A Comparison of Two Methods for Recovering the Release History of a Groundwater Contamination Source

by

Roseanna Marie Neupauer

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science in Mathematics
with Operations Research and Statistics Option

New Mexico Institute of Mining and Technology
Socorro, New Mexico
May, 1999

ACKNOWLEDGEMENT

This research was supported in part by the Environmental Protection Agency's STAR Fellowship program under Fellowship No. U-915324-01-0. This work has not been subjected to the EPA's peer and administrative review and therefore may not necessarily reflect the views of the Agency and no official endorsement should be inferred.

Todd Skaggs and Zbigniew Kabala provided their code and input files for Tikhonov regularization. Allan Woodbury provided information about his implementation of the minimum relative entropy inversion method.

I am grateful for the guidance given to me by my advisor, Brian Borchers, and my committee members, Allan Gutjahr, John Wilson, and Bill Stone; and I appreciate their valuable comments on this thesis.

This thesis was typeset with LaTeX[1] by the author.

[1] LaTeX document preparation system was developed by Leslie Lamport as a special version of Donald Knuth's TeX program for computer typesetting. TeX is a trademark of the American Mathematical Society. The LaTeX macro package for the New Mexico Institute of Mining and Technology thesis format was adapted by Gerald Arnold from the LaTeX macro package for The University of Texas at Austin by Khe-Sing The.

ABSTRACT

Determining the release history of a source of groundwater contamination is necessary to assign liability for remediation costs. Inverse methods can be used to reconstruct the release history from spatially- or temporally-distributed measurements of the concentration of the contaminant in the aquifer. The independent application of several inverse methods to solve this problem has been discussed in recent literature. We compare the effectiveness of two of those methods, Tikhonov regularization and minimum relative entropy inversion, in reconstructing the source release history. We evaluate the methods assuming perfect data and perfect knowledge of the transport processes, and then add complications that arise in field situations. Two different source history functions are evaluated—a smooth, Gaussian-shaped function, and a step function. The results show that minimum relative entropy inversion reproduces the source history more effectively than Tikhonov regularization when the step function is used. Also, when the data contain measurement error, both methods perform equally well if the noise level is known exactly; however, if the noise level is underestimated, Tikhonov regularization performs better than minimum relative entropy inversion. In all other situations addressed in this research, the results of the two methods are essentially indistinguishable.

Table of Contents

Acknowledgement ii

Abstract

Table of Contents iii

List of Tables vi

List of Figures viii

1. Introduction 1

1.1 Problem Statement 2

1.2 Objective 3

1.3 Scope 4

1.4 Background on Contaminant Transport in Groundwater 5

1.4.1 Forward Advection-Dispersion Problem 7

1.4.2 Inverse Advection-Dispersion Problem 8

1.5 Literature Review on Source History Problems 9

2. Inverse Methods 18

2.1 Regularization Methods 20

2.1.1 Overview of Regularization Methods 21

2.1.2 Tikhonov Regularization 22

2.2 Maximum Entropy Methods 28

2.2.1 Overview of Maximum Entropy Methods 29

2.2.2 Minimum Relative Entropy Inversion 30

2.3 Comparison of Tikhonov Regularization and Minimum Relative Entropy Inversion 36

3. Application of Inverse Methods to the Source History Reconstruction Problem 40

3.1 Source History Reconstruction using Tikhonov Regularization . 41

3.1.1 Implementation of Tikhonov Regularization 41

3.1.2 Verification of Tikhonov Regularization Routine 42

3.2 Source History Reconstruction using Minimum Relative Entropy Inversion 50

3.2.1 Implementation of Minimum Relative Entropy Inversion 50

3.2.2 Verification of Minimum Relative Entropy Routine . . . 61

3.3 Summary of Implementation and Verification 72

4. Evaluation of Parameter Selection for the Inverse Methods 73

4.1 Evaluation of Parameter Selection for Tikhonov Regularization . 73

4.1.1 Evaluation of the Regularization Parameter Selection . . 74

4.1.2 Evaluation of the Order of Regularization 81

4.2 Evaluation of Parameter Selection for Minimum Relative Entropy Inversion 90

4.2.1 Evaluation of the Expected Value Function 90

4.2.2 Evaluation of the Upper Bound 96

5. Comparison of Tikhonov Regularization and Minimum Relative Entropy Inversion 103

5.1 Results of Simulations of the Ideal Scenario 104

5.2 Evaluating the Effects of Individual Factors 108

5.2.1 Measurement Error 108

5.2.2 Sampling Frequency 113

5.2.3 Errors in Transport Parameters 117

5.2.4 Smoothness of the Input Function 123

5.3 Evaluating the Effects of Multiple Factors 123

5.4 Summary of Effects of Factors 147

6. Conclusions 149

References 156

A. Determining Lagrange Multipliers for the Prior Distribution 161

B. Determining Lagrange Multipliers for the Posterior Distribution 164

C. Minimum Relative Entropy Inversion Program 168

C.1 Program Overview 168

C.2 Source Code 171

C.3 Variable Definitions 196

List of Tables

3.1 Details of Tikhonov regularization verification runs. 42

3.2 Details of MRE verification runs. 61

4.1 Test scenarios for the analysis of the regularization parameter selection methods. 75

4.2 Results of the analysis of the regularization parameter selection methods. 78

4.3 Test scenarios for the analysis of the regularization order. 83

4.4 Results of the analysis of the regularization order. 86

4.5 Test scenarios for the analysis of the prior expected value functions. 91

4.6 Results of the analysis of the prior expected value functions. . . 91

4.7 Test scenarios for the analysis of the upper bounds. 96

4.8 Results of the analysis of the upper bounds. 97

5.1 Coding chart for the fractional factorial design. 125

5.2 Design matrix for the fractional factorial design. 126

5.3 Results of multi-factor simulations. 128

5.4 Average difference of residual norms for runs with ideal and non-ideal values of each factor. 145

5.5 Main effects of the multi-factor simulations. 146

C.1 Variable definitions for MATLAB program. 196

C.2 Variable names and symbols for MATLAB program. 197

List of Figures

1.1 True source history function used by Skaggs and Kabala (1994). 11

1.2 True plume at $T = 300$ and $T = 600$ used by Skaggs and Kabala (1994). 12

2.1 Sample L-curve plot. 27

3.1 Results of Tikhonov regularization verification Run TV-1. . . . 44

3.2 Results of Tikhonov regularization verification Run TV-2. . . . 45

3.3 Results of Tikhonov regularization verification Run TV-3. . . . 46

3.4 Results of Tikhonov regularization verification Run TV-4. . . . 47

3.5 Results of Tikhonov regularization verification Run TV-5. . . . 48

3.6 Results of Tikhonov regularization verification Run TV-6. . . . 49

3.7 Results of Tikhonov regularization verification Run TV-7. . . . 51

3.8 Results of Tikhonov regularization verification Run TV-8. . . . 52

3.9 Results of Tikhonov regularization verification Run TV-9. . . . 53

3.10 Results of Tikhonov regularization verification Run TV-10. . . . 54

3.11 Results of MRE verification Run MV-1. 64

3.12 Results of MRE verification Run MV-2. 65

3.13 Results of MRE verification Run MV-3. 66

3.14 Results of MRE verification Run MV-4. 68

3.15 Results of MRE verification Run MV-5. 69

3.16 Results of MRE verification Run MV-6. 70

3.17 Results of MRE verification Run MV-7. 71

4.1 True plume and exact and inexact sample measurements for regularization parameter selection Runs RP-1 and RP-3 with a square input function. 76

4.2 True plume and exact and inexact sample measurements for regularization parameter selection Runs RP-2 and RP-4 with a smooth input function. 76

4.3 Results of generalized cross-validation for regularization parameter selection Run RP-1. 77

4.4 Tikhonov regularization results for regularization parameter selection Run RP-1. 78

4.5 Results of generalized cross-validation for regularization parameter selection Run RP-2. 79

4.6 Tikhonov regularization results for regularization parameter selection Run RP-2. 80

4.7 Results of generalized cross-validation for regularization parameter selection Run RP-3. 81

4.8 Tikhonov regularization results for regularization parameter selection Run RP-3. 82

4.9 Results of generalized cross-validation for regularization parameter selection Run RP-4. 83

4.10 Tikhonov regularization results for regularization parameter selection Run RP-4. 84

4.11 Tikhonov regularization results for regularization order Run RO-1. 85

4.12 Tikhonov regularization results for regularization order Run RO-2. 87

4.13 Tikhonov regularization results for regularization order Run RO-3. 88

4.14 Tikhonov regularization results for regularization order Run RO-4. 89

4.15 MRE results for the expected value function analysis Run EV-1. 92

4.16 MRE results for the expected value function analysis Run EV-2. 93

4.17 MRE results for the expected value function analysis Run EV-3. 94

4.18 MRE results for the expected value function analysis Run EV-4. 95

4.19 MRE results for the upper bound analysis Run UB-1. 97

4.20 MRE results for the upper bound analysis Run UB-2. 99

4.21 MRE results for the upper bound analysis Run UB-3. 100

4.22 MRE results for the upper bound analysis Run UB-4. 101

5.1 Tikhonov regularization and MRE results for the ideal scenario. 105

5.2 True plume at $T = 600$ and sampled data used in the ideal scenario. 106

5.3 Tikhonov regularization and MRE results for the ideal scenario sampled at $T = 600$. 107

5.4 Uncertainty in the MRE results for the ideal scenario sampled at $T = 600$. 108

5.5 True plume at $T = 300$ and sampled data used in the error evaluation runs. 110

5.6 Tikhonov regularization and MRE results using inexact data (error model E1). 111

5.7 Tikhonov regularization and MRE results using inexact data (error model E2). 112

5.8 True plume at $T = 300$ and sampling locations for Run SF-1. . 113

5.9 True plume at $T = 300$ and sampling locations for Run SF-2. . 114

5.10 Tikhonov regularization and MRE results with an undersampled plume (Run SF-1). 115

5.11 Tikhonov regularization and MRE results with an undersampled plume (Run SF-2). 116

5.12 Uncertainty in the MRE results with an undersampled plume (Run SF-1). 117

5.13 Uncertainty in the MRE results with an undersampled plume (Run SF-2). 118

5.14 Tikhonov regularization and MRE results with an overestimated velocity. 119

5.15 Tikhonov regularization and MRE results with an underestimated velocity. 120

5.16 Tikhonov regularization and MRE results with an overestimated dispersion coefficient. 121

5.17 Tikhonov regularization and MRE results with an underestimated dispersion coefficient. 122

5.18 Tikhonov regularization and MRE results for the ideal scenario with a non-smooth source history function. 124

5.19 Tikhonov regularization and MRE results for multi-factor Run MF-1. 129

5.20 Tikhonov regularization and MRE results for multi-factor Run MF-2. 130

5.21 Tikhonov regularization and MRE results for multi-factor Run MF-3. 131

5.22 Tikhonov regularization and MRE results for multi-factor Run MF-4. 132

5.23 Tikhonov regularization and MRE results for multi-factor Run MF-5. 133

5.24 Tikhonov regularization and MRE results for multi-factor Run MF-6. 134

5.25 Tikhonov regularization and MRE results for multi-factor Run MF-7. 135

5.26 Tikhonov regularization and MRE results for multi-factor Run MF-8. 136

5.27 Tikhonov regularization and MRE results for multi-factor Run MF-9. 137

5.28 Tikhonov regularization and MRE results for multi-factor Run MF-10. 138

5.29 Tikhonov regularization and MRE results for multi-factor Run MF-11. 139

5.30 Tikhonov regularization and MRE results for multi-factor Run MF-12. 140

5.31 Tikhonov regularization and MRE results for multi-factor Run MF-13. 141

5.32 Tikhonov regularization and MRE results for multi-factor Run MF-14. 142

5.33 Tikhonov regularization and MRE results for multi-factor Run MF-15. 143

5.34 Tikhonov regularization and MRE results for multi-factor Run MF-16. 144

Chapter 1

Introduction

Environmental contamination is a widespread problem that may affect the utility of an environmental resource such as a groundwater aquifer or a surface water body. To minimize the hazardous effects of this contamination, these media must be remediated to acceptable levels. A common problem encountered in environmental restoration is in allocating responsibility for remediation costs. If the contaminant source location is known and only one party was responsible for the contamination, that party should be held liable for the entire cost of remediation. If multiple parties contributed to the contamination, an equitable approach to assigning liability is to allocate the remediation costs among all responsible parties according to the proportion of the total contamination that was contributed by each party. Thus, to allocate the remediation costs in an equitable manner, the locations of all contaminant sources and the amounts contributed from each of these sources must be identified. Furthermore, if a contaminant source is located at a facility that has changed ownership, the release history of the source must be determined to correctly distribute the remediation costs among the owners.

In practice, contamination is often unintentional and can go undetected for many years. Therefore, the locations and release histories of the contaminant sources are often unknown. In many cases, the only available information concerning the contamination is the present spatial distribution of the contaminant concentration, or a time history of contaminant concentration at one or more locations downgradient of the (possibly unknown) sources. These concentration data can be used in an inverse model to reconstruct a likely source history or to identify possible source locations.

In this research, we study a source history reconstruction problem, with a point source of contamination at a known location in a one-dimensional flow field. The spatial distribution of the contaminant concentration is sampled at an unknown time after the initial source release. These data are used with inverse methods to reconstruct the temporal distribution of the contaminant concentration at the source.

The source history reconstruction inverse problem is ill-posed. Concentration data are sampled only at discrete points; therefore, the unique continuous concentration distribution is unknown. An infinite number of functions can fit the measured data; and therefore an infinite number of source histories can produce the measured discrete concentration distribution. The problem becomes more complicated in practice because of measurement error in the data, unknown parameter values in the transport model, and the inability to accurately model all of the transport processes.

1.1 Problem Statement

The source history reconstruction problem that is evaluated in this research has recently been addressed by several researchers. The problem involves a point source of groundwater contamination at a known location in a one-dimensional, saturated, homogeneous porous medium. A known source history is input into a forward model to obtain the spatial concentration distribution of the contaminant at some later time. These data are sampled at discrete locations, and used in the inverse problem to reconstruct a discrete (in time) release history at the source. Skaggs and Kabala (1994, 1995) solved this problem with Tikhonov regularization (TR) and the method of quasi-reversibility, respectively. Woodbury and Ulrych (1996) solved the problem using minimum relative entropy (MRE) inversion; and Snodgrass and Kitanidis (1997) used a geostatistical approach to solve the same problem. These studies are summarized in Section 1.5.

Several controversial statements have been made recently regarding the strengths and limitations of the TR and MRE methods to reconstruct the source history of a contamination source (Woodbury and Ulrych, 1996; Kabala and Skaggs, 1998; Woodbury and Ulrych, 1998b). Woodbury and Ulrych (1996) state that the MRE approach appears "qualitatively superior" to the TR approach. However, as indicated by Kabala and Skaggs (1998), the methods have not been tested using the same data sets. The methods should be thoroughly compared before any conclusions can be made concerning the superiority of one of the methods.

1.2 Objective

The objective of this research is to provide a thorough and unbiased comparison of the TR and MRE approaches for reconstructing the release history of a groundwater contaminant. The MRE method was implemented and tested against results Woodbury and Ulrych (1996) to ensure that it was correctly implemented. For the TR method, we used the code and input files used by Skaggs and Kabala (1994); and we verified that we could reproduce their results. The methods were then evaluated and compared to determine their relative effectiveness in handling complications that are intrinsic to or may be encountered in field situations.

1.3 Scope

To allocate remediation costs of a contaminated aquifer, the locations of all contamination sources and the time-varying release rates from these sources must be identified. If concentration data are available, inverse methods can be used to identify contamination sources and release histories. Data can include samples taken at one location over a period of time, samples taken at many locations at one time, or a combination of the two. Due to the recent interest of the source history reconstruction problem (Skaggs and Kabala, 1994, 1994; Woodbury and Ulrych, 1996, 1998; Snodgrass and Kitanidis, 1997), the scope of this research is limited to reconstructing the release history from a known contamination source, using data sampled at many locations at a single time. We address transport of a conservative solute in a one-dimensional, saturated flow system. Although many methods are available for solving inverse problems, only two (Tikhonov regularization and minimum relative entropy) are addressed due to the recent interest in these two methods in the hydrology literature.

1.4 Background on Contaminant Transport in Groundwater

Groundwater contamination broadly defines any constituent that reduces the quality of groundwater. Contamination can be chemical, physical (particulate matter), or biological (viruses, bacteria). Chemical contamination can be broken down further into soluble components and non-aqueous phase liquid components. Soluble components are dissolved in the groundwater and are transported with the groundwater as it moves. Non-aqueous phase liquids are bodies of liquid that are separate from the water and are generally not transported with bulk groundwater movement. This work addresses transport of dissolved chemicals in water–saturated porous media.

Transport of soluble chemicals in groundwater is controlled by four main processes: advection, dispersion, transformation, and sorption. Advection describes the movement of a contaminant along with the bulk movement of groundwater. Dispersion describes the spreading of a contaminant as it moves through the porous media. Transformation includes any process which converts the contaminant into another chemical; examples of such processes are radioactive decay and biodegradation. Sorption describes the interaction of a contaminant with the porous media, and tends to retard the movement of the plume. The processes are described thoroughly in Fetter (1993). Contaminant transport may also be affected by other processes, such as colloidal transport, that are not described here.

All soluble contaminants are subject to advection and dispersion. The extent to which a contaminant is sorbed or transformed depends on the chemical nature of the contaminant and the porous media. Therefore, only the processes of advection and dispersion are considered here. Other processes can be added if necessary; however, if the processes are not modeled accurately, the uncertainty in the solution to the inverse problem increases. Since our goal is to compare the results of two inverse methods, and not to evaluate the ability of an inverse method to handle all naturally-occurring transport phenomena, we limit our evaluation to only the processes of advection and dispersion.

The governing equation for contaminant transport in groundwater is the advection-dispersion equation (ADE), which, for a one-dimensional semi-infinite domain, is:

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} - v\frac{\partial C}{\partial x} \tag{1.1}$$

$$C(0,t) = C_{in}(t)$$

$$C(x,t) \to 0 \ \text{ as } \ x \to \infty$$

$$C(x,0) = C_o(x)$$

where C is the contaminant concentration as a function of space (x) and time (t), D is the dispersion coefficient, v is the average groundwater velocity, C_{in} is the source history for a source located at $x = 0$, and C_o is the initial spatial distribution of the contaminant concentration. This form of the ADE is valid if D and v are constant in space.

The term on the left-hand side of Equation 1.1 describes the mass accumulation over time. The first term on the right-hand side describes dispersion. Dispersion is modeled as a Fickian process, with the dispersive flux proportional to the concentration gradient. Physically, dispersion depends on the spatial variation in local velocity (i.e. the deviation from the mean groundwater velocity, v), and therefore is not strictly a function of the concentration gradient. However, dispersion is often modeled as a Fickian process because it is a close approximation and no simple model of the true behavior has been developed. The dispersion coefficient is commonly modeled as $D = \alpha v$, where α is called the dispersivity; however, we assume that D is independent of v. With v constant, this assumption seems unnecessary; however, it affects the sensitivity analysis presented in Section 5.2.3. The dispersion coefficient, D, also accounts for molecular diffusion (which is a Fickian process); however, we neglect molecular diffusion, since it is generally small relative to dispersion. The second term on the right-hand side of Equation 1.1 describes the advective movement of the contaminant. Advective flux is proportional to the product of the mean groundwater velocity and the contaminant concentration.

1.4.1 Forward Advection-Dispersion Problem

If the contaminant source location and release history are known, the advection-dispersion equation can be solved directly using analytical techniques or numerical simulation to obtain an expression for $C(x,t)$. This problem is the forward advection-dispersion problem.

In a forward model, information about the source location and release history are specified in the boundary conditions for Equation 1.1. Then, after specifying the other boundary condition and the initial condition, the ADE is solved to obtain an expression for $C(x,t)$.

This problem has a unique solution. For simple source history functions, the equation can be solved analytically. For problems involving more complicated source history functions, Equation 1.1 can often be solved using convolution, because the equation is linear. Numerical solutions are necessary if the convolution integral is complicated. In any case, a unique solution is obtained (neglecting truncation and round-off errors).

1.4.2 Inverse Advection-Dispersion Problem

In the source identification problem, the source information is not available, but measurements of the spatial or temporal distribution of contamination are available. We assume that the source location is known, but the release history is unknown. Therefore, our goal is to obtain a (discrete) function for $C_{in}(t)$. We use the discrete concentration measurements, $C(x_j, T)$, (where x_j is the j^{th} sampling location, T is the sampling time, and $j = 1, 2, \ldots, N$, where N is the number of sampled data), along with Equation 1.1, to estimate the unknown source release history.

This inverse problem is ill-posed and inherently difficult to solve for many reasons. Even if the model parameters are known exactly and the sampled data are error-free, the solution to the inverse problem is non-unique. A more complete discussion is given in Chapter 2.

The inverse problem is further complicated by inexact information. In reality, the data contain measurement error, so the true solution will not fit the data exactly. Also, the parameter values (v and D) are not known exactly. In Equation 1.1, these values are assumed to be constant in space and time; however, they generally vary in space-time and cannot be modeled accurately. Furthermore, the Fickian model for dispersion is known to be incorrect, and we have not accounted for all the processes that affect contaminant transport.

An additional complication is that dispersion is a smoothing process; therefore the structure in the source history will be smoothed out over time and some information about these features will be lost. If data are sampled after some structure of the input function (source) has been lost, it may not be possible to reproduce the exact structure with an inverse model.

Inverse problems are more difficult to solve than forward problems. The inverse solution cannot be obtained directly, and the solution, if it exists, is not necessarily unique. Several techniques have been developed to solve inverse problems. A brief description of two classes of techniques (regularization and maximum entropy) is given in Chapter 2.

1.5 Literature Review on Source History Problems

Several researchers have used inverse methods to identify the location and release history of a source of groundwater contamination. In all of the studies, a hypothetical true source was generated, and the ability of the inverse technique to reproduce the true solution was evaluated. No studies were found that involved real data. The source history reconstruction problem studied in this research follows the work of Skaggs and Kabala (1994), who used Tikhonov regularization to solve the inverse problem, and Woodbury and Ulrych (1996), who used minimum relative entropy inversion. In other studies, different methods have been used to solve the same inverse problem. For example, Skaggs and Kabala (1995) used the method of quasi-reversibility, and Snodgrass and Kitanidis (1997) used a geostatistical approach. Later, Woodbury *et al.*, (1998) extended the MRE approach to a three-dimensional plume. In addition, studies have been done on source history reconstruction for other groundwater contamination problems (e.g. Gorelick *et al.*, 1983; Wagner, 1992). These works are described here.

Skaggs and Kabala (1994) used Tikhonov regularization to reconstruct the release history for a point source of contamination in a one-dimensional flow field. They solved the Fredholm integral equation

$$C(x,T) = \int_0^T C_{in}(t) f(x, T-t)\, dt \ , \tag{1.2}$$

where C is concentration, x is the spatial location (dimensionless), T is sampling time (dimensionless), $C_{in}(t)$ is the input concentration at the source (taken to be at $x = 0$), t is the source release time (dimensionless), $f(x,t)$ is the solution to Equation 1.1 with a pulse input at $x = 0$ and $t = 0$, and $C_o(x) = 0$. The true source history function was

$$\begin{aligned} C_{in}(t) \quad = \quad & \exp\left[-\frac{(t-130)^2}{2(5)^2}\right] + 0.3\exp\left[-\frac{(t-150)^2}{2(10)^2}\right] + \\ & 0.5\exp\left[-\frac{(t-190)^2}{2(7)^2}\right] \ , \end{aligned} \tag{1.3}$$

and is shown in Figure 1.1. Skaggs and Kabala reconstructed the source history using two different spatial data sets—one sampled at $T = 300$ and one sampled at $T = 600$. They used a discretized form of the integral equation (1.2). Twenty-five discrete sampling locations were used. The complete plume distribution and the sampling locations are shown in Figure 1.2 for $v = 1.0$ and $D = 1.0$. The regularized source history function was discretized into 100 points, uniformly-spaced in time (from $t = 0.01$ to $t = 250$ for data sampled at $T = 300$; and from $t = 0.01$ to $t = 300$ for data sampled at $T = 600$).

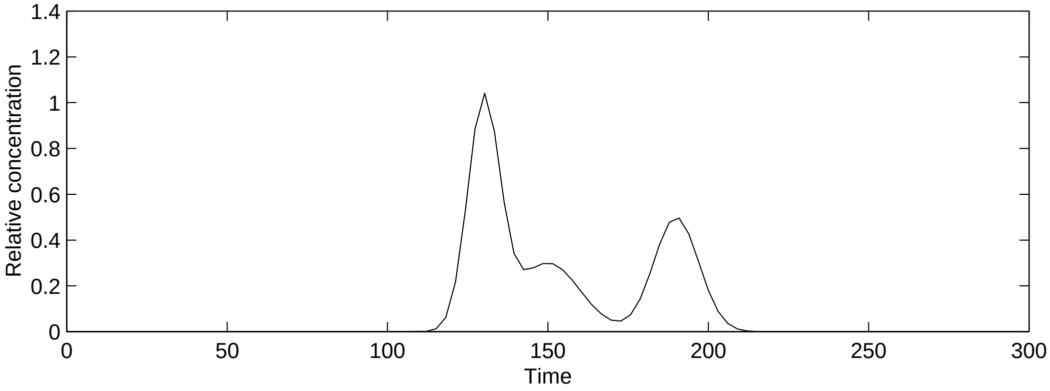

Figure 1.1: True source history function used by Skaggs and Kabala (1994).

Note that at $T = 300$, only two of the three peaks in the source function are distinguishable; and at $T = 600$, the plume appears to have only one peak. Thus, dispersion results in the loss of information over time. Skaggs and Kabala (1994) obtained a reasonable reconstruction of the source history when exact data (at either $T = 300$ or $T = 600$) and accurate parameter values were used.

Skaggs and Kabala (1994) also reconstructed the source history using noisy data. The sampled data were calculated by $C_{\text{meas}}(x_j, T) = C_{\text{exact}}(x_j, T) + \epsilon\delta_j C_{\text{exact}}(x_j, T)$, where $C_{\text{meas}}(x_j, T)$ is the measured concentration at location x_j at time T, x_j is the spatial coordinate of the j^{th} sample, $C_{\text{exact}}(x_j, T)$ is the true concentration at x_j at time T, ϵ is the noise level, and δ_j is the j^{th} independent random deviate (standard normal). They used a moderate noise level of $\epsilon = 0.05$, and a high noise level of $\epsilon = 0.2$. With noisy data (moderate or high noise) sampled at $T = 300$, the reconstructed source history matches the temporal position of the peaks in the true source history, but the magnitudes are less accurate. With noisy data (moderate or high noise)

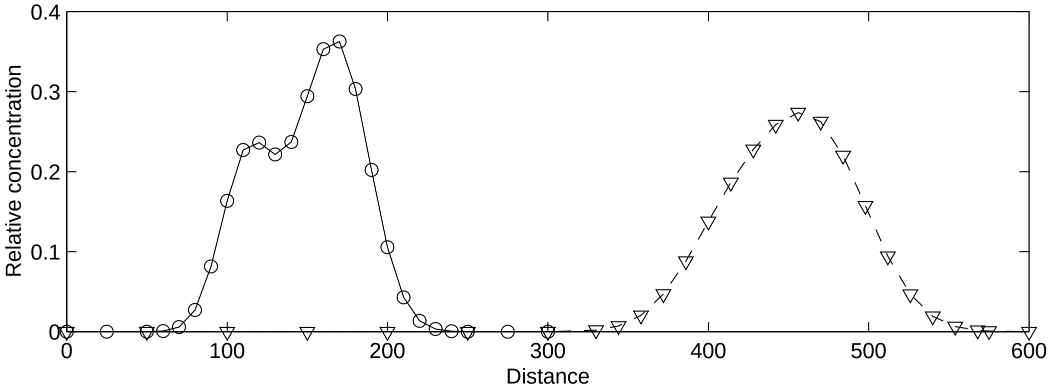

Figure 1.2: True plume at $T = 300$ and $T = 600$ used by Skaggs and Kabala (1994). Solid line: true plume at $T = 300$; dashed line: true plume at $T = 600$; circles: sample locations at $T = 300$; triangles: sample locations at $T = 600$.

sampled at $T = 600$, the combined effects of measurement error and loss of information due to dispersion result in a poorly-reconstructed source history. These results are presented in Section 3.1.2, in which the examples of Skaggs and Kabala (1994) are reproduced.

Skaggs and Kabala (1994) also tested the effects of inexact parameter values by varying v and D by ±5% from their true values, while using exact data ($\epsilon = 0$). Their results show that overestimating v resulted in a slightly more disperse source function that is shifted to later times; while the opposite occurred for an underestimation of v. Overestimating D resulted in a less disperse source history function; while underestimating D resulted in a more disperse source function.

Woodbury and Ulrych (1996) used minimum relative entropy inversion to reconstruct the release history for the problem shown in Equation 1.2 with the true source history function shown in Equation 1.3. They discretized the equation using 60 discrete spatial sampling locations. For a sampling time of $T = 300$ days, the data were uniformly spaced between $x = 5$ m and $x = 300$ m; and the regularized source history function was discretized into 100 points, uniformly-spaced in time, from $t = 0$ to $t = 300$ days. For sampling at $T = 600$ days, sampled data were uniformly spaced between $x = 300$ m and $x = 600$ m; and the source function was discretized into 200 uniformly-spaced points between $t = 0$ and $t = 600$ days.[1]

Woodbury and Ulrych (1996) used MRE to reconstruct the source history using sampled data at $T = 300$ days with three different prior expected value functions (boxcar, exponential, and Gaussian). In all cases, the reconstructed source functions matched well with the true function. They also used sampled data from $T = 600$ days with a boxcar prior expected value function. The reconstructed source history matched the first and third peaks in the true history; however, the second peak was missed. Results from equivalent simulations are presented in Section 3.2.2, in which the code used in this research is verified by comparing the results to those of Woodbury and Ulrych (1996). Note that the spatial and temporal discretizations of Woodbury and Ulrych (1996) are different from those of Skaggs and Kabala (1994).

Woodbury and Ulrych (1996) also reconstructed the source history using noisy data. They calculated the sampled data from $C_{\text{meas}}(x_j, T) = C_{\text{exact}}(x_j, T) + \epsilon\delta_j$. Unlike Skaggs and Kabala, the magnitude of their noise was independent of the true concentration. Woodbury and Ulrych (1996) used two different methods to account for noisy data. The first method involved

[1] Personal communication with A.D. Woodbury, February 2, 1999.

modifying the MRE algorithm to account for noise; this approach is presented in Appendix B of this thesis. The second method involved filtering the data with a Butterworth low-pass filter to remove the high-frequency noise components. Woodbury and Ulrych (1996) used noise levels of $\epsilon = 0.005$, 0.01, and 0.05 with a Gaussian prior distribution and sampled data from $T = 300$ days to reconstruct the release history using the first method for handling noisy data. The results show that the three peaks of the distribution were captured; however the magnitudes of the peaks were not preserved. Using a noise level of $\epsilon = 0.01$ and the filtering method to account for noisy data, they obtained a source function that matched the timing of the first and third peaks, but not the magnitudes.

Skaggs and Kabala (1995) used the method of quasi-reversibility to solve the inverse problem studied by Skaggs and Kabala (1994), using perfect data sampled at $T = 300$. In its usual form, the advection-dispersion equation is unstable for a negative time step because of the diffusion term. In the method of quasi-reversibility, the diffusion operator is replaced by a similar operator that is stable for a negative time step. Then, a forward problem is solved in reversed time, yielding an approximation to the source release history. This method was computationally more efficient than Tikhonov regularization, but it did not recover the source history as well (Skaggs and Kabala, 1995).

Snodgrass and Kitanidis (1997) used a geostatistical approach to solve the same inverse problem. They treated the source history as a random function, and solved for its expected value. This procedure is carried out by first selecting a model covariance function, with a known functional form and unknown parameter values. A covariance matrix is created from this function, and the values of the unknown parameters are estimated through maximum likelihood estimation. Using these estimates the expected value and covariance of the source history function are obtained. If the results are unconstrained, the solution may contain negative values; however, the results can be constrained to be positive. Other inversion techniques such as Tikhonov regularization are special cases of this method, if the covariance function is chosen appropriately (Snodgrass and Kitanidis, 1997).

Snodgrass and Kitanidis (1997) solved the source history reconstruction problem using data sampled at $T = 300$, at the same sampling locations used by Skaggs and Kabala (1994, 1995) (shown in Figure 1.2). The reconstructed source history function was discretized into 300 uniformly-spaced points between $t = 0$ and $t = 300$. With exact data, a Gaussian covariance, and implementing the non-negativity constraint, they obtained a reasonable reconstruction of the source history function. With sparse data and large measurement error, they reproduced the early and late peaks of the true source history function; however, the magnitudes of these peaks were underestimated.

Woodbury *et al.* (1998) used minimum relative entropy inversion to reconstruct the source history function shown in Equation 1.2 for a three-dimensional plume. In one case, they used perfectly sampled data from one location (directly downgradient from the source) at 36 uniformly-spaced time intervals between $t = 0$ days and $t = 360$ days. They discretized the source history function into 360 uniformly-spaced points between $t = 0$ days and $t = 360$ days. The reconstructed source functions agreed well with the true function. The authors also introduced the relative entropy measurement, which is a measure of the resolution of the reconstructed source function. For this problem the resolution function showed that although the reconstructed solution agreed well with the true solution, the resolution was poor near the peaks. In another case, they used noisy data (with the same formulation as Woodbury and Ulrych, 1996) with $\epsilon = 0.01$. The solution showed three peaks of approximately the same magnitude as the true source function; however the positions of the peaks were shifted slightly.

Gorelick *et al.* (1983) used linear programming to estimate the steady state contaminant flux from a leaking pipe system. They assumed that a pipe could be divided into discrete sections, with pipe leakage defined by the mass flux rate out of each section of the pipe. Using known leakage rates, they used a forward model to calculate concentration at a few hypothetical sampling wells, then added noise to the data. Assuming perfect knowledge of the system parameters, they used linear programming to attempt to determine the leakage rate from each pipe section. When the number of leaky pipe sections were known exactly, the model was fairly accurate. However, the results were sensitive to measurement errors.

Gorelick *et al.* (1983) also used linear programming and multiple regression to estimate time-dependent contaminant releases from five disposal wells. The releases from the disposal wells were assumed to be constant over discrete time intervals, and a known number of time intervals was used. They created true release histories from the five wells, and generated temporal concentration distributions for three locations downstream of the disposal wells. With noisy data and perfect knowledge of the system parameters, they reproduced the releases from the disposal well using linear programming and multiple regression. Their solution matched the true releases well; however, they used a large quantity of time-dependent data, which is not generally available for real situations.

Wagner (1992) used maximum likelihood estimation to determine flow and transport parameter values, including source characteristics. The author specified parameter values and then solved a two-dimensional coupled flow (steady-state) and transport (transient) model to obtain hydraulic head and concentration fields for three different times. Heads and concentrations were sampled at fifteen locations at the three available times, and normally-distributed random noise was added to the measurements. These data were used in the maximum likelihood method to estimate values of six flow and transport parameters, and from zero to two source characteristics. In one case, they estimated only the flow and transport parameters, and assumed that the source characteristics were known exactly. In a second case, they assumed that the source location and release time were known, and estimated the values of the flow and transport parameters and the uniform flux rate at the source. In a third case, the source release time was discretized into two periods, and the uniform flux rate for each period was estimated. In a fourth case, two possible source locations were identified, and the flux rate from each location was estimated. The results showed that the parameter uncertainty increases with the number of unknown parameters (Wagner, 1992). Using this method to reproduce the source history function shown in Equation 1.2 would require many discrete time periods. Although the parameter uncertainty would increase with the additional unknown parameters, we cannot speculate as to the magnitude of this increase, and this method would probably work poorly in practice.

Chapter 2

Inverse Methods

The Fredholm integral equation of the first kind is a general equation describing a linear system:

$$d(x,T) = \int_0^T f(x,t)m(t)dt \ , \tag{2.1}$$

where $d(x,T)$ is measured data as a function of location, x, at sampling time, T; $f(x,t)$ is the kernel function that describes the physics of the linear system; and $m(t)$ is the model of the system, as a function of time, t. In a forward problem, the model and kernel are known, and the state of the system at a later time, T, can be obtained by evaluating the integral. In an inverse problem, the data and kernel are known, and the Fredholm integral equation must be used to estimate the model, $m(t)$.

To solve the inverse problem, the Fredholm integral equation can be discretized into a matrix equation:

$$\mathbf{d} = \mathbf{Gm} \ , \tag{2.2}$$

where $\mathbf{d}$ is an $N \times 1$ vector of measured data sampled at discrete locations, N is the number of data points, $\mathbf{G}$ is an $N \times M$ matrix of values of the kernel function scaled by the temporal discretization, $\mathbf{m}$ is an $M \times 1$ vector of unknown model parameters, and M is the number of model parameters.

In the source history reconstruction problem, the $i^{\rm th}$ element of the vector $\mathbf{m}$ represents the source concentration at time t_i $(i = 1, 2, \ldots, M)$; the $j^{\rm th}$ element of the vector $\mathbf{d}$ is the sampled concentration at location x_j $(j = 1, 2, \ldots, N)$; and the $j, i^{\rm th}$ entry in the matrix $\mathbf{G}$ represents the kernel function evaluated at time $T - t_i$ and location x_j. The kernel function is the solution to the advection-dispersion equation for a pulse input at the source location, scaled by the temporal discretization.

Most inverse problems are ill-posed and therefore difficult to solve. For a problem to be well-posed, it must satisfy the requirements of existence and uniqueness, and the solution must depend continuously on the data. In most inverse problems, at least one of these three requirements is not satisfied. Existence is satisfied in an inverse problem with perfect data, because the data must be from a physically plausible situation. However, if the data contain measurement error, an accurate solution may not exist (Sun, 1994). In general, inverse problems are non-unique. Since the data are not exact, many solution can match the data to within a reasonable tolerance. Also, because the integral equation is discretized, the functions corresponding to the data and kernel are only known at a discrete number of points. No information is available regarding the behavior of the system at locations that were not sampled. Many alternative models exist that match the data at the sampled locations; therefore, the solution to the discretized inverse problem in non-unique. A problem is ill-posed if a small change in the data results in a corresponding small change in the solution, i.e. if the solution does not depend continuously on the data (Parker, 1994). In general, solutions to inverse problems do not depend continuously on the data; therefore, the unavoidable measurement errors and numerical errors often lead to inaccurate results.

Many methods have been developed to solve inverse problems. Some methods solve for one possible solution to the inverse problem that matches the data to within a reasonable tolerance; other methods solve for a probability density function of possible solutions. Because of measurement error, the true model will produce results that do not match the measurements exactly. If the solution does match the data, then the model is reproducing some of the measurement error.

The two inverse methods that we use in this research are Tikhonov regularization and minimum relative entropy (MRE) inversion. Tikhonov regularization is a classical inverse method based on the least squares solution, with regularization to improve the stability of the equation. Minimum relative entropy inversion belongs to the class of Bayesian methods in which the solution parameters are treated as random variables with a joint probability density function. In this remaining sections in this chapter, a brief, general discussion of inverse methods are presented, and Tikhonov regularization and MRE inversion are described in more detail.

2.1 Regularization Methods

In regularization methods, the ill-posed inverse problem is replaced with a family of similar well-posed problems through the introduction of a regularization operator and a regularization parameter. The goal is to find the model that best approximates the solution of the ill-posed problem by finding an exact solution to a similar well-posed problem.

In this section, we define the necessary properties of the regularization operator and parameter for any regularization method. Then, we discuss the selection of the regularization operator and parameter for the method of Tikhonov regularization, which is the regularization method used in this research.

2.1.1 Overview of Regularization Methods

The exact solution of Equation 2.2 cannot be obtained directly because the measured data are noisy, and the data vector, $\mathbf{d}$, is not known exactly. The actual data vector can be expressed as $\mathbf{d}^\epsilon$, where ϵ is the noise level given by $||\mathbf{d}^\epsilon - \mathbf{d}|| \leq \epsilon$. The solution to the actual problem is $\mathbf{m}^\dagger$, where $\mathbf{m}^\dagger = \mathbf{G}^\dagger\mathbf{d}^\epsilon$, and $\mathbf{G}^\dagger$ is the generalized inverse of $\mathbf{G}$. The operator, $\mathbf{G}^\dagger$, is unstable; therefore $||\mathbf{m} - \mathbf{G}^\dagger\mathbf{d}^\epsilon||$ may be large, even if $||\mathbf{d}^\epsilon - \mathbf{d}||$ is small,

The regularized solution of Equation 2.2 can be expressed as $\mathbf{m}_\alpha^\epsilon = R_\alpha\mathbf{d}^\epsilon$, where α is the regularization parameter, R is the family of regularization operators, and R_α is a particular regularization operator from the family, R, that is used to obtain the regularized solution. The regularization operator is a non-negative stabilizer that controls the sensitivity of $\mathbf{m}_\alpha^\epsilon$ to perturbations in the true data, $\mathbf{d}$ (Lukas, 1980). The regularization operator must satisfy the following properties:

- The operator must be stable.

- The operator must depend on $\mathbf{G}$.

- The regularized solution must approach the true solution as $\epsilon \rightarrow 0$.

The first property ensures that the solution depends continuously on the noisy data, $\mathbf{d}^{\epsilon}$. The second and third properties ensure that the behavior of the regularized solution is similar to that of the true solution. The particular member (R_{α}) of the family of operators (R) is defined through the selection of the regularization parameter, α, which must be chosen in such a way that the third property is met. Therefore, the regularization parameter must have some dependence on the noise level, ϵ, or on the noisy data, $\mathbf{d}^{\epsilon}$, or both (Engl *et al.*, 1996).

Several regularization methods have been developed (cf. Dimri, 1992; Engl *et al.*, 1996; Lukas, 1980). The methods differ in their formulation of the regularization operator and selection of the regularization parameter. Tikhonov regularization was used in this research, and is described in detail in the following discussion.

2.1.2 Tikhonov Regularization

Regularization Operator

With Tikhonov regularization (Tikhonov and Arsenin, 1977), the discretized form of the ill-posed Fredholm integral equation (Equation 2.2) is replaced by a well-posed minimization problem. The expression to be minimized is

$$R_{\alpha}(\mathbf{d}^{\epsilon}) = ||\mathbf{Gm} - \mathbf{d}^{\epsilon}||^2 + \alpha^2\, ||\mathbf{L\ m}||^2 \ , \tag{2.3}$$

where α is the regularization parameter, $\mathbf{L}$ is an operator matrix, and $||\cdot||$ denotes the L_2 (Euclidian) norm. The first term on the right-hand side of

Equation 2.3 is the discrepancy term and represents the square of the norm of the difference between the measured data and the model-predicted system state. The second term on the right-hand side of Equation 2.3 is the regularization term and represents the square of the norm of a specific property of the model that depends on the operator matrix, **L**.

Useful choices of the operator matrix, **L**, are the derivatives of the discrete form of $m(t)$ with respect to t. With $\mathbf{L} = \mathbf{I}_M$ (0^{th} derivative), where $\mathbf{I}_M$ is the $M \times M$ identity matrix, the norm of the model is minimized (zero-order regularization). To minimize the first derivative of the model (first-order regularization), we have

$$\mathbf{L} = \begin{bmatrix} -1 & 1 & 0 & 0 & \dots \\ 0 & -1 & 1 & 0 & \dots \\ 0 & 0 & -1 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{2.4}$$

where **L** is an $M - 1 \times M$ matrix. To minimize the second derivative of the model, or to maximize the smoothness of the model (second-order regularization), we use

$$\mathbf{L} = \begin{bmatrix} 1 & -2 & 1 & 0 & \dots \\ 0 & 1 & -2 & 1 & \dots \\ 0 & 0 & 1 & -2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{2.5}$$

where **L** is an $M - 2 \times M$ matrix. The choice of L depends on the desired properties of the regularized solution.

The regularization term stabilizes the problem; therefore, the minimization of R_α in Equation 2.3 is a trade-off between matching the data and stabilizing the solution. A large value of α produces a stable solution; however it may not adequately satisfy the original problem. A small value of α

could be expected to minimize the discrepancy; however, the problem then is approaching the original ill-posed problem, and is unstable (Lukas, 1980).

The Tikhonov regularization solution is found by minimizing $R_\alpha(\mathbf{d}^\epsilon)$. For zero-order regularization, this is a modified least squares solution, $\mathbf{m}_\alpha^\epsilon = (\mathbf{G}^T\mathbf{G} + \alpha^2\mathbf{I})^{-1}\mathbf{G}^T\mathbf{d}^\epsilon$ (Engl *et al.*, 1996). In general terms, the minimum-norm least squares solution to the regularized problem is $\mathbf{m}_\alpha^\epsilon = (\mathbf{G}^T\mathbf{G} + \alpha^2\mathbf{L}^T\mathbf{L})^{-1}\mathbf{G}^T\mathbf{d}^\epsilon$. With this basic theory, no constraints are made on the solution; however, with some modified approaches, the solution can be constrained to physically reasonable ranges. For example, we know that concentration is non-negative; therefore, we can constrain the solution so that each element of $\mathbf{m}_\alpha^\epsilon$ is non-negative.

Selecting the Regularization Parameter

Several methods can be used to select the optimal value of the regularization parameter, α. These methods include the discrepancy principle, generalized cross-validation, F-test, and L-curve.

The discrepancy principle states that the quality of the solution must be comparable to the quality of the input data (Groetsch, 1984). With this method, the regularization parameter should be selected so that

$$||\mathbf{G}\mathbf{m}_\alpha^\epsilon - \mathbf{d}^\epsilon|| = \epsilon \ . \tag{2.6}$$

There is a unique value of $\alpha(\epsilon)$ that satisfies this equation (Groetsch, 1984).

In the generalized cross-validation approach (Wahba, 1977), the regularization parameter, α, is determined through the ratio

$$G(\alpha) = \frac{||\mathbf{G}\mathbf{m}_\alpha^\epsilon - \mathbf{d}^\epsilon||^2}{[\mathrm{Tr}(\mathbf{I} - \mathbf{G}\mathbf{G}^\#)]^2} \tag{2.7}$$

where "Tr" denotes the trace and $\mathbf{G}^\#$ is the regularized pseudo-inverse of $\mathbf{G}$, such that $\mathbf{m}_\alpha^\epsilon = \mathbf{G}^\#\mathbf{d}^\epsilon$. The optimal choice of α is the value that minimizes $G(\alpha)$ (Wahba, 1977).

The F-test method (Provencher, 1982a; Obenchain, 1977) for selecting the regularization parameter, α, uses the F-distribution, with $P(\alpha) = P(F(\alpha), \nu_1, \nu_2)$, where P denotes probability, F is the ratio to be tested, ν_1 is the number of degrees of freedom associated with the numerator of F, ν_2 is the number of degrees of freedom associated with the denominator of F, and

$$F(\alpha) = \frac{(R_\alpha - R_{\alpha_o})/M_o}{R_{\alpha_o}/(N - M_o)} \tag{2.8}$$

where R_α is shown in Equation 2.3, α_o is a small value of α such that $R_{\alpha_o} \approx R_0$, N is the number of data points, and M_o is the number of degrees of freedom of R_α (described below). The numerator of $F(\alpha)$ characterizes the bias from regularization, and the denominator characterizes the error associated with the least squares solution (i.e. no regularization). A good choice for the regularization parameter is the value that balances the regularization error and the least squares error. Thus, α is chosen such that $P(\alpha) = 0.5$ (Provencher, 1982a).

The parameter M_o, in Equation 2.8, is the number of degrees of freedom of R_α, which can be expressed as the number of independent parameters in $\mathbf{m}$. The model vector $\mathbf{m}$ contains M parameters; however, the regularization makes the model parameters correlated, so we have $M_o < M$ (Provencher, 1982a). Provencher (1982a) suggests using an estimate of the scaled sum of squared errors of the data:

$$M_o = \sum_{k=1}^{K} \delta_k \, , \tag{2.9}$$

where $\delta_k = s_k^2/(s_k^2 + \alpha^2)$, s_k are the singular values of the regularized matrix problem (which can be obtained with singular value decomposition), and K is $\min(M, N)$. Other methods of calculating M_o yield similar results (Provencher, 1982a).

In the L-curve method (Hansen, 1992), the norm of the regularization term $||\mathbf{L}\mathbf{m}_\alpha^\epsilon||$ is plotted on a log-log plot against the residual norm $||\mathbf{G}\mathbf{m}_\alpha^\epsilon - \mathbf{d}^\epsilon||$, for many values of the regularization parameter α. The resulting curve is called the L-curve, and illustrates the trade-off between minimizing the residual and minimizing the regularization. A sample L-curve plot is shown in Figure 2.1.

The L-curve has two distinct segments; one segment is nearly horizontal, and the other is nearly vertical. In the flat segment, corresponding to large values of α, the regularization error dominates and the effects of the residuals are dampened. The steep segment of the L-curve corresponds to small values of α, and the residuals dominate over the regularization. The point of maximum curvature on the L-curve is near the intersection of the flat and steep segments, and is called the “corner”. The value of the regularization parameter at the corner is often the optimal parameter value, in that it balances the trade-off between the regularization and residual errors (Hansen, 1992).

