x represents light source intensity as a function of spatial position, and the function d represents image intensity. The function g is referred to as the kernel of the convolution and characterizes the blurring effects that occur during the formation of the image.

In this thesis, we consider the particular case when the kernel g is a mixture of two Gaussian functions:

$$g(s; b_1, b_2, \sigma_1, \sigma_2) = b_1 \frac{\exp\left(-\frac{s^2}{2\sigma_1^2}\right)}{\int \exp\left(-\frac{s^2}{2\sigma_1^2}\right) ds} + b_2 \frac{\exp\left(-\frac{s^2}{2\sigma_2^2}\right)}{\int \exp\left(-\frac{s^2}{2\sigma_2^2}\right) ds}. \tag{1.2}$$

Notice that each Gaussian kernel is normalized. The normalization is important in applications such as imaging problems because to correctly model the image deblurring process, it is required that the intensity of each pixel does not become greater than its true intensity after being blurred.

The convolution problem (1.1) can be transformed into its discrete form by using midpoint quadrature and can then be formulated in matrix-vector notation,

$$d = G(y_{true})x_{true},$$

where $G(y_{true})$ is an $m \times m$ matrix whose entries depend on the kernel g and the vector of true parameters denoted by $y_{true} = (b_1, b_2, \sigma_1, \sigma_2)$ that defines it. We call G the blurring operator. The matrix-vector product $G(y_{true})x_{true}$ then represents the convolution (or blur) of the true signal. In coefficient form, the matrix $G(y_{true})$ can be expressed as

$$\left[G\big((b_1, b_2, \sigma_1, \sigma_2)\big)\right]_{ij} = b_1 \frac{\exp\left(\frac{-(i-j)^2}{2\sigma_1^2}\right)}{\sum_k \exp\left(\frac{-k^2}{2\sigma_1^2}\right)} + b_2 \frac{\exp\left(\frac{-(i-j)^2}{2\sigma_2^2}\right)}{\sum_k \exp\left(\frac{-k^2}{2\sigma_2^2}\right)}.$$

In the application of imaging, during the capture of an image, it is inevitable that small random errors (or noise) may become present in the recorded signal [2]. Gaussian noise is added to the convoluted signal to model when the signal d contains noisy observations. Therefore, a better formulation of the process is given by

$$d = G(y_{true})x_{true} + \epsilon_{noise}. \tag{1.3}$$

The aim is now to approximate the true signal, $x_{true} \in \mathbb{R}^m$, given the blurred signal, $d \in \mathbb{R}^m$, provided that there is a parameterized model for the kernel. When the blurring operator is not explicitly known, the problem is called blind deconvolution. In this case, a parameterized kernel model is assumed, and it defines a blurring operator. Therefore, this problem is referred to as a semi-blind deconvolution problem.

The blurring matrix becomes increasingly ill-conditioned as the number of discretization points used to construct it increases. Thus, errors in the blurred signal caused by noise become greatly amplified during the inversion process, ultimately leading to poor approximations of the true signal [1]. Therefore, regularization is introduced to penalize the noise present in the blurred signal. More specifically, regularization by spectral filtering can be employed. To explain the need for regularization, suppose the discrete model (1.3) is given with fixed parameter y. Then, we

use the singular value decomposition (SVD) [2, p. 9] of $G(y) \in \mathbb{R}^{m\times m}$ defined as

$$G(y) = U\Sigma V^T, \tag{1.4}$$

where U and V are orthogonal matrices, satisfying $U^TU = I$ and $V^TV = I$, and $\Sigma = \operatorname{diag}(s_i)$ is a diagonal matrix whose elements are nonnegative and appear in nonincreasing order,

$$s_1 \geq s_2 \geq \cdots \geq s_m \geq 0. \tag{1.5}$$

The quantities s_i are called the singular values of $G(y)$. The columns u_i of U are called the left singular vectors, while the columns v_i are called the right singular vectors. If all singular values are assumed to be strictly positive, then it follows that

$$G(y)^{-1} = V\Sigma^{-1}U^T. \tag{1.6}$$

And, since Σ is a diagonal matrix, its inverse Σ^{-1} is also diagonal, with entries $1/s_i$ for $i = 1, \ldots, m$. Moreover, using the SVD of $G(y)$, the blurring operator can be written as a linear combination of rank-one matrices. That is,

$$\begin{aligned} G(y) &= U\Sigma V^T \\ &= \begin{bmatrix} u_1 & \ldots & u_m \end{bmatrix} \begin{bmatrix} s_1 & & \\ & \ddots & \\ & & s_m \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_m^T \end{bmatrix} \\ &= u_1 s_1 v_1^T + \cdots + u_m s_m v_m^T \\ &= \sum_{i=1}^{m} s_i u_i v_i^T. \end{aligned}$$

Let $G^{-1}(y)$ denote the inverse of the matrix $G(y)$. Then similarly,

$$G^{-1}(y) = \sum_{i=1}^{m} \frac{1}{s_i} v_i u_i^T.$$

Next, if the noise in the signal is ignored and solving for x_{true} in a naïve way is considered, then,

$$\begin{aligned} G^{-1}(y)d &= G^{-1}(y)(G(y)x_{true} + \epsilon_{noise}) \\ &= x_{true} + G^{-1}(y)\epsilon_{noise} \\ &= x_{true} + \sum_{i=1}^{m} \frac{(u_i^T \epsilon_{noise})v_i}{s_i}. \end{aligned}$$

Notice that if the noise term ϵ_{noise} were not present, then taking the inverse of $G(y)$ would solve for x_{true} exactly; however, noise is indeed present in the signal. Instability in the solution arises from the division of singular values, s_i, which are close to zero. The errors in the noisy signal and the small singular values from the ill-conditioned matrix $G(y)$ contaminate the solution. A regularization filter function can be used to filter out these troublesome singular values and thus approximate a solution to x_{true}.

An example of what the solution to x_{true} looks like when regularization is not used, is provided in Figure 1.1 where it can be observed how the noise contaminates the solution. According to [2, p. 10], for discrete ill-posed problems as ours, the vectors u_i and v_i tend to have more sign changes as the index i increases, representing high-frequency information. An example of this can be seen in Figure 1.2. The authors explain that the higher-frequency information is amplified as the magnitude of the coefficients $u_i^T \epsilon_{noise}/s_i$ grows. Regularization aims to filter out this higher-frequency information.

In this thesis, the Tikhonov regularization filter is implemented,

$$w_\lambda(s) = \frac{s^2}{s^2 + \lambda^2},$$

resulting in the approximate solution to x_{true} given by

$$x = \sum_{i=1}^{m} w_\lambda(s_i) u_i^T dv_i = \sum_{i=1}^{m} \frac{s_i}{s_i^2 + \lambda^2} u_i^T dv_i. \tag{1.7}$$

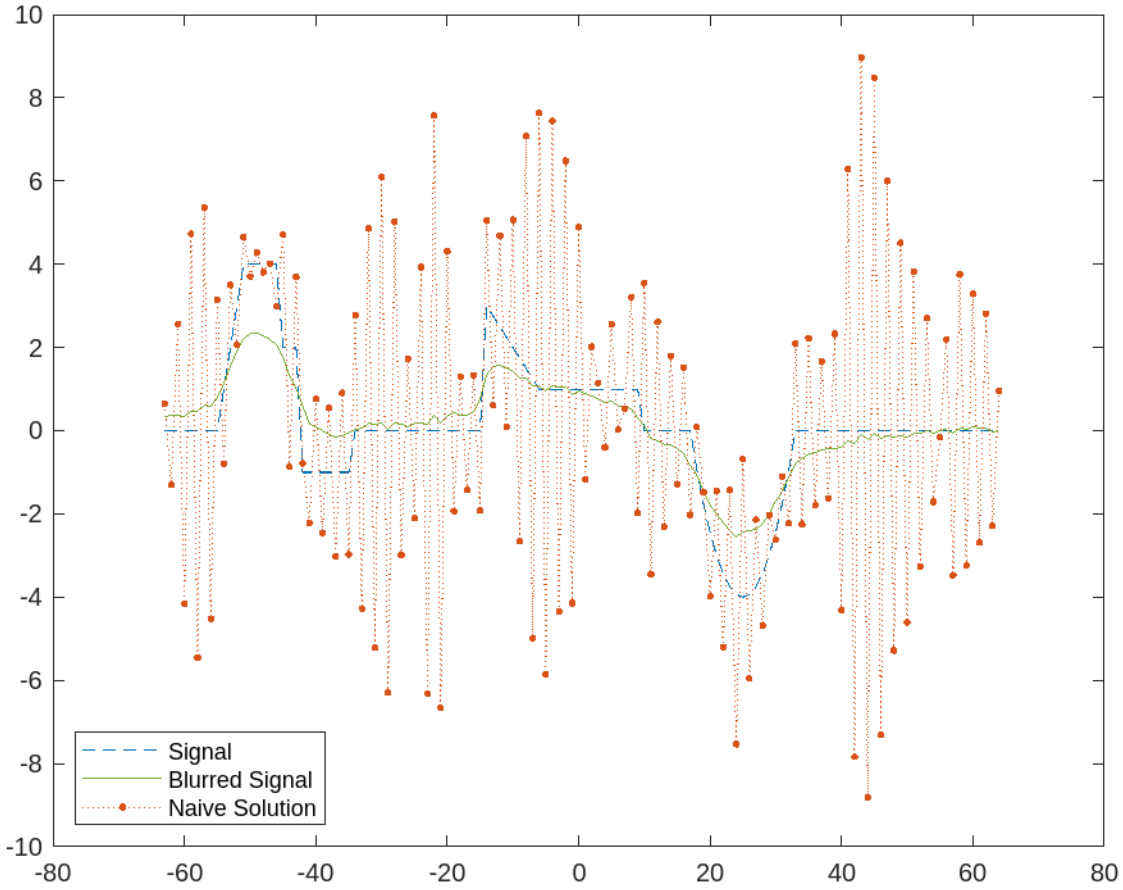

Figure 1.1: Naïve solution. The noisy signal is produced by using convolution on the true signal and then adding noise. The naïve solution is achieved by inverting the blur operator, G, resulting in a poor approximation to the ground truth.

