

A STATISTICAL METHOD FOR REGULARIZING NONLINEAR INVERSE PROBLEMS

by

Chad Clifton Hammerquist

A thesis

submitted in partial fulfillment

of the requirements for the degree of

Master of Science in Mathematics

Boise State University

August 2012

© 2012

Chad Clifton Hammerquist

ALL RIGHTS RESERVED

BOISE STATE UNIVERSITY GRADUATE COLLEGE

DEFENSE COMMITTEE AND FINAL READING APPROVALS

of the thesis submitted by

Chad Hammerquist

Thesis Title: A statistical method for regularizing nonlinear inverse problems

Date of Final Oral Examination: 09 May 2012

The following individuals read and discussed the thesis submitted by student Chad Hammerquist, and they evaluated his presentation and response to questions during the final oral examination. They found that the student passed the final oral examination.

Jodi Mead, Ph.D.

Chair, Supervisory Committee

Grady Wright, Ph.D,

Member, Supervisory Committee

Dr. Paul Micheals, Ph.D.,PE

Member, Supervisory Committee

The final reading approval of the thesis was granted by Jodi Mead, Ph.D., Chair, Supervisory Committee. The thesis was approved for the Graduate College by John R. Pelton, Ph.D., Dean of the Graduate College.

ABSTRACT

Inverse problems are typically ill-posed or ill-conditioned and require regularization. Tikhonov regularization is a popular approach and it requires an additional parameter called the regularization parameter that has to be estimated. The χ^2 method introduced by Mead in [8] uses the χ^2 distribution of the Tikhonov functional for linear inverse problems to estimate the regularization parameter. However, for nonlinear inverse problems the distribution of the Tikhonov functional is not known. In this thesis, we extend the χ^2 method to nonlinear problems through the use of Gauss Newton iterations and also with Levenberg Marquardt iterations. We derive approximate χ^2 distributions for the quadratic functionals that arise in Gauss Newton and Levenberg Marquardt iterations. The approach is illustrated on two ill-posed nonlinear inverse problems: a nonlinear cross-well tomography problem and a subsurface electrical conductivity estimation problem. We numerically test the validity of assumptions in this approach by demonstrating that the theoretical χ^2 distributions agree closely with actual distributions. The nonlinear χ^2 method is implemented in two algorithms, based on Gauss Newton and the Levenberg Marquardt methods, that dynamically estimate the regularization parameter using χ^2 tests. We compare parameter estimates from the nonlinear χ^2 method with estimates found using Occams inversion and the discrepancy principle on the cross-well tomography problem and on the subsurface electrical conductivity estimation problem. The χ^2 method is shown to provide similar parameter estimates to estimates found using the discrepancy principle and is computationally less expensive. In addition, the χ^2 method provided much better parameter estimates than Occams Inversion.

TABLE OF CONTENTS

ABSTRACT	iii
LIST OF TABLES	vi
LIST OF FIGURES	vii
1 INTRODUCTION	1
1.1 Inverse Problems	1
1.1.1 Formulation of the Problem	1
1.2 Ill-Posed Problems	3
2 LEAST SQUARES	6
2.1 Ordinary Least Squares (OLS)	6
2.2 Nonlinear Least Squares	7
2.2.1 Gauss-Newton Method and Levenberg-Marquardt	9
2.3 Generalized Least Squares	12
3 REGULARIZATION AND THE χ^2 METHOD	16
3.1 Choice of Regularization Parameter	17
3.1.1 Nonlinear Regularization	17
3.1.2 Statistical Framework	18
3.1.3 Linear χ^2 Method	20
3.2 χ^2 Tests for Gauss-Newton Method	21

3.2.1	χ^2 Tests for Levenberg-Marquardt Method	26
3.3	Nonlinear χ^2 Method	29
3.3.1	Numerical Implementation of Algorithms	32
4	APPLICATION, SOLUTIONS, AND NUMERICAL EXPERIMENTS	34
4.1	Nonlinear Cross-Well Tomography	35
4.1.1	Numerical Experiments	36
4.1.2	Inversion Results	38
4.2	Subsurface Electrical Conductivity Estimation	42
4.2.1	Numerical Experiments	43
4.2.2	Inversion Results	44
5	CONCLUSIONS AND FUTURE WORK	48
A	ADDITIONAL THEOREMS	52

LIST OF TABLES

4.1	Comparison of discrepancy principle to χ^2 method on the cross-well tomography problem, $\mu = \text{mean}(\ x_M - x_{true}\ /\ x_{true}\)$, $\sigma = \text{sqr}t(\text{var}(\ x_M - x_{true}\ /\ x_{true}\))$	41
4.2	Comparison of the χ^2 method to Occam's inversion for the estimation of subsurface conductivities, $\mu = \text{mean}(\ x_M - x_{true}\ /\ x_{true}\)$, $\sigma = \text{sqr}t(\text{var}(\ x_M - x_{true}\ /\ x_{true}\))$	47

LIST OF FIGURES

1.1	Contours of $\ d - F(x)\ _2^2$ for two linear inverse problems. Left: Well-posed problem. Right: Ill-posed problem.	4
4.1	The setup of the cross-well tomography problem. (Left) Shown here is the true velocity model(m/s). (Right) The ray paths crossing through region of interest (background is faded to make the ray paths more clear).	35
4.2	Histograms of $\tilde{\mathcal{J}}_k(x_{k+1})$. (Left): $\tilde{\mathcal{J}}_k$ with $L = \mathbf{I}$, (Right): $\tilde{\mathcal{J}}_k$ with $L = \tilde{\Delta}$. The mean of the sample is shown as the middle tick, and χ_{64}^2 probability density function is shown as the solid blue line.	38
4.3	Solutions found for the tomography problem with $L = \tilde{\Delta}$ (Left) Solution found using the discrepancy principle. (Right) Solution found with Algorithm 1.	40
4.4	Solutions found for the tomography problem when $L = \mathbf{I}$. (Left) Solution found using the discrepancy principle. (Right) Solution found with Algorithm 1.	40
4.5	A representation of the soil-conductivity estimation. The instrument depicted in the top of image represents a ground conductivity meter creating a time-varying electromagnetic field in the layered earth beneath.	42

4.6	Histograms of $\tilde{\mathcal{J}}_k(x_{k+1})$. (Left) $\tilde{\mathcal{J}}_k$ with $L = I$, (Right) $\tilde{\mathcal{J}}_k$ with $L = \tilde{D}^{(2)}$. The mean of the sample is shown as the middle tick, and the χ^2_{18}, χ^2_{16} density functions are shown as the solid blue line.	44
4.7	(Left) The unregularized solution. (Right) The solution found with Occam's inversion.	46
4.8	The parameters found using the χ^2 method. (Left) $L = \tilde{D}^{(2)}$ (Right) $L = I$	46

CHAPTER 1

INTRODUCTION

1.1 Inverse Problems

If you have ever driven a car, watched the weather channel, had a CAT scan or MRI then it is likely that your life has benefited in some way from the solution of inverse problems. Solving an inverse problem is the process of recovering some hidden information such as a set of parameters from indirect noisy measurements. For example, geoscientists use inverse theory to determine some information about the structure of the earth, such as possible oil deposits, from measurements taken at the surface of the earth [1]. Inverse theory is widely used in many applied sciences such as image processing, medical imaging, weather forecasting, climate modeling, and astrophysics [1, 4, 6, 7], to name a few.

1.1.1 Formulation of the Problem

Most scientific study of a physical system can be represented with the following components: a minimal set of parameters that completely describes the system, a mathematical model, and some observations. Let $F : \mathbb{R}^m \rightarrow \mathbb{R}^n$ represent the mathematical model that is referred to as the forward problem, $x \in \mathbb{R}^m$ represent the parameters we are trying to estimate, and $d \in \mathbb{R}^n$ represent our data. Then we have the following:

$$d = F(x) + \varepsilon \tag{1.1}$$

where ε represents noise in the data and is an unknown random variable. F can be an analytical equation, an algorithm, or even a “black box” software with inputs and outputs. Often the parameters we are trying to recover are actually a discretized function and F is a discretization of some continuous operator. Determining the model itself is also a type of inverse problem. However, in many practical cases the mathematical model is known and the term “inverse problem” typically refers to the process of determining a set of parameters from a set of data.

In an ideal universe, perhaps the universe of introductory algebra textbooks, the model parameters could be found using:

$$x = F^{-1}(d - \varepsilon). \tag{1.2}$$

In practice the inverse of F is not known or it may not exist. Even if the inverse of F is known, it is likely that it is very sensitive to noise in the data. This noise ε , or ‘error’ as it will be called from now on, generally comes from three main sources:

- Measurement error. No matter what the process or what is being measured, there is going to be some error that is a result of the measurement.
- Modelling error. Tractable mathematical models almost always involve simplification and idealized assumptions and thus do not completely model the physical system.
- Computation error. Even if the model exactly describes the physical system, the model is computed with finite precision.

Due to the elusive nature of F^{-1} and the unavoidable error in the data, x is almost always unknowable and we must be satisfied with an estimate \hat{x} of x . A common way to estimate x is to use the following equation:

$$\hat{x} = \arg \min_x \|d - F(x)\|_2^2. \quad (1.3)$$

Not surprisingly, this type of solution is called the least squares solution and is discussed in more detail in Chapter 2. However, in many applications, finding the estimate from (1.3) is an ill-posed problem and so the solution of (1.3) will not always provide useful answers.

1.2 Ill-Posed Problems

An ill-posed problem is defined as a problem that is not well-posed. In general, there are three criteria for classifying a math problem as well-posed. A problem is said to be well-posed [12] if:

- the problem has a solution,
- the solution is unique,
- the solution depends continuously on the data.

While the first criterion is not usually an issue, the second and third criterion can plague inverse problems. In addition, even if a problem is well-posed mathematically, these criteria must also hold true with respect to the computational precision of a computer. For example, even if there is a solution for the continuous version of (1.3), there might be a range of equivalent solutions at finite precision. Similarly, even if

(1.3) depends continuously on the data, in order to obtain a useful solution it must also be computationally stable with respect to small perturbations of the data. A problem that is not stable with respect to small perturbations is termed ill-conditioned [12] and the degree of instability is quantified with a large condition number. Likewise, a problem that is stable with respect to small perturbations has a small condition number and is termed a well-conditioned problem. So being ill-conditioned is one way an inverse problem can fail to be well-posed.

Figure 1.1 compares a well-posed linear least squares problem to an ill-posed problem. The functional plotted on the left is from the well-posed problem and has a nice well-defined minimum. The functional plotted on the right is from the ill-posed problem and does not have such a well-defined minimum. In fact, there is a entire range of values for which the functional is a minimum. Even if (1.3) is an

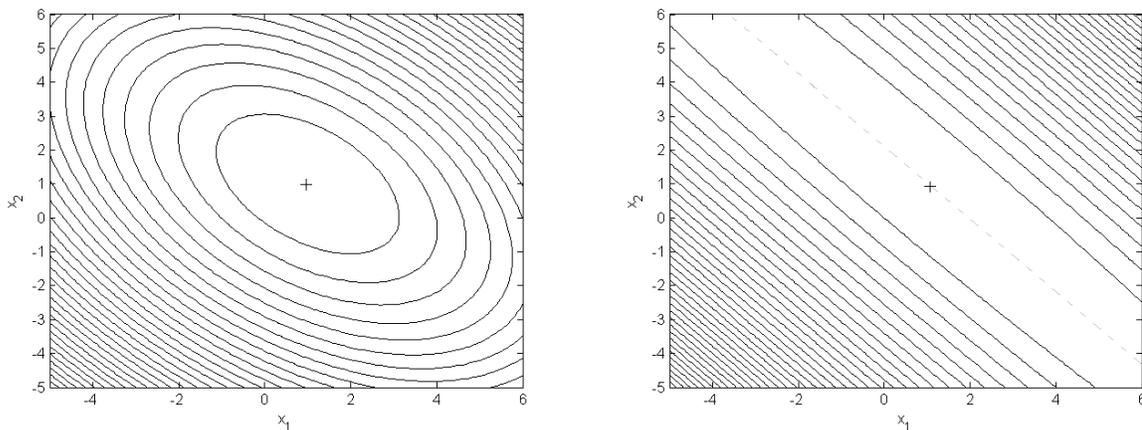


Figure 1.1: Contours of $\|d - F(x)\|_2^2$ for two linear inverse problems. Left: Well-posed problem. Right: Ill-posed problem.

ill-posed problem, this does not necessarily mean that a good estimate of x is not possible. In this case, to estimate x it is necessary to change the ill-posed problem into a well-posed problem by adding some additional information. To obtain a useful

estimate, it is desirable to change the problem just enough to make it well-posed. However, determining how much the problem needs to be regularized is not trivial and this is what this thesis will cover: a method for determining the amount of regularization for nonlinear inverse problems.

