Aerosol Data Inversion: Optimal Solutions and Information Content

Thesis by
J. Kenneth Wolfenbarger

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

California Institute of Technology
Pasadena, California

1990

(Submitted November 27, 1989)

To Mom, Dad, and LaReesa

Acknowledgements

I am most grateful to my advisor, John Seinfeld, for his support and advice throughout this project. Additionally, I thank John for the freedom, trust, and respect that he extended to me during this project, and I am thankful for all of the contributions that he has made to my professional health.

I also thank Rick Flagan for taking the time to exchange ideas with me about my project. Rick is *eager* to share his ideas with the graduate students, and enthusiasm like his is precisely why I have enjoyed my brief stay at the Institute.

I also wish to thank David Huang, Hung Nyguen, and Shih-Chen Wang for their help.

The financial support of the National Science Foundation is greatly appreciated.

Last but not least, a big "thank you" goes to Pam for too many things to mention here.

Abstract

The determination of an aerosol size distribution is presently difficult because current aerosol instruments cannot perfectly discriminate aerosols based on size and because a only limited number of data can be obtained. As a result, for a given set of data the relationship between the unknown distribution and the data is a finite Fredholm integral equation. If the size distribution is desired, then one should answer the following

- What measurements should be taken?
- How should the measurements be used to determine a size distribution?

In this thesis, we shed some light on the answers to these questions by finding optimal solutions to the Fredholm integral equation, and by characterizing the size of the solution set.

The questions of existence and uniqueness of solutions subject to linear inequality constraints are examined. Optimal solutions based on regularization are developed, and numerical methods for finding these solutions are described. Numerical experiments are presented that demonstrate the importance of

- describing dependent error sources.
- considering the magnitude of the errors in the data when there are few data.
- using generalized cross validation when there are many data and the magnitude of the errors is unknown.

An analysis that uses some simple information concepts is presented for examining the size of the solution set. An example is presented that demonstrates the effect of dependent errors on the information provided by the data, and some illustrative experiment design studies are presented.

Contents

List of Figures

Nomenclature

k_C	number of inequality constraints
m	number of measurements or data
n	number of linear splines used to represent the size distribution function
x	log of particle diameter
y_i	datum corresponding to the i^{th} instrument response
ε	measurement errors
λ	smoothing or regularization parameter
$k_i(x)$	i^{th} instrument response or kernel function
$f(x)$	aerosol size distribution function
$\bar{g}(x)$	basis for linear splines
K	operator that maps $f(x)$ to y_i
C	matrix of inequality constraints
D	finite difference matrix
E	matrix that relates the standard deviations of the error sources to the data
H	Hessian of quadratic functional
U	left-hand singular eigenvectors in singular value decomposition
V	right-hand singular eigenvectors in singular value decomposition
$R(f)$	$\|\bar{y}^R - \bar{y}^M\|^2$
$J_2(f)$	regularization functional, $\int (f^{(2)}(x))^2 \, dx$
$\mathcal{N}(0,1)$	the set of normal random variables with zero mean and unit variance
$\mathbf{E}$	expectation operator
s_E^2	$\mathbf{E}\ [\|\bar{y}^t - \bar{y}^M\|^2]$
$\mathcal{C}_s$	the set of $f(x)$ that satisfy $\|\bar{y}^M - \bar{y}^R\| \leq s$
$\mathcal{I}$	size interval over which the solution will be computed
Subscripts	
λ	obtained from the regularized solution with smoothing parameter λ

Superscripts

T	transpose
R	predicted by a particular $f(x)$
t	true value
M	measured value
$\dagger$	pseudo - inverse

Symbols

$\bar{a}$	overbar denotes a is a column vector

Chapter 1

Introduction

1.1 Motivation

A large component of research in aerosol science is devoted to understanding the physics and chemistry associated with the evolution of the aerosol size distribution, $f(x)$, that describes how the aerosol is distributed with respect to size x. A frequent problem in this regard is experimentally measuring the size distribution. The instruments that are designed to measure the size distribution in general separate particles based on size into a finite number of channels or bins, and then generate a signal that is in most cases proportional to the amount of aerosol in that channel. The instruments are not able to discriminate perfectly among the particles, and thus one is left with the following problem: find $f(x)$ given

$$\int_I f(x) k_i(x)\, dx = y_i^M + \varepsilon_i \quad i = 1, n \tag{1.1}$$

subject to inequality constraints, where $k_i(x)$ is proportional to the instrument's response to an impulse distribution of size x, and y_i^M and ε_i are the corresponding datum and error respectively. The constraints are generally linear, and reflect $f(x) \geq 0$ or $|y_i^t - y_i^M| \leq \delta$ for example.

This finite Fredholm integral equation is ill-posed. In particular, observe that (1) for a continuum the solution will be unstable, (2) for a finite number of data, then the solution is most likely not unique, and (3) because the data are in error, solutions need not exist.

Much of the motivation for working on this problem comes from the lack of an acceptable solution procedure. Since 1962, several solution techniques have been published [13]. Many of these methods have concentrated on the numerical procedure, and ignore the lack of uniqueness. As a result, the solution obtained is an uncharacterized, random element of the solution set. Even at the time of this writing, these approaches are still being proposed to the aerosol community because of a general lack of understanding of the aerosol data inversion problem. Crump and Seinfeld [13,12] made a significant contribution to the area in 1982 with the use of regularization and generalized cross validation in the computer codes INVERSE and CINVERSE. Their work, in addition to containing numerical deficiencies, was incomplete, and in many cases the proposed solutions were unrealistic.

Additionally, many researchers have sought solutions to the inversion problem when the number of data are limited. Here, the size of the set of reasonable distributions is large, and a single size distribution cannot adequately describe the "solution" to the inversion problem. A similar problem is describing a random variable with an undesirably large variance. The mean value of the random variable is only part of the story; the variance is also important. Thus, in the inversion problem, it is also desirable to have an estimate of variance of the solution, and this aspect has been ignored in the literature.

1.2 Thesis outline

The goal of this thesis is to describe optimal solutions to the aerosol data inversion problem and present techniques for finding these solutions, and to demonstrate the value and ease of estimating the variance of these solutions.

In Chapter 2, some aspects of existence and uniqueness to the inversion problem and the regularized approximations are examined. Regularized solutions are described, and several improvements over the regularized solution shown in [12] are presented. The problem of dependent errors in the data is examined, and numerical experiments are presented that demonstrate the value of including a description of the dependent errors. Theoretical and numerical comparisons are made between the regularized solution and other techniques in use today.

In Chapter 3, different methods for choosing the regularization parameter are examined and numerical techniques are developed for the use of these methods when inequality constraints are important. Numerical experiments are presented that demonstrate the importance of

- using the inequality constraints to choose the regularization parameter.
- using generalized cross validation when the magnitude of the errors is unknown.
- having sufficient data when using generalized cross validation.

In Chapter 4, the size distribution is viewed as an element of a random process with a known autocorrelation function. This information could be used to describe an optimal solution, but the resulting solution is too sensitive to the estimated mean and autocorrelation function. Instead the information is used to measure the size of the solution space in terms of the variance of linear functionals of the size distribution and to measure the amount of information provided by a set of measurements. Examples are presented that demonstrate the effect of dependent errors on the information provided by the data. Also some techniques for improving the amount of information provided by common instruments are experimentally investigated.

Finally, a computer program that makes the theory and calculations described in this thesis readily accessible was developed and is currently being distributed to the aerosol community around the world. A user's manual for this program is in the appendix.

Chapter 2

Inversion of Aerosol Size Distribution Data

Abstract

A comprehensive analysis of the problem of determining aerosol size distributions on the basis of data from conventional measuring instruments is presented. An inversion algorithm based on regularization is developed that enables one to find size distributions that are both smooth and faithful to a set of data generated by a combination of instruments. The algorithm includes consideration of the dependent nature of errors in the data. Several numerical examples are presented that compare the present technique with previously available methods.

2.1 Introduction

Aerosol size distributions are determined by passing the aerosol through instruments that classify the particles according to size based on their optical, electrical, or dynamic behavior. The raw data from these instruments must then be inverted to obtain the size distribution. The inversion of aerosol size distribution data to find the size distribution is not a straightforward problem, however, and some evidence of this are the many different methods proposed in the literature for finding size distributions.

The instruments used to determine the aerosol size distribution usually generate data as follows:

- the particles are separated into m channels based on size.
- a nonnegative signal, y_i^M, that is related to the number of particles in each channel is measured.

The relationship between the error-free data and the size distribution is generally given by

$$y_i^t = \int_{\mathcal{I}} k_i(x) f(x)\, dx \quad i = 1, \ldots m, \tag{2.1}$$

where the nonnegative kernel function for instrument channel i, $k_i(x)$, is determined from calibration data or theoretical models and represents the channel's response to a monodisperse aerosol sample of size x, and $\mathcal{I}$ is an interval such that the product $f(x)k_i(x)$ is zero outside $\mathcal{I}$. We can rewrite Eq. (2.1) compactly as

$$Kf = \bar{y}^t = \bar{y}^M + \bar{\varepsilon}^t \tag{2.2}$$

where overbars represent column vectors and $\bar{\varepsilon}^t$ represents the error of measurement.

Instruments are ideally designed to minimize the range of particle sizes that contribute to each datum, that is with the goal of having

$$k_i(x) \longrightarrow \alpha\delta(x - x_i),$$

and they are designed so that particles of each size contribute only to a single datum. In this case the relationship between $f(x)$ and $\bar{y}^M$ can be approximated by a matrix that has a stable inverse, and an approximate solution is easily found. This technique is used to invert data from the differential mobility analyzer (DMA) [3,23] that has kernel functions similar to those shown in Figure 1. The position of the primary peak is controlled by the strength of the electric field that collects charged particles in a given electrical mobility interval. The smaller secondary peaks are present in addition to the primary peak because larger particles that carry two or more charges can have the same mobility as a small particle carrying a single charge. If the variance of the peaks in the size distribution is large in comparison to the variance of the DMA peaks, as in the case of the distribution shown in Figure 1, then the size distribution can be approximated as a constant in the interval of the DMA peak. Additionally, if the remaining DMA kernel functions are chosen so that the primary peaks overlap the secondary peaks then the size distribution can be found by inverting a matrix that is nearly diagonal. Even though the inverse is stable here, we will see that often this inversion procedure is inadequate.

In general, however, the kernel functions of aerosol instruments are broad and can have considerable overlap. This can be seen, for example, from the kernel functions for a screen-type diffusion battery [2] also shown in Figure 1. Each stage consists of screens that collect the particles by diffusion, impaction, and interception; here the kernel functions represent the fraction of particles collected by the stages at each diameter. An $m \times m$ matrix approximation relating the data from this instrument to the size distribution yields an inverse that is overly sensitive to errors in the data. This instability due to kernel function overlap is compounded when data are available from more than one instrument responding to particles in the same size interval.

The difficulty of the aerosol data inversion problem lies in the failure of the inverse of K to exist; for most $\bar{y}^t$ there are infinitely many solutions to Eq. (2.2). Additionally, the domain of K is constrained to include only nonnegative functions. This constraint not only adds difficulty to the task of finding solutions, but also introduces the possibility that solutions to Eq. (2.2) do not exist for a given $\bar{y}^M$. Approximate inverses obtained by discretization usually generate unrealistic results because of their sensitivity to errors in the data.

The goal of this paper is to present a comprehensive analysis of the aerosol data inversion problem and to develop an inversion algorithm based on regularization that enables one to find smooth size distributions that are faithful to data from any combination of instruments that satisfy Eq. (2.1). We also show how information on the dependent nature of the errors in the data can be included in the inversion and show the importance of biasing the data to reflect dependent errors. The results of some numerical experiments are presented along with theoretical and numerical comparisons with other inversion techniques currently used.

2.2 The ill-posedness of the aerosol inverse problem

In this section some theoretical aspects of the linearly constrained inversion problem that apply to the aerosol inversion are discussed. We will show that solutions to Eq. (2.1) do not exist for all $\bar{y}^t$ and that when solutions do exist, they are rarely unique. The theorems we present will help explain why some of the inversion methods described in the literature can fail to converge.

A tool that we use in this paper is the singular value decomposition. Any $m \times n$ matrix E satisfies

$$E = [U_p, U_0] \begin{bmatrix} \Sigma_p, & 0 \\ 0, & 0 \end{bmatrix} \begin{bmatrix} V_p^T \\ V_0^T \end{bmatrix} \tag{2.3}$$

where Σ_p is a $p \times p$ diagonal matrix whose diagonals are the square roots of the positive eigenvalues of EE^T, $[U_p, U_0]$ is an $m \times m$ orthonormal matrix, and $[V_p, V_0]$ is an $n \times n$ orthonormal matrix. The matrices U_p and V_p have p columns, and this leads to

$$E = U_p \Sigma_p V_p^T.$$

The pseudo-inverse of E, written $E^\dagger$, is

$$E^\dagger = V_p \Sigma_p^{-1} U_p^T,$$

and arises in solutions to least-squares problems [28].

In the remainder of this paper we assume $f(x)$ lies in an n-dimensional vector space, $H_1(0,1)$, with basis vectors $g_i(x)$, and we write

$$f(x) = \bar{f}^T \bar{g}(x)$$

where $\bar{f}$ is the coordinate vector. The more general case of $f(x)$ in an infinite-dimensional Hilbert space is not considered here; we only assume that n is large enough to approximate *all* size distributions of interest as closely as needed. For example, it is difficult to justify choosing $n < m$, since we would be arbitrarily eliminating a large number of reasonable solutions and restricting ourselves to a set that contained no solution to Eq. (2.1). We also assume the kernel functions are linearly independent and that n is large enough to capture this independence.

We assume the size distribution must satisfy

1. $f(x) \geq 0$. This constraint differs from the remaining constraints in that it must be satisfied *by definition.* If the data are accurate and the problem is well-posed, then solutions to the unconstrained inversion will automatically satisfy this constraint, but this is rarely the case when inverting aerosol data.

2. constraints that arise from "accurate" linear combinations of data. We will discuss these further in Section 2.5.

3. constraints that are supplied by the experimentalist. These may include box constraints on the error or bounds on the total number or mass. The experimentalist should be aware of the danger of placing too many constraints on the solution, because one could inadvertently eliminate all of the feasible solutions.

4. $\sum_{i=1}^{m}(y_i^M - y_i^R) \leq 0$ in some cases. We discuss this constraint in more detail in Section 2.5.

To ease the analysis we assume

$$f(x) \geq 0 \quad \text{if and only if} \quad f_i \geq 0 \quad i = 1, \ldots n.$$

In other words one can constrain $f(x)$ to be nonnegative with only linear inequality constraints on the n f_i values. Thus all of the constraints listed above can be written compactly as the k_C inequality constraints

$$C\bar{f} \geq \bar{b} \tag{2.4}$$

where C is a $k_C \times n$ matrix. We will use $\mathcal{C}_\infty$ to represent the set of all $f(x)$ in $H_1(0,1)$ that satisfy Eq. (2.4). Observe that $k_C = 0$ implies $\mathcal{C}_\infty = H_1(0,1)$. Unless otherwise stated, we will assume $f(x)$ is at least constrained to be nonnegative.

Since $\bar{y}^M$ is in error, we are not only interested in solutions to Eq. (2.1), but more generally we are interested in all solutions that satisfy

$$||Kf - \bar{y}^M|| \leq s \tag{2.5}$$

where s is a nonnegative constant, and $||\cdot||$ represents the Euclidean norm. We use the notation $\mathcal{C}_s$ to represent the set of $f(x)$ in $\mathcal{C}_\infty$ that satisfy Eq. (2.5); thus $\mathcal{C}_0$ is the set of $f(x)$ in $\mathcal{C}_\infty$ that satisfy Eq. (2.1). Note also that in this notation

$$s \leq \hat{s} \quad \text{implies} \quad \mathcal{C}_s \subseteq \mathcal{C}_{\hat{s}}$$

Size distributions that satisfy Eq. (2.5) may not exist, or in other words $\mathcal{C}_s$ may be empty. It is clear, for example, that there are no $f(x) \geq 0$ that satisfy Eq. (2.1) when there are two data, $y_1 < y_2$, and $k_1(x) \geq k_2(x)$. This leads to the observation for any m that if we find a linear combination of data that is negative and the same linear combination of kernel functions is nonnegative, then $\mathcal{C}_0$ must be empty. Theorem 5 in the Appendix generalizes these observations to arbitrary C and offers an alternative characterization of the existence of solutions to Eq. (2.5).

If we assume the aerosol size distribution is unconstrained, then $\bar{b}, C^{\dagger}$, and U_0 are 0, $V_0 = I$, and Theorem 5 reduces to the Fredholm alternative of linear algebra:

One and only one of the following is true:

$$\begin{array}{lll} \text{(i)} & \|K\bar{f} - \bar{y}^M\| \leq s & \text{has a solution} \quad \bar{f} \\ \text{(ii)} & \bar{x}^T K = 0, \ \ \bar{x}^T\bar{y}^M > s, & \text{has a solution} \quad \|\bar{x}\| = 1 \end{array}$$

If the rows of K are linearly independent as we have assumed, then the columns of K span R^m, or $\bar{x} = 0$ is the only vector satisfying $\bar{x}^T K = 0$, and a solution to the aerosol size distribution can always be found when $f(x)$ is incorrectly assumed to be unconstrained.

If the only constraints on the solution are $\bar{f} \geq 0$, then U_0, V_0, and $\bar{b}$ are 0, $C = C^{\dagger} = I$, and Theorem 5 restates the Farkas alternative [15] of linear programming:

One and only one of the following is true:

$$\begin{array}{lll} \text{(i)} & \|K\bar{f} - \bar{y}^M \ \| \leq s & \text{has a solution} \quad \bar{f} \geq 0 \\ \text{(ii)} & \bar{x}^T K \geq 0, \bar{x}^T\bar{y}^M > s & \text{has a solution} \quad \|\bar{x}\| = 1 \end{array}$$

This coincides with the previous observation that if we find a linear combination of kernel functions that is nonnegative, then no size distribution can recover a data vector if the resulting linear combination of data is negative. Note also if $C = I$, and if there exists intervals $\mathcal{X}_i \subset \mathcal{I} \quad i = 1, \ldots m$ such that $k_i(x)$ is the only nonzero kernel function on $\mathcal{X}_i$, then for large n the Farkas alternative guarantees $\mathcal{C}_0$ is not empty. This follows from the observation that the existence of $\mathcal{X}_i$ implies $\bar{x}^T K \geq 0$ is true only if $\bar{x} \geq 0$, and this implies $\bar{x}^T\bar{y}^M \geq 0$ since $\bar{y}^M$ is nonnegative. Thus the second alternative is false, or Eq. (2.5) has a solution.

One property of the aerosol inversion problem that has been noted is the nonuniqueness of a solution to Eq. (2.5). This observation is based on the fact that $n > m$ implies the kernel of K is not empty. This analysis is not complete, however, because the set of solutions must lie in $\mathcal{C}_\infty$. The statement given in Theorem 6 in the Appendix is more precise and shows that the solution to Eq. (2.5) is unique in some rare circumstances. For example if $f(x)$ is constrained to be nonnegative, the single datum $y_1 = 0$, and $k_1 > 0$, then $f(x) = 0$ is the unique size distribution that satisfies $\|Kf - \bar{y}^M\| \leq 0$.