The positive constant λ is called the regularization parameter and plays a fundamental role in the reconstruction, which we discuss below.

Now, we adopt the notation presented in [2, Chapter 6] and reformulate (1.7) in matrix notation, $x = V\Phi\Sigma^{-1}U^T$, using Φ to denote the diagonal matrix consisting of the diagonal elements $w_\lambda(s_i)$. The relationship between the regularization and perturbation errors can then be examined. Following the analysis presented in [2,

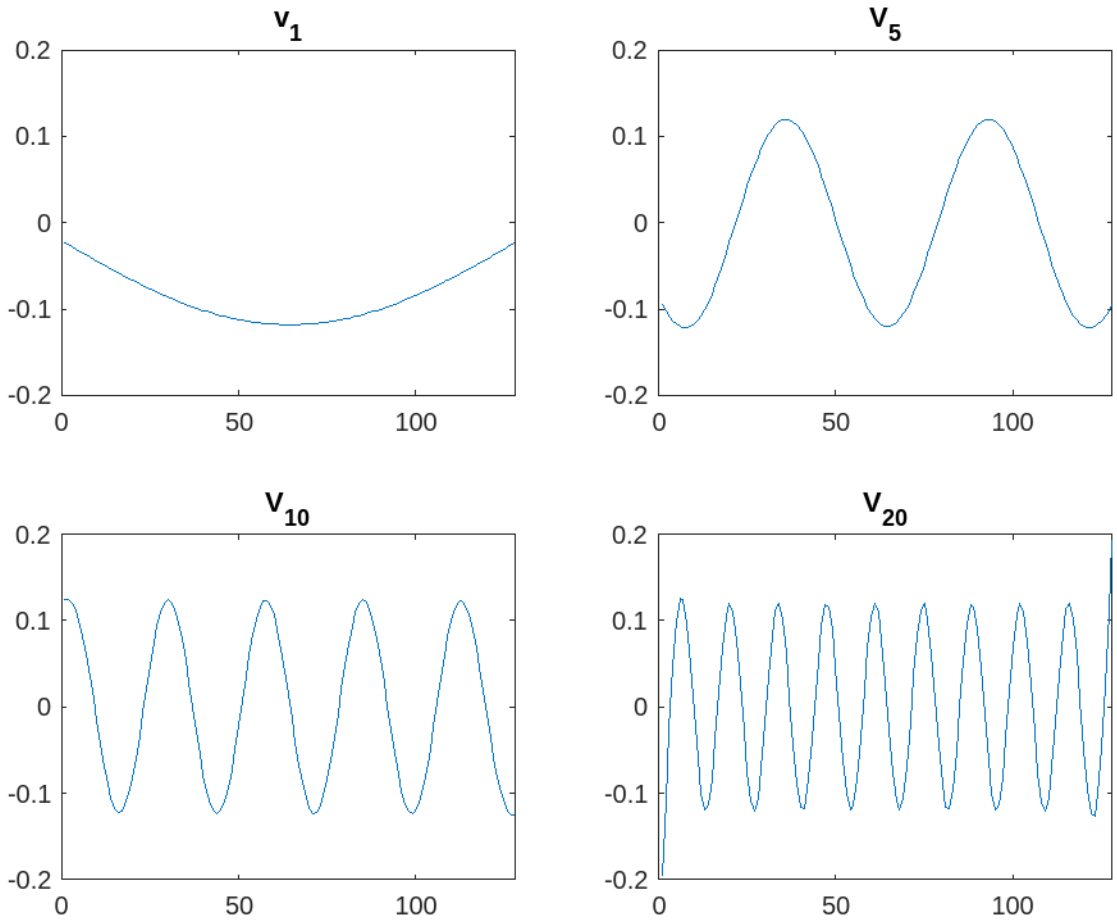

Figure 1.2: High-frequency behavior of singular vectors. The vectors v_i oscillate more as i increases.

p. 78], we write

$$\begin{aligned} x &= V\Phi\Sigma^{-1}U^T d \\ &= V\Phi\Sigma^{-1}U^T G(y)x_{true} + V\Phi\Sigma^{-1}U^T\epsilon_{noise} \\ &= V\Phi V^T x_{true} + V\Phi\Sigma^{-1}U^T\epsilon_{noise}. \end{aligned}$$

Then, we compare the true solution with the filtered solution,

$$\begin{aligned} x_{true} - x &= x_{true} - V\Phi V^T x_{true} - V\Phi\Sigma^{-1}U^T\epsilon_{noise} \\ &= (I - V\Phi V^T)x_{true} - V\Phi\Sigma^{-1}U^T\epsilon_{noise}. \end{aligned}$$

The authors in [2] refer to the first component $(I - V\Phi V^T)x_{true}$ as the regularization

error and the second component $V\Phi\Sigma^{-1}U^T\epsilon_{noise}$ as the perturbation error. The two different errors depend on Φ. The matrix Φ is determined by the regularization filter $w_\lambda(s_i)$. If too many filter factors $w_\lambda(s_i)$ are close to one, then Φ is closer to the identity matrix. Therefore, the regularization error is smaller, but the perturbation error is larger. If too many filter factors are close to zero, then Φ is close to the zero matrix. Similarly, the perturbation error is zero, but the regularization error increases. Picking an appropriate regularization parameter λ is necessary to balance the regularization and perturbation errors.

Moreover, the choice of the regularization parameter can be informed by the following analysis by the authors in [2, p. 73]. Using Taylor expansions, consider first the case when $s_i >> \lambda$:

$$w_\lambda(s_i) = \frac{1}{1+\lambda^2/s_i^2} = 1 - \frac{\lambda^2}{s_i^2} + \frac{1}{2}\frac{\lambda^4}{s_i^4} + \dots,$$

then, consider the case when $s_i << \lambda$:

$$w_\lambda(s_i) = \frac{s_i^2}{\lambda^2}\frac{1}{1+s_i^2/\lambda^2} = \frac{s_i^2}{\lambda^2}\Big(1 - \frac{s_i^2}{\lambda^2} + \frac{1}{2}\frac{s_i^4}{\lambda^4} + \dots\Big).$$

Next, it can be concluded that the Tikhonov filter factors satisfy the following:

$$w_\lambda(s_i) = \begin{cases} 1 - \left(\frac{\lambda}{s_i}\right)^2 + O\left(\left(\frac{\lambda}{s_i}\right)^4\right), & s_i >> \lambda, \\ \left(\frac{s_i}{\lambda}\right)^2 + O\left(\left(\frac{s_i}{\lambda}\right)^4\right), & s_i << \lambda. \end{cases}$$

Therefore, choosing $\lambda \in [s_m, s_1]$ results in $w_\lambda(s_i) \approx 1$ for small indices of i while $w_\lambda(s_i) \approx s_i^2/\lambda^2$ for large indices. The authors also demonstrate that, for a given λ, the point at which the filter factors change behavior is at the index for which $s_i \approx \lambda$. There is no reason to choose λ outside the interval $[s_m, s_1]$. Moreover, the regularization parameter in the Tikhonov filter is used to dampen the coefficients that correspond to the high-frequency singular vectors.