CHAPTER 2

LEAST SQUARES

Least squares is a straightforward, computationally inexpensive method that is widely used to solve inverse problems [15]. Even though regularization is the focus of this thesis, we begin with a discussion of unregularized least squares to establish the framework and notation needed for later chapters. In addition, we will explain and exploit some nice statistical properties of this method in this chapter.

The unregularized least squares estimate \hat{x} is given as:

$$\begin{aligned} \mathcal{J}(x) &= \|d - F(x)\|_2^2 \\ \hat{x} &= \arg \min_x \mathcal{J}(x). \end{aligned} \tag{2.1}$$

In a purely mathematical sense, the *arg min* above should be *arg inf*. However, since all real problems are solved in the computational realm and the numbers accessible to the computer are a finite subset of \mathbb{R} , it is valid to use *min* instead of *inf*. This convention will be used throughout the rest of the paper.

2.1 Ordinary Least Squares (OLS)

If F is a linear function, then it can be represented as a matrix $A \in \mathbb{R}^{n \times m}$ so (2.1) becomes

$$\begin{aligned}\mathcal{J}(x) &= \|d - Ax\|_2^2 \\ \hat{x} &= \arg \min_x \mathcal{J}(x).\end{aligned}\tag{2.2}$$

This is a quadratic functional and the minimum can be found directly by setting:

$$\nabla \mathcal{J} = -\frac{1}{2}A^T(d - Ax) = 0.\tag{2.3}$$

Solving for x gives the ordinary least squares estimate:

$$\hat{x} = (A^T A)^{-1} A^T d.\tag{2.4}$$

If $(A^T A)$ is invertible and the problem is well-conditioned, then this is a straightforward way to estimate x . In addition, if we assume that the error ε in the problem from (1.1) is a random variable with a mean of zero, then \hat{x} is an unbiased estimate of x since the expected value of \hat{x} is equal to x , i.e.

$$\begin{aligned}E(\hat{x}) &= E((A^T A)^{-1} A^T d) \\ &= (A^T A)^{-1} A^T E(d) \\ &= (A^T A)^{-1} A^T E(Ax + \varepsilon) \\ &= (A^T A)^{-1} A^T Ax \\ &= x.\end{aligned}\tag{2.5}$$

2.2 Nonlinear Least Squares

The least squares estimate is not as simple for nonlinear problems. The minimum cannot be solved for analytically as in (2.4) so some type of iterative method must

be used. Finding the solution to:

$$\hat{x} = \arg \min_x \mathcal{J}(x) \quad (2.6)$$

falls under a whole field of mathematics called optimization. There are many different methods that can be used to solve this nonlinear unconstrained optimization problem including: genetic algorithms, stochastic algorithms, particle swarm optimization algorithms, quantum optimization algorithms (need a quantum computer), pattern search algorithms, direct search methods, steepest descent algorithms, and conjugate gradient algorithm. However, if the function is well-behaved (i.e. continuous and twice differential in the domain), then Newton's method style algorithms are among the fastest and most efficient, and can even offer quadratic convergence [2, 13].

If we are given a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ that is Fréchet differentiable and a starting point that is sufficiently close to the root then Newton's Method estimates the roots of $f(x)$ by iterating

$$\begin{aligned} x_{k+1} &= x_k + \Delta x_k \\ \Delta x_k &= -\nabla f^{-1}(x_k)f(x_k) \end{aligned} \quad (2.7)$$

until some criterion of convergence is reached, assuming that the Jacobian of $f(x)$ is invertible at each x_k . Here $\nabla f^{-1}(x_k)$ represents the inverse of the Jacobian matrix. Applying this to (2.6), if $\mathcal{J}(x)$ is locally convex, then a local minimum can be found using Newton's method to find:

$$\nabla \mathcal{J}(x) = 0. \quad (2.8)$$

In this case, the Newton iteration becomes:

$$\begin{aligned}
x_{k+1} &= x_k + \Delta x_k \\
\Delta x_k &= -\nabla^2 \mathcal{J}^{-1}(x_k) \nabla \mathcal{J}(x_k)
\end{aligned}
\tag{2.9}$$

where $\nabla^2 \mathcal{J}(x)$ is the Hessian of \mathcal{J} . However, this classical Newton algorithm is not robust and it has been shown that it can even diverge in some cases [13]. Also, it has several other problems in that the Hessian of \mathcal{J} can be difficult to obtain computationally and might not be positive definite at some points. To overcome some of these limitations, there are many modifications of Newton's method. For example, in the modified Newton method, the step length is scaled at each iteration with a positive scalar ρ . Then the step becomes: $\Delta x = -\rho \nabla^2 \mathcal{J}^{-1} \nabla \mathcal{J}$ and a line search is used at each iteration to find the best ρ [13]. Alternatively, Quasi-Newton methods use an approximation for the Hessian, which is updated at each iteration in a way that ensures that it is positive definite and invertible. Restricted-step methods modify the Hessian by $H = \nabla^2 \mathcal{J} + \lambda^2 I$, where λ is chosen to ensure that the H is invertible and to ensure the step Δx_k leads to a reduction in $\mathcal{J}(x)$ [13].

2.2.1 Gauss-Newton Method and Levenberg-Marquardt

The Gauss-Newton method is an adaptation of Newton's method, which exploits the structure of least squares problems. This method has the benefit that it doesn't require the calculation or storage of the Hessian, which can be computationally expensive. The Gauss-Newton method is the basis for much of the theory developed later in this thesis and so is considered here in more detail.

The Gauss-Newton step is given as:

$$\Delta x_k = - (J_k^T J_k)^{-1} J_k^T (d - F(x_k)) \quad (2.10)$$

where J_k is the Jacobian of F at x_k . This is derived from Newton's method as follows:

The first and second Fréchet derivatives of \mathcal{J} are given as:

$$\nabla \mathcal{J}(x) = 2J_k^T (d - F(x)) \quad (2.11)$$

$$\nabla^2 \mathcal{J}(x) = 2(J_k^T J_k + Q(x)) \quad (2.12)$$

where $Q(x) = \sum_{i=1}^m \nabla^2 F_i(x) [d - F(x)]_i$ and $\nabla^2 F_i(x)$ is the Hessian of the i th component of $F(x)$. Ignoring $Q(x)$ from $\nabla^2 \mathcal{J}$ in the Newton iteration gives (2.10). So the Gauss-Newton method approximates $\nabla^2 \mathcal{J}(x)$ with just the first-order part $J_k^T J_k$. If necessary, the Jacobian can be calculated with finite differences without affecting the performance of the method [2]. The Gauss-Newton method, when it converges, can be more efficient than the full Newton method. It also can ultimately achieve a quadratic rate of convergence [2]. In addition, it usually converges faster and is more efficient than the Quasi-Newton method [2],[13]. However, it is based on the fact that $\|J_k^T J_k\| \gg \|Q(x)\|$, which is true for small residual problems and is not a good approximation when the largest eigenvalue of $J^T J_k$ is comparable to $\|d - F(x_k)\|_2^2$ [2].

Levenberg-Marquardt

The Gauss-Newton step (2.10) can fail to reduce $\mathcal{J}(x)$ if $J_k^T J_k$ is close to singular or if $J_k^T J_k$ is a poor approximation of the Hessian of \mathcal{J} . The Levenberg-Marquardt

(LM) algorithm [13] is a modification of the Gauss-Newton method that allows for singular or ill-conditioned matrices $J^T J$ and takes smaller, safer steps by introducing a parameter λ_k and a diagonal matrix D with positive diagonal elements:

$$\Delta x_k = -(J_k^T J_k + \lambda D)^{-1} J_k^T (d - F(x_k)) \quad (2.13)$$

where D is a diagonal matrix with positive diagonal elements. For simplicity, $D = I$ is often used and in this case Δx_k is an interpolation between the steepest descent step and the Gauss-Newton step. Alternatively, a typical choice for D is a matrix with diagonal elements equal to those of $J^T J$. Note that $\Delta x = -M^{-1} \nabla \mathcal{J}$ is guaranteed to be a descent direction as long as M is positive definite [2].

The Levenberg-Marquardt parameter λ_k is chosen so that $\mathcal{J}(x_{k+1}) < \mathcal{J}(x_k)$. If λ_k is too small, then Δx_k might not lead to a reduction in the value of \mathcal{J} . If λ_k is too large, then the algorithm will take small steps and its progress will be slow. A common way to determine λ_k is as follows: start with a small value for λ_1 , i.e. $\lambda_1 = 0.1$. If Δx_k leads to a reduction of $\mathcal{J}(x)$, then update $\lambda_{k+1} = \lambda_k/10$. However, if Δx_k doesn't reduce $\mathcal{J}(x)$, then increase $\lambda_k = 10\lambda_k$ and recompute Δx_k . Repeat this until the choice of λ_k leads to a reduction in $\mathcal{J}(x)$, update $x_{k+1} = x_k + \Delta x_k$ [13]. More complex implementations use a trust-region methodology [13], which chooses a λ_k such that $\|\Delta x_k\|_2^2 \leq \mu$, where μ is the radius of the “trust region.” These methods update this trust region at each iteration based on the success of the previous iterations in reducing \mathcal{J} [2]. In Chapter 3, we will implement the LM algorithm to solve the regularized least squares problem. While the λ_k appears to be regularizing the solution in a similar fashion to Tikhonov regularization, it only regularizes at each iteration and so it doesn't regularize the solution. There is general agreement that

LM algorithm is in general a robust method and works well for many nonlinear least squares problems [2].

2.3 Generalized Least Squares

If the data are of varying scales or if the measurements have different variances, or if the errors in the data are correlated, then these factors can be taken into account in the estimation of x . Generalized least squares does this by weighting the least squares problem with the inverse of the covariance matrix of the error. The generalized least squares estimate \hat{x}_{GLS} is:

$$\begin{aligned}\mathcal{J}_{\text{GLS}}(x) &= \|d - F(x)\|_{C_\varepsilon^{-1}}^2 \\ \hat{x}_{\text{GLS}} &= \arg \min_x \mathcal{J}_{\text{GLS}}(x)\end{aligned}\tag{2.14}$$

where $\|d - F(x)\|_{C_\varepsilon^{-1}}^2$ is the weighted 2-norm $(d - F(x))^T C_\varepsilon^{-1} (d - F(x))$. This is an intuitive addition to least squares, because if we have some measurements with a large variance then it makes sense that these points should have less weight. Also, if $\varepsilon \sim N(0, C_\varepsilon)$, then \hat{x}_{GLS} from (2.14) is equivalent to the maximum likelihood estimate.

All the previous least squares results can be applied to the generalized least square estimate by first converting the generalized least squares problem into an OLS problem:

$$\begin{aligned}\tilde{F}(x) &= C_\varepsilon^{-1/2} F(x), \\ \tilde{d} &= C_\varepsilon^{-1/2} d.\end{aligned}\tag{2.15}$$

Then \hat{x}_{GLS} becomes the OLS estimate of the new problem:

$$\hat{x} = \arg \min_x \| \tilde{d} - \tilde{F}(x) \|_2^2. \quad (2.16)$$

We now introduce a theorem describing an important statistical property of \mathcal{J}_{GLS} from (2.14) at its minimum value. This theorem will provide much of the basis needed for the theory developed later in this work.

Theorem 1. *If $F : \mathbb{R}^m \rightarrow \mathbb{R}^n$ is a linear function, and $\varepsilon \sim N(0, C_\varepsilon)$, then $\mathcal{J}_{\text{GLS}}(\hat{x}_{\text{GLS}}) \sim \chi_{n-m}^2$.*

Proof. Since $F : \mathbb{R}^m \rightarrow \mathbb{R}^n$ is a linear function we can write it as a matrix A with dimension $n \times m$. Also, $x \in \mathbb{R}^m$, $d \in \mathbb{R}^n$, and C_ε has dimension $n \times n$.