In the remainder of this paper we will be concerned primarily with solutions in the solution set $\mathcal{C}_{s_E}$ where s_E is the standard deviation of the sum of the errors,

$$s_E^2 = \mathbf{E}[\|\bar{y}^t - \bar{y}^M\|^2]. \tag{2.6}$$

Here $\mathbf{E}$ denotes the expectation operator. We assume the error sources are normal random variables and that $\bar{y}^M$ is bounded. This implies if $\mathcal{C}_\infty$ is not empty, then s_E is bounded. Theorem 7 in the Appendix confirms that the set we are interested in usually is not empty and that it usually does not define a unique element.

2.3 Constrained regularization

Regularization is a method for finding solutions to ill-posed problems [21,34,40]. To find a regularized solution, the ill-posed linear inverse problem given by Eq. (2.2) is approximated by a well-posed λ-family of problems that have stable inverses, $K_\lambda^\dagger$. The well-posed family of problems is chosen to approximate the ill-posed problem in the sense that

$$K_\lambda^\dagger \bar{y}^M \longrightarrow f_0(x) \text{ as } \lambda \longrightarrow 0$$

where $f_0(x)$ is a least-squares solution, and where $\lambda > 0$ is referred to as the regularization (or smoothing) parameter. The regularized solution to Eq. (2.2), $K_\lambda^\dagger \bar{y}^M$, is stable with respect to perturbations in $\bar{y}^M$ and can be chosen arbitrarily close to a least-squares solution.

In our case, we replace the ill-posed problem of finding solutions to Eq. (2.2) by the following:
Find $f(x) \in \mathcal{C}_\infty$ that minimizes $q(\lambda, f)$, where

$$q(\lambda, f) = R(f) + \lambda J_2(f), \tag{2.7}$$

and

$$\begin{aligned} R(f) &= \|Kf - \bar{y}^M\|^2 \\ J_2(f) &= \textstyle\int_I (f''(x))^2\, dx. \end{aligned}$$

The first term in Eq. (2.7), $R(f)$, penalizes solutions that disagree with the measured data. The second term, $J_2(f)$, is called the regularization (or smoothing) functional, and here we set $J_2(f)$ equal to the semi-norm originally proposed by Phillips [38] that penalizes solutions that are not smooth. This form of penalty seems justified when determining aerosol size distributions because aerosol processes often tend to smooth rough distributions. Also, in the absence of *a priori* information, it is difficult to justify presenting a solution with structure and oscillations that are beyond the resolving capabilities of the data.

The application of regularization to the aerosol inversion problem is more difficult than to the linear inverse problem described by Phillips because of the inequality constraints imposed by C. If the set of constraints is empty then the solution to the minimization of Eq. (2.7) is well known [42]:

$$\bar{f} = (K^T K + \lambda D^T D)^{-1} K^T \bar{y}^M$$

where

$$J_2(f) = \bar{f}^T D^T D \bar{f}$$

In our case, however, C at least contains the n inequality constraints $f \geq 0$, and these prevent us from finding an analytical solution.

It is easy to show that a minimum of Eq. (2.7) always exists in contrast to solutions defined by Eq. (2.2), as long as $\mathcal{C}_\infty$ is not empty. A unique minimum of Eq. (2.7), however, is not guaranteed. For the particular regularization functional used in this paper, we have a convenient sufficient condition for uniqueness: *if linear functions are not in the kernel of K, then* $\mathbf{f}_\lambda$ *is unique.* To prove this, note that the assumed condition implies $K^T K + \lambda D^T D$ is positive definite, and this implies the solution is unique [15].

The solution to Eq. (2.2) is still not defined because λ is not yet specified. To help guide us we note that choosing a larger λ increases the amount of smoothing at the expense of fidelity of the solution to the data. Theorem 8 in the Appendix makes this precise; this theorem tells us that over some interval of $R(\lambda)$ (or $J_2(f_\lambda)$), the question of choosing λ can be replaced by the problem of choosing $R(\lambda)$ (or $J_2(f_\lambda)$), and that over this interval the smoothness increases as λ increases. It is unlikely that one can *a priori* define an appropriate value for λ or $J_2(f_\lambda)$, but it seems reasonable to define a target $R(\lambda)$.

We proceed as suggested by Morozov [35] and define the target $R(\lambda)$, R_t, as

$$(R_t - s_E)^2 \leq \mathbf{VAR}[\|\bar{y}^t - \bar{y}^M\|] \qquad (2.8)$$

where s_E is defined in Eq. (2.6), and **VAR** denotes variance. Theorem 8 does not guarantee the existence of λ that satisfies Eq. (2.8); in fact the following observation shows that λ may not exist: *Let $\mathcal{J}_2$ be the set of all $f(x)$ in $\mathcal{C}_\infty$ that minimizes $J_2(f)$,* *and let $R_{0,\infty}$ be the minimum value of $R(f)$ on this set. If $s_E > R_{0,\infty}$, then* $\mathbf{f}_\lambda$ *satisfying Eq. (2.8) does not exist.* This claim can be argued by noting that if it is not true, then we can find a λ and a solution to Eq. (2.7), $\mathbf{f}_\lambda$, such that

$$R(\lambda) > R(f_J)$$

where f_J is an element of $\mathcal{J}_2$. This leads to

$$R(\lambda) + \lambda\, J_2(f_\lambda) > R(f_J) + \lambda\, J_2(f_J)$$

and contradicts the assumption that $\mathbf{f}_\lambda$ is a minimum of Eq. (2.7). The difficulty here is that when the expected errors are large, there can be an infinite number of solutions that both minimize $J_2(f)$ and satisfy $R(f) < s_E$; $J_2(f)$ is unable to discriminate among solutions when large errors are acceptable. Here we choose the solution in $\mathcal{J}_2$ that minimizes $R(f)$.

Calculating s_E is usually difficult, and to simplify the calculations we assume Eq. (2.1) has been rescaled by $\sqrt{m}\sigma_i$. When $k_C = 0$, then the calculation is trivial and yields

$$s_E = 1. \tag{2.9}$$

Using Eq. (2.9), as suggested in [12], is a simplification that has caused us difficulties because sometimes the minimum of $R(f)$ on $\mathcal{C}_\infty$ is greater than one. One must remember that the errors may not be normally distributed after $\bar{y}^M$ is measured if $k_C \neq 0$. Here we simplify the calculation of s_E by assuming that the range of K, $\mathcal{R}(K)$, is radially distributed about $\bar{y}^M$, or

$$\bar{y} \in \mathcal{R}(K) \quad \text{implies} \quad \alpha\frac{\bar{y}}{\|\bar{y}\|} \in \mathcal{R}(K) \tag{2.10}$$

where

$$R_0 \leq \alpha \leq R_{max}\ , \tag{2.11}$$

and R_0 and R_{max} are the minimum and maximum of $R(f)$ on $\mathcal{C}_\infty$. If C only implies $f(x) \geq 0$, then for m even we find

$$s_E{}^2 = 1 + \frac{R_0{}^2}{m\sum_{i=0}^{\hat{m}} M_i R_0{}^{-2i}}$$

where

$$M_i = \frac{\hat{m}!2^i}{(\hat{m}-i)!},$$

and $\hat{m} = m/2 - 1$. If in addition C implies $\sum_{i=1}^{m}(y_i^M - y_i^R) = 0$, then we find for m odd

$$s_E{}^2 = 1 + \frac{\beta - 1}{\sum_{i=0}^{\hat{m}} M_i(\beta R_0{}^{-2i-2} - R_{max}{}^{-2i-2})}$$

where

$$\beta = e^{(R_{max}{}^2 - R_0{}^2)/2}(R_0/R_{max})^{2m-2}$$

and $\hat{m} = (m-3)/2$.

One could alternatively account for positive R_0 by assuming that the range of K is bounded by a single plane. Here we find

$$s_E{}^2 = 1 + R_0^2.$$

One can show s_E computed from this equation is always greater than the results obtained from the assumption described by Eq. (2.10).

Alternative smoothing functionals

The penalty term $J_2(f)$ biases the solution to reduce unrealistic properties in the size distribution. As mentioned previously, the penalty term $J_2(f)$ is justified here because highly oscillatory size distributions are not desirable. For any given experiment, however, the penalty term $J_2(f)$ may not be the most appropriate. For example if one expects most of the particles to be concentrated in a small size interval, then biasing the inversion procedure to favor solutions that are overly smooth in this region is not appropriate, or if we expect a small number of particles near the ends of the distribution, then in this region it is more appropriate to include $\int(f(x))^2dx$ as part of the penalty term.

A more general penalty term is

$$J(f) = \int \sum_{i=0} w_i(x)(f^{(i)}(x) - p_i(x))^2\,dx$$

where $w_i(x) \geq 0$ is a weighting function, and $p_i(x)$ is a desired function. The analytical and numerical details of using a penalty function of this form are similar to those required for $J_2(f)$. We do not compare the sensitivity of $\mathbf{f}_\lambda$ to different forms of $J(f)$ or discuss the advantages of using different regularization functionals. We only note that $J(f)$ provides a justifiable method for comparing different solutions that are equally faithful to the data, and when the experimental conditions suggest an appropriate $J(f)$, the inverted distributions obtained using this penalty term will be more desirable than solutions obtained using $J_2(f)$, for example.

2.4 Equivalent optimization statements

In this section we will discuss some optimization problems that are equivalent to finding the minimum of Eq. (2.7) [39]. This will help motivate and add validity to some of the statements presented later in this paper.

Theorem 1 *If* $\mathbf{f}_\lambda$ *is the unique minimizer of Eq. (2.7) and* $\hat{f}$ *is a solution to*

$$\begin{array}{ll} \underset{f \in \mathcal{C}_\infty}{\textit{minimize}} & \|Kf - \bar{y}^M\| \\ \textit{subject to} & J_2(f) = J_2(f_\lambda) \ , \end{array}$$

then $\hat{f} = \mathbf{f}_\lambda$.

Proof: Assume instead $\mathbf{f}_\lambda \neq \hat{f}$. Since $\mathbf{f}_\lambda$ is the unique minimizer of Eq. (2.7), and $J_2(f) = J_2(f_\lambda)$, we find

$$\|K\hat{f} - \bar{y}^M\| > R(\lambda)$$

and this contradicts the definition of $\hat{f}$. □

This claim says that if one finds a solution to the aerosol inversion problem, $f(x)$, and the solution has smoothness $J_2(f)$, then regularization can provide a solution with the same smoothness, and the solution will match the measured data better.

Theorem 2 *Let* $p_{\bar{\varepsilon}^t}$ *represent the probability density function of* $\bar{\varepsilon}^t$. *Assume that the unconditioned errors in the data are normal random variables with unit variance and zero mean. If* $\mathbf{f}_\lambda$ *is the unique minimizer of Eq. (2.7), and* $\hat{f}$ *is a solution to*

$$\begin{array}{ll} \underset{f(x) \in \mathcal{C}_\infty}{\textit{maximize}} & p_{\bar{\varepsilon}^t}(Kf - \bar{y}^M) \\ \textit{subject to} & J_2(f) = J_2(f_\lambda) \ , \end{array}$$

then $\mathbf{f}_\lambda = \hat{f}$.

Proof: To see that this is true, note that if $f_1(x)$ and $f_2(x)$ lie in $\mathcal{C}_\infty$ then

$$p_{\bar{\varepsilon}^t}(Kf_1 - \bar{y}^M) = p_{\bar{\varepsilon}^t}(Kf_2 - \bar{y}^M) \iff R(f_1(x)) = R(f_2(x))$$

and

$$p_{\bar{\varepsilon}^t}(Kf_1 - \bar{y}^M) > p_{\bar{\varepsilon}^t}(Kf_2 - \bar{y}^M) \iff R(f_1(x)) < R(f_2(x)) \ .$$

Thus $\hat{f}$ defined by Theorem 2 is equivalent to $\hat{f}$ defined by Theorem 1, and thus Theorem 2 is just a restatement of Theorem 1. □

Note that $p_{\bar{\varepsilon}^t}(Kf - \bar{y}^M)$ is a more accurate description of the fidelity of the solution to the data than $\| Kf - \bar{y}^M \|$, and that the descriptions are equivalent when the error sources can be modeled as random numbers with zero mean and unit variance. Thus, as before, if we find a solution to the inversion problem by another algorithm, and the solution has smoothness $J_2(f)$, then regularization can provide a solution whose roughness is $J_2(f)$, and the solution will be more probable in the sense that the probability density function of the recovered errors is larger.

Theorem 3 *If* $\mathbf{f}_\lambda$ *is the unique minimizer of Eq. (2.7), and* $\hat{f}$ *is a solution to the following optimization problem:*

$$\begin{array}{ll} \underset{f(x)\in\mathcal{C}_\infty}{\textit{minimize}} & J_2(f) \\ \textit{subject to} & \| Kf - \bar{y}^M \| = R(\lambda) \ , \end{array}$$

then $\hat{f} = \mathbf{f}_\lambda$.

This can be verified in the same way as Theorem 1. This restatement says that if we find a solution to the inversion problem by another algorithm, and the norm of the recovered errors equals $R(f)$ then regularization can provide a solution with the same recovered errors, and the regularized solution will be smoother.

2.5 Data with independent errors

As mentioned earlier, $R(f)$ in Eq. (2.7) penalizes solutions that are unfaithful to the data. Theorem 2 suggests that $R(f)$ takes on more meaning as a penalty term when $p_{\bar{\varepsilon}^t}(Kf - \bar{y}^M)$ is a function only of $R(f)$ for $f(x) \in \mathcal{C}_\infty$, because here $R(f)$ reflects the probability that $\bar{y}^M$ would be observed if the true solution is $f(x)$. The importance of this is easy to see in the case $m = 2$, where the data are independent and normally distributed with $\sigma_1 = 1$ and $\sigma_2 = 10$, and two solutions, $f(x)$ and $\hat{f}(x)$ are available that generate the recovered errors

$$\bar{\varepsilon}^R = \begin{pmatrix} 1 \\ 10 \end{pmatrix} \text{ and } \hat{\bar{\varepsilon}}^R = \begin{pmatrix} 10 \\ 1 \end{pmatrix} .$$

Both solutions have identical values of $R(f)$, but clearly the probability that $\bar{y}^M$ would be observed is orders of magnitude larger if $f(x)$ is the solution instead of $\hat{f}(x)$. When the errors in the data are independent and normally distributed, this undesirable property of $R(f)$ is corrected by rescaling the data by the standard deviations σ.

The errors in aerosol size distribution data, however, are dependent. The reason is that several data can be generated by a single instrument, and a given instrument usually has operating parameters and noise sources that affect all the readings. The need to account for dependent errors becomes more apparent when data from multiple instruments are inverted. One can find that the data from a single instrument are self-consistent, but that the data from different instruments disagree due to error sources that affect all the data from a single instrument.

One can model dependent errors in the data by assuming there exist k_E independent and normally distributed sources of error that affect the data. Each source of error is capable of adding noise to any datum, and the relationship between the error sources and the data is given by the matrix E, where the ij^{th} element of E represents the standard deviation of the error added to the i^{th} datum by the j^{th} source. The goal is to find a transformation, $E^\ddagger$, such that $p_{E^\ddagger\bar{\varepsilon}^t}(E^\ddagger(Kf - \bar{y}^M))$ is a function only of $\|E^\ddagger\,(Kf - \bar{y}^M)\|$. The following claim makes short work of our search for $E^\ddagger$.

Theorem 4 *Assume the relationship between the error sources and the data is given by the matrix E, and let U, Σ, and V refer to the matrices obtained from the singular decomposition of E. Then the probability density function of the errors in $E^\ddagger\bar{y}^M$ is a function only of $\|E^\ddagger\,(Kf - \bar{y}^M)\|$, where $E^\ddagger = Q_p\Sigma_p^{-1}U_p^T$, and Q_p is any $p \times p$ rotation matrix.*

Proof: The probability density function of $E^\ddagger\varepsilon^t$ is the same as the probability density function of $Q_pV_p^Tz$, where $z_i \in \mathcal{N}(0,1)$ (normal with zero mean and unit standard deviation). Since V is orthonormal, the probability density function of $Q_pV_p^Tz$ is a function only of $\|V_p^Tz\|$. Thus given two feasible data vectors, $\bar{y}_1^R$ and $\bar{y}_2^R$,

$$p_{E^\ddagger\bar{\varepsilon}^t}(E^\ddagger(\bar{y}^M - \bar{y}_1^R)) = p_{E^\ddagger\bar{\varepsilon}^t}(E^\ddagger(\bar{y}^M - \bar{y}_2^R))$$

if and only if

$$\|E^\ddagger(\bar{y}^M - \bar{y}_1^R)\| = \|E^\ddagger(\bar{y}^M - \bar{y}_2^R)\|.\Box$$

The solution obtained using the transformed data reflects the dependent nature of the errors in the data, and for a given smoothness the regularized solution obtained using the transformation $E^\ddagger = Q_p\Sigma^{-1}U^T$ is the most probable as measured by $p_{E^\ddagger\bar{\varepsilon}^t}$.

Note that $E^\ddagger$ is $p \times m$, and this implies the regularization is performed with $p \leq m$ data, or $m - p$ pieces of data are lost when Eq. (2.2) is transformed by $E^\ddagger$. This reflects that the rank of E equals p, or that there are some linear combinations of data that have negligible error. These linear combinations are given by $U_0^T\bar{y}^M$, where U_0 is defined by Eq. (2.3). Since there is negligible error in the equations

$$U_0Kf = \bar{y}^M,$$

these must be included in the set of linear constraints.

The transformation $E^{\ddagger}$ also reveals difficulties in the data inversion procedure we have described. As an example, assume C only implies $f(x) \geq 0$, and that the relationship between the errors and the data is given by

$$E = \begin{pmatrix} \alpha y_1 & & & & \beta y_1 \\ & \alpha y_2 & & 0 & \beta y_2 \\ & & \ddots & & \vdots \\ & 0 & & \ddots & \vdots \\ & & & & \beta y_m \end{pmatrix}, \qquad (2.12)$$

where $\alpha \ll 1$, and $\beta \ll 1$. Also assume $y_i^M = 1$ and that there are a large number of data so that $\sqrt{m}\beta \geq 1$ is satisfied. Then one can show that $f_0(x) = 0$ satisfies $R_t \leq s_E$ by noting that

$$E\bar{z} = \bar{y}^M - Kf_0$$

if $z_i = \delta_{i,m}\sqrt{m}$. In other words, even though the data are accurate, as more data are included in the inversion the solution defined by Eq. (2.8) can collapse to zero, *independent of the kernel functions.*

A problem occurred while finding a solution for the given E because there existed a feasible error vector $\|\bar{\varepsilon}^R\| = s_E$ that also satisfied

$$(E^{\ddagger})^{-1}\bar{\varepsilon}^R = -\bar{y}^M. \qquad (2.13)$$

The regularization algorithm will choose this solution if it is available because solutions that satisfy $0 \leq \bar{y}^M \ll \bar{y}^R$ are small in magnitude and therefore are smooth. Note that this problem cannot occur if E is diagonal and σ_i is a constant fraction of y_i^M. The solution obtained when Eq. (2.13) is satisfied is undesirable not because it is rough or because the recovered data are improbable, but because *all* of the recovered data are much less than the measured data. This reflects the bias that regularization has for choosing solutions that satisfy $\bar{y}^M > \bar{y}^R$ simply because they are smoother.