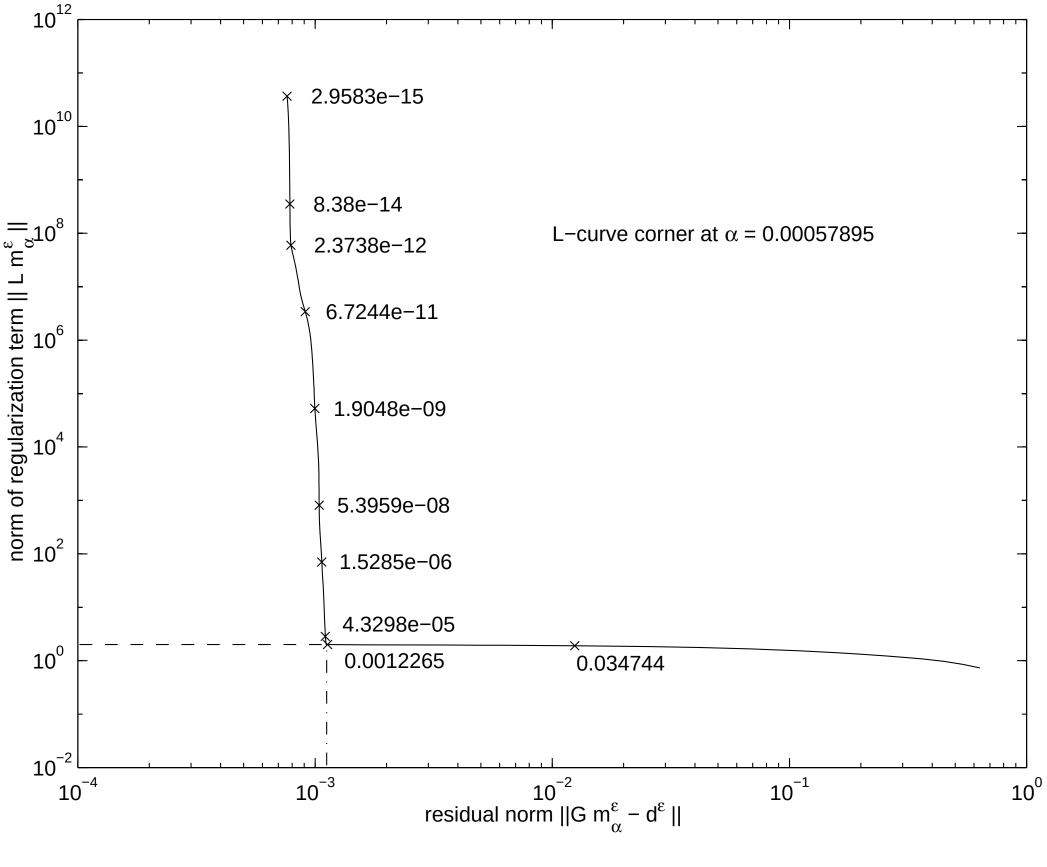

Figure 2.1: Sample L-curve plot. Numerical labels identify the value of α at each point marked with an "x".

Uncertainty in the Tikhonov Regularization Solution

The Tikhonov regularization solution can be written as $\mathbf{m}_\alpha^\epsilon = \mathbf{G}^\#\mathbf{d}^\epsilon$, where $\mathbf{m}_\alpha^\epsilon$ is the solution, $\mathbf{G}^\#$ is the generalized pseudo-inverse of $\mathbf{G}$, and $\mathbf{d}^\epsilon$ is the data vector. The uncertainty of the Tikhonov regularization solution can be defined by the model covariance matrix, $\mathbf{\Sigma_m}$, is given by $\mathbf{\Sigma_m} = \mathbf{G}^\#\mathbf{\Sigma_d}(\mathbf{G}^\#)^\mathbf{T}$, where $\mathbf{\Sigma_d}$ is the covariance matrix of the data. Using the model covariance matrix, we can define the $100(1-\alpha^*)\%$ confidence interval for the solution. Assuming normally–distributed errors, the $100(1-\alpha^*)\%$ confidence interval is $m_i \pm \beta^*\sigma_{ii}^2$, where σ_{ii}^2 is the i, i^{th} entry in $\mathbf{\Sigma_m}$, β^* is a value such that $P(Z \leq -\beta^*) = \alpha^*/2$ and $P(Z \leq \beta^*) = 1-\alpha^*/2$, and $P(Z \leq z)$ defines the probability that $Z \leq z$, and Z is a standard normal random variable. For example, the 90% confidence interval is $m_i \pm 1.645\sigma_{ii}$, and $P(Z < 1.645) = 0.95$.

For regularized solutions, the covariance matrix tends to underestimate the level of uncertainty, and depends on the value of the regularization parameter (Provencher, 1982a); therefore, the error bounds are not absolute. They can be used to compare the relative uncertainty in different regions of one solution; however, it is not valid to compare the error bounds with those of other Tikhonov regularization solutions, or with those of solutions using other inverse methods.

2.2 Maximum Entropy Methods

The Bayesian approach to solving inverse problems is to treat each unknown parameter as a random variable that has a true but unknown probability density function (pdf), $q^\dagger$. Bayesian methods use available data to develop an estimate, q, of the true pdf for each parameter value. Jaynes (1957) proposed the maximum entropy approach for selecting a pdf. The general procedure for maximum entropy methods is described in the next section. The minimum relative entropy inversion method, which is a special case of maximum entropy, is described in more detail in the subsequent section. Minimum relative entropy is the Bayesian approach used in this research.

2.2.1 Overview of Maximum Entropy Methods

The basis of maximum entropy methods is to obtain a probability density function that agrees with the available data and that avoids bias. To avoid bias, the probability density function that maximizes the uncertainty should be chosen (Jaynes, 1957). According to Shannon (1948), the measure of uncertainty that should be maximized is

$$H = -\sum_i q(m_i) \ln q(m_i) \ , \tag{2.10}$$

where $q(m_i)$ denotes the probability of occurrence of event m_i and H is the uncertainty. Since Equation 2.10 has the same form as the expression for entropy in statistical mechanics, the term "entropy" has been adopted for this quantity (Jaynes, 1957).

If a prior estimate, p, of the true probability density function is available, one can use the prior estimate and the available data to obtain a posterior estimate of the pdf. According to the principle of minimum relative entropy (Kullback, 1959), if a prior distribution is available, the posterior distribution that avoids bias is the pdf that minimizes the relative entropy, which is expressed as

$$H = \sum_i q(m_i) \ln \left[\frac{q(m_i)}{p(m_i)} \right] \ . \tag{2.11}$$

For continuous distributions, the relative entropy is expressed as (Shore and Johnson, 1980)

$$H(q,p) = \int q(m) \ln \left[\frac{q(m)}{p(m)} \right] \ dm \ . \tag{2.12}$$

The minimum relative entropy (MRE) inversion approach has been used in groundwater hydrology for parameter estimation (Woodbury and Ulrych, 1993; Woodbury *et al.*, 1995) and for the source history reconstruction problem (Woodbury and Ulrych, 1996; Woodbury *et al.*, 1998; Woodbury and Ulrych, 1998a). The MRE method is described in the following discussion, based on the approach taken by Woodbury and Ulrych (1993, 1996).

2.2.2 Minimum Relative Entropy Inversion

With the minimum relative entropy (MRE) approach, the Fredholm integral equation is discretized into matrix form. The MRE method treats the elements of the model vector, m_i, as random variables. A probability density function (pdf) is generated for each m_i using measured data and prior information about the ranges and expected values. The solution to the inverse problem is the best estimate of m_i, based on this pdf. In the approach described here, we use the expected value as the best estimate of m_i; other possible choices include the median and the mode of the distribution.

Since each m_i is a random variable, the possible states of the vector $\mathbf{m}$ can be characterized by a multivariate probability density function. Let

$\mathbf{x}$ denote the set of possible states of $\mathbf{m}$. There exists a true, but unknown, multivariate pdf, $q^{\dagger}(\mathbf{x})$. The goal of MRE is to obtain a reasonable estimate of $q^{\dagger}(\mathbf{x})$.

Woodbury and Ulrych (1996) first specified a range of possible values and an expected value for each element of the model vector, $\mathbf{m}$. Next, they obtained a prior distribution, $p(\mathbf{m})$, based on the range and expected value. Finally, they used the measured data, $\mathbf{d}$, and the prior distribution to obtain a posterior estimate, $q(\mathbf{m})$, of the true distribution.

Estimating the Prior Distribution

For many parameters, a reasonable upper and lower bound can be obtained. With these bounds, the base level of information about the model, $\mathbf{m}$, is a multivariate uniform distribution between these bounds:

$$\begin{aligned} b(\mathbf{m}) &= \prod_{i=1}^{M} \frac{1}{U_i - L_i} \quad \text{for } L_i \leq m_i \leq U_i \\ b(\mathbf{m}) &= 0 \quad \text{otherwise} \end{aligned} \tag{2.13}$$

where U_i is the upper bound of parameter m_i, L_i is the lower bound, and M is the total number of model parameters.

The entropy of the prior distribution, $p(\mathbf{m})$, relative to $b(\mathbf{m})$ is

$$H(p, b) = \int_{\mathbf{m}} p(\mathbf{m}) \ln \left[\frac{p(\mathbf{m})}{b(\mathbf{m})} \right] d\mathbf{m} \; . \tag{2.14}$$

The above integral is an M-fold integral, with each integrated from L_i to U_i, for $i = 1, 2, \ldots, M$. The relative entropy, $H(p, b)$, is minimized, subject to two constraints—the normalization requirement, which ensures that $\int_{\mathbf{m}} p(\mathbf{m}) d\mathbf{m} =$

1, and an expected value constraint. For many parameters, an estimate of the expected value of the parameter is known from prior knowledge, calibration, or other means. When this estimate of the expected value is available, entropy minimization is subject to the following expected value constraint:

$$\int_{\mathbf{m}} m_i p(\mathbf{m}) d\mathbf{m} = s_i \quad i = 1, 2, \ldots, M \tag{2.15}$$

where s_i is the expected value of model parameter m_i. With these constraints, the objective function to be minimized is

$$\begin{aligned} \phi &= H(p, b) + \mu \left[\int_{\mathbf{m}} p(\mathbf{m}) d\mathbf{m} - 1 \right] \\ &\quad + \sum_{i=1}^{M} \beta_i \left[\int_{\mathbf{m}} m_i p(\mathbf{m}) d\mathbf{m} - s_i \right] , \end{aligned} \tag{2.16}$$

where μ and β_i are Lagrange multipliers. The objective function is minimized with respect to $p(\mathbf{m})$ when the following equality holds:

$$0 = \ln \left[\frac{p(\mathbf{m})}{b(\mathbf{m})} \right] + 1 + \mu + \sum_{i=1}^{M} \beta_i m_i \ . \tag{2.17}$$

Solving this equation for $p(\mathbf{m})$ gives

$$p(\mathbf{m}) = b(\mathbf{m}) \exp \left[-1 - \mu - \sum_{i=1}^{M} \beta_i m_i \right] \ . \tag{2.18}$$

The values of the Lagrange multipliers are obtained from the two constraints. The derivation is shown in Appendix A. The values for β_i must be obtained numerically. Substituting the Lagrange multipliers into the previous expression, we obtain the prior distribution, $p(\mathbf{m})$:

$$\begin{aligned} p(m_i) &= \frac{\beta_i \exp(-\beta_i m_i)}{\exp(-\beta_i L_i) - \exp(-\beta_i U_i)} \ \text{ for } \beta_i \neq 0 \\ p(m_i) &= \frac{1}{U_i - L_i} \ \text{ for } \beta_i = 0 \ ; \\ p(\mathbf{m}) &= \prod_{i=1}^{M} p(m_i) \ . \end{aligned} \tag{2.19}$$

Estimating the Posterior Distribution

To obtain the posterior estimate of the true distribution, we minimize the entropy of the posterior distribution, $q(\mathbf{m})$, relative to the prior distribution, $p(\mathbf{m})$. The entropy to be minimized is

$$H(q,p) = \int_{\mathbf{m}} q(\mathbf{m}) \ln \left[\frac{q(\mathbf{m})}{p(\mathbf{m})} \right] d\mathbf{m} \; , \tag{2.20}$$

where $p(\mathbf{m})$ is given in Equation 2.19. The entropy minimization is subject to two constraints—the normalization requirement ($\int_{\mathbf{m}} q(\mathbf{m}) d\mathbf{m} = 1$) and the measured data. The data constraint ensures that the model agrees with the data.

According to the discretization of the Fredholm integral equation (Equation 2.2), the relationship between the model and the data is

$$d_j = \sum_{i=1}^{M} g_{ji} m_i \; , \tag{2.21}$$

where g_{ji} is the j, i^{th} element of the $\mathbf{G}$ matrix, and $j = 1, 2, \ldots, N$. Since the model parameters, m_i, are random variables, the data elements, d_j, are also random variables. To ensure that the model agrees with the data, the expected values of the data variables must be equal to the measured data, i.e.

$$\bar{d}_j = \int_{\mathbf{m}} d_j q(\mathbf{m}) d\mathbf{m} \; , \tag{2.22}$$

where $\bar{d}_j$ is the j^{th} measured data point. Substituting Equation 2.21 into the previous equation results in the data constraint to be used in the minimization problem:

$$\bar{d}_j = \int_{\mathbf{m}} q(\mathbf{m}) \sum_{i=1}^{M} g_{ji} m_i d\mathbf{m} \; . \tag{2.23}$$

With these constraints, the objective function to be minimized is

$$\begin{aligned} \phi \;&=\; H(q,p) + \mu\left[\int_{\mathbf{m}} q(\mathbf{m})d\mathbf{m} - 1\right] + \\ &\qquad \sum_{j=1}^{N}\lambda_j\left[\int_{\mathbf{m}} q(\mathbf{m})\sum_{i=1}^{M} g_{ji}m_i d\mathbf{m} - \bar{d}_j\right] \;, \end{aligned} \tag{2.24}$$

where μ and λ_j are Lagrange multipliers. The objective function is minimized with respect to $q(\mathbf{m})$ when the following equality holds:

$$0 = \ln\left[\frac{q(\mathbf{m})}{p(\mathbf{m})}\right] + 1 + \mu + \sum_{j=1}^{N}\left(\lambda_j\sum_{i=1}^{M} g_{ji}m_i\right) \;. \tag{2.25}$$

Solving this equation for $q(\mathbf{m})$ gives

$$q(\mathbf{m}) = cp(\mathbf{m})\exp\left[-\sum_{j=1}^{N}\left(\lambda_j\sum_{i=1}^{M} g_{ji}m_i\right)\right] \;, \tag{2.26}$$

where $c = \exp(-1-\mu)$.

The values of the Lagrange multipliers are obtained from the two constraints—the normalization constraint and the data constraint. Note that the measured values are now implicitly contained in the Lagrange multipliers, λ_j. The calculations for solving for the Lagrange multipliers are shown in Appendix B. The values for λ_j must be obtained numerically. Substituting the Lagrange multipliers into the previous expression, we obtain the posterior distribution, $q(\mathbf{m})$:

$$\begin{aligned} q(m_i) \;&=\; \frac{a_i\exp(-a_i m_i)}{\exp(-a_i L_i) - \exp(-a_i U_i)} \;\text{ for } a_i \neq 0 \\ q(m_i) \;&=\; \frac{1}{U_i - L_i} \;\text{ for } a_i = 0 \;; \\ q(\mathbf{m}) \;&=\; \prod_{i=1}^{M} q(m_i) \;, \end{aligned} \tag{2.27}$$

where $a_i = \beta_i + \sum_{j=1}^{N} \lambda_j g_{ji}$.

The solution to the inverse problem is a likely value of $\mathbf{m}$, based on the probability density function, $q(\mathbf{m})$. In this research, we use the expected value of $\mathbf{m}$ as the solution. For the i^{th} model parameter, the expected value of m_i is $\hat{m}_i = \int_{\mathbf{m}} m_i q(\mathbf{m}) d\mathbf{m}$. Substituting Equation 2.27 into this expression and integrating, we obtain

$$\begin{aligned} \hat{m}_i &= \frac{(a_i L_i + 1)\exp(-a_i L_i) - (a_i U_i + 1)\exp(-a_i U_i)}{a_i \left[\exp(-a_i L_i) - \exp(-a_i U_i)\right]} \quad \text{for } a_i \neq 0 \\ \hat{m}_i &= \frac{U_i + L_i}{2} \quad \text{for } a_i = 0 \ . \end{aligned} \tag{2.28}$$

Note that the measured data invariably contain measurement error. Therefore, the equalities in Equations 2.22 and 2.23 are not strictly correct. Equations 2.27 and 2.28 for the posterior distribution and $\hat{\mathbf{m}}$, respectively, are correct as written, whether or not the sampled data contain measurement error (Johnson and Shore, 1984; Woodbury and Ulrych, 1996). However, the measurement error is accounted for in the values of the Lagrange multipliers, λ_j (See Appendix B).

Uncertainty in the Minimum Relative Entropy Method Solution

For each model parameter, the MRE algorithm calculates the probability density function, $q(m_i)$. All values of m_i for which $m_i \neq 0$ are possible solution. We take the expected value of $\mathbf{m}$ to be the solution. The expected value is the 50^{th} percentile probability level, expressed as the value of m_i such that $\int_0^{m_i} q(m_i')\, dm_i' = 0.5$. The uncertainty in the solution can be expressed with other probability levels. We define the $100(1-\alpha^*)\%$ probability interval using the $\alpha^*/2$ percentile probability level and the $1 - \alpha^*/2$ percentile probability level. Then, the true value lies within this interval $100(1 - \alpha^*)\%$ of the time. For example, the 90% probability interval has as its bounds the 5^{th} and 95^{th} percentile probability levels, which are given by the values of m_i such that $\int_0^{m_i} q(m_i')\, dm_i' = 0.05$ and $\int_0^{m_i} q(m_i')\, dm_i' = 0.95$, respectively.

2.3 Comparison of Tikhonov Regularization and Minimum Relative Entropy Inversion

Tikhonov regularization and minimum relative entropy inversion are two methods of solving an inverse problem defined by an ill-posed Fredholm integral equation of the first kind. In both methods, the integral equation is discretized into a matrix equation. The physics of the process is assumed to be known and modeled exactly. Both methods use discrete measurements of concentration, along with an expression describing the physics of the process, to reproduce an input function that matches the data to within some defined measurement error.

With Tikhonov regularization, the Fredholm integral equation is replaced by a well-posed minimization problem whose solution is close to that of the original problem. The stabilizer has two components—a regularization parameter and an operator matrix. The choice of operator matrix defines the order of regularization, which specifies the feature of the input function that is minimized (e.g. zero-order regularization minimizes the norm of the input function; second-order regularization minimizes the oscillatory nature of the input function). The choice of operator matrix is somewhat subjective, and depends on the desired nature of the input function. The regularization parameter describes the relative contribution of the data and the regularizor to the solution of the inverse problem. With a small value of the regularization parameter, the solution is unstable but similar to the least squares solution. With a large regularization parameter, the resulting minimization problem will be stable, but might not adequately represent the true solution.

With minimum relative entropy inversion, the model parameters to be estimated are treated as random variables. A probability density function is obtained for each model parameter, and the expected value is selected as the most likely value of the model parameter. The posterior distribution is obtained by minimizing the entropy of the distribution relative to a prior distribution, while constraining the results to match the measured data. The prior distribution can be developed so that the mean value and the known bounds on the parameter values are included in the solution; however, the choice of these bounds is subjective.

Both methods contain some subjectivity. With Tikhonov regularization, the operator matrix and the value of the regularization parameter must be selected. Methods have been developed to select the value of the regularization parameter with some physical or mathematical justification; however, it still remains somewhat subjective. The solution depends heavily on the value of the regularization parameter; therefore, the dependence of the solution on the subjectivity of the parameter choice cannot be avoided. With minimum relative entropy inversion, the selection of the prior distribution is subjective. Although some information, such as the mean value or range of values, is available to refine the prior distribution, the choice of distribution is still subjective.

Both methods provide a mechanism for refining the solution according to prior knowledge of the result. With Tikhonov regularization, the choice of operator matrix can affect the shape of the solution. If the solution is known to be smooth, choosing second-order regularization will preserve the smooth nature of the solution. With minimum relative entropy inversion, the prior distribution can be chosen to include only those values that fall within the known range of parameter values. Also, if the expected value of the parameter is known, this information can be incorporated into the prior distribution.

One weakness of the basic Tikhonov regularization theory is that the parameters cannot be constrained with upper or lower bounds. Although we know that the concentration of a contaminant is always non-negative, the Tikhonov regularization solution can produce negative results. Modified approaches of Tikhonov regularization can account for non-negativity constraints, and the code used in this research, CONTIN (Provencher, 1982a, 1982b, 1984) can handle these constraints.

One potentially significant difference between the two methods is in the treatment of inexact data. With Tikhonov regularization, the noise level need not be specified, unless the discrepancy principle is used to select the value of the regularization parameter. If the discrepancy principle is not used; the inexact data are used directly in the procedure. With MRE, the noise level must be specified in the data constraint (Equations 2.23 and B.9, for exact and inexact data, respectively); therefore, a reasonable approximation of the noise level must be specified. Since the noise level is not known in practice, an incorrect noise level is likely to be used, resulting in an inaccurate data constraint, and possibly an inaccurate solution.

Both methods contain unavoidable subjectivity, and both methods can incorporate prior knowledge of the model or its parameters. Based on theory alone, we cannot determine which of the two methods better solves the source history reconstruction problem.

Chapter 3

Application of Inverse Methods to the Source History Reconstruction Problem

Tikhonov regularization and minimum relative entropy inversion are used to solve the source history reconstruction problem, described by the following integral equation:

$$C(x,T) = \int_0^T C_{in}(t) f(x, T-t)\, dt \;, \tag{3.1}$$

where C is concentration, x is the spatial location (dimensionless), T is sampling time (dimensionless), $C_{in}(t)$ is the input concentration at the source (taken to be at $x = 0$), t is the source release time (dimensionless), and $f(x, T-t)$ is the solution to the advection-dispersion equation (Equation 1.1) with a pulse input at $x = 0$ and $t = 0$, given by

$$f(x, T-t) = \frac{x}{2\sqrt{\pi D (T-t)^3}} \exp\left\{ -\frac{[x - v(T-t)]^2}{4D(T-t)} \right\} \;, \tag{3.2}$$

where v is the groundwater velocity and D is the dispersion coefficient. To implement the inverse methods, the integral equation is discretized as (Equation 2.2)

$$\mathbf{d} = \mathbf{G}\mathbf{m} \;, \tag{3.3}$$

where $\mathbf{d}$ corresponds to the N-length vector of sampled concentrations ($d_j = C(x_j, T)$, $j = 1, 2, \ldots, N$), $\mathbf{m}$ corresponds to the M-length vector of model parameters ($m_i = C_{in}(t_i)$, $i = 1, 2, \ldots, M$), and $\mathbf{G}$ corresponds to the $N \times M$ kernel matrix ($g_{ji} = \Delta t f(x_j, T - t_i)$), x_j are the sample locations, t_i are the input times, and $\Delta t = t_{k+1} - t_k$.

In this chapter, we discuss the implementation and verification of the methods used to solve this inverse problem. For all of the verification runs described in this chapter, the true source history is

$$\begin{aligned} C_{in}(t) &= \exp\left[-\frac{(t-130)^2}{2(5)^2}\right] + 0.3\exp\left[-\frac{(t-150)^2}{2(10)^2}\right] + \\ & \quad 0.5\exp\left[-\frac{(t-190)^2}{2(7)^2}\right] , \end{aligned} \tag{3.4}$$

and is shown in Figure 1.1. The values of the transport parameters are $v = 1$ m/d, and $D = 1$ m^2/d.

3.1 Source History Reconstruction using Tikhonov Regularization

Skaggs and Kabala (1994) used commercially-available software to solve the inverse problem using the Tikhonov regularization method. In this research, we used the same software package, including the input files created by Skaggs and Kabala. In this section, we discuss some details of the code and the reproduction of the results of Skaggs and Kabala (1994) using the code.

3.1.1 Implementation of Tikhonov Regularization

The Tikhonov regularization method was carried out using CONTIN (Provencher, 1982a; 1982b; 1984), a general purpose Fortran program for obtaining the regularized solution of linear integral equations of the first kind. CONTIN can solve up to fifth-order regularization, and uses the F-test to

Table 3.1: Details of Tikhonov regularization verification runs.

Run Number	Sample Time (T)	Noise Level (ϵ)	Parameter Error (v or D)
TV-1	300	0	none
TV-2	600	0	none
TV-3	300	0.05	none
TV-4	300	0.2	none
TV-5	600	0.05	none
TV-6	600	0.2	none
TV-7	300	0	+5% v
TV-8	300	0	-5% v
TV-9	300	0	+5% D
TV-10	300	0	-5% D

choose the value of the regularization parameter. With CONTIN, the solution can be constrained to be positive. CONTIN estimates the error in the solution by calculating the covariance matrix of the solution. Skaggs and Kabala (1994) used this code for their simulations, with second-order regularization, and with a modification to include a function to evaluate the kernel[1] (Equation 3.2).

3.1.2 Verification of Tikhonov Regularization Routine

We obtained the input files and source code that Skaggs and Kabala (1994) used in their analysis, and we reproduced the results of their ten examples. The details of these examples are outlined in Table 3.1. In all cases, the non-negativity constraint was used. The sampling locations are shown in

[1] Personal communication with T.H. Skaggs, October 19, 1998.

Figure 1.2. For runs with measurement error, the data were generated using $C_{\text{meas}}(x_j,T) = C_{\text{exact}}(x_j,T) + \epsilon\delta_j C_{\text{exact}}(x_j,T)$, where δ_j is a standard normal random variable. For runs that used a sampling time of $T = 300$ (dimensionless time), the time domain was discretized into 100 uniformly-spaced intervals between $t = 0.01$ and $t = 250$. For runs using data sampled at $T = 600$, the time domain was discretized into 100 uniformly-spaced intervals between $t = 0.01$ and $t = 300$. The results are presented in Figures 3.1—3.10, and are consistent with the results presented by Skaggs and Kabala (their Figures 2—9).

Runs TV-1 and TV-2 used perfect data sampled at $T = 300$ and $T = 600$, respectively. The results show that the reconstructed source history agrees well with the true source history. The only differences are that the magnitude of the first peak is slightly underestimated and the shape of the second peak is not quite accurate. The regularized solution using data sampled at $T = 300$ shows better agreement with the true source history than does the solution using data sampled at $T = 600$. This is expected because the plume at $T = 600$ is more disperse, and therefore some of the features of the true source history have been lost. In both cases, the reconstructed plume is indistinguishable from the sampled data.

Runs TV-3—TV-6 used inexact data, and the results show that the measurement error affects the accuracy of the regularized solution. With data sampled at $T = 300$ and either moderate ($\epsilon = 0.05$) or high ($\epsilon = 0.2$) noise levels (Figures 3.3 and 3.4, respectively), the reconstructed source history has three peaks that somewhat match the timing of the true peaks but do not match the magnitudes. Using data sampled at $T = 600$ and moderate noise (Figure 3.5), the reconstructed source history still has three peaks, although the magnitudes

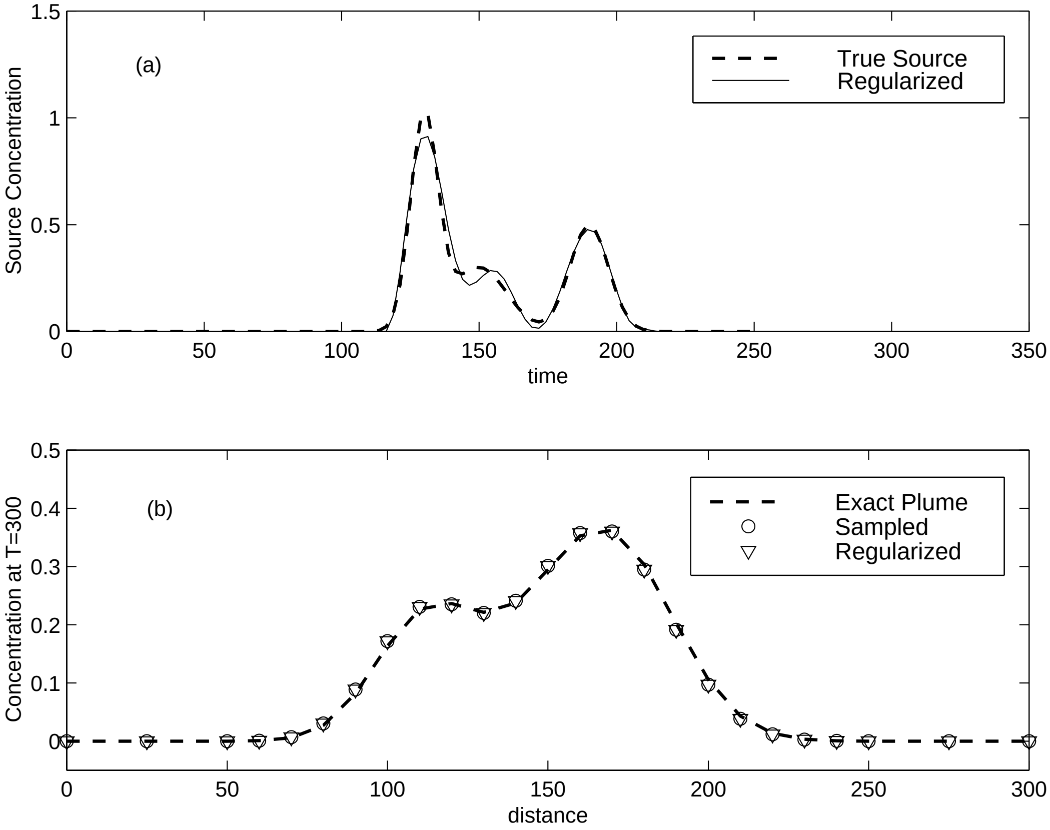

Figure 3.1: Results of Tikhonov regularization verification Run TV-1. (a) Reconstructed source history. (b) Plume at $T = 300$. Parameters are: $\epsilon = 0.0$, $v = 1.0$, $D = 1.0$, and $T = 300$.

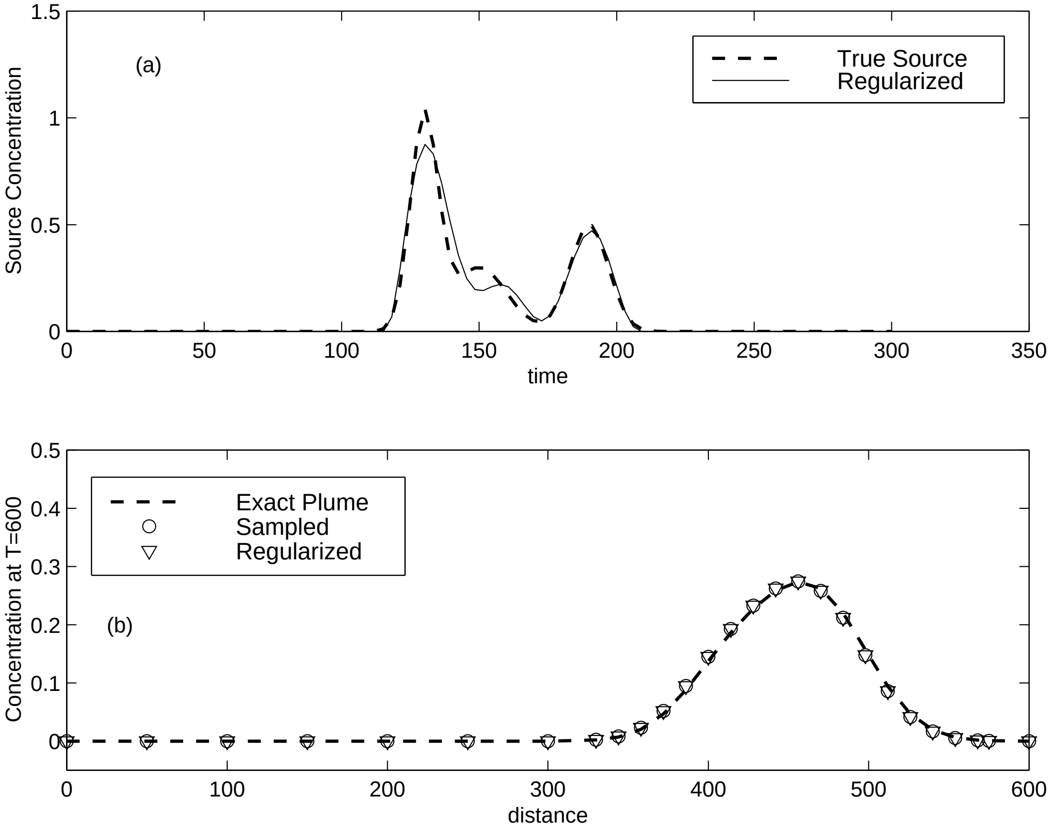

Figure 3.2: Results of Tikhonov regularization verification Run TV-2. (a) Reconstructed source history. (b) Plume at $T = 600$. Parameters are: $\epsilon = 0.0$, $v = 1.0$, $D = 1.0$, and $T = 600$.

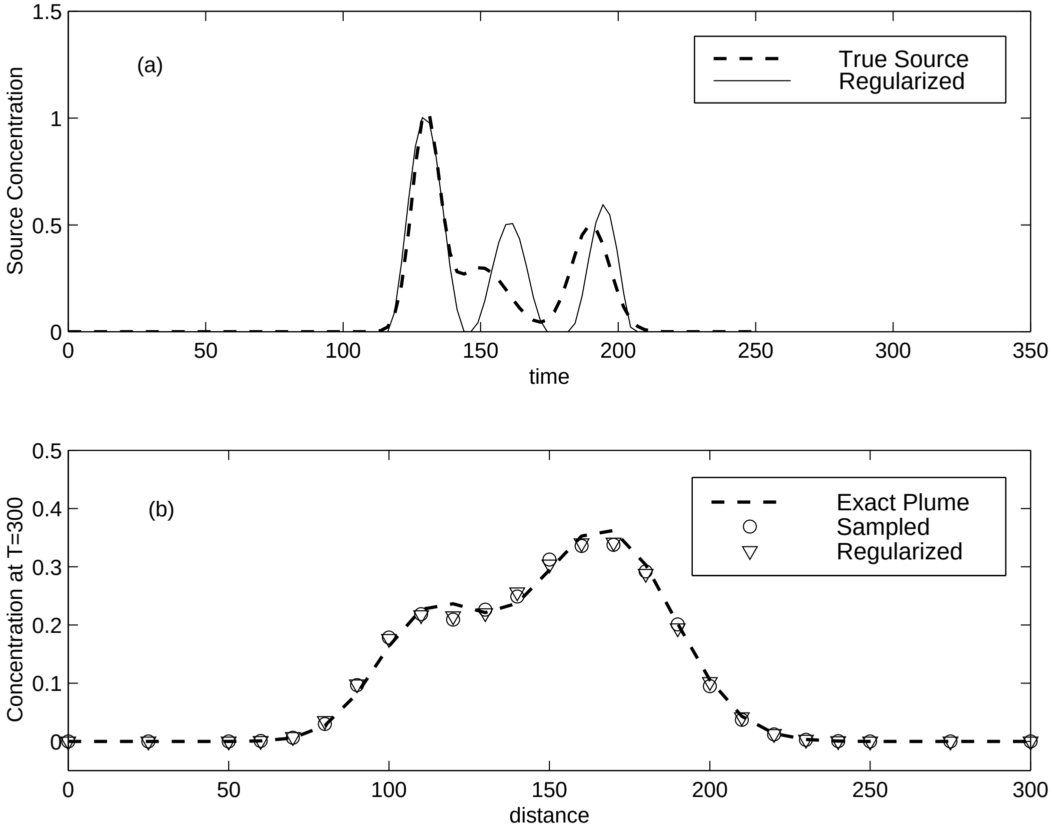

Figure 3.3: Results of Tikhonov regularization verification Run TV-3. (a) Reconstructed source history. (b) Plume at $T = 300$. Parameters are: $\epsilon = 0.05$, $v = 1.0$, $D = 1.0$, and $T = 300$.

are much worse. With high noise (Figure 3.6), the important features of the source history cannot be reconstructed. The reconstructed plumes in all cases are less accurate than in Runs TV-1 and TV-2.

Runs TV-7—TV-10 used perfect data sampled at $T = 300$ to determine the effects of using incorrect estimates of the parameter values (v or D) in the inversion method. In these runs, either the velocity or the dispersion coefficient was changed by 5% (increased in one run; decreased in another

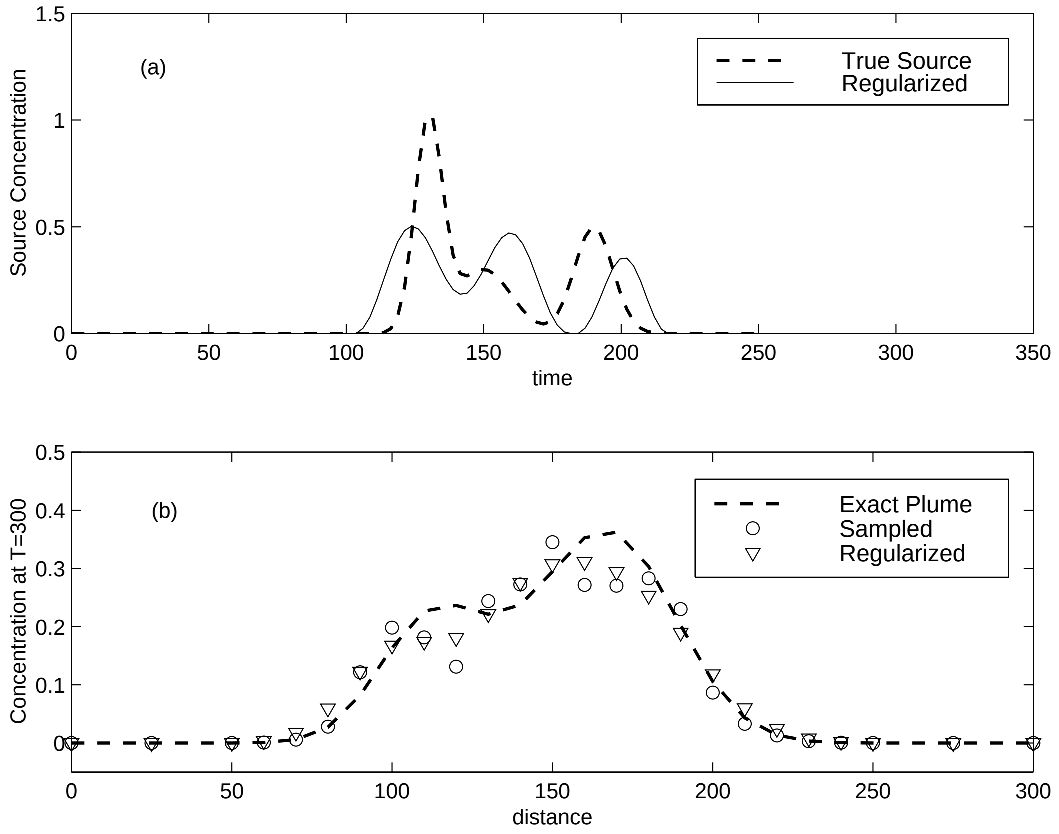

Figure 3.4: Results of Tikhonov regularization verification Run TV-4. (a) Reconstructed source history. (b) Plume at $T = 300$. Parameters are: $\epsilon = 0.2$, $v = 1.0$, $D = 1.0$, and $T = 300$.

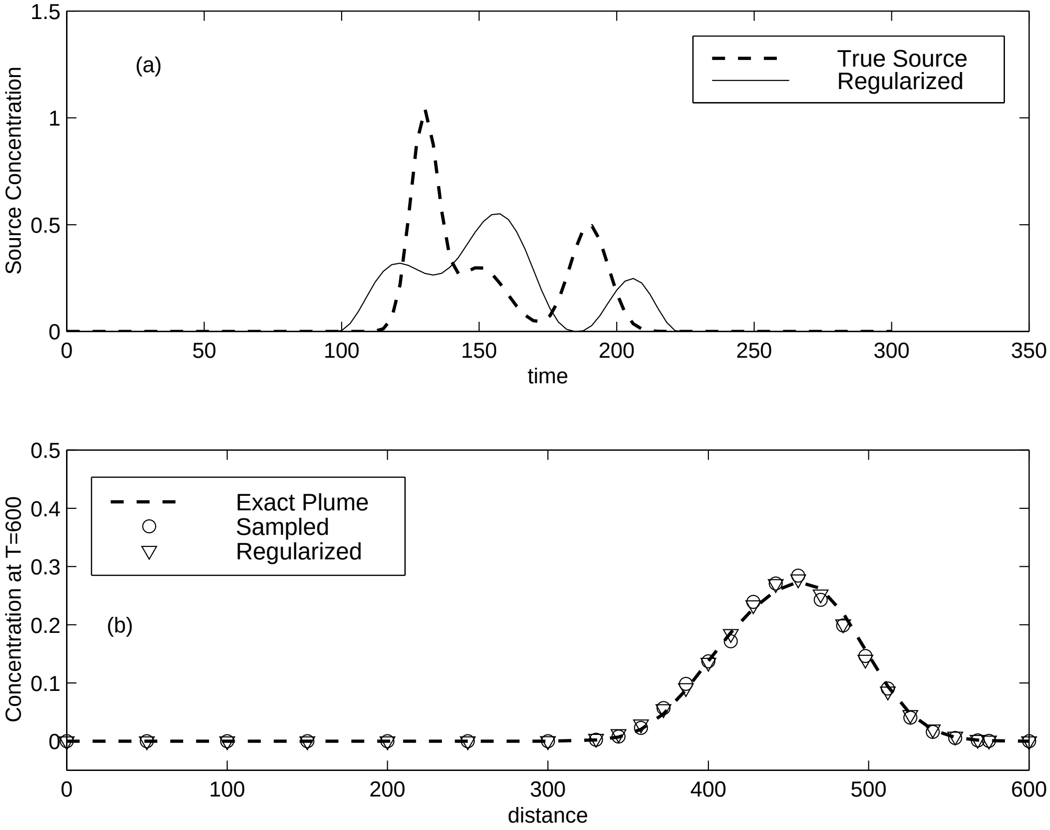

Figure 3.5: Results of Tikhonov regularization verification Run TV-5. (a) Reconstructed source history. (b) Plume at $T = 600$. Parameters are: $\epsilon = 0.05$, $v = 1.0$, $D = 1.0$, and $T = 600$.

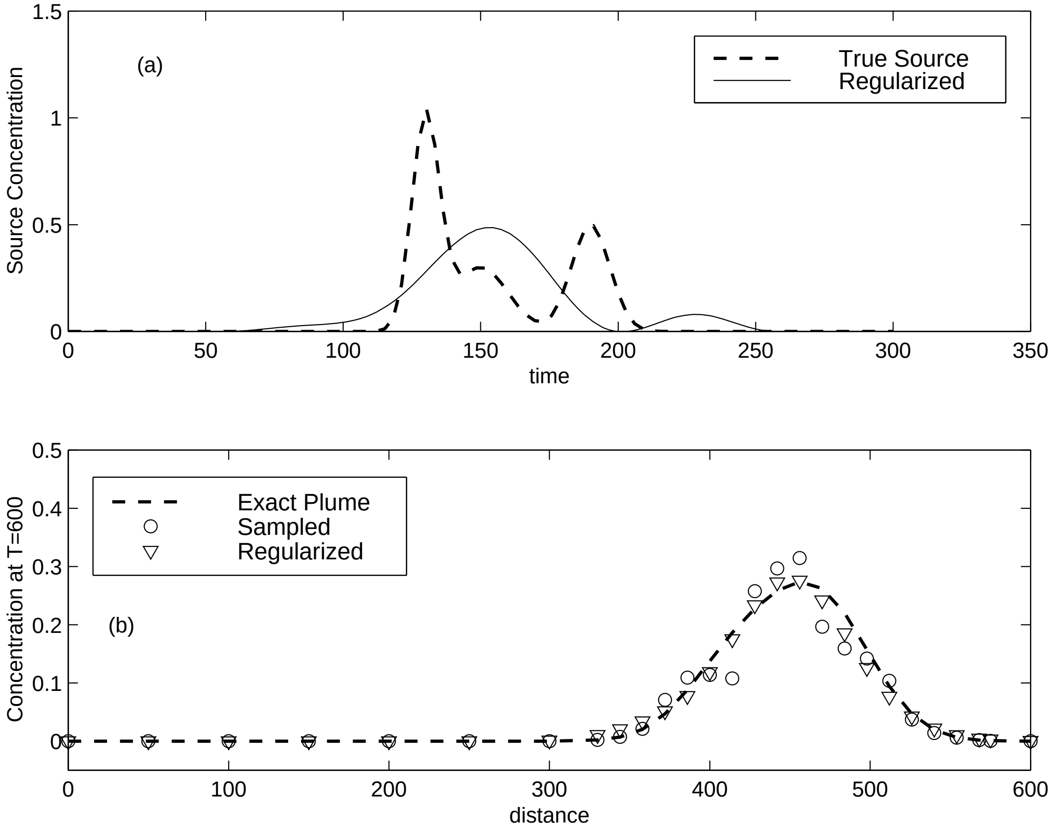

Figure 3.6: Results of Tikhonov regularization verification Run TV-6. (a) Reconstructed source history. (b) Plume at $T = 600$. Parameters are: $\epsilon = 0.2$, $v = 1.0$, $D = 1.0$, and $T = 600$.

run), while the value of the other parameter was unchanged (e.g. Run TV-7 uses $v = 1.05$ and $D = 1.0$). If velocity is overestimated (Figure 3.7), the regularized solution is shifted later in time and is more disperse; the opposite occurs if velocity is underestimated (Figure 3.8). If the dispersion coefficient is overestimated (Figure 3.9), the regularized solution is less disperse; and it is more disperse if the dispersion coefficient is underestimated (Figure 3.10). For these simulations, the reconstructed plume is indistinguishable from the measured data.

3.2 Source History Reconstruction using Minimum Relative Entropy Inversion

We wrote a new MRE program for this research, and we discuss its implementation and verification in this section. The implementation follows the approach described in Section 2.2.2; however several numerical issues arose that may not be apparent from the previous discussion, or from Woodbury and Ulrych (1996). Unless otherwise noted, the issues addressed here were not documented in Woodbury and Ulrych (1996).

3.2.1 Implementation of Minimum Relative Entropy Inversion

We wrote a MATLAB program to implement the minimum relative entropy inversion method, following the general approach described in Section 2.2.2. The program is presented in Appendix C.The basic algorithm is:

1. Obtain the data. The data includes a vector of sampled concentrations, a vector of sampling locations, and the sampling time. All data must be

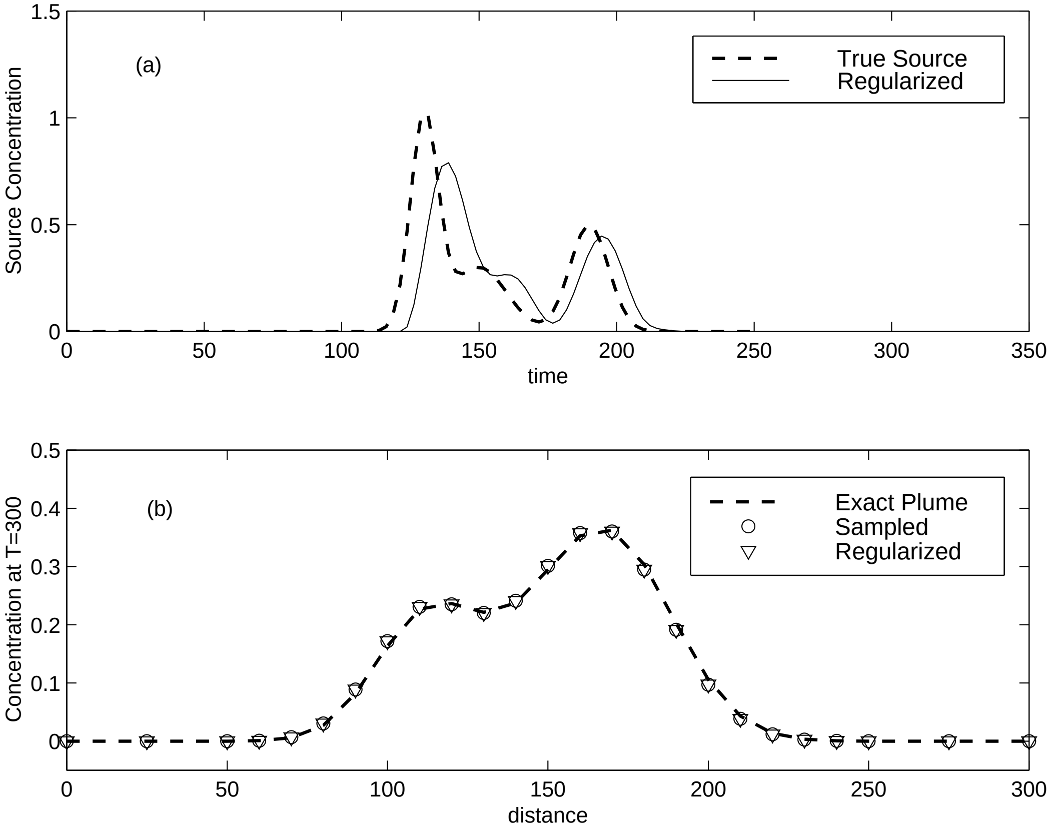

Figure 3.7: Results of Tikhonov regularization verification Run TV-7. (a) Reconstructed source history. (b) Plume at $T = 300$. Parameters are: $\epsilon = 0.0$, $v = 1.05$, $D = 1.0$, and $T = 300$.