So we have:

$$\begin{aligned} \tilde{d} &= C_\varepsilon^{-1/2} Ax + C_\varepsilon^{-1/2} \varepsilon \\ &= \tilde{A}x + \tilde{\varepsilon}. \end{aligned} \quad (2.17)$$

Theorem 7 in Appendix A implies: $\tilde{\varepsilon} \sim N(0, \mathbf{I}_n)$. Now $\mathcal{J}_{\text{GLS}}(x) = \| \tilde{d} - \tilde{A}x \|_2^2$ and $\hat{x}_{\text{GLS}} = \hat{x} = (\tilde{A}^T \tilde{A})^{-1} \tilde{A}^T \tilde{d}$, which gives:

$$\begin{aligned} \| \tilde{d} - \tilde{A}\hat{x} \|_2^2 &= \| \tilde{d} - \tilde{A}(\tilde{A}^T \tilde{A})^{-1} \tilde{A}^T \tilde{d} \|_2^2 \\ &= \| (\mathbf{I}_n - \tilde{A}(\tilde{A}^T \tilde{A})^{-1} \tilde{A}^T) \tilde{d} \|_2^2 \\ &= \| (\mathbf{I}_n - P) \tilde{d} \|_2^2. \end{aligned} \quad (2.18)$$

Then, $P = \tilde{A}(\tilde{A}^T \tilde{A})^{-1} \tilde{A}^T$ is a projection matrix and it orthogonally projects \tilde{d} onto the range space of \tilde{A} [14]. It is easy to see that P is symmetric and idempotent and that $P\tilde{A} = \tilde{A}$. This implies that

$$\begin{aligned}
(\mathbf{I}_n - P)\tilde{d} &= (\mathbf{I}_n - P)\tilde{d} - \tilde{A}x + \tilde{A}x \\
&= (\mathbf{I}_n - P)\tilde{d} - \tilde{A}x + P\tilde{A}x \\
&= (\mathbf{I}_n - P)(\tilde{d} - \tilde{A}x) \\
&= (\mathbf{I}_n - P)\tilde{\varepsilon}.
\end{aligned} \tag{2.19}$$

Combining this result and the fact that $(\mathbf{I}_n - P)$ is idempotent and symmetric implies:

$$\|\tilde{d} - \tilde{A}(\hat{x})\|_2^2 = \tilde{\varepsilon}^T (\mathbf{I}_n - P) \tilde{\varepsilon}. \tag{2.20}$$

Theorem 5 from Appendix A says that the rank of a matrix that is idempotent and symmetric is equal to its trace. Using this result:

$$\begin{aligned}
\text{rank}(\mathbf{I}_n - P) &= \text{trace}(\mathbf{I}_n - P) \\
&= \text{trace}(\mathbf{I}_n) - \text{trace}(P) \\
&= n - \text{trace}(\tilde{A}(\tilde{A}^T \tilde{A})^{-1} \tilde{A}^T) \\
&= n - \text{trace}((\tilde{A}^T \tilde{A})^{-1} \tilde{A}^T \tilde{A}) \\
&= n - m
\end{aligned} \tag{2.21}$$

Finally, applying the Theorem 8 from Appendix A:

$$\tilde{\varepsilon}^T (\mathbf{I}_n - P) \tilde{\varepsilon} \sim \chi_{n-m}^2. \tag{2.22}$$

□

The use of Theorem 1 to analyze the least squares solution is sometimes called the χ^2 test. If $\mathcal{J}(\hat{x})$ is much larger than the mean of χ_{n-m}^2 or is outside of some confidence interval, for example say a 95 % confidence interval, then this would suggest that the

errors in the data are larger than expected. This indicates that either the covariance of the data errors is too small or the forward problem does not accurately model the physical system. Therefore, Theorem 1 supplies some very useful information about the solution to the inverse problem. This result only applies to linear problems, however, it is shown in [13] that it approximately holds for nonlinear problems in a region around \hat{x} .

CHAPTER 3

REGULARIZATION AND THE χ^2 METHOD

As mentioned in previous chapters, inverse problems are often ill-posed in practice and finding a solution requires some form of regularization. A common way to regularize inverse problems is to add a second term to the functional being minimized in order to stabilize and add uniqueness to the solution. This is known as Tikhonov regularization and this modified functional is sometimes called the Tikhonov functional:

$$\mathcal{J}_{\text{tkh}}(x) = \|d - F(x)\|^2 + \alpha^2 \|Lx - z\|^2, \quad (3.1)$$

where L is a linear operator $L : \mathbb{R}^m \rightarrow \mathbb{R}^q$, $z \in \mathbb{R}^q$, and α is a scalar.

The matrix L is commonly chosen to be the identity operator or an approximate first or second derivative operator. If L is the identity, then z could be an initial estimate of x . In this case, the regularization parameter α controls the compromise between how far the solution deviates from the original estimate and how well the solution fits the data. Alternatively, when x is the discretization of a continuous function, then the expected structure of x can be exploited by choosing L to represent a derivative operator. In this case, z represents the desired slope in the solution and is often set to be 0 to obtain smooth solutions. In this case, α controls the compromise between how smooth the solution is and how well the solution fits that data.

The value of the parameter α controls how much (3.1) changes the original inverse

problem. It is desirable to choose α so that it changes the original problem just enough that a good estimate for x can be obtained. Choosing the value of α that accomplishes this, however, is not trivial.

3.1 Choice of Regularization Parameter

There is a voluminous amount of literature on how to determine the regularization parameter for linear least squares problems. Common methods include L-Curve, Generalized Cross Validation, and the discrepancy principle. For a complete treatment of these methods, the reader is referred to the literature, specifically [1, 4]. A relatively new method, called the χ^2 method, is proposed by Mead in [8] and developed further in [10, 9]. The focus of this thesis is to extend this method for solving nonlinear inverse problems.

3.1.1 Nonlinear Regularization

Regularization methods for linear problems do not straightforwardly extend to nonlinear least squares problems. Since the nonlinear problems are solved iteratively, the methods for determining the regularization parameter generally breakdown into two approaches.

In the first approach, α remains fixed throughout the nonlinear inversion process. In these methods, the inversion is done multiple times for different values of α until the solution meets some criterion. Some criteria used for evaluating the solution are the discrepancy principle [1] and Generalized Cross Validation (GCV) [3].

In the second approach, α is estimated dynamically at each iteration. In this approach, the nonlinear inverse problem is solved only once, but the optimization

procedure has to be integrated with the method for estimating α . Some examples of this type of method include Occam's inversion and an implementation of GCV as proposed in [3]. The nonlinear χ^2 method which uses this second approach and is an alternative to these methods and is developed in this thesis.

3.1.2 Statistical Framework

A popular data assimilation method in weather forecasting based on a Bayesian framework is known as the three-dimensional variational method (3DVAR) [7]. It starts with the following assumptions:

$$\begin{aligned} d &= F(x) + \varepsilon, \\ x &= x_p + f, \end{aligned} \tag{3.2}$$

where $\varepsilon \sim N(0, C_\varepsilon)$, $f \sim N(0, C_f)$ and x_p is an initial estimate of x . This differs from traditional inverse problems in that we have both noisy data and a prior probability distribution for the parameter set.

Since both ε, f are normal, it is straightforward to find the maximum a posteriori (MAP) estimate for x for a given data set. The MAP estimate x_M is:

$$x_M = \arg \max_x (P(x|d)), \tag{3.3}$$

where $P(x|d)$ represents the conditional probability density function for x given the data d . Using Baye's theorem, it possible to write $P(x|d)$ in terms of prior distributions

$$P(x|d) = \frac{P(d|x)P(x)}{P(d)}, \tag{3.4}$$

where $P(x)$ represents the prior distribution for x , $P(d)$ represents the distribution for d and $P(d|x)$ represents the conditional probability density function for d , given the data x .

So x_M becomes:

$$x_M = \arg \max_x \left(\frac{P(d|x)P(x)}{P(d)} \right). \quad (3.5)$$

Since ε is normally distributed, $P(d|x)$ can be written as

$$P(d|x) = \frac{1}{2\pi^{\frac{n}{2}} |C_\varepsilon|^{\frac{1}{2}}} \exp \left(-\frac{1}{2} (d - F(x))^T C_\varepsilon^{-1} (d - F(x)) \right).$$

In addition, since the prior distribution is a multivariate normal, that we can write:

$$P(x) = \frac{1}{2\pi^{\frac{m}{2}} |C_f|^{\frac{1}{2}}} \exp \left(-\frac{1}{2} (x - x_p)^T C_f^{-1} (x - x_p) \right).$$

Using these distributions and the fact that $P(d)$ does not depend on x , the MAP estimate becomes:

$$\begin{aligned} x_M &= \arg \max_x \left\{ \exp \left(-\frac{1}{2} (d - F(x))^T C_\varepsilon^{-1} (d - F(x)) \right) \exp \left(-\frac{1}{2} (x - x_p)^T C_f^{-1} (x - x_p) \right) \right\} \\ &= \arg \min_x \left\{ (d - F(x))^T C_\varepsilon^{-1} (d - F(x)) + (x - x_p)^T C_f^{-1} (x - x_p) \right\}. \end{aligned} \quad (3.6)$$

The MAP estimate x_M minimizes

$$\mathcal{J}_M(x) = \|d - F(x)\|_{C_\varepsilon^{-1}}^2 + \|x - x_p\|_{C_f^{-1}}^2, \quad (3.7)$$

which is very similar to the Tikhonov functional \mathcal{J}_{tkh} from (3.1) when L is the identity and z is the initial estimate. In (3.7), each term in the functional is weighted with its

respective inverse covariance, whereas in (3.1) the second term is weighted with α^2 .

3.1.3 Linear χ^2 Method

If $\varepsilon \sim N(0, \sigma_\varepsilon^2 \mathbf{I})$ and $f \sim N(0, \sigma_f^2 \mathbf{I})$, then $x_{\mathbf{M}}$ is identical to the estimate found by minimizing the Tikhonov functional with L as the identity, z as the initial estimate, and with $\alpha = \sigma_\varepsilon / \sigma_f$. Of course, many times in inverse problems, the prior covariance for f is not available. However, all is not lost. Mead in [8] suggested capitalizing on Theorem 2 to estimate α .

Theorem 2. *If $F : \mathbb{R}^m \rightarrow \mathbb{R}^n$ is a linear function and the following holds:*

$$\begin{aligned} d &= F(x) + \varepsilon, \\ x &= x_p + f, \end{aligned} \tag{3.8}$$

then $\mathcal{J}_{\mathbf{M}}(x)$ from (3.7) at its minimum value, i.e. $\mathcal{J}_{\mathbf{M}}(x_{\mathbf{M}})$, follows a χ^2 distribution with n degrees of freedom.

Proof. Since $F : \mathbb{R}^m \rightarrow \mathbb{R}^n$ is a linear function, we can write it as a matrix A with dimension $n \times m$. Also, $x \in \mathbb{R}^m$, $d \in \mathbb{R}^n$, C_ε has dimension $n \times n$ and C_f has dimension $m \times m$.

First, rewrite (3.7) as:

$$\begin{aligned} \mathcal{J}_{\mathbf{M}}(x) &= \left\| \begin{array}{c} C_\varepsilon^{-1/2}(Ax - d) \\ C_f^{-1/2}(x - x_p) \end{array} \right\|_2^2 \\ x_{\mathbf{M}} &= \arg \min_x \mathcal{J}_{\mathbf{M}}(x). \end{aligned} \tag{3.9}$$

This can be written as an ordinary least squares problem:

$$\mathcal{J}(x) = \left\| \begin{bmatrix} C_\varepsilon^{-1/2} A \\ C_f^{-1/2} \end{bmatrix} x - \begin{bmatrix} C_\varepsilon^{-1/2} d \\ C_f^{-1/2} x_p \end{bmatrix} \right\|_2^2. \quad (3.10)$$

For the sake of simplicity, let

$$A^* = \begin{bmatrix} C_\varepsilon^{-1/2} A \\ C_f^{-1/2} \end{bmatrix}, \quad d^* = \begin{bmatrix} C_\varepsilon^{-1/2} d \\ C_f^{-1/2} x_p \end{bmatrix}, \quad \varepsilon^* = \begin{bmatrix} C_\varepsilon^{-1/2} \varepsilon \\ C_f^{-1/2} f \end{bmatrix}. \quad (3.11)$$

Then:

$$\begin{aligned} \mathcal{J}(x) &= \|d^* - A^*x\|_2^2 \\ \hat{x} &= \arg \min_x \mathcal{J}(x), \end{aligned} \quad (3.12)$$

where A^* has dimension $(n+m) \times m$, d^* has dimension $(n+m) \times 1$, and $\varepsilon^* \sim N(0, I_{n+m})$. By Theorem 1, in the previous chapter, $\mathcal{J}(\hat{x}) \sim \chi_n^2$. \square

The method proposed in [8] is called the χ^2 method and it says choose α such that the minimum of the functional (3.7) has a value that is consistent with its χ^2 distribution. This is implemented in [8] as finding the α that makes the minimum of the functional equal to the mean of the χ^2 distribution. Also, Mead showed in [8] that Theorem 2 holds asymptotically when ε and f are not normally distributed, which allows this method to be applied in a more general sense.