The bias regularization has for choosing $cf(x)$ instead of $f(x)$, $c < 1$, simply because $cf(x)$ is smoother is undesirable and can be eliminated by imposing the constraint

$$\Sigma_{i=1}^{m}(y_i^M - y_i^R) \geq \mathbf{E}[\Sigma_{i=1}^{m}(y_i^M - y_i^t)], \qquad (2.14)$$

which usually equals zero. The importance of using this constraint for the error model described by Eq. (2.12) was tested by simulating 18 data for the DMA while assuming the true distribution satisfied $f^t = \mathrm{lgn}(1, .07, 1.8)$, where $\mathrm{lgn}(a, b, c)$ refers to a log-normal distribution with concentration a, log-mean diameter b, and geometric standard deviation c. Error was added to the data as described by Eq. (2.12) for $\alpha = 2\%$ while β ranged from 0 to 10%. Many sets of erroneous data were inverted for each value of β, and the average values of $\|f^R - f^t\|_{TV}$ and $\|f^R - f^t\|_1$ were computed for the solutions obtained, imposing and ignoring the constraint given by Eq. (2.14) where

$$\|f\|_{TV} = \min_{x \in \mathcal{I}} f(x) + \int_{\mathcal{I}} |f'(x)|\, dx$$

and

$$\|f\|_1 = \int_{\mathcal{I}} |f(x)|\, dx.$$

The results in Figure 2 show that in this example the solution obtained using the constraint defined by Eq. (2.14) remains accurate for all values of β, and that the constraint prevents the solution from collapsing as β becomes large. It is not surprising in light of the preceding discussion that for the unconstrained curve and large β,

$$d\,\|f^R - f^t\|_1 / d\beta \approx \sqrt{18}$$

2.6 Numerical approach

To solve the minimization described by Equations 2.7 and 2.8 we assume $f(x) \in R^n$ is a linear spline with equally spaced knots. It would be more realistic to restrict $f(x)$ to lie in some other vector space such as cubic splines or Legendre polynomials; however, these require the use of nonlinear constraints in order to ensure $f(x) \geq 0$, while the linear spline only requires linear constraints, and this leads to a considerable savings in the computational effort. We find

$$f(x) = \bar{f}^t \bar{g}(x)$$

$\bar{f} \in R^n$, and

$$\begin{aligned}
g_i(x) &= h_i(x)(1 - d_i(x))(1 - \delta_{i,n}) + d_{i-1}(x) h_{i-1}(x)(1 - \delta_{1,i}) \\
h_i(x) &= H(x - x_{i+1}) - H(x - x_i) \\
d_i(x) &= (x - x_i)/\Delta x \\
x_i &= x_1 + (i-1)\Delta x \\
\Delta x &= (x_n - x_1)/(n-1).
\end{aligned}$$

$H(x)$ is the Heaviside unit step function, $\delta_{i,j}$ is the Kronecker delta, and x_1 and x_n are the endpoints of the inversion interval.

If $\bar{f} \in R^n$, then the linear functional of $f(x)$, $k_i(x)$ can also be represented as an element in R^n, and one finds

$$K_{ij} = \int_I k_i(x) g_j(x) dx. \tag{2.15}$$

We should note that Eq. (2.15) is exact for $f(x)$ in the set of linear splines in contrast to the commonly used approximation

$$K_{ij}(x) = k_i(x_j) \tag{2.16}$$

The quadrature rule defined by Eq. (2.16) can be a major source of error even when n is large enough to accurately represent the size distribution function. This approximation, for example, helped make the inversion of DMA data with CINVERSE [12] unnecessarily expensive.

The second derivative of $\bar{f}^T \bar{g}(x)$ is zero where defined; therefore, we define the second term in Eq. (2.7) using finite difference and the trapezoidal integration rule. We find

$$J_2(f) \;=\; \bar{f}^T D^T D \bar{f}$$

where

$$\sqrt{\Delta x} D_{ij} = (1 - \delta_{i,1})(1 - \delta_{n,j})(\delta_{i,j-1} - 2\delta_{i,j} + \delta_{i,j+1}).$$

Thus our solution to the aerosol size distribution problem is the $\bar{f}_\lambda$ that minimizes

$$\frac{1}{2} \bar{f}^T H \bar{f} + \bar{c}^T \bar{f} \tag{2.17}$$

subject to the constraint

$$\|E^\ddagger (K \bar{f}_\lambda - \bar{y}^M)\| = \mathbf{E} \|E^\ddagger \bar{\varepsilon}^t\|$$

where

$$\begin{aligned} H &= K^T E^{\ddagger T} E^\ddagger K + \lambda D^T D \\ \bar{c} &= K^T E^{\ddagger T} E^\ddagger \bar{y}^M \end{aligned}$$

We calculate the solution, $\mathbf{f}_\lambda$, and R_0 in Eq. (2.11) using the iterative two-phase active-set QP solver described by Gill *et al.* [18]. The two phases are a feasibility phase used to minimize the sum of the squares of the violated constraints, and an optimality phase used to minimize the quadratic functional. At each iteration of either phase a working set of constraints is chosen to be satisfied as equalities. Based on the results of the minimization using the current working set, a constraint may be added and (or) deleted from the working set for the next iteration. This method of solving the QPs described here is superior to linear based QP solvers, such as the one used by Crump and Seinfeld [12] because small changes in λ lead to small changes in the final working set. This implies the active-set QP solver will require only a few iterations of constraint swapping to compute solutions for future values of λ.

The target smoothing parameter is found by trial and error, and by taking advantage of the monotonicity of $R(\lambda)$ as a function of λ. A simple calculation shows that for a fixed set of constraints,

$$dR(\lambda)/d\log(\lambda) \longrightarrow 0 \text{ when } \lambda \longrightarrow 0 \text{ or } \lambda \longrightarrow \infty,$$

and this is demonstrated by a typical plot of $R(\lambda)$ as a function of λ shown in Figure 3, where we assumed f^t and the data were the same as those used to generate Figure 2, and error was added to the data assuming $\sigma_1 = 0.01y_i$. This suggests that if one uses a derivative based root finder to determine the target smoothing parameter, then it is important to add safeguards that limit the distance between successive iterates [6]. The failure to do this was one source of trouble for the inversion routine CINVERSE [12].

2.7 Examples

In this section we present the results of two numerical experiments. First we present an example of inverting data with dependent errors and show the importance of taking the dependence of errors into account when using regularization. In the second example we show the effect of incorrectly specifying the amount of error in the data. Both of these examples are similar because they demonstrate the relationship between the errors (or perceived errors) and the regularized solution.

In the first example, data were simulated for an 11 stage screen-type diffusion battery [7,8] and a DMA [27,3]. The detector for each instrument was assumed to be a condensation nuclei counter (CNC) with the response given in [2]. The datum for channel $2 \leq i \leq 12$ of the diffusion battery was defined as the number of particles that penetrated stage $i - 2$ minus the number of particles that penetrated stage $i - 1$, and the datum for channel 1 was defined as the total concentration detected by the diffusion battery's CNC. For the DMA, the evenly spaced voltage sequence described in [23] was used in this experiment for convenience; unlike the commonly used methods for inverting DMA data, the inversion algorithm described in this paper does not require the use of a particular voltage sequence. We also assumed a second CNC was used as a detector for the DMA.

Error was added to the data, assuming that two major sources of error existed for each instrument. First, we assumed that each CNC incorrectly counted particles according to the rule

$$N^M = N^t(1 + \beta_j z_j)$$

where the magnitude of β_j represents the standard deviation of the j^{th} dependent error source and the index j references either the DMA or the diffusion battery; $z_j \in \mathcal{N}(0,1)$ is assumed to be constant for a given set of data. The second source of errors was assumed to be caused by fluctuations in the aerosol source. In other words, the distribution is incorrectly assumed to be at steady state, but undetected variations in temperature, humidity, etc., cause fluctuations in the true distribution. We assumed these fluctuations in the size distribution occur on a time scale smaller than the data collection time scale and cause counting errors of the form

$$N^M = N^t(1 + \alpha)z_i,$$

where i denotes i^{th} datum, and $z_i \in \mathcal{N}(0,1)$.

Given these assumptions, the error matrix E is 29 × 31 and has the form

$$E = \begin{bmatrix} E_{DMA}, & 0 \\ 0, & E_{DB} \end{bmatrix},$$

where E_{DMA} is 18 × 19 and has the form

$$E_{DMA} = \begin{pmatrix} \alpha y_1^t & & & & \beta y_1^t \\ & \alpha y_2^t & & 0 & \beta y_2^t \\ & & \ddots & & \vdots \\ & 0 & & \ddots & \vdots \\ & & & \alpha y_{18}^t & \beta y_{18}^t \end{pmatrix},$$

and E_{DB} is 12 × 13 and has the form

$$E_{DB} = \begin{pmatrix} \alpha N_0^t & & & & & \beta y_1^t \\ \alpha N_0^t & -\alpha N_1^t & & 0 & & \beta y_2^t \\ & \ddots & \ddots & & & \vdots \\ & 0 & \ddots & \ddots & & \vdots \\ & & & \alpha N_{11}^t & -\alpha N_{12}^t & \beta y_{13}^t \end{pmatrix}.$$

where N_i^t is the number of particles detected by an error free CNC.

In the numerical experiment we assumed $f^t = \text{lgn}(1.0, .07, 1.8)$, $\alpha_i = 2\%$ for all i and that β was equal for the two CNCs. Error was generated with a pseudo-random number generator and added to the true data according to the error model described above while allowing β to vary between 0 and 10%. Many sets of noisy data were inverted for each value of β in the following two cases:

1. the information provided by E was used.

2. the errors were assumed to be independent.

In the second case, the standard deviation of the i^{th} measurement is defined by the i^{th} row of E,

$$\sigma_i^2 = \sum_{j=1}^{k} E_{ij}^2,$$

where E_{ij} is the j^{th} element of the i^{th} row of E. The average value of $||f^t - \mathbf{f}_\lambda||_{TV}$ was computed for each β, and the results are shown in Figure 4. Here the information provided by E about the error sources helped bias the solution to prevent unrealistic recovered errors, and this improved the accuracy of the regularized solution.

The second example shows the importance of correctly specifying the amount of error in the data. This example is similar to the first because both demonstrate the effect of incorrectly characterizing the error sources. A single set of noisy data was simulated for an optical particle counter, impactor, electrical aerosol analyzer, a diffusion battery, and a DMA. The errors in the data were independent, and $\sigma_i = 0.15y_i$. The data were inverted while the assumed σ_i varied between $0.05y_i$ and $0.25y_i$. The results are shown in Figure 5; the true distribution is accurately reconstructed in the figure when the assumed $\sigma_i/y_i = 15\%$. Notice that the solution exhibits nonexistent structure when the amount of error is underestimated, and that true peaks are smoothed out as the amount of error in the data is overestimated.

The sensitivity of the regularized solution to the specified error model may seem to be a weakness in the algorithm described in this paper; however as brought out in Section 2.4, this sensitivity must be present in any algorithm that targets an acceptable amount of error. This sensitivity reflects the importance of knowing the amount of error in the data. The perceived accuracy of the data is a measure of one's confidence in the data's ability to choose among solutions. When the accuracy of the data is overestimated, then too much confidence has been placed in the data's ability to determine a highly structured solution, or when the accuracy is underestimated, one needlessly loses confidence that the structure observed in the solution really exists.

2.8 Comparison with other aerosol inversion techniques

The inversion technique described in this paper can be compared with other inversion algorithms that have been proposed. The comparison is based on fundamental observations where possible, and a numerical comparison is made with another method recently proposed. The inversion algorithms we consider are constrained least-squares [9], Twomey's [41], EM [32] and STWOM [33].

We point out that the algorithm described in this paper is similar to CINVERSE [12] with the following extensions:

- correcting the definition of the target smoothing parameter to include the possibility that Eq. (2.5) has no solution.
- eliminating the assumption that the errors in the data are independent.
- allowing arbitrary inequality constraints.
- eliminating the numerical errors caused by discretizing the kernel functions.

It is convenient to view the process of finding a solution to the aerosol inversion problem as having three parts:

1. define the set of feasible solutions.
2. define the optimal solution in the feasible set.
3. find the optimal solution.

In the algorithm we have developed, the set of feasible solutions is the set of nonnegative linear splines. The dimension of the solution vector is chosen so that we have the largest set possible from which to choose the optimal solution. The optimal solution is the smoothest solution that is within the expected error, and we have shown that this definition usually is unique. We concentrate our comparison with other inversion algorithms based on the first two items listed because most inversion algorithms fail to properly define the solution set and fail to suitably define an optimal element in the feasible set. In these cases the numerical details of finding the optimal solution are less important.

The set of feasible solutions

Defining the feasible set usually involves defining a vector space in which the solution will lie and specifying the constraints. Sometimes we observe that inversion techniques constrain the solution to lie in an unrealistically small vector space. This includes all methods that attempt to fit an equation with a small number of parameters to the data, for example the log-normal curve fitting solution described by Helspar *et al.* [24]. Another example of this is the EM algorithm described by Maher and Laird [32], where inversion techniques are compared while the solutions were restricted to a 6-dimensional vector space. These restrictions are undesirable because they eliminate without justification an infinitely large set of possible solutions and because they introduce the possibility that the true solution is far from the artificially small set of solutions. Artificially small feasible sets are used by some inversion techniques to ensure the optimal solution is unique; however, if the optimal solution is not unique, then it seems more reasonable to redefine the target solution than to make unrealistic restrictions on the feasible set.

The optimal solution

Constrained least squares, EM, and Twomey's algorithm suffer from a poorly defined optimal solution. These methods define the optimal solution as the solution that minimizes the recovered errors. Section 2.2 shows this solution is rarely unique unless the feasible set has been artificially constrained. This lack of uniqueness is undesirable because one finds a solution that is a random element of a possibly large solution set. Even if the solution was unique, finding the solution that minimizes the error is a questionable objective; Figure 5 shows that the inverted distribution reflects the error in the data through artificial peaks as $||\bar{\varepsilon}^R||$ tends to its minimum value.

A numerical example

We compare the results of the inversion method described in this paper with STWOM [33]. STWOM is an extension of Twomey's algorithm that smooths a solution found using Twomey's algorithm.

Twomey's algorithm is an iterative method that corrects the size distribution f^0 so that it agrees with $\bar{y}^M$. The corrected distribution is found by calculating

$$f^i(x) = f^{i-1}(x)(1 + \alpha_i k_i(x)) \quad i = 1, \ldots m$$

where

$$\alpha_i = \frac{y_i^M}{\int_I f^{i-1}(x) k_i(x) dx} - 1,$$

and where $k_i(x)$ is rescaled to satisfy $k_i(x) \leq 1$. If f^m does not agree with the data, then the procedure is repeated as necessary with f^0 equal to the most recently calculated f^m. Any size distribution that agrees with the data will cause the procedure to stop, and depending on the initial guess and kernel functions, the final solution may be unreasonably oscillatory [13].

Given a Twomey solution $f^{m,0}$ that satisfies $\sum_{i=1}^m (\varepsilon_i/\sigma_i)^2 \leq m$, STWOM improves the solution by repeating the following procedure $p = 1, \ldots$:

1. repeatedly smooth $f^{m,p-1}$ to obtain $f^{0,p}$ that satisfies $\sum_{i=1}^m (\varepsilon_i/\sigma_i)^2 \geq m$,
2. use Twomey's method to correct $f^{0,p}$ to obtain $f^{m,p}$ that satisfies $\sum_{i=1}^m (\varepsilon_i/\sigma_i)^2 \leq m$.

A single smoothing in step 1 is carried out by calculating

$$f_s = \int w(x) f(x) dx$$

where $w(x)$ is a positive averaging function satisfying $\int w(x)dx = 1$. The entire procedure is terminated after step 2 if $\int |(f^{m,p})''(x)| dx$ does not change with successive values of p. Thus STWOM improves Twomey's solutions by eliminating oscillatory solutions via step 1 while keeping the solution faithful to the data.

Although STWOM is simple to implement and is an improvement over some of the available algorithms, this method is still flawed because the solution is not optimal and poorly characterized. Since many solutions are possible, the experimentalist needs to know *of all the possible size distributions that agree with the data, which one do I have?* With STWOM the answer is not clear because the solution is defined only in terms of the numerical routine; this is in contrast to regularized solutions that are defined in terms of observables before the numerical search begins (see Sec. 2.3). STWOM's lack of a well characterized solution is made evident by the haphazard path taken to reach the solution (too much smoothing followed by too much roughening etc.) instead of a path defined by a gradient that leads to a minimum.

We numerically tested the effects of error in the data on the algorithm developed in the present paper and STWOM. In the first experiment, data from an EAA were simulated while assuming $f^t = \text{lgn}(1.0, 0.08, 2.2)$. The errors in all the channels were assumed to be normally distributed with $\sigma_i = \alpha y_i^t$, and α was assumed to range between 1 and 20%. The errors were generated with a pseudo-random number generator, and many sets of erroneous data were inverted for each value of α. STWOM was given the initial distribution that was suggested in [33]. Figure 6 shows the average $||f^t - f^R||_{TV}$ as a function of α for both inversion techniques and shows in this case that the regularized solution is better able to reconstruct the true solution at all levels of added error. Here the average $||f^t - f^R||_{TV}$ for the STWOM solutions is about twice that of the regularized solutions.

We also compared the ability of these two algorithms to reconstruct a distribution with a larger $||f^t||_{TV}$ as data from more instruments were included in the inversion. STWOM was originally developed to invert EAA or impactor data, but the algorithm on which STWOM is based is independent of the instrument as long as the relationship between the data and the size distribution is given by Eq. (2.2). Data were simulated for an optical particle counter, MOUDI impactor, DMA, low pressure impactor, and a diffusion battery while assuming the tri-modal distribution shown in Figure 7. The DMA and diffusion battery respond to the smaller particles represented by the larger peak, and the remaining instruments repond to the larger particles. Random error was added to each measurement with $\alpha = 15\%$, and this was used to define the stopping criteria for STWOM. The initial STWOM guess was the same as the true distribution except the mean diameters of the peaks were adjusted 10%.

Three inversions were performed with data from the following combinations of instruments:

(a) the diffusion battery and the OPC

(b) same as (a) except DMA data were included

(c) same as (b) except the data from both impactors were added.

The results of the inversions are shown in Figure 7. Note that as more data are added, the regularized solution better reconstructs the true distribution in contrast to STWOM, which shows little improvement between steps **a** and **c**. STWOM is unable to take advantage of all the information provided by the many pieces of noisy data because the corrections are inconsistent and lack overall direction; the algorithm literally goes in circles looking for a solution. It is not surprising that in this example STWOM required about 10 times as much computing time as the method based on regularization.

2.9 Conclusions

Solutions to the aerosol data inversion problem are nonunique. A variety of algorithms have been previously proposed to select one solution from the many possible solutions. Each of the prior methods has important shortcomings. We have presented an algorithm for finding a solution that is both smooth and faithful to the data. We have noted how the dependence of the errors in the data can be modeled and have shown how to transform the data to bias the solutions to reflect this dependence. MICRON, a computer program that is based on the work described in this paper and enables one to invert data from any combination of linear instruments, is available from the authors. If smooth solutions are not desirable, if information regarding the magnitude of the errors is lacking, or if significant statistical information is available *a priori* about the size distribution, then the solution presented in this paper will have to be modified to reflect the new conditions.