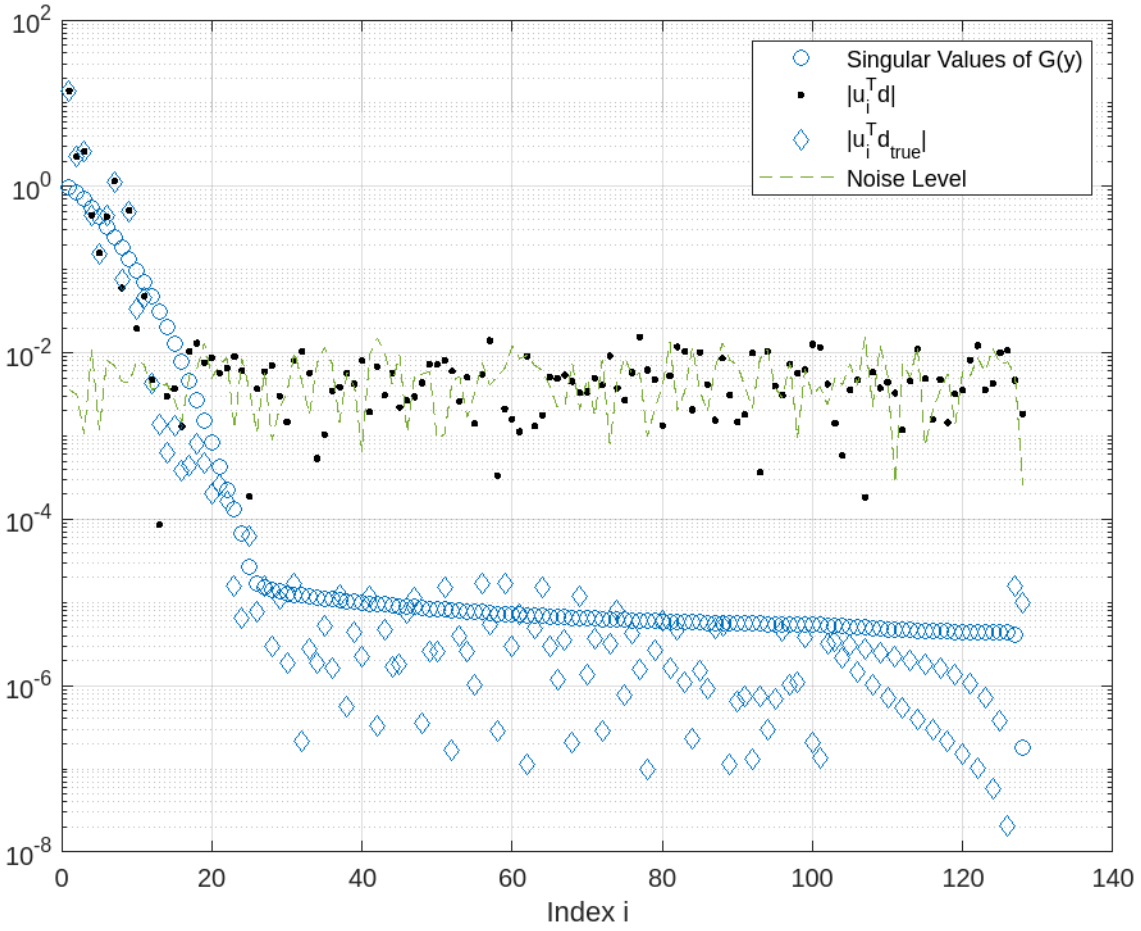

Figure 1.3: Illustration of the Spectral Coefficients Involved in the Error Analysis and the Singular Values.

Furthermore, a regularized approximation to the true signal can be computed because spectral filtering can suppress much of the inverted noise while keeping the regularization error small [2, p. 78]. The filtering can achieve this goal because the discrete Picard condition is satisfied: the coefficients $|u_i^T d|$ decay faster (on average) than the corresponding singular values [2, p. 69]. When the true solution is approximated, only the components corresponding to the coefficients $|u_i^T d|$ above the noise level should be included. An illustration of this is depicted in Figure 1.3. In the figure, the components corresponding to the signal without noise are denoted by $|u_i^T d_{true}|$.

It can also be seen from the figure that an appropriate regularization parameter for filtering out the noise would be $\lambda \approx 0.1$ since this value dampens all the components that correspond to the coefficients $|u_i^T d|$ which are within the noise level. Any value of $\lambda < 10^{-2}$ would result in poor regularization. Solving for x_{true} without regularization leads to a solution dominated by high-frequency behavior. So, regularization is an effective tool for filtering out errors, such as noise, that contaminate the solution.

In addition, the filtered solution in the spectral form (1.7) has the equivalent matrix-vector form,

$$x = (G(y)^T G(y) + \lambda^2 I)^{-1} G(y)^T d, \tag{1.8}$$

where x is the solution to the minimization problem:

$$\min_x \|G(y)x - d\|_2^2 + \lambda^2 \|x\|_2^2. \tag{1.9}$$

When the Tikhonov problem is formulated as it is in (1.9) it is said to be in standard form. The regularization used in this thesis is called generalized regularization and has the form:

$$\min_{x,y} \|G(y)x - d\|_2^2 + \lambda^2 \|Lx\|_2^2. \tag{1.10}$$

Here L is an $m \times m$ regularization matrix. The damped least squares problem or the general Tikhonov regularization problem in (1.10) can be equivalently written using the identity provided by

$$\|w\|_2^2 + \|z\|_2^2 = w^T w + z^T z = \begin{bmatrix} w \\ z \end{bmatrix}^T \begin{bmatrix} w \\ z \end{bmatrix} = \left\| \begin{bmatrix} w \\ z \end{bmatrix} \right\|_2^2, \tag{1.11}$$

to obtain the least squares problem:

$$\min_{x,y} \left\| \begin{pmatrix} G(y) \\ \lambda L \end{pmatrix} x - \begin{pmatrix} d \\ 0 \end{pmatrix} \right\|_2^2. \tag{1.12}$$

The advantage of realizing that (1.10) can be written in the form of (1.12) is that the latter formulation is better for developing algorithms [2, p. 91].

In this work, we assume that the solution x_{true} has zero boundary conditions, therefore, the matrix $G(y)$ has a Toeplitz structure:

$$G(y) = \begin{pmatrix} g_0 & g_{-1} & g_{-2} & \cdots & g_{-(m-1)} \\ g_1 & g_0 & g_{-1} & \cdots & g_{-(m-2)} \\ g_2 & g_1 & g_0 & \cdots & g_{-(m-3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{m-1} & g_{m-2} & g_{m-3} & \cdots & g_0 \end{pmatrix}.$$

The structure of $G(y)$ depends on the assumed boundary conditions [2, Chapter 4]; and, an algorithm for building a Toeplitz matrix given a discrete kernel is provided in [2, p. 118].

Although the aim is to find a solution that provides a good estimate for the true signal, it would also be advantageous to predict a good estimate for the true kernel. This thesis will review how the general variable projection method can be used in conjunction with the Gauss-Newton method to solve the semi-blind deconvolution problem with mixed Gaussian kernels, demonstrate how to find the expressions for the partial derivatives of the kernel mixture, which will be necessary for the Gauss-Newton iteration, and present the numerical results of implementing the algorithm for solving this problem.

Chapter 2

METHODS

2.1 The Generalized Variable Projection Method

The variable projection (VarPro) method was originally introduced by Golub and Pereyra in [3] for solving separable nonlinear least squares problems. It is often considered in applications where two sets of unknown parameters exist, where one set is dependent on the other and can be explicitly eliminated. In [4], Golub and Pereyra review 30 years of developments and applications of the variable projection method for solving separable nonlinear least squares problems. The problems range from applications in applied chemistry, neural networks, and biological imaging. The authors also mention that "it would be worthwhile to consider a variable projection algorithm for the regularization of ill-conditioned separable nonlinear least squares problems [4]." In [5], the authors introduced a regularized variable projection algorithm for the regularization in the standard form. In [6], the authors presented a regularized approach for large-scale inverse problems. This thesis will consider the generalized variable projection (GenVarPro) method proposed by the authors in [7]. They propose a modified VarPro method for regularization in the general form. The authors also propose an approach for reconstructing images in semi-blind deblurring problems. The GenVarPro method is now reviewed; it aims to solve the minimization problem given by (1.12).

For ease in notation, let

$$G_L(y) := \begin{pmatrix} G(y) \\ \lambda L \end{pmatrix} \qquad \text{and} \qquad \tilde{d} := \begin{pmatrix} d \\ 0 \end{pmatrix}.$$

Then, the minimization problem can be written in this notation as,

$$\min_{x,y} \|G_L(y)x - \tilde{d}\|_2^2. \tag{2.1}$$

It is important to notice that (2.1) is a linear least squares problem if the vector of parameters y is assumed to be known. The linear variable x can then be obtained in terms of y by solving the linear least squares problem. The solution is provided in closed form by

$$x = G_L^\dagger(y)\tilde{d}, \tag{2.2}$$

where $G_L^\dagger(y)$ is the Moore-Penrose generalized inverse of $G_L(y)$. Define $r(x,y) = G_L(y)x - \tilde{d}$. Then, the functional $\mathcal{F}(x,y) = \|r(x,y)\|_2^2$ is often referred to as the full functional or the original functional. Next, substituting (2.2) into (2.1) provides the following minimization problem:

$$\min_y \left\|G_L(y)G_L^\dagger(y)\tilde{d} - \tilde{d}\right\|_2^2. \tag{2.3}$$

Notice that the linear variable x has been eliminated and that a nonlinear least squares problem in terms of the parameters y is left. The variable projection method has eliminated the linear variable x by exploiting the separability inherent in the model. According to [3, 4], using the variable projection method, the matrix

$$\mathcal{P}_{G_L(y)}^\perp := I - G_L(y)G_L^\dagger(y)$$

is the projector on the orthogonal complement of the column space of $G_L(y)$. Then, the vector $r_2(y) = \mathcal{P}_{G_L(y)}^\perp \tilde{d}$ is called the variable projection of $\tilde{d}$. Furthermore, the functional $\tilde{\mathcal{F}}(y) = \|r_2(y)\|_2^2$ may be called the variable projection functional or may often be referred to as the reduced functional. Notice that the minimization of the reduced functional in (2.3) is a nonlinear least squares problem and can be solved using the Gauss-Newton method, which we explain below.