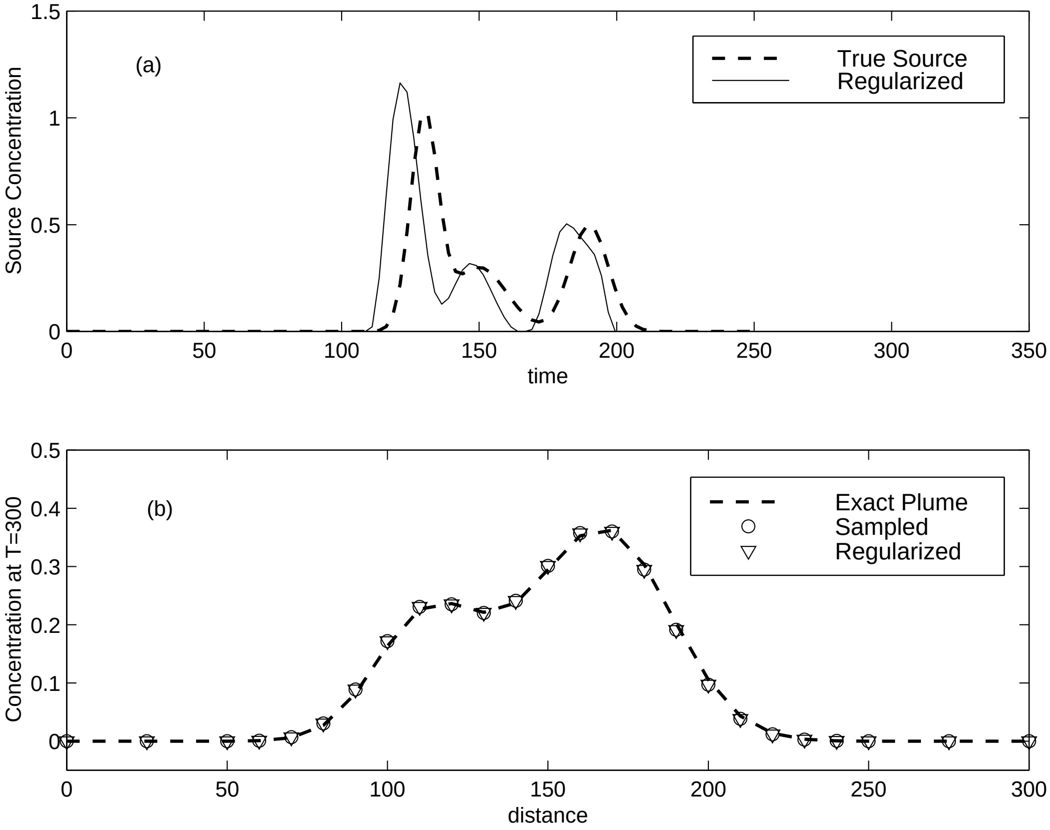

Figure 3.8: Results of Tikhonov regularization verification Run TV-8. (a) Reconstructed source history. (b) Plume at $T = 300$. Parameters are: $\epsilon = 0.0$, $v = 0.95$, $D = 1.0$, and $T = 300$.

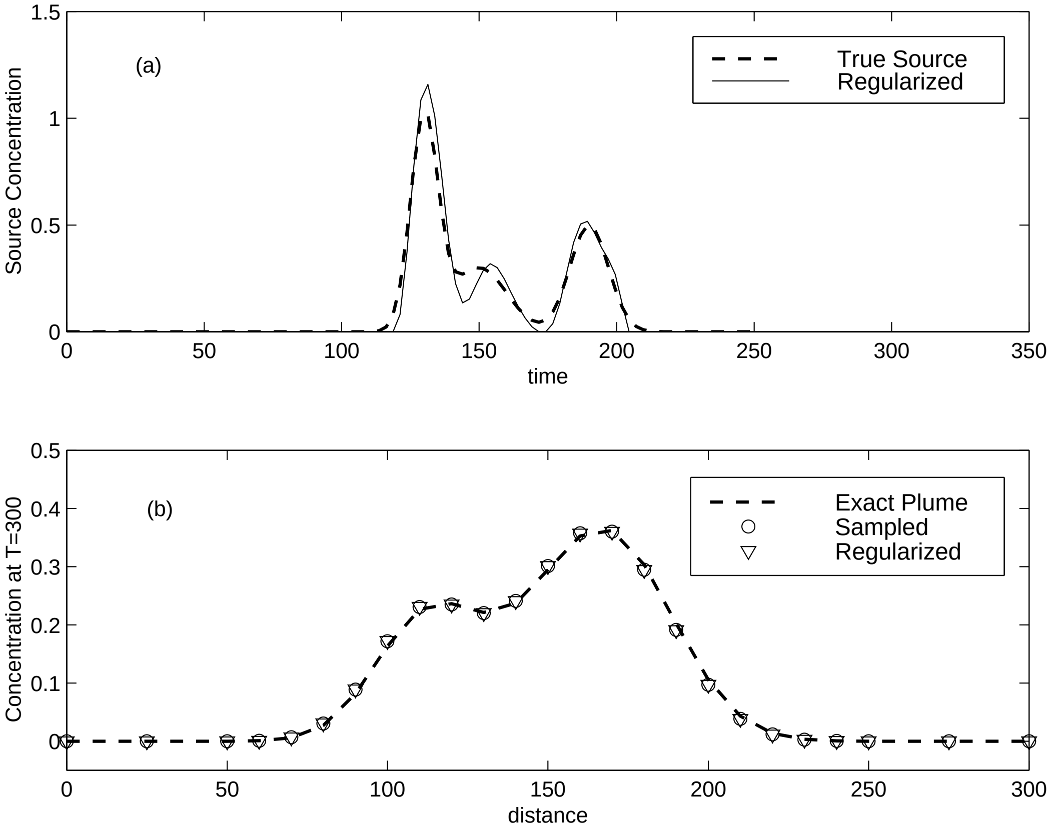

Figure 3.9: Results of Tikhonov regularization verification Run TV-9. (a) Reconstructed source history. (b) Plume at $T = 300$. Parameters are: $\epsilon = 0.0$, $v = 1.0$, $D = 1.05$, and $T = 300$.

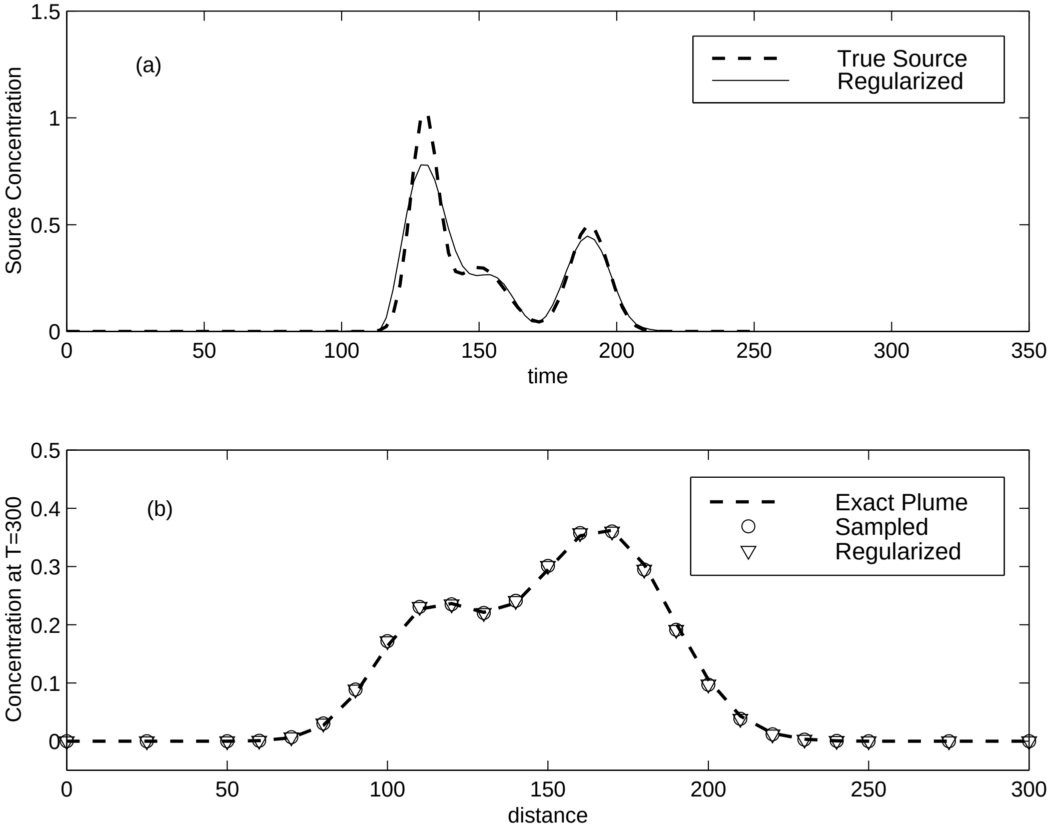

Figure 3.10: Results of Tikhonov regularization verification Run TV-10. (a) Reconstructed source history. (b) Plume at $T = 300$. Parameters are: $\epsilon = 0.0$, $v = 1.0$, $D = 0.95$, and $T = 300$.

sampled at the same time.

2. Select the time discretization for the solution, and choose the upper and lower bounds and the expected value for each element of the solution vector. In the remaining discussion, the elements of the solution vector will be called solution parameters.

3. Calculate the prior distribution of each solution parameter (Equation 2.19).

4. Calculate the posterior distribution of each solution parameter (Equation 2.27).

5. Determine the expected value of the solution parameter from the posterior distribution. This value is the best estimate of the true solution.

Many equations shown in Section 2.2.2 are written in terms of the upper and lower bounds (U_i and L_i, respectively) on the solution parameters. The code was written with $L_i = 0$, for all i. For $L_i \neq 0$, the problem can be re-scaled so that the MATLAB program can be used. Rescaling is done by defining $\hat{\mathbf{m}} = \hat{\mathbf{m}}_o + \mathbf{L}$, where $\hat{\mathbf{m}}$ is the true solution, $\hat{\mathbf{m}}_o$ is the corresponding model solution for a zero lower bound and expected values of $s_i - L_i$, $\mathbf{L}$ is the vector of lower bounds, and s_i is the prior expected value of the i^{th} solution parameter. We can solve for $\hat{\mathbf{m}}_o$ using the MATLAB routine, with the data modified as $\bar{\mathbf{d}}_L = \bar{\mathbf{d}} - \mathbf{GL}$, where $\bar{\mathbf{d}}_L$ is the data vector used in the MRE program, $\bar{\mathbf{d}}$ is the true data vector, and $\mathbf{G}$ is the matrix of kernel values. The upper bounds and expected values must be replaced by $U_i - L_i$ and $s_i - L_i$, respectively. After $\hat{\mathbf{m}}_o$ is computed with the MRE algorithm, the true solution is obtained from $\hat{\mathbf{m}} = \hat{\mathbf{m}}_o + \mathbf{L}$ (Woodbury and Ulrych, 1996).

Items 1 and 2 in the MRE algorithm are done by the user prior to calling the MRE program, and these data are passed into the program. The prior distribution (Item 3) is calculated from Equation 2.19, which minimizes the relative entropy between the prior distribution and the user-specified ranges and expected values of the solution parameters. Equation 2.19 is written in terms of the Lagrange multipliers, β_i. The values for each β_i are determined individually from Equation A.12, which is rewritten here, with $L_i = 0$:

$$s_i = \frac{-(\beta_i U_i + 1)\exp(-\beta_i U_i) + 1}{\beta_i\,[1 - \exp(-\beta_i U_i)]}\ , \tag{3.5}$$

where U_i is the upper bound of solution parameter i, s_i is the expected value of the i^{th} solution parameter, and $i = 1, 2, \ldots, M$, where M is the number of solution parameters.

We use the bisection method (Gill *et al.*, 1981) to solve for the values of β_i is the previous equation. Woodbury and Ulrych (1996) also used the bisection method. The user specifies the initial interval to be used in the bisection method. The bisection method is repeated until the size of the interval is less than a user-specified tolerance, or until the maximum number of iterations (user-specified value) is reached. The final value of β_i is the midpoint of the final interval. Note that the right-hand side of Equation 3.5 is indeterminate for $\beta_i = 0$. In the limit as $\beta_i \to 0$, we have

$$\lim_{\beta_i \to 0} s_i = \frac{U_i}{2}\ . \tag{3.6}$$

Thus, if $s_i = U_i/2$, then the MRE program assigns $\beta_i = 0$. If β_i is small, but non-zero (i.e. if $s_i \approx U_i/2$), the left-hand side of Equation 3.5 is subject to numerical error. To avoid this error, we use an asymptotic approximation to Equation 3.5 for $|\beta_i| < \epsilon_0$:

$$s_i \approx \frac{12U_i - 8\beta_i U_i^2 + 3\beta_i^2 U_i^3}{24 - 12\beta_i U_i + 4\beta_i^2 U_i^2 - \beta_i^3 U_i^3} \; . \tag{3.7}$$

In the MRE algorithm, we use $\epsilon_0 = 10^{-4}$.

The left-hand side of Equation 3.5 is also subject to numerical error when $s_i \approx U_i$. If $s_i \approx U_i$, then $\beta_i \to -\infty$ ($\beta_i < -10^2$ in our MRE algorithm), and we use the asymptotic approximation of $\beta_i = -1/(U_i - s_i)$. If s_i is identically equal to U_i, the corresponding value of β_i would be $-\infty$. The prior distribution, $p(m_i)$, is defined such that $s_i = \mathrm{E[m_i]}$, given that m_i has the probability distribution $p(m_i)$ over the range from $m_i = 0$ to $m_i = U_i$. Thus, if $s_i = U_i$, $p(m_i) = \delta(m_i - U_i)$ (where $\delta(m_i)$ is the Dirac delta function), and the true solution will always be equal to the upper bound. Therefore, we must have $s_i < U_i$ for all i. Note that if $s_i \approx L_i$, then $\beta_i \to \infty$; however, since $L_i = 0$, this situation does not cause numerical error and no asymptotic approximations are necessary. With these approximations, the bisection method converges quickly to a solution.

The prior distribution, $p(\mathbf{m})$, is shown in Equation 2.19 in terms of the Lagrange parameters, β_i. If $\beta_i \to -\infty$, the expressions shown in Equation 2.19 cannot be evaluated numerically, and asymptotic approximations must be used (Recall that if $\beta_i \to \infty$, then $s_i \approx 0$, and an asymptotic approximation is not necessary). For $\beta_i \to -\infty$ ($\beta_i < -10^2$), the prior distribution is given by

$$p(m_i) = \begin{cases} -\beta_i \left[1 + \beta_i \left(U_i - m_i\right)\right] & U_i + 1/\beta_i \le m_i \le U_i \; , \\ 0 & \text{otherwise} \; . \end{cases} \tag{3.8}$$

The posterior distribution (Item 4) is calculated from Equation 2.27, which minimizes the relative entropy between the posterior distribution and the prior distribution. Equation 2.27 is written in terms of $a_i = \beta_i + \sum_{j=1}^{N} \lambda_j g_{ji}$, where β_i were addressed in Item 3, λ_j are Lagrange parameters, g_{ji} is the j, i^{th} element in the discretized kernel matrix, and $i = 1, 2, \ldots, M$.

Note that Equation 2.27 was obtained by minimizing the entropy of the posterior distribution relative to the prior distribution shown in Equation 2.19. However, if $\beta_i \rightarrow -\infty$, an asymptotic approximation to the prior distribution is used (Equation 3.8). For $\beta_i \rightarrow -\infty$ ($\beta_i < -10^2$), the asymptotic approximation of the posterior distribution is

$$q(m_i) = \begin{cases} \begin{array}{l} \frac{2}{3}\beta_i[1 + (b_i + \beta_i)(U_i - m_i) + \\ \qquad b_i(U_i - m_i)^2(\frac{b_i}{2} + \beta_i)] \end{array} & U_i - 1/\beta_i \le m_i \le U_i \ , \\ 0 & \text{otherwise} \ , \end{cases} \tag{3.9}$$

where $b_i = \sum_{j=1}^{N} \lambda_j g_{ji}$.

The values of $\boldsymbol{\lambda}$ ($\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_N]^T$, where T denotes transpose) are calculated from Equation B.11 using the Newton-Raphson method. Equation B.11 is restated here for completeness:

$$F(\boldsymbol{\lambda})_j = \bar{d}_j - \sum_{i=1}^{M} g_{ji}\hat{m}_i(\boldsymbol{\lambda}) + \xi\epsilon\frac{\lambda_j}{||\boldsymbol{\lambda}||} \ , \tag{3.10}$$

where $\bar{d}_j$ is the j^{th} measured data point, ϵ is the standard deviation of the measurement error, ξ is the scaling factor in the data constraint (see Equation B.9), and $\hat{m}_i(\boldsymbol{\lambda})$ is the expected value of the i^{th} solution parameter. Equation 2.28 shows the expression for $\hat{m}_i$ when the posterior distribution can be expressed as Equation 2.27. However, when the asymptotic approximation shown in Equation 3.9 is used for the posterior distribution, an asymptotic approximation of

$\hat{m}_i$ must be used. For $\beta_i \rightarrow -\infty$ ($\beta_i < -10^2$), we have

$$\hat{m}_i = \frac{2}{3}U_i - \frac{1-(b_i+\beta_i)U_i}{3\beta_i} + \frac{2}{9\beta_i^2}\left[U_i b_i \left(\frac{b_i}{2}+\beta_i\right) - b_i - \beta_i\right] \ . \qquad (3.11)$$

When a_i approaches zero, or $\pm\infty$, solving for $\hat{m}_i$ using Equation 2.28 is subject to numerical error. To avoid this error, we use asymptotic approximations to these limiting values. For $a_i \rightarrow 0$ ($|a_i| < 10^{-4}$), an asymptotic approximation to Equation 2.28 is

$$\hat{m}_i \approx \frac{12U_i - 8a_iU_i^2 + 3a_i^2U_i^3}{24 - 12a_iU_i + 4a_i^2U_i^2 - a_i^3U_i^3} \ . \qquad (3.12)$$

For $a_i \rightarrow -\infty$ ($a_i < -10^2$), the asymptotic approximation is $\hat{m}_i \approx U_i - 1/a_i$; and for $a_i \rightarrow +\infty$ ($a_i > 10^2$), the asymptotic approximation is $\hat{m}_i \approx 1/a_i$.

The Newton-Raphson method iteratively solves for the zeroes of Equation 3.10 using (Equation B.12)

$$\boldsymbol{\lambda}^k = \boldsymbol{\lambda}^{k-1} - \left(\left.\frac{\partial \mathbf{F}}{\partial \boldsymbol{\lambda}}\right|_{\boldsymbol{\lambda}^{k-1}}\right)^{-1} \mathbf{F}^{k-1} \ , \qquad (3.13)$$

where the superscripts denote the iteration, and the terms of the Jacobian matrix, $\partial\mathbf{F}/\partial\boldsymbol{\lambda}$, are shown in Equation B.13. The initial values of $\boldsymbol{\lambda}$ are $\lambda_j = 1$. Equation 3.10 is sufficiently non-linear so that the step size calculated from the previous equation is not necessarily optimal. Therefore, we use the Newton-Raphson method to calculate the optimal step direction; then, we use a univariate golden section search (Gill *et al.*, 1981) in that direction to calculate the step length that minimizes $||\mathbf{F}(\boldsymbol{\lambda})||$. The process is repeated until $||\mathbf{F}||/(1+||\bar{\mathbf{d}}||)$ is less than a user-specified tolerance. We have found that the condition number of the Jacobian matrix can be high when $\epsilon = 0$. To reduce the condition number, we perform row-scaling on the Jacobian matrix.

It is possible to substantially reduce the condition number by including a small amount of noise, at a level that is negligible compared to the true data values.

The equation for the elements of the Jacobian matrix (Equation B.13) contains the partial derivative $\partial\hat{m}_i/\partial a_i$, which is given by Equation B.14 and repeated here (with $L_i = 0$):

$$\begin{aligned} \frac{\partial\hat{m}_i}{\partial a_i} &= \frac{a_i^2 U_i^2 \exp\left(-a_i U_i\right) - \left[1 - \exp(-a_i U_i)\right]^2}{a_i^2 \left[1 - \exp(-a_i U_i)\right]^2} \qquad \text{for } a_i \neq 0 \\ \frac{\partial\hat{m}_i}{\partial a_i} &= -\frac{U_i^2}{12} \text{ for } a_i = 0 \ . \end{aligned} \tag{3.14}$$

This equation is subject to numerical error when a_i approaches zero, or $\pm\infty$; therefore, we use asymptotic approximations to these limiting values. For $a_i \rightarrow 0$ ($|a_i| < 10^{-4}$), an asymptotic approximation to Equation 3.14 is

$$\frac{\partial\hat{m}_i}{\partial a_i} = -U_i^2 \frac{15 - 15a_i U_i + 8a_i^2 U_i^2}{180 - 180a_i U_i + 105a_i^2 U_i^2} \tag{3.15}$$

For $a_i \rightarrow \pm\infty$ ($|a_i| > 10^2$), an asymptotic approximation to Equation 3.14 is

$$\frac{\partial\hat{m}_i}{\partial a_i} = -\frac{1}{a_i^2} \ . \tag{3.16}$$

When $\beta_i \rightarrow -\infty$, the partial derivative, $\partial\hat{m}_i/\partial a_i$, in Equation B.13 is replaced with $\partial\hat{m}_i/\partial b_i$. From Equation 3.11, we see that as $\beta_i \rightarrow -\infty$ ($\beta_i < -10^2$),

$$\frac{\partial\hat{m}_i}{\partial b_i} = \frac{5U_i}{9\beta_i} - \frac{2(1 - 2b_i U_i)}{9\beta_i^2} \ . \tag{3.17}$$

After the optimal values of the Lagrange parameters, $\boldsymbol{\lambda}$, are obtained, these values are used in Equation 2.28 or Equation 3.11 to calculate the best estimate of the solution parameters, $\hat{\mathbf{m}}$. This step (Item 5) completes the basic MRE algorithm.

Table 3.2: Details of MRE verification runs.

Run Number	Expected Value (s_i) [-]	Sample Time (T) [days]	Noise Level (ϵ) [-])
MV-1	boxcar	300	0
MV-2	exponential	300	0
MV-3	Gaussian	300	0
MV-4	boxcar	600	0
MV-5	Gaussian	300	0.005
MV-6	Gaussian	300	0.01
MV-7	Gaussian	300	0.05

3.2.2 Verification of Minimum Relative Entropy Routine

We verified the MATLAB MRE program by running the same examples described in Woodbury and Ulrych (1996), and comparing our results to their reported results. Woodbury and Ulrych (1996) presented results of seven examples. The details of these examples are outlined in Table 3.2.

For runs using data sampled at $T = 300$ days, the sampling locations are at 5 m-intervals between 5 m and 300 m. For sampling at $T = 600$ days, the sampling locations were at 5 m-intervals between 300 m and 600 m (Woodbury and Ulrych, 1996). For runs with inexact data, random measurement error was added using $C_{\text{meas}}(x_j, T) = C_{\text{exact}}(x_j, T) + \epsilon\delta_j$. Woodbury and Ulrych (1996) used two methods for handling inexact data with the MRE. In the first method, described in Section 2.2.2 and Appendix B, the data constraint in the MRE method is modified to account for the inexact data (the noise level must be specified). In the second method, the raw data are pre-filtered using a Butterworth low-pass filter to remove the high-frequency noise components.

The filtered data are then treated as noise-free in the MRE method. In this research, we used the first method for handling inexact data; we were unable to reproduce the results of the filtering method.

For sampling at $T = 300$ days, the time domain was discretized into 100 uniformly-spaced intervals between $t = 0$ days and $t = 300$ days (Woodbury and Ulrych, 1996). For sampling at $T = 600$ days, the time domain was discretized into 200 uniformly-spaced intervals between $t = 0$ days and $t = 600$ days.[2] The expected value functions were discretized to correspond to these discretized times. The equation used for the boxcar expected value function was

$$s_i = \begin{cases} \zeta & 100 \leq t_i \leq 225 \ , \\ 0 & \text{otherwise} \ , \end{cases} \tag{3.18}$$

where $\zeta = 0.8$ for Run MV-1 (Woodbury and Ulrych, 1996), $\zeta = 0.49$ for Run MV-4, [3] and t_i is in units of days. The equation used for the exponential expected value function was [4]

$$s_i = \exp\left[-\frac{0.69 t_i}{50}\right] \ , \tag{3.19}$$

where t_i is in units of days. The equation used for the Gaussian expected value function was [5]

$$s_i = \exp\left[-\frac{(t_i - 150)^2}{2(20)^2}\right] \ , \tag{3.20}$$

[2] Personal communication with A.D. Woodbury, February 2, 1999.

[3] Ibid.

[4] Ibid.

[5] Ibid.

where t_i is in units of days. The lower and upper bounds on all model parameters were 0 and 1.1, respectively. [6] For all runs, the stopping tolerance was 10^{-4} for the β calculations, and 10^{-2} for the $\boldsymbol{\lambda}$ calculations.

The results of the seven verification examples shown in Table 3.2 were compared to those of Woodbury and Ulrych (1996). For each simulation, we present plots showing the true and fitted source history functions, along with the expected value function used in the simulation, and the 5th and 95th percentile probability levels, which corresponds to the 90% probability interval. For each simulation, we also present the true plume, sampled data, and fitted plume.

In Runs MV-1—MV-3, we used perfect data sampled at $T = 300$ days and three different expected value functions (boxcar, exponential, and Gaussian, respectively). The results are shown in Figures 3.11—3.13, for Runs MV-1 through MV-3, respectively. In all cases, the reconstructed source history agrees well with the true source history; the only difference is that the magnitude of the first peak is slightly increased. These results are similar to those of Woodbury and Ulrych (their Figures 2, 4, and 5). The 5th and 95th percentile probability levels bracket the true source history function at all times, although the results of Woodbury and Ulrych show a narrower spread in the probability levels. The sampled and fitted spatial plume concentrations are nearly identical.

In Run MV-4, we used perfect data sampled at $T = 600$ days, and a

[6] Ibid.

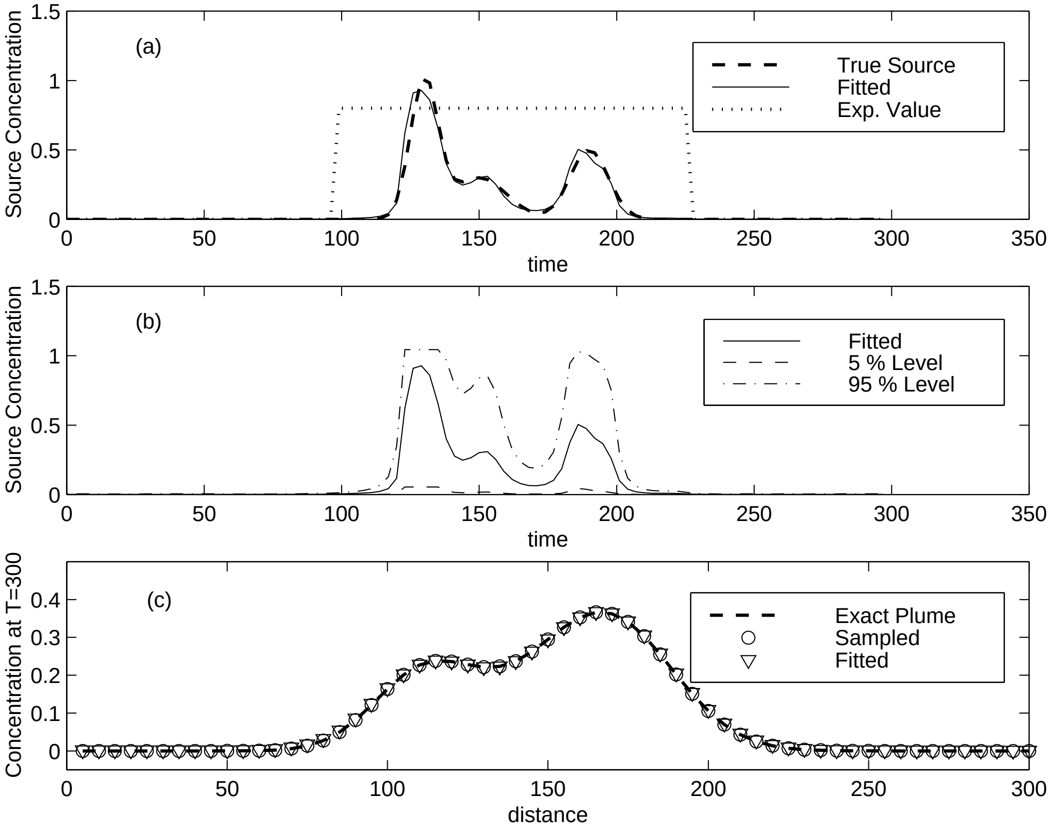

Figure 3.11: Results of MRE verification Run MV-1. (a) Reconstructed source history. (b) Reconstructed solution and the 5^{th} and 95^{th} percentile probability levels. (c) Plume at $T = 300$ days. Parameters are: $\epsilon = 0.0$, s_i = boxcar, and $T = 300$.

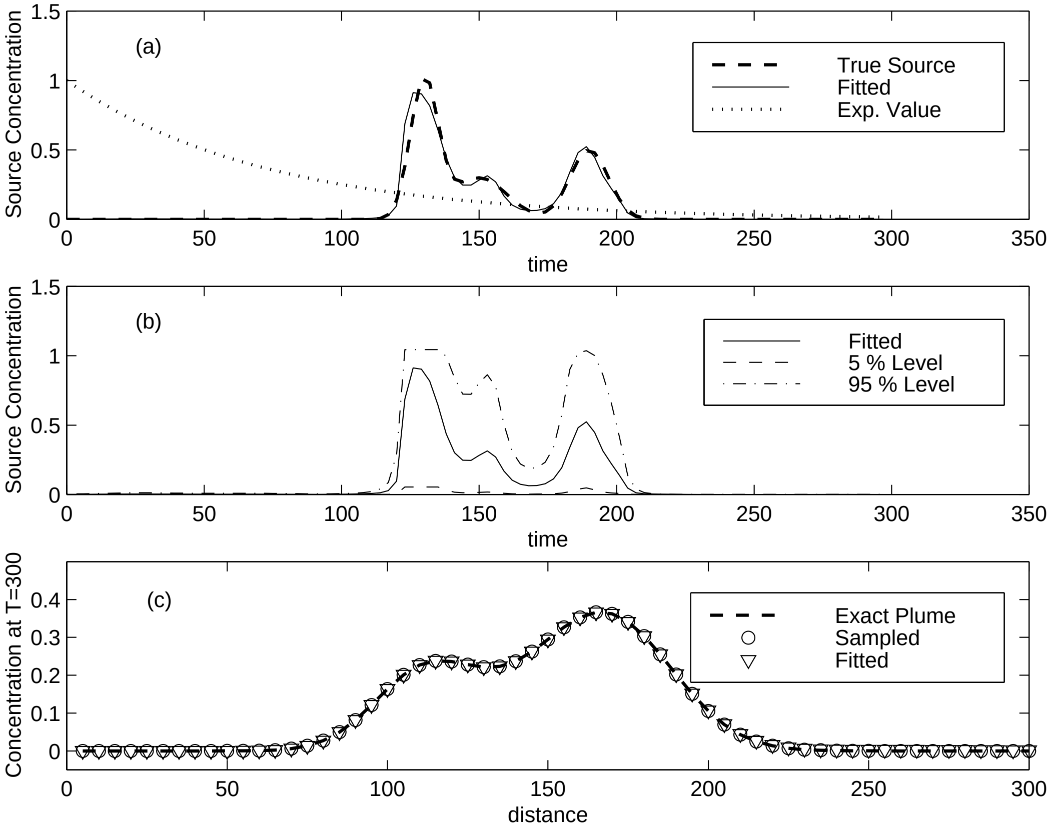

Figure 3.12: Results of MRE verification Run MV-2. (a) Reconstructed source history. (b) Reconstructed solution and the 5th and 95th percentile probability levels. (c) Plume at $T = 300$ days. Parameters are: $\epsilon = 0.0$, s_i = exponential, and $T = 300$.

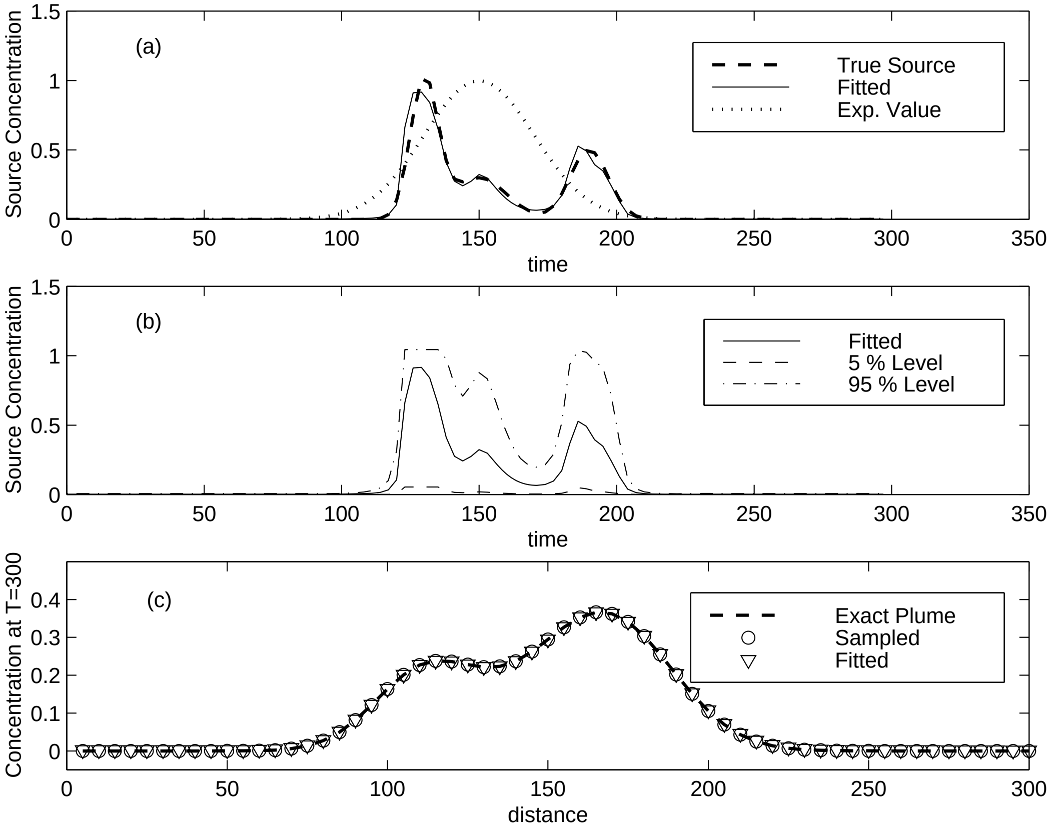

Figure 3.13: Results of MRE verification Run MV-3. (a) Reconstructed source history. (b) Reconstructed solution and the 5^{th} and 95^{th} percentile probability levels. (c) Plume at $T = 300$ days. Parameters are: $\epsilon = 0.0$, s_i = Gaussian, and $T = 300$.

boxcar expected value function. The results are shown in Figure 3.14. The reconstructed source history has a similar structure as the true solution; however, the timing of the first peak is shifted slightly, and the magnitudes of the first and third peaks are slightly inaccurate. The results presented by Woodbury and Ulrych (their Figure 7) for this case are smoother, with the first and second peaks indistinguishable. The difference may be caused by using different stopping criteria, different expressions for the asymptotic approximations, or by using different solution methods for computing the optimal values of $\boldsymbol{\lambda}$. As with Runs MV-1—MV-3, the 5^{th} and 95^{th} percentile probability levels bracket the true source history functions at all times, but those of Woodbury and Ulrych have a narrower spread. Again, the sampled and fitted spatial plume concentration are nearly identical.

In Runs MV-5—MV-7, we used inexact data sampled at $T = 300$ days and a Gaussian expected value function. The noise levels were 0.005, 0.01, and 0.05, respectively. The results are shown in Figures 3.15—3.17, for Runs MV-5 through MV-7, respectively. In all cases, the middle peak is essentially indistinguishable from the first peak. The only exception is the reconstructed source history for Run MV-7, which shows the middle peak; however, this peak is likely a result of the overestimated value of the concentration sampled near $x = 140$ (See Figure 3.17c). In general, the timing of the first and third peaks in Runs MV-5—MV-7 agree well with the true source history; however the magnitudes of these peaks are less accurate. The reconstruction of the magnitudes of the peaks becomes worse as the noise level increases. The 5^{th} and 95^{th} percentile probability levels bracket the true source at all times, and the fitted spatial plume concentrations match the data to within the noise level. These

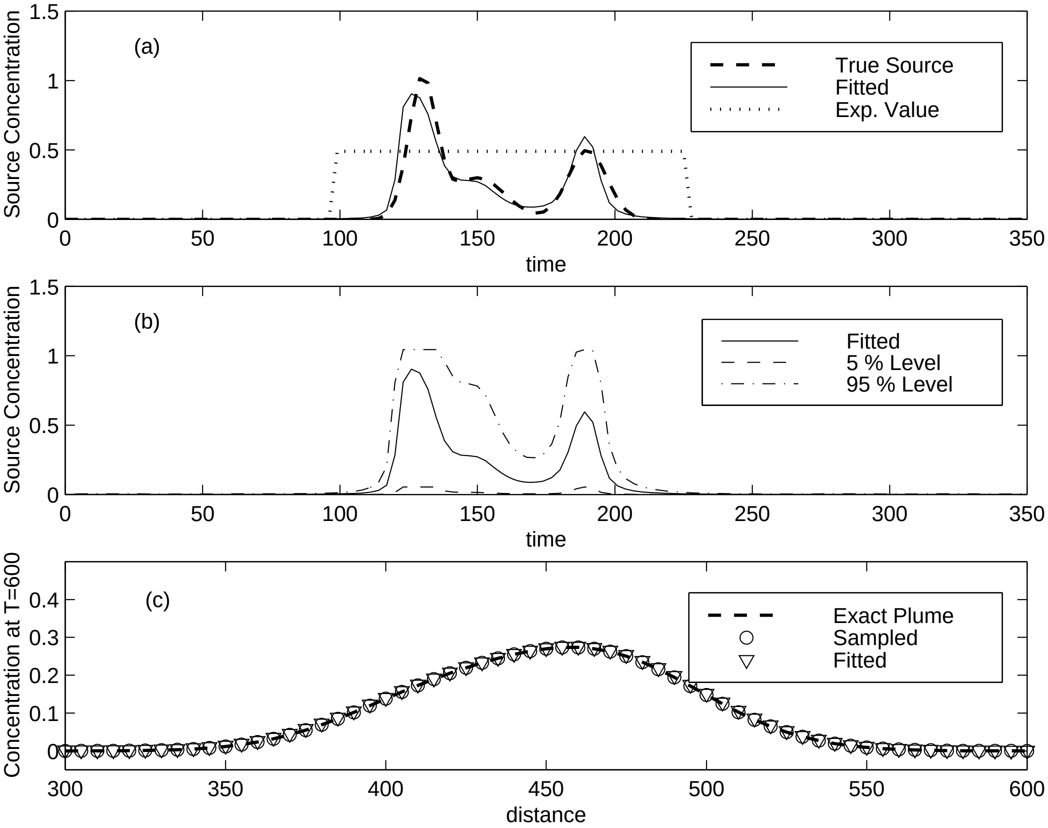

Figure 3.14: Results of MRE verification Run MV-4. (a) Reconstructed source history. (b) Reconstructed solution and the 5^{th} and 95^{th} percentile probability levels. (c) Plume at $T = 600$ days. Parameters are: $\epsilon = 0.0$, s_i = boxcar, and $T = 600$.

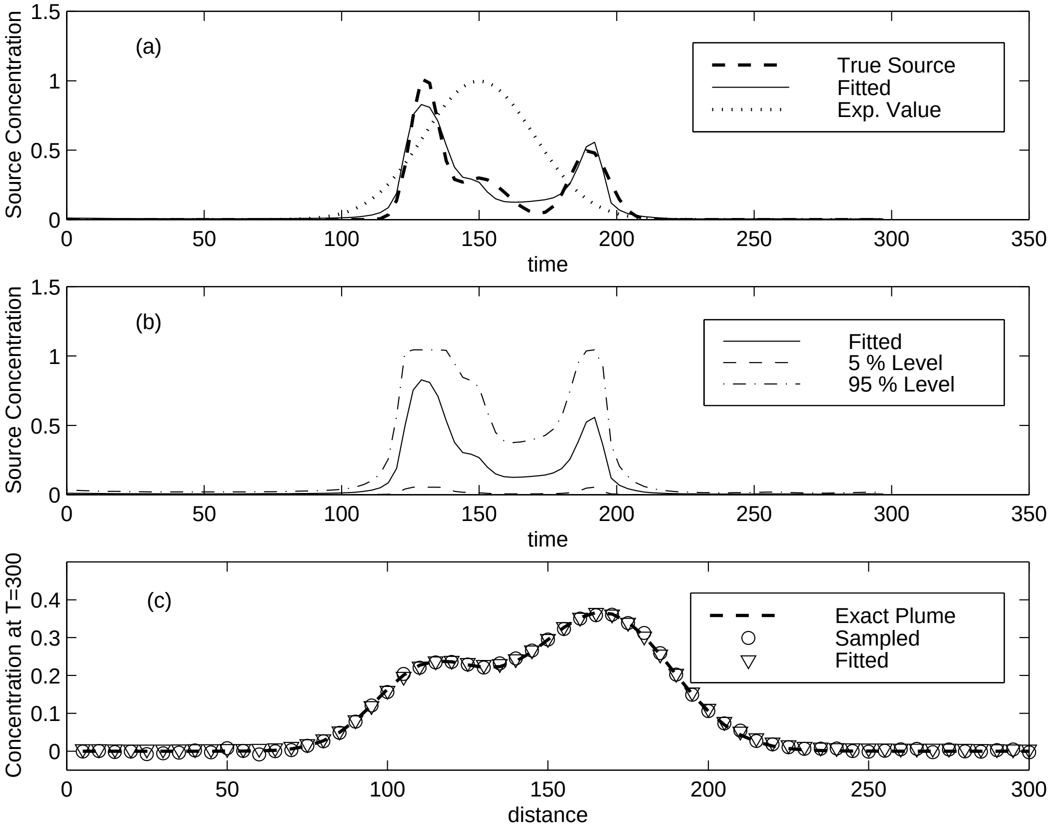

Figure 3.15: Results of MRE verification Run MV-5. (a) Reconstructed source history. (b) Reconstructed solution and the 5$^{\text{th}}$ and 95$^{\text{th}}$ percentile probability levels. (c) Plume at $T = 300$ days. Parameters are: $\epsilon = 0.005$, $s_i =$ Gaussian, and $T = 300$.

results are similar to those of Woodbury and Ulrych (their Figures 10, 11, and 14); however they were able to recapture to some degree the middle peak in the true source history function.

We cannot directly compare our results to those of Woodbury and Ulrych (1996) because the random noise is different in each case. Each set of random noise produces a different set of inexact data, and therefore produces a different solution to the source history reconstruction problem. Therefore,

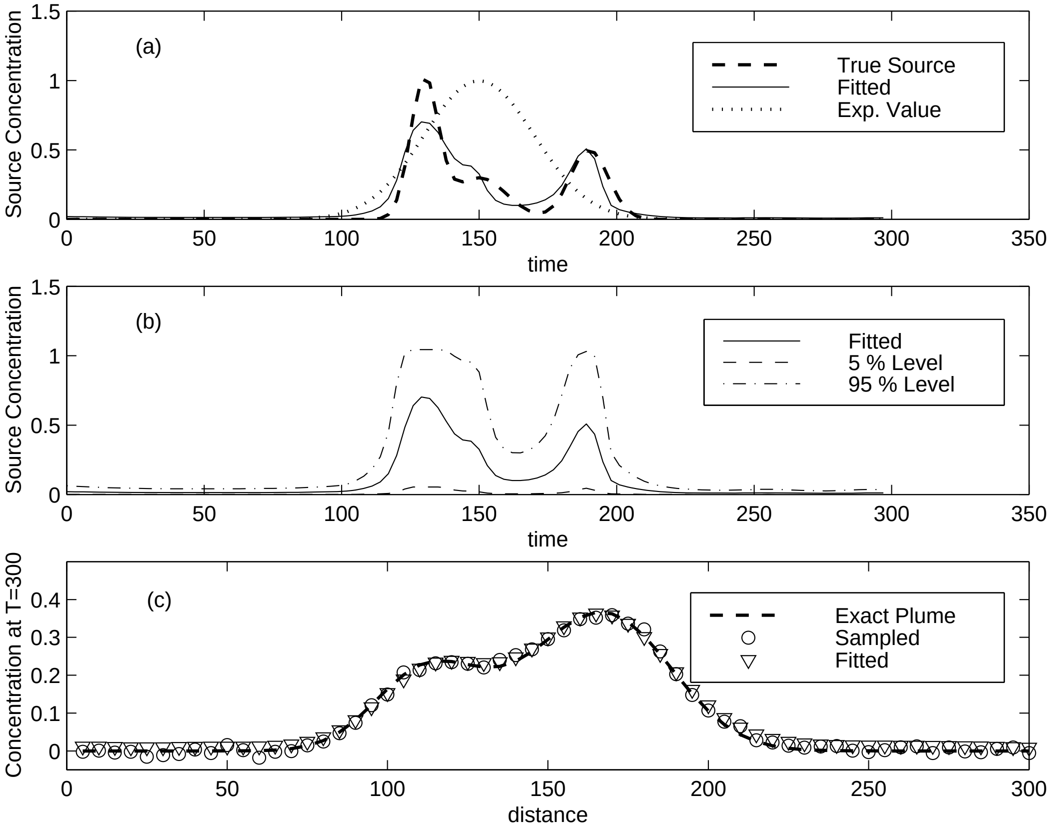

Figure 3.16: Results of MRE verification Run MV-6. (a) Reconstructed source history. (b) Reconstructed solution and the 5$^{\text{th}}$ and 95$^{\text{th}}$ percentile probability levels. (c) Plume at $T = 300$ days. Parameters are: $\epsilon = 0.01$, $s_i =$ Gaussian, and $T = 300$.

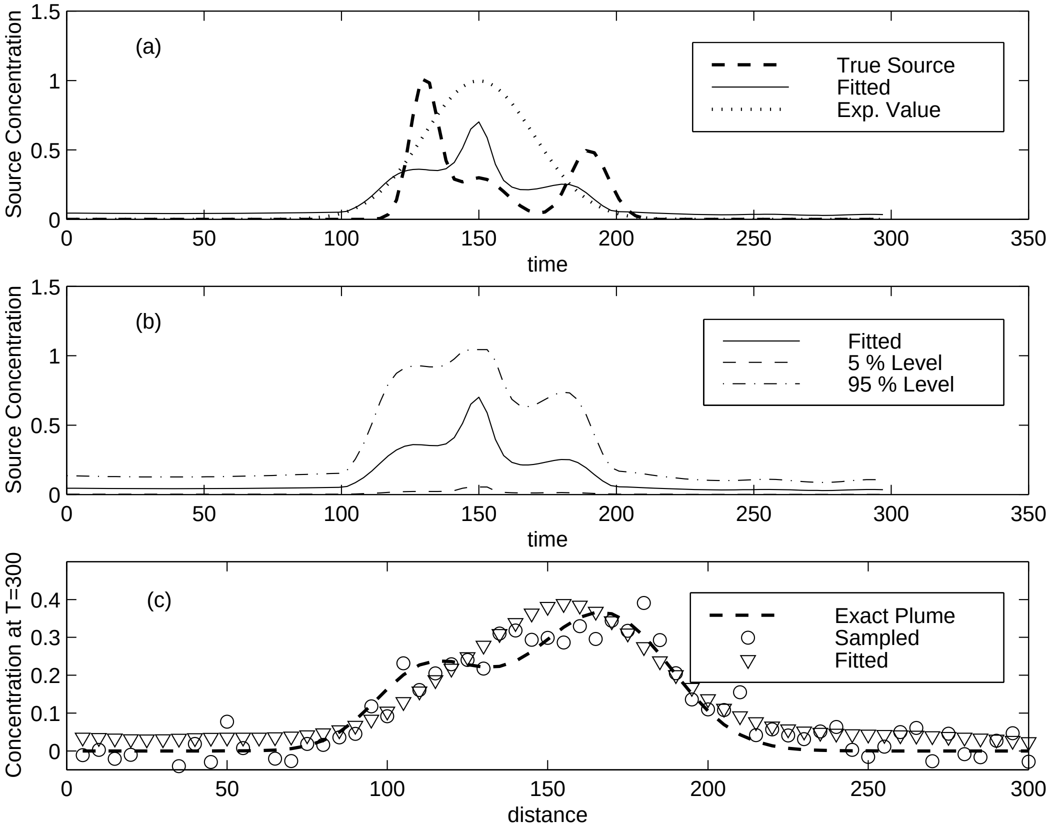

Figure 3.17: Results of MRE verification Run MV-7. (a) Reconstructed source history. (b) Reconstructed solution and the 5th and 95th percentile probability levels. (c) Plume at $T = 300$ days. Parameters are: $\epsilon = 0.05$, s_i = Gaussian, and $T = 300$.

the shape of the reconstructed release history and its agreement with the true source history will be different for each set of random noise, and we cannot compare the results.

3.3 Summary of Implementation and Verification

The results presented here verify that we can reproduce the results of Skaggs and Kabala (1994) using their input files and the CONTIN code. The results also verify that the MRE algorithm has been implemented correctly, since our results are similar to those of Woodbury and Ulrych (1996) for the same cases. Also, the results are consistent with the expected results—i.e. the reconstructed source history is very close to the true source history when perfect data are used, and becomes less accurate as the noise level of the data increases.

Chapter 4

Evaluation of Parameter Selection for the Inverse Methods

The purpose of this research is to provide a thorough and unbiased comparison of two methods (Tikhonov regularization and minimum relative entropy inversion) for reconstructing the release history of a groundwater contamination source. As discussed in Section 2.3, each of the methods contain some subjectivity. With Tikhonov regularization, the choice of the regularization parameter and the order of regularization is subjective; and with MRE, the upper and lower bounds and expected values for the solution parameters are subjective. In this chapter, we present results of several analyses regarding the sensitivity of the inverse solution to these subjective parameters.