Appendix

Theorem 5 *C_s is empty if and only if there exists a vector $\begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix}$, that satisfies*

$$\begin{aligned} \|\bar{x}_1\| &= 1 \\ \bar{x}_1^T K V_0 &= 0 \\ \bar{x}_1^T K C^\dagger \; + \bar{x}_2^T U_0^T &\geq 0 \\ \bar{x}_1^T \bar{y}^M \; - \bar{x}_1^T K C^\dagger \bar{b} - \bar{x}_2^T U_0^T \bar{b} &< -s \end{aligned}$$

where $C^\dagger$, U_0, and V_0, are defined in Eq. (2.3).

Note that V_0 is nonzero whenever some planes in R^n are not constrained by C, and this eases the task of finding a solution, or alternatively adds difficulty to the search for $\bar{x}_1$ and $\bar{x}_2$ defined in the theorem. U_0 in the last two equations is present whenever the rows of C are linearly dependent, as, for example, if

$$y_1^L \leq \int k_1(x) f(x)\, dx \leq y_1^U$$

must be satisfied, and reflects in some cases an added difficulty in finding solutions to Eq. (2.5). For example, when $\bar{b} \neq 0$ and the rows of C are linearly dependent, it is possible to have no solutions to Eq. (2.5) *independent* of K and $\bar{y}^M$.

Proof: The equations

$$\left.\begin{aligned} K\bar{f} &= \bar{y}^M \\ C\bar{f} &\geq \bar{b} \end{aligned}\right\} \tag{2.18}$$

can be written

$$\begin{aligned} K\bar{f} &= \bar{y}^M \\ [U_p, U_0]\begin{bmatrix} \Sigma_p, & 0 \\ 0, & 0 \end{bmatrix}\begin{bmatrix} V_p^T \\ V_0^T \end{bmatrix}\bar{f} &\geq \bar{b} \end{aligned}$$

which has a solution if and only if

$$\begin{aligned} K[V_p, V_0]\begin{pmatrix} \bar{f}_p \\ \bar{f}_0 \end{pmatrix} &= \bar{y}^M \\ [U_p, U_0]\begin{bmatrix} \Sigma_p, & 0 \\ 0, & 0 \end{bmatrix}\begin{pmatrix} \bar{f}_p \\ \bar{f}_0 \end{pmatrix} &\geq \bar{b}. \end{aligned}$$

$\bar{f}_0$ is not constrained by the previous equation and can be replaced by

$$\bar{f}_0 = \bar{f}_{0,1} - \bar{f}_{0,2}$$

$$\bar{f}_{0,1} \geq 0 \quad \bar{f}_{0,2} \geq 0$$

Thus Eq. (2.18) has a solution if and only if

$$\begin{aligned} [KV_p, KV_0, -KV_0]\begin{pmatrix} \bar{f}_p \\ \bar{f}_{0,1} \\ \bar{f}_{0,2} \end{pmatrix} &= \bar{y}^M \\ \begin{bmatrix} U_p, & U_0, & 0, & 0 \\ 0, & 0, & I, & 0 \\ 0, & 0, & 0, & I \end{bmatrix}\begin{bmatrix} \Sigma_p, & 0, & 0 \\ 0, & 0, & 0 \\ 0, & I, & 0 \\ 0, & 0, & I \end{bmatrix}\begin{pmatrix} \bar{f}_p \\ \bar{f}_{0,1} \\ \bar{f}_{0,2} \end{pmatrix} &\geq \begin{pmatrix} \bar{b} \\ 0 \\ 0 \end{pmatrix} \end{aligned}$$

These are equivalent to the following pair of equations

$$\begin{aligned} \begin{bmatrix} KV_p, & 0, & KV_0, & -KV_0 \\ 0, & I, & 0, & 0 \end{bmatrix}\begin{pmatrix} \bar{f}_p \\ \bar{f}_{0,0} \\ \bar{f}_{0,1} \\ \bar{f}_{0,2} \end{pmatrix} &= \begin{pmatrix} \bar{y}^M \\ 0 \end{pmatrix} \\ \begin{bmatrix} U_p, & U_0, & 0, & 0 \\ 0, & 0, & I, & 0 \\ 0, & 0, & 0, & I \end{bmatrix}\begin{bmatrix} \Sigma_p, & 0, & 0, & 0 \\ 0, & I, & 0, & 0 \\ 0, & 0, & I, & 0 \\ 0, & 0, & 0, & I \end{bmatrix}\begin{pmatrix} \bar{f}_p \\ \bar{f}_{0,0} \\ \bar{f}_{0,1} \\ \bar{f}_{0,2} \end{pmatrix} &\geq \begin{pmatrix} \bar{b} \\ 0 \\ 0 \end{pmatrix} \end{aligned}$$

or $\mathcal{C}_0$ is empty if and only if

$$\begin{bmatrix} KC^{\dagger}, & KV_0, & -KV_0 \\ U_0^T, & 0, & 0 \end{bmatrix} \bar{f}_2 = \begin{pmatrix} \bar{y}^M \\ 0 \end{pmatrix} - \begin{pmatrix} KC^{\dagger}\bar{b} \\ U_0^T\bar{b} \end{pmatrix}$$
$$\bar{f}_2 \geq 0$$

has no solution, or $\mathcal{C}_s$ is empty if and only if

$$\begin{bmatrix} KC^{\dagger}, & KV_0, & -KV_0 \\ U_0^T, & 0, & 0 \end{bmatrix} \bar{f}_2 = \begin{pmatrix} (\bar{y}^M + \bar{s}) \\ 0 \end{pmatrix} - \begin{pmatrix} KC^{\dagger}(\bar{b} + \bar{s}) \\ U_0^T\bar{b} \end{pmatrix}$$
$$\bar{f}_2 \geq 0$$

has no solution for all $\|\bar{s}\| \leq s$. If $\mathcal{C}_s$ is empty, the cutting plane theorem guarantees there exists $\begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix}$ independent of $\bar{s}$ that satisfies

$$\begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix}^T \begin{bmatrix} KC^{\dagger}, & KV_0, & -KV_0 \\ U_0^T, & 0, & 0 \end{bmatrix} \geq 0$$
$$\begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix}^T \begin{bmatrix} (\bar{y}^M + \bar{s}) - KC^{\dagger}\bar{b} \\ -U_0^T\bar{b} \end{bmatrix} < 0$$

Setting $\bar{s} = \bar{x}_1$ we find if $\mathcal{C}_s$ is empty then the inequalities listed in the theorem statement hold.

Conversely if $\mathcal{C}_s$ is not empty, then there exists $\bar{s}$ and $\bar{f}_2$ such that Eq. (2.18) is satisfied. Thus if Eq. (2.18) holds for $\begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix}$ then

$$\begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix} \begin{pmatrix} (\bar{y}^M + \bar{s}) - KC^{\dagger}\bar{b} \\ -U_0^T\bar{b} \end{pmatrix} \geq 0$$

must also hold. □

Theorem 6 *Assume $f_p(x) \in \mathcal{C}_s$, $\mathcal{C}_{\hat{s}} = \emptyset$ whenever $\hat{s} < s$, and let $\hat{C}$ be the matrix of constraints that are active at f_p. Then $\mathcal{C}_s$ contains a unique element if and only if*

$$K\bar{f} = 0,\ \hat{C}\bar{f} \geq 0 \ \textit{ implies } \ f = 0. \tag{2.19}$$

Proof: $\mathcal{C}_s$ is convex, and $\mathcal{C}_{\hat{s}} = \emptyset$ implies all elements of $\mathcal{C}_s$ have the form $f_p + f_0$ where $Kf_0 = 0$. If Eq. (2.19) holds then

$$\hat{C}(f_p + f_0) = \hat{\bar{b}} + \hat{C}f_0 \geq \hat{\bar{b}},$$

which implies $f_0 = 0$, or f_p is unique. Similarly if f_p is unique then all vectors of the form

$$f_p \; + \epsilon f_0, \quad \|f_0\| = 1, \quad \epsilon > 0$$

must violate at least one constraint. For small ϵ the inactive constraints remain inactive, or $\hat{C} f_0 \not\geq 0$ if $f_0 \neq 0$. □

Theorem 7 *$\mathcal{C}_{s_E}$ is empty if and only if $\mathcal{C}_\infty$ is empty. Additionally, $\mathcal{C}_{s_E}$ contains a unique element if and only if $\mathcal{C}_\infty$ contains a unique element.*

Proof: The first statement is clear from the definitions. The second statement follows from the convexity of $\mathcal{C}_\infty$ and the mean value theorem. □

Theorem 8 *Let $\mathcal{L}$ be an interval of positive λ, and assume that the relationship between λ and $\mathbf{f}_\lambda$ on $\mathcal{L}$ is one to one. Then the functions $J_2(f_\lambda)$ and $R(\mathbf{f}_\lambda)$ are continuous, and $R(\mathbf{f}_\lambda)$ [$J_2(f_\lambda)$] is monotone increasing [decreasing] on $\mathcal{L}$.*

Note that the existence of $\mathcal{L}$ is not guaranteed.
Proof: q defined by Eq. (2.7) is a convex function on the convex sets $\mathcal{C}_\infty$ and $\mathcal{L}$, and this implies the continuity of $\mathbf{f}_\lambda$ as a function of λ, or $J_2(f_\lambda)$ and $R(\mathbf{f}_\lambda)$ are continuous as functions of λ. Observe also that for $\lambda_2 > \lambda_1$, the definition of $\mathbf{f}_\lambda$ implies

$$q(\lambda_2, f_{\lambda_1}) > q(\lambda_2, f_{\lambda_2}) > q(\lambda_1, f_{\lambda_2}) > q(\lambda_1, f\lambda_1).$$

These inequalities lead directly to the monotonicity of $R(\lambda)$ and $J_2(f_\lambda)$.

List of Figures

Figure 7. The inverted solutions obtained using the regularization algorithm described in this paper and STWOM as more data are used in the inversion. ——— represents the true distribution; random error was added to each measurement while assuming $\sigma_i = 0.15y_i$.

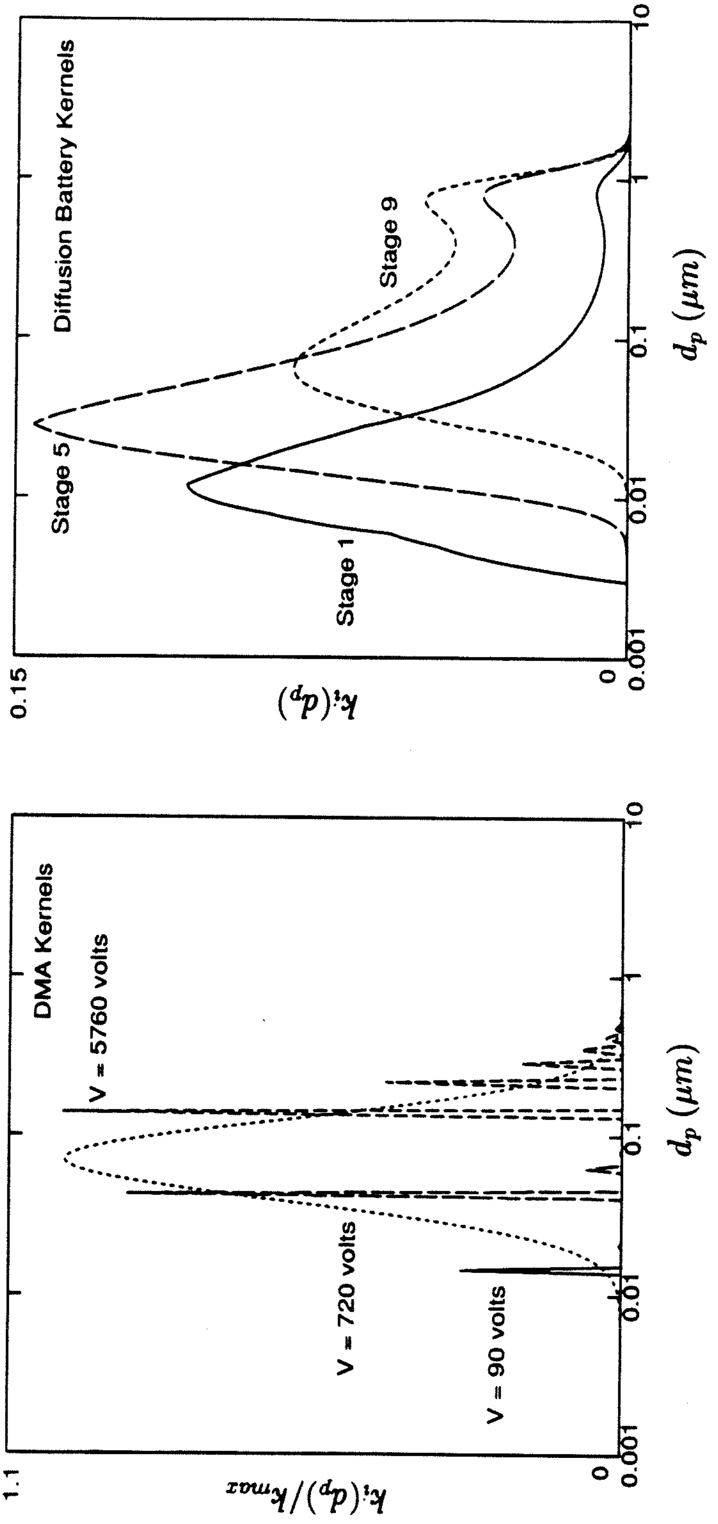

Diffusion Battery Kernels
Stage 1
Stage 5
Stage 9
0.15
0
$k_i(d_p)$
0.001
0.01
0.1
1
10
$d_p\ (\mu m)$
DMA Kernels
V = 5760 volts
V = 720 volts
V = 90 volts
1.1
0
$k_i(d_p)/k_{max}$
0.001
0.01
0.1
1
10
$d_p\ (\mu m)$

Figure 1

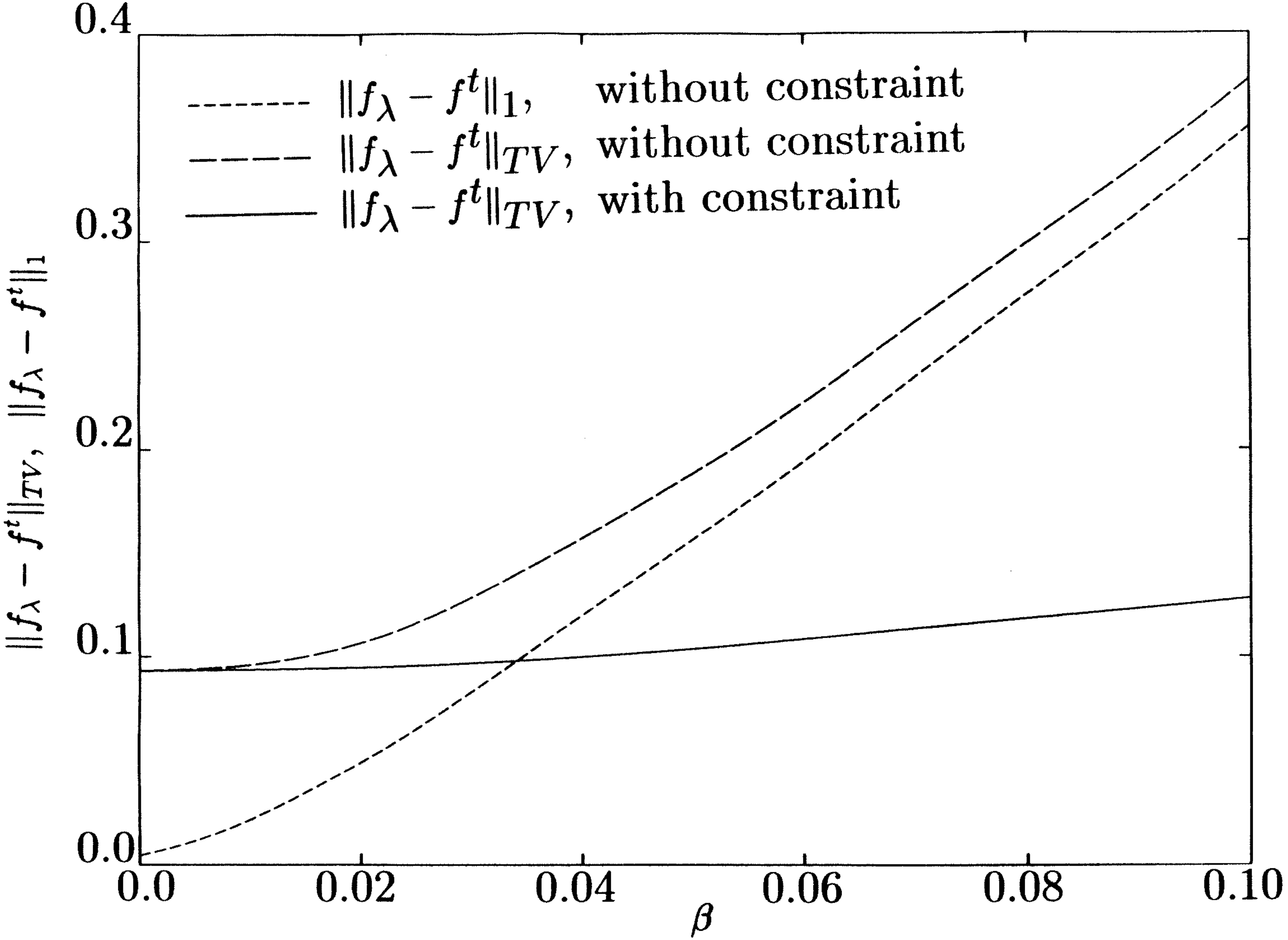

$\|f_\lambda - f^t\|_1$, without constraint
$\|f_\lambda - f^t\|_{TV}$, without constraint
$\|f_\lambda - f^t\|_{TV}$, with constraint
$\|f_\lambda - f^t\|_{TV}$, $\|f_\lambda - f^t\|_1$
0.4
0.3
0.2
0.1
0.0
0.0
0.02
0.04
0.06
0.08
0.10
β