In short, the variable projection method consists of first minimizing the reduced functional $\tilde{\mathcal{F}}(y)$ and then using the optimal value obtained for y to solve for x in (2.2). The main advantage of this reduced functional is that the minimization always converges in fewer iterations than the full functional [4]. Moreover, the authors in [4] emphasize that using the variable projection method to exploit the separability of the problem is more powerful than methods that alternate between the two sets of variables.

2.2 The Gauss-Newton Method

The Gauss-Newton method is a quasi-Newton iterative method developed to solve nonlinear least squares problems [8]. Suppose the full functional is reduced to obtain the minimization problem provided by (2.3). Let $f(y) = G_L(y)G_L^{\dagger}(y)\tilde{d}$. Then, the resulting unconstrained nonlinear optimization problem is:

$$\min_{y} \|f(y) - \tilde{d}\|_2^2. \tag{2.4}$$

The Gauss-Newton method starts with an initial guess, $y^{(0)}$, and aims to obtain $y^{(k+1)}$ given some current iterate $y^{(k)}$. Consider that $f(y^{(k+1)})$ is approximated linearly by $f(y^{(k)}) + J(y^{(k)})\tilde{s}^{(k)}$ where $J(y^{(k)})$ is the Jacobian matrix of f and $\tilde{s}^{(k)}$ is the correction vector. Substituting this approximation into the optimization problem in (2.4), the linear least squares problem is obtained:

$$\min_{\tilde{s}^{(k)}} \|J(y^{(k)})\tilde{s}^{(k)} - \left(\tilde{d} - f(y^{(k)})\right)\|_2^2. \tag{2.5}$$

Then, solve for $\tilde{s}^{(k)}$ in (2.5) by utilizing the normal equations:

$$J(y^{(k)})^T J(y^{(k)})\tilde{s}^{(k)} = J(y^{(k)})^T(\tilde{d} - f(y^{(k)})). \tag{2.6}$$

Next set the update using a line search strategy $y^{(k+1)} = y^{(k)} + \alpha^{(k)}\tilde{s}^{(k)}$ where $\alpha^{(k)}$ is known as the step length.

In summary, the Gauss-Newton iteration as it pertains to GenVarPro can be described as follows: first, solve the linear least squares problem provided by (2.5); second, set $y^{(k+1)} = y^{(k)} + \alpha^{(k)}\tilde{s}^{(k)}$.

2.3 Computing the Jacobian

A key component of the Gauss-Newton iteration is the Jacobian matrix. Golub and Pereyra developed in [3] expressions for the derivatives of the reduced functional resulting in a formulation for the Jacobian. The authors in [7] refer to this formulation as the Golub-Pereyra Jacobian:

$$\mathcal{D}\mathcal{P}^{\perp}_{G_L(y)}d = (-\mathcal{P}^{\perp}_{G_L(y)}\mathcal{D}(G_L(y))G_L^{\dagger}(y) - (\mathcal{P}^{\perp}_{G_L(y)}\mathcal{D}(G_L(y))G_L^{\dagger}(y))^T)d,$$

where $\mathcal{D}$ denotes the Fréchet derivative. Various authors have proposed several different approximations of the Golub-Pereyra Jacobian. One simplification of the Golub-Pereyra Jacobian that has been proposed is the Kaufman simplification which has been shown to reduce cost and increase the speed of convergence [4]. The Kaufman simplification has the form:

$$\mathcal{D}\mathcal{P}^{\perp}_{G_L(y)}d \approx -\mathcal{P}^{\perp}_{G_L(y)}\mathcal{D}(G_L(y))G_L^{\dagger}(y)d.$$

Also, a simplification was proposed by Ruano, Jones, and Fleming [9], which is the Jacobian that is implemented in this thesis:

$$\mathcal{D}\mathcal{P}^{\perp}_{G_L(y)}d \approx -\mathcal{D}(G)G^{\dagger}(y)d.$$

This Jacobian will be referred to as RJF Jacobian. In [7], it is further described how the columns of the RJF Jacobian are defined by

$$J^{(i)}(:,j) = \frac{\partial G(y)}{\partial y_j}x^{(i)}, \quad \text{where} \quad y_1 = b_1,\ y_2 = b_2,\ y_3 = \sigma_1,\ y_4 = \sigma_2.$$

Here $\frac{\partial G(y)}{\partial_{y_j}}$ is the Toeplitz matrix built from the partial derivative of the kernel with respect to y_j. This then requires us to compute the expressions for the partial derivatives of the kernel function with respect to each parameter. In the following, it will be demonstrated how expressions for the partial derivatives of the kernel can be determined. Suppose the kernel is given by

$$\kappa(r; y) = b_1\hat{\kappa}_1(r; \sigma_1) + b_2\hat{\kappa}_2(r; \sigma_2),$$

where

$$\hat{\kappa}_j(r; \sigma_j) = \frac{\exp\left(-r^2/(2\sigma_j^2)\right)}{\int \exp\left(-r^2/(2\sigma_j^2)\right) dr} \text{ for } j = 1, 2.$$

Then, the partial derivatives are given by

$$\partial_{b_1}\kappa = \hat{\kappa}_1, \quad \partial_{b_2}\kappa = \hat{\kappa}_2, \quad \partial_{\sigma_1}\kappa = b_1\partial_{\sigma_1}\hat{\kappa}_1, \quad \partial_{\sigma_2}\kappa = b_2\partial_{\sigma_2}\hat{\kappa}_2.$$

Recall the quotient rule:

$$\left(\frac{f(x)}{g(x)}\right)' = \frac{f'(x)g(x) - f(x)g'(x)}{[g(x)]^2}.$$

Then,

$$\partial_{\sigma_j}\hat{\kappa}_i = \frac{\left(\partial_{\sigma_j} \exp\left(-r^2/(2\sigma_j^2)\right)\right)\left(\int \exp\left(-r^2/(2\sigma_j^2)\right) dr\right)}{\left(\int \exp\left(-r^2/(2\sigma_j^2)\right) dr\right)^2} - \frac{\left(\exp\left(-r^2/(2\sigma_j^2)\right)\right)\left(\partial_{\sigma_j} \int \exp\left(-r^2/(2\sigma_j^2)\right) dr\right)}{\left(\int \exp\left(-r^2/(2\sigma_j^2)\right) dr\right)^2}.$$

Let $\kappa_j = \exp\left(-r^2/(2\sigma_j^2)\right)$. Then, simplifying,

$$\partial_{\sigma_j}\hat{\kappa}_j = \frac{\partial_{\sigma_j}\kappa_j}{\int \kappa_j\, dr} - \frac{\kappa_j \int \partial_{\sigma_j}\kappa_j\, dr}{\left(\int \kappa_j\, dr\right)^2}.$$

So the partial derivatives of the kernel function can be written as

$$\partial_{b_j}\kappa = \hat{\kappa}_j = \frac{\exp\left(-r^2/(2\sigma_j^2)\right)}{\int \exp\left(-r^2/(2\sigma_j^2)\right) dr} \tag{2.7}$$

and

$$\partial_{\sigma_j}\kappa = b_j\left(\frac{\partial_{\sigma_j}\kappa_j}{\int \kappa_j\, dr} - \frac{\kappa_j \int \partial_{\sigma_j}\kappa_j\, dr}{\left(\int \kappa_j\, dr\right)^2}\right). \tag{2.8}$$

After discretizing (2.7) and (2.8) and building the corresponding Toeplitz matrices one obtains the RJF Jacobian matrix required for the Gauss-Newton iteration.

2.4 Algorithm

The GenVarPro method implemented in this thesis is a Gauss-Newton method (with line search strategy) on the reduced functional with a simplified Jacobian matrix called the RJF Jacobian.

In the following, the algorithm used in this thesis provided by Algorithm 1 is discussed. In line 2, the linear variable $x^{(i)}$ is updated by solving the linear least squares problem with iterate $y^{(i)}$, which is done by utilizing Equation (2.2). Then, in line 3, the residual of the variable projection functional is updated. Note that line 3 is essentially in reduced form since the iterate $x^{(i)}$ has been solved in terms of $y^{(i)}$ in the manner mentioned in (2.2). In line 4, the RJF Jacobian is updated with the new iterate $y^{(i)}$ and then computed. Finally, a Gauss-Newton iteration is performed on lines 5 and 6, thus updating the iterate $y^{(i)}$.

The algorithm requires an initial guess of the parameters $y^{(0)} = (b_1^{(0)}, b_2^{(0)}, \sigma_1^{(0)}, \sigma_2^{(0)})$, the regularization matrix L, an appropriate value for the regularization parameter λ, and the blurred signal denoted by d. For the implementation used in this thesis, the regularization matrix L is chosen to be a discrete approximation to the first-order

Algorithm 1 GenVarPro Method with Gauss-Newton Iteration

Require: $G(y^{(0)})$, L, λ, and d.

1: **for** $i = 0, 1, \ldots$ **do**

2: $\quad x^{(i)} = \arg\min_x \|G(y^{(i)})x - d\|_2^2 + \lambda^2 \|Lx\|_2^2$

3: $\quad r^{(i)} = G(y^{(i)})x^{(i)} - d$

4: $\quad$ Compute/Update the Jacobian matrix $J_r^{(i)}$

5: $\quad \tilde{s}^{(i)} = \arg\min_{\tilde{s}} \left\| J_r^{(i)} \tilde{s} + r^{(i)} \right\|_2^2$

6: $\quad y^{(i+1)} = y^{(i)} + \alpha^{(i)} \tilde{s}^{(i)}$

derivative operator, which is given by the difference matrix:

$$L = \begin{pmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ & & & -1 \end{pmatrix}.$$

The motivation for using this regularization matrix is that differentiation is known to magnify high-frequency components, so penalizing high-frequency components leads to a smoother solution [2, p. 92].