4.1 Evaluation of Parameter Selection for Tikhonov Regularization

The solution to the source history reconstruction problem using Tikhonov regularization depends on the value selected for the regularization parameter, α, and on the regularization order. In this section, we evaluate the effects of these two parameters on the solution to the inverse problem.

4.1.1 Evaluation of the Regularization Parameter Selection

The regularization parameter, α, determines the trade-off between matching the data and regularizing the problem. A high value of the regularization parameter results in a smooth solution; while a low value results in a solution that matches the data more closely. Several methods of selecting the optimal regularization parameter were discussed in Section 2.1.2. In general, for a given problem, each method will define a different optimal value of the regularization parameter. To analyze the sensitivity of the reconstructed source history to the regularization parameter selection method, we used two different selection methods—the F-test method and generalized cross-validation (GCV).

In CONTIN, the regularization parameter is selected using the F-test method. CONTIN systematically chooses $2n_\alpha$ different values of α, where n_α is user-specified (the default value is $n_\alpha = 6$), and obtains the regularized solution for each of these values. For each α, the value of the F ratio (Equation 2.8) is calculated, and the regularized solution is printed in the output file. The optimal solution is the one for which $P(\alpha) = P(F(\alpha), \nu_1, \nu_2)$ is closest to 0.5 (parameters are defined in Section 2.1.2).

To implement the generalized cross-validation method, we used each of the $2n_\alpha$ values of the regularization parameter selected by CONTIN and the corresponding regularized solutions, to evaluate the ratio, $G(\alpha)$, shown in Equation 2.7. The chosen value of α was the value of these that minimized $G(\alpha)$.

We tested four different scenarios to determine the effect of the regularization parameter selection method on the solution to the inverse problem.

Table 4.1: Test scenarios for the analysis of the regularization parameter selection methods.

Run Number	True Source History Function	Noise Level (ϵ)
RP-1	Square	0
RP-2	Smooth	0
RP-3	Square	0.05
RP-4	Smooth	0.05

The four scenarios are shown in Table 4.1. In all cases, twenty-five data points were sampled at $T = 300$ ($\mathbf{x} = [0.01, 25, 50, 60, 70, \ldots, 240, 250, 275, 300]$), and the transport parameters were v=1.0 and D=1.0. The "square" input function was

$$C_{in}(t_i) = \begin{cases} 1.0 & 125 \leq t_i \leq 225 \ , \\ 0 & \text{otherwise} \ . \end{cases} \tag{4.1}$$

The "smooth" input function was used in the verification runs in Chapter 3, and is shown in Equation 3.4 and Figure 1.1. The source history function was discretized into 100 uniformly-spaced intervals between $t = 0.01$ and $t = 250$. The inexact data were generated using $C_{\mathrm{meas}}(x_j, T) = C_{\mathrm{exact}}(x_j, T) + \epsilon\delta_j C_{\mathrm{exact}}(x_j, T)$. The true plumes and sampled data are shown in Figures 4.1 and 4.2, for the square input and smooth input, respectively.

The results were evaluated by comparing the L$_2$ norm of the differences between the true input function and the regularized solution. These norms were normalized by the total number of discretized intervals in the source history function (100).

Run RP-1 used perfect data and a square input function. The optimal

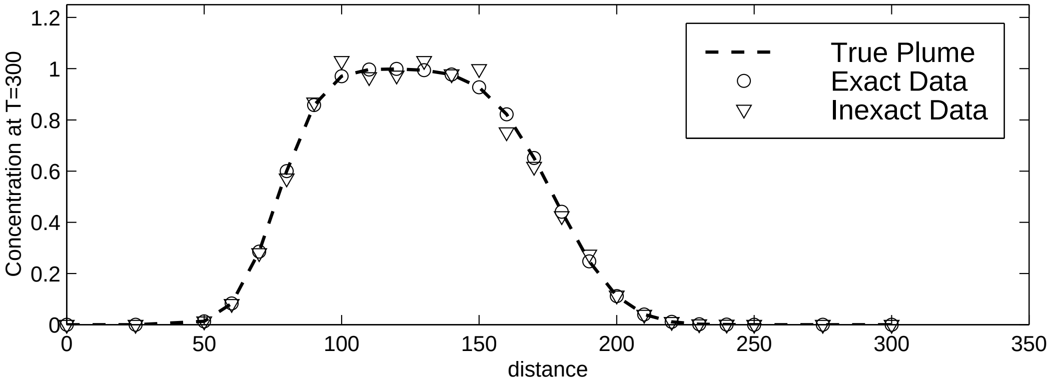

Figure 4.1: True plume and exact and inexact sample measurements for regularization parameter selection Runs RP-1 and RP-3 with a square input function.

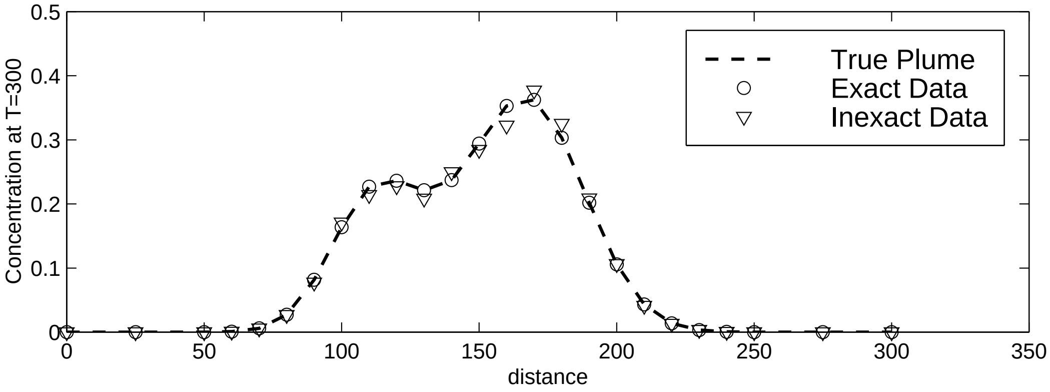

Figure 4.2: True plume and exact and inexact sample measurements for regularization parameter selection Runs RP-2 and RP-4 with a smooth input function.

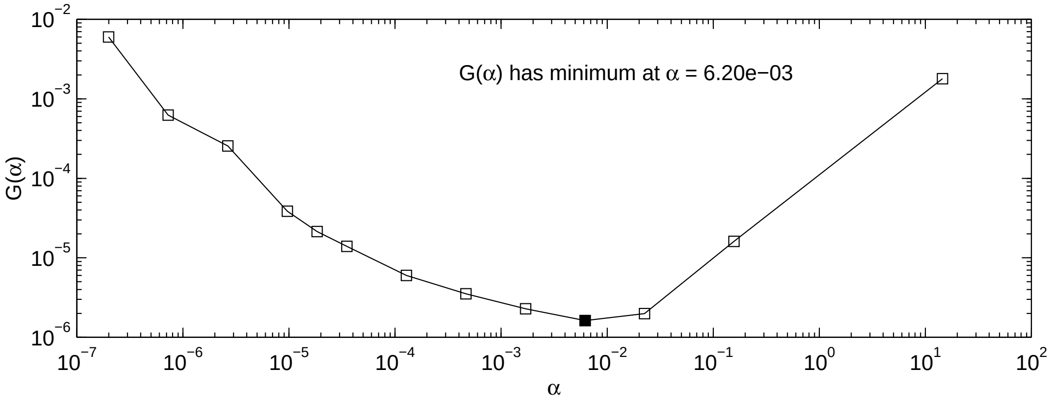

Figure 4.3: Results of generalized cross-validation for regularization parameter selection Run RP-1. The filled square corresponds to the optimal value of α.

value of α using the F-test method from CONTIN was $\alpha = 1.84 \times 10^{-5}$, and $P(\alpha) = 0.639$. The results of the generalized cross-validation method are shown in Figure 4.3, with the optimal value of $\alpha = 6.20 \times 10^{-3}$. The results of the simulations with each of these parameter values are shown in Figure 4.4 and in Table 4.2. The solution using the GCV-selected α is smoother (because α is larger, and the solution is more regularized); but it does not match the sharp increase and decrease as well. The residual norms in both cases are similar; therefore both parameter selection methods perform equally well.

Run RP-2 used perfect data and a smooth input function. The optimal value of α using the F-test method from CONTIN was $\alpha = 3.51 \times 10^{-5}$, and $P(\alpha) = 0.442$. The results of the generalized cross-validation method are shown in Figure 4.5, with the optimal value of $\alpha = 1.70 \times 10^{-3}$. The results of the simulations with each of these parameter values are shown in Figure 4.6 and in Table 4.2. The solution using the GCV-selected α is smoother, and does not match the true source history as well as the solution using the F-test–

Table 4.2: Results of the analysis of the regularization parameter selection methods.

Run Number	Regularization Parameter (α)		Residual Norm	
	F-test	GCV	F-test	GCV
RP-1	1.84×10^{-5}	6.20×10^{-3}	1.05×10^{-2}	1.11×10^{-2}
RP-2	3.51×10^{-5}	1.70×10^{-3}	2.73×10^{-3}	4.91×10^{-3}
RP-3	8.22×10^{-2}	8.22×10^{-2}	1.32×10^{-2}	1.32×10^{-2}
RP-4	7.68×10^{-3}	7.68×10^{-3}	8.85×10^{-3}	8.85×10^{-3}

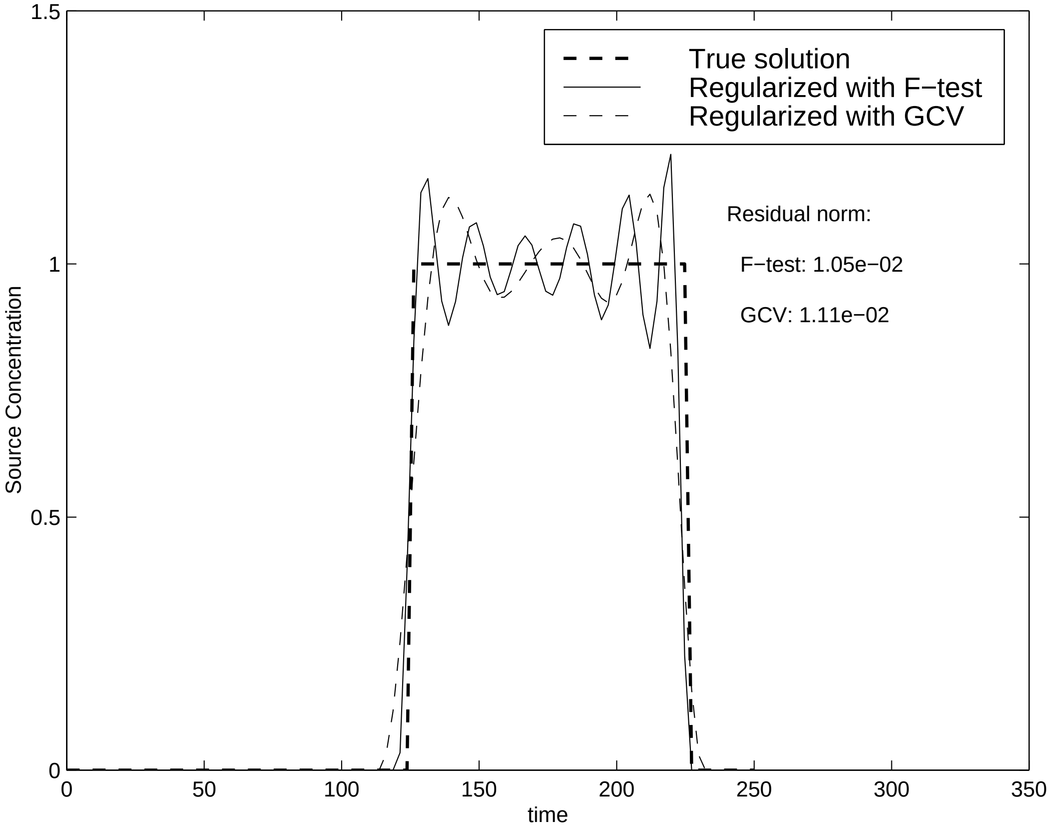

Figure 4.4: Tikhonov regularization results for regularization parameter selection Run RP-1. Parameters are: square input function and $\epsilon = 0.0$.

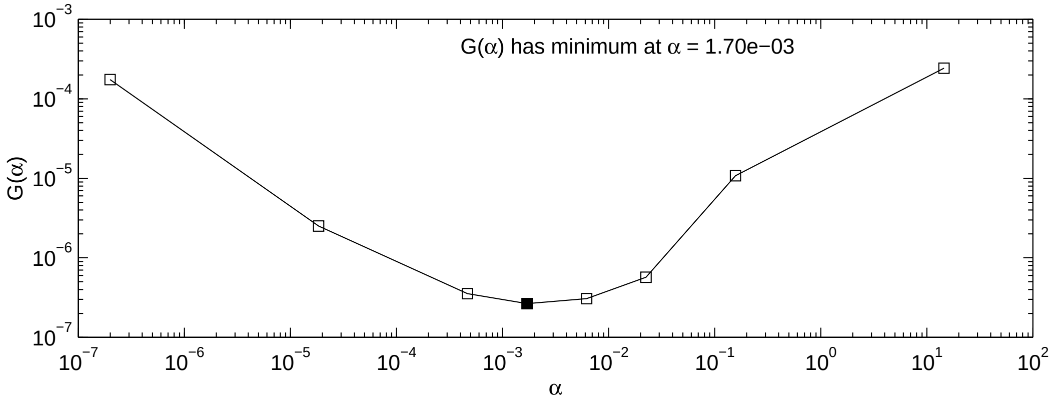

Figure 4.5: Results of generalized cross-validation for regularization parameter selection Run RP-2. The filled square corresponds to the optimal value of α.

selected α. The residual norm from the F-test method is lower than that from the GCV method; therefore, for this scenario, the F-test method of selecting the regularization parameter performs better.

Run RP-3 used inexact data and a square input function. The optimal value of α using the F-test method from CONTIN was $\alpha = 8.22 \times 10^{-2}$, and $P(\alpha) = 0.356$. The results of the generalized cross-validation method are shown in Figure 4.7, with the optimal value of $\alpha = 8.22 \times 10^{-2}$ also. Both methods selected the same value as the optimal value of α; therefore, the results of the simulations are equivalent (see Figure 4.8 and Table 4.2), and both methods of selecting the regularization parameter perform equally well.

Run RP-4 used inexact data and a smooth input function. The optimal value of α using the F-test method from CONTIN was $\alpha = 7.68 \times 10^{-3}$, and $P(\alpha) = 0.728$. The results of the generalized cross-validation method are shown in Figure 4.7, also with an optimal value of $\alpha = 7.68 \times 10^{-3}$. Both methods selected the same value as the optimal value of α; therefore, the results

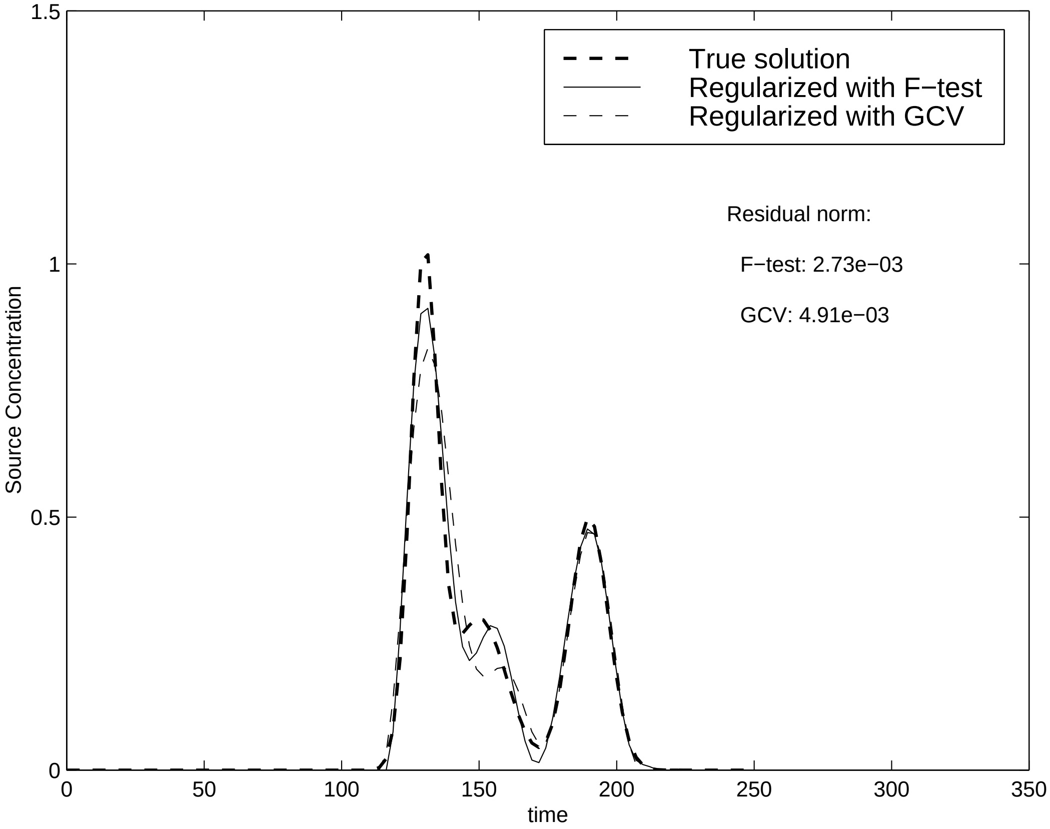

Figure 4.6: Tikhonov regularization results for regularization parameter selection Run RP-2. Parameters are: smooth input function and $\epsilon = 0.0$.

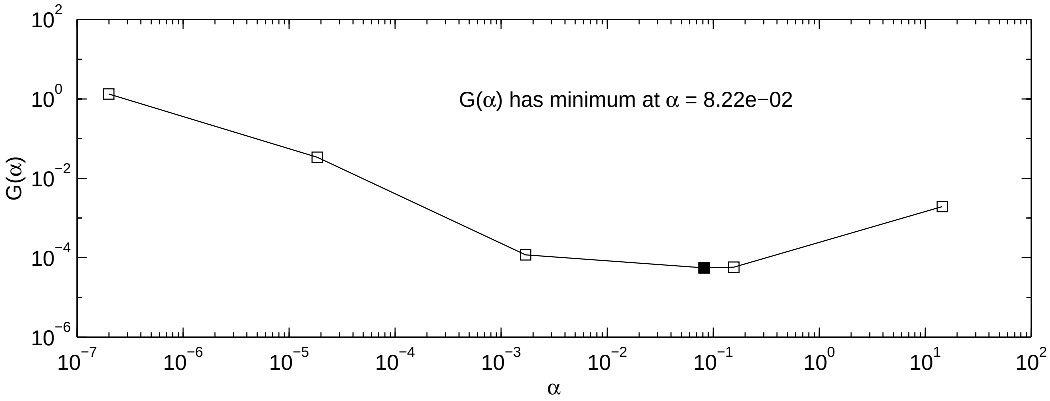

Figure 4.7: Results of generalized cross-validation for regularization parameter selection Run RP-3. The filled square corresponds to the optimal value of α.

of the simulations are equivalent (see Figure 4.10 and Table 4.2), and both methods of selecting the regularization parameter perform equally well.

These results show that the value of the regularization parameter does affect the solution to the inverse problem, and that the chosen value depends on the method of selection that was used. These results also show that the values of α chosen by the F-test and GCV methods result in similar solutions to the inverse problem. Therefore, either method would be a satisfactory choice in practice.

4.1.2 Evaluation of the Order of Regularization

The order of regularization controls some properties of the regularized solution. With zero-order regularization, the norm of the model is minimized; with first-order regularization, the first derivative of the model is minimized; and with second order regularization, the second derivative of the model is minimized, or, in other words, the smoothness of the model is maximized.

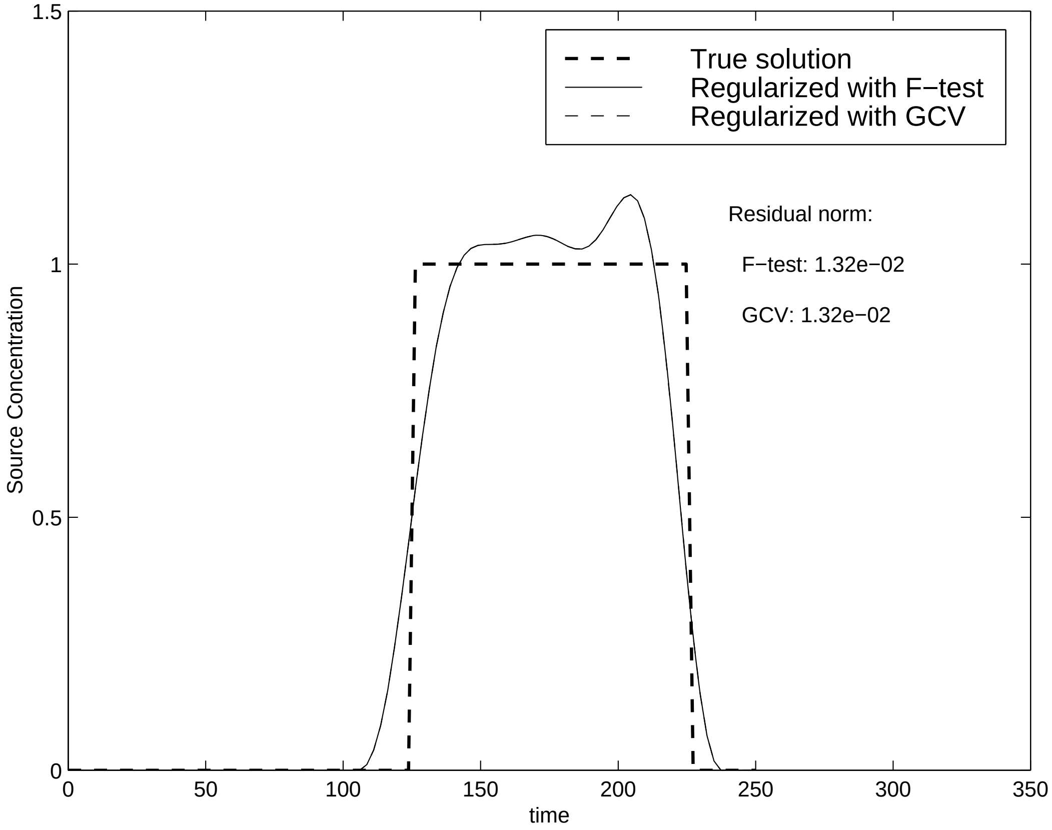

Figure 4.8: Tikhonov regularization results for regularization parameter selection Run RP-3. The two regularized solution are equivalent. Parameters are: square input function and $\epsilon = 0.05$.

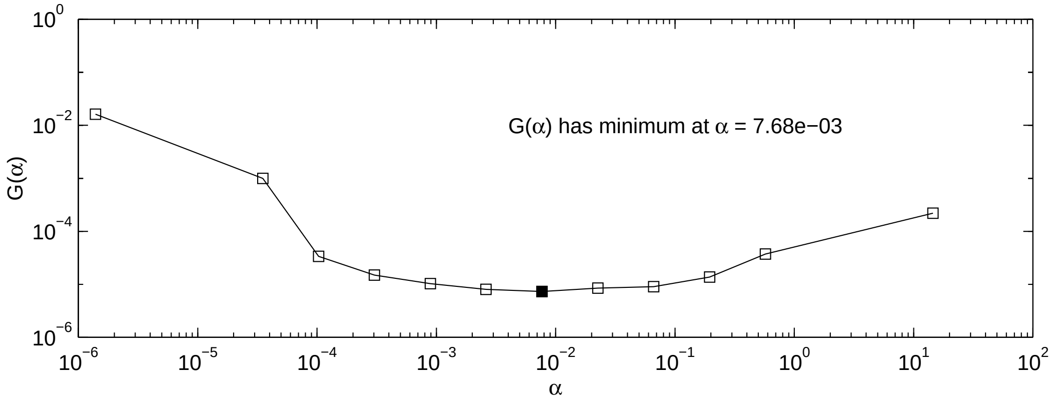

Figure 4.9: Results of generalized cross-validation for regularization parameter selection Run RP-4. The filled square corresponds to the optimal value of α.

Table 4.3: Test scenarios for the analysis of the regularization order.

Run Number	True Source History Function	Noise Level (ϵ)
RO-1	Square	0
RO-2	Smooth	0
RO-3	Square	0.05
RO-4	Smooth	0.05

To analyze the effects of the regularization order on the regularized solution, we tested four different scenarios, shown in Table 4.3, using zero–, first– and second–order regularization. The parameters used in the runs described in Section 4.1.1 were also used here. The regularization parameter was selected using the F-test method. The results were evaluated by comparing the L_2 norm of the differences between the true input function and the regularized solution, normalized by the total number of discretization intervals in the source history function (100).

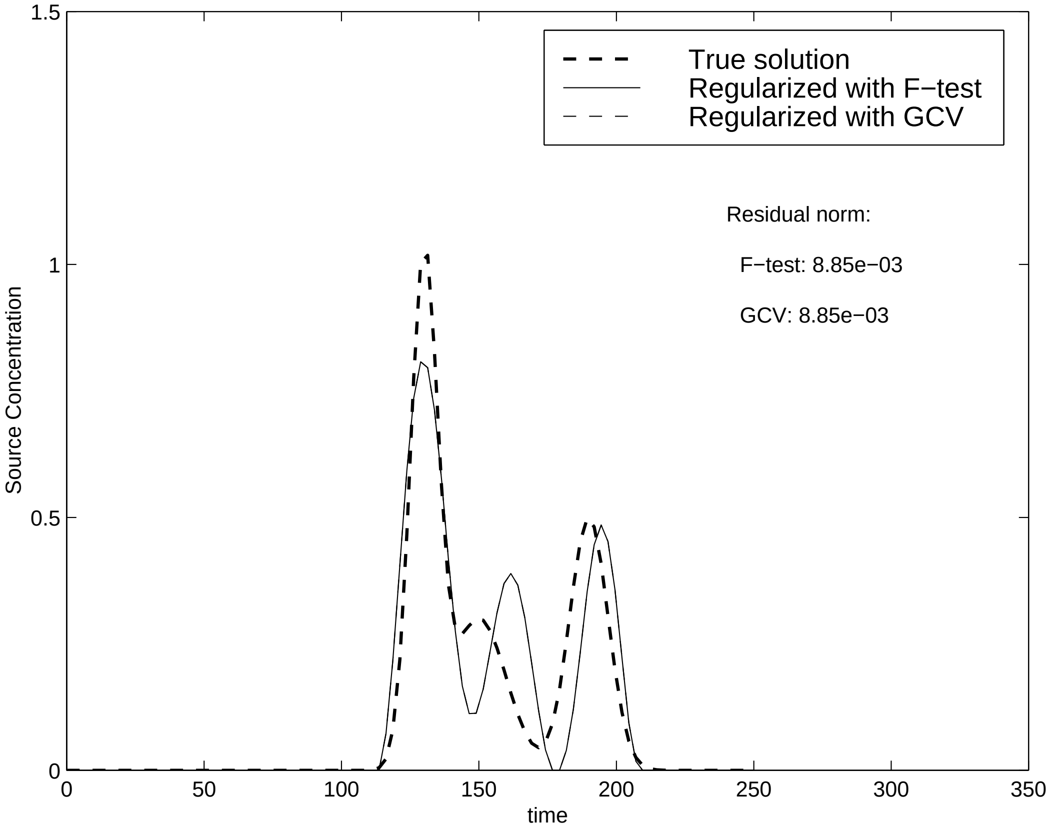

Figure 4.10: Tikhonov regularization results for regularization parameter selection Run RP-4. The two regularized solution are equivalent. Parameters are: smooth input function and $\epsilon = 0.05$.

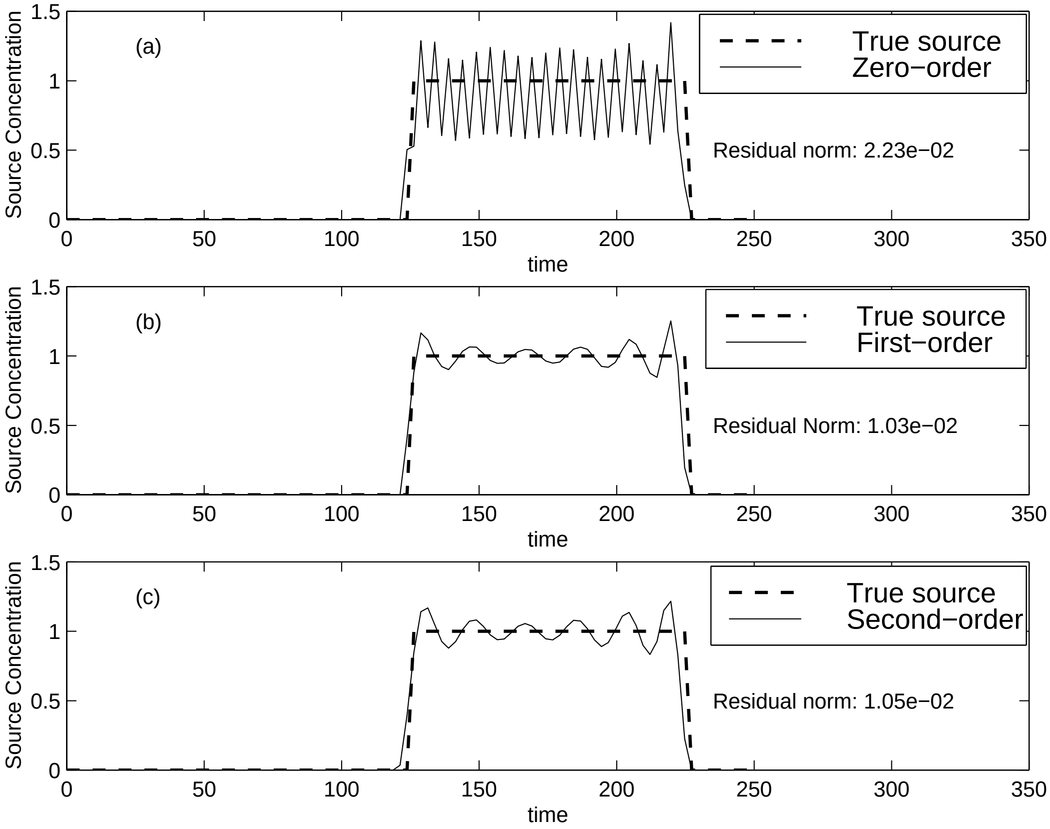

Figure 4.11: Tikhonov regularization results for regularization order Run RO-1. (a) Zero–order regularization. (b) First-order regularization. (c) Second-order regularization. Parameters are: square input function and $\epsilon = 0.0$.

Run RO-1 used perfect data and a square input function. The results are shown in Figure 4.11 and Table 4.4. The results of first– and second–order regularization are nearly indistinguishable, and much smoother than the regularized solution from zero–order regularization. Obviously, first– or second–order regularization produces a more accurate result than zero–order regularization.

Run RO-2 used perfect data and a smooth input function. The results

Table 4.4: Results of the analysis of the regularization order.

Run Number	Residual Norm		
	Zero–order	First–order	Second–order
RO-1	2.23×10^{-2}	1.03×10^{-2}	1.05×10^{-2}
RO-2	8.37×10^{-3}	3.13×10^{-3}	2.73×10^{-3}
RO-3	2.37×10^{-2}	1.38×10^{-2}	1.32×10^{-2}
RO-4	1.23×10^{-2}	8.84×10^{-3}	8.85×10^{-3}

are shown in Figure 4.12 and Table 4.4. Again, the solutions from first– and second–order regularization are much smoother and more accurate than the solution from zero-order regularization. Although the first–order and second–order results appear similar, the residual norm is slightly lower when second–order regularization is used.

Run RO-3 used inexact data and a square input function. The results are shown in Figure 4.13 and Table 4.4. The solutions from first– and second–order regularization are nearly indistinguishable; and they are much smoother and more accurate than the solution from zero-order regularization.

Run RO-4 used inexact data and a smooth input function. The results are shown in Figure 4.14 and Table 4.4. Again, the solutions from first– and second–order regularization are much smoother and more accurate than the solution from zero-order regularization.

These results show that the regularization order does affect the solution to the inverse problem. Zero–order regularization does not produce an accurate solution for this problem; while, first–order and second–order regularizations produce smoother and more accurate results. In Run RO-2, second–order

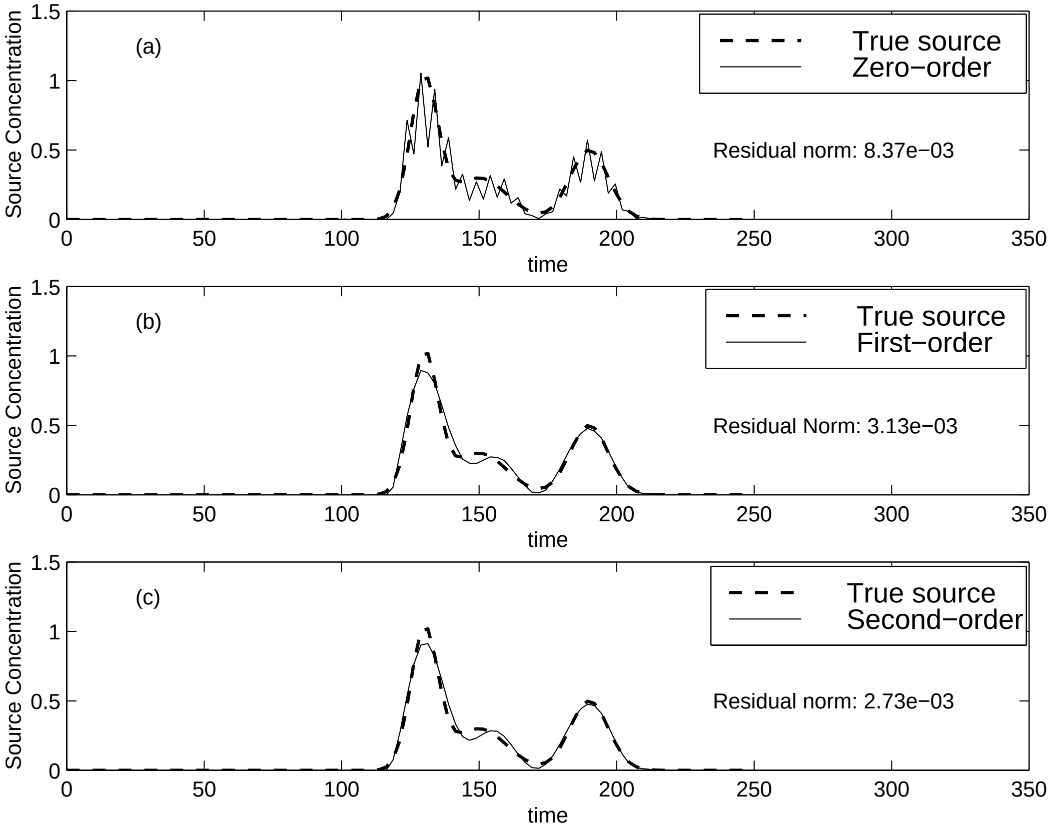

Figure 4.12: Tikhonov regularization results for regularization order Run RO-2. (a) Zero–order regularization. (b) First-order regularization. (c) Second-order regularization. Parameters are: smooth input function and $\epsilon = 0.0$.

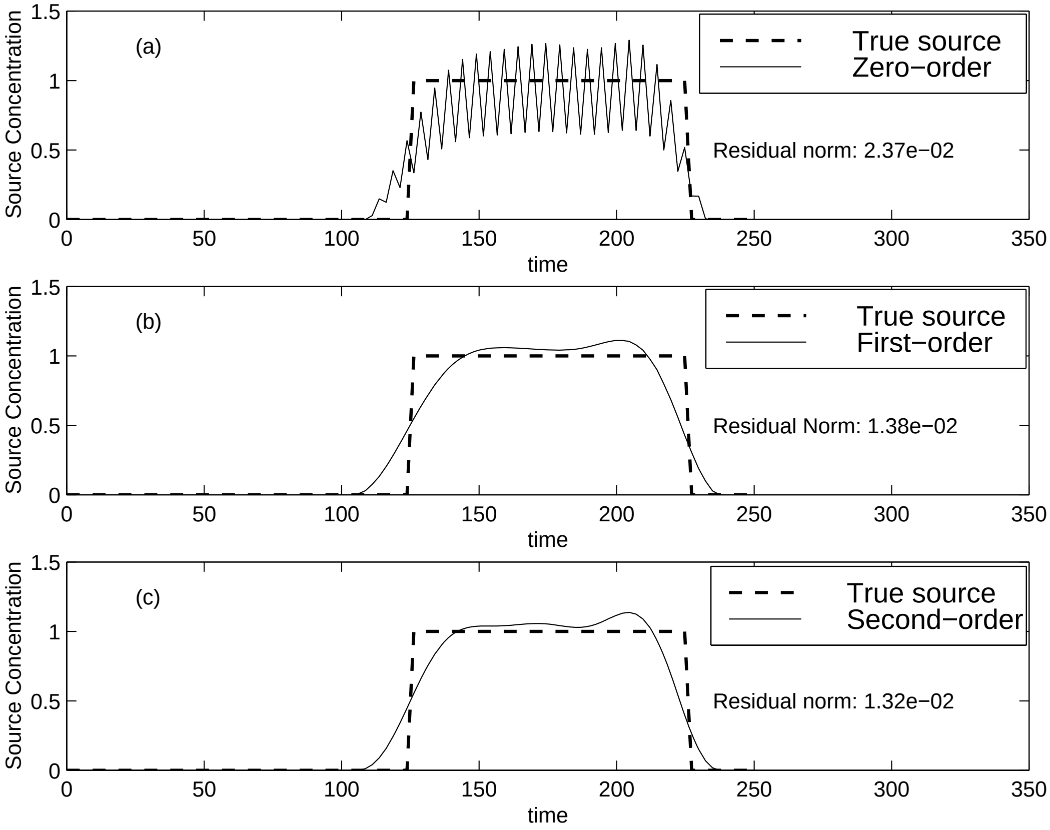

Figure 4.13: Tikhonov regularization results for regularization order Run RO-3. (a) Zero–order regularization. (b) First-order regularization. (c) Second-order regularization. Parameters are: square input function and $\epsilon = 0.05$.

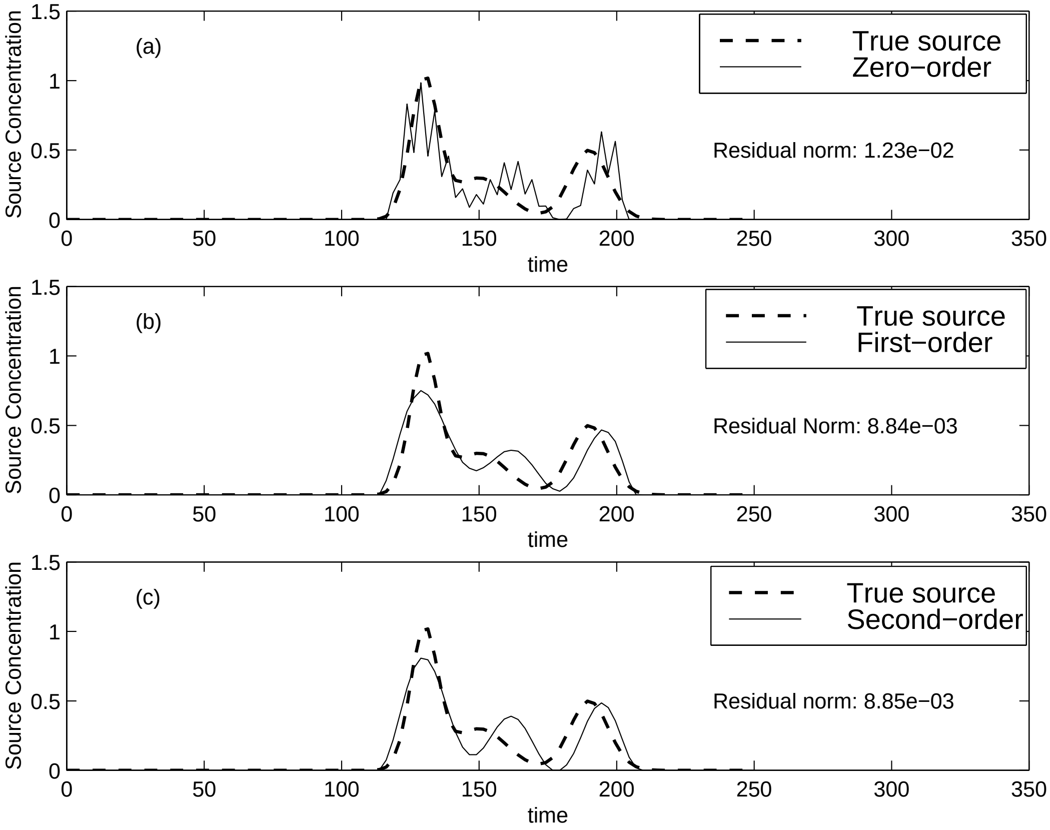

Figure 4.14: Tikhonov regularization results for regularization order Run RO-4. (a) Zero–order regularization. (b) First-order regularization. (c) Second-order regularization. Parameters are: smooth input function and $\epsilon = 0.05$.

regularization produces a slightly more accurate result; while the differences are negligible in all other runs. Therefore, either first–order or second–order regularization would be satisfactory.

4.2 Evaluation of Parameter Selection for Minimum Relative Entropy Inversion

Solving the source history reconstruction problem using minimum relative entropy inversion requires the user to specify lower and upper bounds and an expected value for each solution parameter. These parameter choices can be based on prior knowledge of the system; however, they are subjective. In this section, we evaluate the effects of these parameters on the inverse problem solution.

4.2.1 Evaluation of the Expected Value Function

We analyzed the sensitivity of the reconstructed source history to the prior expected value functions for the four different scenarios shown in Table 4.5. For each of these runs, we used three different expected values functions:

- Gaussian (shown in Equation 3.20),
- Uniform ($s_i = 0.8$),
- Exponential (shown in Equation 3.19).

The parameters that were used in the runs described in Section 4.1 were also used here, except that for these runs, $t_1 = 0$ instead of $t_1 = 0.01$. The

Table 4.5: Test scenarios for the analysis of the prior expected value functions.

Run Number	True Source History Function	Noise Level (ϵ)
EV-1	Square	0
EV-2	Smooth	0
EV-3	Square	0.05
EV-4	Smooth	0.05

Table 4.6: Results of the analysis of the prior expected value functions.

	Residual Norm		
Run Number	Gaussian	Uniform	Exponential
EV-1	8.60×10^{-3}	9.19×10^{-3}	9.59×10^{-3}
EV-2	4.13×10^{-3}	4.58×10^{-3}	4.69×10^{-3}
EV-3	1.35×10^{-2}	1.33×10^{-2}	1.39×10^{-2}
EV-4	6.21×10^{-3}	8.34×10^{-3}	8.74×10^{-3}

upper bound was constant at 1.1; and the lower bound was constant at 0.0. The stopping tolerance for the $\boldsymbol{\lambda}$ iterations was 10^{-2}. The results were evaluated by comparing the L_2 norm (normalized by the total number of discretization intervals in the source history function) of the differences between the true input function and the inverse solution.

Run EV-1 used perfect data and a square input function. The results are shown in Figure 4.15 and Table 4.6. The three solutions are similar; they match the vertical sections fairly well, and have a slight oscillation in the horizontal section. The residual norms are approximately equal.

Run EV-2 used perfect data and a smooth input function. The results are shown in Figure 4.16 and Table 4.6. The three solutions are nearly indis-

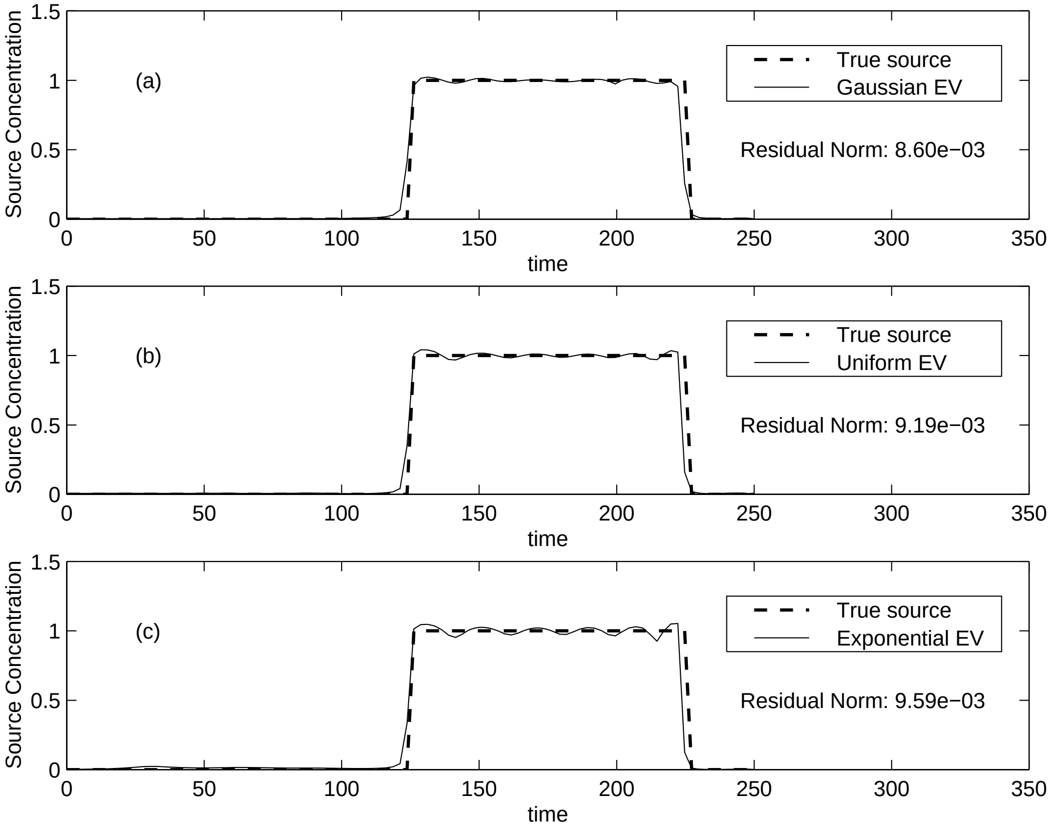

Figure 4.15: MRE results for the expected value function analysis Run EV-1. (a) Results with the gaussian expected value function. (b) Results with the uniform expected value function. (c) Results with the exponential expected value function. Parameters are: square input function and $\epsilon = 0.0$.

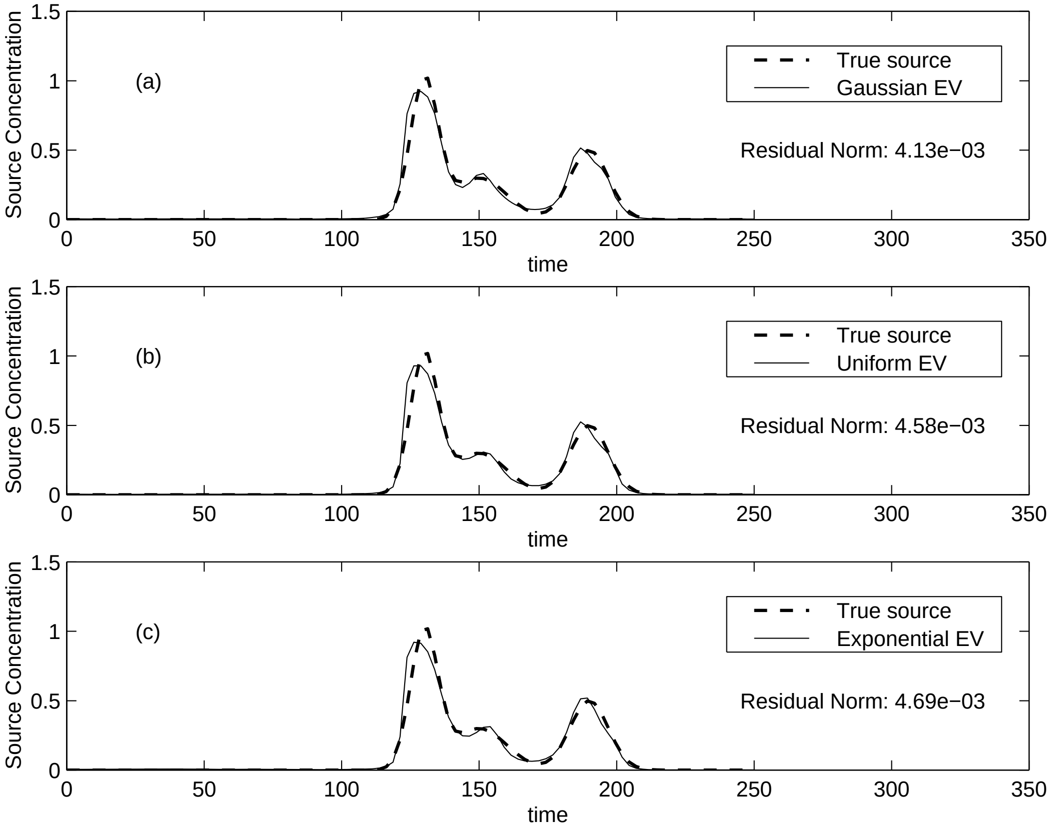

Figure 4.16: MRE results for the expected value function analysis Run EV-2. (a) Results with the gaussian expected value function. (b) Results with the uniform expected value function. (c) Results with the exponential expected value function. Parameters are: smooth input function and $\epsilon = 0.0$.

tinguishable, matching the timing of the three peaks well but showing slightly less accuracy in matching the magnitudes of the peaks. The three solutions differ slightly near the middle peak, and these differences are reflected in the values of the residual norms.