Figure 2

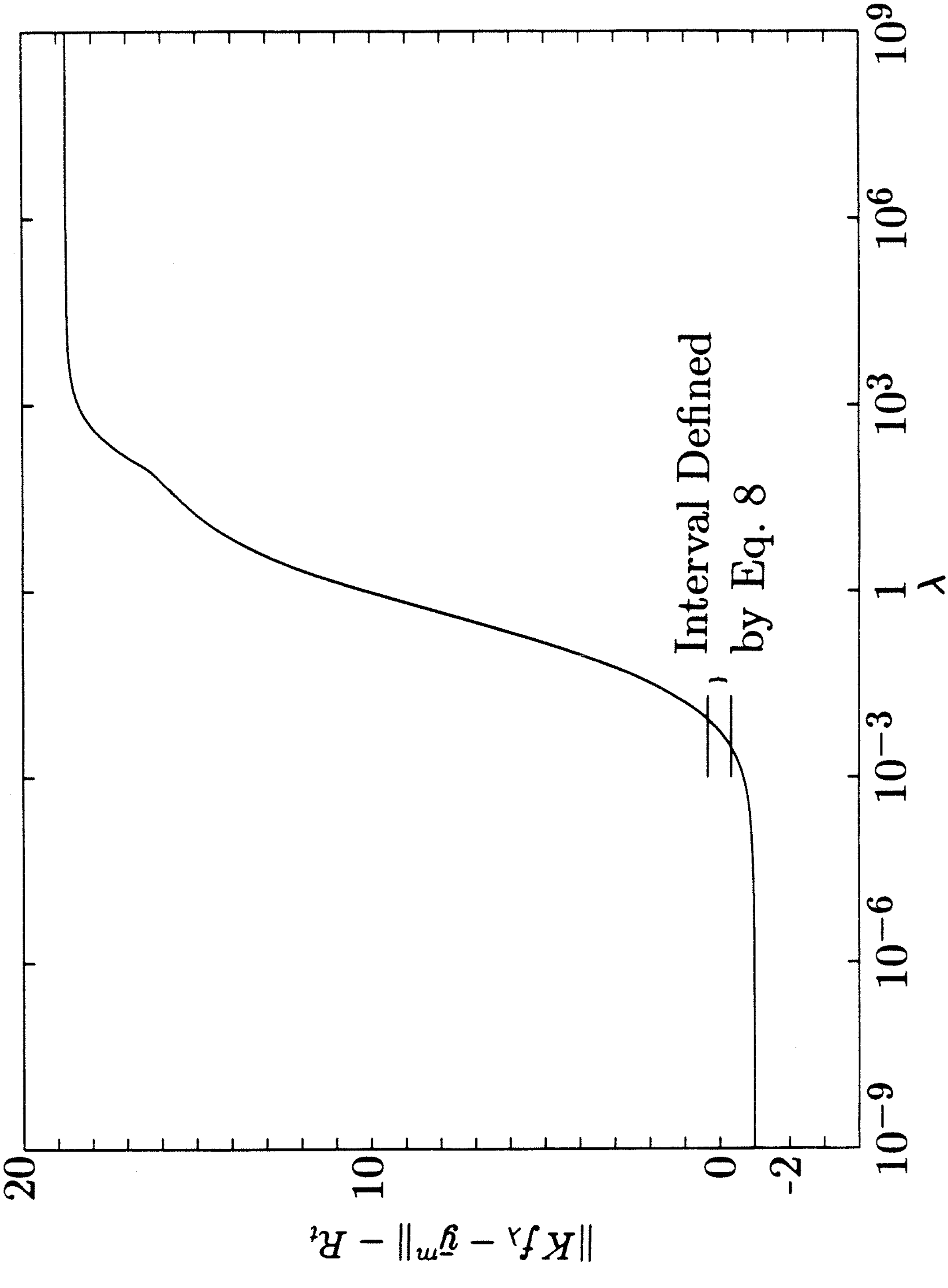

$\|Kf_\lambda - \underline{y}^m\| - R_t$
20
10
0
-2
10^{-9}
10^{-6}
10^{-3}
1
10^{3}
10^{6}
10^{9}
λ
Interval Defined
by Eq. 8

Figure 3

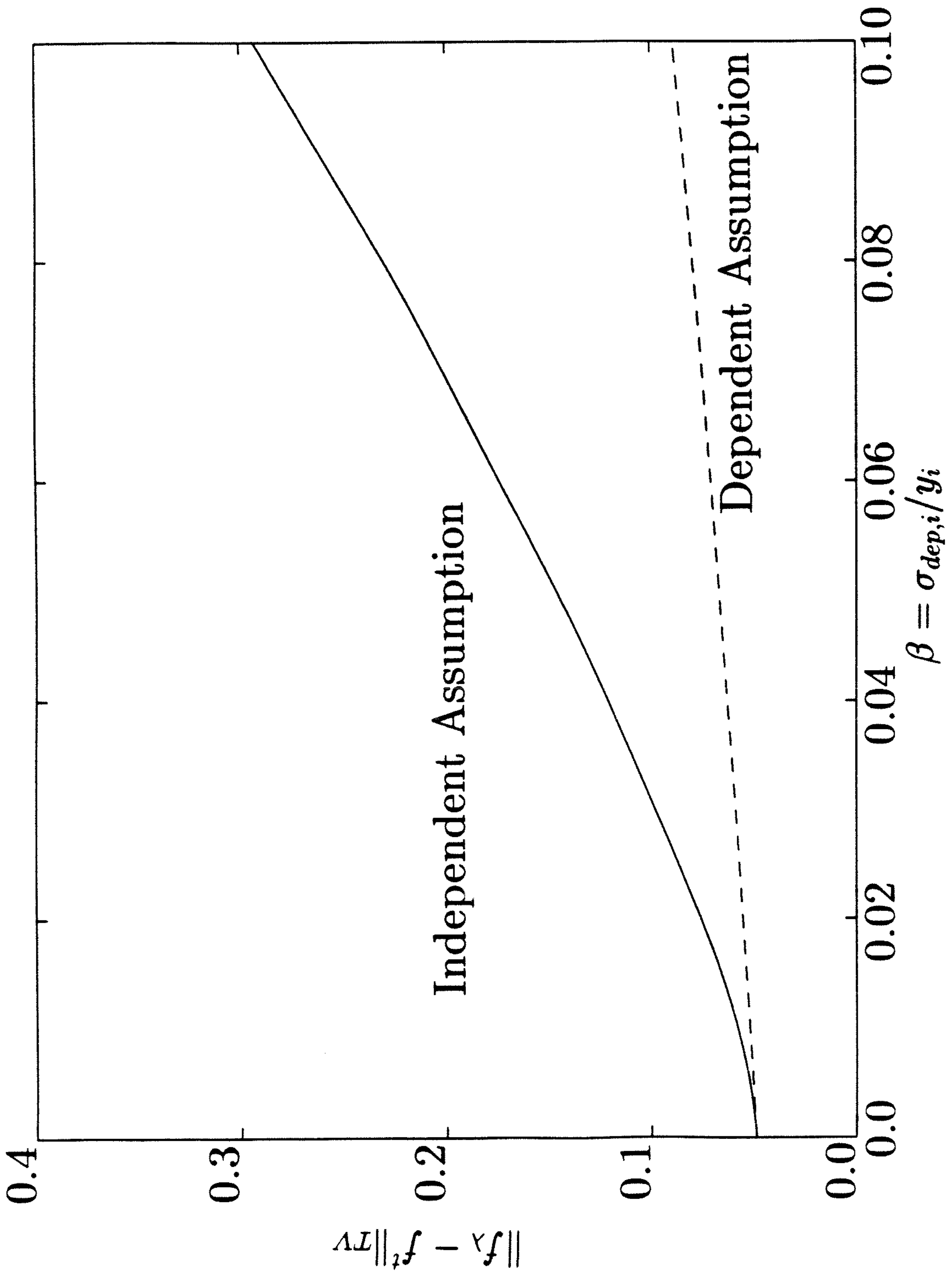

0.4
0.3
0.2
0.1
0.0
$\|f_\lambda - f_t\|_{TV}$
0.0
0.02
0.04
0.06
0.08
0.10
$\beta = \sigma_{dep,i}/y_i$
Independent Assumption
Dependent Assumption

Figure 4

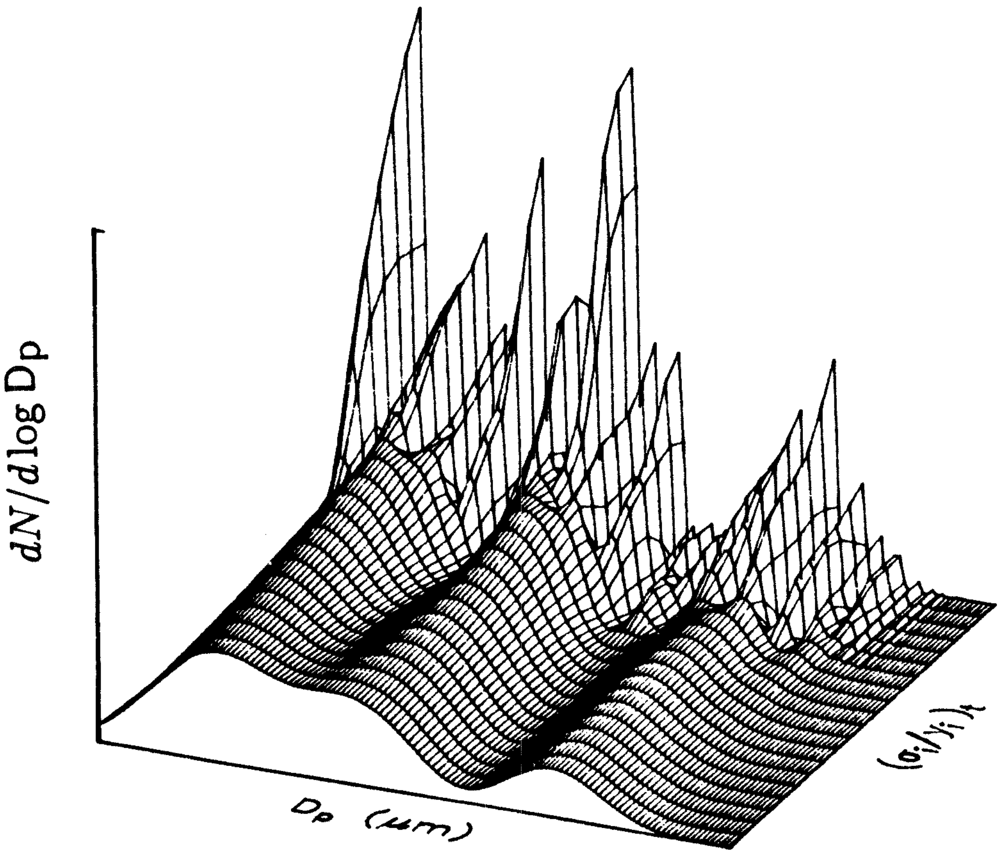


Figure 5

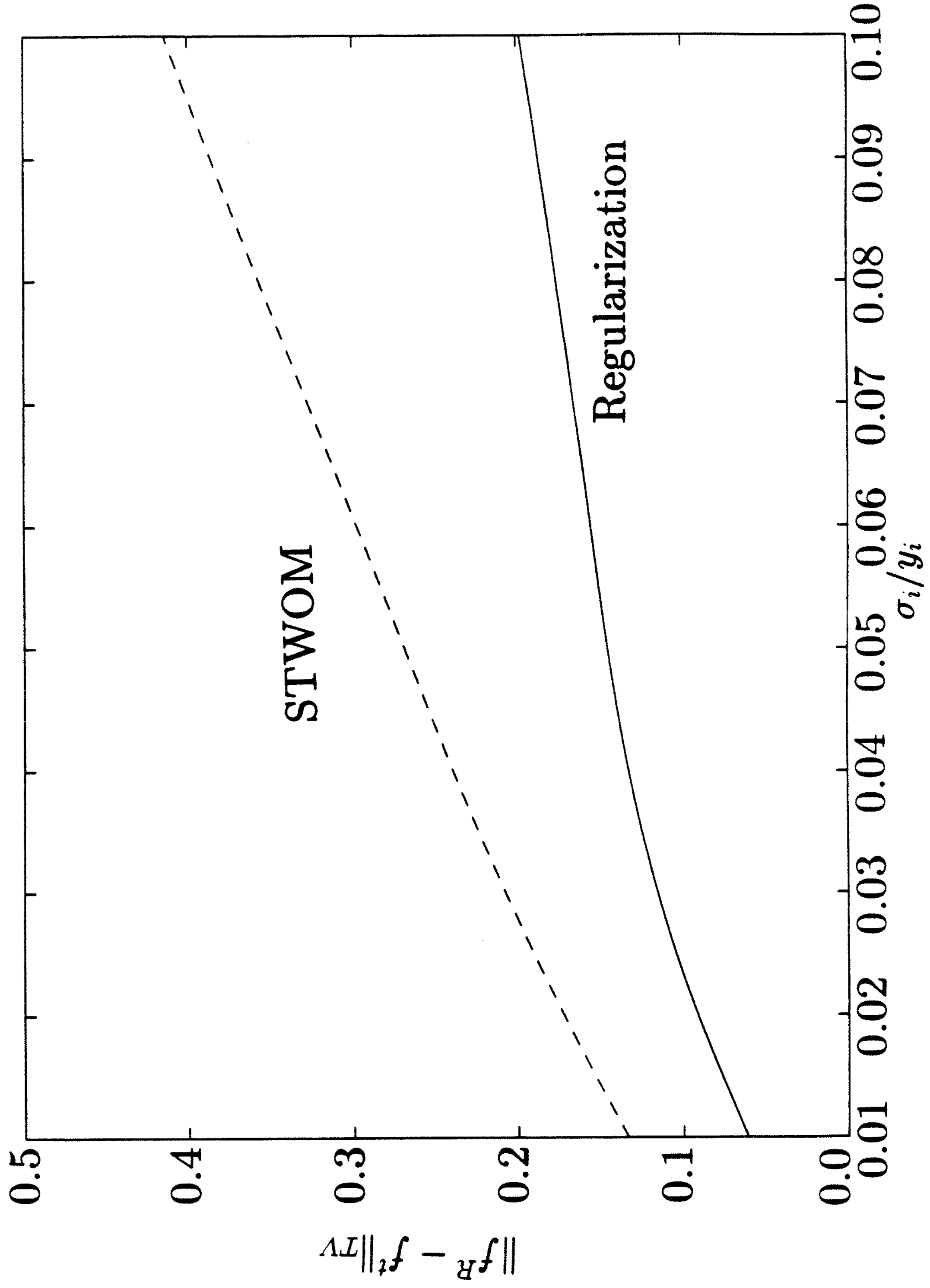

STWOM
Regularization
0.0
0.1
0.2
0.3
0.4
0.5
$\|f^R - f^t\|_{TV}$
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10
σ_i/y_i

Figure 6

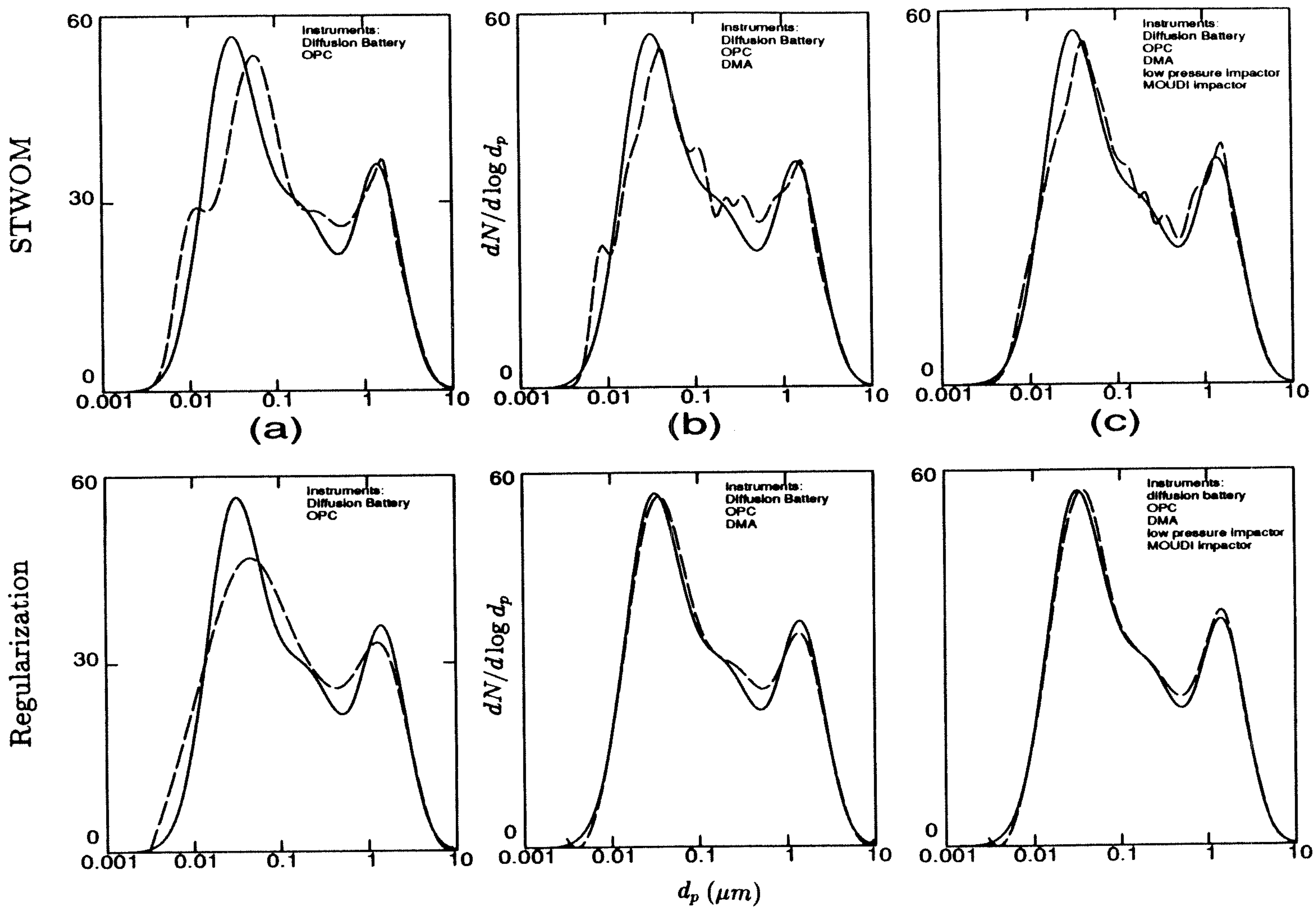

STWOM
Regularization
$dN/d\log d_p$
$dN/d\log d_p$
60
30
0
0.001
0.01
0.1
1
10
(a)
(b)
(c)
Instruments:
Diffusion Battery
OPC
Instruments:
Diffusion Battery
OPC
DMA
Instruments:
Diffusion Battery
OPC
DMA
low pressure impactor
MOUDI impactor
Instruments:
diffusion battery
OPC
DMA
low pressure impactor
MOUDI impactor
d_p (μm)

Figure 7

Chapter 3

Regularized Solutions to the Aerosol Data Inversion Problem

Abstract

Regularized solutions to the aerosol data inversion problem are presented. An approximate form of generalized cross validation is developed that is applicable to this linearly constrained inverse problem. The results obtained with this algorithm for choosing the smoothing parameter are compared with those obtained by the method of discrepancy and by minimizing an unbiased estimate of the inverted errors. Examples are presented that demonstrate the importance of using generalized cross validation to choose the smoothing parameter when the magnitude of the errors in the data is difficult to estimate.

3.1 Introduction

We consider the problem of determining a suitable aerosol size distribution, $f(x)$, from a set of p measurements, $\mathbf{y}^M$. Here, the product $f(x)\,dx$ can represent the mass, surface area, or number of particles in the size interval $(x, x+dx)$. Since the diameter of aerosol particles can range over several orders of magnitude, it is convenient to define particle size by

$$x = \frac{\log(d/d_1)}{\log(d_2/d_1)} \tag{3.1}$$

where d is the particle diameter, and (d_1, d_2) is the interval of particle diameters over which the solution will be computed. For conventional aerosol instruments the relationship between the aerosol size distribution and the data is

$$y_i^M = \int_0^1 k_i(x) f(x)\,dx + \varepsilon_i^t \quad i = 1, \ldots, p \tag{3.2}$$

where $k_i(x)$ is commonly referred to as the kernel function and is proportional to the known response of the instrument to a monodisperse aerosol sample of size x. Note that the interval (d_1, d_2) must be defined so that the support of the product $k_i(x)f(x)$ lies in the interval $(0, 1)$. We will assume that the errors in the data, $\boldsymbol{\varepsilon}^t$, are independent normal random variables. This assumption is rarely valid; however, as discussed in [47], if one can represent $\boldsymbol{\varepsilon}^t$ as a linear combination of independent normal random variables, error sources for example, then Eq. (3.2) can be transformed so that the resulting $\boldsymbol{\varepsilon}^t$ are independent normal random variables with unit variance.