The implementation of Algorithm 1 used in this thesis contains three stopping conditions. The first stopping condition is based on the size of the full functional. A tolerance is provided, and the algorithm stops when the size of the full functional becomes smaller than the tolerance.

The second stopping condition used in this implementation is a value that denotes the maximum number of iterations. This condition is there in case the algorithm does not reach the first stopping condition. In some cases, the algorithm may fail to reach the first stopping condition yet still provide an approximate solution to the true signal.

Finally, the third stopping condition is useful for debugging purposes. This stop-

ping condition is based on the conditioning of the matrix $J(y)^T J(y)$. It has been observed, in the implementation of the algorithm used in this thesis, that in some cases the matrix $J(y)^T J(y)$ becomes singular to working precision. This results in the normal equations of (2.6) being numerically unsolvable.

The regularization parameter, denoted by λ, is provided by the user of the algorithm and remains fixed for each iteration. Different values for the regularization parameters are tested within an interval $[s_m, s_1]$ where s_m and s_1 are the smallest and largest singular values of the matrix $G(y^{(0)})$. Then, each solution is graphed and visually inspected, and the solutions that appear to approximate the true solution well are selected.

Chapter 3

NUMERICAL RESULTS

In this thesis chapter, the numerical results of implementing the GenVarPro method with the RJF Jacobian are presented. Only the case of zero boundary conditions is considered. The results are obtained by creating a synthetic signal representing the true signal, blurring the true signal with the true blurring operator, and then adding noise to obtain what is called the blurred signal. With this blurred signal, the GenVarPro method provided by Algorithm 1 is used to finding a solution that is called the deblurred signal. The aim is that the deblurred signal approximates the true signal accurately, given the blurred signal.

When deciding on a regularization parameter, a value is chosen by inspecting a chart similar to Figure 1.3, running the algorithm with this guess, and then visually inspecting the results.

In Figure 3.2, the algorithm is run on a smooth signal. The true signal is seen in Figure 3.2 as the solid blue curve. Then, the blurred signal is produced using the true kernel provided in Figure 3.1. The true parameters are $y_{true} = (0.4, 0.6, 6, 15)$ and the initial guess is $y^{(0)} = (0.5, 0.5, 8, 17)$. The blurred signal that is obtained is depicted as the solid green line in Figure 3.2.

The GenVarPro method is run with the initial guess and a fixed regularization parameter of approximately 0.12. The deblurred signal is shown in Figure 3.2 as the dotted red circles. The deblurred signal is obtained after running the GenVarPro method for 20 iterations, it stopped once it reached the maximum iteration count. See that the deblurred signal appears to approximate the true signal reasonably well upon inspecting the graph. This appears to be a satisfactory result. So given a

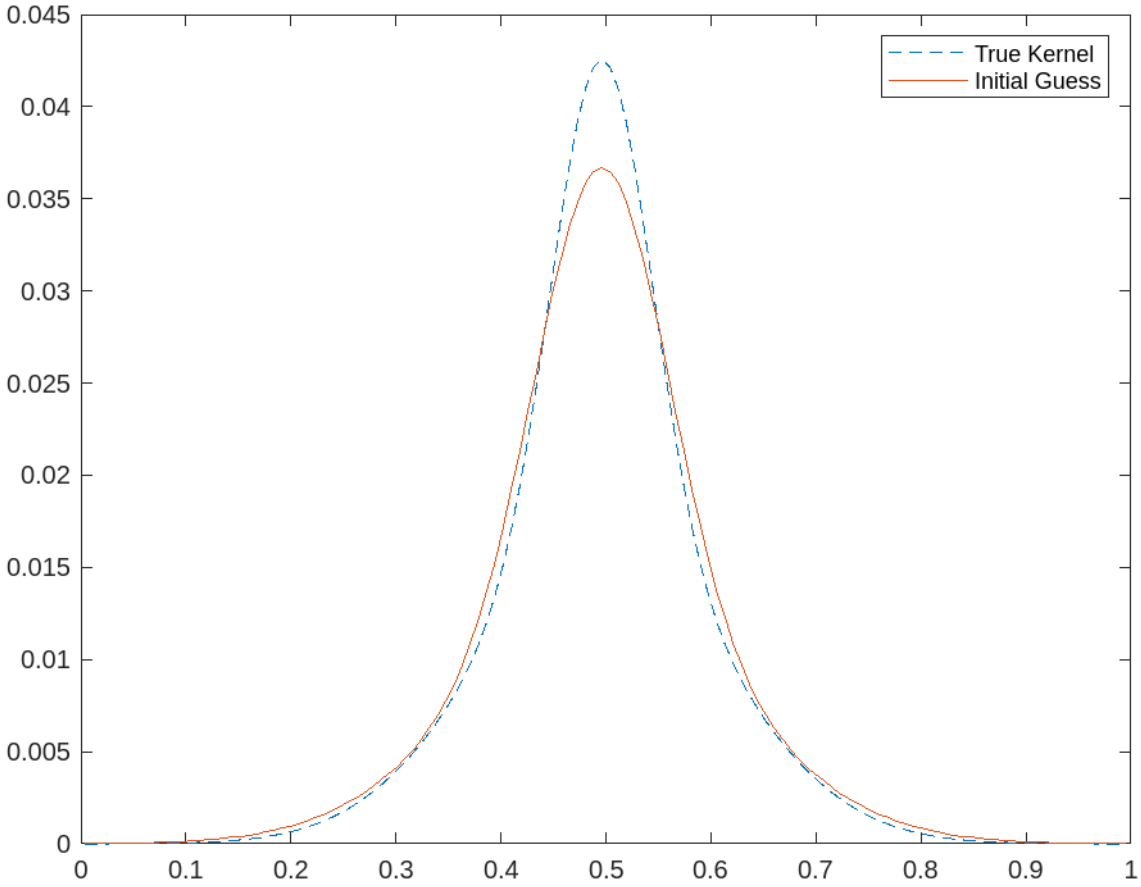

Figure 3.1: Example of the true kernel and a corresponding initial guess. The true parameters are $y_{true} = (0.4, 0.6, 6, 15)$ and the initial guess is $y^{(0)} = (0.5, 0.5, 8, 17)$.

good initial guess and an appropriate regularization parameter the GenVarPro can effectively approximate the true signal even with a fixed value of λ.

In the following, the method is tested on different signals, and the errors in the numerical results are investigated. The algorithm is used on a signal with edges, true parameters $y_{true} = (0.4, 0.6, 3, 10)$, and initial guess $y^{(0)} = (0.5, 0.5, 3.5, 10.5)$. It can be observed from Figure 3.3 that the quality of the deblurred signal depends on an appropriate value of the regularization parameter. See that the deblurred signal is noisy for a small value of the regularization parameter. Also, see that if the value of the regularization parameter is too large, the deblurred signal is very smooth.

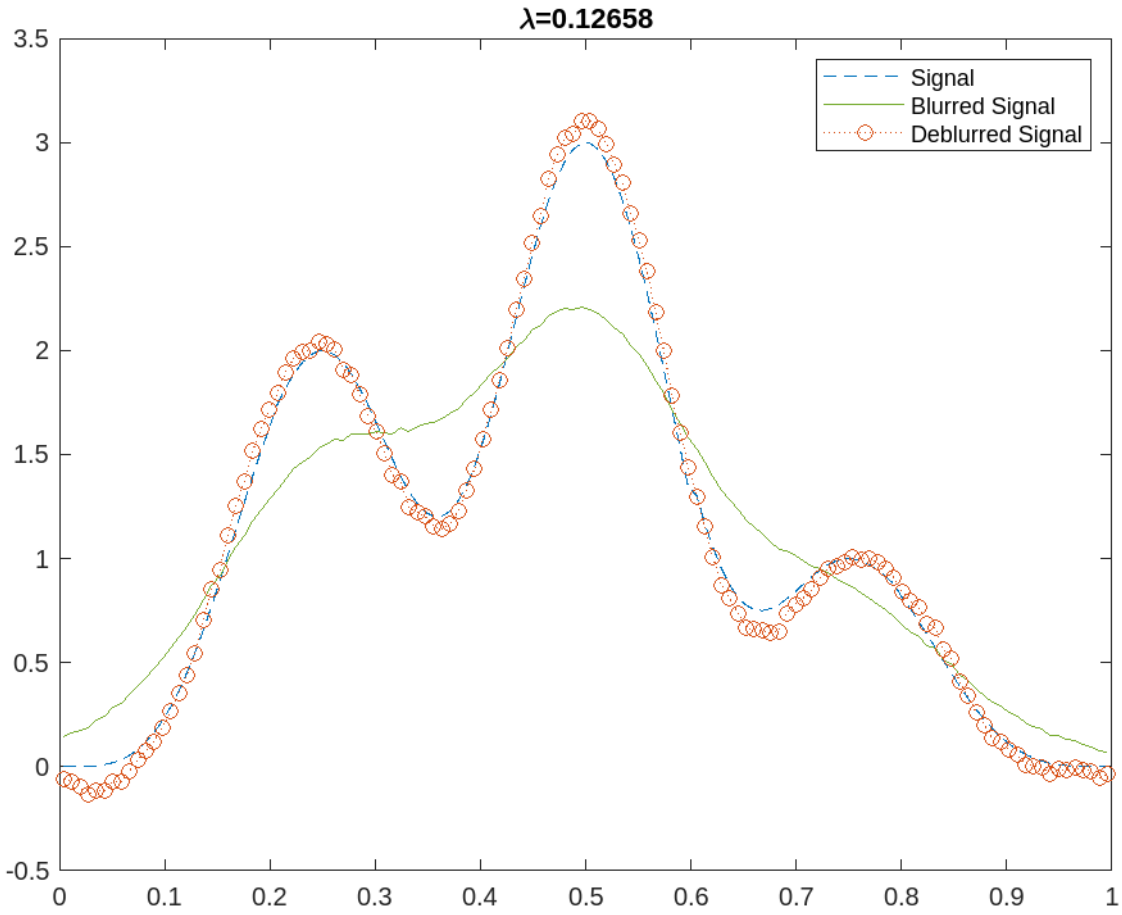

Figure 3.2: The solution obtained by using the GenVarPro method on a smooth signal. The regularization parameter provided is approximately 0.12.