Run EV-3 used inexact data and a square input function. The results are shown in Figure 4.17 and Table 4.6. The three solutions are similar; they capture the general shape of the true input function, but the finer details are

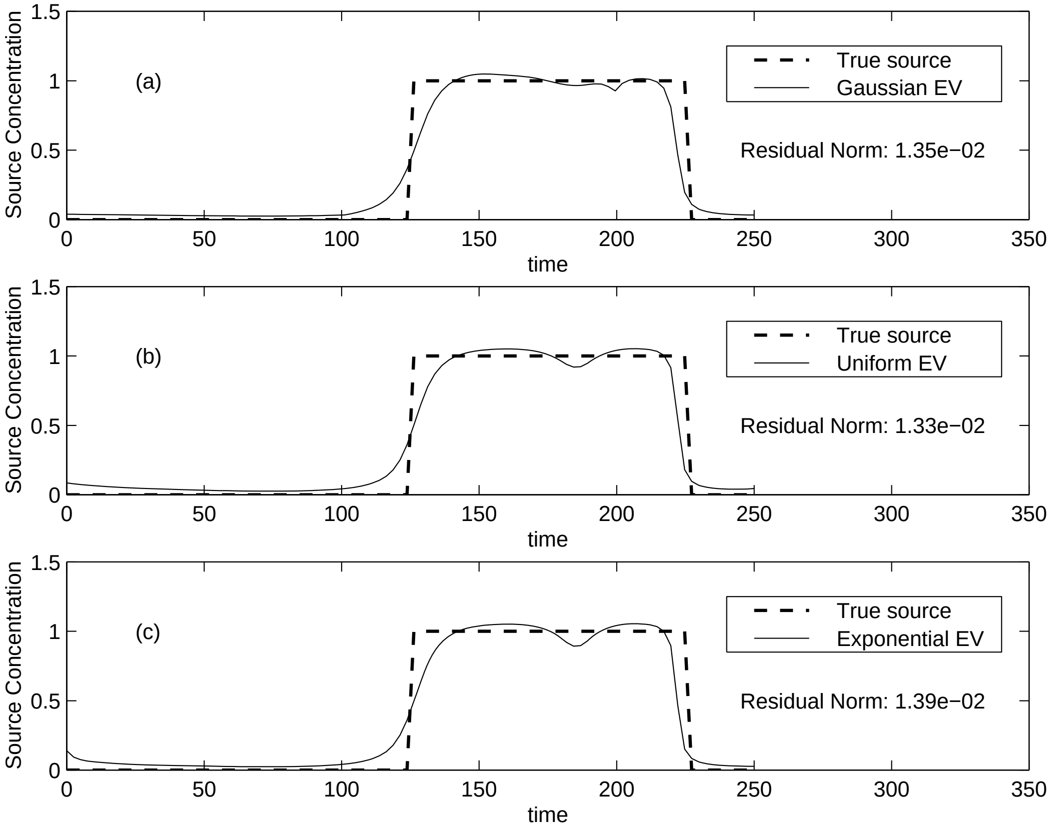

Figure 4.17: MRE results for the expected value function analysis Run EV-3. (a) Results with the gaussian expected value function. (b) Results with the uniform expected value function. (c) Results with the exponential expected value function. Parameters are: square input function and $\epsilon = 0.05$.

not reconstructed. The residual norms for the three cases are approximately equal.

Run EV-4 used inexact data and a smooth input function. The results are shown in Figure 4.18 and Table 4.6. The three solutions are similar near the first and third peaks. Near the middle peak, the solutions from the uniform and exponential expected value functions are nearly identical, and show the opposite behavior as the true function. This is likely due to the slightly over-

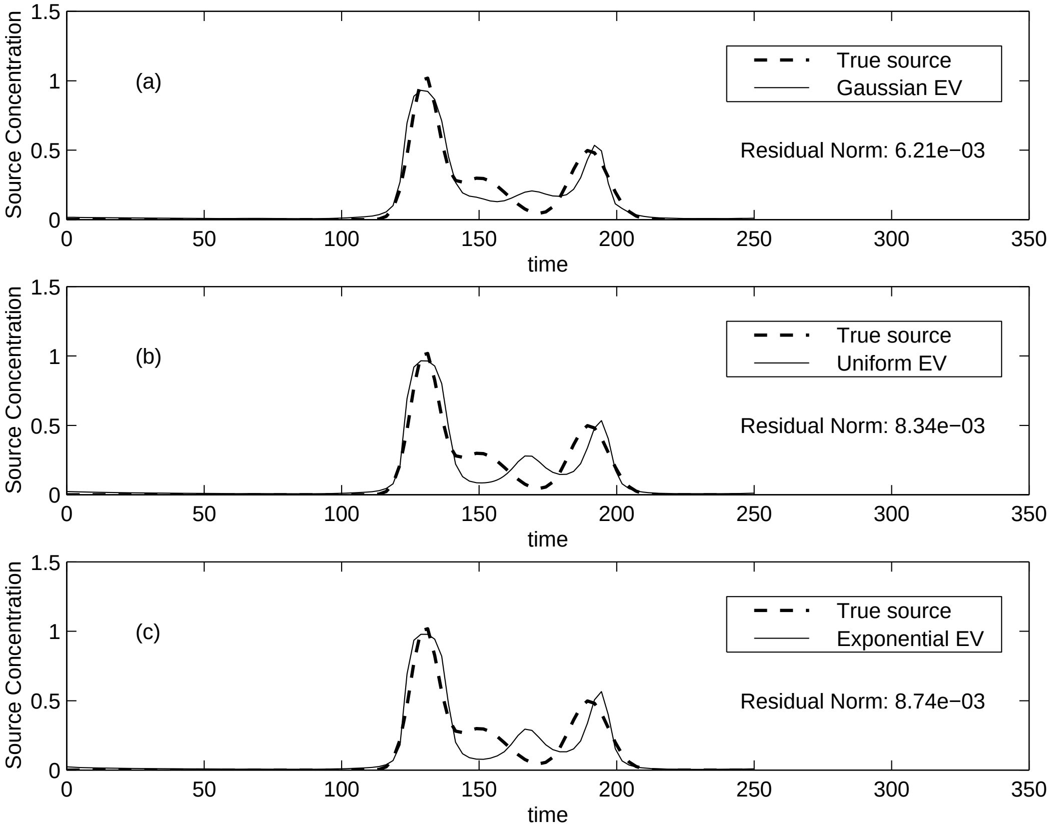

Figure 4.18: MRE results for the expected value function analysis Run EV-4. (a) Results with the gaussian expected value function. (b) Results with the uniform expected value function. (c) Results with the exponential expected value function. Parameters are: smooth input function and $\epsilon = 0.05$.

estimated concentration sampled at $x = 140$, and the underestimated sampled concentrations at $x = 120, 130, 150$, and 160 (See Figure 4.2). Near the middle peak, the solution from the Gaussian expected value function shows similar pattern as the other reconstructed solutions; however, the fluctuation is less pronounced.

The results show that the solutions are relatively insensitive to the prior expected value functions. The solutions were almost indistinguishable in

Table 4.7: Test scenarios for the analysis of the upper bounds.

Run Number	True Source History Function	Noise Level (ϵ)
UB-1	Square	0
UB-2	Smooth	0
UB-3	Square	0.05
UB-4	Smooth	0.05

all but one case (EV-4).

4.2.2 Evaluation of the Upper Bound

We analyzed the sensitivity of the reconstructed source history to the upper bound parameter for the four different scenarios shown in Table 4.7. For each of these runs, we used a constant upper bound (UB) of 1.02, 1.5, and 2.0. Since the upper bound is the upper limit on the value of the reconstructed source history function, the upper bound must be greater than or equal to the true source history. If the value selected for the upper bound is too low, the data constraint cannot be satisfied. The expected value function was Gaussian (Equation 3.20), which has a maximum value of 1.0; therefore the value of the upper bound must be greater than 1.0. For the remaining parameters, we used the same values as for Runs EV-1—EV-4.

Run UB-1 used perfect data and a square input function. The results are shown in Figure 4.19 and Table 4.8. The results get progressively worse as the upper bound increases.

Run UB-2 used perfect data and a smooth input function. The results

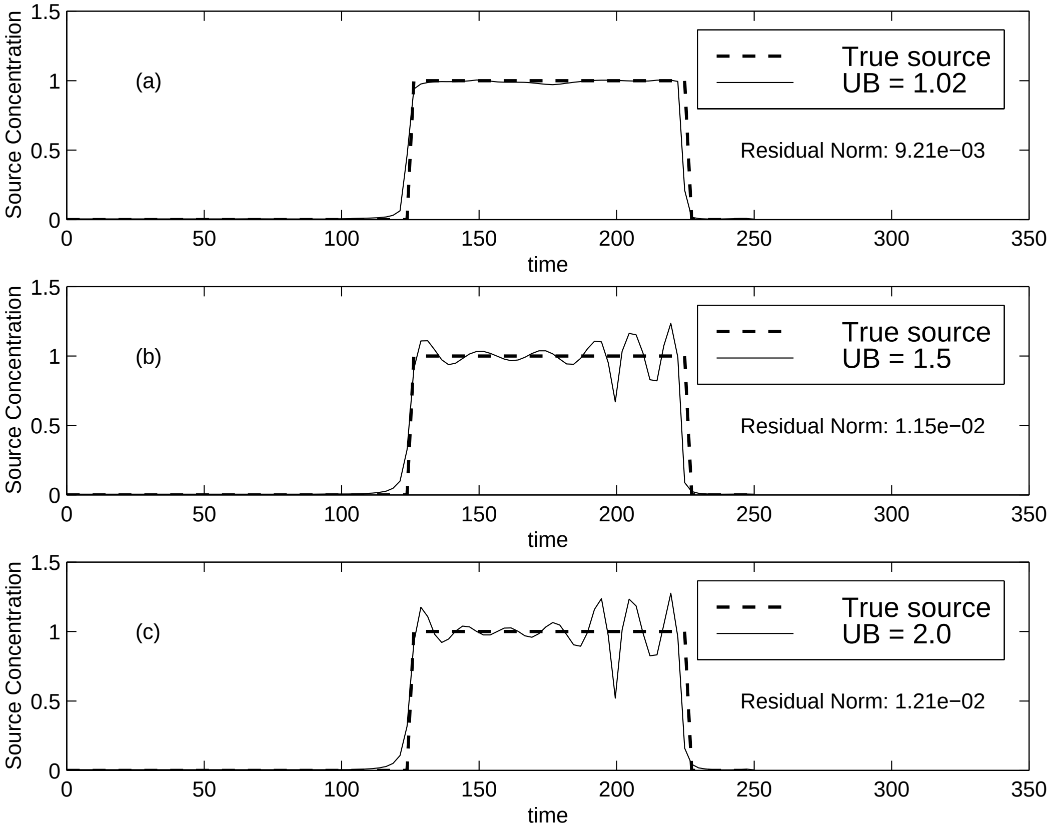

Figure 4.19: MRE results for the upper bound analysis Run UB-1. (a) Results with upper bound UB = 1.02. (b) Results with upper bound UB = 1.5. (c) Results with upper bound UB = 2.0. Parameters are: square input function and $\epsilon = 0.0$.

Table 4.8: Results of the analysis of the upper bounds.

Run Number	Residual Norm		
	UB = 1.02	UB = 1.5	UB = 2.0
UB-1	9.21×10^{-3}	1.15×10^{-2}	1.21×10^{-2}
UB-2	8.00×10^{-3}	4.37×10^{-3}	4.55×10^{-3}
UB-3	1.13×10^{-2}	1.80×10^{-2}	2.19×10^{-2}
UB-4	1.10×10^{-2}	8.96×10^{-3}	1.06×10^{-2}

are shown in Figure 4.20 and Table 4.8. For UB = 1.5 or UB = 2.0, the results are similar and agree well with the true source history. For UB = 1.02, $s_i \approx U_i$ at $t_i = 150$. As discussed in Section 3.2.1, if $s_i = U_i$, the probability density function, $q(m_i)$, is a Dirac delta function and the solution is always equal to the upper bound. In this case, the probability density function, $q(m_i)$, is approximately equal to a Dirac delta function; therefore the solution will always be near the upper bound, and the true solution cannot be attained when UB = 1.02. This problem was not encountered in Run UB-1 because the true solution at $t_i = 150$ was very close to the upper bound. Note that with UB = 1.1 (Run EV-2, Figure 4.16a), the residual norm was 4.13×10^{-3}. This, along with the results in Figure 4.20, indicates that the solution is better for lower values of the upper bound, provided that the upper bound is sufficiently greater than the expected value.

Run UB-3 used inexact data and a square input function. The results, shown in Figure 4.21 and Table 4.8, show that the solution gets progressively worse as the upper bound increases.

Run UB-4 used inexact data and a smooth input function. The results are shown in Figure 4.22 and Table 4.8. Similar to the results of UB-2, the solution is poor when UB = 1.02 because the prior expected value, s_i, and the upper bound, U_i, are approximately equal at $t = 150$. For UB = 1.5 and UB = 2.0, the results match the first and third peaks, but do not match the timing of the second peak. This is likely due to the measured concentration being overestimated at $x = 140$ and underestimated at $x = 120, 130, 150,$ and 160. Recall that with UB=1.1 (Run EV-4, Figure 4.18a), the residual norm was 6.21×10^{-3}, which is lower than the residual norms for any of these runs. This

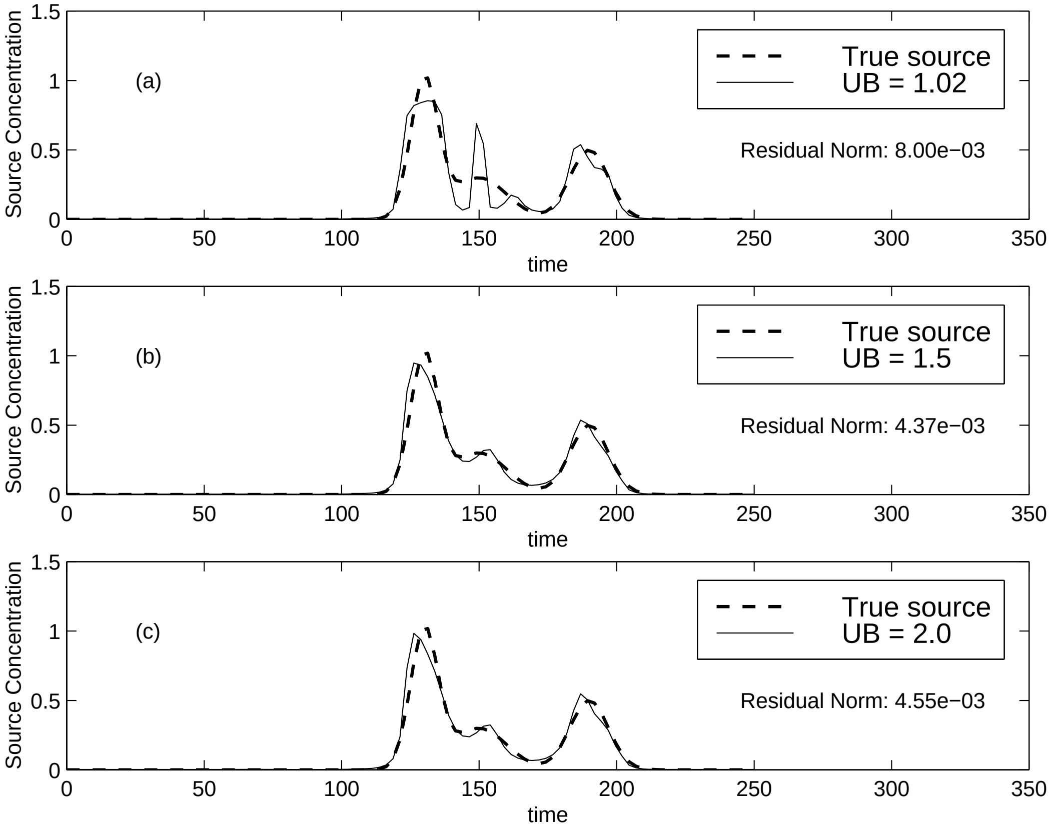

Figure 4.20: MRE results for the upper bound analysis Run UB-2. (a) Results with upper bound UB = 1.02. (b) Results with upper bound UB = 1.5. (c) Results with upper bound UB = 2.0. Parameters are: smooth input function and $\epsilon = 0.0$.

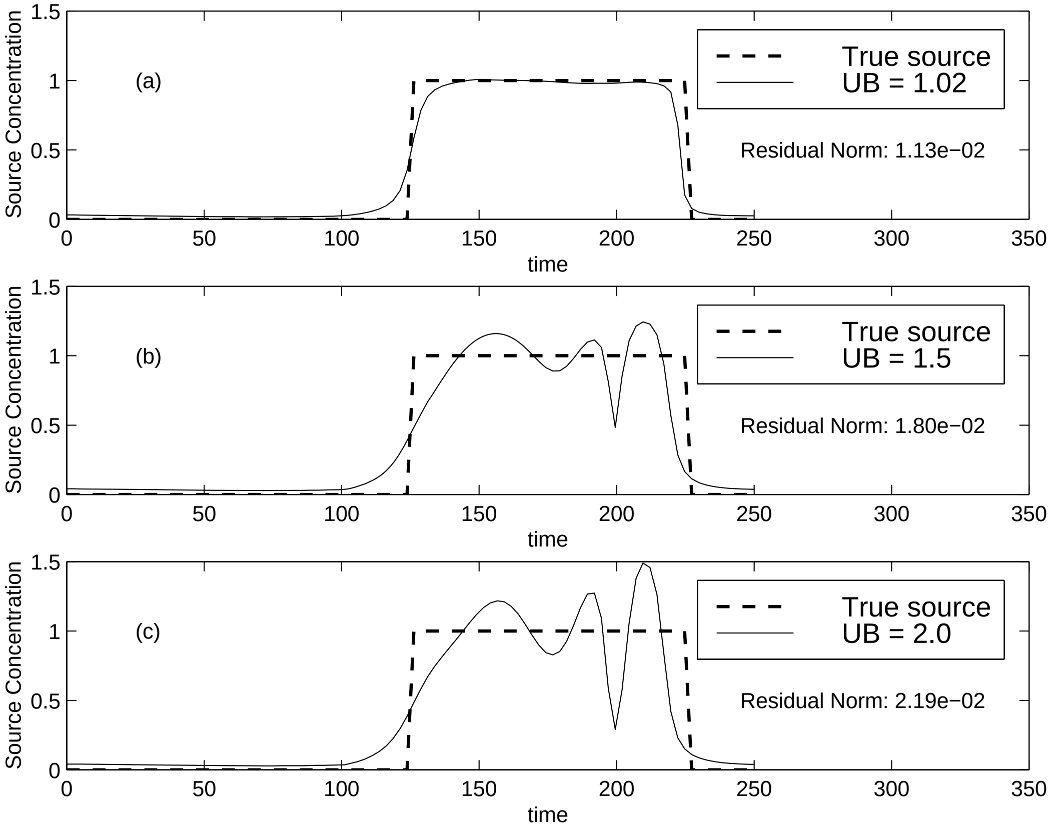

Figure 4.21: MRE results for the upper bound analysis Run UB-3. (a) Results with upper bound UB = 1.02. (b) Results with upper bound UB = 1.5. (c) Results with upper bound UB = 2.0. Parameters are: square input function and $\epsilon = 0.05$.

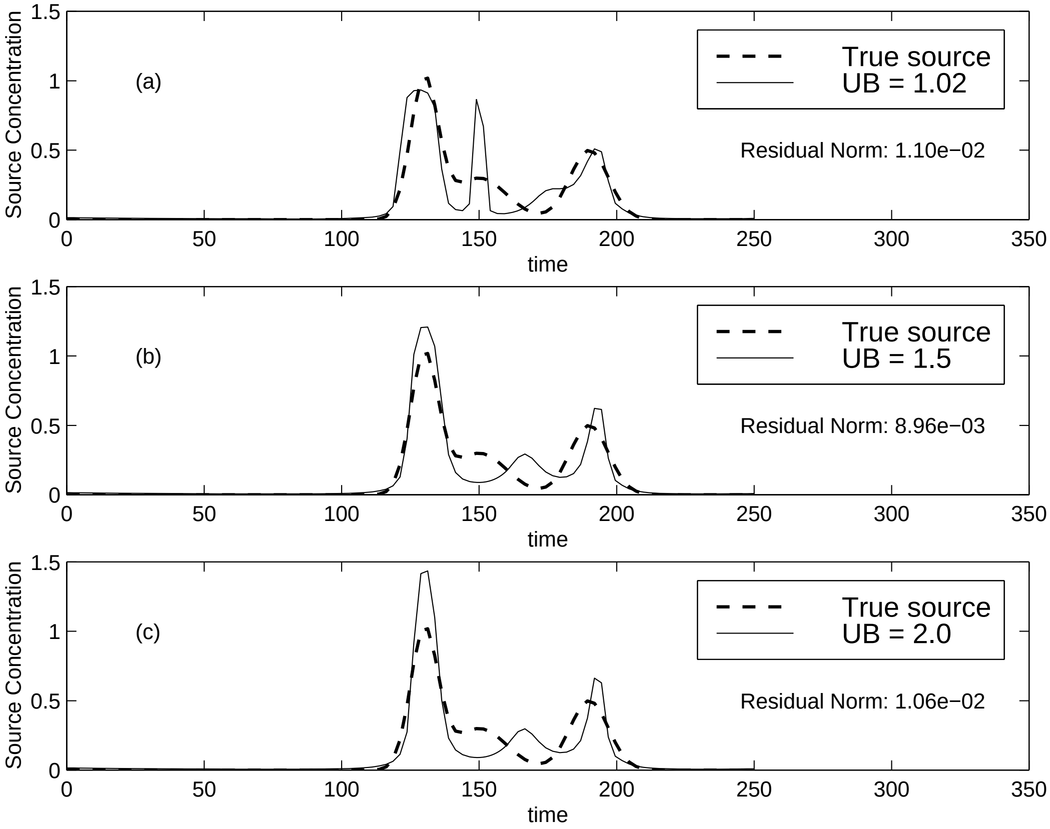

Figure 4.22: MRE results for the upper bound analysis Run UB-4. (a) Results with upper bound UB = 1.02. (b) Results with upper bound UB = 1.5. (c) Results with upper bound UB = 2.0. Parameters are: smooth input function and $\epsilon = 0.05$.

indicates that the solution is more accurate when the upper bound is closer to the true maximum value; however, if the upper bound is equal to the expected value function, the accuracy of the solution is diminished due to numerical instabilities.

The results show that the solution is affected by the upper bound. The upper bound must be greater than or equal to the true source history; otherwise the data cannot be matched. The solution is most accurate when the upper bound is near the maximum value of the true source history function, and the accuracy decreases as the upper bound increases away from this value. The accuracy is poor when the upper bound is equal or approximately equal to the expected value function.

Chapter 5

Comparison of Tikhonov Regularization and Minimum Relative Entropy Inversion

In this chapter, we evaluate the results of Tikhonov regularization and minimum relative entropy inversion for the source history reconstruction problem to compare the relative effectiveness of the two methods in handling complications that are intrinsic to or may be encountered in field situations. In an ideal situation, the plume would be sampled completely (i.e. the sampling would capture both the leading and trailing edges of the plume), the sampled concentrations would contain no measurement error, and the transport parameters would be known exactly. In this chapter we evaluate the ability of the two methods to reconstruct the release history for an ideal situation. We also address several non-ideal factors including:

- Measurement error.
- Incomplete spatial sampling of the plume.
- Errors in transport parameter estimates.

We also evaluate the ability of each method to reproduce a smooth source history function and a non-smooth source history function. We use the smooth source history function in the ideal case, and the non-smooth function in the non-ideal case, although this decision was entirely arbitrary.

We examine the results of the two methods for an ideal scenario; then, we run individual simulations addressing each possible non-ideality (including the non-smooth source history function), independent of the others. Since it is possible that the compound effects of multiple non-ideal factors affect the results differently than the individual factors, we also evaluate the results of various combinations of the ideal and non-ideal factors.

5.1 Results of Simulations of the Ideal Scenario

For the ideal scenario, we used a smooth input function (shown in Equation 1.3 and Figure 1.2), and complete sampling of the plume at $T = 300$ ($\mathbf{x} = [0.01, 25, 50, 60, 70, \ldots, 240, 250, 275, 300]$). The transport parameters were exact, with $v = 1.0$ and $D = 1.0$. The reconstructed source history was uniformly discretized into 100 intervals between $t = 0.01$ and $t = 250$.

For Tikhonov regularization, we used second–order regularization, and the F-test method for selecting the regularization parameter ($n_\alpha = 8$). For the MRE method, we used a uniform expected value function ($s_i = 0.8$), and lower and upper bounds of 0.0 and 1.1, respectively. The stopping tolerances were 10^{-4} for the β_i calculations and 10^{-2} for the $\boldsymbol{\lambda}$ calculations (see Section 3.2.1).

The results are shown in Figure 5.1. Both solutions are very close to the true source history for this case. The timing of three peaks is reproduced, and the magnitude is reproduced well except for the first peak.

We also ran an inverse problem for the ideal scenario, with data sampled at $T = 600$. The data were sampled at 21 uniformly-spaced points between

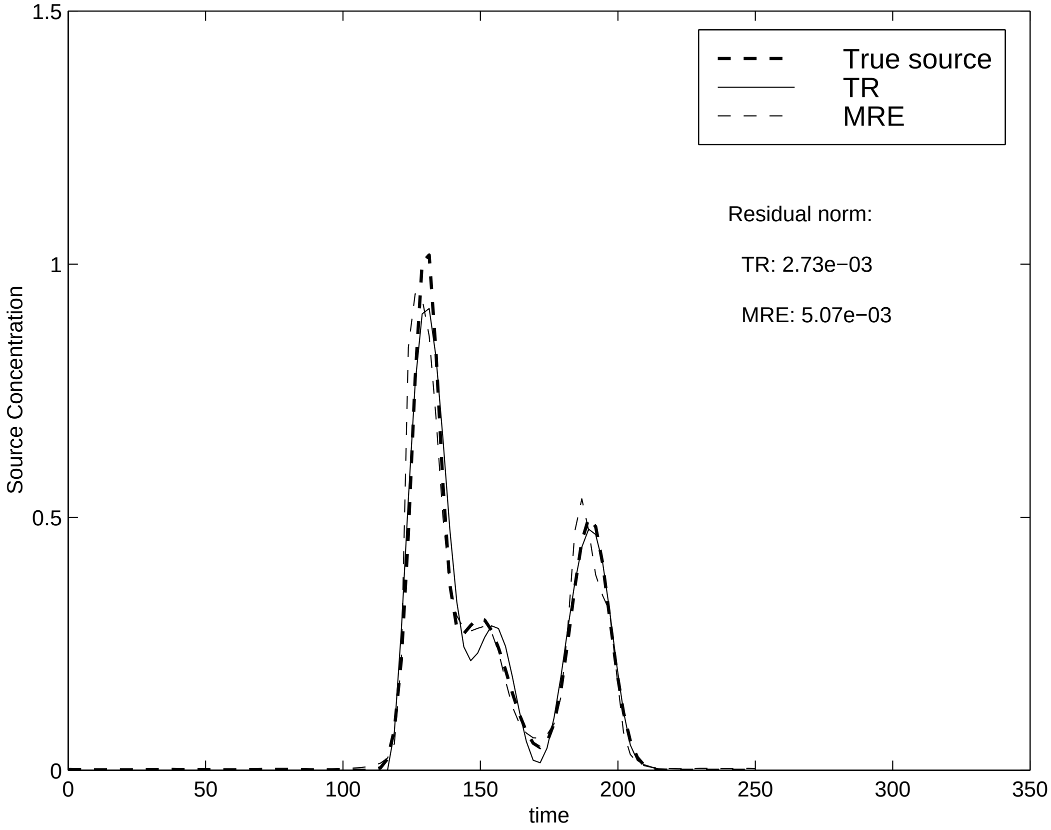

Figure 5.1: Tikhonov regularization and MRE results for the ideal scenario. Parameters are: smooth input function, complete sampling frequency, $\epsilon = 0.0$, $v = 1.0$, $D = 1.0$, and $T = 300$.

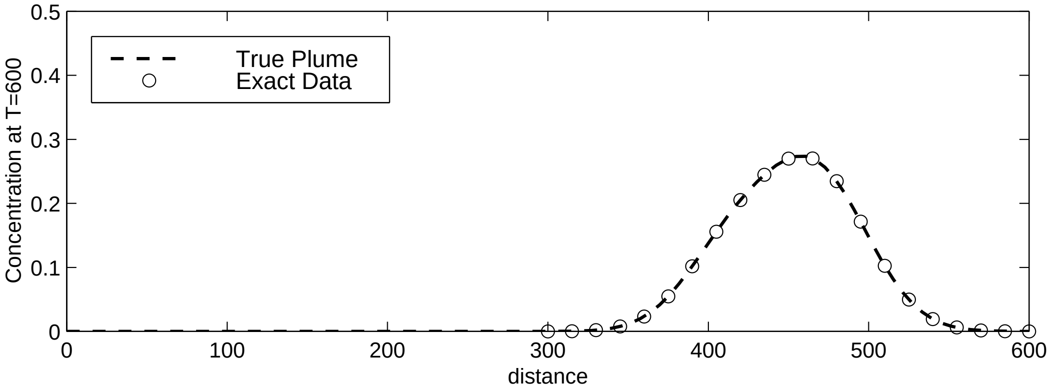

Figure 5.2: True plume at $T = 600$ and sampled data used in the ideal scenario.

$x = 300$ and $x = 600$ (shown in Figure 5.2). At $T = 600$, the plume has dispersed so that the three-peaked structure of the input function is no longer apparent. The reconstructed source history was uniformly discretized into 100 intervals between $t = 0.01$ and $t = 450$. The results are shown in Figure 5.3. The MRE solution matches the timing of the first and third peaks well, but miscalculates the magnitudes of all peaks. Based on the residual norms (calculated for $t < 275$), both methods perform equally well. Both methods produce a non-zero solution for $t > 350$, indicating that the sampled data provide no information about the source history there. With $v = 1.0$ and $D = 1.0$, contamination released from the source after $t \approx 350$ would not reach the first sampling location ($x = 300$) before sampling occurred at $T = 600$. For the MRE results, the uncertainty in the solution for $t > 350$ can be seen in Figure 5.4, which shows the 5^{th} and 95^{th} percentile probability levels. The uncertainty is high for $t > 350$. The uncertainty in the Tikhonov regularization model is based on the model covariance, $\mathbf{\Sigma_m} = \mathbf{G^{\#}\Sigma_d(G^{\#})^T}$, where $\mathbf{\Sigma_d}$ is the covariance matrix of the data and $\mathbf{G^{\#}}$ is the generalized inverse. However, in this problem, we used

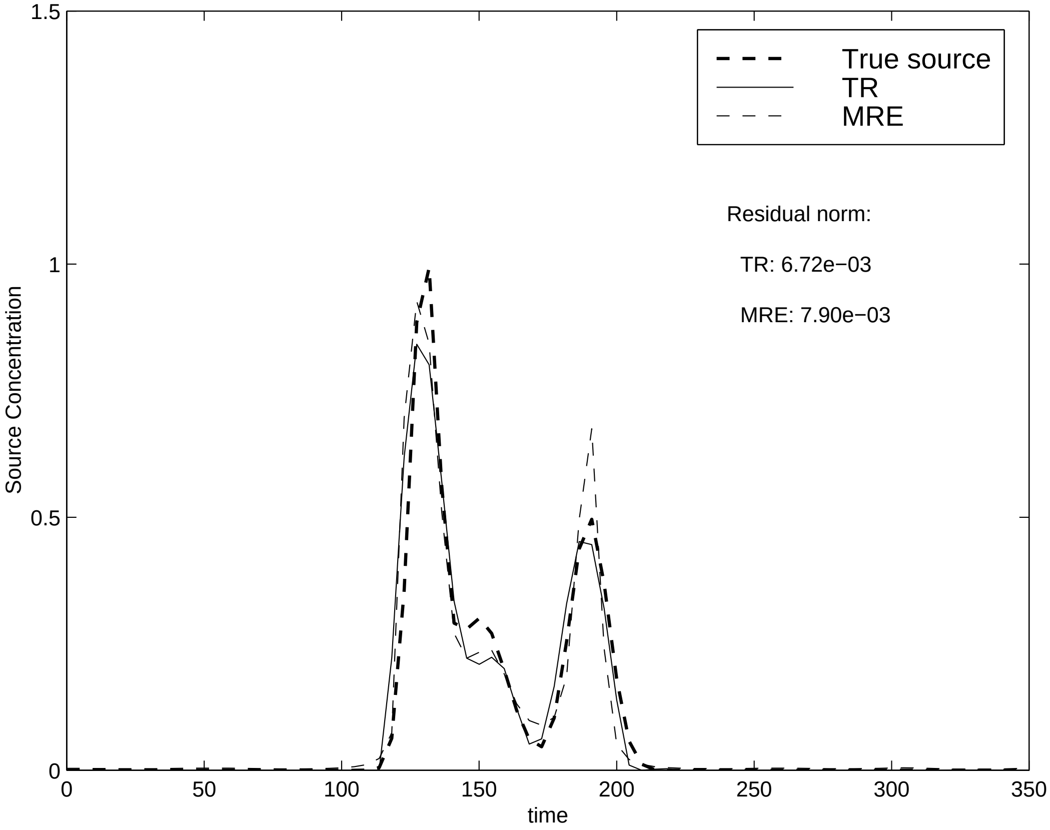

Figure 5.3: Tikhonov regularization and MRE results for the ideal scenario sampled at $T = 600$. Parameters are: smooth input function, complete sampling frequency, $\epsilon = 0.0$, $v = 1.0$, $D = 1.0$, and $T = 600$.

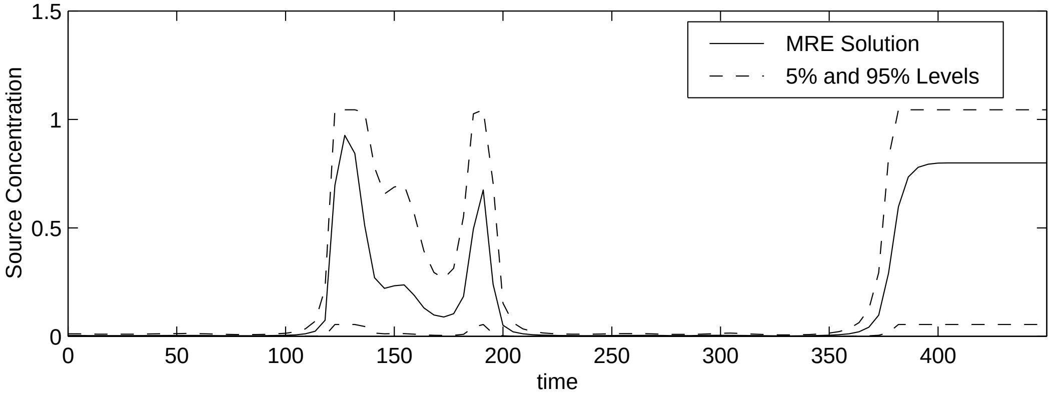

Figure 5.4: Uncertainty in the MRE results for the ideal scenario sampled at $T = 600$.

perfect data, so $\mathbf{\Sigma_d} = \mathbf{\Sigma_m} = \mathbf{0}$.

5.2 Evaluating the Effects of Individual Factors

The ideal scenario was modified to account for each of the non-ideal factors individually. For each of these cases, the inverse problem was solved using the two inverse methods, and the residual norms of the solutions were compared. Except for the non-ideal factor being tested, we used the parameters from the ideal scenario for these runs.

5.2.1 Measurement Error

Sampled data always contains measurement error; therefore, to be effective, inverse methods must be able to handle inexact data. Skaggs and Kabala (1994) used the following error model:

$$C_{\text{meas}}(x_j, T) = C_{\text{exact}}(x_j, T) + \epsilon\delta_j C_{\text{exact}}(x_j, T) \;, \tag{5.1}$$

where $C_{\text{meas}}(x_j, T)$ is the measured concentration at location x_j at time T, x_j is the spatial coordinate of the j^{th} sample, $C_{\text{exact}}(x_j, T)$ is the true concentration at x_j at time T, ϵ is the noise level, and δ_j is the j^{th} independent random deviate (standard normal). We will call this error model E1. Woodbury and Ulrych (1996) used a different error model:

$$C_{\text{meas}}(x_j, T) = C_{\text{exact}}(x_j, T) + \epsilon\delta_j \ . \tag{5.2}$$

We will call this error model E2. In error model E1, the magnitude of the measurement error is proportional to the true concentration; and in error model E2, the magnitude of the error is independent of the true concentration.

To evaluate the ability of the inverse methods to reconstruct the release history using inexact data, we created two sets of measurements with error—one each with E1 and E2. For E1, we used $\epsilon = 0.05$; and for E2, we used $\epsilon = 0.01$. These data are shown in Figure 5.5. Note that with E2, some of the measurements are negative. This is an artifact of the error model since negative measurements are unrealistic in practice.

Tikhonov regularization uses the inexact data directly, and no estimation of the noise level is necessary. With the MRE method, however, an estimate of the noise level must be provided. In practice, the noise level is not known; and therefore an incorrect estimate is likely to be used. To evaluate the effects of incorrectly estimating the noise level, we ran the MRE method with three different noise levels—the exact value, an underestimated value ($\epsilon/2$), and an overestimated value (2ϵ). The results are shown in Figures 5.6 and 5.7, for error models E1 and E2, respectively.

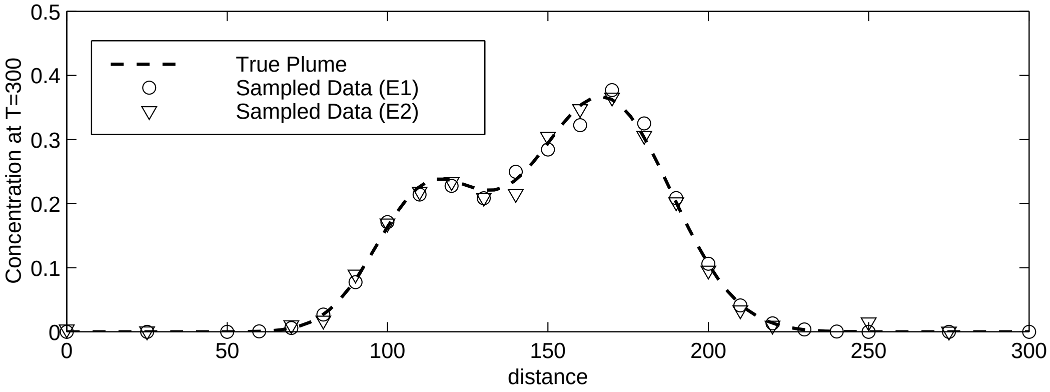

Figure 5.5: True plume at $T = 300$ and sampled data used in the error evaluation runs.

With both error models, the results from Tikhonov regularization are similar (based on the residual norms) to those of the MRE method when the exact noise level or the overestimated noise level is used. However, when the noise level is underestimated, the MRE solution is poor. For the simulations with an underestimated noise level, the stopping tolerance for the $\boldsymbol{\lambda}$ iterations (10^{-2}) in the MRE routine was not met. The solution was converging slowly; so the process was stopped when the maximum number of iterations (27, in this case) was reached. The simulation for E1 stopped when the value of the stopping criteria was 4.05×10^{-2}; and for E2, the simulation was stopped when the value of the stopping criteria was 2.25×10^{-2}. These results show that the particular error model used (Equation 5.1 or Equation 5.2) does not affect the relative results. The results also show that if the data contain measurement error, Tikhonov regularization is more accurate than the MRE method in reconstructing the release history, unless the noise level is known exactly or overestimated. When the noise level was underestimated, the MRE routine converged slowly. Therefore, it is possible that a practitioner could attribute

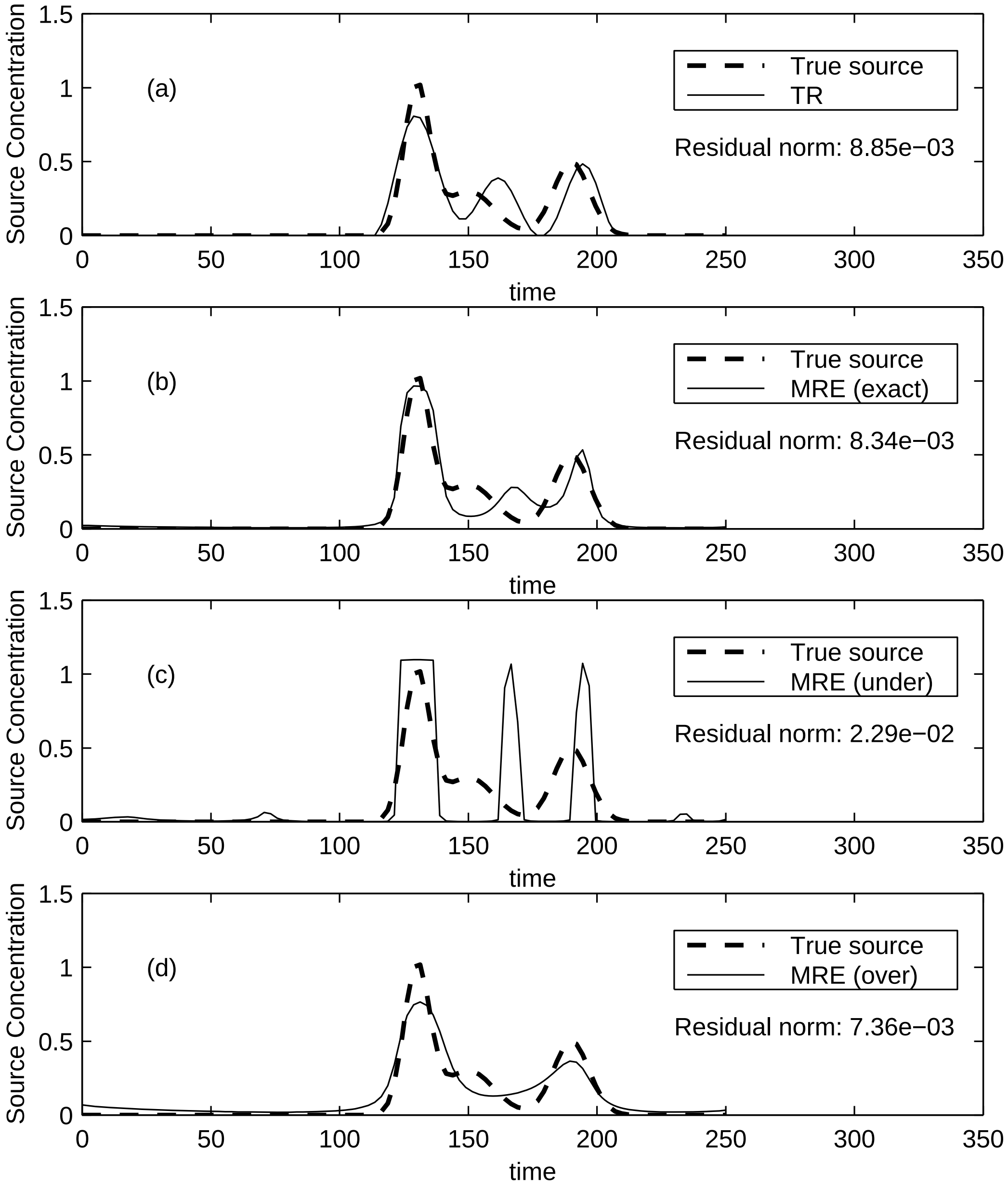

Figure 5.6: Tikhonov regularization and MRE results using inexact data (error model E1). (a) Tikhonov regularization results. (b) MRE results with the exact noise level. (c) MRE results with the underestimated noise level. (d) MRE results with the overestimated noise level. Parameters are: smooth input function, complete sampling frequency, true $\epsilon = 0.05$, $v = 1.0$, $D = 1.0$, and $T = 300$.

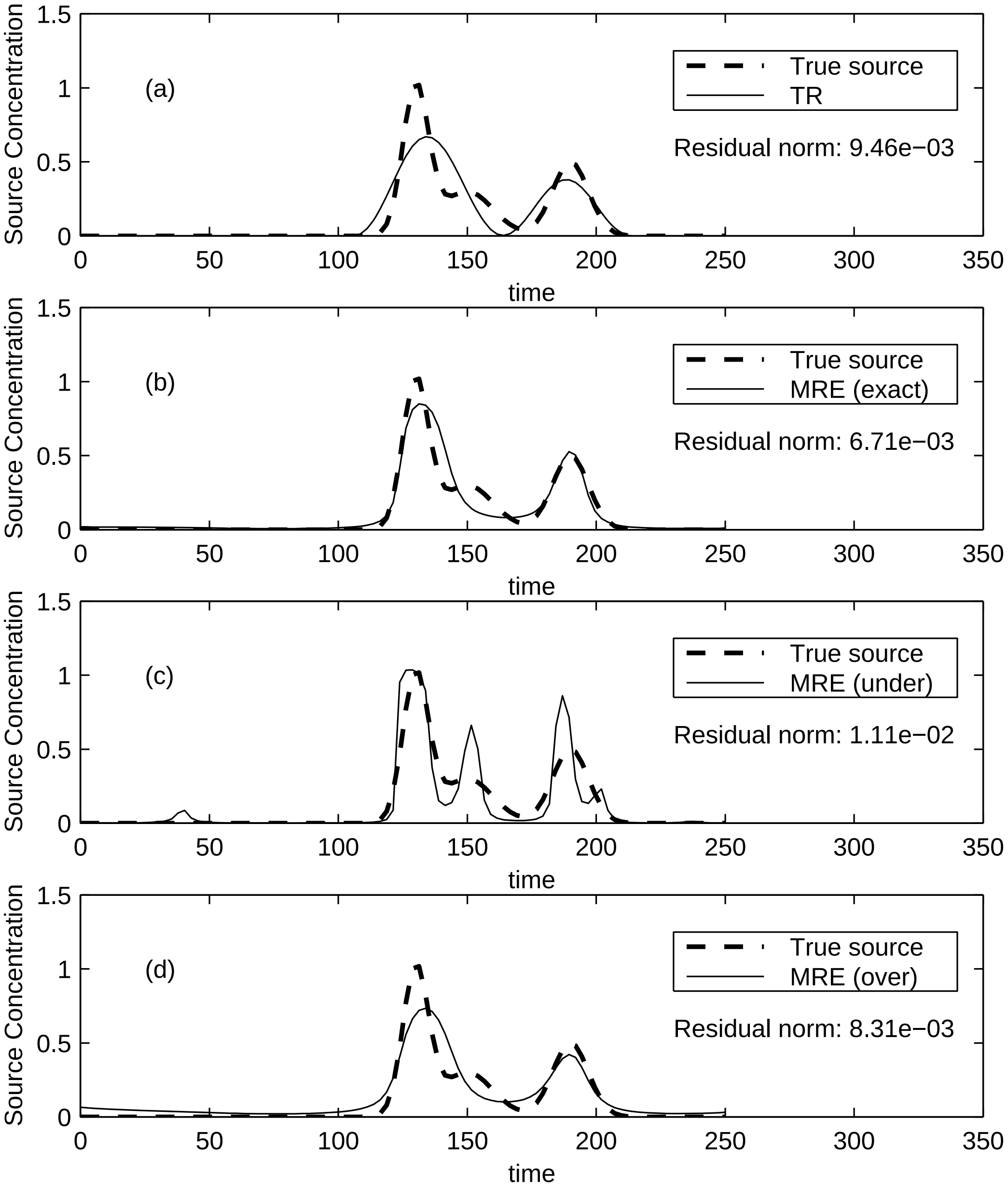

Figure 5.7: Tikhonov regularization and MRE results using inexact data (error model E2). (a) Tikhonov regularization results. (b) MRE results with the exact noise level. (c) MRE results with the underestimated noise level. (d) MRE results with the overestimated noise level. Parameters are: smooth input function, complete sampling frequency, true $\epsilon = 0.05$, $v = 1.0$, $D = 1.0$, and $T = 300$.

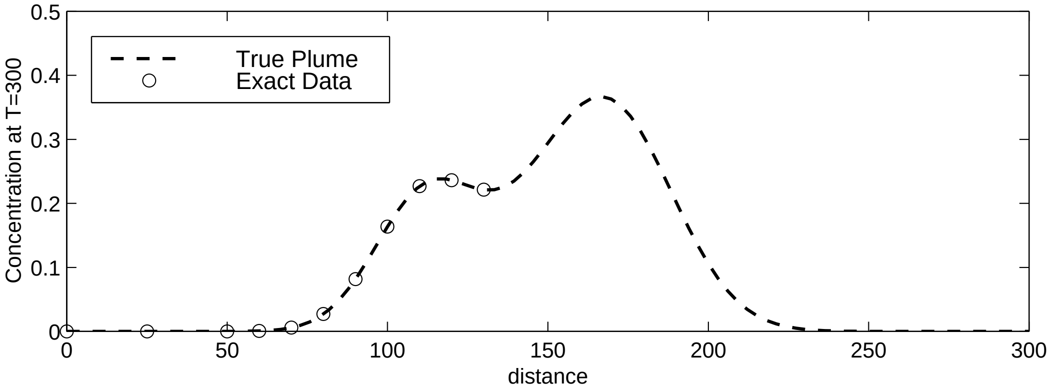

Figure 5.8: True plume at $T = 300$ and sampling locations for Run SF-1.

the slow convergence of the MRE algorithm to an underestimated noise level, and increase the noise level to a more appropriate value.