3.2 χ^2 Tests for Gauss-Newton Method

Theorem 2 has only been shown for linear problems. For nonlinear problems, the distribution of $\mathcal{J}_M(x_M)$ is not usually known. However, if the nonlinear inverse problem

is solved with a sequence of linearizations, then it is possible to find appropriate χ^2 tests at each iteration.

The Gauss-Newton method to find $x_M = \arg \min_x \mathcal{J}_M(x)$ from (3.7) is as follows. First find the first and second Fréchet derivative of \mathcal{J}_M ,

$$\nabla \mathcal{J}_M(x) = J^T C_\varepsilon^{-1} (d - F(x)) - C_f^{-1} (x - x_p), \quad (3.13)$$

where J is the Jacobian of F at x . Now

$$\nabla^2 \mathcal{J}_M(x) = J^T C_\varepsilon^{-1} J + Q(x) + C_f^{-1}, \quad (3.14)$$

where $Q(x_k)$ is the second-order information of \mathcal{J}_M . The Gauss-Newton method ignores this Q , so we get the following iteration:

$$\begin{aligned} x_{k+1} &= x_k + \Delta x_k \\ \Delta x_k &= - \left(J_k^T C_\varepsilon^{-1} J_k + C_f^{-1} \right)^{-1} \left(J_k^T C_\varepsilon^{-1} (d - F(x_k)) - C_f^{-1} (x_k - x_p) \right). \end{aligned} \quad (3.15)$$

The Gauss-Newton method can be converted to a sequence of linear OLS problems with the following manipulations:

$$x_{k+1} = x_k + \left(J_k^T C_\varepsilon^{-1} J_k + C_f^{-1} \right)^{-1} \left(J_k^T C_\varepsilon^{-1} r_k - C_f^{-1} (x_k - x_p) \right), \quad (3.16)$$

where $r_k = d - F(x_k)$. Now multiplying both sides with $\left(J_k^T C_\varepsilon^{-1} J_k + C_f^{-1} \right)$ gives:

$$\left(J_k^T C_\varepsilon^{-1} J_k + C_f^{-1} \right) x_{k+1} = \left(J_k^T C_\varepsilon^{-1} J_k + C_f^{-1} \right) x_k + \left(J_k^T C_\varepsilon^{-1} r_k - C_f^{-1} (x_k - x_p) \right). \quad (3.17)$$

$C_f^{-1} x_k$ subtracts out and gives

$$(J_k^T C_\varepsilon^{-1} J_k + C_f^{-1}) x_{k+1} = (J_k^T C_\varepsilon^{-1} J_k) x_k + (J_k^T C_\varepsilon^{-1} r_k + C_f^{-1} x_p). \quad (3.18)$$

Rewrite and factor out $J_k^T C_\varepsilon^{-1}$ on right-hand side

$$(J_k^T C_\varepsilon^{-1} J_k + C_f^{-1}) x_{k+1} = J_k^T C_\varepsilon^{-1} ((J_k x_k + r_k) + C_f^{-1} x_p). \quad (3.19)$$

This can be factored again into the normal equations:

$$\begin{bmatrix} C_\varepsilon^{-1/2} J_k \\ C_f^{-1/2} \end{bmatrix}^T \begin{bmatrix} C_\varepsilon^{-1/2} J_k \\ C_f^{-1/2} \end{bmatrix} x_{k+1} = \begin{bmatrix} C_\varepsilon^{-1/2} J_k \\ C_f^{-1/2} \end{bmatrix}^T \begin{bmatrix} C_\varepsilon^{-1/2} (\tilde{d}_k) \\ C_f^{-1/2} x_p \end{bmatrix}, \quad (3.20)$$

where $\tilde{d}_k = d - F(x_k) + J_k x_k$. Finally, this can be written as the sequence of linear OLS problems:

$$\begin{aligned} \tilde{\mathcal{J}}_k(x) &= \|\tilde{d}_k - J_k x\|_{C_\varepsilon^{-1}}^2 + \|x - x_p\|_{C_f^{-1}}^2 \\ \hat{x}_{k+1} &= \arg \min_x \tilde{\mathcal{J}}_k(x) \end{aligned} \quad (3.21)$$

The sequence of OLS problems in (3.21) solves the following linear inverse problem at each iteration.

$$\begin{aligned} \tilde{d}_k &= J_k x_{k+1} + \varepsilon_k, \\ x_{k+1} &= x_p + f \end{aligned} \quad (3.22)$$

where $\varepsilon_k = \varepsilon + \nu_k$ with $Cov(\varepsilon_k) = C_{\varepsilon_k}$ and ν_k represents error introduced by the linearization, i.e. $\nu_k = F(x) - F(x_k - J_k(x - x_k))$. The following theorem gives χ^2 distribution for the Gauss-Newton functional $J_k(x)$ under several assumptions.

Theorem 3. If $\tilde{\mathcal{J}}_k(x) = \|\tilde{d}_k - J_k x\|_{C_\varepsilon^{-1}}^2 + \|x - x_p\|_{C_f^{-1}}^2$, $\hat{x}_{k+1} = \arg \min_x \tilde{\mathcal{J}}_k(x)$, the nonlinear error is zero, and the following are true:

$$\begin{aligned} d_k &= J_k x_{k+1} + \varepsilon_k & \varepsilon_k &\sim N(0, C_{\varepsilon_k}) \\ x_{k+1} &= x_p + f & f &\sim N(0, C_f) \end{aligned} \tag{3.23}$$

then $\tilde{\mathcal{J}}_k(\hat{x}_{k+1}) \sim \chi_n^2$.

Proof. This follows trivially from Theorem 2. □

If the nonlinear error is zero, then the problem is likely linear. However, this theorem can still be used to develop the χ^2 test for nonlinear problems by making the assumption that $C_{\varepsilon_k} \approx C_\varepsilon$. This approximation will get better as the iterations gets closer to the solution and the nonlinear error is reduced. Under this assumption, the χ^2 method can be applied at each iterations to achieve increasingly better estimates for C_f^{-1} . In the next chapter, we show that this assumption works well for two inverse problems given in [1].

Now we consider a more general case where L is used as in (3.1). It is not difficult to see that in a similar way we can minimize $\mathcal{J}_M(x) = \|d - F(x)\|_{C_\varepsilon^{-1}}^2 + \|Lx - z\|_{C_f^{-1}}^2$ with the sequence of linear OLS problems:

$$\tilde{\mathcal{J}}_k(x) = \|\tilde{d}_k - J_k x\|_{C_\varepsilon^{-1}}^2 + \|Lx - z\|_{C_f^{-1}}^2. \tag{3.24}$$

Often when L is chosen to represent a derivative operator, it is not a square matrix. In this case, the χ^2 distribution of $\tilde{\mathcal{J}}_k(x)$ has different degrees of freedom given in Theorem 4.

Theorem 4. If $\tilde{\mathcal{J}}_k(x) = \|\tilde{d}_k - J_k x\|_{C_\varepsilon^{-1}}^2 + \|Lx - z\|_{C_f^{-1}}^2$, where $L : \mathbb{R}^m \rightarrow \mathbb{R}^q$ is a linear operator and $\hat{x}_{k+1} = \arg \min_x \tilde{\mathcal{J}}_k(x)$, the invertibility condition holds:

$\mathcal{N}(J_k) \cap \mathcal{N}(L) = 0$ where $\mathcal{N}(A)$ is the null space of A , the nonlinear error is zero, and the following are true:

$$\begin{aligned} d_k &= J_k x + \varepsilon_k, & \varepsilon_k &\sim N(0, C_{\varepsilon_k}) \\ Lx &= z + f & f &\sim N(0, C_f) \end{aligned} \quad (3.25)$$

Then, $\tilde{\mathcal{J}}_k(\hat{x}) \sim \chi_{n-m+q}^2$.

Proof. $L : \mathbb{R}^m \rightarrow \mathbb{R}^q$ is a linear operator and $x \in \mathbb{R}^m$, $z \in \mathbb{R}^q$, $d \in \mathbb{R}^n$, C_ε has dimension $n \times n$, and C_f has dimension $q \times q$.

Rewrite as an ordinary least squares problem:

$$\mathcal{J}(x) = \left\| \begin{bmatrix} C_\varepsilon^{-1/2} J_k \\ C_f^{-1/2} L \end{bmatrix} x - \begin{bmatrix} C_\varepsilon^{-1/2} \tilde{d}_k \\ C_f^{-1/2} z \end{bmatrix} \right\|_2^2. \quad (3.26)$$

For sake of simplicity, let

$$A^* = \begin{bmatrix} C_\varepsilon^{-1/2} J_k \\ C_f^{-1/2} L \end{bmatrix}, \quad d^* = \begin{bmatrix} C_\varepsilon^{-1/2} \tilde{d}_k \\ C_f^{-1/2} z \end{bmatrix}, \quad \varepsilon^* = \begin{bmatrix} C_\varepsilon^{-1/2} \varepsilon \\ C_f^{-1/2} f \end{bmatrix}. \quad (3.27)$$

The least squares problem can be written as:

$$\begin{aligned} \tilde{\mathcal{J}}_k(x) &= \|d^* - A^* x\|_2^2, \\ \hat{x} &= \arg \min_x \mathcal{J}(x), \end{aligned} \quad (3.28)$$

where A^* has dimension $(n+q) \times m$, d^* has dimension $(n+q) \times 1$, and $\varepsilon^* \sim N(0, I_{n+q})$.

By Theorem 1 in Chapter 2, $\tilde{\mathcal{J}}_k(\hat{x}) \sim \chi_{n-m+q}^2$. □

3.2.1 χ^2 Tests for Levenberg-Marquardt Method

Recall that the Levenberg-Marquardt method is a modification of the Gauss-Newton method to help ensure the convergence of the algorithm. The LM step to find $x_M = \arg \min_x \mathcal{J}_M(x)$ from (3.7) is as follows:

$$\begin{aligned} x_{k+1}^{\text{LM}} &= x_k + \Delta x_k, \\ \Delta x_k &= - \left(J_k^T C_\varepsilon^{-1} J_k + C_f^{-1} + \lambda^2 D \right)^{-1} \left(J_k^T C_\varepsilon^{-1} (d - F(x_k)) - C_f^{-1} (x_k - x_p) \right). \end{aligned} \quad (3.29)$$

It is possible to write the regularized LM method as a sequence of OLS problems in a similar way as the Gauss-Newton method. These iterates become:

$$\begin{aligned} \tilde{\mathcal{J}}_k^{\text{LM}}(x) &= \|\tilde{d}_k - J_k x\|_{C_\varepsilon^{-1}}^2 + \|x - x_p\|_{C_f^{-1}}^2 + \lambda^2 \|D(x - x_k)\|_2^2, \\ x_{k+1}^{\text{LM}} &= \arg \min_x \tilde{\mathcal{J}}_k^{\text{LM}}(x). \end{aligned} \quad (3.30)$$

The χ^2 test is not clear for this more complicated functional because there is no statistical information about the third term in the functional. However, it is possible to derive an approximate χ^2 test for the LM method. First, to simplify the following manipulations, we convert $\mathcal{J}_M(x)$ into a nonlinear OLS problem. Let:

$$F(x)^* = \begin{bmatrix} C_\varepsilon^{-1/2} F(x) \\ C_f^{-1/2} Lx \end{bmatrix}, \quad d^* = \begin{bmatrix} C_\varepsilon^{-1/2} \tilde{d}_k \\ C_f^{-1/2} z \end{bmatrix}, \quad \varepsilon^* = \begin{bmatrix} C_\varepsilon^{-1/2} \varepsilon \\ C_f^{-1/2} f \end{bmatrix}. \quad (3.31)$$