In either case, it can be seen from inspecting the graph that the deblurred signal does not provide a good solution for regularization parameters that are too small or too large. According to [2, p. 91], this behavior is expected if the regularization parameter is too large. Too much emphasis is put on dampening the noise in the solution, which causes a loss of detail. In contrast, if the regularization parameter is too small, the solution is influenced too much by the noise in the blurred signal.

In Figure 3.4, a different signal with edges is chosen. In this case, the signal is strictly non-negative. The stopping condition for the tolerance of the full functional

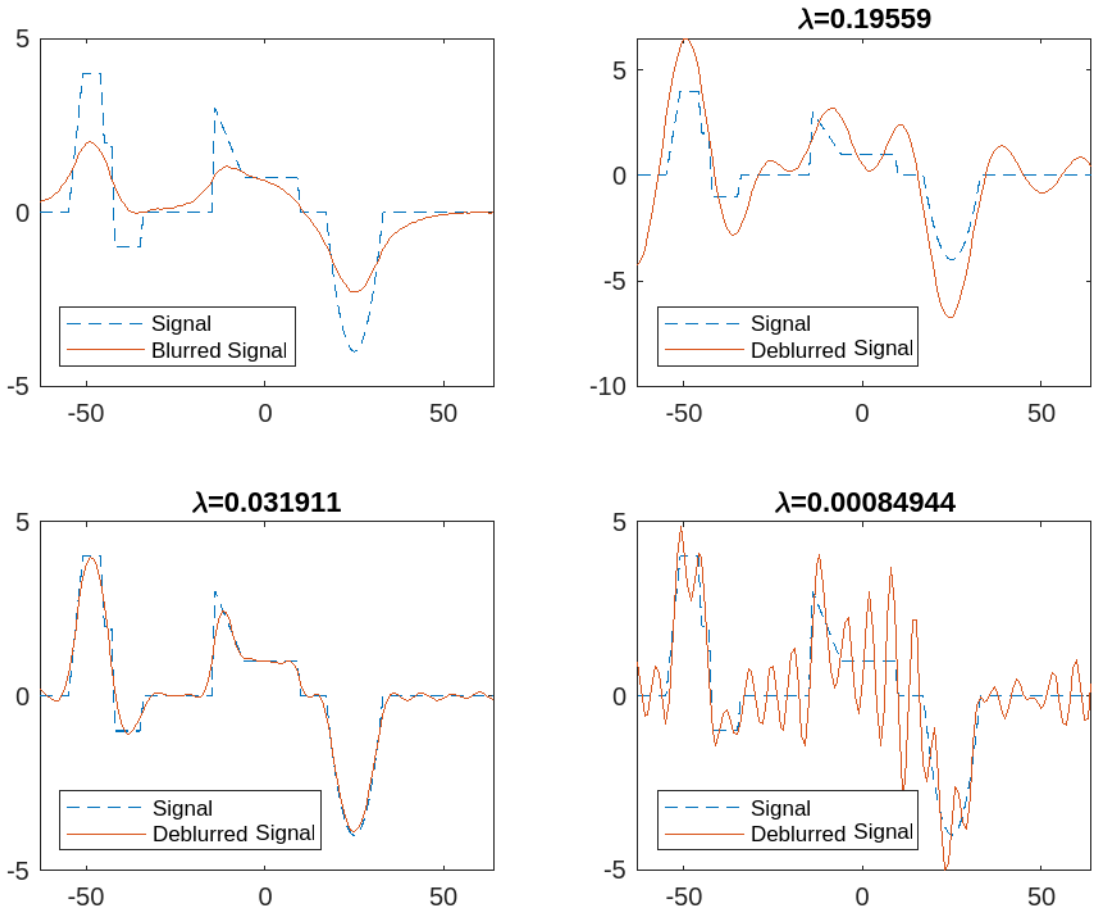

Figure 3.3: The variable projection method for a signal that contains edged sections and smooth sections with different values for the regularization parameter. The upper left shows the true solution x_{true} and the signal d. The rest of the figures show the approximations of the signal and the true solution for comparison.

is chosen to be too small to be triggered so that the method can run for 50 iterations in order to observe convergence behavior. The upper left subplot shows the true signal as a dashed blue line, the blurred signal as a solid red line, and the deblurred signal as red dots. The value of the regularization parameter is approximately 0.019. The upper right subplot shows the value of the full functional for each iteration. The lower left subplot shows the relative error of the deblurred signal, which is given by

$$\mathrm{ERR}(x^{(i)}) = \frac{\|x_{true} - x^{(i)}\|_2}{\|x_{true}\|_2},$$

where $x^{(i)}$ is the deblurred signal at the ith iterate. The lower right subplot shows

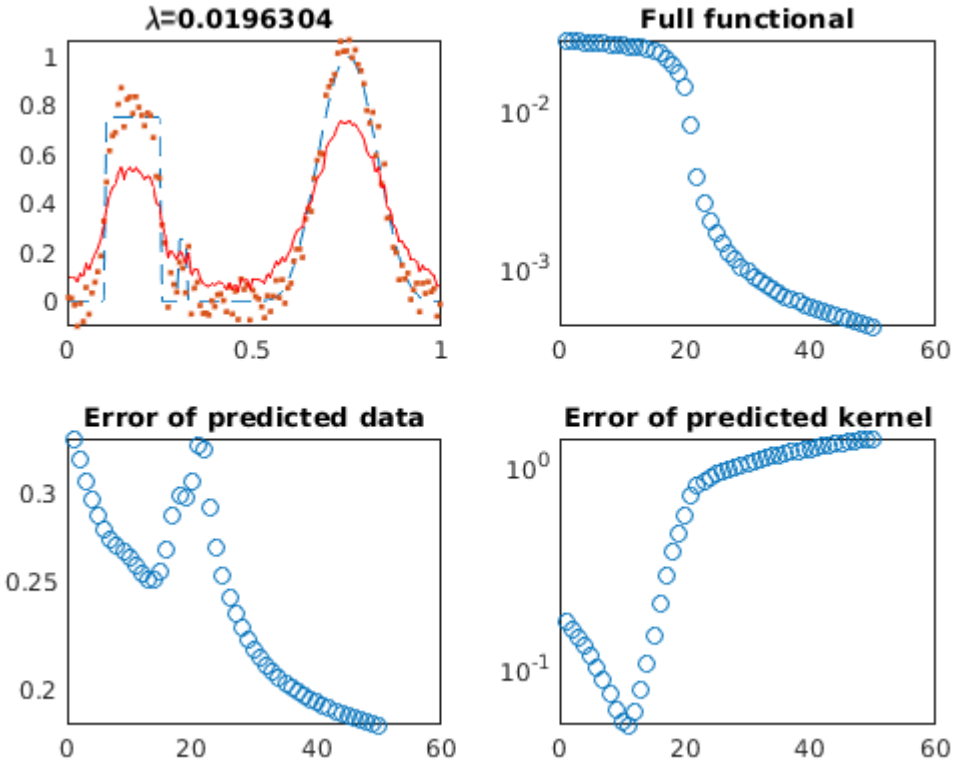

Figure 3.4: The variable projection method for a nonnegative signal with both smooth and edged sections and with full functional and errors for deblurred signal and predicted kernel per iteration.

the error between the true kernel and the predicted kernel. The predicted kernel is the value of the kernel for each iterate $y^{(i)}$, and its relative error is defined by

$$\mathrm{ERR}(g(s;y^{(i)})) = \frac{\|g(s;y_{true}) - g(s;y^{(i)})\|_2}{\|g(s;y_{true})\|_2}.$$

See from Figure 3.4 that the full functional appears to be getting smaller for each iteration. Also, notice the semi-convergence behavior of the predicted kernel and that even though the predicted kernel begins to diverge from the true kernel, the error of the deblurred signal eventually decreases. The decreasing values of the full functional are expected since the aim was to minimize it. This suggests that the method is indeed minimizing the full functional. However, the semi-convergence of the predicted kernel