5.2.2 Sampling Frequency

In practical situations, it is often not possible to sample the entire plume. For example, suppose a property owner detects contamination in a monitoring well on his property. The owner can collect samples from other locations on his property, but it might not be possible to collect samples from off-site locations. To analyze this situation, we ran two test cases using different subsets of the sampled data used in the ideal scenario. The sample locations are shown in Figures 5.8 and 5.9, for sampling near the source (Run SF-1) and away from the source (Run SF-2), respectively. In each subset, one of the two peaks in the true plume is sampled.

The results are shown in Figures 5.10 and 5.11 for Runs SF-1 and SF-2, respectively. For SF-1, the two methods produce similar results; the late peak is reproduced well because the data sampled near the source at $T =$

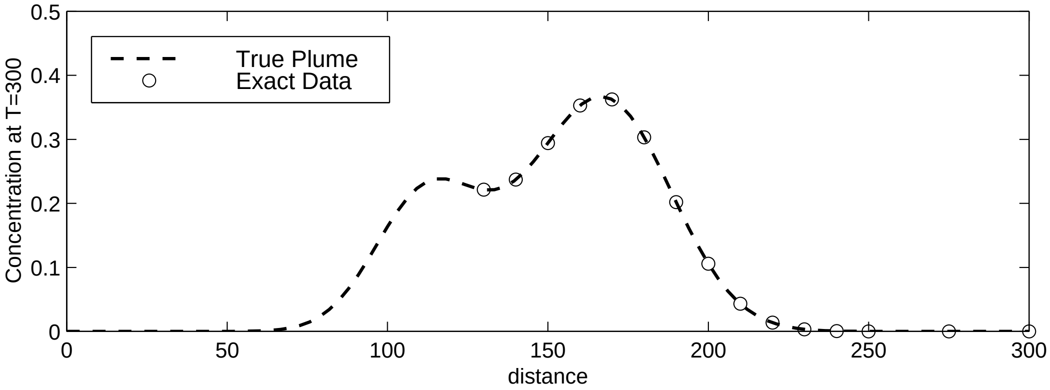

Figure 5.9: True plume at $T = 300$ and sampling locations for Run SF-2.

300 correspond to this late source release. The non-zero solution at early times indicates that the data is unable to provide any information about the source history at these times. This is further supported by the MRE solution uncertainty shown in Figure 5.12. The 5^{th} and 95^{th} percentile probability levels for $t < 150$ cover almost the entire range of possible values ($L_i = 0.$; $U_i = 1.1$). Since perfect data was used in this simulation, the model covariance (error) for the Tikhonov regularization results is zero.

For SF-2, both methods reproduce the first and second peaks fairly well, and even partially reproduce the third peak. The MRE solution indicates that the data does not provide any information about the source history for $t > 200$; the resulting non-zero solution is reflected in the value of the residual norm. The Tikhonov regularization solution appears to provide a reasonable solution for $t > 200$, and therefore has a lower residual norm. The uncertainty in the MRE solution is shown in Figure 5.13. The 5^{th} and 95^{th} percentile probability levels of the MRE solution cover the entire range of possible values for $t > 210$, indicating a high level of uncertainty in the solution in that region.

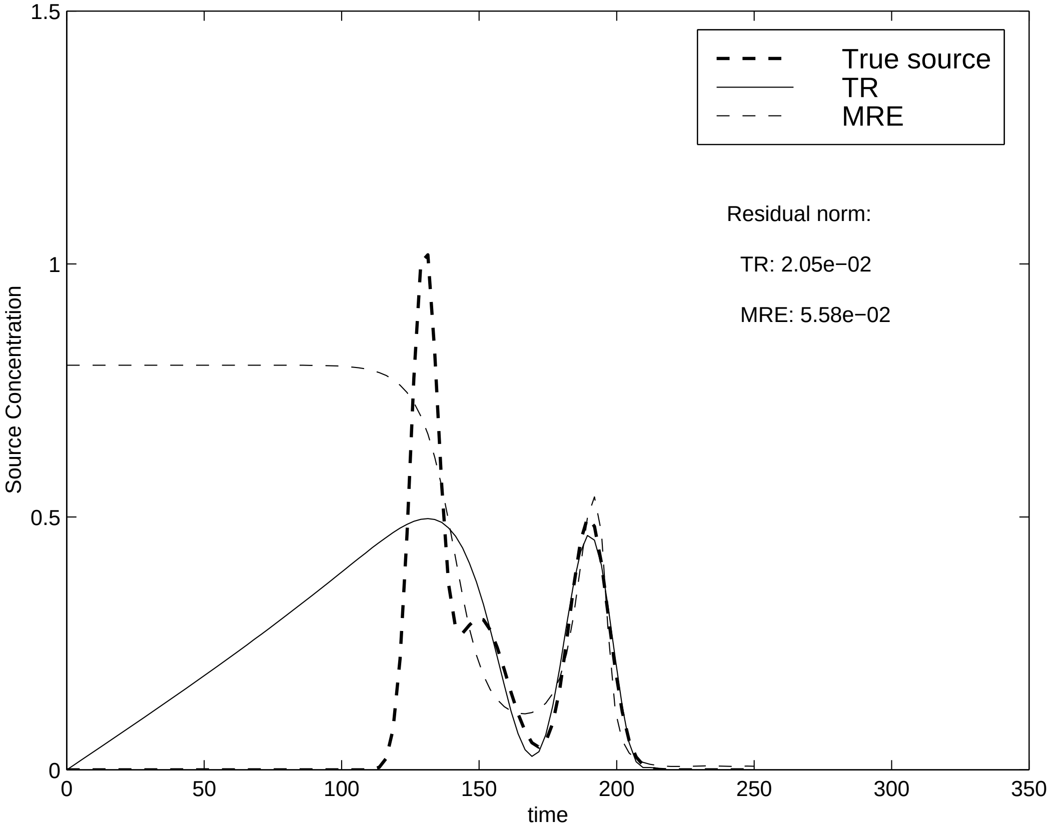

Figure 5.10: Tikhonov regularization and MRE results with an undersampled plume (Run SF-1). Parameters are: smooth input function, incomplete sampling frequency (near-source sampling), $\epsilon = 0.0$, $v = 1.0$, $D = 1.0$, and $T = 300$.

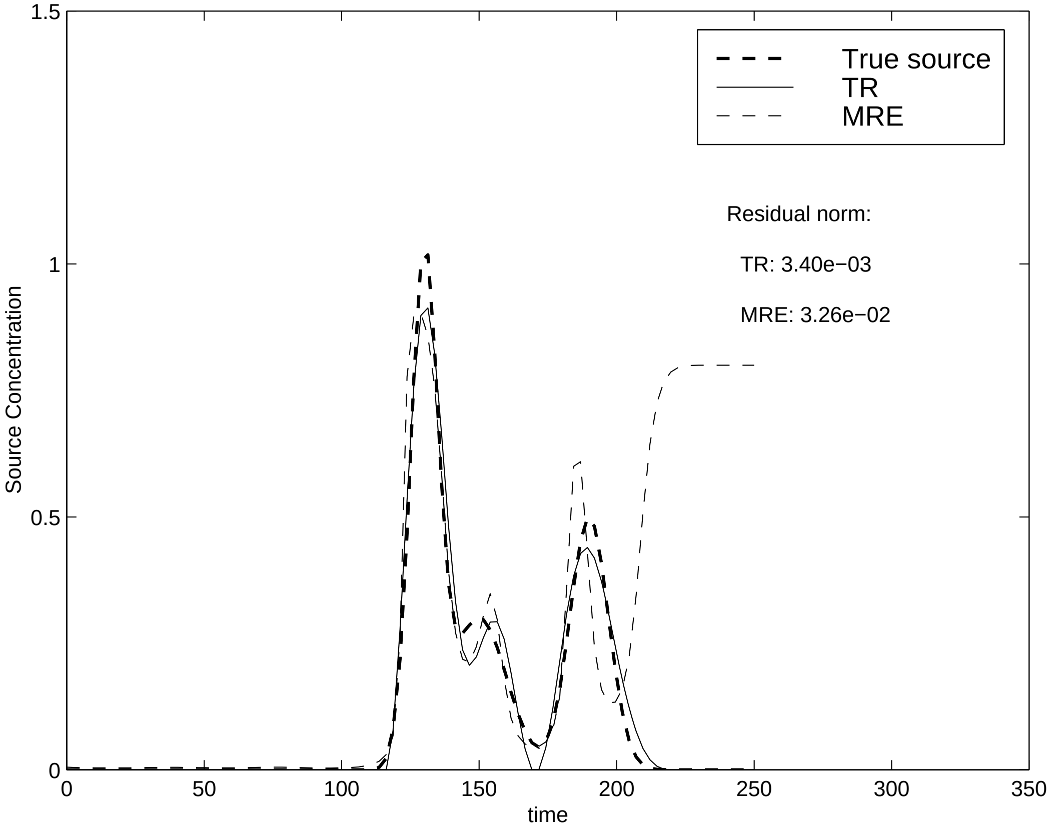

Figure 5.11: Tikhonov regularization and MRE results with an undersampled plume (Run SF-2). Parameters are: smooth input function, incomplete sampling frequency (distant sampling), $\epsilon = 0.0$, $v = 1.0$, $D = 1.0$, and $T = 300$.

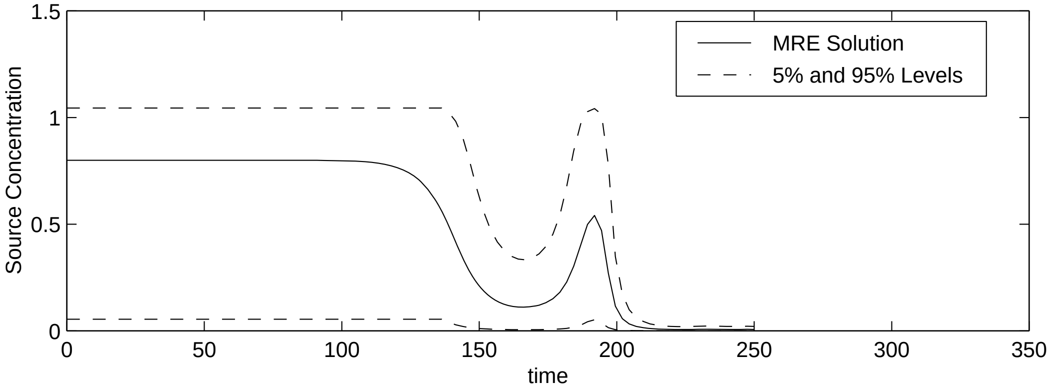

Figure 5.12: Uncertainty in the MRE results with an undersampled plume (Run SF-1).

5.2.3 Errors in Transport Parameters

In field situations, the transport parameters (v and D) are not known exactly. To approximate this situation, we ran four test cases with incorrect parameter values. In these runs, either the velocity or the dispersion coefficient was changed by 5% (increased in one run; decreased in another run), while the value of the other parameter was unchanged. All other parameters from the ideal scenario (sampled at $T = 300$) were unchanged. Although D is often modeled as a function of v, we assume that D and v are independent; therefore, increasing or decreasing the value of v did not affect the value of D.

Figure 5.14 shows the results of simulation with v overestimated ($v = 1.05$ and $D = 1.0$). The results of the two methods are essentially equivalent; the reconstructed solution is shifted to later times, and is more disperse than the true solution. The solution is shifted to later times because,

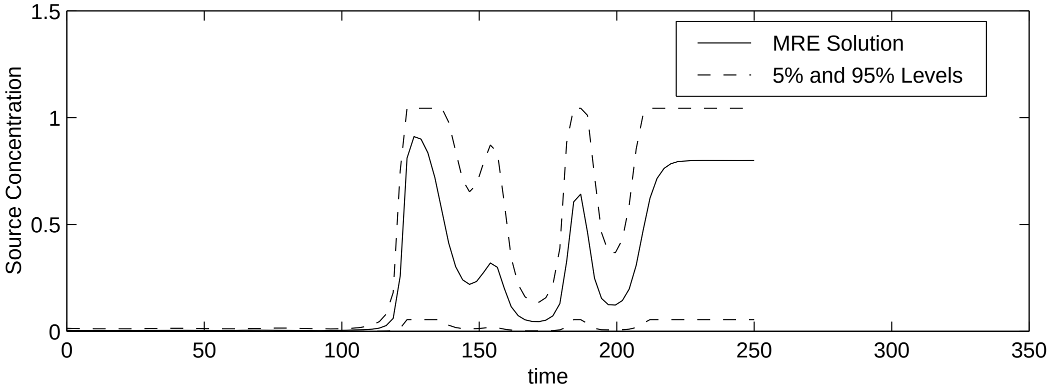

Figure 5.13: Uncertainty in the MRE results with an undersampled plume (Run SF-2).

with v overestimated, the plume is thought to travel faster than it actually does; therefore, the data can be matched only if a later release time compensates for the faster travel time. The reconstructed solutions are more disperse than the true solution because the faster travel time results in less spreading of the original plume. In order to match the spread indicated by the data, the source history must be more spread out. Figure 5.15 shows the results of the simulation with v underestimated ($v = 0.95$ and $D = 1.0$). In this case, the reconstructed source history functions are shifted to earlier times and are less disperse. The results of the two methods are approximately equal.

Figure 5.16 shows the results of the simulation with D overestimated ($v = 1.0$ and $D = 1.05$). The results of the two methods are similar; however, based on the residual norms, Tikhonov regularization performs slightly better. In both methods, the reconstructed solution is less disperse than the true solution. Since D is overestimated, the spread of the data can be matched only if the the reconstructed solution is less disperse. Figure 5.17 shows the results

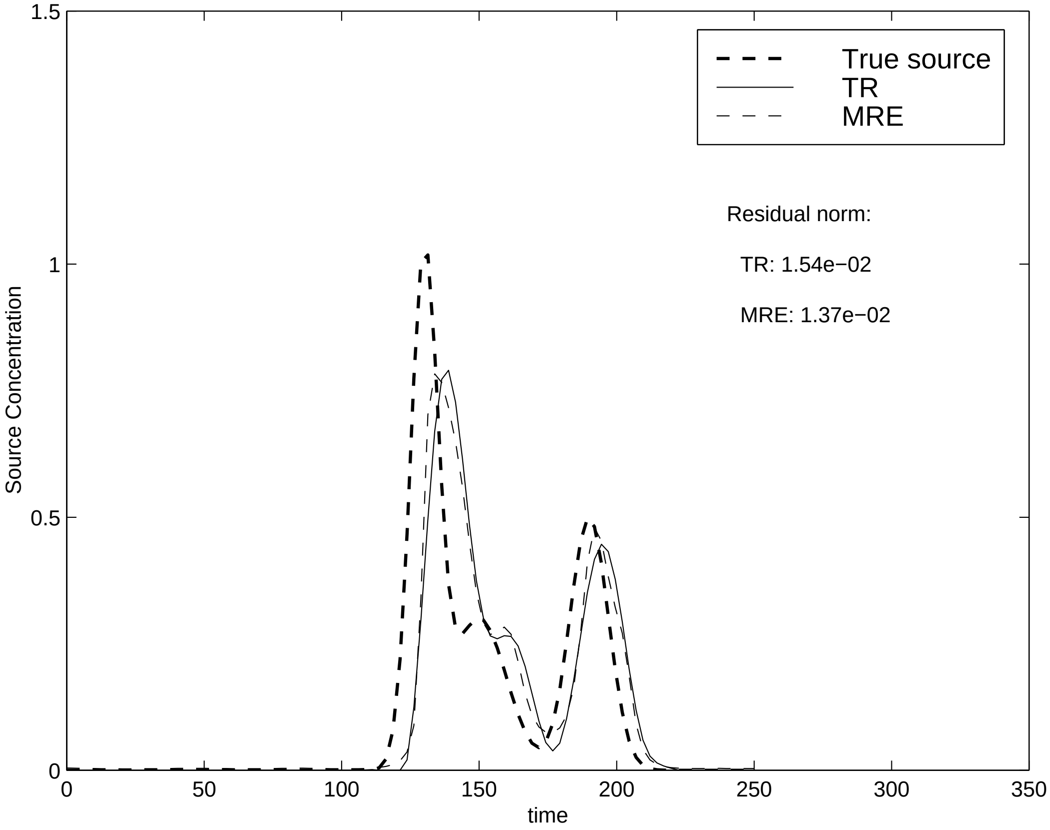

Figure 5.14: Tikhonov regularization and MRE results with an overestimated velocity. Parameters are: smooth input function, complete sampling frequency, $\epsilon = 0.0$, $v = 1.05$, $D = 1.0$, and $T = 300$.

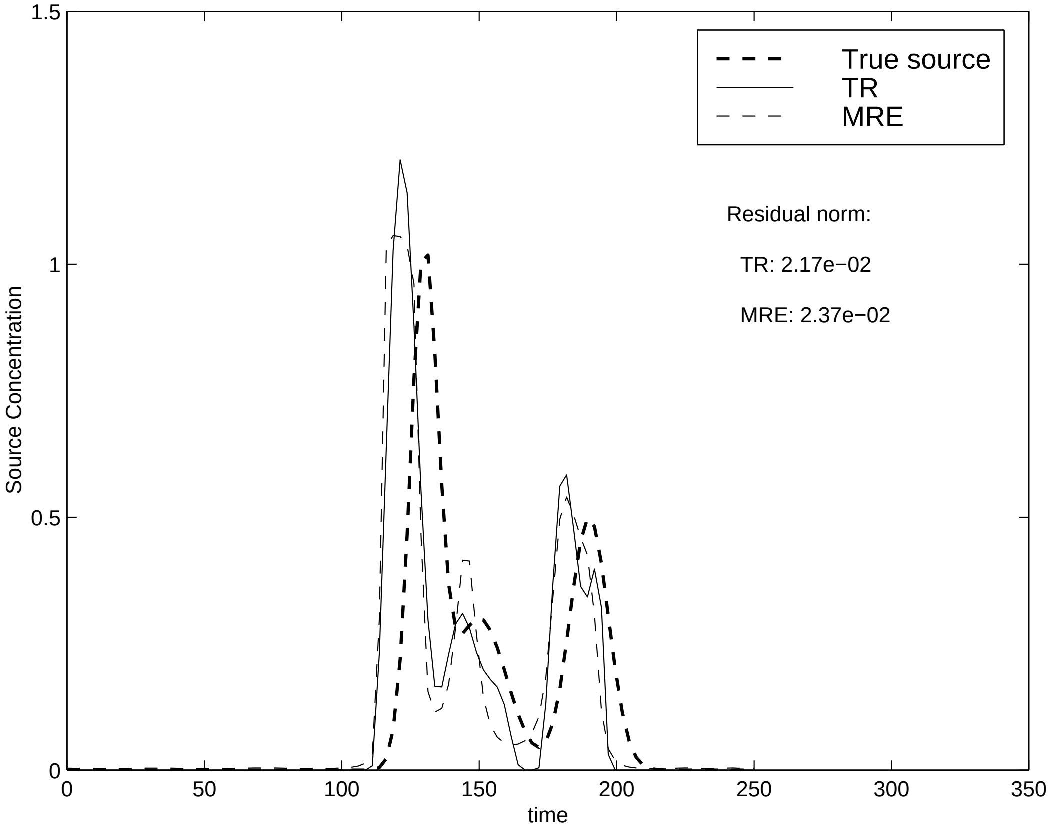

Figure 5.15: Tikhonov regularization and MRE results with an underestimated velocity. Parameters are: smooth input function, complete sampling frequency, $\epsilon = 0.0$, $v = 0.95$, $D = 1.0$, and $T = 300$.

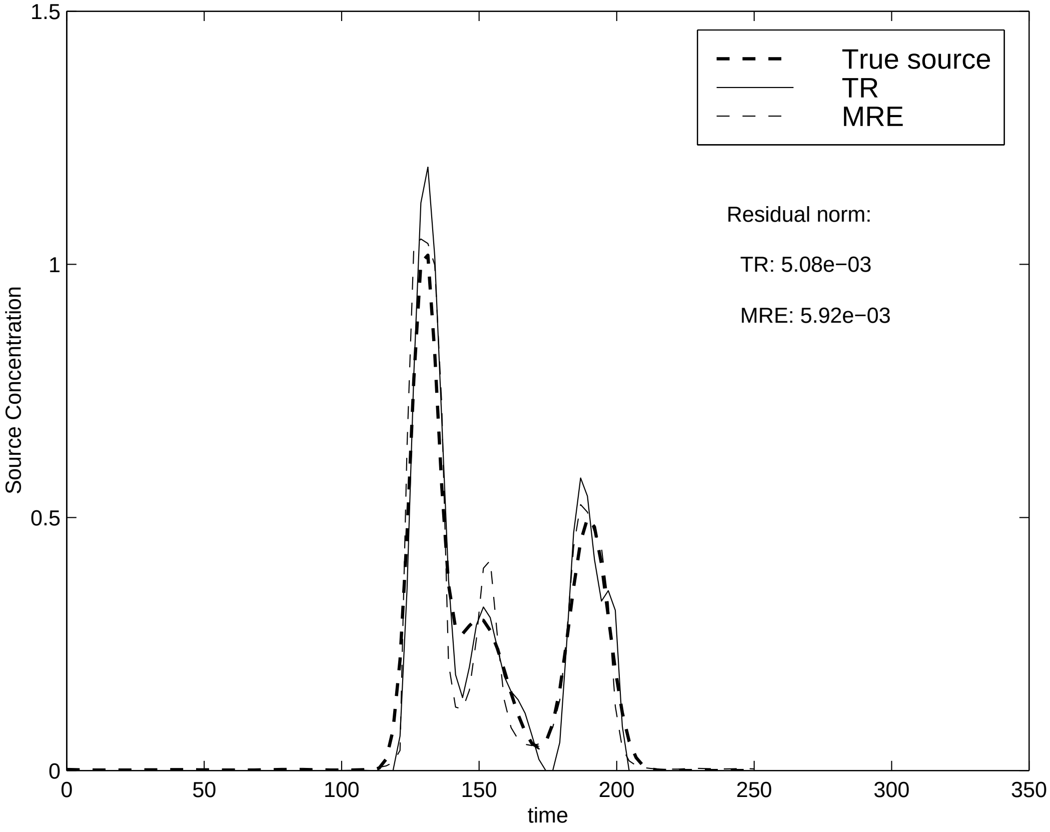

Figure 5.16: Tikhonov regularization and MRE results with an overestimated dispersion coefficient. Parameters are: smooth input function, complete sampling frequency, $\epsilon = 0.0$, $v = 1.0$, $D = 1.05$, and $T = 300$.

of the simulation with D underestimated ($v = 1.0$ and $D = 0.95$). In this case, the reconstructed source history functions are more disperse. Again, the results of the two methods are approximately equal; however, Tikhonov regularization performs slightly better.

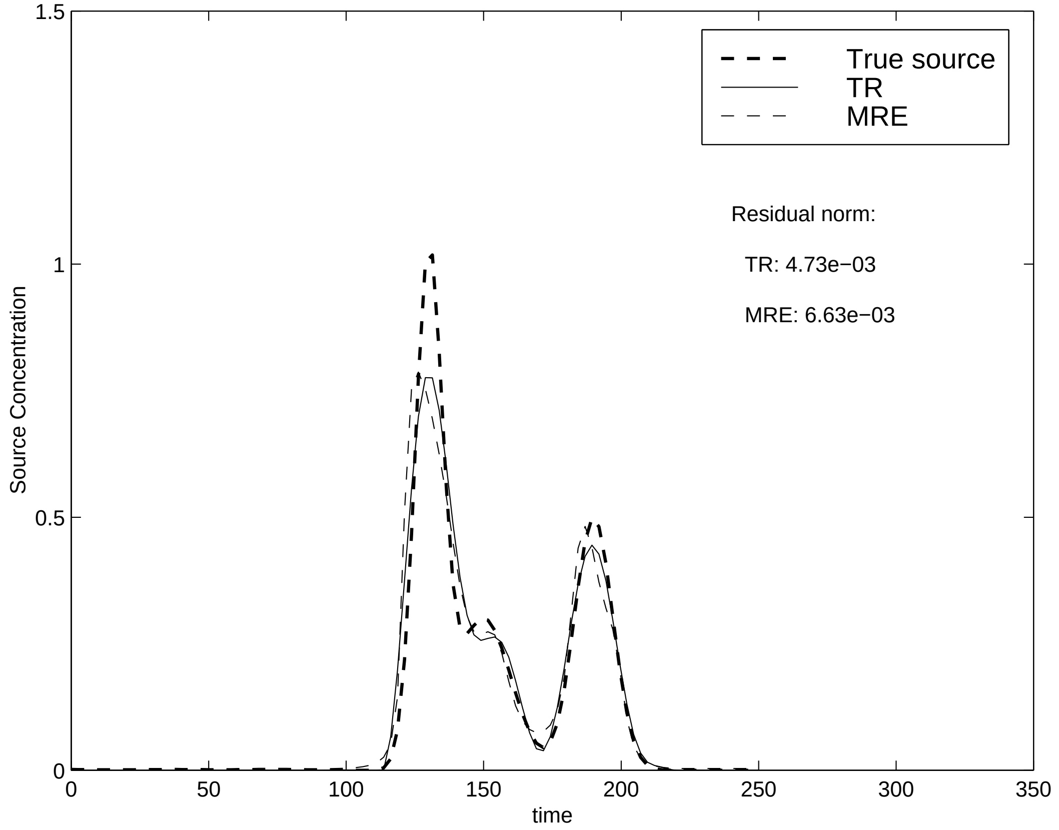

Figure 5.17: Tikhonov regularization and MRE results with an underestimated dispersion coefficient. Parameters are: smooth input function, complete sampling frequency, $\epsilon = 0.0$, $v = 1.0$, $D = 0.95$, and $T = 300$.

5.2.4 Smoothness of the Input Function

To evaluate the ability of each of the inverse methods to reconstruct a non-smooth source history function, we modified the ideal scenario so that the true source history was a square function given by Equation 4.1. The results are shown in Figure 5.18. The solution from the MRE method reproduces the true source history well. However, the solution from Tikhonov regularization cannot reproduce the constant, non-zero input between $t = 125$ and $t = 225$; instead the solution fluctuates around the true source history. This occurs because Tikhonov regularization attempts to fit a smooth function, and the best-fit smooth function that can capture the step change in input concentration must fluctuate. With the MRE method, the solution parameters are independent; therefore, it is possible to reproduce the sharp rise and fall and the constant section of the true source history function.

5.3 Evaluating the Effects of Multiple Factors

In the previous section, we evaluated the effectiveness of the inverse methods in reconstructing the release history under non-ideal situations. The non-ideal factors included measurement error in the sampled data, incomplete sampling of the plume, and incorrect estimations of the transport parameters. In practice, many or all of these non-ideal factors are present; therefore, we must evaluate the accuracy of the inverse solutions when multiple non-ideal factors are present.

We used a fractional factorial design to select the combinations of ideal and non-ideal factors to be evaluated. We selected five factors, and defined an

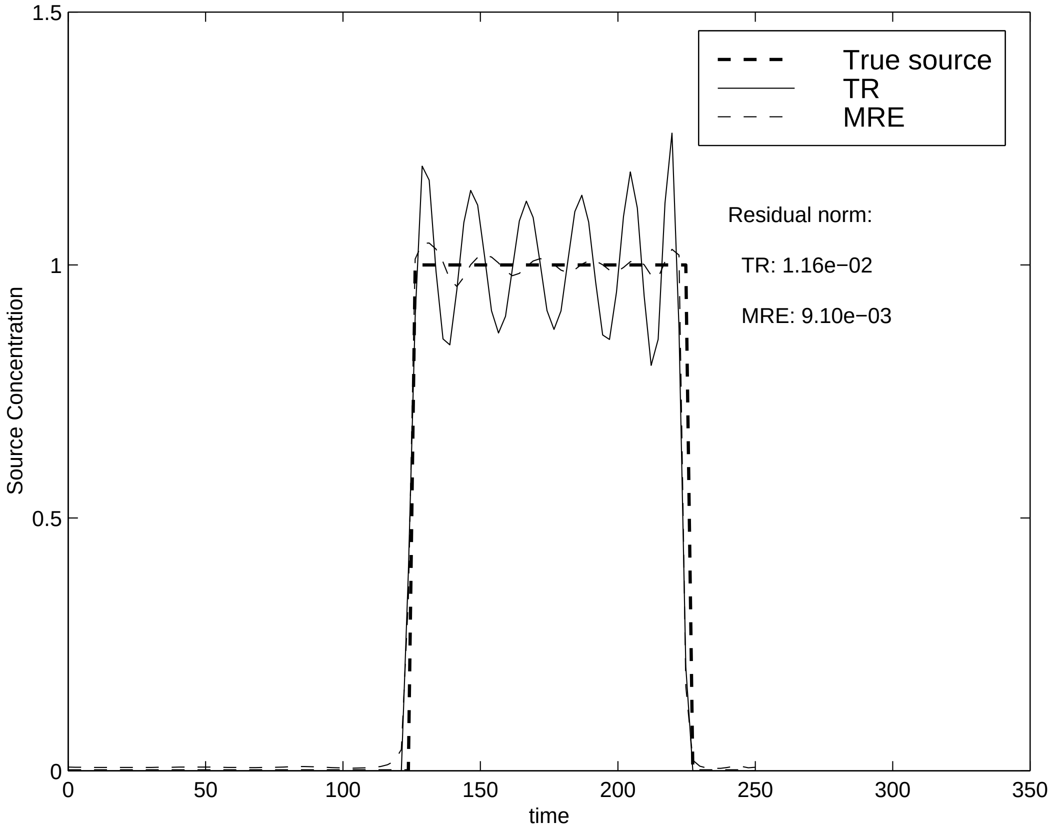

Figure 5.18: Tikhonov regularization and MRE results for the ideal scenario with a non-smooth source history function. Parameters are: square input function, complete sampling frequency, $\epsilon = 0.0$, $v = 1.0$, $D = 1.0$, and $T = 300$.

Table 5.1: Coding chart for the fractional factorial design.

Factor No.	Factor	Ideal Value	Non-ideal Value
1	Noise level (ϵ)	0.0	0.05
2	Sampling frequency	complete	incomplete
3	Value of v	correct ($v = 1.0$)	incorrect ($v = 1.05$)
4	Value of D	correct ($D = 1.0$)	incorrect ($D = 0.95$)
5	Input function	smooth	square

ideal and non-ideal value for each. The coding chart for these factors is shown in Table 5.1.

Since the results of Section 5.2 show that the relative success of the two inverse methods does not depend on the particular error model used to generate inexact data, we used error model E1 (Equation 5.1) for these multi-factor simulations. Also, we used the exact noise level with the MRE method because overestimating the noise level did not have a significant effect on the results, and the solution did not converge for an underestimated noise level. For an underestimated noise level, the measurements are accurate to within ϵ of the true data, but the model is trying to fit a solution to within $\epsilon/2$ of the true data. Since this solution does not exist, the model does not converge. For the incomplete sampling frequency, we used the sampling location shown in Figure 5.9, for sampling away from the source.

Evaluating all possible combinations of these five factors would require $2^5 = 32$ simulations. By using a fractional factorial design, we can systematically select a subset of these combinations to reduce the number of simulations, while still obtaining good results. We used a 2_V^{5-1} fractional factorial design,

Table 5.2: Design matrix for the fractional factorial design.

	Factor				
Run Number	1	2	3	4	5
MF-1	-	-	-	-	+
MF-2	-	-	-	+	-
MF-3	-	-	+	-	-
MF-4	-	-	+	+	+
MF-5	-	+	-	-	-
MF-6	-	+	-	+	+
MF-7	-	+	+	-	+
MF-8	-	+	+	+	-
MF-9	+	-	-	-	-
MF-10	+	-	-	+	+
MF-11	+	-	+	-	+
MF-12	+	-	+	+	-
MF-13	+	+	-	-	+
MF-14	+	+	-	+	-
MF-15	+	+	+	-	-
MF-16	+	+	+	+	+

in which all possible combinations of four ($5 - 1 = 4$) of the factors are used, resulting in $2^4 = 16$ different combinations. These are shown in Table 5.2 for the first four factors, where "+" indicates that the ideal value of the factor was used, and "—" indicates that the non-ideal value was used. The value used for the fifth factor was the product of the values used for the first four factors. The products are calculated by substituting -1 for "—" and $+1$ for "+" (Law and Kelton, 1991).

With this design, the main effects and two-way interactions can be evaluated adequately. The main effects measure the average change in response (the residual norm) to a change in one of the factors. Two-way interactions measures the degree to which the effect of one factor depends on the value of another factor. Higher-order interactions measure the degree of interaction between more factors. In the fractional factorial design, we do not run simulations for all possible combinations; therefore, some information is sacrificed and the resolution of the design decreases. The resolution is given by the Roman number subscript, e.g. 2_V^{5-1} is a resolution V design. The resolution describes the levels of interactions that can be reliably evaluated. For a resolution V design, main effects (one-way interactions) can be evaluated reliably if fourth-order effects and higher are negligible; and two-way interactions can be evaluated reliably if three-way interactions and higher are negligible (Law and Kelton, 1991).

For each design point shown in Table 5.2, we solved the corresponding inverse problem using both Tikhonov regularization and minimum relative entropy inversion. We compared the residual norms of the results to evaluate the main effects, and we compared the differences in the residual norms of the two methods. In these runs, all unspecified parameters (e.g. sampling time, upper bounds, etc.) have the same values as in the ideal scenario.

The results are shown in Figure 5.19–5.34 and in Table 5.3, which lists the residual norms of the solutions of both methods. For each run, the smaller of the two norms is printed in bold-faced type. The absolute and relative differences of the two norms are also shown in the table. The difference is calculated by subtracting the MRE norm from the TR norm; and the relative difference of the norms is this difference normalized by the MRE norm. Note that the MRE method performs better in eleven of the 16 runs. Of the five runs in which Tikhonov regularization performed better (MF-6, MF-9, MF-11,

Table 5.3: Results of multi-factor simulations.

	Residual Norm			
Run Number	TR	MRE	Difference	Rel. Difference
MF-1	3.04×10^{-2}	$\mathbf{2.26} \times \mathbf{10^{-2}}$	7.80×10^{-3}	3.45×10^{-1}
MF-2	2.52×10^{-2}	$\mathbf{1.85} \times \mathbf{10^{-2}}$	6.70×10^{-3}	3.62×10^{-1}
MF-3	3.71×10^{-2}	$\mathbf{1.21} \times \mathbf{10^{-2}}$	2.50×10^{-2}	2.07×10^{0}
MF-4	2.26×10^{-2}	$\mathbf{1.36} \times \mathbf{10^{-2}}$	9.00×10^{-3}	6.62×10^{-1}
MF-5	1.67×10^{-2}	$\mathbf{1.59} \times \mathbf{10^{-2}}$	8.00×10^{-4}	5.03×10^{-2}
MF-6	$\mathbf{1.71} \times \mathbf{10^{-2}}$	1.89×10^{-2}	-1.80×10^{-3}	-9.52×10^{-2}
MF-7	8.98×10^{-3}	$\mathbf{8.02} \times \mathbf{10^{-3}}$	9.60×10^{-4}	1.20×10^{-1}
MF-8	1.39×10^{-2}	$\mathbf{1.33} \times \mathbf{10^{-2}}$	6.00×10^{-4}	4.51×10^{-2}
MF-9	$\mathbf{1.59} \times \mathbf{10^{-2}}$	1.68×10^{-2}	-9.00×10^{-4}	-5.36×10^{-2}
MF-10	1.88×10^{-2}	$\mathbf{1.70} \times \mathbf{10^{-2}}$	1.80×10^{-3}	1.06×10^{-1}
MF-11	$\mathbf{6.09} \times \mathbf{10^{-3}}$	1.04×10^{-2}	-4.31×10^{-3}	-4.14×10^{-1}
MF-12	6.94×10^{-3}	$\mathbf{4.63} \times \mathbf{10^{-3}}$	2.31×10^{-3}	4.99×10^{-1}
MF-13	1.40×10^{-2}	$\mathbf{1.23} \times \mathbf{10^{-2}}$	1.70×10^{-3}	1.38×10^{-1}
MF-14	$\mathbf{1.55} \times \mathbf{10^{-2}}$	1.57×10^{-2}	-2.00×10^{-4}	-1.27×10^{-2}
MF-15	1.00×10^{-2}	$\mathbf{9.99} \times \mathbf{10^{-3}}$	1.00×10^{-5}	1.00×10^{-3}
MF-16	$\mathbf{2.73} \times \mathbf{10^{-3}}$	5.03×10^{-3}	-2.30×10^{-3}	-4.57×10^{-1}

MF-14, and MF-16), the residual norms were very close for two of those runs (MF-9 and MF-14); in the remaining three runs, the common factor was the smooth input function.

Table 5.4 shows the difference in the residual norms, separated out by the values of each factor. For each factor, the column labeled "Ideal" lists the average value of the difference of the residual norms for runs in which that factor was at its ideal value; and the column labeled "Non-ideal" lists the average difference of the residual norms for runs in which that factor was at its non-ideal value. Recall that a positive value of the difference indicates that the MRE method performed better; while a negative value indicates that

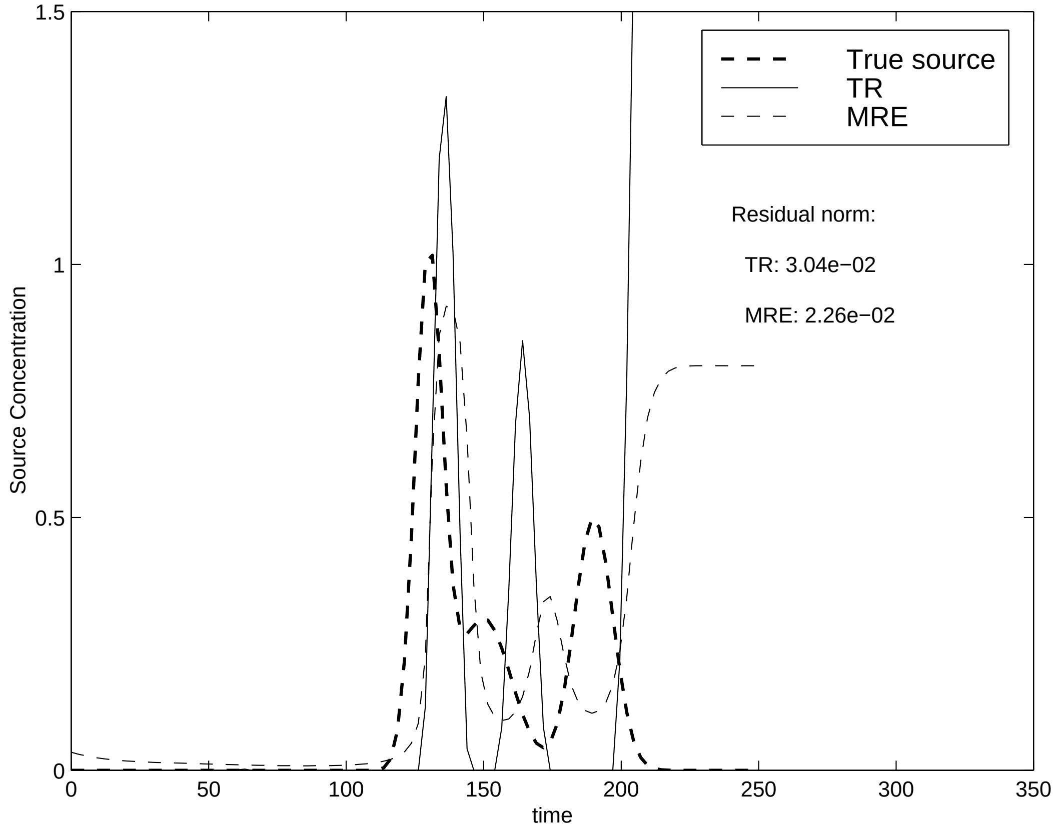

Figure 5.19: Tikhonov regularization and MRE results for multi-factor Run MF-1. Parameters are: smooth input function, incomplete sampling frequency, $\epsilon = 0.05$, $v = 1.05$, $D = 1.05$, and $T = 300$.

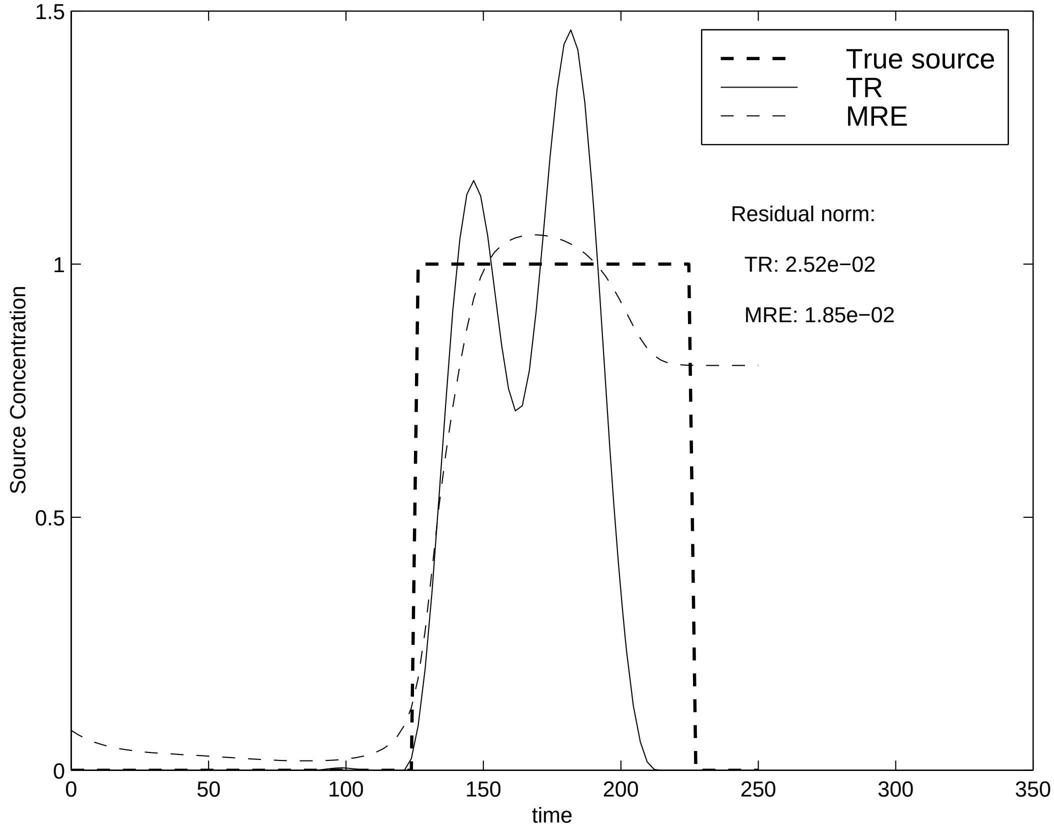

Figure 5.20: Tikhonov regularization and MRE results for multi-factor Run MF-2. Parameters are: square input function, incomplete sampling frequency, $\epsilon = 0.05$, $v = 1.05$, $D = 1.0$, and $T = 300$.

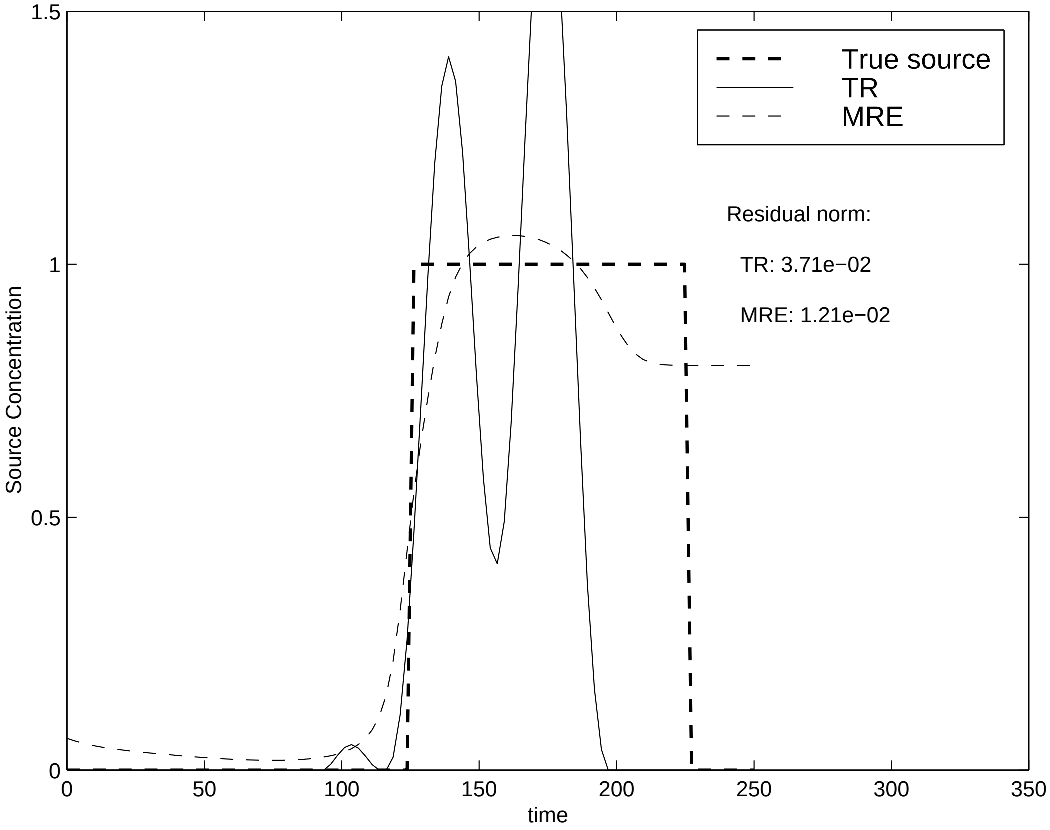

Figure 5.21: Tikhonov regularization and MRE results for multi-factor Run MF-3. Parameters are: square input function, incomplete sampling frequency, $\epsilon = 0.05$, $v = 1.0$, $D = 0.95$, and $T = 300$.

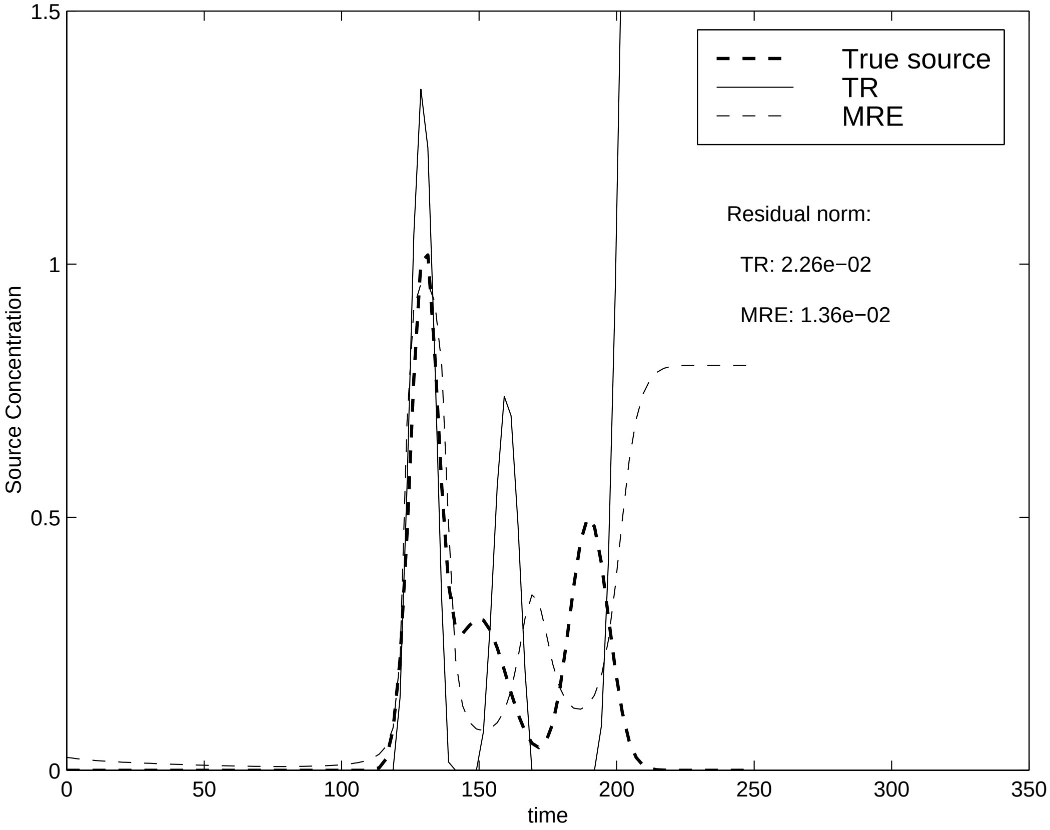

Figure 5.22: Tikhonov regularization and MRE results for multi-factor Run MF-4. Parameters are: smooth input function, incomplete sampling frequency, $\epsilon = 0.05$, $v = 1.0$, $D = 1.0$, and $T = 300$.