Further information about the size distribution can be provided by linear inequality constraints on $f(x)$ that arise because the size distribution is nonnegative, bounds may exist for the true values of the measured data, the value of the distribution may be known at certain xs (e.g., the endpoints), and as discussed in [47], certain linear combinations of data may be known.

The problem of finding solutions to Eq. (3.2) is ill-posed: for a given $\mathbf{y}^M$ and $\boldsymbol{\varepsilon}^t$, a solution may not exist, and if a solution exists then it is usually not unique. When the number of data is large, one may attempt to eliminate the ill-posedness by approximating Eq. (3.2) by an invertible system of linear equations, but this approach has long been known to yield unstable results [38].

Regularization is a powerful technique for finding solutions to the ill-posed Eq. (3.2). Here, the set of solutions defined by Eq. (3.2) is replaced by the set of smooth solutions, $f_\lambda(x)$, $\lambda \geq 0$, that minimize

$$\sum_{i=0}^{p}\Big(\int_0^1 k_i(x) f(x)\,dx - y_i^M\Big)^2 + \lambda J(f) \tag{3.3}$$

subject to constraints. $J(f)$ is the regularizing functional that penalizes undesirable solutions. For example, in this paper we use

$$J(f) = \int_0^1 (f^{(m)}(x))^2 \, dx \tag{3.4}$$

since high frequency oscillations are undesirable. The unspecified λ measures the relative importance of the solution's smoothness and its fidelity to the data. If λ is too small then smoothness becomes unimportant, and the solution will be too sensitive to perturbations in the data and will likely exhibit unrealistic oscillations. On the other hand, if λ is too large then the structure that exists in the true distribution will be suppressed.

One technique for choosing λ that we have used with some success and that is straightforward to apply to this constrained inversion is the method of discrepancy [12]: set $\lambda = \lambda_D$ where

$$\|\mathbf{y}^R_{\lambda_D} - \mathbf{y}^M\|^2 = \mathbf{E}\|\mathbf{y}^t - \mathbf{y}^M\|^2. \tag{3.5}$$

and where the superscript t denotes true, the superscript R denotes predicted, boldface variables represent column vectors, and $\|\cdot\|$ denotes the Euclidean norm. Wahba and others [44] have presented theoretical and numerical evidence on related inversion problems that shows λ_D tends to oversmooth $f_\lambda(x)$. The weakness of choosing the solution to match the statistics of the error in the data is made clear in the following example presented in [47]. If

$$y_i^t - y_i^M = (\varepsilon_i + \varepsilon p + 1) y_i^t \qquad i = 1, \ldots, p$$

where ε_i is an independent normal random variable with a variance of σ^2, then one can show that

$$\lambda_D \longrightarrow \infty \quad \text{as} \quad \sigma p \longrightarrow 1$$

independent of instruments and the data. Here λ_D is a poor regularization parameter: it overestimates the amount of smoothing, and the oversmoothing becomes worse as more data are included in the inversion.

An additional difficulty with using λ_D is that the standard deviations of the errors in the data are usually not well known, and a poor estimate of the magnitude of these errors can easily lead to unrealistic solutions [47].

In this paper we use methods similar to those of Kimeldorf and Wahba [26], and Villalobos and Wahba [43] to find solutions to the aerosol data inversion problem subject to inequality constraints. Smoothing parameters are chosen to minimize an unbiased estimate of the inverted errors, and by generalized cross validation (GCV) when the number of data is large. We present examples that demonstrate the importance of taking the constraints into consideration when using GCV to choose λ. Additional examples compare the solutions obtained while using λ_D with those obtained for the regularization parameters described in this paper, and demonstrate the importance of using GCV as one's ability to estimate the magnitude of the errors diminishes.

3.2 Solution to the regularized problem

The solution to Eq. (3.2) is not easy to find because the subset $\{f(x) : f(x) \geq 0\}$, of $\mathcal{L}^2(0,1)$ for example, defines an infinite cone, and minimization techniques are not well developed on this type of set.

One approach we can use is described in [43] in relation to multi-dimensional interpolation. Here, we assume $f(x)$ lies in the Sobolev space $W_2^m(0,1)$, where $f^m(x) \in \mathcal{L}^2(0,1)$ and $f^{(i)}(x)$, $i = 1, \ldots, m-1$, is absolutely continuous [1]. The inner product is defined as

$$\langle f, g \rangle = \sum_{i=0}^{m-1} f^{(i)}(0) g^{(i)}(0) + \int_0^1 f^{(m)}(x) g^{(m)}(x)\, dx \tag{3.6}$$

The constraint that the solution be nonnegative is replaced with the finite set of constraints:

$$f(x_i) \geq 0 \quad i = 1, \ldots, \bar{n} \;\leq\; n \tag{3.7}$$

so that there are a finite number of linear inequality constraints, and we write these as

$$c_i(f) \geq b_i \quad i = 1, \ldots, n \tag{3.8}$$

If $T(f)$ is a linear functional on $W_2^m(0,1)$, then the Riesz representation theorem [1] guarantees the existence of the representer of T such that

$$T(f) = \langle \theta, f \rangle \tag{3.9}$$

for all $f \in W_2^m(0,1)$. For example, if

$$T(f) = \int_0^1 t^0(x) f(x)\, dx \tag{3.10}$$

then integration by parts and the independence on f shows that θ satisfies the differential equation

$$\theta^{(2m)}(x) = (-1)^m t^0(x) \tag{3.11}$$

with the boundary conditions

$$(-1)^i\theta^{(m+i)}(0) - \theta^{(m-1-i)}(0) = \theta^{(m+i)}(1) = 0 \quad i = 0,\dots,m-1 \tag{3.12}$$

One readily obtains

$$(-1)^m\theta(x) = t^{2m}(x) + \sum_{i=0}^{m-1}\frac{x^i}{i!}(t^{i-1}(0) + (-1)^{m-k}t^{2m-i}(0)) \tag{3.13}$$

where

$$t^i(x) = \int_x^1 t^{i-1}(s)\,ds \quad i = 1,\dots,2m \tag{3.14}$$

Thus, if we define the representers $\boldsymbol{\kappa}(x)$ and $\boldsymbol{\gamma}(x)$ by

$$\langle f, \kappa_i\rangle = \int_0^1 k_i(x)f(x)\,dx \tag{3.15}$$

$$\langle f, \gamma_i\rangle = c_i(f) \tag{3.16}$$

then $f_\lambda(x)$, the minimizer of Eq. (3.3), is also the minimizer of

$$\sum_{i=0}^{p}(\langle\kappa_i, f\rangle - y_i^M)^2 + \lambda\langle P_1 f, P_1 f\rangle \tag{3.17}$$

subject to

$$\langle\gamma_i, f\rangle = b_i \quad i = 1,\dots,n \tag{3.18}$$

where P_1 is the projection on $W_2^m(0,1)$ defined by

$$P_1(f(x)) = f(x) - \sum_{i=0}^{m-1}\frac{f^{(i)}(0)x^i}{i!} \tag{3.19}$$

We denote the range and kernel of P_1 on $W_2^m(0,1)$ by $H_0(0,1)$ and $H_1(0,1)$, respectively.

There exists coefficients k, c, and w such that the minimizer of Eq. (3.3) as well as any element of $W_2^m(0,1)$ can be written as

$$f_\lambda(x) = \sum_{i=0}^{p}\mathsf{k}_i P_1\kappa_i(x) + \sum_{i=0}^{n}\mathsf{c}_i P_1\gamma_i(x) + \sum_{i=0}^{m-1}\mathsf{w}_i\omega_i(x) + f_\perp(x) \tag{3.20}$$

where

$$\omega_i(x) = \frac{x^{i-1}}{i!} \quad i = 1, \ldots, m \tag{3.21}$$

is a basis for $H_0(0,1)$, and where $f_\perp(x) \in H_1(0,1)$ is orthogonal to all $\kappa_i(x)$ and $\gamma_i(x)$. This representation may not be unique however because the set of representers $\{\boldsymbol{\gamma}(x), \boldsymbol{\kappa}(x)\}$ can be linearly independent. For example, we may require the solution to satisfy

$$b_{Li} \leq \int_0^1 k_i(x) f(x)\, dx \leq b_{Ui} \tag{3.22}$$

Therefore, it is desirable to identify a subset of the representers, $\{\hat{\boldsymbol{\gamma}}(x), \hat{\boldsymbol{\kappa}}(x)\}$, with $\hat{p}+\hat{n} \leq p+n$ elements, such that $\{P_1\hat{\boldsymbol{\gamma}}, P_1\hat{\boldsymbol{\kappa}}\}$ are linearly independent, and such that the span of $\{P_1\hat{\boldsymbol{\gamma}}, P_1\hat{\boldsymbol{\kappa}}\}$ equals the span of $\{P_1\boldsymbol{\gamma}, P_1\boldsymbol{\kappa}\}$. With $\{\hat{\boldsymbol{\gamma}}(x), \hat{\boldsymbol{\kappa}}(x)\}$ identified, elements of $W_2^m(0,1)$ can be written uniquely as

$$f_\lambda(x) = \sum_{i=0}^{p} \hat{\mathsf{k}}_i P_1 \hat{\kappa}_i(x) + \sum_{i=0}^{n} \hat{\mathsf{c}}_i P_1 \hat{\gamma}_i(x) + \sum_{i=0}^{m-1} \mathsf{w}_i \omega_i(x) + f_\perp(x) \tag{3.23}$$

Substituting Eq. (3.23) into Eq. (3.17) shows that $f_\perp(x) = 0$ and that $\hat{\mathsf{k}}$, $\hat{\mathsf{c}}$, and w minimize

$$\|\mathcal{K}\mathsf{x} - \mathbf{y}^M\|^2 + \lambda \|\mathcal{D}\mathsf{x}\|^2 \tag{3.24}$$

subject to

$$\mathcal{C}\mathsf{x} \geq \mathbf{b} \tag{3.25}$$

where

$$\begin{aligned}
\mathcal{K} &= [\langle P_1\boldsymbol{\kappa}, P_1\hat{\boldsymbol{\kappa}}^T\rangle, \langle P_1\boldsymbol{\kappa}, P_1\hat{\boldsymbol{\gamma}}^T\rangle, \langle P_0\boldsymbol{\kappa}, \boldsymbol{\omega}^T\rangle] && (3.26)\\
\mathcal{D} &= [\langle P_1\hat{\boldsymbol{\kappa}}, P_1\hat{\boldsymbol{\kappa}}^T\rangle, \langle P_1\hat{\boldsymbol{\gamma}}, P_1\hat{\boldsymbol{\gamma}}^T\rangle, 0] && (3.27)\\
\mathcal{C} &= [\langle P_1\boldsymbol{\gamma}, P_1\hat{\boldsymbol{\kappa}}^T\rangle, \langle P_1\boldsymbol{\gamma}, P_1\hat{\boldsymbol{\gamma}}^T\rangle, \langle P_0\boldsymbol{\gamma}, \boldsymbol{\omega}^T\rangle] && (3.28)\\
\mathsf{x} &= \begin{pmatrix} \hat{\mathsf{k}} \\ \hat{\mathsf{c}} \\ \mathsf{w} \end{pmatrix} && (3.29)
\end{aligned}$$

Here, $\langle P_1\boldsymbol{\gamma}, P_1\boldsymbol{\kappa}^T\rangle$, for example, represents a matrix with ij^{th} component equal to $\langle P_1\gamma_i, P_1\kappa_j\rangle$, and if A is an $n \times m$ matrix and B is an $n \times p$ matrix then $[A, B]$ denotes the $n \times m + p$ augmented matrix with the first m columns corresponding to A and the last p columns corresponding to B.

If the feasible set defined by Eq. (3.25) is not empty, then a minimizer of Eq. (3.24) must exist. If the rank of the matrix $\langle \boldsymbol{\omega}, \boldsymbol{\kappa}^T \rangle$ is m, then it is straightforward to show that the minimizer is unique; this condition is sufficient though not necessary. A more detailed discussion on the existence of a unique solution can be found in [47].

3.2.1 Some numerical considerations

To calculate the elements of these matrices, $\langle P_1\kappa_i, P_1\kappa_j \rangle$, for example, Eq. (3.6) and Eq. (3.13) are not suitable for numerical calculations, and simpler expressions are readily obtained for a general linear functional by introducing the reproducing kernel for $W_2^m(0,1)$ [26]. The reproducing kernel, $K(x,y)$, for the Hilbert space H satisfies:

1. $K_x(y) \in H$.
2. $K_x(y) = K_y(x)$.
3. $\langle K_x(y), u(y) \rangle = u(x)$.

where we write $K_x(y)$ to mean that $K(x,y)$ should be viewed as a function of y. From this definition one can see the representer $\theta_i(x)$ for the linear functional T_i satisfies

$$\theta_i(x) = T_i K_x(y) \tag{3.30}$$

and

$$\langle \theta_i, \theta_j \rangle = T_i T_j K(x,y) \tag{3.31}$$

Note also that if K_i is the reproducing kernel for H_i where $W_2^m(0,1) = \sum_\oplus H_i$, then one can show [22] that $K = \sum K_i$.

From the definitions we can write

$$\begin{aligned} \int_0^1 u(y)\delta_x(y)\,dy &= \langle K_{1;x}(y), u(y) \rangle & (3.32) \\ &= \int_0^1 K_{1;x}^{(m)}(y) u^{(m)}(y)\,dy & (3.33) \end{aligned}$$

where $\delta_x(y)$ is the dirac-delta function. Integration by parts plus the independence of $K_{1;x}(y)$ on $u(y)$ shows that $K_{1;x}(y)$ satisfies the distributional equation

$$K_{1;x}^{(2m)}(y) = (-1)^m \delta_x(y) \tag{3.34}$$

subject to the boundary conditions

$$K_{1;x}^{(m+i)}(1) = K_{1;x}^{(i)}(0) = 0 \quad i = 0, \ldots, m-1 \tag{3.35}$$

or

$$K_1(x,y) = \sum_{i=0}^{m-1} \frac{(-1)^k x^{m-1-i} y^{m+i}}{(m-1-i)!(m+i)!} \qquad x \geq y \tag{3.36}$$

One can easily verify that

$$K_0(x,y) = \sum_{i=0}^{m-1} \frac{x^i y^i}{2(i!)} \tag{3.37}$$

Thus

$$\begin{aligned} \langle P_1\kappa_i, P_1\kappa_j\rangle &= \int_0^1 \int_0^x (k_i(x)k_j(y) + k_i(y)k_j(x)) \times \\ & \sum_{\nu=0}^{m-1} \frac{(-1)^\nu x^{m-1-\nu} y^{m+\nu}}{(m-1-\nu)!(m+\nu)!}\, dx\, dy \end{aligned} \tag{3.38}$$

or, if it is not efficient to keep the sum inside the integral, one can show that

$$\begin{aligned} \langle P_1\kappa_i, P_1\kappa_j\rangle &= \frac{(-1)^m}{(2m-1)!} \int_0^1 \int_0^x k_i(x)k_j(y)(x-y)^{2m-1}\, dx\, dx \\ & + \sum_{\nu=0}^{m-1} \frac{(-1)^\nu M_{ki}^{m-\nu-1} M_{kj}^{m+\nu}}{(m-\nu-1)!(m+\nu)!} \end{aligned} \tag{3.39}$$

where M_{ki}^j is the j^{th} moment of $k_i(x)$ over the interval $(0,1)$. Efficient algorithms for these 2-D integrals have been developed [14,25,29]. Also, if $\gamma_i(x)$ is the evaluation functional then

$$\begin{aligned} \langle P_1\gamma_i, P_1\kappa_j\rangle &= \frac{(-1)^m}{(2m-1)!} \int_0^{x_i} k_j(y)(x_i-y)^{2m-1}\, dy \\ & + \sum_{\nu=0}^{m-1} \frac{(-1)^k x_i^{m+\nu} M_{kj}^{m-1-\nu}}{(m-1-\nu)!(m+\nu)!} \end{aligned} \tag{3.40}$$

We find the minimizer of Eq. (3.24) with the active-set algorithm described in [19,18]. Briefly, a minimum is found by iteratively choosing a set of active constraints, constraints that will be treated as equalities, and performing the minimization while ignoring the remaining inequalities. A single constraint is added and/or deleted per iteration.

The upper triangular matrix, R, which satisfies

$$R^T R = \mathcal{K}^T\mathcal{K} + \lambda\mathcal{D}^T\mathcal{D} = H^0 \tag{3.41}$$

is used in place of the Hessian H^0 to reduce the computation and increase the numerical stability. The straightforward method of computing R is to simply form the

Hessian and then perform the Cholesky decomposition; however, for small λ this approach can lead to a troublesome loss of precision. Instead, if λ is small then we compute upper triangular Δ_R that satisfies

$$R^T R = (\hat{R} + \Delta_R)^T (\hat{R} + \Delta_R) \tag{3.42}$$

where $\hat{R}$ is the upper triangular matrix that satisfies

$$\hat{R}^T \hat{R} = \mathcal{K}^T \mathcal{K} \tag{3.43}$$

Note that $\hat{R}$ will be available if the minimizer of

$$\sum_{i=1}^{p} (\int_0^1 k_i(x) f(x)\, dx - y_i^M)^2 \tag{3.44}$$

is computed. In the Cholesky decomposition, one computes the i^{th} row of R given the first $i-1$ rows by solving

$$R_{i,i} R_{i,j} = H_{i,j}^{i-1} \quad j = i, \ldots, n \tag{3.45}$$

where

$$H^i = H^{i-1} - \mathbf{r}_i \mathbf{r}_i^T \quad i = 1, \ldots, n \tag{3.46}$$

The column vector $\mathbf{r}_i$ corresponds to the i^{th} column of R^T. Similarly, one can compute the i^{th} row of Δ_R given the first $i-1$ rows by solving

$$\Delta_{Ri,i} \Delta_{Ri,j} = D_{i,j}^{i-1} \quad j = i, \ldots, n \tag{3.47}$$

where

$$D^i = D^{i-1} - \hat{\mathbf{r}}_i \Delta_{\mathbf{r}_i}^T - \Delta_{\mathbf{r}_i} \hat{\mathbf{r}}_i^T - \Delta_{\mathbf{r}_i} \Delta_{\mathbf{r}_i}^T \tag{3.48}$$

and

$$D^0 = \lambda \mathcal{D}^T \mathcal{D} \tag{3.49}$$

where the column vector $\hat{\mathbf{r}}_i$ corresponds to the i^{th} column of $\hat{R}^T$, and the column vector $\Delta_{\mathbf{r}_i}$ corresponds to the i^{th} column of $\Delta_R{}^T$. Note that the product $\hat{R}^T \hat{R}$ is not formed. Roughly speaking, we avoid precision loss by working with terms on the order of $\sqrt{\lambda}\mathcal{D}$ relative to $\hat{R}$ in contrast to working with terms on the order of $\lambda \mathcal{D}^T \mathcal{D}$ relative to $\hat{R}^T \hat{R}$.

One feature of the algorithm described in [19] is that the inequality constraints are divided into general constraints and bound constraints:

$$\mathbf{b}_U \leq \begin{bmatrix} I \\ A \end{bmatrix} \mathsf{x} \leq \mathbf{b}_L \tag{3.50}$$

where the matrix I corresponds to the bound constraints and the A corresponds to the general constraints. If, for example, x is a $\hat{p}$-vector, $\hat{m}$ general constraints are active, and $\hat{n}$ bound constraints are active, one can show [19] that $\hat{n}(6\hat{p} - 2\hat{m} - \frac{3}{2}\hat{n})$ multiplications are saved per iteration when a single general constraint is added if the constraints are treated separately instead of treating them all as general constraints.