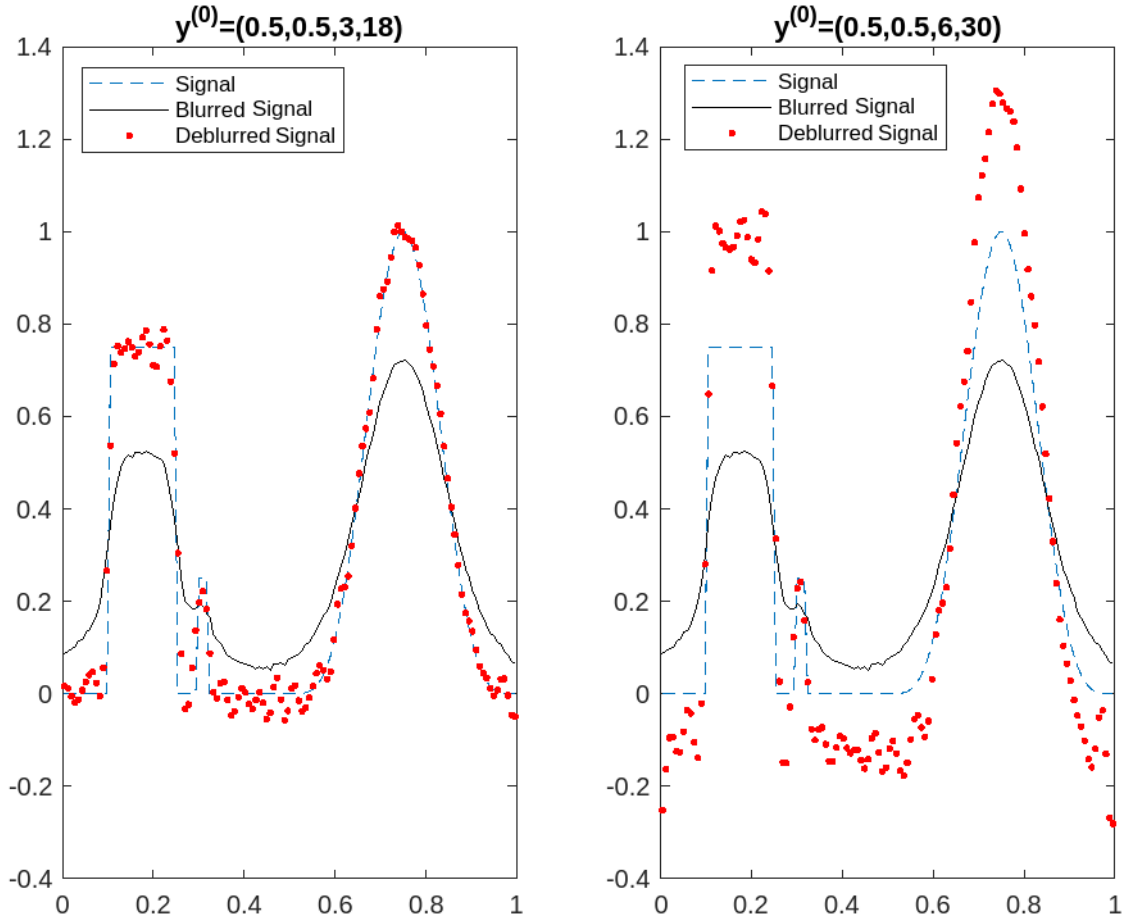

Figure 3.5: The variable projection with different initial guesses of kernel parameters.

requires more investigation. It appears that the implementation of the method in this thesis provides a decent prediction of the true signal but not a good prediction of the true kernel. In Figure 3.5, the predicted solution is shown for two initial guesses of the parameters for the kernel. The subplot on the left has an initial guess of $y^{(0)} = (0.5, 0.5, 3, 18)$. The subplot on the right has an initial guess $y^{(0)} = (0.5, 0.5, 6, 30)$. The true parameters for each case are given by $y_{true} = (0.4.0.6, 2, 15)$. Also, for each subplot, the regularization parameter is the same and is approximately 0.0192. Both methods ran for a total of 50 iterations. The figure shows that the initial guess, which is closer to the true parameters, approximates the true signal better than the initial guess that is further away.

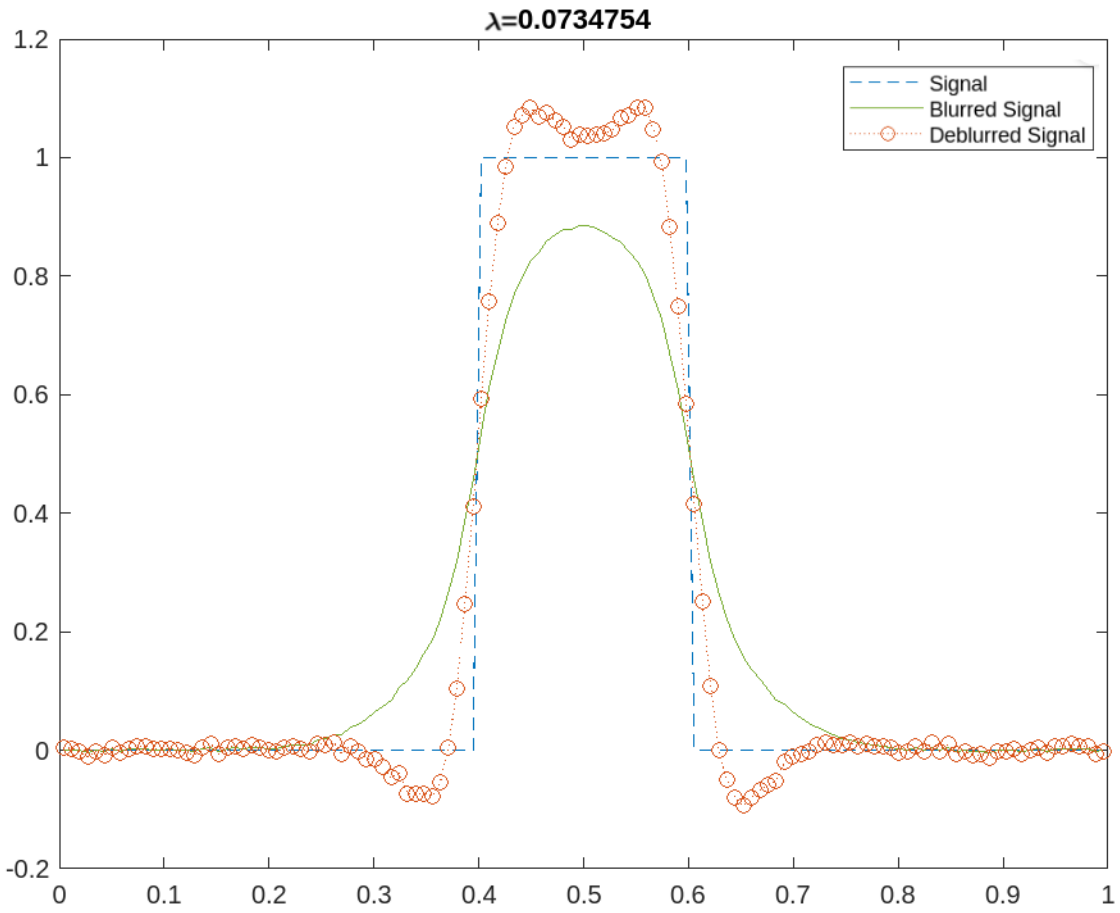

Figure 3.6: The approximated solution to a signal that is a single edge.

For a signal that contains a single edge, it can be observed in Figure 3.6, that the approximate solution contains a decent amount of oscillation. According to [1], the total variation functional can be used to penalize this high oscillatory behavior. These oscillatory artifacts are not present in Figure 3.2 because the true signal is smooth. In [1, Chapter 8], the author describes how the total variation functional is better for "blocky" signals. These observations illustrate how the implementation in this thesis may work well for smooth signals but not as well for signals that have straight-edged sections. Next, how the parameters of the iterate $y^{(i)}$ evolve over time is investigated. In Figure 3.7, each parameter is plotted for b_1, b_2, σ_3, and σ_4 over each iteration. The horizontal line that can be seen in each subplot is the value of