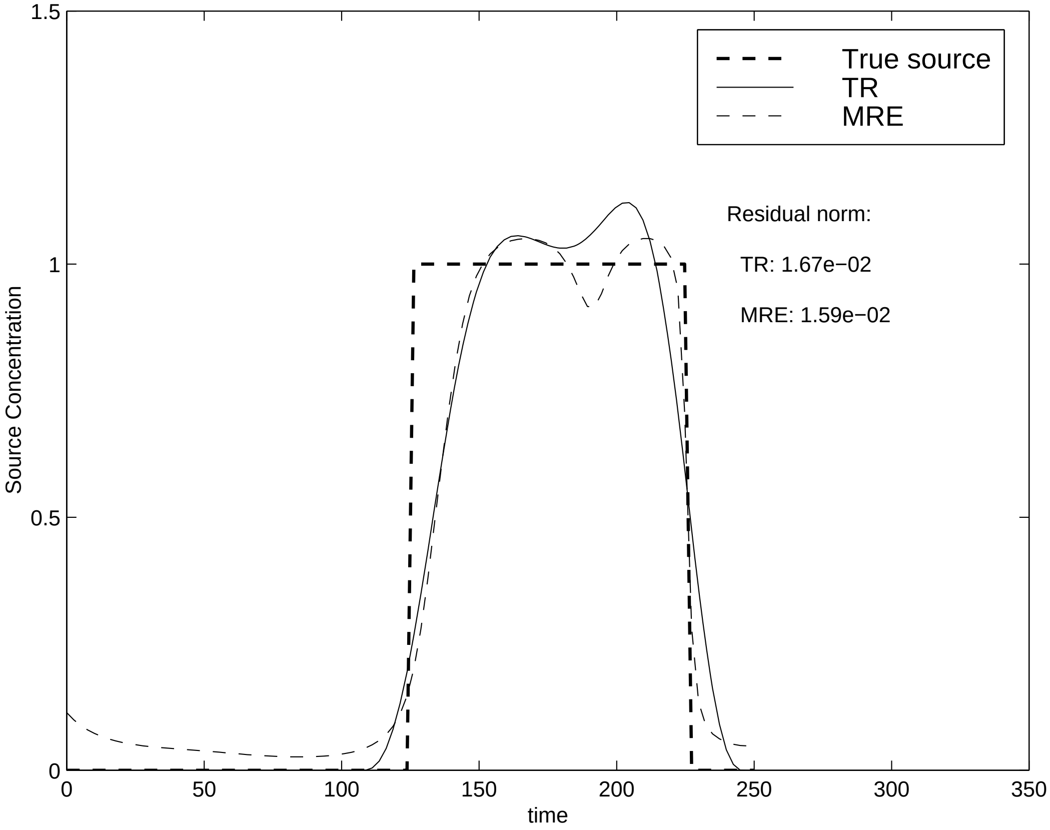

Figure 5.23: Tikhonov regularization and MRE results for multi-factor Run MF-5. Parameters are: square input function, complete sampling frequency, $\epsilon = 0.05$, $v = 1.05$, $D = 0.95$, and $T = 300$.

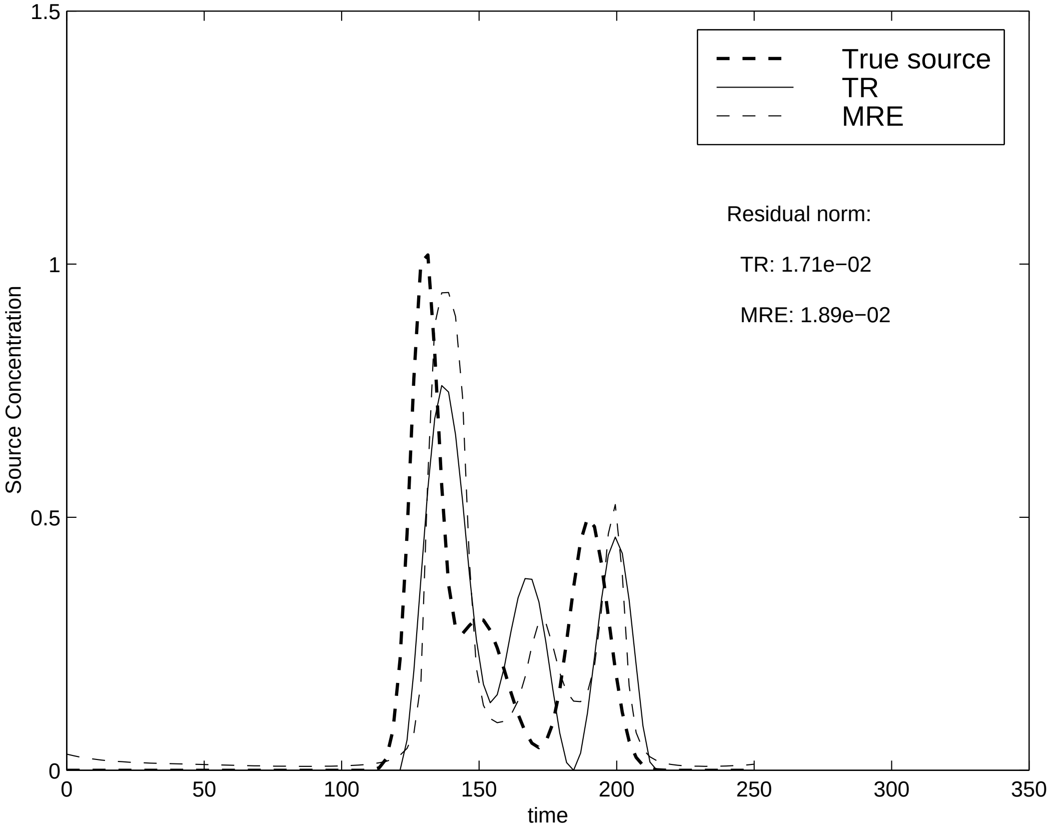

Figure 5.24: Tikhonov regularization and MRE results for multi-factor Run MF-6. Parameters are: smooth input function, complete sampling frequency, $\epsilon = 0.05$, $v = 1.05$, $D = 1.0$, and $T = 300$.

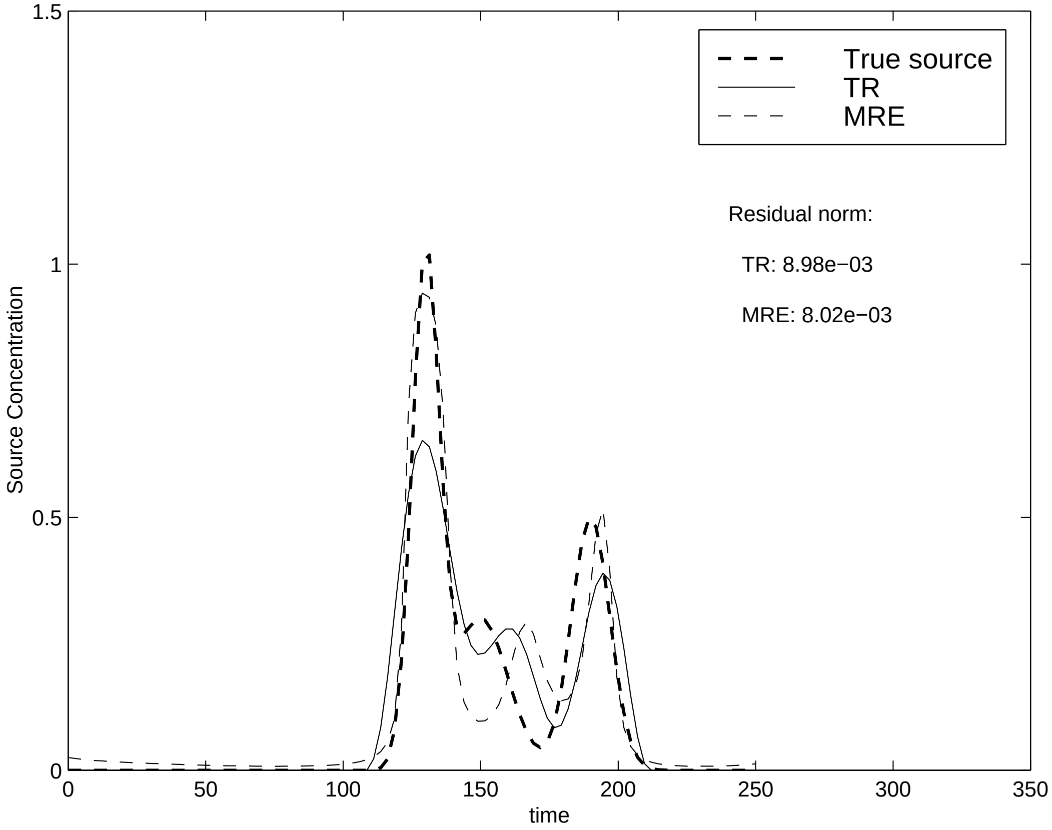

Figure 5.25: Tikhonov regularization and MRE results for multi-factor Run MF-7. Parameters are: smooth input function, complete sampling frequency, $\epsilon = 0.05$, $v = 1.0$, $D = 0.95$, and $T = 300$.

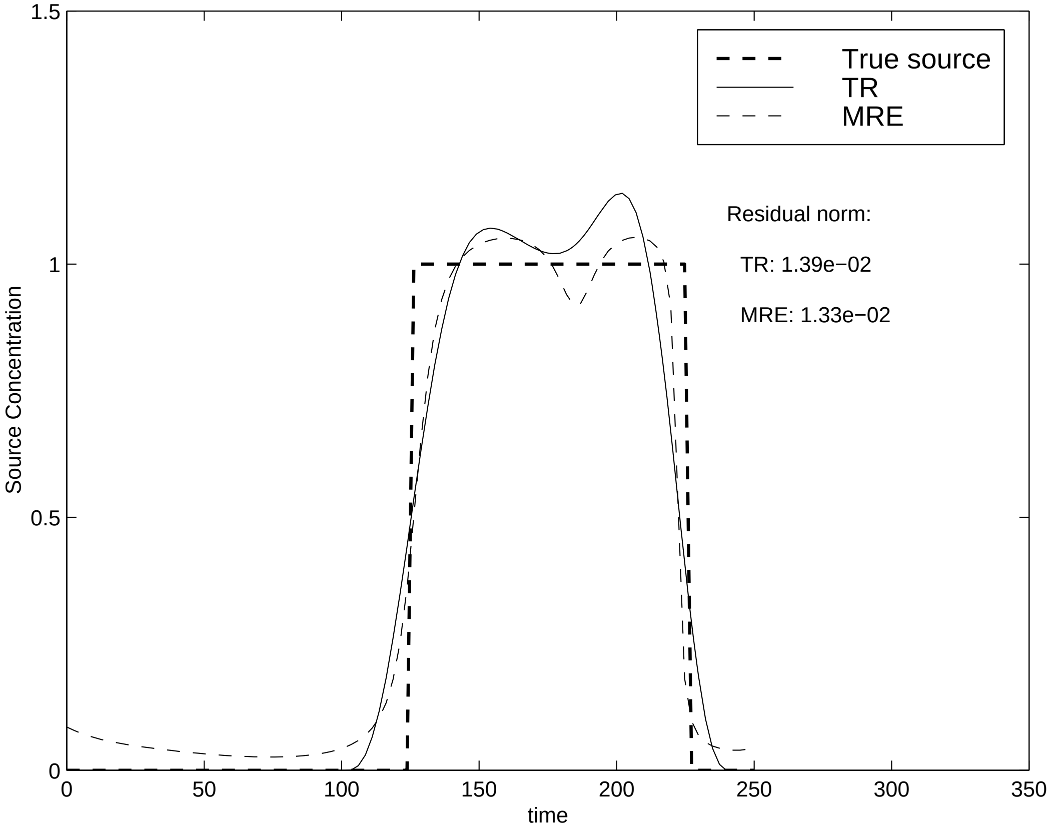

Figure 5.26: Tikhonov regularization and MRE results for multi-factor Run MF-8. Parameters are: square input function, complete sampling frequency, $\epsilon = 0.05$, $v = 1.0$, $D = 1.0$, and $T = 300$.

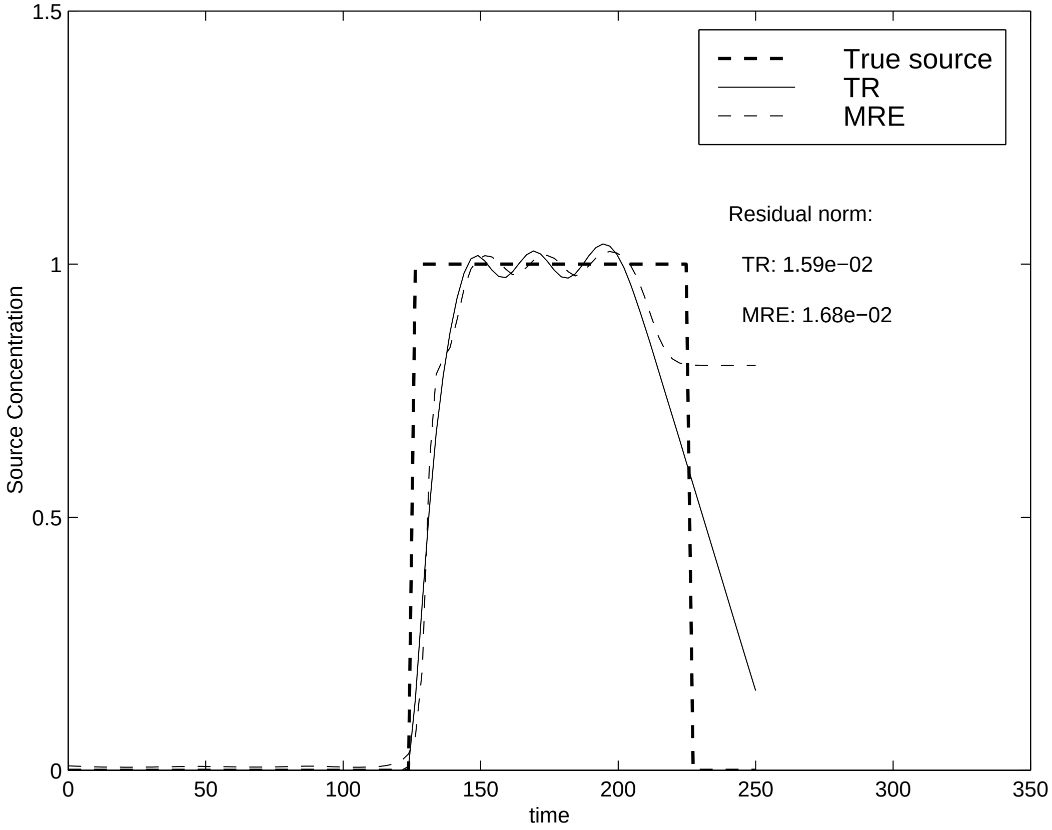

Figure 5.27: Tikhonov regularization and MRE results for multi-factor Run MF-9. Parameters are: square input function, incomplete sampling frequency, $\epsilon = 0.0$, $v = 1.05$, $D = 0.95$, and $T = 300$.

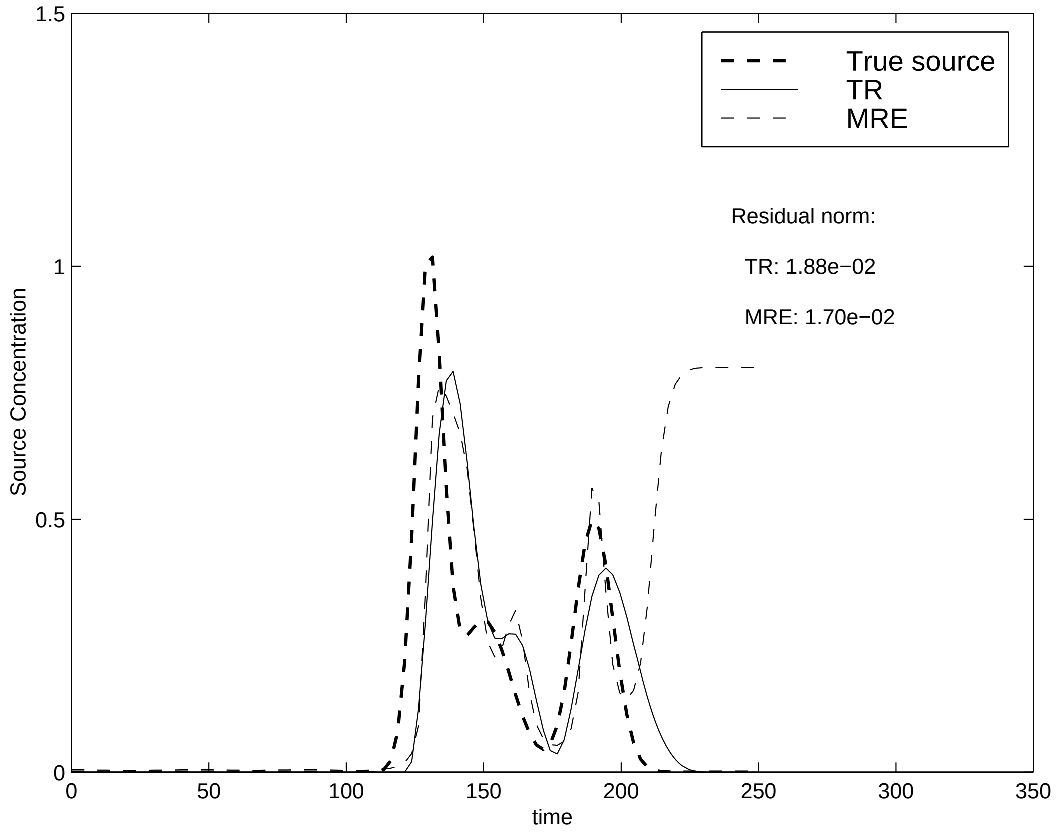

Figure 5.28: Tikhonov regularization and MRE results for multi-factor Run MF-10. Parameters are: smooth input function, incomplete sampling frequency, $\epsilon = 0.0$, $v = 1.05$, $D = 1.0$, and $T = 300$.

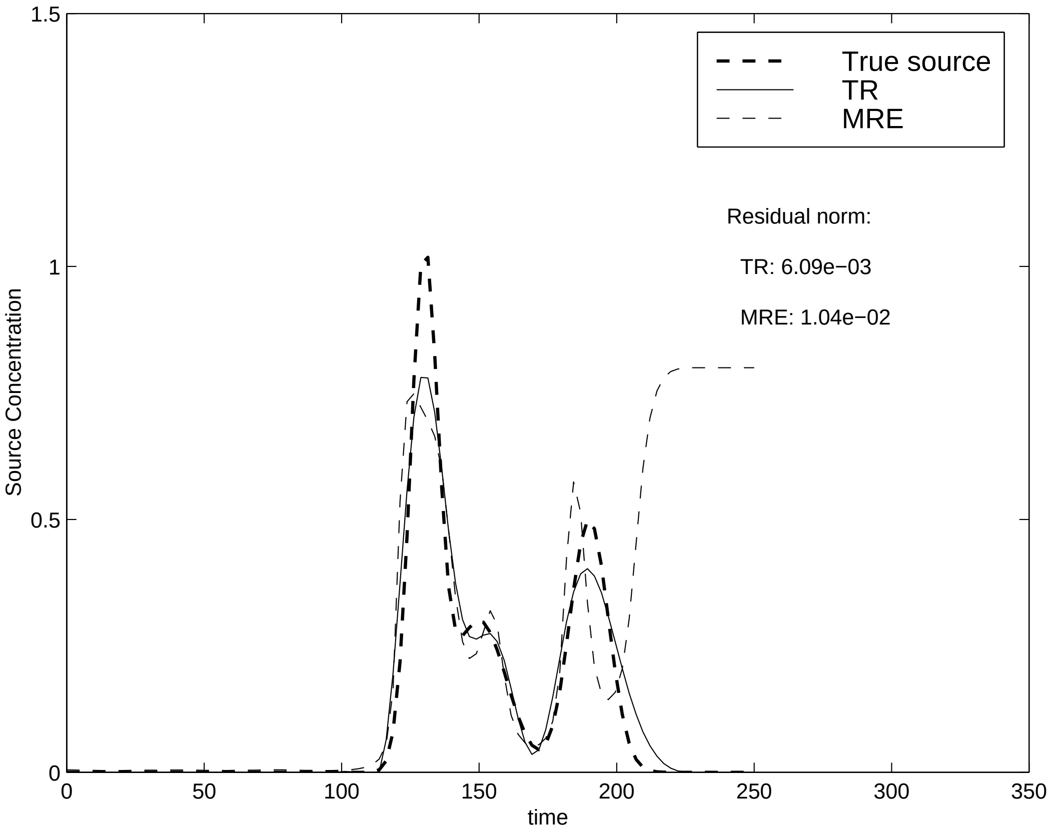

Figure 5.29: Tikhonov regularization and MRE results for multi-factor Run MF-11. Parameters are: smooth input function, incomplete sampling frequency, $\epsilon = 0.0$, $v = 1.0$, $D = 0.95$, and $T = 300$.

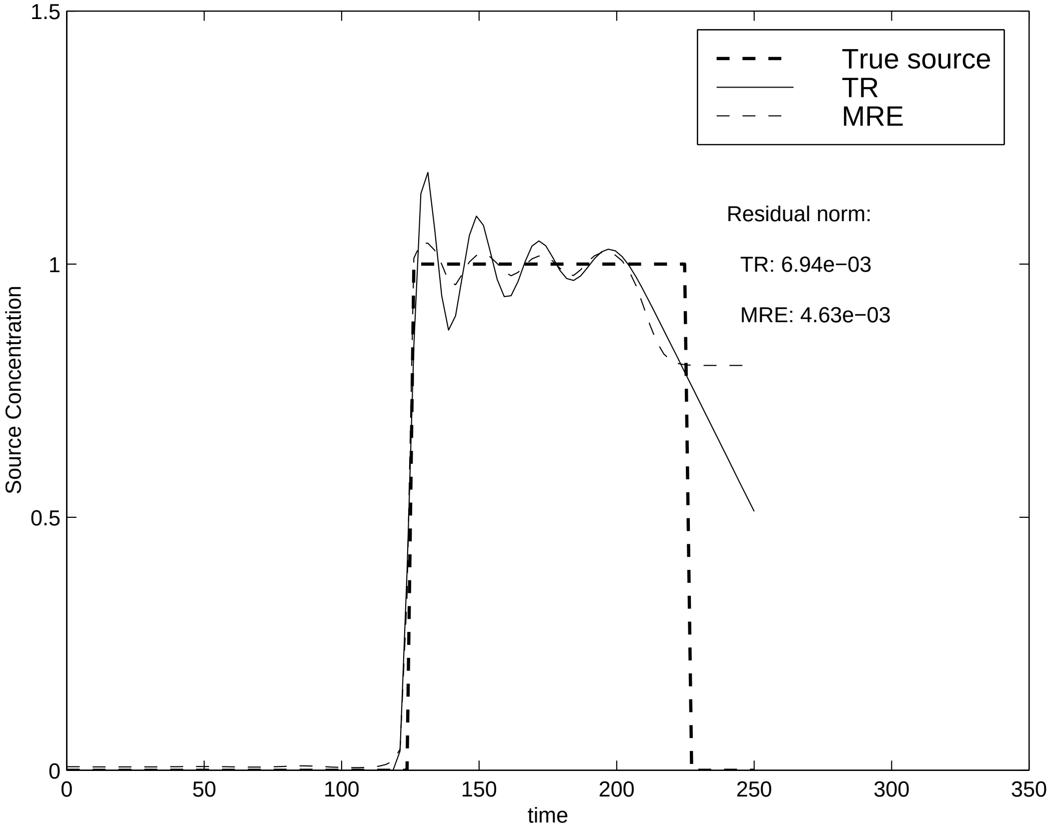

Figure 5.30: Tikhonov regularization and MRE results for multi-factor Run MF-12. Parameters are: square input function, incomplete sampling frequency, $\epsilon = 0.0$, $v = 1.0$, $D = 1.0$, and $T = 300$.

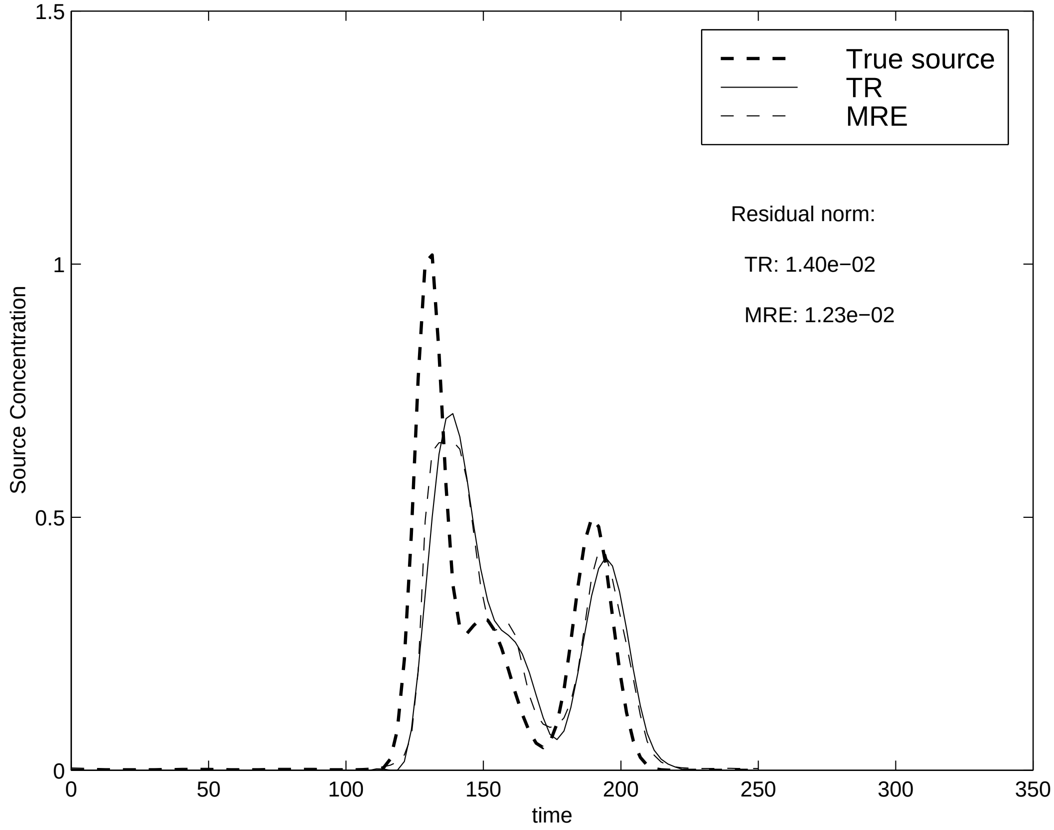

Figure 5.31: Tikhonov regularization and MRE results for multi-factor Run MF-13. Parameters are: smooth input function, complete sampling frequency, $\epsilon = 0.0$, $v = 1.05$, $D = 0.95$, and $T = 300$.

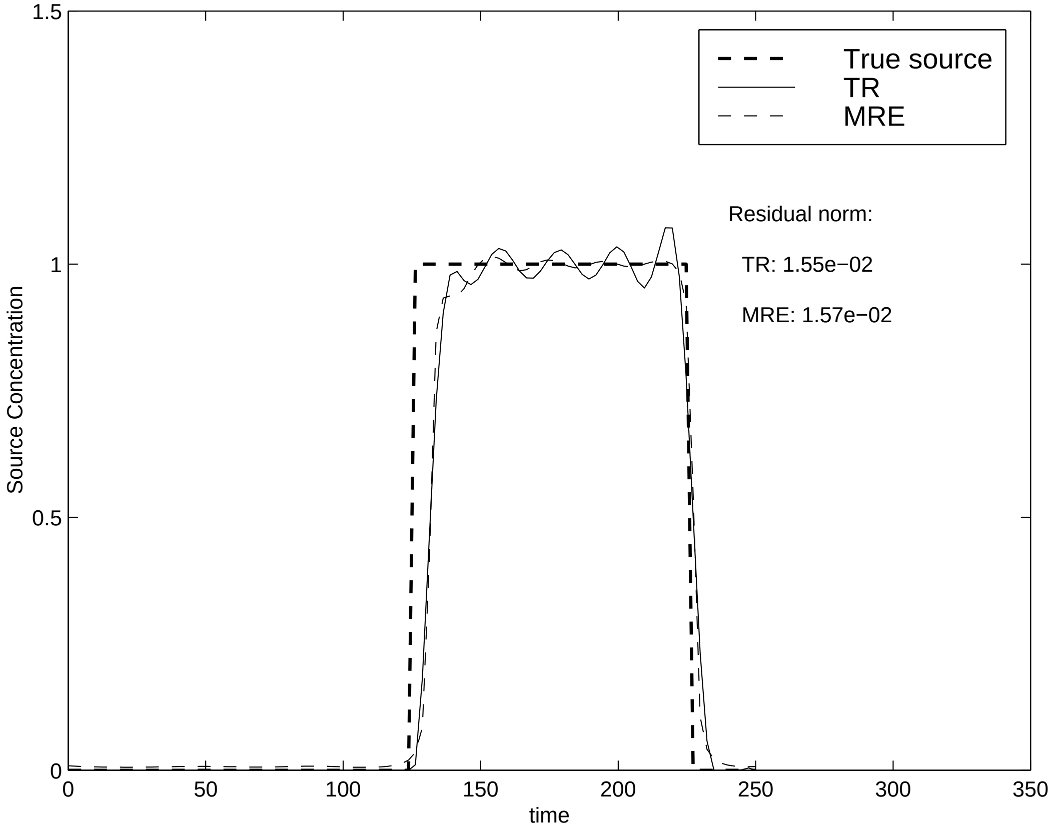

Figure 5.32: Tikhonov regularization and MRE results for multi-factor Run MF-14. Parameters are: square input function, complete sampling frequency, $\epsilon = 0.0$, $v = 1.05$, $D = 1.0$, and $T = 300$.

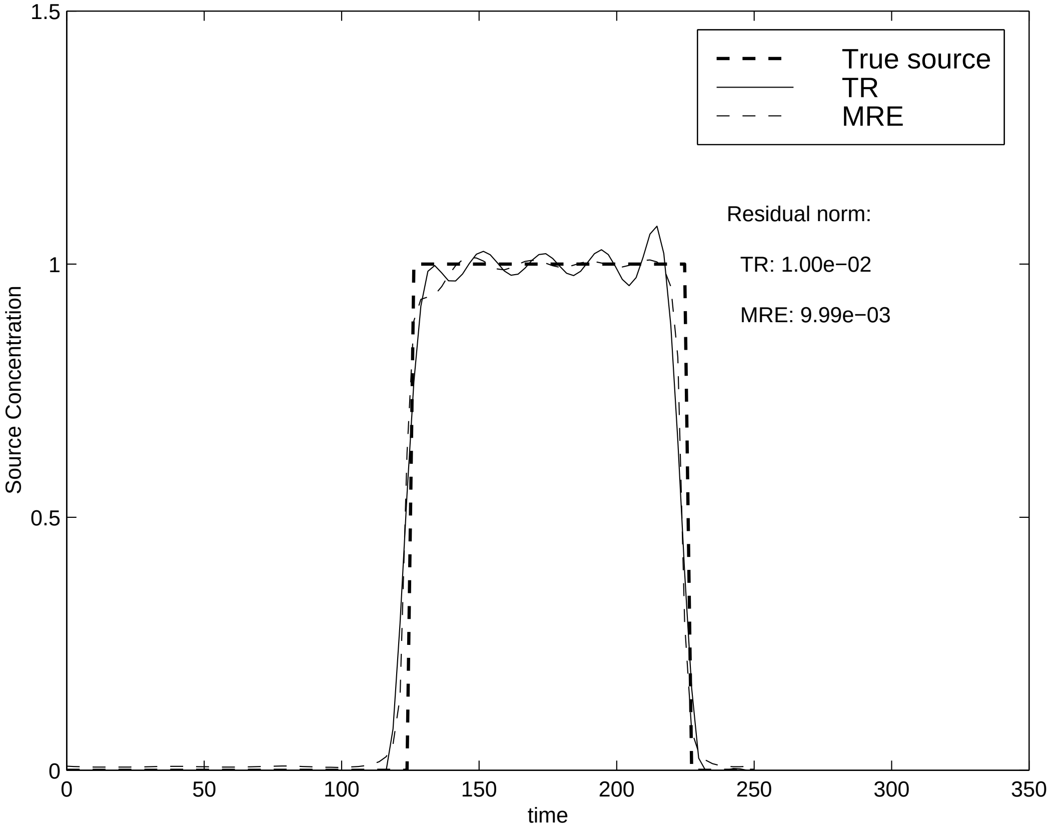

Figure 5.33: Tikhonov regularization and MRE results for multi-factor Run MF-15. Parameters are: square input function, complete sampling frequency, $\epsilon = 0.0$, $v = 1.0$, $D = 0.95$, and $T = 300$.

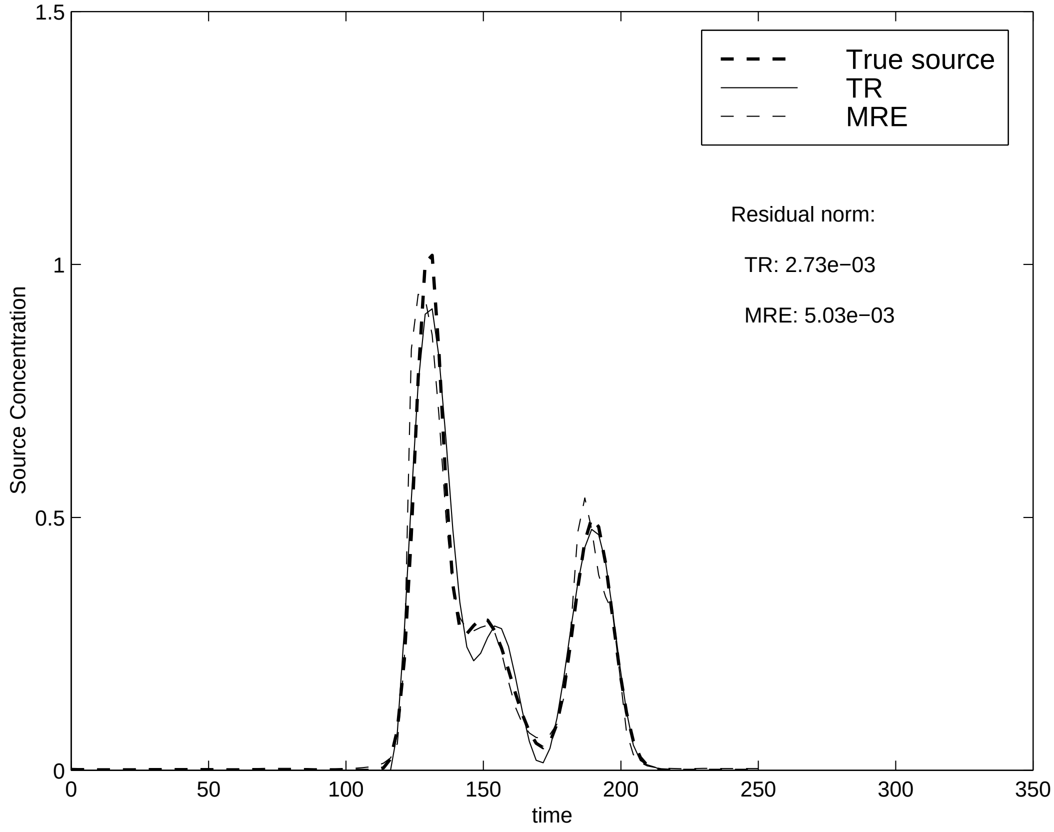

Figure 5.34: Tikhonov regularization and MRE results for multi-factor Run MF-16. Parameters are: smooth input function, complete sampling frequency, $\epsilon = 0.0$, $v = 1.0$, $D = 1.0$, and $T = 300$.

Table 5.4: Average difference of residual norms for runs with ideal and non-ideal values of each factor.

Factor	Average Difference of Residual Norms	
	Ideal	Non-ideal
1	-4.72×10^{-4}	1.23×10^{2}
2	-5.75×10^{-5}	1.19×10^{-2}
3	7.82×10^{-3}	3.98×10^{-3}
4	4.03×10^{-3}	7.77×10^{-3}
5	3.21×10^{-3}	8.58×10^{-3}

Tikhonov regularization performed better. The results show that the MRE method performs better, on average, than Tikhonov regularization whenever the non-ideal value of any factor is used. Tikhonov regularization only performs better than the MRE when data is exact, and when the plume is sampled completely.

The main effects for each factor are calculated by

$$e_k = \sum_{i=1}^{n} f_{k,i} R_i \ , \tag{5.3}$$

where e_k is the main effect of the k^{th} factor; n is the number of runs, $f_{k,i}$ is the level ("+" for the ideal value; "—" for the non-ideal value) of the k^{th} factor in the i^{th} run, and R_i is the response (residual norm) of the i^{th} run. The main effects are shown separately in Table 5.5 for the Tikhonov regularization results and the MRE results. Here, a negative value indicates that the residual norm of runs with the non-ideal factor is larger than the residual norm of runs with the ideal factor; i.e. the results are more accurate if the ideal value of the factor is used.

Table 5.5: Main effects of the multi-factor simulations.

Factor	TR	MRE
1	-1.03e-02	-3.88e-03
2	-8.01e-03	-2.06e-03
3	-5.66e-03	-7.58e-03
4	-2.05e-03	-1.81e-04
5	-2.57e-03	1.16e-04

The results show that, for each factor, the residual norms are lower when the ideal value is used, which is to be expected. The only exception is that the MRE method has smaller residual norms when a square input function is used instead of a smooth input function. The magnitudes of the main effects of the Tikhonov regularization results are larger than those of the MRE results, except for the velocity estimate (Factor 3). This indicates that, in general, the Tikhonov regularization is more sensitive than the MRE method to using non-ideal values. For Tikhonov regularization, the factor that has the largest effect on the residual norm is measurement error, followed in decreasing order by sampling frequency, velocity estimate, smoothness of the input function, and estimate of the dispersion coefficient. For the MRE method, the velocity estimate has the largest effect on the residual norm. The main effect of the dispersion estimate and of the smoothness of the input function are very low. These results show that the MRE method can perform equally well with a smooth or non-smooth input function; while the Tikhonov regularization method performs significantly worse with a non-smooth input function than with a smooth input function.

5.4 Summary of Effects of Factors

We compared the relative effectiveness of Tikhonov regularization and minimum relative entropy inversion in solving the source history reconstruction problem when complications exist. Some complications that commonly arise in field situation include measurement error, incomplete sampling of the plume, inaccurate estimates of parameters values, unknown source location, inability to accurately model all transport processes, spatial and temporal heterogeneity of the transport parameters, etc. In this research, we addressed three of these complications: measurement error, incomplete spatial sampling of the plume, and inaccurate estimates of transport parameters. We ran several simulations of the inverse methods to determine the effects of each of these complications on the reconstructed source history. We also considered two different types of input functions—smooth and non-smooth.

With perfect data, no complications, and a smooth input function, Tikhonov regularization performed slightly more accurately than the MRE method; However, when a non-smooth input function was used, the MRE method performed better than Tikhonov regularization. When undersampling of the plume was the only complication, Tikhonov regularization performed slightly better than the MRE method. When the transport parameters were incorrectly estimated, both methods performed at a similar level of accuracy. When measurement error was the only complication, the results of the two methods were approximately equivalent if the noise level was known exactly. In the MRE method, the noise level must be specified, or estimated if it is not known. If the noise level is underestimated, the MRE results are much worse than those of Tikhonov regularization.

Combinations of multiple complicating factors were addressed. These results show that, in general, the MRE method was more accurate than Tikhonov regularization under these circumstances. Note, however, that in the runs with measurement error, the noise level was assumed to be known exactly; thus eliminating the situation in which Tikhonov regularization performs much better than the MRE method. In addition, the results of the MRE method are sensitive to the value of the upper bound. In these simulations, we used an upper bound of 1.1, which is near the maximum of the true solution and therefore produces accurate results. If a larger upper bound were used, the results of the MRE method would be less accurate. Specifically, the results of the multi-factor simulations re-emphasize that the MRE method is more accurate than Tikhonov regularization when the source history is a non-smooth function.

Chapter 6

Conclusions

Determining the release history of a source of groundwater contamination is necessary in assessing liability for remediation costs. In many cases, the only available information concerning the contamination is the present spatial distribution of the contaminant concentration. Inverse methods can be used to reconstruct the release history from these measurements of the contaminant concentration.

The source history reconstruction problem that was evaluated in this research involves a point source of groundwater contamination at a known location in a one-dimensional, saturated, homogeneous porous medium. A known source history was input into a forward model to obtain the spatial concentration distribution of the contaminant at some later time. These data were sampled at discrete locations, and used in the inverse problem to reconstruct a discrete (in time) release history at the source. Several methods for solving this problem have been presented in the literature:

- Tikhonov regularization (Skaggs and Kabala, 1994)
- Method of quasi-reversibility, (Skaggs and Kabala, 1995)
- Minimum relative entropy inversion (Woodbury and Ulrych, 1996)
- Geostatistical approach (Snodgrass and Kitanidis, 1997)

Several controversial statements have been made recently regarding the strengths and limitations of using Tikhonov regularization and minimum relative entropy inversion to solve the source history reconstruction problem (Woodbury and Ulrych, 1996; Kabala and Skaggs, 1998; Woodbury and Ulrych, 1998b); however, the methods were not directly compared. The objective of this research was to provide a thorough and unbiased comparison of these two inverse methods, Tikhonov regularization and minimum relative entropy inversion, in solving the source history reconstruction problem.

Tikhonov regularization and minimum relative entropy inversion solve inverse problems defined by an ill-posed Fredholm integral equation of the first kind, by discretizing the integral equation into a matrix equation. Both methods use discrete measurements of concentration, along with an expression describing the physics of the process (assumed to be known exactly and modeled correctly), to reproduce an input function that matches the data to within some defined measurement error.

With Tikhonov regularization, the discretized Fredholm integral equation is replaced by a well-posed minimization problem whose solution is close to that of the original problem. In this research, we used the code, CONTIN, to solve the Tikhonov regularization problem. Skaggs and Kabala (1994) also used CONTIN; they provided us with their input files, so we were able to reproduce their results exactly.

With minimum relative entropy inversion, the model parameters to be estimated are treated as random variables. A probability density function is obtained for each model parameter, and the expected value of the model parameter, based on this probability density function, is chosen as the model solution. The posterior distribution is obtained by minimizing the entropy of the distribution relative to a prior distribution, while constraining the results to match the measured data. The prior distribution can be developed so that the mean value and the known bounds on the parameter values are included in the solution. We wrote a MATLAB program to implement the MRE method. To verify our program, we used it to solve the same problems that were solved by Woodbury and Ulrych (1996). We were able to reproduce most of their solutions; however our probability bounds were wider and we could not reproduce their pre-filtering approach to handling measurement error.

Both inverse methods contain some subjectivity. The stabilizer in Tikhonov regularization has two components—a regularization parameter and an operator matrix; the choice of these components is somewhat subjective. The regularization parameter describes the relative trade-off between matching the concentration measurements and stabilizing the problem. Several methods have been developed for selecting this optimal value. The choice of operator matrix defines the order of regularization, which specifies the feature of the input function that is minimized. With minimum relative entropy inversion, the selection of the prior expected value function and the upper bounds are subjective.

We evaluated the effects of the subjective parameters on the solution to the source history reconstruction problem. We found that the regularization order does affect the solution to the inverse problem. Zero–order regularization produces a solution with a very oscillatory structure; while, second–order regularization produces smoother and more accurate results. Since the true source history function is not likely to be oscillatory, second–order regularization is appropriate. We evaluated two methods for selecting the regularization parameter—the F-test method and generalized cross-validation. Although the value of the regularization parameter does affect the accuracy of the solution, the regularization parameter values chosen by the F-test and GCV methods were similar; and the resulting solutions to the inverse problem were also similar. Therefore, although the choice of regularization parameter selection method may be subjective, the results are fairly consistent.

We evaluated the effects of the prior expected value function and the upper bounds on the solutions obtained using the MRE method. We found that the solutions are relatively insensitive to the prior expected value functions, but more sensitive to the upper bound. The accuracy is poor when the upper bound is equal or approximately equal to the expected value function. If the expected value of a solution parameter is equal to the upper bound of its range, then the solution must be equal to the upper bound. This case is unrealistic, since it implies that we know the exact value of the solution parameter. If the expected value is very close to the upper bound, the probability density function shows a high probability that the solution is near the upper bound. If the true solution is not near the upper bound, then it will likely be overestimated. With the MRE method, we also found that the solution is most accurate when the upper bound is near the maximum value of the true source history function, and the accuracy decreases as the upper bound increases away from this value.

The main goal of this research was to compare the effectiveness of the two inverse method in solving the source history reconstruction problem. We evaluated the two methods under an ideal situation (perfect knowledge of all parameters except for the release history) and with some complications that are common in field situations (measurement error, incomplete spatial sampling of the plume, and inaccurate estimates of transport parameters). We considered two different types of input functions—smooth and non-smooth, and we ran several simulation of the inverse methods to determine the effects of each of these complications.

With perfect data, no complications, and a smooth input function, Tikhonov regularization performed slightly more accurately than the MRE method; However, when a non-smooth input function was used, the MRE method performed better than Tikhonov regularization. Second-order Tikhonov regularization produces a smooth solution which cannot capture the sharp rise and fall of the non-smooth function.

When measurement error was the only complication, the results of the two methods were approximately equivalent if the noise level was known exactly. In the MRE method, the noise level must be specified, or estimated if it is not known. When the noise level was underestimated, the MRE results were much worse than those of Tikhonov regularization. We evaluated two different error models for adding random noise to the data. In the first error model, the random measurement error was proportional to the true concentration (as in Skaggs and Kabala, 1994); and in the second error model, the random measurement error was independent of the true concentration (as in Woodbury and Ulrych, 1996). We found that the choice of error model does not affect the relative accuracy of the two inverse methods.

When undersampling of the plume was the only complication, Tikhonov regularization performed slightly better than the MRE method. When the plume is undersampled, the sampled data cannot provide information about the entire source history; therefore, only incomplete reconstruction of the source history is possible.

When transport parameters were incorrectly estimated, both methods performed at a similar level of accuracy. When velocity was overestimated, the reconstructed source history was shifted to later times and was more disperse than the true source history. The opposite effect occurred when the velocity was underestimated. When the dispersion coefficient was overestimated, the reconstructed source history was less disperse than the true source history; and the opposite effect occurred when the dispersion coefficient was underestimated.

We addressed combinations of multiple complicating factors. These results show that, in general, the MRE method was more accurate than Tikhonov regularization under these circumstances. Note, however, that in the runs with measurement error, the noise level was assumed to be known exactly; thus eliminating the situation in which Tikhonov regularization performs much better than the MRE method. In addition, the upper bound used in the MRE simulations was near the maximum value of the true solution; if a larger upper were used, the MRE results would be less accurate.

Overall, the results show that Tikhonov regularization and minimum relative entropy inversion produce similar results for many of the source history reconstruction problems evaluated. Two exceptions are that the MRE method performs better than Tikhonov regularization when the true source history is a non-smooth function; and if the noise level is underestimated in the MRE method, the results of Tikhonov regularization are more accurate than those of the MRE method. Although we can identify situations in which one method outperforms the other, the shape of the true source history function and the noise level in the measured data are not known in a practical situation; therefore, based on the information that is commonly available, we cannot determine which inverse method will perform better.

In all cases, the accuracy of the solutions decreases when additional uncertainty is added to the problem. Under ideal circumstances, both methods produce results that are nearly indistinguishable from the true solution. However, perfect knowledge of the system cannot be expected in practice. Uncertainty exists in the measured data, estimates of values of transport parameters, source location, models of transport processes, spatial and temporal variability of the transport parameters, the extent of the plume, etc. Both inverse methods are affected by this uncertainty, and the practitioner should be cautious of the results of either method.

References

[1] Dimri, V. 1992. *Deconvolution and Inverse Theory: Application to Geophysical Problems*, Elsevier Science, New York.

[2] Engl, H.W., M. Hanke, and A. Neubauer. 1996. *Regularization of Inverse Problems*, Kluwer Academic Publishers, Norwell, Massachusetts.

[3] Fetter, C.W. 1993. *Contaminant Hydrogeology*, Prentice-Hall, Inc., Upper Saddle River, New Jersey.

[4] Gill, P.E., W. Murray, and M.H. Wright. 1981. *Practical Optimization*, Academic Press, Inc., San Diego, California.

[5] Gorelick, S.M., B. Evans, and I. Remson. 1983. Identifying sources of groundwater pollution: an optimization approach. *Water Resources Research*, 19(3), 779–790.

[6] Groetsch, C.W. 1984. *The Theory of Tikhonov Regularization for Fredholm Integral Equations of the First Kind*, Pitman Publishing Inc., Marshfield, Massachusetts.

[7] Hansen, P.C. 1992. Numerical tools for analysis and solution of Fredholm integral equations of the first kind. *Inverse Problems*, 8, 849–872.

[8] Jaynes, E.T., 1957. Information theory and statistical mechanics. *Physical Review*, 106(4), 620–630.

[9] Johnson, R.W. and J.E. Shore. 1984. Relative- entropy minimization with uncertain constraints—Theory and application to spectrum analysis. *Report 5478*, Naval Research Laboratory, Washington, D.C.

[10] Kabala, Z.L. and T.H. Skaggs. 1998. Comment on "Minimum relative entropy inversion: Theory and application to recovering the release history of a groundwater contaminant" by Allan D. Woodbury and Tadeusz J. Ulrych, *Water Resources Research*, 34(8), 2077–2079.

[11] Kullback, S. 1959. *Information Theory and Statistics*, John Wiley, New York.

[12] Law, A.M. and W.D. Kelton. 1991. *Simulation Modeling and Analysis*, 2nd edition, McGraw-Hill, Inc., New York.

[13] Lukas, M.A. 1980. Regularization, in *The Application and Numerical Solution of Integral Equations*, edited by R.S. Anderson, F.R. de Hoog, and M.A. Lukas, Sijthoff and Noordoff, Germantown, Maryland.

[14] Obenchain, R.L. 1977. Classical F-tests and confidence regions for ridge regression. *Technometrics*, 19(4), pp. 429–439.

[15] Parker, R.L. 1994. *Geophysical Inverse Theory*, Princeton University Press, Princeton, New Jersey.

[16] Provencher, S.W. 1982a. A constrained regularization method for inverting data represented by linear algebraic or integral equation. *Computer Physics Communications*, 27, 213–227.