In the preceding formulation for the solution to Eq. (3.3), note that all of the constraints must be treated as general constraints since the condition $f(x_i) \geq 0$ leads to a matrix with entries given in Eq. (3.28) instead of the identity matrix. The calculations will be expensive if a large number of constraints of this type are chosen and the initial active set chosen is not close to the final active set. Additionally, computing the entries for the matrices $\mathcal{K}$, $\mathcal{D}$, and $\mathcal{C}$ can become expensive. For example, if we have 40 measurements and 40 constraints, then we will need to numerically evaluate 800 double and single integrals.

An undesirable feature of the previous formulation is that for any finite number of constraints of the form $f(x_i) \geq 0$, we always have

$$\sum_{i=1}^{p}\left(\int_0^1 k_i(x) f_\lambda(x)\, dx - y_i^M\right)^2 \longrightarrow 0 \tag{3.51}$$

as $\lambda \longrightarrow 0$, when in fact we encounter cases where the magnitude of

$$\inf_{\mathrm{f} \geq 0} \quad \textstyle\sum_{i=0}^{p}(\int_0^1 k_i(x) f(x)\, dx - y_i^M)^2 \tag{3.52}$$

is significant. This can occur, for example, if

$$\sum_{i \neq j} k_i(x) > k_j(x) \quad \text{and} \quad \sum_{i \neq j} y_i^M \leq \alpha y_j^M \tag{3.53}$$

where $\alpha < 1$. The inability of the previous formulation to predict that Eq. (3.52) can be nonzero could become important if the method used to choose λ depends on the difference between the predicted data and the measured data.

3.2.2 A finite-dimensional solution

An alternative, though similar, approach to solving Eq. (3.3) is to assume that the solution can be approximated as a n-dimensional linear spline. The difficulty associated with this assumption is obvious: how can one be certain that n is large enough to capture all of the information provided by the data? For now we simply assume that n is of the same order of magnitude as p, for example $n = 2p$. Note that in this setting, $f_\lambda(x)$ is able to reflect the lack of a positive solution to Eq. (3.2) for $\boldsymbol{\varepsilon}^t = \mathbf{0}$.

With this assumption we seek $\mathbf{f}_\lambda$, the solution to the following quadratic program:

$$\text{minimize} \quad \mathbf{f}^T({\mathcal{K}_n}^T\mathcal{K}_n + p\lambda {D^{(m)}}^T D^{(m)})\mathbf{f} - 2{\mathbf{y}^M}^T\mathcal{K}_n\mathbf{f} \tag{3.54}$$

$$\text{subject to} \quad C\mathbf{f} \geq b \tag{3.55}$$

where

$$f_\lambda(x) = \mathbf{g}^T(x)\mathbf{f}_\lambda \tag{3.56}$$

and

$$\mathcal{K}_{ni,j} = \int_0^1 g_j(x)k_i(x)\,dx \tag{3.57}$$

$\mathcal{K}_n$ is the matrix that represents the integral operator for the linear splines and $\{g_i(x), i = 1, \ldots, n\}$ is a basis for the n-dimensional linear splines. $D^{(m)}$ is a difference matrix such that if $\mathbf{f}^T\mathbf{g}(x)$ is an approximation to $f(x)$, then

$$\int_0^1 (f^{(m)}(x))^2\,dx \approx \mathbf{f}^T {D^{(m)}}^T D^{(m)}\mathbf{f} \tag{3.58}$$

For example, if the splines are equally spaced we use

$$D^{(1)}_{i,j} = \frac{\delta_{i,i} - \delta_{i,i-1}}{n-1} \tag{3.59}$$

and for m even

$$D^{(m)} = (D^{(1),T}D^{(1)})^{m/2} \tag{3.60}$$

In the remainder of this paper we assume the solution can be approximated as a n-dimensional linear spline. If the data are consistent and few constraints are required to keep the solution positive, then it would be better to keep the solution in $W_2^m(0,1)$; however, the following analysis is similar for both approaches.

One can show that the solution in the absence of constraints, $C = 0$, is

$$D^{(m)}\mathbf{f}_\lambda = {K_D}^T(K_D{K_D}^T + \lambda pI)^{-1}\mathbf{y}^M \tag{3.61}$$

where $K_D = D^{-(m)}\mathcal{K}_n$ and $D^{-(m)}$ is the inverse of $D^{(m)}$. Note that K_D can be computed from $\mathcal{K}_n$ with only $\frac{1}{2}mn(n-1)$ additions and that the inverse defined in Eq. (3.61) is order p instead of the larger n. Following [11] we observe that the predicted data are linearly related to the measured data,

$$\mathbf{y}_\lambda^R = K_D {K_D}^T (K_D {K_D}^T + p\lambda I)^{-1} \mathbf{y}^M = A(\lambda)\mathbf{y}^M. \tag{3.62}$$

If the set of constraints that are satisfied at $\mathbf{f}_\lambda$ are known, then a simple expression for the solution can be found by projecting the quadratic functional onto the null-space of the active constraints. Let the set of linearly independent constraints that are active at the minimizing $\mathbf{f}_\lambda$ be denoted by

$$C_a\mathbf{f} = \mathbf{b}_a, \tag{3.63}$$

then one can show that

$$D^{(m)}\mathbf{f}_\lambda = V_{Ca}{\Lambda_{Ca}}^{-1}{U_C}^T\mathbf{b}_a + {K_{V0}}^T(K_{V0}{K_{V0}}^T + p\lambda I_0)^{-1}\mathbf{y}_0^M \tag{3.64}$$

where

$$\begin{aligned} \left[\ K_{V0}, K_{Va}\ \right] &= K_D \left[\ V_{C0}, V_{Ca}\ \right] \qquad (3.65) \\ \mathbf{y}_0^M &= \mathbf{y}^M - K_{Va}{\Lambda_{Ca}}^{-1}{U_C}^T\mathbf{b}_a \qquad (3.66) \end{aligned}$$

and where the singular value decomposition of C_a is represented by

$$C_a = U_C \left[\ \Lambda_{Ca}, 0\ \right] \begin{bmatrix} {V_{Ca}}^T \\ {V_{C0}}^T \end{bmatrix} \tag{3.67}$$

Here we find an expression similar to Eq. (3.62) for the predicted data,

$$\begin{aligned} \mathbf{y}_0^R &= \mathbf{y}_\lambda^R - K_{Va}{\Lambda_{Ca}}^{-1}{U_C}^T\mathbf{b}_a \qquad (3.68) \\ &= K_{V0}{K_{V0}}^T(K_{V0}{K_{V0}}^T + p\lambda I_0)^{-1}\mathbf{y}_0^M \qquad (3.69) \end{aligned}$$

or we can write

$$\mathbf{y}_0^R = A_0(\lambda)\mathbf{y}_0^M. \tag{3.70}$$

3.3 Minimizing the expected recovered error

If the standard deviation of the errors in the data is known (and in this case equal to 1), then one can calculate an unbiased estimate of the expected errors in the predicted data, $\mathbf{E}R(\lambda)$, and the regularization parameter that minimizes $\mathbf{E}R(\lambda)$, λ_R, can be found.

If $C = 0$ then the analysis in [11] can be applied directly. The expression we want to minimize is

$$\mathbf{E}R(\lambda) = \mathbf{E}\|\mathbf{y}^t - \mathbf{y}^R_\lambda\|^2 \tag{3.71}$$

which, when combined with Eq. (3.2) and Eq. (3.62), yields

$$\mathbf{E}R(\lambda) = \mathbf{E}\|(I - A(\lambda))\mathbf{y}^t\|^2 + \mathbf{tr}A^2(\lambda). \tag{3.72}$$

Observe that an unbiased estimate of $\mathbf{E}R(\lambda)$ is given by $\hat{R}(\lambda)$,

$$\hat{R}(\lambda) = \|(I - A(\lambda))\mathbf{y}^M\|^2 + \mathbf{tr}A^2(\lambda) - \mathbf{tr}(I - A(\lambda))^2. \tag{3.73}$$

The singular value decomposition of K_D,

$$K_D = U_K \Lambda V^T \tag{3.74}$$

can be used to show that Eq. (3.73) is equivalent to

$$\hat{R}(\lambda) = p + \sum_{j=1}^{p} r_i(\lambda)^2 y_{U;i}^{M\,2} - 2r_i(\lambda) \tag{3.75}$$

where

$$\begin{aligned} \mathbf{y}^M_U &= U_K \mathbf{y}^M &(3.76)\\ r_i(\lambda) &= \frac{p\lambda}{\lambda_i + p\lambda} &(3.77)\\ \lambda_i &= \sqrt{\Lambda_{ii}} &(3.78) \end{aligned}$$

Therefore a good choice of λ is λ_R, the minimizer Eq. (3.75), and λ_R is easily found since $\mathbf{y}^M_U$ and λ_i are independent of λ.

If the regularized solution has active constraints, defined by Eq. (3.63), then Eq. (3.70) must be substituted for Eq. (3.62) in the previous analysis. Thus we find

$$\mathbf{E}R(\lambda) = \mathbf{E}\|\mathbf{y}^R_\lambda - \mathbf{y}^t\|^2 \tag{3.79}$$

or

$$\mathbf{E}R(\lambda) = \mathbf{E}\|(I - A_0(\lambda))\mathbf{y}^M_0\|^2 + \|\ \varepsilon^t A_0(\lambda)\|^2 \tag{3.80}$$

Thus an unbiased estimate of $\mathbf{E}R(\lambda)$is $\hat{R}_0(\lambda)$, given by

$$\hat{R}_0(\lambda) = p + \sum_{j=1}^{p} r_{0;i}(\lambda)^2 y_{0;i}^{M2} - 2r_{0;i}(\lambda) \tag{3.81}$$

where

$$\mathbf{y}_0^M = U_{K0}\mathbf{y}^M \tag{3.82}$$

$$r_{0;i}(\lambda) = \frac{p\lambda}{\lambda_{0;i} + p\lambda} \tag{3.83}$$

The minimizer of $\hat{R}_0(\lambda)$ is not as easily found as the minimizer of Eq. (3.75) because $\lambda_{0;i}$ and $\mathbf{y}_0^M$ are functions of the active set that can vary with λ.

The most common case in aerosol data inversion is $C = I$, $\mathbf{b} = \mathbf{0}$. Here the addition of a constraint corresponds to deleting the variable f_i and the i^{th} column of K_D from Eq. (3.2), and we can write $K_D = [K_a, K_0]$, where K_a corresponds to the constrained variables. Observe that both $\hat{R}_0(\lambda)$ and $\hat{R}(\lambda)$ tend to $\|\mathbf{y}^M\|/p$ as $\lambda \longrightarrow \infty$, and if there is a nonnegative solution to Eq. (3.2), $\boldsymbol{\varepsilon}^t = 0$, then $\hat{R}_0(\lambda)$ and $\hat{R}(\lambda)$ both tend to 1 as λ tends to zero. Additionally, one can show that

$$\lambda_i \geq \lambda_{0;i}, \qquad i = 1, \ldots, p \tag{3.84}$$

where strict inequality holds if the rank of K_a equals p. Equation Eq. (3.84) implies that for small λ, we can expect $\hat{R}(\lambda) > \hat{R}_0(\lambda)$. The relationship between $\hat{R}_0(\lambda)$ and $\hat{R}(\lambda)$ is more clear if the left-hand singular eigenvectors of K_a and K_D are approximately equal; in this case one can show that

$$\hat{R}(\lambda) - \hat{R}_0(\lambda) = \sum_{i=1}^{p} 2p\lambda(1 - \alpha_i) - (p\lambda \mathbf{y}_U^M)^2(1 - \alpha_i^2) \tag{3.85}$$

where

$$\alpha_i = \frac{\lambda_{0;i} + p\lambda}{\lambda_{0;i} + \hat{\lambda}_i + p\lambda} \tag{3.86}$$

and $\hat{\lambda}_i$ is an eigenvalue of $K_a{K_a}^T$. The difference defined by Eq. (3.85) is positive for small λ and negative for large λ, and because of this one might expect the minimizer of $\hat{R}_0(\lambda)$ to be less than the minimizer of $\hat{R}(\lambda)$.

In Figure 1 we compare $\hat{R}_0(\lambda)$ and $\hat{R}(\lambda)$ as functions of λ to $R(\lambda)$, $\|f_\lambda(x) - f^t(x)\|_1$, and $\|f^t(x) - f_\lambda(x)\|_{TV}$, where

$$\|f(x)\|_1 = \int_0^1 |f(x)|\, dx \tag{3.87}$$

$$\|f(x)\|_{TV} = \min \ \{f(x)\} + \int_0^1 |f'(x)|\, dx \tag{3.88}$$

Data with a signal-to-noise ratio of 20 (S/N = 20) were artificially generated for the differential mobility analyzer (DMA) [4] at 54 different inner rod voltages varying between 45 and 4500 volts for the distribution shown in Figure 1a. Since the test distribution is smooth, the inequality constraints add little information to the problem; thus, each $f_\lambda(x)$ has only a few active constraints, and the difference between $\hat{R}(\lambda)$ and $\hat{R}_0(\lambda)$ should be negligible as shown in Figure 1b. The difference between $R(\lambda)$ and the estimator $\hat{R}_0(\lambda)$ is noticeable, but the minimizers are nearly equal. Also one can note that the minimizer of $R(\lambda)$ is approximately the same as the minimizer of $\|f_\lambda(x) - f^t(x)\|_1$ and $\|f^t(x) - f_\lambda(x)\|_{TV}$. Some meaning to these norms is provided by Figure 1a, which shows $f_\lambda(x)$ for $\lambda = 1\times 10^{-5}$ (oversmoothed) and for $\lambda = 1\times 10^{-10}$ (undersmoothed).

Similar plots are shown in Figure 1d where we attempt to reconstruct the trimodal distribution shown in Figure 1c. Data were generated as in the previous example with S/N = 20. In this case, the inequality constraints are an important part of the solution; at the optimal λ we found that 50 percent of the constraints were active. However, even here we find only a slight difference between $\hat{R}(\lambda)$ and $\hat{R}_0(\lambda)$. Although $\hat{R}_0(\lambda)$ is a better estimator of $R(\lambda)$ in this case for most λ, it does not provide a better estimate of the minimizer of $R(\lambda)$. We do see the expected $\hat{R}(\lambda) < \hat{R}_0(\lambda)$ at large λ and $\hat{R}(\lambda) > \hat{R}_0(\lambda)$ at small λ. Here, $\lambda_D = 2 \times 10^{-10}$, and the corresponding solution is shown along with the solution corresponding to the minimizer of $\hat{R}_0(\lambda)$. Here we see that the method of discrepancy causes a significant amount of oversmoothing.

In a large number of realistic test aerosol inversions, we find only a small difference in the minimizers of $\hat{R}(\lambda)$ and $\hat{R}_0(\lambda)$, and often the minimizer of $\hat{R}_0(\lambda)$ can be found with only a few iterations of a successive substitution scheme. Here given λ^i, an estimate of the minimizer of $\hat{R}_0(\lambda)$, we

1. identify the set of active constraints $C_a(\lambda^i)$

2. find λ^{i+1} by minimizing $\tilde{R}$, where $\tilde{R}$ approximates $\hat{R}_0(\lambda)$ by assuming the set of active constraints, $C_a(\lambda^i)$, is fixed.

This procedure will not always converge, and in some cases we are forced to use a more robust minimization algorithm to choose successive λs.

3.4 Generalized cross validation

In the case of aerosol measurements, we often do not have reliable estimates of the magnitude of the errors available. Here, the analysis in the previous section leads to trouble because λ depends on the unknown or poorly known σ.

If $C = 0$, then one can predict σ and choose λ from the data by minimizing $V(\lambda)$ [11],

$$V(\lambda) = \sum_{i=1}^{p} (y_{[i]}^{R}(\lambda) - y_i^M)^2 w_i(\lambda) \tag{3.89}$$

where $y_{[i]}^{R}(\lambda)$ is the i^{th} datum predicted by the regularized solution when the i^{th} measurement is omitted from Eq. (3.2). The weights, $w_i(\lambda)$, are needed to ensure that $V(\lambda)$ reflects the rotation invariant nature of Eq. (3.2). The idea behind choosing λ to minimize Eq. (3.89) is simple: a good choice of λ is the one that best enables the data to predict an omitted datum. As the number of data increases, this technique for choosing λ becomes optimal as described in [31,30].

One can show that [11]

$$y_{[i]}^{R}(\lambda) - y_i^M = (y_i^R(\lambda) - y_i^M)/(1 - a_{i,i}(\lambda)) \tag{3.90}$$

where $a_{i,i}(\lambda)$ is the i^{th} diagonal element of $A(\lambda)$. From this and the observation that $w_i(\lambda)$ should equal 1 if $A(\lambda)$ is circulant, one can show that an appropriate expression for $w_i(\lambda)$ is [20]

$$w_i(\lambda) = \frac{(1 - a_{i,i}(\lambda))^2}{(p - \mathbf{tr}[A(\lambda)])^2} \tag{3.91}$$

This leads to

$$V(\lambda) = \frac{\|(I - A(\lambda))\|^2}{(\mathbf{tr}[I - A(\lambda)])^2} \tag{3.92}$$

Eq. (3.92) can be combined with Eq. (3.74) to yield

$$V(\lambda) = \frac{\sum_{i=1}^{p} r_i(\lambda)^2 {y_{U;i}^{M}}^2}{(\sum_{i=1}^{p} r_i(\lambda))^2} \tag{3.93}$$

where $r_i(\lambda)$ and $y_{U;i}^M$ are defined in Eq. (3.76). Note that

$$\begin{array}{lcll} V(\lambda) & \longrightarrow & \|\mathbf{y}^M\|^2 & \text{as} \quad \lambda \longrightarrow \infty \\ V(\lambda) & \longrightarrow & \dfrac{\sum_{i=1}^{p} {y_{U;i}^{M}}^2/\lambda_i^2}{\sum_{i=1} p 1/\lambda_i^2} & \text{as} \quad \lambda \longrightarrow 0 \end{array} \tag{3.94}$$

It is not uncommon for $\lambda = 0$ to minimize $V(\lambda)$.

The analysis for the constrained case is similar if we assume the set of constraints that is active at $\mathbf{f}_\lambda$ is not affected by the removal of one of the data. Here we choose the λ that minimizes $V_0(\lambda)$,

$$V_0(\lambda) = \sum_{i=1} (y_{0[i]}^{R}(\lambda) - y_{0i}^{M})^2 w_{0i}(\lambda) \tag{3.95}$$

where $y^R_{0[i]}(\lambda)$ is the constrained analog of $y^R_{[i]}(\lambda)$, and y^M_{0i} is defined by Eq. (3.66). Since we assumed that the constraints are not affected by the removal of a single datum, we find

$$y^R_{0[i]}(\lambda) - y^M_{0i} = (y^R_{0i}(\lambda) - y^M_{0i})/(1 - a_{0i,i}(\lambda)) \tag{3.96}$$

Similarly, an appropriate expression for the weights is given by

$$w_{0i}(\lambda) = \frac{(1 - a_{0i,i}(\lambda))^2}{(\mathbf{tr}[I - A_0(\lambda)])^2} \tag{3.97}$$

and

$$V_0(\lambda) = \frac{\|(A_0(\lambda) - I)\mathbf{y}^M_0\|^2}{(\mathbf{tr}[I - A_0(\lambda)])^2} \tag{3.98}$$

In the constrained case, the calculation can become lengthy if the set of active constraints depends on λ.