Then we have the following new problem:

$$d^* = F^*(x) + \varepsilon^* \quad \text{where} \quad \varepsilon^* \sim N(0, I_{n+q}). \quad (3.32)$$

In Chapter 2, we derived the Gauss-Newton method as a modification of Newton's

method, but it is helpful here to derive it in a slightly different way. Consider the Taylor series expansion of F^* around a point x_k :

$$F^*(x) = F^*(x_k) + J_k^*(x - x_k) + \text{higher order terms}, \quad (3.33)$$

where J_k^* is the Jacobian of F^* at x_k . Plugging this back into (3.32), we get:

$$d^* = F^*(x_k) + J_k^*(x - x_k) + \varepsilon_k^*, \quad (3.34)$$

where $\varepsilon_k^* = \varepsilon^* + \nu_k$ and ν_k represents the error introduced by ignoring the higher-order terms. Rewriting:

$$r_k^* = J_k^* \Delta x_k + \varepsilon_k^*, \quad (3.35)$$

where $r_k^* = d^* - F^*(x_k)$ and $\Delta x_k = x_{k+1} - x_k$. We see that the OLS estimate for Δx_k is

$$\widehat{\Delta x_k} = (J_k^{*T} J_k^*)^{-1} J_k^{*T} r_k^*. \quad (3.36)$$

As in the proof of Theorem 1,

$$\begin{aligned} \|r_k^* - J_k^* \widehat{\Delta x_k}\|_2^2 &= \|(I - P)r_k^*\|_2^2, \\ &= \|(I - P)\varepsilon_k^*\|_2^2, \end{aligned} \quad (3.37)$$

where $P = J_k^*(J_k^{*T} J_k^*)^{-1} J_k^{*T}$. Now replace $\widehat{\Delta x_k}$ in (3.37) with the LM step $\Delta x_k = (J_k^{*T} J_k^* + \lambda_k^2 D^T D)^{-1} J_k^{*T} r_k^*$, so

$$\begin{aligned} \|r_k^* - J_k^* \Delta x_k\|_2^2 &= \|(r_k^* - J_k^*(J_k^{*T} J_k^* + \lambda_k^2 D^T D)^{-1} J_k^{*T} r_k^*)\|_2^2 \\ &= \|(I - \hat{P})r_k^*\|_2^2. \end{aligned} \quad (3.38)$$

where $\hat{P} = J_k^*(J_k^{*T}J_k^* + \lambda_k^2 D^T D)^{-1}J_k^{*T}$. \hat{P} is symmetric but it is not idempotent and so is not an orthogonal projection. However, if $\|J_k^{*T}J_k^*\| \gg \|\lambda_k^2 D^T D\|$, then $(J_k^{*T}J_k^*)^{-1} \approx (J_k^{*T}J_k^* + \lambda_k^2 D^T D)^{-1}$ and so \hat{P} is approximately equal to the projection P from (3.37). Then,

$$\begin{aligned}
\|r_k^* - J_k^* \Delta x_k\|_2^2 &= \|(I - \hat{P})r_k^*\|_2^2 \\
&\approx \|(I - P)r_k^*\|_2^2 \\
&= \|(I - P)\varepsilon_k^*\|_2^2 \\
&\sim \chi_n^2 \quad (\text{if } \varepsilon_k^* = \varepsilon^*).
\end{aligned} \tag{3.39}$$

Now, rewriting:

$$\begin{aligned}
\|r_k^* - J_k^* \Delta x_k\|_2^2 &= \|d^* - F^*(x_k) - J_k^*(x_{k+1}^{\text{LM}} - x_k)\|_2^2 \\
&= \|(d^* - F^*(x_k) + x_k) - J_k^* x_{k+1}^{\text{LM}}\|_2^2 \\
&= \left\| \begin{bmatrix} C_\varepsilon^{-1/2} J_k \\ C_f^{-1/2} \end{bmatrix} x_{k+1}^{\text{LM}} - \begin{bmatrix} C_\varepsilon^{-1/2} \tilde{d}_k \\ C_f^{-1/2} x_p \end{bmatrix} \right\|_2^2 \\
&= \tilde{\mathcal{J}}_k(x_{k+1}^{\text{LM}}).
\end{aligned} \tag{3.40}$$

So the Gauss-Newton functional $\tilde{\mathcal{J}}_k$ at the LM estimate x_{k+1}^{LM} approximately follows a χ_n^2 distribution and will be a better approximation as the LM iterates progress because λ_k will go to zero, as will the nonlinear error. We show experimentally in Chapter 4 that $\tilde{\mathcal{J}}_k(x_{k+1}^{\text{LM}})$ closely follows a χ_n^2 distribution.

3.3 Nonlinear χ^2 Method

In Section 3.2 and Theorems 3 and 4, we derived approximate χ^2 distributions for the regularized Gauss-Newton functional J_k at \hat{x}_{k+1} , i.e. $\tilde{\mathcal{J}}_k(\hat{x}_{k+1})$. Also, in Section 3.2.1, we found an approximate χ^2 distribution for $\tilde{\mathcal{J}}_k(x_{k+1}^{\text{LM}})$. In keeping with the approach proposed by Mead in [8] for the linear χ^2 method, we suggest using these χ^2 distributions to estimate the regularization parameter. However, when solving real problems, only one sample of $\tilde{\mathcal{J}}_k$ is available because there is only one realization of the error ε in the data. Therefore, the best we can do is to use a single characteristic of the distribution to find the regularization parameter.

In [10], they suggest using the mean of the χ^2 distribution to estimate α . However, for a χ^2 distribution the median is approximately equal to the mean. This implies that if a perfectly weighted $\tilde{\mathcal{J}}_k(\hat{x}_{k+1})$ is sampled multiple times, about half of these samples will be greater than the mean. If we estimate the regularization parameter such that $\tilde{\mathcal{J}}_k(\hat{x}_{k+1})$ is always equal to the mean, then about half of the time the regularization parameter will have to be made smaller to compensate for the realization of data error that makes a perfectly weighted $\tilde{\mathcal{J}}_k(\hat{x}_{k+1})$ larger than the mean. This means that choosing α such that $\tilde{\mathcal{J}}_k(\hat{x}_{k+1})$ is equal to the mean will under regularize the problems about half of the time. To avoid under-regularization, we suggest using the upper bound of the desired confidence interval for the χ^2 distribution. For example, if the desired confidence level is 95%, then this upper bound is the number at which a correctly weighted $\tilde{\mathcal{J}}_k(\hat{x}_{k+1})$ will be less than or equal to 95% of the time. We suggest choosing the regularization parameter such that $\tilde{\mathcal{J}}_k(\hat{x}_{k+1})$ is equal to this number. This approach is implemented in Algorithm 1, which uses the Gauss-Newton method to solve the nonlinear inverse problem and dynamically estimates the regularization

parameter at each Gauss-Newton iteration.

Algorithm 1 Nonlinear χ^2 method with Gauss-Newton step

Input $L, C_\varepsilon, x_p, \text{tol}$

for $k=1,2,3,\dots$ **do**

 Calculate J_k and \tilde{d}_k

 Define:

$$\tilde{\mathcal{J}}_k(x, \alpha) = \|\tilde{d}_k - J_k x\|_{C_\varepsilon^{-1}}^2 + \alpha^2 \|L(x - x_p)\|_2^2$$

 Choose α_k such that:

$$\tilde{\mathcal{J}}_k(x, \alpha_k) \approx \Phi_{n-m+q}^{-1}(95\%)$$

 where Φ_{n-m+q}^{-1} is the inverse cumulative distribution function (CDF) of χ_{n-m+q}^2 .

$$\begin{aligned} \hat{x}_{k+1} &= \arg \min_x \tilde{\mathcal{J}}_k(x, \alpha_k) \\ &= (J_k^T C_\varepsilon^{-1} J_k + \alpha_k^2 L^T L)^{-1} (J_k^T C_\varepsilon^{-1} \tilde{d}_k + \alpha_k^2 L^T L x_p) \end{aligned}$$

if $\frac{|\mathcal{J}_M(\hat{x}_{k+1}) - \mathcal{J}_M(x_k)|}{\mathcal{J}_M(x_k)} < \text{tol}$ **then**

 converged and $x_M = \hat{x}_{k+1}$

return

end if

end for

When the Gauss-Newton method (3.15) fails to converge, the Levenberg-Marquardt method can be used to solve the inverse problem. Algorithm 2 uses the regularized Levenberg-Marquardt method from (3.29) and estimates the regularization parameter using the approximate χ^2 distribution derived in Section 3.2.1.

Algorithm 2 Nonlinear χ^2 method with LM step

 Input $L, C_\varepsilon, x_p, \lambda_1, D$
for $k=1,2,3\dots$ **do**

 Calculate J_k and \tilde{d}_k
if $k > 1$ **then**

Define:

$$\begin{aligned}\tilde{\mathcal{J}}_k^{\text{LM}}(x, \lambda) &= \|\tilde{d}_k - J_k x\|_{C_\varepsilon^{-1}}^2 + \alpha_k^2 \|L(x - x_p)\|_2^2 + \lambda^2 \|D(x - x_k)\|_2^2 \\ x_{k+1}^{\text{LM}} &= \arg \min_x \tilde{\mathcal{J}}_k^{\text{LM}}(x, \lambda) \\ &= (J_k^T C_\varepsilon^{-1} J_k + \alpha_k^2 L^T L + \lambda_k^2 D)^{-1} (J_k^T C_\varepsilon^{-1} \tilde{d}_k + \alpha_k^2 L^T L x_p + \lambda_k^2 x_k)\end{aligned}$$

 Update LM parameter by finding a small λ_{k+1} that still ensures

$$\mathcal{J}_M(x_{k+1}^{\text{LM}}) < \mathcal{J}_M(x_k^{\text{LM}})$$

end if

Define:

$$\begin{aligned}\tilde{\mathcal{J}}_k^{\text{LM}}(x, \alpha) &= \|\tilde{d}_k - J_k x\|_{C_\varepsilon^{-1}}^2 + \alpha^2 \|L(x - x_p)\|_2^2 + \lambda_k^2 \|D(x - x_k)\|_2^2 \\ \tilde{\mathcal{J}}_k(x, \alpha) &= \|\tilde{d}_k - J_k x\|_{C_\varepsilon^{-1}}^2 + \alpha^2 \|L(x - x_p)\|_2^2\end{aligned}$$

 Choose α_{k+1} such that:

$$\tilde{\mathcal{J}}_k(x_{k+1}^{\text{LM}}, \alpha_{k+1}) \approx \Phi_{n-m+q}^{-1}(95\%)$$

 where $\Phi_{n-m+q}^{-1}()$ is the inverse CDF of χ_{n-m+q}^2 .

$$x_{k+1}^{\text{LM}} = \arg \min_x \tilde{\mathcal{J}}_k^{\text{LM}}(x, \alpha_{k+1}):$$

if $\frac{|\mathcal{J}_M(x_{k+1}^{\text{LM}}) - \mathcal{J}_M(x_k^{\text{LM}})|}{\mathcal{J}_M(x_{k+1}^{\text{LM}})} < \text{tol}$ **then**

 converged and $x_M = x_{k+1}^{\text{LM}}$
return
end if
end for

Algorithm 2 has the additional complication of determining the LM parameter λ_k . This can be found using the methods from the LM implementations discussed in Chapter 2. This algorithm has more computational overhead than the previous algorithm, but this is simply the price for a more robust method that is needed to solve more difficult problems.

3.3.1 Numerical Implementation of Algorithms

In Algorithm 1 and 2, it is necessary to do some type of line search at each iteration to find α_{k+1} . To do this, any standard root finding algorithm can be used such as the bisection method, inverse quadratic interpolation, secant method, or Newton's method. In [10], the authors introduce an exact Newton root-finding algorithm that uses SVD of the linear inverse problem which would work for Algorithm 1.

In Algorithm 1 and 2, the Gauss-Newton method and the Levenberg-Marquardt method are written as a sequence of OLS problems. Since the algorithms require that these OLS problems to be solved multiple times at each iteration, it is important that the OLS solution is computed in an efficient matter. We saw in Chapter 2 that the OLS estimate is given as:

$$\begin{aligned}\hat{x} &= \arg \min_x \|d - Ax\|_2^2, \\ \hat{x} &= (A^T A)^{-1} A^T d.\end{aligned}\tag{3.41}$$

In practice, $(A^T A)^{-1}$ should never be computed as this can be computationally expensive and is not stable with respect to round off errors because $\text{cond}(A^T A) \approx \text{cond}(A)^2$. There are a number of efficient methods for solving OLS problems. We use the backslash operator (same function as `mldivide`) in MATLAB, which uses a robust

implementation of QR factorization to solve overdetermined problems [11].