[17] Provencher, S.W. 1982b. CONTIN: A general purpose constrained regularization program for inverting noisy linear algebraic and integral equations, *Computer Physics Communications*, 22, 229–242.

[18] Provencher, S.W. 1984. *CONTIN (Version 2) Users Manual*, Technical Report EMBL-DA07, Max-Planck- Institut für biophysikalische Chemie, Göttingen, Federal Republic of Germany.

[19] Shannon, C.E. 1948. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27, 379–423, 623–659.

[20] Shore, J.E. and R.W. Johnson. 1980. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *IEEE Transactions on Information Theory*, IT-26(1), 26–37.

[21] Skaggs, T.H. and Z.L. Kabala. 1994. Recovering the release history of a groundwater contaminant. *Water Resources Research*, 30(1), 71–79.

[22] Skaggs, T.H. and Z.L. Kabala. 1995. Recovering the history of a groundwater contaminant plume: Method of quasi-reversibility. *Water Resources Research*, 31(11), 2669–2673.

[23] Snodgrass, M.R. and P.K. Kitanidis. 1997. A geostatistical approach to contaminant source identification, *Water Resources Research*, 33(4), 537–546.

[24] Sun, N.-Z. 1994. *Inverse Problems in Groundwater Modeling*, Kluwer Academic Publishers, Norwell, Massachusetts.

[25] Tikhonov, A.N. and V.Y. Arsenin. 1977. *Solutions of Ill-Posed Problems*, John Wiley, New York.

[26] Wagner, B.J. 1992. Simultaneous parameter estimation and contaminant source characterization for coupled groundwater flow and contaminant transport modelling, *Journal of Hydrology*, 135, 275–303.

[27] Wahba, G. 1977. Practical approximate solutions to linear operator equations when the data are noisy. *SIAM Journal on Numerical Analysis*, 14, 651–667.

[28] Woodbury, A.D., F. Render, and T. Ulrych. 1995. Practical probabilistic ground-water modeling. *Ground Water*, 33(4), 532–538.

[29] Woodbury, A.D., E. Sudicky, T.J. Ulrych, and R. Ludwig. 1998. Three-dimensional plume source reconstruction using minimum relative entropy inversion. *Journal of Contaminant Hydrology*, 32, 131–158.

[30] Woodbury, A.D. and T.J. Ulrych. 1993. Minimum relative entropy: Forward probabilistic modeling, *Water Resources Research*, 29(8), 2847–2860.

[31] Woodbury, A.D. and T.J. Ulrych. 1996. Minimum relative entropy inversion: Theory and application to recovering the release history of a groundwater contaminant, *Water Resources Research*, 32(9), 2671–2681.

[32] Woodbury, A.D. and T.J. Ulrych. 1998a. Minimum relative entropy and probabilistic inversion in groundwater hydrology, *Stochastic Hydrology and Hydraulics*, 12, 317–358.

[33] Woodbury, A.D. and T.J. Ulrych. 1998b. Reply, *Water Resources Research*, 34(8), 2081–2084.

Appendix A

Determining Lagrange Multipliers for the Prior Distribution

In Section 2.2.2, we obtained an expression for the prior distribution, $p(\mathbf{m})$, in terms of the Lagrange multipliers, μ and β_i, $i = 1, 2, \ldots, M$ (Equation 2.18):

$$p(\mathbf{m}) = b(\mathbf{m}) \exp\left[-1 - \mu - \sum_{i=1}^{M} \beta_i m_i\right] \;, \tag{A.1}$$

where M is the number of model parameters and $b(\mathbf{m})$ is the multivariate uniform distribution (Equation 2.13):

$$\begin{aligned} b(\mathbf{m}) &= \prod_{i=1}^{M} \frac{1}{U_i - L_i} \quad \text{for} \;\; L_i \leq m_i \leq U_i \\ b(\mathbf{m}) &= 0 \;\;\; \text{otherwise} \end{aligned} \tag{A.2}$$

where U_i is the upper bound of parameter m_i and L_i is the lower bound of m_i. In this appendix, we solve for the Lagrange multipliers, following the approach of Woodbury and Ulrych (1993).

To solve for the Lagrange multiplier, μ, we use the normalization requirement, i.e. $\int p(\mathbf{m}) d\mathbf{m} = 1$. Let $c = \exp(-1 - \mu) \prod_{i=1}^{M} (U_i - L_i)^{-1}$. With this substitution, Equation A.1 becomes

$$p(\mathbf{m}) = c \prod_{i=1}^{M} \exp(-\beta_i m_i) \;. \tag{A.3}$$

Integrating this expression over all $\mathbf{m}$ results in

$$c \int_{\mathbf{m}} \prod_{i=1}^{M} \exp(-\beta_i m_i) d\mathbf{m} = 1 \; . \tag{A.4}$$

Evaluating this integral gives

$$c \prod_{i=1}^{M} \left[-\frac{1}{\beta_i} \exp(-\beta_i U_i) + \frac{1}{\beta_i} \exp(-\beta_i L_i) \right] = 1 \; . \tag{A.5}$$

Solving for c and substituting the result into Equation A.3 gives the following expression for $p(\mathbf{m})$:

$$p(\mathbf{m}) = \prod_{i=1}^{M} \frac{\beta_i \exp(-\beta_i m_i)}{\exp(-\beta_i L_i) - \exp(-\beta_i U_i)} \; . \tag{A.6}$$

The previous equation is indeterminate if $\beta_i = 0$ for any i. For any i, Equation A.6 shows that

$$p(m_i) = \frac{\beta_i \exp(-\beta_i m_i)}{\exp(-\beta_i L_i) - \exp(-\beta_i U_i)} \; . \tag{A.7}$$

Taking the limit of this expression as $\beta_i \to 0$ gives

$$p(m_i) = \frac{1}{U_i - L_i} \; . \tag{A.8}$$

Therefore, the appropriate form of the prior distribution in terms of the Lagrange multipliers, β_i, is

$$\begin{aligned} p(m_i) &= \frac{\beta_i \exp(-\beta_i m_i)}{\exp(-\beta_i L_i) - \exp(-\beta_i U_i)} \;\; \text{for } \beta_i \neq 0 \\ p(m_i) &= \frac{1}{U_i - L_i} \;\; \text{for } \beta_i = 0 \; ; \\ p(\mathbf{m}) &= \prod_{i=1}^{M} p(m_i) \; . \end{aligned} \tag{A.9}$$

The values of β_i can be obtained from the expected value constraint (Equation 2.15):

$$\int_{\mathbf{m}} m_k p(\mathbf{m}) d\mathbf{m} = s_k \quad k = 1, 2, \ldots, M \tag{A.10}$$

where s_k is the expected value of model parameter m_k. By the normalization requirement, the integrals over m_i for $i \neq k$ evaluate to unity. Therefore, the expected value constraint simplifies to

$$\int_{L_i}^{U_i} m_i \frac{\beta_i \exp(-\beta_i m_i)}{\exp(-\beta_i L_i) - \exp(-\beta_i U_i)} dm_i = s_i \ , \tag{A.11}$$

for $i = 1, 2, \ldots, M$ and $\beta_i \neq 0$. Evaluating this integral, we obtain the final form of the expected value constraint:

$$\frac{-(\beta_i U_i + 1) \exp(-\beta_i U_i) + (\beta_i L_i + 1) \exp(-\beta_i L_i)}{\beta_i \left[\exp(-\beta_i L_i) - \exp(-\beta_i U_i)\right]} = s_i \ , \tag{A.12}$$

for $i = 1, 2, \ldots, M$. This equation can be solved numerically to obtain the values of the Lagrange multipliers, β_i. From Equations A.9 and A.11, we see that if $s_i = (U_i + L_i)/2$, then $\beta_i = 0$.

Appendix B

Determining Lagrange Multipliers for the Posterior Distribution

In Section 2.2.2, we obtained an expression for the posterior distribution, $q(\mathbf{m})$, in terms of the Lagrange multipliers, μ and λ_j, $j = 1, 2, \ldots, N$ (Equation 2.26):

$$q(\mathbf{m}) = cp(\mathbf{m}) \exp\left[-\sum_{j=1}^{N} \left(\lambda_j \sum_{i=1}^{M} g_{ji} m_i \right) \right] , \tag{B.1}$$

where $c = \exp(-1 - \mu)$, N is the number of data points, M is the number of model parameters, m_i are the model parameters, g_{ji} are the kernel function values, and $p(\mathbf{m})$ is the prior distribution (Equation 2.19):

$$\begin{aligned} p(m_i) &= \frac{\beta_i \exp(-\beta_i m_i)}{\exp(-\beta_i L_i) - \exp(-\beta_i U_i)} \text{ for } \beta_i \neq 0 \\ p(m_i) &= \frac{1}{U_i - L_i} \text{ for } \beta_i = 0 \; ; \\ p(\mathbf{m}) &= \prod_{i=1}^{M} p(m_i) \; , \end{aligned} \tag{B.2}$$

where β_i are Lagrange multipliers (whose values are obtained numerically), U_i is the upper bound of model parameter m_i, and L_i is the lower bound of m_i. In this appendix, we solve for the Lagrange multipliers, μ and λ_j, following the approach of Woodbury and Ulrych (1996).

To solve for the Lagrange multiplier, μ, we use the normalization requirement, i.e. $\int q(\mathbf{m}) d\mathbf{m} = 1$. Reversing the order of summation in Equation B.1 and substituting Equation B.2 for $p(\mathbf{m})$ (assuming that $\beta_i \neq 0$), we obtain

$$q(\mathbf{m}) = c\prod_{i=1}^{M} \frac{\beta_i \exp(-m_i a_i)}{\exp(-\beta_i L_i) - \exp(-\beta_i U_i)} \ , \tag{B.3}$$

where $a_i = \beta_i + \sum_{j=1}^{N} g_{ji}\lambda_j$. Integrating this expression over all $\mathbf{m}$ results in

$$c\int_{\mathbf{m}} \prod_{i=1}^{M} \frac{\beta_i \exp(-m_i a_i)}{\exp(-\beta_i L_i) - \exp(-\beta_i U_i)} \, d\mathbf{m} = 1 \ . \tag{B.4}$$

Evaluating this integral gives

$$c\prod_{i=1}^{M} \frac{\beta_i \left[\exp(-a_i L_i) - \exp(-a_i U_i)\right]}{a_i \left[\exp(-\beta_i L_i) - \exp(-\beta_i U_i)\right]} = 1 \ . \tag{B.5}$$

Solving for c and substituting the result into Equation B.3 gives the following expression for $q(\mathbf{m})$:

$$\begin{aligned} q(m_i) &= \frac{a_i \exp(-m_i a_i)}{\exp(-a_i L_i) - \exp(-a_i U_i)} \ \text{ for } a_i \neq 0 \\ q(m_i) &= \frac{1}{U_i - L_i} \ \text{ for } a_i = 0 \ ; \\ q(\mathbf{m}) &= \prod_{i=1}^{M} q(m_i) \ . \end{aligned} \tag{B.6}$$

This same result is obtained if $\beta_i = 0$, and $p(m_i) = 1/(U_i - L_i)$ is used.

The values of λ_j can be obtained from the data constraint (Equation 2.23):

$$\bar{d}_j = \int_{\mathbf{m}} q(\mathbf{m}) \sum_{i=1}^{M} g_{ji} m_i d\mathbf{m} \ , \tag{B.7}$$

where $\bar{d}_j$ is the value of the j^{th} measured data point. Reversing the order of summation and integration and evaluating the integral, we see that

$$\bar{d}_j = \sum_{i=1}^{M} g_{ji} \hat{m}_i(\boldsymbol{\lambda}) \ , \tag{B.8}$$

where $\hat{m}_i$ is the expected value of the model parameter m_i, and is a function of the Lagrange multipliers, $\boldsymbol{\lambda}$ ($\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_N]^T$, where T denotes transpose). Since the data is likely to contain measurement error, the equality in the previous equation does not hold. To account for measurement error, ϵ, we add the following constraint:

$$\sum_{j=1}^{N} \left[\bar{d}_j - \sum_{i=1}^{M} g_{ji} \hat{m}_i(\boldsymbol{\lambda}) \right]^2 \leq \xi^2 \epsilon^2 \ . \tag{B.9}$$

If measurement error is not proportional to the measured concentration (i.e. $C_{\text{meas}}(x_n, T) = C_{\text{exact}}(x_n, T) + \epsilon\delta_n$), then $\xi^2 = N$. If measurement error is proportional to the measured concentration, then $\xi^2 = ||\bar{\mathbf{d}}||^2$, where $||\cdot||$ denotes the L_2 norm.

With this constraint, Equation B.8 should be modified as (Johnson and Shore, 1984)

$$\bar{d}_j - \sum_{i=1}^{M} g_{ji} \hat{m}_i(\boldsymbol{\lambda}) = -\xi\epsilon \frac{\lambda_j}{||\boldsymbol{\lambda}||} \ , \tag{B.10}$$

where the errors are taken as independent and identically distributed with mean zero.

The values of the Lagrange multipliers, $\boldsymbol{\lambda}$, can be calculated using the Newton-Raphson method. From Equation B.10, we define

$$F(\boldsymbol{\lambda})_j = \bar{d}_j - \sum_{i=1}^{M} g_{ji} \hat{m}_i(\boldsymbol{\lambda}) + \xi\epsilon \frac{\lambda_j}{||\boldsymbol{\lambda}||} \ , \tag{B.11}$$

and use Newton-Raphson method to solve for the zeroes of $\mathbf{F}(\boldsymbol{\lambda})$ iteratively using:

$$\boldsymbol{\lambda}^k = \boldsymbol{\lambda}^{k-1} - \left(\left. \frac{\partial \mathbf{F}}{\partial \boldsymbol{\lambda}} \right|_{\boldsymbol{\lambda}^{k-1}} \right)^{-1} \mathbf{F}^{k-1} \ , \tag{B.12}$$

where the superscripts denote the iteration. The terms of the $\partial \mathbf{F}/\partial \boldsymbol{\lambda}$ matrix are

$$\frac{\partial F_j}{\partial \lambda_l} = -\sum_{i=1}^{M} g_{ji}\left[\frac{\partial \hat{m}_i}{\partial a_i} g_{li}\right] + \frac{\xi\epsilon}{||\boldsymbol{\lambda}||}\left[\delta_{jl} - \frac{\lambda_j \lambda_l}{||\boldsymbol{\lambda}||^2}\right] \ , \tag{B.13}$$

where $l = 1, 2, \ldots, N$, δ_{jl} is the Kronecker delta, and

$$\begin{aligned} \frac{\partial \hat{m}_i}{\partial a_i} &= \frac{a_i^2(U_i - L_i)^2 \exp\left[-a_i(U_i + L_i)\right] - \left[\exp(-a_i L_i) - \exp(-a_i U_i)\right]^2}{a_i^2\left[\exp(-a_i L_i) - \exp(-a_i U_i)\right]^2} \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{for } a_i \neq 0 \\ \frac{\partial \hat{m}_i}{\partial a_i} &= -\frac{(U_i - L_i)^2}{12} \ \text{ for } a_i = 0 \ . \end{aligned} \tag{B.14}$$

The iterations are carried out until a reasonable tolerance is met. To solve these equations numerically, several asymptotic approximations must be made; these approximations are described in Chapter 3.

Appendix C

Minimum Relative Entropy Inversion Program

We wrote a MATLAB program to implement the minimum relative entropy inversion method. The implementation was described in Chapters 2 and 3, and in Appendices A and B. The source code for the MATLAB program is included in this Appendix. We also include an overview of the program and a table relating the program variable names to those used in this thesis.

C.1 Program Overview

The program was written as a MATLAB function. The user calls the function, and passes in the necessary parameter values, and the function returns the results. The main function is `domre.m`, which calls many other functions to implement the MRE method. The input to the function includes:

- the name of a MATLAB function that calculates the kernel
- array of sample locations
- array of sampled concentrations
- sampling time
- array of upper bounds for the prior distribution
- array of prior expected values

- transport parameters, v and D
- time array for the reconstructed source function
- noise level
- true source history, if available
- lower and upper limits of interval for bisection method for β_i calculations
- stopping tolerance for β_i calculations
- maximum number of iterations for β_i calculations
- stopping tolerance for $\boldsymbol{\lambda}$ calculations
- maximum number of iterations for $\boldsymbol{\lambda}$ calculations
- stopping tolerance for golden section search
- cutoff value for asymptotic approximations near zero
- cutoff value for asymptotic approximations near $\pm\infty$
- flag specifying plot options

The output from the function includes the reconstructed source history, the model fit to the measured data, the values of the Lagrange multipliers, and the 5^{th} and 95^{th} percentile probability levels. During each iteration of Newton's method in the $\boldsymbol{\lambda}$ calculations, the following information is displayed in the command window: the iteration number, the condition number of the Jacobian matrix, the optimal step length from the golden section search, the norm of **F** (See Equation B.11), and the value of the stopping criteria. In addition, the user can select to display plots at each iteration of Newton's method ($\boldsymbol{\lambda}$ calculations). The items that can be plotted are the true and reconstructed source history functions, the measured and fitted data, the values of a_i ($a_i = \beta_i + \sum_{j=1}^{N} \lambda_j g_{ji}$), and the Lagrange multipliers, $\boldsymbol{\lambda}$.

The functions called by `domre.m` and its subfunctions include:

- `bisection.m` – calculates the values of the Lagrange multipliers, β_i, using bisection method
- `getrep.m` – calculates the values of the kernel matrix, **G**
- `donewton.m` – calculates the values of the Lagrange multipliers, $\boldsymbol{\lambda}$, using Newton's method, and the model solution, $\hat{\mathbf{m}}$.
- `snfcn.m` – evaluates the function whose zero will be found using the bisection method
- `kernel.m` – evaluates the value of the kernel function (Equation 3.2)
- `getfk.m` – evaluates the vector **F** (Equation B.11)
- `getjac.m` – evaluates the Jacobian matrix (Equation B.13)
- `plotops.m` – creates plots of information at each Newton's method iteration in the $\boldsymbol{\lambda}$ calculations
- `rscale.m` – performs row scaling
- `golden.m` – performs the golden section search

C.2 Source Code

```
%%%%%%%%%% start of function domre.m %%%%%%%%%%
%
% Function to solves an inverse problem using Minimum Relative
%     Entropy Inversion
%
% Written by: Roseanna M. Neupauer
% Modification Date: April 24, 1999
%
% [csource,lambda,beta,fittedc,p5,p95] = ...
%     domre(kernfn,xsample,csample,tsample,upper,expvalue,...
%     params,t,noise,cin,leftbegin,rightbegin,tolbeta,...
%     maxiter,tollam,lamiter,tolls,nearzero,large,pflag);
%
% Inputs
%     kernfn       name of matlab function that calculates the
%                    value of the kernel
%                    This function is called with the
%                    following command:
%                         y=kernfn(xx,tt,params)
%                    where xx is a location and tt is a time
%     xsample      array containing sampling locations
%     csample      array containing sampled concentrations
%     tsample      time of sampling
%     upper        array containing upper limit of prior
%                    distributions
%     expvalue     array containing prior expected value
%     params       array of all transport parameters values
%     t            array containing solution times
%     noise        standard deviation of normally-distributed
%                    random noise in measurements
%                    for noise > 0, noise is absolute
%                    for noise < 0, the absolute value of
%                         noise is proportional to
%                         sample concentration
%     cin          array containing true source history
%     leftbegin    lower limit of range for bisection method
%     rightbegin   upper limit of range for bisection method
%     tolbeta      tolerance for beta-fitting
%     maxiter      maximum number of iterations in beta-fitting
%     tollam       tolerance for lambda-fitting
%     lamiter      maximum number of iterations in
%                    lambda-fitting
%     tolls        tolerance for golden section search
%     nearzero     value below which the asymptotic
%                    approximation to zero is used
%     large        value above which the asymptotic
%                    approximation to infinity is used
%     pflag            plot flag (0-15)
%                      pflag = sum of the following
%                        1 - to plot true source history and
%                             fitted source history after
%                             each iteration
%                        2 - to plot measured data and fitted
%                             data after each iteration
%                        4 - to plot a vector at each iteration
%                        8 - to plot lambda vector at each
%                             iteration
%
% Outputs
%     csource          array containing reconstructed solution of
%                        source history
%     lambda           array containing the Lagrange multipliers
%     beta             array containing the Lagrange multipliers
%     fittedc          array containing the model fit to the
%                        measured data
%     p5               array containing the 5th percentile
%                        probability level
%     p95              array containing the 95th percentile
%                        probability level
%
% Functions called
%     bisection        calculates the values of the Lagrange
%                        multipliers, beta, using bisection method
%     getrep           calculates the values of the kernel matrix
%     donewton         calculates the values of the Lagrange
%                        multipliers, lambda, using Newton's
%                        method, and, in the process, calculates
%                        the expected value of the posterior
%                        distribution

function [csource,lambda,beta,fittedc,p5,p95] = ...
     domre(kernfn,xsample,csample,tsample,upper,expvalue,...
     params,t,noise,cin,leftbegin,rightbegin,tolbeta,...
     maxiter,tollam,lamiter,tolls,nearzero,large,pflag);

% define size variables
     nt=size(t,2);
     ndata=size(csample,1);

% calculate betas using bisection method

     beta=bisection(upper,expvalue,leftbegin,rightbegin,...
          maxiter,tolbeta,nt,nearzero,large);

% discretize representers
     g=getrep(kernfn,t,xsample,tsample,params);
```

```
% use Newton-Raphson method to calculate values of Lagrange
% multipliers, lambda, and, in the process, the expected value
% of the distribution

     [lambda,fittedc,csource,a]=...
          donewton('getfk','getjac',lamiter,tollam,tolls,...
          xsample,g,upper,t,cin,noise,csample,ndata,beta,...
          nearzero,large,pflag);

% calculate the probability levels of the distribution
% (page 36)
     p5=-log(0.05 .*(exp(-a.*upper)-1)+1)./a;
     p95=-log(0.95 .*(exp(-a.*upper)-1)+1)./a;
     alist=find(a < nearzero);
     p5(alist)=0.05*upper(alist);
     p95(alist)=0.95*upper(alist);

%%%%%%%%%%  end of function domre.m  %%%%%%%%%%
```

```
%%%%%%%%%% start of function bisection.m %%%%%%%%%%
%
% Function to calculate the values of the Lagrange
% multipliers, beta, using bisection method
%
% Written by: Roseanna M. Neupauer
% Modification Date: April 24, 1999
%
% function beta=bisection(upper,expvalue,leftbegin,...
%     rightbegin,maxiter,tolbeta,nt,nearzero,large)
%
% Inputs
%     upper        array containing upper limit of prior
%                    distributions
%     expvalue     array containing prior expected value
%     leftbegin    lower limit of range for bisection method
%     rightbegin   upper limit of range for bisection method
%     maxiter      maximum number of iterations in beta-fitting
%     tolbeta      tolerance for beta-fitting
%     nt           number of elements in the time array
%     nearzero     value below which the asymptotic
%                    approximation to zero is used
%     large        value above which the asymptotic
%                    approximation to infinity
%
% Outputs
%     beta         array containing the Lagrange multipliers
%
% Functions called
%     snfcn        evaluates the function whose zero will be
%                    found using the bisection method

function beta=bisection(upper,expvalue,leftbegin,rightbegin,...
          maxiter,tolbeta,nt,nearzero,large)

beta=zeros(nt,1);

for i=1:nt
     u=upper(i);
     sn=expvalue(i);
     left=leftbegin;
     right=rightbegin;
     for j=1:maxiter
          bold=(left+right)/2;
          snfcnl=snfcn(left,u,sn,nearzero,large);
          snfcnr=snfcn(right,u,sn,nearzero,large);
          snfcnb=snfcn(bold,u,sn,nearzero,large);
          if (abs(2*sn - u)<eps)
               beta(i)=0.;
               break
          elseif (snfcnl < 0 & snfcnr < 0)
               if (snfcnl > snfcnr)
                    beta(i)=leftbegin;
               else
                    beta(i)=rightbegin;
               end
               break
          elseif (snfcnl > 0 & snfcnr > 0)
               if (snfcnl > snfcnr)
                    beta(i)=rightbegin;
               else
                    beta(i)=leftbegin;
               end
               break
          else
               if (snfcnb*snfcnl) > 0
                    left=bold;
               else
                    right=bold;
               end
               if (right-left) < tolbeta
                    beta(i)=(left+right)/2;
                    break
               else
                    if (j == maxiter)
                      disp(’Beta iterations did not converge’)
                      beta(i)=(right+left)/2;
                    end
               end
          end
     end % for j
     if (beta(i) == leftbegin & u-sn ~= 0)
          beta(i)=-1/(u-sn);
     end
end % for i
%%%%%%%%%%  end of function bisection.m  %%%%%%%%%%
```

```
%%%%%%%%%% start of function getrep.m %%%%%%%%%%
%
% Function to evaluate the kernel matrix
%
% Written by: Roseanna M. Neupauer
% Modification Date: April 24, 1999
%
% f=getrep(kernfn,t,xsample,tsample,params);
%
% Inputs
%     kernfn       name of matlab function that calculates the
%                    value of the kernel
%                    This function is called with the
%                    following command:
%                         y=kernfn(xx,tt,params)
%                    where xx is a location and tt is a time
%     t            array containing solution times
%     xsample      array containing sampling locations
%     tsample      time of sampling
%     params       array of all transport parameters values
%
% Outputs
%     f            matrix of kernel function values, scaled by
%                    time interval
%
% Functions called
%     kernfn       calculated the value of the kernel at one
%                    location and time

function f=getrep(kernfn,t,xsample,tsample,params);

nt=size(t,2);
ns=size(xsample,2);
weight=t(2)-t(1);
f=zeros(ns,nt);

% G matrix in Equation 3.3
for i=1:nt
     for j=1:ns
          if (t(i) <= tsample) ...
               f(j,i)=weight*...
               feval(kernfn,xsample(j),tsample-t(i),params);
          end
     end % for j
end     % for i

return
%%%%%%%%%%  end of function getrep.m  %%%%%%%%%%
```

```
X%%%%%%%%%% start of function donewton.m %%%%%%%%%%
%
% Function to calculate the Lagrange multipliers, lambda,
% using Newton's method
%
% Written by: Roseanna M. Neupauer
% Modification Date: April 24, 1999
%
% [lambda,fittedc,mhat,a]=...
%          donewton(funfcn,jacfcn,lamiter,tollam,tolls,...
%          xsample,g,upper,t,cin,noise,csample,ndata,beta,...
%          nearzero,large,pflag);
%
% Inputs
%     funfcn       name of matlab function that calculates the
%                    F vector (Equation 3.10) funfcn='getfk'
%     jacfcn       name of matlab function that calculates the
%                    Jacobian matrix (Equation B.13)
%                    funfcn='getjac'
%     lamiter      maximum number of iterations in
%                    lambda-fitting
%     tollam       tolerance for lambda-fitting
%     tolls        tolerance for golden section search
%     xsample      array containing sampling locations
%     g            matrix of scaled kernel functions
%     upper        array containing upper limit of prior
%                    distributions
%     t            array containing solution times
%     cin          array containing true source history
%     noise        standard deviation of normally-distributed
%                    random noise in measurements
%                    for noise > 0, noise is absolute
%                    for noise < 0, the absolute value of
%                         noise is proportional to
%                         sample concentration
%     csample      array containing sampled concentrations
%     ndata        number of measured data points
%     beta         array of Lagrange multipliers
%     nearzero     value below which the asymptotic
%                    approximation to zero is used
%     large        value above which the asymptotic
%                    approximation to infinity is used
%     pflag        plot flag (0-15)
%                  pflag = sum of the following
%                    1 - to plot true source history and
%                         fitted source history after
%                         each iteration
%                    2 - to plot measured data and fitted
%                         data after each iteration
%                    4 - to plot a vector at each iteration
%                    8 - to plot lambda vector at each
%                         iteration
%
% Outputs
%     lambda       array containing the Lagrange multipliers
%     fittedc      array containing the model fit to the
%                    measured data
%     mhat         array containing the expected value of the
%                    model, based on the posterior distribution
%     a            array containing the vector, a
%
% Functions called
%     getfk        calculates the F vector
%     getjac       calculates the Jacobian matrix
%     plotops      creates some intermediate plots
%     rscale       row scales the Jacobian
%     golden       performs a golden section search

function [lambda,fittedc,mhat,a]=...
          donewton(funfcn,jacfcn,lamiter,tollam,tolls,...
          xsample,g,upper,t,cin,noise,csample,ndata,beta,...
          nearzero,large,pflag);

k=0;
fignum=1;
normcs=norm(csample);
lambda=ones(ndata,1);

for i=1:lamiter

% calculate the F matrix (Equation 3.10)
     fprintf('Newton method: ITERATION %i\n',i)
     [fe,mhat,a]=feval(funfcn,lambda,g,beta,upper,csample,...
          noise,nearzero,large);

% display some plots
     if pflag
          [k,fignum]=plotops(pflag,k,fignum,t,mhat,cin,i,...
               lambda,csample,noise,ndata,xsample,a,fe,large);
     end

% calculate the Jacobian matrix (Equation B.13)
     dfe=feval(jacfcn,lambda,g,upper,a,beta,csample,noise,...
               nearzero,large);

% perform row scaling on the Jacobian matrix (page 59)
     [dfe,fe]=rscale(dfe,fe);
     fprintf('   Condition number of Jacobian: %e \n',cond(dfe))
```

```
% solve the matrix equation to obtain step direction
% (Equations 3.13 and B.12)
     dellame=-dfe\fe;

% Use Golden section search to find the optimal step length
% (page 59)
     alpha=golden(funfcn,lambda,dellame,tolls,g,beta,upper,...
          csample,noise,nearzero,large);
     fprintf('   Step length is %f\n',alpha);

     lambda=lambda+alpha*dellame;
     [fe,mhat,a]=feval(funfcn,lambda,g,beta,upper,...
          csample,noise,nearzero,large);
        fprintf('   new norm(F) is %e \n',norm(fe));

% check if convergence criteria is met (page 59)
     ratio=norm(fe)/(1+norm(csample));
     fprintf('   Stopping criteria is %e \n',ratio);
     if ratio < tollam | isnan(ratio)
% calculate the fitted plume data
          norml=norm(lambda);
          if (noise > 0)
               fittedc=-fe+csample+sqrt(ndata)*noise*...
                    lambda/norml;
          else
               fittedc=-fe+csample+normcs*abs(noise)*...
                    lambda/norml;
          end
          break;
     end
end

return
%%%%%%%%%%  end of function donewton.m  %%%%%%%%%%
```

```
%%%%%%%%%% start of function snfcn.m %%%%%%%%%%
%
% Function to evaluate the equations whose zero will be found
%     using the bisection method
%
% Written by: Roseanna M. Neupauer
% Modification Date: April 24, 1999
%
% function f=snfcn(beta,u,sn,nearzero,large);
%
% Inputs
%     beta         estimation of the ith beta
%     u            ith upper value
%     sn           ith expected value
%     nearzero     value below which the asymptotic
%                    approximation to zero is used
%     large        value above which the asymptotic
%                    approximation to infinity
%
% Outputs
%     f            value of function whose zero will be found
%                    using the bisection method

function f=snfcn(beta,u,sn,nearzero,large);

bu=beta*u;
if bu == 0

     % Equation 3.6
     f=sn-u/2;

elseif (abs(bu) < nearzero)

     % Equation 3.7
     f=sn-(u*(12.-8.*bu+3.*bu^2)/(24.-12.*bu+4.*bu^2-bu^3));

elseif bu < -large

     % page 56
     f=sn-(u+1/beta);

elseif (bu > large)

     % page 57
     f=sn-1/beta;

else

     % Equation 3.5 and Equation A.12
     f=sn-((exp(-bu).*(-bu-1)+1)/(beta.*(1-exp(-bu))));

end

return
%%%%%%%%%%  end of function snfcn.m  %%%%%%%%%%
```

```
%%%%%%%%%% start of function kernel.m %%%%%%%%%%
%
% Function to evaluate the kernel for one x and one time
%
% Written by: Roseanna M. Neupauer
% Modification Date: April 24, 1999
%
% f=kernel(xx,tt,params)
%
% Inputs
%     xx           location
%     tt           difference between sample time and
%                    calculation time
%     params       array of all transport parameters values
%
% Outputs
%     f          value of kernel function at xx,tt
%

function f=kernel(xx,tt,params)

% Equation 3.2
f=xx./(2*sqrt(pi*params(2)*tt.^3)).*...
     exp(-(xx-params(1).*tt).^2./(4*params(2)*tt));

return
%%%%%%%%%%  end of function kernel.m  %%%%%%%%%%
```

```
%%%%%%%%%% start of function getfk.m %%%%%%%%%%
%
% Function to calculate the F vector (Equation 3.10)
%
% Written by: Roseanna M. Neupauer
% Modification Date: April 24, 1999
%
% [fe,mhat,a]=getfk(lambda,g,beta,upper,csample,noise,...
%          nearzero,large)
%
% Inputs
%     lambda       array containing the Lagrange multipliers
%     g            matrix of scaled kernel functions
%     beta         array of Lagrange multipliers
%     upper        array containing upper limit of prior
%                    distributions
%     csample      array containing sampled concentrations
%     noise        standard deviation of normally-distributed
%                    random noise in measurements
%                    for noise > 0, noise is absolute
%                    for noise < 0, the absolute value of
%                         noise is proportional to
%                         sample concentration
%     nearzero     value below which the asymptotic
%                    approximation to zero is used
%     large        value above which the asymptotic
%                    approximation to infinity is used
%
% Outputs
%     fe           array containing the F vector
%     mhat         array containing the expected value of the
%                    model, based on the posterior distribution
%     a            array containing the vector, a

function [fe,mhat,a]=getfk(lambda,g,beta,upper,csample,...
     noise,nearzero,large)

% define array sizes and create new arrays
[ndata,nt]=size(g);
fe=zeros(ndata,1);
b=zeros(nt,1);
a=zeros(nt,1);
mhat=zeros(nt,1);
blarge=100.;

% calculate the a vector (page 35)
for k=1:nt
     aa=0.;
     for l=1:ndata
```

```
          aa=aa+lambda(l)*g(l,k);
     end
     a(k)=beta(k)+aa;
     b(k)=aa;
end

% calculate mhat
mau=a.*upper;
for j=1:nt
     if mau(j) == 0.

          % Equation 2.28
          mhat(j)=upper(j)/2;

     elseif (beta(j) < -blarge)

          % Equation 3.11
          mhat(j)=2*upper(j)/3-1/(3*beta(j))*(1-upper(j)*...
               (beta(j)+b(j)))+...
               2/(9*beta(j)^2)*(-beta(j)-b(j)+upper(j)*b(j)*...
               (beta(j)+b(j)/2));

     elseif (abs(mau(j)) < nearzero)
          au=a(j)*upper(j);

          % Equation 3.12
          mhat(j)=upper(j)*(12.-8.*au+3.*au^2)/...
               (24.-12.*au+4.*au^2-au^3);

     elseif mau(j) < -1*10^2

          % page 59
          mhat(j)=upper(j)+1/a(j);

     elseif (mau(j) > 10^4)

          % page 59
          mhat(j)=1/a(j);

     else

          % Equation 2.28
          mhat(j)=(exp(-mau(j)).*(-mau(j)-1)+1)./...
               (a(j).*(1-exp(-mau(j))));

     end
end

% calculate the F matrix
```

```
if noise > 0
     errfactor=sqrt(ndata)*noise/norm(lambda);
else
     errfactor=norm(csample)*abs(noise)/norm(lambda);
end
% Equations 3.10 and B.11
for k=1:ndata
     ff=csample(k);
     for l=1:nt
          ff=ff-g(k,l)*mhat(l);
     end
     fe(k)=ff+errfactor*lambda(k);
end

return
%%%%%%%%%%  end of function getfk.m  %%%%%%%%%%
```

```
%%%%%%%%%% start of function getjac.m %%%%%%%%%%
%
% Function to calculate the Jacobian matrix (Equation B.13)
%
% Written by: Roseanna M. Neupauer
% Modification Date: April 24, 1999
%
% dfe=getjac(lambda,g,upper,a,beta,csample,noise,nearzero,...
%     large);
%
% Inputs
%     lambda       array containing the Lagrange multipliers
%     g            matrix of scaled kernel functions
%     upper        array containing upper limit of prior
%                    distributions
%     a            array containing the vector, a
%     beta         array of Lagrange multipliers
%     csample      array containing sampled concentrations
%     noise        standard deviation of normally-distributed
%                    random noise in measurements
%                    for noise > 0, noise is absolute
%                    for noise < 0, the absolute value of
%                         noise is proportional to
%                         sample concentration
%     nearzero     value below which the asymptotic
%                    approximation to zero is used
%     large        value above which the asymptotic
%                    approximation to infinity is used
%
% Outputs
%     dfe          array containing the Jacobian matrix

function dfe=getjac(lambda,g,upper,a,beta,csample,noise,...
       nearzero,large);

% calculate array sizes and define new arrays
[ndata,nt]=size(g);
dfe=zeros(ndata);
dmndan=zeros(nt,1);
b=zeros(nt,1);
norml=norm(lambda);
blarge=100.;

% calculate the b vector (page 58)
for k=1:nt
     aa=0.;
     for l=1:ndata
          aa=aa+lambda(l)*g(l,k);
     end
     b(k)=aa;
end

% calculate the derivative of mhat with respect to a or b
mau=a.*upper;
for j=1:nt
     if beta(j) < -blarge

          % Equation 3.17
          dmndan(j)=5*upper(j)/(9*beta(j))-...
               2*(1-2*b(j)*upper(j))/(9*beta(j)^2);

     elseif abs(mau(j)) < nearzero
          au=a(j)*upper(j);

          % Equation 3.15
          dmndan(j)=-upper(j)^2*(15.-15.*au+8*au^2)/...
               (180.-180*au+105*au^2);

     elseif abs(mau(j)) > large

          % Equation 3.16
          dmndan(j)=-a(j)^(-2);

     else

          % Equations 3.14 and B.14
          dmndan(j)=((a(j)^2)*(upper(j)^2)*exp(-mau(j))...
               -1+2*exp(-mau(j))-exp(-2*mau(j)))/...
               ((a(j)^2)*(1-exp(-mau(j)))^2);

     end
end

% calculate the derivative of F with respect to lambda

if noise > 0
     errfactor=sqrt(ndata)*noise/norml;
else
     errfactor=norm(csample)*abs(noise)/norml;
end
% Equation B.13
for j=1:ndata
     for l=1:ndata
          fl=0;
          for n=1:nt
               fl=fl-g(j,n)*dmndan(n)*g(l,n);
          end
          dfe(j,l)=fl-errfactor*lambda(j)*...
```

```
               lambda(l)/(norml^2);
     end
end
dfe=dfe+errfactor*eye(ndata);

return
%%%%%%%%%%  end of function getjac.m  %%%%%%%%%%
```

```
%%%%%%%%%% start of function plotops.m %%%%%%%%%%
%
% Function to create intermediate plots during each
%     iteration of Newton's method
%
% Written by: Roseanna M. Neupauer
% Modification Date: April 24, 1999
%
% [k,fignum]=plotops(pflag,k,fignum,t,mhat,cin,i,lambda,...
%               csample,noise,ndata,xsample,a,fe,large)
% Inputs
%     pflag        plot flag (0-15)
%                  pflag = sum of the following
%                    1 - to plot true source history and
%                         fitted source history after
%                         each iteration
%                    2 - to plot measured data and fitted
%                         data after each iteration
%                    4 - to plot a vector at each iteration
%                    8 - to plot lambda vector at each
%                         iteration
%     k            plot number on plot of true solution and
%                    fitted solution
%     fignum       figure identifier on plot of true solution
%                    and fitted solution
%     t            array containing solution times
%     mhat         array containing the expected value of the
%                    model, based on the posterior distribution
%     cin          array containing true source history
%     i            iteration number
%     lambda       array containing the Lagrange multipliers
%     csample      array containing sampled concentrations
%     noise        standard deviation of normally-distributed
%                    random noise in measurements
%                    for noise > 0, noise is absolute
%                    for noise < 0, the absolute value of
%                         noise is proportional to
%                         sample concentration
%     ndata        number of measured data points
%     xsample      array containing sampling locations
%     a            array containing the vector, a
%     fe           array containing the F vector
%     large        value above which the asymptotic
%                     approximation to infinity is used
%
% Outputs
%     k            plot number on plot of true solution and
%                    fitted solution
%     fignum       figure identifier on plot of true solution
%                    and fitted solution

function [k,fignum]=plotops(pflag,k,fignum,t,mhat,cin,i,...
               lambda,csample,noise,ndata,xsample,a,fe,large)

% plot user-selected plots
     if mod(pflag,2)

%     plot true and fitted solution at each iteration

          k=k+1;
          funstr=strcat(’subplot(33’,num2str(k),’)’);
          if k == 1
               figure(fignum)
               subplot(111)
               clf
               fignum=fignum+1;
          elseif k == 9
               k=0;
          end
          eval(funstr)
          if ~isempty(cin)
               plot(t,mhat,t,cin,’--’)
          else
               plot(t,mhat)
          end
          xlabel(’time’),ylabel(’Source Concentration’);
          text(240,1.2,num2str(i-1))
          pause(2)
     end

     if (mod(pflag,4) >=2)

%     plot fitted and measured data at each iteration

          figure(10)
          norml=norm(lambda);
          normcs=norm(csample);
          if (noise > 0)
               fittedc=-fe+csample+sqrt(ndata)*noise*...
                    lambda/norml;
          else
               fittedc=-fe+csample+normcs*abs(noise)*...
                    lambda/norml;
          end
          plot(xsample’,csample,’o’,xsample’,fittedc,’V’)
          legend(’Sampled Data’,’Fitted Data’,5)
          text(50,.352,num2str(i-1))
          pause(2)
     end

     if (mod(pflag,8) >= 4)

%     plot a vector at each iteration

          figure(20)
          tp=find(a > 0);
          tn=find(a < 0);
          semilogy(t,abs(a),':',t(tp),a(tp),'o',...
               t(tn),-a(tn),'V',t,large*ones(size(t)))
          xlabel('Time')
          ylabel('values of a')
          legend('Absolute a','Positive a','Negative a',...
               'Asymptotic Cutoff',5)
          pause(2)
     end

     if (pflag >= 8)

%      plot lambdas

          figure(30)
          nl=size(lambda,1);
          tp=find(lambda > 0);
          tn=find(lambda < 0);
          tt=linspace(1,nl,nl)';
          semilogy(tt,abs(lambda),':',tt(tp),lambda(tp),'o',...
               tt(tn),-lambda(tn),'V')
          xlabel('x')
          ylabel('values of lambda')
          legend('Absolute lambda','Positive lambda',...
               'Negative lambda',5)
          pause(2)

     end
     figure(fignum-1)

return
%%%%%%%%%%  end of function plotops.m  %%%%%%%%%%
```

```
%%%%%%%%%% start of function rscale.m %%%%%%%%%%
%
% Function to perform row scaling of Jacobian matrix
%
% Written by: Roseanna M. Neupauer
% Modification Date: April 24, 1999
%
% [Aout, vout]=rscale(Ain,vin)
%
% Inputs
%     Ain          unscaled Jacobian matrix
%     vin          unscaled F vector
%
% Outputs
%     Aout         scaled Jacobian matrix
%     vout         scaled F vector

function [Aout, vout]=rscale(Ain,vin)

[nr,nc]=size(Ain);
maxval=max(Ain,[],2);
for i=1:nr
     Aout(i,:)=Ain(i,:)/maxval(i);
end
vout=vin./maxval;

return
%%%%%%%%%%  end of function rscale.m  %%%%%%%%%%
```

```
%%%%%%%%%% start of function golden.m %%%%%%%%%%
%
% Function to perform a Golden section search to calculate
%           optimal step length
%
% Written by: Roseanna M. Neupauer
% Modification Date: April 24, 1999
%
% alpha=golden(funfcn,lambda,dellame,tolls,g,beta,upper,...
%          csample,noise,nearzero,large)
% Inputs
%     funfcn       name of matlab function that calculates the
%                    F vector (Equation 3.10) funfcn='getfk'
%     lambda       array containing the Lagrange multipliers
%     dellame      step direction
%     tolls        tolerance for golden section search
%     g            matrix of scaled kernel functions
%     beta         array of Lagrange multipliers, beta
%     upper        array containing upper limit of prior
%                    distributions
%     csample      array containing sampled concentrations
%     noise        standard deviation of normally-distributed
%                    random noise in measurements
%                    for noise > 0, noise is absolute
%                    for noise < 0, the absolute value of
%                         noise is proportional to
%                         sample concentration
%     nearzero     value below which the asymptotic
%                    approximation to zero is used
%     large        value above which the asymptotic
%                    approximation to infinity is used
%
% Outputs
%     alpha        optimal step length
%
% Functions called
%     getfk        calculates the F vector

function alpha=golden(funfcn,lambda,dellame,tolls,g,beta,...
          upper,csample,noise,nearzero,large)

% bracket the step length to within a .1-length interval
normdel=ones(11,1);
for brack=1:11
     del=0.1*(brack-1);
     [newfe,dummy,dummy1]=feval(funfcn,lambda+del*dellame,...
          g,beta,upper,csample,noise,nearzero,large);
     normdel(brack)=norm(newfe);
end
```

```
minbrack=find(normdel==min(normdel));
minbrack=minbrack(1);
if (minbrack >=  10)
     aa=0.9;
     bb=1.0;
elseif (minbrack <= 2)
     aa=0.;
     bb=0.1;

else
     aa=0.1*(minbrack-1);
     bb=0.1*(minbrack+1);
end

% Using the 0.1-length interval as a starting interval,
%     perform a Golden section search in the step direction

[newfe,dummy,dummy1]=feval(funfcn,lambda+aa*dellame,g,beta,...
     upper,csample,noise,nearzero,large);
fa=norm(newfe);
[newfe,dummy,dummy1]=feval(funfcn,lambda+bb*dellame,g,beta,...
     upper,csample,noise,nearzero,large);
fb=norm(newfe);

tau=2/(1+sqrt(5));
F1=aa+(1-tau)*(bb-aa);
F2=aa+tau*(bb-aa);
[newfe,dummy,dummy1]=feval(funfcn,lambda+F1*dellame,g,beta,...
     upper,csample,noise,nearzero,large);
fF1=norm(newfe);
[newfe,dummy,dummy1]=feval(funfcn,lambda+F2*dellame,g,beta,...
     upper,csample,noise,nearzero,large);
fF2=norm(newfe);
while abs(F1-F2) > tolls
     if (fF1 > fF2 & fa > fF1)
          aa=F1;
          temp=F2;
          F1=F2;
          fF1=fF2;
          F2=temp+tau*(bb-temp);
          [newfe,dummy,dummy1]=feval(funfcn,...
               lambda+F2*dellame,g,beta,upper,...
               csample,noise,nearzero,large);
          fF2=norm(newfe);
     else
          bb=F2;
          temp=F1;
          F2=F1;
          fF2=fF1;
          F1=aa+(1-tau)*(temp-aa);
          [newfe,dummy,dummy1]=feval(funfcn,...
               lambda+F1*dellame,g,beta,upper,...
               csample,noise,nearzero,large);
          fF1=norm(newfe);
     end
end
alpha=(F1+F2)/2;
return
%%%%%%%%%%  end of function golden.m  %%%%%%%%%%
```

Table C.1: Variable definitions for MATLAB program.

Variable	Description	Page
`alpha`	optimal step length in Newton's method	59
`fittedc`	model fit to measured data	NA
`lamiter`	maximum number of iterations in Newton's method	NA
`large`	asymptotic cutoff for $\pm\infty$	57
`leftbegin`	lower limit of initial interval for bisection method	56
`maxiter`	maximum number of iterations in bisection method	NA
`p5`	5th percentile probability levels	36
`p95`	95th percentile probability levels	36
`pflag`	flag to select plotting options	NA
`rightbegin`	upper limit of initial interval for bisection method	56
`tolbeta`	stopping tolerance for bisection method	56
`tollam`	stopping tolerance for Newton's method	60
`tolls`	stopping tolerance for golden section search	NA

C.3 Variable Definitions

The variables used in the MATLAB program are defined in the following tables. The variables that are not defined by a symbol in this thesis are shown in Table C.1, with a brief description of the variable and a reference to the page number in this thesis where the variable is described. For the variables that have a corresponding symbol in this thesis, Table C.2 shows the variable names in the MATLAB program, the symbol used in this thesis, and a reference for the equation or page number where the symbol is defined.

Table C.2: Variable names and symbols for MATLAB program.

Variable name	Symbol	Reference
`a`	a_i	Page 35
`beta`	β_i	Equation 2.16
`cin`	$C_{in}(t)$	Equation 3.1
`csample`	$d = C(x_j, T)$	Equation 3.3
`csource`	$\mathbf{m}$	Equation 3.3
`dfe`	$\partial \mathbf{F} / \partial \boldsymbol{\lambda}$	Equations B.12 and 3.13
`dmndan`	$\partial \hat{m}_i / \partial a_i$ and $\partial \hat{m}_i / \partial b_i$	Equations B.14, 3.14, 3.15, 3.16, and 3.17
`expvalue`	s_i	Equation 2.15
`fe`	$\mathbf{F}$	Equations B.11 and 3.10
`g`	$\mathbf{G}$	Equation 3.3
`kernfn`	$f(x_j, T - t_i)$	Equation 3.2
`lambda`	$\boldsymbol{\lambda}$	Equation 2.24
`mhat`	$\hat{\mathbf{m}}$	Equations 2.28, 3.11, and 3.12
`ndata`	N	Page 18
`nearzero`	ϵ_0	Page 57
`noise`	ϵ	Equation 3.10
`nt`	M	Page 18
`params`	$[v, D]$	Equation 3.2
`t`	t_i	Page 41
`tsample`	T	Equation 3.1
`upper`	U_i	Equation 2.13
`xsample`	x_j	Page 41