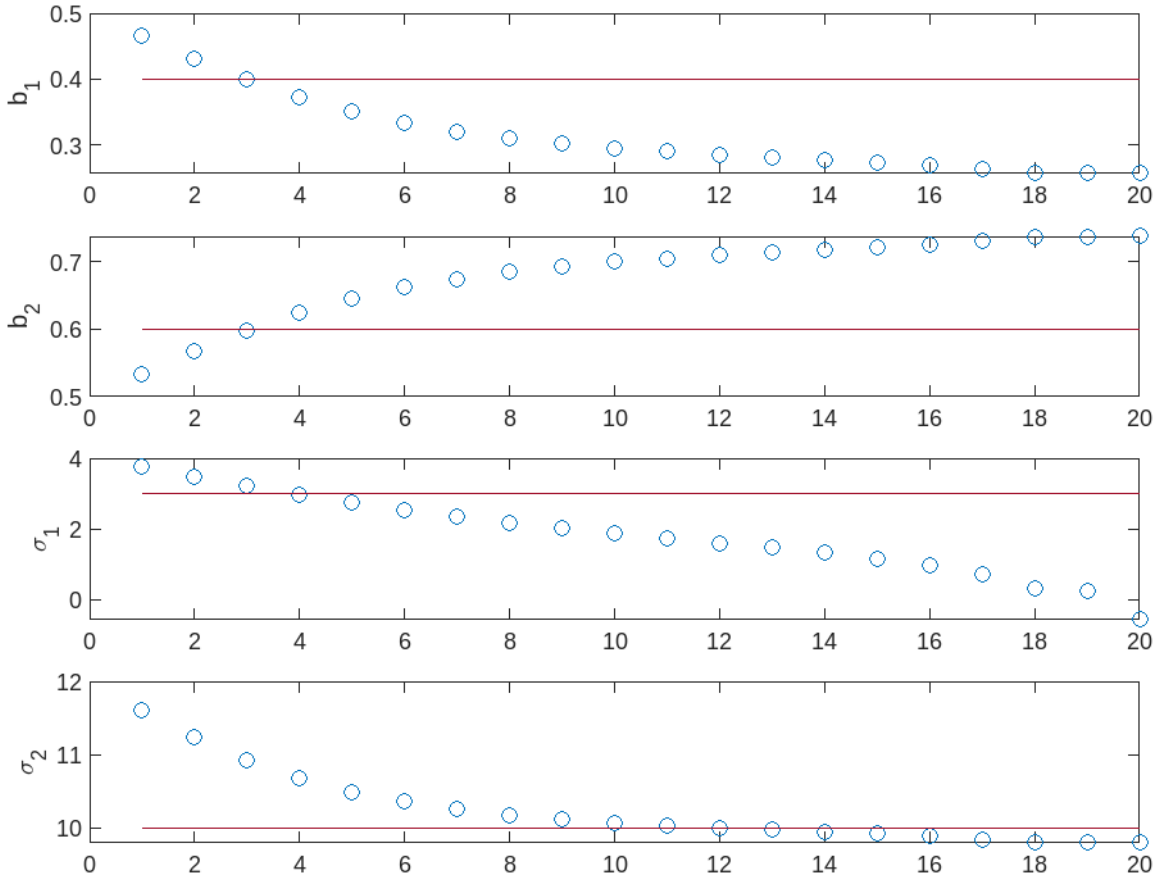

Figure 3.7: Evolution of parameters over each iteration. The evolution of the parameter corresponding to the approximate solution of the signal is presented in Figure 3.6.

the true parameters of the kernel mixture y_{true}. Notice that each parameter does not converge to its corresponding true parameter. Also, see how the parameters b_1 and b_2 behave when b_1 goes down, and b_2 goes up. The parameters σ_1 and σ_2 appear to be getting smaller per iteration. In particular, notice that σ_1 approaches zero and then becomes negative. In Figure 3.8, the predicted kernel corresponding to the approximate solution of the edge signal provided by Figure 3.6 is shown. See that even though the initial guess is close to the true kernel, the predicted kernel does not approximate the true kernel well. Also, notice the "spike" in the center of the predicted kernel near zero. The cause of this artifact needs to be investigated

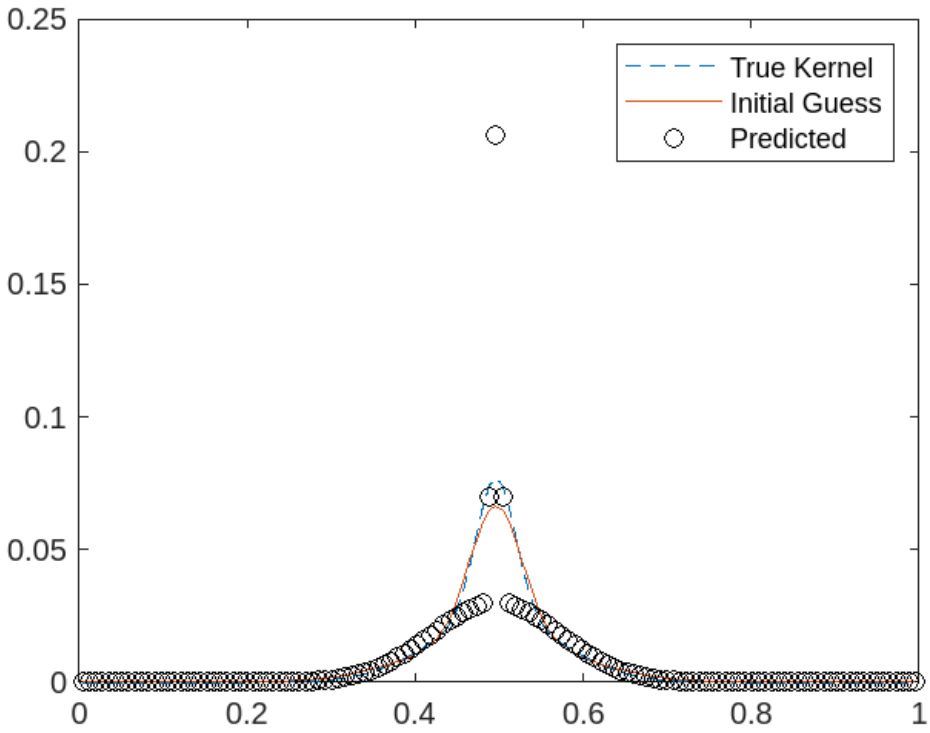

Figure 3.8: The predicted kernel corresponding to the approximate solution of the signal presented in Figure 3.6.

further, however, it seems that the "spike" may be formed because the parameter σ_1 is near zero. This behavior might be eliminated by incorporating constraints onto the parameters in the kernel mixture.

In short, examples pertaining to how GenVarPro performs on different signals have been provided. The method works well for an appropriate choice of the regularization parameter and a good initial guess of the parameter y_{true}. The method appears to behave better for smooth signals than it does for "blocky" signals. However, the method does not predict the true kernel well.

Chapter 4

CONCLUSION

In summary, this thesis has reviewed a semi-blind deconvolution problem with mixed Gaussian kernels. It has explored using the GenVarPro method for reducing the problem into a form that can be solved using the Gauss-Newton method. Furthermore, this thesis illustrates how to compute the required partial derivatives of the mixed Gaussian kernel, and it shows the numerical results of implementing the GenVarPro method with a simplified Jacobian. This thesis provides figures to illustrate the behavior of the method for various signals, regularization parameters, and initial guesses of the parameters for the true kernel.

Future work might include extending the semi-blind model to include more Gaussian functions, mixing the Gaussian kernels with other kernel functions, modifying the regularization to use total variation, incorporating constraints on the parameters of the kernel mixture, assuming other boundary conditions of the true signal, extending the model to be used on images, and testing the model against real-world data. There are also improvements to be made in implementing methods for appropriately choosing the regularization parameter. The current implementation has a fixed regularization parameter and relies on the user to experiment with various values which is inefficient and time-consuming. In [2, Chapter 6], the authors discuss parameter choice methods such as the discrepancy principle, generalized cross validation, and the L-curve criterion. The robustness and effectiveness of the current implementation may be increased by using a parameter choice method such as the generalized cross validation for each time the parameters to the kernel mixture are updated. Furthermore, future work could include exploring different techniques for normalizing the

kernel mixture. In this implementation, each Gaussian function has been normalized independently using the sum of the corresponding Gaussian, however, it could be possible to normalize each function using a normalization constant or by normalizing the entire kernel rather than each individual Gaussian.

In conclusion, for appropriate values of the regularization parameter and a good initial guess of the parameters in the true kernel, the GenVarPro method provides a decent approximate solution. The method does not approximate signals with edges as well as it does smooth signals. The method exhibits a semi-convergence behavior towards the true kernel which requires further investigation.

REFERENCES

[1] C. R. Vogel, *Computational methods for inverse problems*. SIAM, 2002.

[2] P. C. Hansen, J. G. Nagy, and D. P. O'leary, *Deblurring images: matrices, spectra, and filtering*. SIAM, 2006.

[3] G. H. Golub and V. Pereyra, "The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate," *SIAM Journal on numerical analysis*, vol. 10, no. 2, pp. 413–432, 1973.

[4] G. Golub and V. Pereyra, "Separable nonlinear least squares: the variable projection method and its applications," *Inverse problems*, vol. 19, no. 2, p. R1, 2003.

[5] G. Chen, M. Gan, C. P. Chen, and H. Li, "A regularized variable projection algorithm for separable nonlinear least-squares problems," *IEEE Trans. Automat. Control*, vol. 64, no. 2, pp. 526–537, 2018.

[6] J. Chung and J. G. Nagy, "An efficient iterative approach for large-scale separable nonlinear inverse problems," *SIAM J. Sci. Comput.*, vol. 31, no. 6, pp. 4654–4674, 2010.

[7] M. I. Español and M. Pasha, "Variable projection methods for separable nonlinear inverse problems with general-form Tikhonov regularization," *Inverse Problems*, 2023. [Online]. Available: http://iopscience.iop.org/article/10.1088/1361-6420/acdd1b

[8] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[9] A. E. B. Ruano, D. I. Jones, and P. J. Fleming, "A new formulation of the learning problem of a neural network controller," in *[1991] Proceedings of the 30th IEEE Conference on Decision and Control*. IEEE, 1991, pp. 865–866.