Also, in many of the previous equations the inverse of the square root of the covariance matrix is taken in order to factor the steps into an OLS problem. When the matrix is a diagonal matrix this operation is trivial. However, if the covariance matrices have nonzero off-diagonal elements, taking the matrix square root can be expensive and unstable. Instead of taking the matrix square root, we can split the matrix with Cholesky factorization, which can be more accurate and computationally cheaper. The following example shows why this is valid. If we have

$$d = Ax + \varepsilon \quad \varepsilon \sim N(0, C_\varepsilon) \quad (3.42)$$

and if R represents the Cholesky decomposition of C_ε , i.e. $RR^T = C_\varepsilon$, then we can normalize the problem using R :

$$R^{-1}d = R^{-1}Ax + R^{-1}\varepsilon. \quad (3.43)$$

If we let $\tilde{A} = R^{-1}A$, $\tilde{d} = R^{-1}d$, and $\tilde{\varepsilon} = R^{-1}\varepsilon$, then Theorem 7 in Appendix A implies: $\tilde{\varepsilon} \sim N(0, I_n)$ and (3.43) becomes the normalized OLS problem:

$$\tilde{d} = \tilde{A}x + \tilde{\varepsilon} \quad \tilde{\varepsilon} \sim N(0, I_n) \quad (3.44)$$

CHAPTER 4

APPLICATION, SOLUTIONS, AND NUMERICAL EXPERIMENTS

In this chapter, we consider two ill-posed nonlinear inverse problems and use these problems to test the theorems and algorithms developed in Chapter 3. These test problems are from Chapter 10 of [1] where the authors both describe problems and provide the corresponding solutions. Conveniently, the authors included Matlab codes along with the text that set up the forward problem and solve the inverse problems with existing methods. This allowed us to recreate their results and use them as a basis for comparison.

These problems are a 2-D nonlinear cross-well tomography problem and a 1-D electromagnetic sounding problem. The mathematical models for both of these systems are quite involved and are not developed in this thesis. Instead, the forward models from the codes provided by [1] are treated as the proverbial “black box” functions and the Jacobians of the functions are calculated using finite differences. Using these “functions,” we run some numerical experiments to see if Theorems 3 and 4 hold under the necessary assumptions. In addition, we compare the solutions found using Algorithms 1 and 2 to solutions found by [1] using existing methods.

4.1 Nonlinear Cross-Well Tomography

The first problem is an implementation of nonlinear cross-well tomography. The forward model includes ray path refraction where the refracted rays tend to travel through high-velocity regions and avoid low-velocity regions, which adds nonlinearity to the problem. The problem is set up with two wells spaced 1600 m apart, and there are pairs of sources and receivers at equally spaced depths down the wells. The travel time between each pair of opposing sources and receivers is recorded, and the objective is to recover the two-dimensional velocity structure between the two wells. The true velocity structure has a background of 2.9 km/s with an embedded Gaussian shaped region that is about 10% faster than the background and another Gaussian-shaped region that is about 15% slower. The observations for this particular problem consist

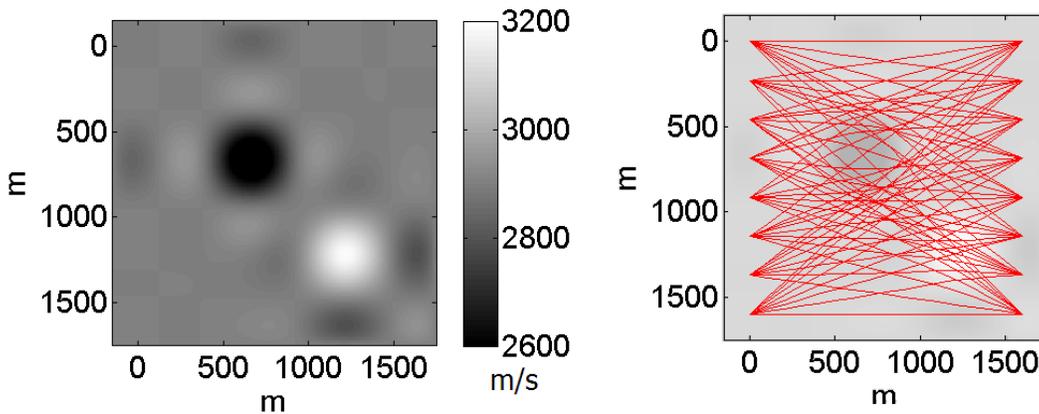


Figure 4.1: The setup of the cross-well tomography problem. (Left) Shown here is the true velocity model(m/s). (Right) The ray paths crossing through region of interest (background is faded to make the ray paths more clear).

of 64 travel times between each pair of opposing sources and receivers. The true velocity model along with the 64 ray paths are plotted in Figure 4.1. The faster regions are represented by the lighter areas and the slower regions are darker. The entire region between the two wells is discretized into 64 square blocks so there are

64 model parameters (the slowness of each block) and 64 observations (the ray path travel times).

4.1.1 Numerical Experiments

In Chapter 3, it was shown that the Gauss-Newton method can be written as the sequence of linear inverse problems:

$$\begin{aligned} \tilde{\mathcal{J}}_k(x) &= \|\tilde{d}_k - J_k x\|_{C_\varepsilon^{-1}}^2 + \|x - x_p\|_{C_f^{-1}}^2 \\ x_{k+1} &= \arg \min_x \tilde{\mathcal{J}}_k(x) \end{aligned} \tag{4.1}$$

and the assertion was made that $\tilde{\mathcal{J}}_k(x_{k+1})$ approximately follows a χ_n^2 distribution. To test this assertion, we carried out the following numerical experiment. First, we generated a set of synthetic data from an initial parameter set. Then we added 1000 different realizations of ε and f to the synthetic data and initial parameters, respectively. The added noise ε was sampled from $N(0, (.001)^2 \mathbf{I}_{64})$ and f from $N(0, (.00001)^2 \mathbf{I}_{64})$. For perspective, the values for d are $\mathcal{O}(.1)$ and the values for x are $\mathcal{O}(.0001)$. This means the data had about 1% noise added and the initial parameter estimate had 10% noise added. We then used the Gauss-Newton method to solve the nonlinear inverse problem 1000 times, once for each realization of noise. Essentially, this is equivalent to sampling $\tilde{\mathcal{J}}_k(x_{k+1})$ 1000 times. Each of these converged in 6 iterations. A histogram of these samples of $\tilde{\mathcal{J}}_k(x_{k+1})$ at each iteration is given below in Figure 4.2. Since there are 64 observations and 64 model parameters, the theory says that $\tilde{\mathcal{J}}_k \sim \chi_{64}^2$ and so $E(\tilde{\mathcal{J}}_k(x_{k+1})) = 64$.

In [1] the authors use a discrete approximation of the Laplacian operator $\tilde{\Delta}$ to regularize this problem. So if $L = \tilde{\Delta}$ and $Lx_p = 0$, then the Gauss-Newton method

becomes:

$$\begin{aligned}\tilde{\mathcal{J}}_k(x) &= \|\tilde{d}_k - J_k x\|_{C_\varepsilon^{-1}}^2 + \|Lx - 0\|_{C_f^{-1}}^2, \\ x_{k+1} &= \arg \min_x \tilde{\mathcal{J}}_k(x).\end{aligned}\tag{4.2}$$

Using this operator and the same assumptions as above, we found approximate distributions of $\tilde{\mathcal{J}}_k$ when $L = \tilde{\Delta}$. The approximate distributions found are shown in Figure 4.2. In the top-left histogram in Figure 4.2 the sampled distribution of $\tilde{\mathcal{J}}_k(x_{k+1})$ for the first iteration is shifted slightly right of the theoretical χ^2 distribution predicted in Theorem 3. This is what would be expected if C_ε underestimates C_{ε_k} . However, by the second iteration, the two distributions are almost identical as the nonlinear error is decreased. In fact, in each of the other histograms shown in Figure 4.2, the sampled distribution agrees very well with the theoretical distribution. This indicates that the theoretical χ^2 distributions established in Theorem 3 are a good approximation to the actual distributions.

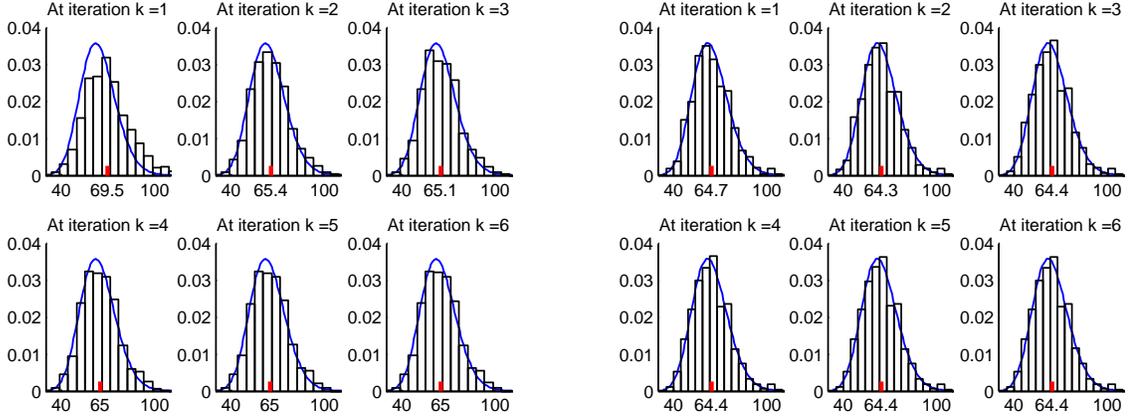


Figure 4.2: Histograms of $\tilde{\mathcal{J}}_k(x_{k+1})$. (Left): $\tilde{\mathcal{J}}_k$ with $L = I$, (Right): $\tilde{\mathcal{J}}_k$ with $L = \tilde{\Delta}$. The mean of the sample is shown as the middle tick, and χ_{64}^2 probability density function is shown as the solid blue line.

4.1.2 Inversion Results

In [1] the authors solve the nonlinear tomography problem by minimizing

$\mathcal{J}_{\mathcal{M}}(x) = \|d - F(x)\|_{C_{\varepsilon}^{-1}}^2 + \alpha^2 \|Lx - 0\|^2$, where $L = \tilde{\Delta}$ and the discrepancy principle is used to estimate α . They implemented the discrepancy principle as finding α so that $\mathcal{J}_{\text{data}}(x_{\mathcal{M}}) = \|d - F(x_{\mathcal{M}})\|_{C_{\varepsilon}^{-1}}^2 = m$. The data used in this inversion was created by generating synthetic data from the “true” parameter set and adding a realization of random noise ε to this synthetic data. We use this same approach and data to solve the inverse problem by minimizing $\mathcal{J}_{\mathcal{M}}(x) = \|d - F(x)\|_{C_{\varepsilon}^{-1}}^2 + \alpha^2 \|L(x - x_p)\|^2$, where $L = I$. We did this to create another case to which to compare the χ^2 method. Using the same data set, we found solutions using Algorithm 1 from Chapter 3 with both $L = \tilde{\Delta}$ and for $L = I$. The solutions found with $L = \tilde{\Delta}$ and the discrepancy principle are plotted in Figure 4.3 next to the solution found with $L = \tilde{\Delta}$ and Algorithm 1. The plot of the solution found with Algorithm 1 in Figure 4.3 is very similar to

the solution found with the discrepancy principle. The solutions found with both methods using $L = I$ are plotted next to each other in Figure 4.4. These are also very similar to each other. It is evident from these figures that solutions found with $L = \tilde{\Delta}$ are smoother than the solutions found with $L = I$.

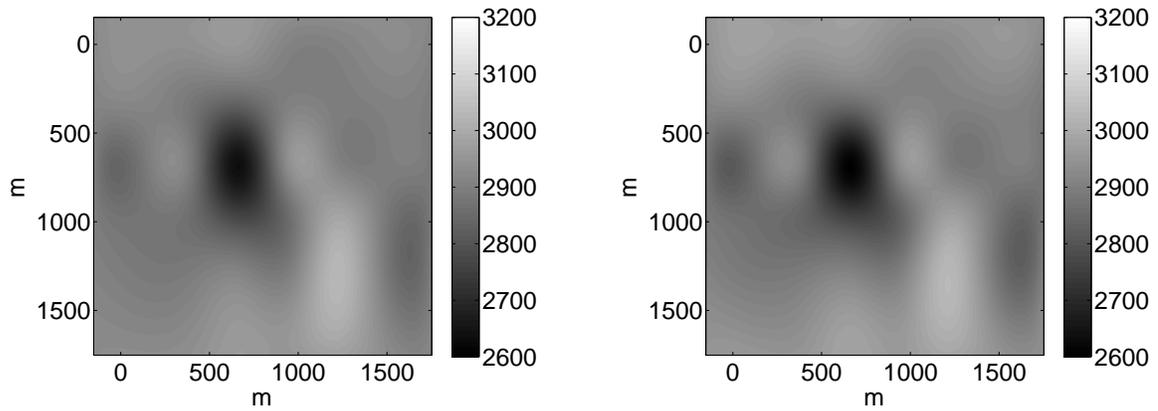


Figure 4.3: Solutions found for the tomography problem with $L = \tilde{\Delta}$ (Left) Solution found using the discrepancy principle. (Right) Solution found with Algorithm 1.

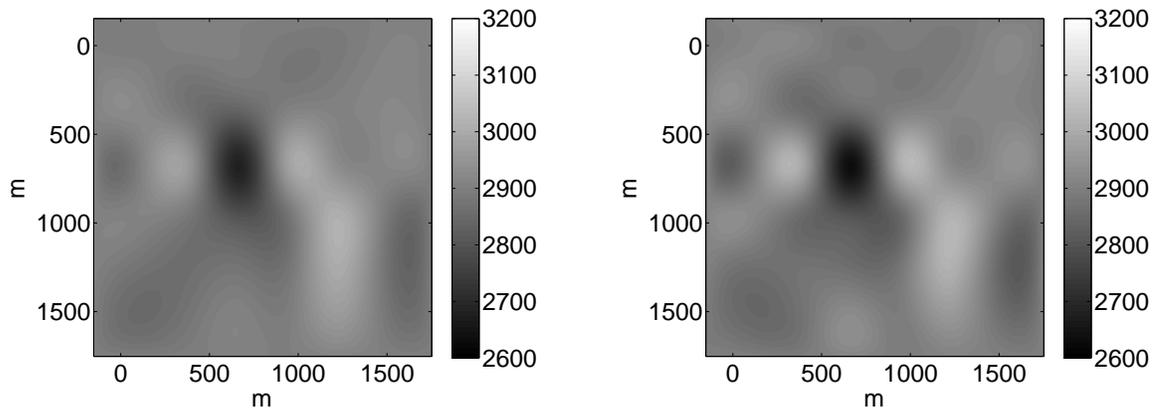


Figure 4.4: Solutions found for the tomography problem when $L = I$. (Left) Solution found using the discrepancy principle. (Right) Solution found with Algorithm 1.

Figures 4.3 and 4.4 are the results from only one realization of ε . In order to establish a good comparison, the above procedure for both the discrepancy principle and for Algorithm 1 was repeated for 200 different realizations of ε . The mean and standard deviation of $\|x_M - x_{true}\|/\|x_{true}\|$ for the 200 trials for each method are given below in Table 4.1. Using this as the basis for comparison, χ^2 method gave better results on average when $L = I$. But the discrepancy principle did better on average when the regularizing operator $L = \tilde{\Delta}$. While these differences are only incremental, the χ^2 method was faster computationally because it only solves the inverse problem once and dynamically estimates α_k . The discrepancy principle requires the inverse problem to be solved multiple times, incurring computational cost. In this test problem, we replaced the brute line search in the code for the discrepancy principle with a secant iteration, which typically converged in 6 or 7 iterations. The inner iteration typically converged in same number of iterations as Algorithm 1. So the discrepancy principle required 6 or 7 times more forward function evaluations than Algorithm 1. However, Algorithm 1 does a search at each iteration to estimate α_k , which doesn't require more forward function evaluations but does add some computational cost. The net result was that Algorithm 1 was about three times faster in terms of wall-clock time.

Table 4.1: Comparison of discrepancy principle to χ^2 method on the cross-well tomography problem, $\mu = mean(\|x_M - x_{true}\|/\|x_{true}\|)$, $\sigma = sqrt(var(\|x_M - x_{true}\|/\|x_{true}\|))$

Method	$L = I$	$L = \tilde{\Delta}$
χ^2 Method	$\mu = 0.01628$ $\sigma = 0.0006$	$\mu = 0.0206$ $\sigma = 0.00456$
Discrepancy Principle	$\mu = 0.01672$ $\sigma = 0.00050$	$\mu = 0.018$ $\sigma = 0.0021$

4.2 Subsurface Electrical Conductivity Estimation

The second problem considered is the estimation of soil electrical conductivity profile from above-ground electromagnetic induction measurements. The forward problem models a Geonics EM38 ground conductivity meter that has two coils on a 1 meter long bar. Alternating current is sent in one of the coils which induces currents in soil and both coils measure the magnetic field that is created by the subsurface currents. For a complete treatment of the instrument and corresponding mathematical model see [5]. Measurements are taken at 9 different heights above the soil and with two different orientations of the instrument, resulting in a total of 18 observations. The subsurface electrical conductivity of the ground is discretized into 10 layers, 20 cm thick, with a semi-infinite layer below 2m, resulting in 11 conductivities to be estimated. An illustration of the setup of this problem is given in Figure 4.5.

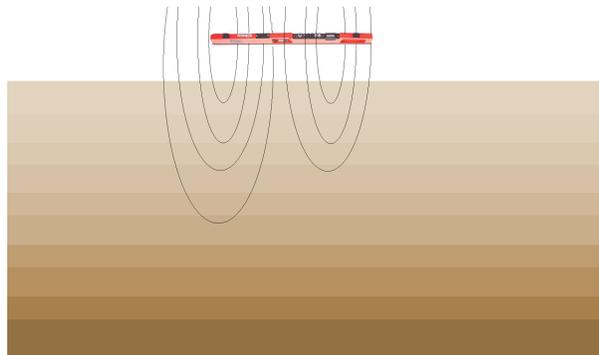


Figure 4.5: A representation of the soil-conductivity estimation. The instrument depicted in the top of image represents a ground conductivity meter creating a time-varying electromagnetic field in the layered earth beneath.

4.2.1 Numerical Experiments

We found in solving this inverse problem, the Gauss-Newton method does not always converge. Therefore, finding the solution necessitated the use of the Levenberg-Marquardt algorithm. This provided the opportunity to test the validity of the approximations discussed in Section 3.2.1. In that section, it was shown that $\tilde{\mathcal{J}}_k(x_{k+1}^{\text{LM}}) = \|\tilde{d}_k - J_k x_{k+1}^{\text{LM}}\|_2^2 + \|x_{k+1}^{\text{LM}} - x_p\|_2^2$ approximately follows χ_n^2 distribution. Once again we ran some numerical experiments to test this. In a similar way as before, we generated a synthetic data set from a set of parameters and then added 1000 realizations of noise to this data and parameters. This added noise ε was sampled from $N(0, (1)^2 \mathbf{I}_{18})$ and f from $N(0, (100)^2 \mathbf{I}_{11})$. For perspective, the values for d are $\mathcal{O}(100)$ and the values for x are $\mathcal{O}(100)$. This means the data had about 1% noise added and the initial parameter estimate had 100% noise added. We then used the Levenberg-Marquardt method to solve the regularized nonlinear inverse problem 1000 times for each realization of noise and recorded the samples of $\tilde{\mathcal{J}}_k(x_{k+1}^{\text{LM}})$. All of these LM iterations converged within 6 iterations. Histograms of these 1000 samples of $J_k(x_{k+1}^{\text{LM}})$ are shown below in Figure 4.6. There were 18 observations for this problem and 11 parameters so $E(\tilde{\mathcal{J}}_k(x_{k+1}^{\text{LM}})) = 18$.

In [1] the authors solve this inverse problem using an approximate 2nd order differential operator $L = \tilde{D}^{(2)}$ to regularize the inversion. We carried out the same experiment as above, except using this operator to regularize the problem. This matrix L has dimension 9×11 so $E(\tilde{\mathcal{J}}_k(x_{k+1}^{\text{LM}})) = 16$. Also, in this experiment, f was sampled from $N(0, (10)^2 \mathbf{I}_9)$ since the elements of Lx are $\mathcal{O}(10)$. These LM iterations converged in 5 iterations. Histograms of these 1000 samples of $J_k(x_{k+1}^{\text{LM}})$ are shown in Figure 4.6. The histograms of the samples of $J_k(x_{k+1})$ in Figure 4.6

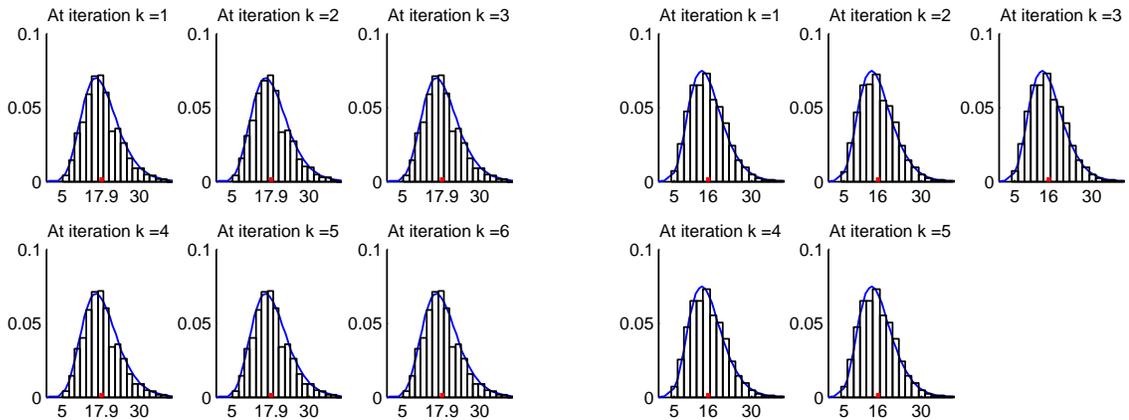


Figure 4.6: Histograms of $\tilde{\mathcal{J}}_k(x_{k+1})$. (Left) $\tilde{\mathcal{J}}_k$ with $L = I$, (Right) $\tilde{\mathcal{J}}_k$ with $L = \tilde{D}^{(2)}$. The mean of the sample is shown as the middle tick, and the χ_{18}^2 , χ_{16}^2 density functions are shown as the solid blue line.

coincided closely with the theoretical χ^2 distributions also plotted. This suggests that the approximations used in Section 3.2.1 are good approximations, at least for this problem.

4.2.2 Inversion Results

In some ways, this inverse problem is more difficult than the cross-well tomography problem. The Gauss-Newton step doesn't always lead to a reduction in the non-linear cost function and it is not always possible to find a regularizing parameter for which the solution satisfies the discrepancy principle. In [1] the authors used an implementation of the Levenberg-Marquardt algorithm to minimize the unregularized least squares problem to demonstrate the ill-posedness of this problem. This solution, plotted below in Figure 4.7, is wildly oscillating, has extreme values and is not even close to being a physically possible solution. However, this isn't evident from just looking at the data misfit as this solution actually fits the data quite well.

The inverse problem was also solved in [1] using Occam's Inversion method. Occam's Inversion is given as the following algorithm.

Algorithm 3 Occam's Inversion

Start with an initial estimate x_p

for $k=1,2,3,\dots$ **do**

Define $x_{k+1}^{\text{oc}} = (J_k^T C_\varepsilon^{-1} J_k + \alpha_k^2 L^T L)^{-1} (J_k^T C_\varepsilon^{-1} \tilde{d}_k)$

Choose largest value of α_k such the $\|d - F(x_{k+1}^{\text{oc}})\|_{C_\varepsilon^{-1}} \leq m$

If no such α_k exists, then chose a α_k that minimizes $\|d - F(x_{k+1}^{\text{oc}})\|_{C_\varepsilon^{-1}}$

Stop when $\|d - F(x_{k+1}^{\text{oc}})\|_{C_\varepsilon^{-1}} = m$

end for

In this implementation of Occam's Inversion, L was chosen to be $\tilde{D}^{(2)}$. As in the previous problem, the data used in this inversion was created by generating synthetic data from the "true" parameter set and adding a realization of random noise ε to this synthetic data. The solution found using this algorithm and data set is plotted in Figure 4.7. We implemented Occam's inversion using $L = I$ to solve this problem, however, the algorithm diverged with this choice for L .

We used the same data set and Algorithm 2 to find the solution using both $L = I$ and $L = \tilde{D}^{(2)}$. These solutions are plotted below in Figure 4.8. Comparing the solutions found using $L = \tilde{D}^{(2)}$ for the both the χ^2 method and Occam's inversion in Figures 4.7 and 4.8, it is apparent that both estimate the true solution fairly well for this realization of ε . While the χ^2 method was still able to find a solution with $L = I$, it can be seen in Figure 4.8 that it doesn't estimate the true solution as well.

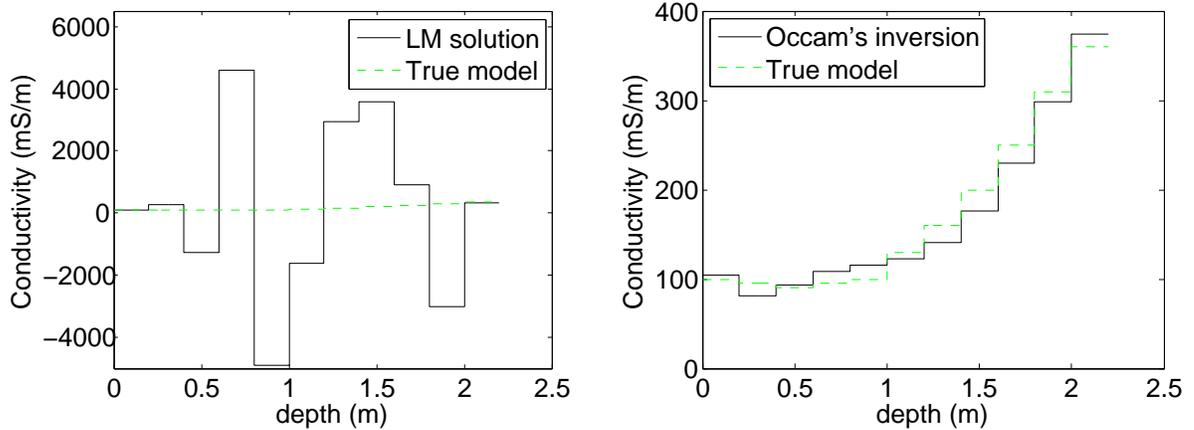


Figure 4.7: (Left) The unregularized solution. (Right) The solution found with Occam's inversion.

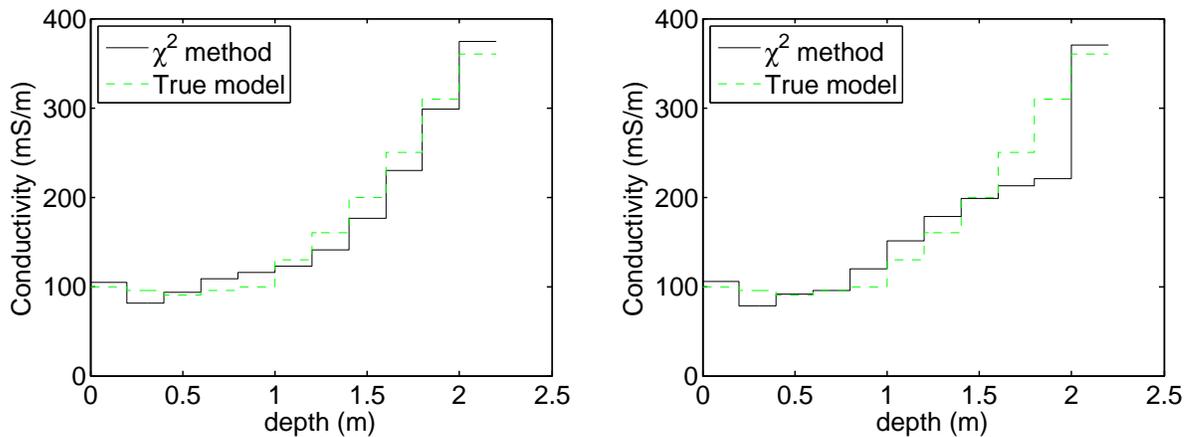


Figure 4.8: The parameters found using the χ^2 method. (Left) $L = \tilde{D}^{(2)}$ (Right) $L = I$.

Once again, in order to establish a good comparison, each of these methods were run for 200 different realizations of ε . The mean and standard deviation of $\|x_M - x_{true}\|/\|x_{true}\|$ for the 200 trials for each method are given below in Table 4.1. While

Occam's inversion was able to find good solutions for some realization of ε , such as the solution plotted in Figure 4.7, the results in Table 4.1 indicate that sometimes it found poor estimates. The mean of $\|x_M - x_{true}\|/\|x_{true}\|$ for the χ^2 method is almost an order or magnitude smaller than for Occam's inversion for this problem. Even the χ^2 method with $L = I$ found better solutions on average. Also, the relatively small values for σ in Table 4.2 for the χ^2 method suggest that the solutions found were fairly consistent with each other. Conversely, the large value of σ in Table 4.1 for Occam's inversion indicates that these solutions were not consistent with each other. Since both methods estimate the regularization parameter dynamically, the computational cost should be about the same and both methods took about the same speed in terms of wall-clock time.

Table 4.2: Comparison of the χ^2 method to Occam's inversion for the estimation of subsurface conductivities, $\mu = \text{mean}(\|x_M - x_{true}\|/\|x_{true}\|)$, $\sigma = \text{sqrt}(\text{var}(\|x_M - x_{true}\|/\|x_{true}\|))$

Method	$L = I$	$L = \tilde{D}^{(2)}$
χ^2 Method	$\mu = 0.1827$	$\mu = 0.0308$
	$\sigma = 0.0295$	$\sigma = 0.0281$
Occam's Inversion	Diverged	$\mu = 0.4376$
		$\sigma = 0.6615$

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

We presented a method regularizing nonlinear inverse problems that we call the nonlinear χ^2 method. This approach uses statistical information about the data to determine the proper level of regularization and is an extension of the linear χ^2 method proposed by Mead in [8]. The χ^2 tests used in the linear χ^2 method were extended to nonlinear problems in Section 3.2. The χ^2 method was extended to nonlinear problems using the Gauss-Newton method and the Levenberg-Marquardt method in Algorithms 1 and 2, respectively. We gave numerical results in Sections 4.1.1 and 4.2.1 illustrating the statistical theory developed in Chapter 3 and demonstrated that it was valid for two complex nonlinear problems

Two new algorithms were implemented on two nonlinear problems from [1] and compared against several existing methods for nonlinear regularization. It was shown that Algorithm 1 provided parameter estimates that were of similar accuracy as the discrepancy principle in a nonlinear cross-well tomography problem from [1]. In a subsurface electrical conductivity problem from [1], Algorithm 2 proved to be more robust than Occam's inversion, providing parameter estimates without the use of a smoothing operator. Algorithm 2 also provided much better estimates than Occam's inversion on average when the smoothing operator was used.

The high computational cost of the first forward problems should be considered

and this is where the χ^2 method prevails. The discrepancy principle solves the nonlinear inverse problem several times for different regularization parameters and thus it requires more forward model evaluations, making it computationally expensive. The nonlinear χ^2 method is cheaper because it only solves the inverse problem once and dynamically updates the regularization parameter.

We conclude that the nonlinear χ^2 method is an attractive alternative to the discrepancy principle and Occam's inversion. However, it does share a disadvantage with these methods in that they all require the covariance of the data to be known. If an estimate of the data covariance is not known, then the nonlinear χ^2 method will not be appropriate for solving such a problem. Future work includes estimating more complex covariance matrices for the parameter estimates. In [9], Mead shows that it is possible to use multiple χ^2 tests to estimate such a covariance and it seems likely that this could also be extended to solving nonlinear problems.

REFERENCES

- [1] Richard C. Aster, Brian Borchers, and Clifford H. Thurber. *Parameter Estimation and Inverse Problems*. Elsevier Academic Press, 2005.
- [2] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization*. Academic Press, 1981.
- [3] Eldad Haber and Douglas Oldenburg. A GCV based method for nonlinear ill-posed problems. *Computational Geosciences*, 4(1):41–63, 2000.
- [4] Per Christian Hansen. *Discrete Inverse Problems, Insight and Algorithms*. SIAM, 2010.
- [5] B; Corwin D L; Lesch S M; et al Hendrickx, J M H; Borchers. Inversion of soil conductivity profiles from electromagnetic induction measurements: Theory and experimental verification. *Soil Science Society of America Journal*, 66(3):673–685, 2002.
- [6] Enting Ian G. *Inverse Problems in Atmospheric Constituent Transport*. Cambridge University Press, 2002.
- [7] John M. Lewis, S. Lakshmivarahan, and Sudarshan Dhall. *Dynamic Data Assimilation*. Cambridge University Press, 2006.
- [8] Jodi L Mead. Parameter estimation: A new approach to weighting a priori information. *Journal of Inverse and Ill-Posed Problems*, 16(2):175–194, 2008.

- [9] Jodi L Mead;. Discontinuous parameter estimates with least squares estimators. *Applied Mathematics and Computation*, In Revision, 2011.
- [10] Jodi L Mead and Rosemary A Renault. A Newton root-finding algorithm for estimating the regularization parameter for solving ill-conditioned least squares problems. *Inverse Methods*, 2009.
- [11] Cleve Moler. *Numerical Computing with Matlab*. SIAM, 2006.
- [12] Sybil P Parker. *McGraw-Hill dictionary of scientific and technical terms*. New York : McGraw-Hill, 1994.
- [13] G. A. F. Seber and C. J. Wild. *Nonlinear Regression*. Wiley, 2003.
- [14] George A.F. Seber and Alan J. Lee. *Linear Regression Analysis*. John Wiley and Sons, Inc, 2003.
- [15] Albert Tarantola. *Inverse Problem Theory, and Methods for Model Parameter Estimation*. SIAM, 2005.

APPENDIX A

ADDITIONAL THEOREMS

For the convenience of the reader, we include some distribution theory and linear algebra that was used in the proofs of the Theorems 1, 2, 3, and 4 in Chapters 2 and 3. Theorems 5, 6, and 7 are from [14]. The last theorem listed here, Theorem 8, is an important theorem that gives χ^2 distribution of a variable that arises in the proof of Theorem 1 in Chapter 2. Since understanding Theorem 8 is helpful in establishing an intuitive understanding of much of the χ^2 theory presented in this thesis, its proof is included here.

Theorem 5. *If P is symmetric and idempotent matrix then $\text{rank}(P) = \text{trace}(P)$. (Theorem A.6.2 [14])*

Theorem 6. *Let A be a symmetric matrix. Then A has r eigenvalues equal to 1 and the rest equal to zero iff $A^2 = A$ and $\text{rank } A=r$. (Theorem 2.7 [14])*

Theorem 7. *Let Y be normal random vector with dimension $n \times 1$ with mean μ and variance Σ , i.e. $Y \sim N(\mu, \Sigma)$, and let C be an $m \times n$ matrix of rank m and d be an $m \times 1$ vector. Then $(CY + d) \sim N(C\mu + d, C\Sigma C^T)$. (Theorem 2.2 [14])*

Theorem 8. *Let Y be normal random vector with dimension $n \times 1$ with mean 0 and variance \mathbf{I}_n , i.e. $Y \sim N(0, \mathbf{I}_n)$ and let A be a $n \times n$ symmetric idempotent matrix with rank r , then $Y^T A Y \sim \chi_r^2$.*

Proof. Since A is symmetric, it can be written in terms of its spectral decomposition: $A = T^T D T$ where D is a diagonal matrix whose entries are the eigenvalues of A and T is an orthogonal matrix. Then $Y^T A Y = Z^T D Z$, where $Z = T^T Y$. By Theorem 7, $Z \sim N(0, \mathbf{I}_n)$. Since A is symmetric, idempotent, and with rank r , Theorem 6 implies that A has r unit eigenvalues and the rest are zero. So $Y^T A Y = Z^T D Z = \sum_{i=1}^r T_i^2$. Thus $Y^T A Y$ is equal to the sum of r squared standard normal random variables, so $Y^T A Y \sim \chi_r^2$ □