Again we consider in more detail the case $C = I$, $\mathbf{b} = \mathbf{0}$. The difference between $V_0(\lambda)$ and $V(\lambda)$ is not as transparent as the difference between $R(\lambda)$ and $R_0(\lambda)$. We only note that for large λ we can expect that $f_\lambda(x)$ will be very smooth; thus, there will be few active constraints, if any, and here $V_0(\lambda) = V(\lambda)$.

We have examined $V_0(\lambda)$ and $V(\lambda)$ under the same conditions as those of the examples presented in the previous section. In Figure 2a we see $V_0(\lambda)$ and $V(\lambda)$ in the case of the smooth distribution where the inequality constraints are unimportant, and as expected little difference between the two is observed, as was the case in the previous section for $R(\lambda)$ and $R_0(\lambda)$. Also, we note that the minimizer of $V(\lambda)$ is a good estimate of the minimizer of $R(\lambda)$.

In Figure 2b we show the $V_0(\lambda)$ and $V(\lambda)$ that are obtained for the reconstruction of the trimodal distribution shown in Figure 1c. The numerical conditions are the same as those used to generate Figure 1c. Notice that the constraints provide critical information in choosing λ. Here, $\lambda = 0$ is the minimizer of $V(\lambda)$, and the plot of $\|\mathbf{y}^R_\lambda - \mathbf{y}^M\|^2$ and the solution difference norms show that this would result in an unacceptable solution. Similar results were obtained for S/N = 10 and S/N = 100.

It is clear from Figure 2 that the derivative of $V_0(\lambda)$ need not be continuous, and this reflects the discrete addition of information as constraints are added or deleted. For very small λ, the set of constraints can be sensitive to small changes in λ, and one may find a minimum of $V_0(\lambda)$ in this region of λ. This minimum is not desirable because of the instability of the minimum with respect to changes in the data and the constraints; additionally, in this region the assumption that the set of constraints does not change as a datum is deleted is most likely invalid.

If the minimizers of $V_0(\lambda)$ and $V(\lambda)$ are very different, then the technique used to minimize $R_0(\lambda)$ fails when trying to minimize $V_0(\lambda)$; here, minimizing $V_0(\lambda)$ is difficult, and we do not have a simple technique that applies to all cases.

3.5 Numerical comparisons of λ_D, λ_V, and λ_R

In this section we present two examples that compare the regularized solutions obtained with the smoothing parameters λ_D, λ_R, and λ_V. We will ignore the smoothing parameters obtained when the constraints are omitted; if generalized cross validation is being used then, as shown in the previous section, the results can be catastrophic, and if $\mathbf{E}R(\lambda)$ is being minimized then usually the results do not change significantly if the constraints are removed.

In the first example we compare the average difference between the regularized solution and the true solution over a range of signal-to-noise ratios for the two true distributions shown in Figures 3a and 3b. In all cases we assume data from the DMA described in the previous section were available, along with data from the diffusion battery described in [8]. We also assumed that the size distribution was to be zero outside the interval (0.005 μm , 0.5 μm) and that the solution can be represented by 200 linear splines. Many sets of noisy data were generated with the signal-to-noise ratio ranging from 4 and 100, and the regularized solutions were calculated for the smoothing parameters λ_R, λ_D, and λ_V. The average values of $\|f^t(x) - f_\lambda(x)\|_{TV}$ and $\|f_\lambda(x) - f^t(x)\|_1$ are shown in Figures 3c-3f. Also, in Figures 3a and 3b we present representative solutions for S/N = 20.

In this example, one can see a clear advantage in using λ_R or λ_V as opposed to λ_D for the range of conditions tested. As the signal-to-noise ratio increases, the accuracy of the inverted solution must decay on the average; however, this decay is accelerated when λ_D is used to define $\mathbf{f}_\lambda$. Also note the difference between the solutions obtained from λ_R and λ_V is negligible in most cases, and this lends more credence to the claim that generalized cross validation is able to provide a good choice of the smoothing parameter and predict the magnitude of the errors in the data *from the data alone.*

Generalized cross validation works well in this example because the amount of information provided by the data is sufficient to justify minimizing Eq. (3.89); generalized cross validation cannot be expected to perform well in cases where the information content of the data is small relative to the amount of structure in the solution. For example, in the case of only a few impactor data, it is not clear if the minimizer of Eq. (3.92) is desirable.

In the second example we investigate the effects of inaccurately estimating the signal-to-noise ratio. This is important because often the magnitude of the error in aerosol measurement data cannot be accurately estimated, and here we expect as the estimate becomes worse, GCV will become more attractive. In this example, we assumed data were available from the same sources that were used in the previous example. Many sets of erroneous data were generated for unimodal and trimodal analogs of the test distributions shown in Figure 3 while the signal-to-noise ratio varied between 4 and 100. Regularized solutions were calculated for λ_V, λ_R, and λ_D, while the estimated signal-to-noise ratio remained at 20. The difference between the regularized solution and the true solution is shown in Figure 4 over the range of true signal-to-noise ratios. These results clearly demonstrate the power of using λ_V in the regularized solution. The main point here is that generalized cross validation was able to consistently predict an acceptable amount of smoothing in the test cases without being affected by the poorly estimated signal-to-noise ratio; this inaccurate estimation of the true S/N, however, had a disastrous affect on the $\mathbf{f}_\lambda$ corresponding to λ_R and λ_D. One can also see that of the three methods for choosing the smoothing parameter, λ_D is most sensitive to overestimating the signal-to-noise ratio.

The advantages of using λ_V can be further highlighted by examining, for example, the average $\|f^t(x) - f_\lambda(x)\|_{TV}$ that one would obtain if the estimated signal-to-noise ratio was 20 while the logarithm of the true signal-to-noise ratio was a uniform random variable varying between $20c$ and $20/c$, with $1 < c \leq 5$. Thus we seek $\bar{N}(c)$ defined by

$$\bar{N}(c) = \frac{1}{2\log c}\int_{(20/c)}^{20c} N(x) d\log x \tag{3.99}$$

where N is the average value of the norm of interest evaluated at a signal-to-noise level of x given in Figure 4.

The results are shown in Figure 5 and emphasize the average cost of a poor estimate of the signal-to-noise ratio. We see in many cases one is better off using λ_V, even if the estimated signal-to-noise ratio is exact. For the two distributions having the most structure, there may be some advantage to using λ_R if one's estimate of the true signal-to-noise ratio is within a factor of 1.5; however, the difference is slight. For distributions with large amounts of structure relative to the amount of data, the idea behind λ_V is lacking, since it may not be possible for the set of data to predict a key omitted datum as suggested by Eq. (3.89).

The results in Figure 5 point to an advantage of this work over that presented in [47], where the analysis was based on using λ_D. In most realistic situations involving aerosol data, the signal-to-noise ratio cannot be estimated accurately enough to justify using the method of discrepancy.

3.6 Conclusions

Regularization parameters, λ_D, λ_R, and λ_V, based on the method of discrepancy, minimization of an unbiased estimate of the expected errors, and generalized cross validation, respectively, have been investigated for the regularized solution to the constrained linear inverse problem arising in aerosol data inversion. A computationally feasible method of finding λ_R and λ_V when constraints are important has been presented. An example was presented where consideration of the constraints is crucial in defining λ_V. The results of numerical experiments suggest that λ_D is too large and point out the importance of using generalized cross validation if the signal to noise ratio cannot be accurately estimated.

List of Figures

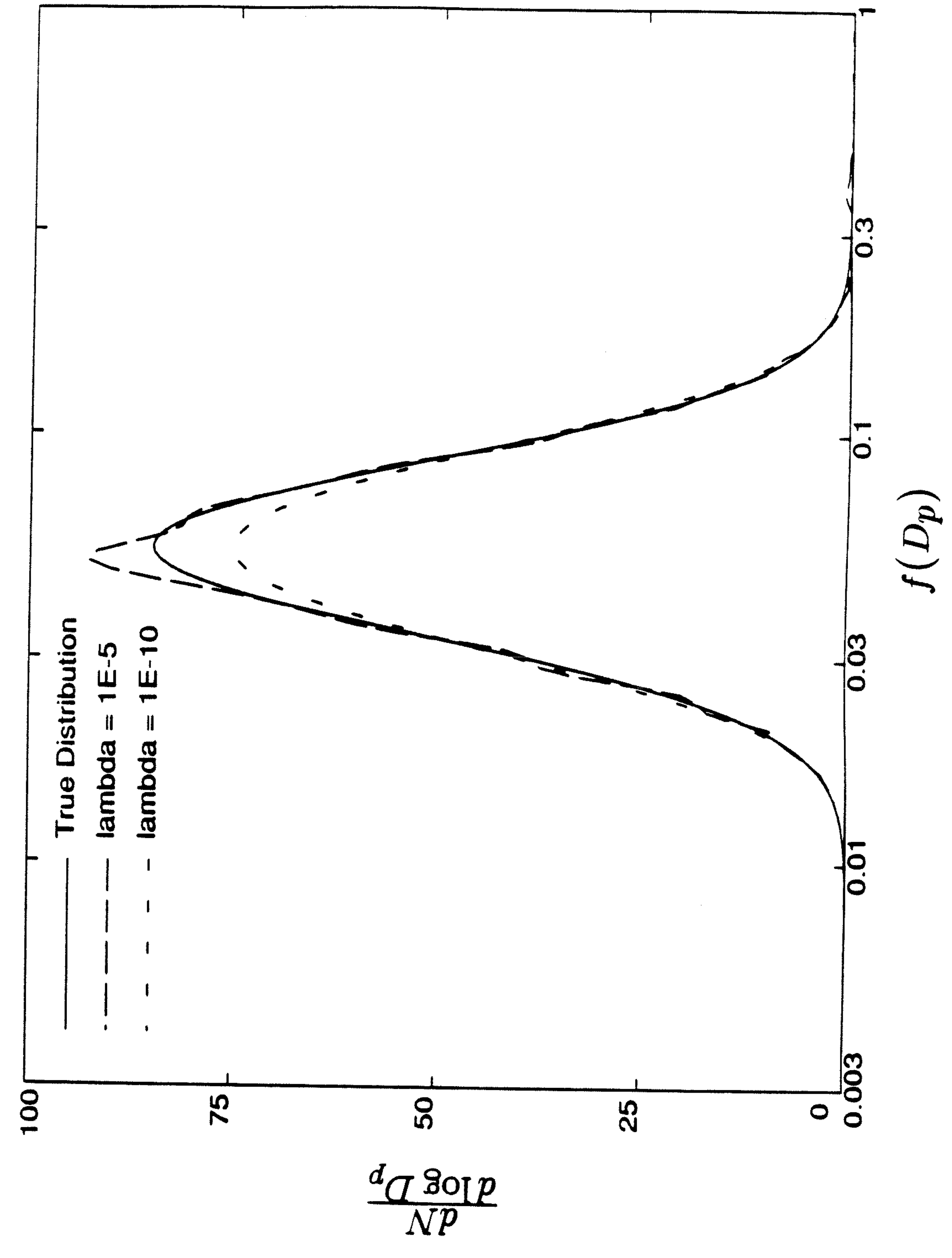

True Distribution
lambda = 1E-5
lambda = 1E-10
100
75
50
25
0
0.003
0.01
0.03
0.1
0.3
1
$\frac{dN}{d\log D_p}$
$f(D_p)$

Figure 1a

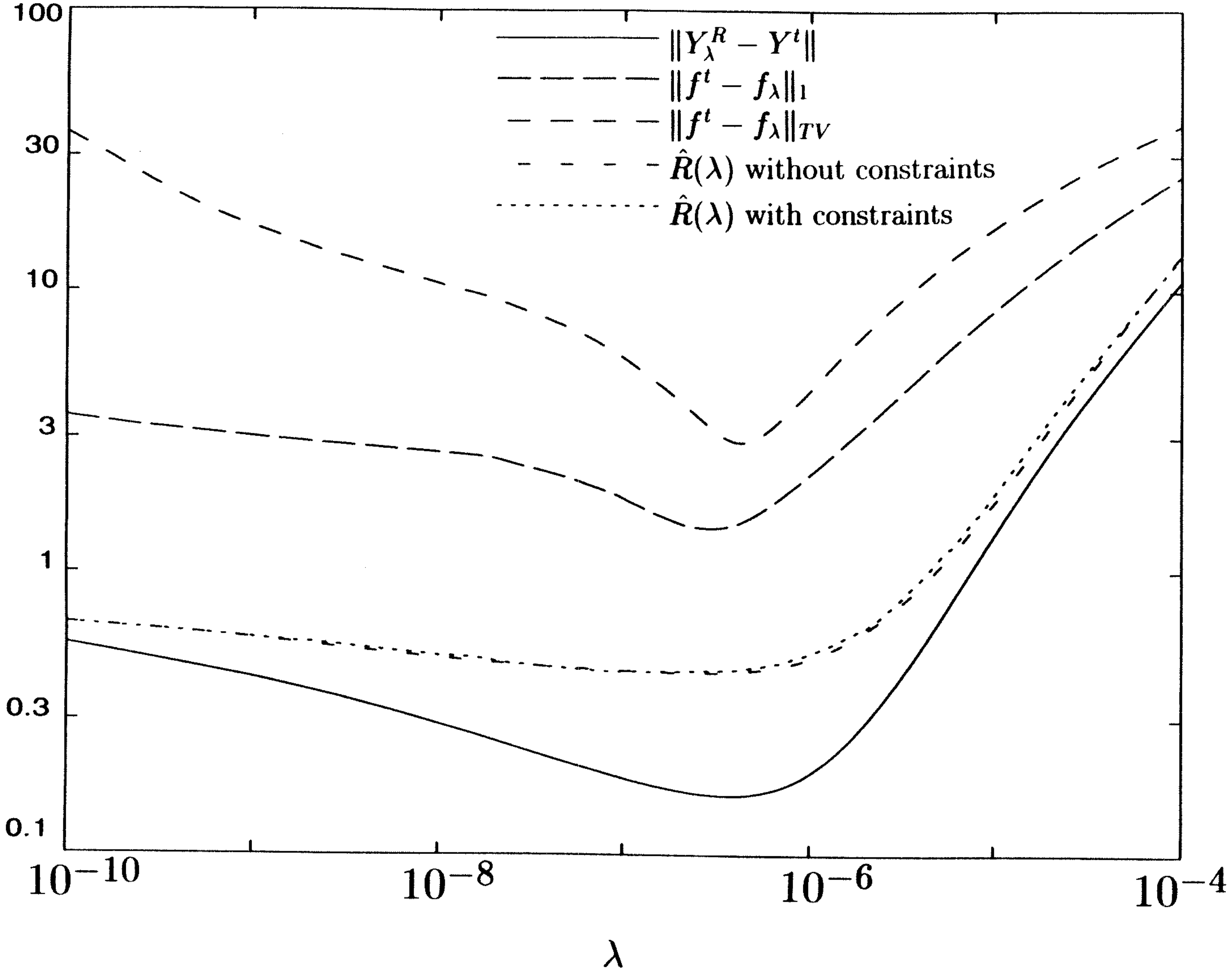

100
30
10
3
1
0.3
0.1
10^{-10}
10^{-8}
10^{-6}
10^{-4}
λ
$\|Y_\lambda^R - Y^t\|$
$\|f^t - f_\lambda\|_1$
$\|f^t - f_\lambda\|_{TV}$
$\hat{R}(\lambda)$ without constraints
$\hat{R}(\lambda)$ with constraints

Figure 1b

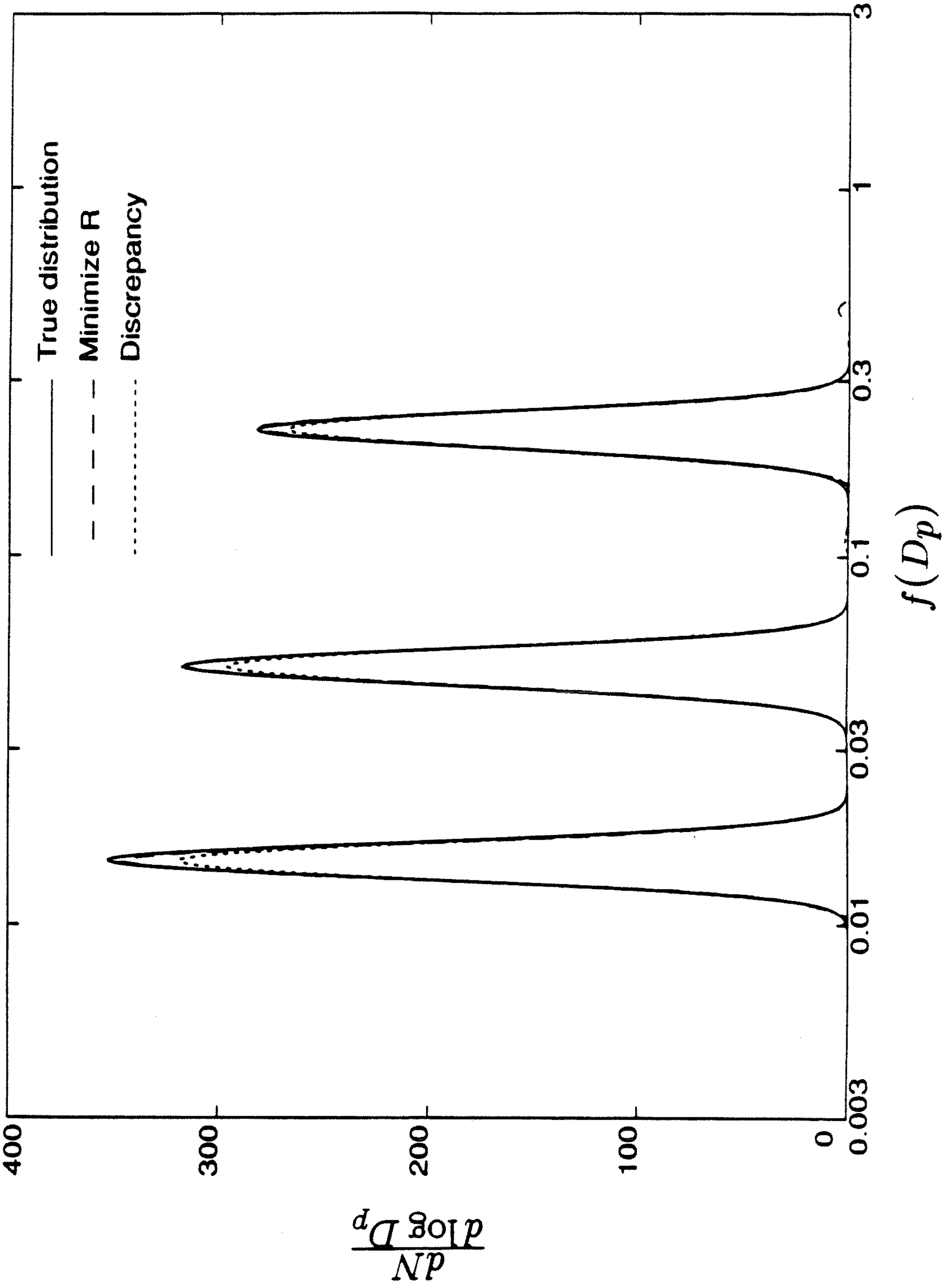
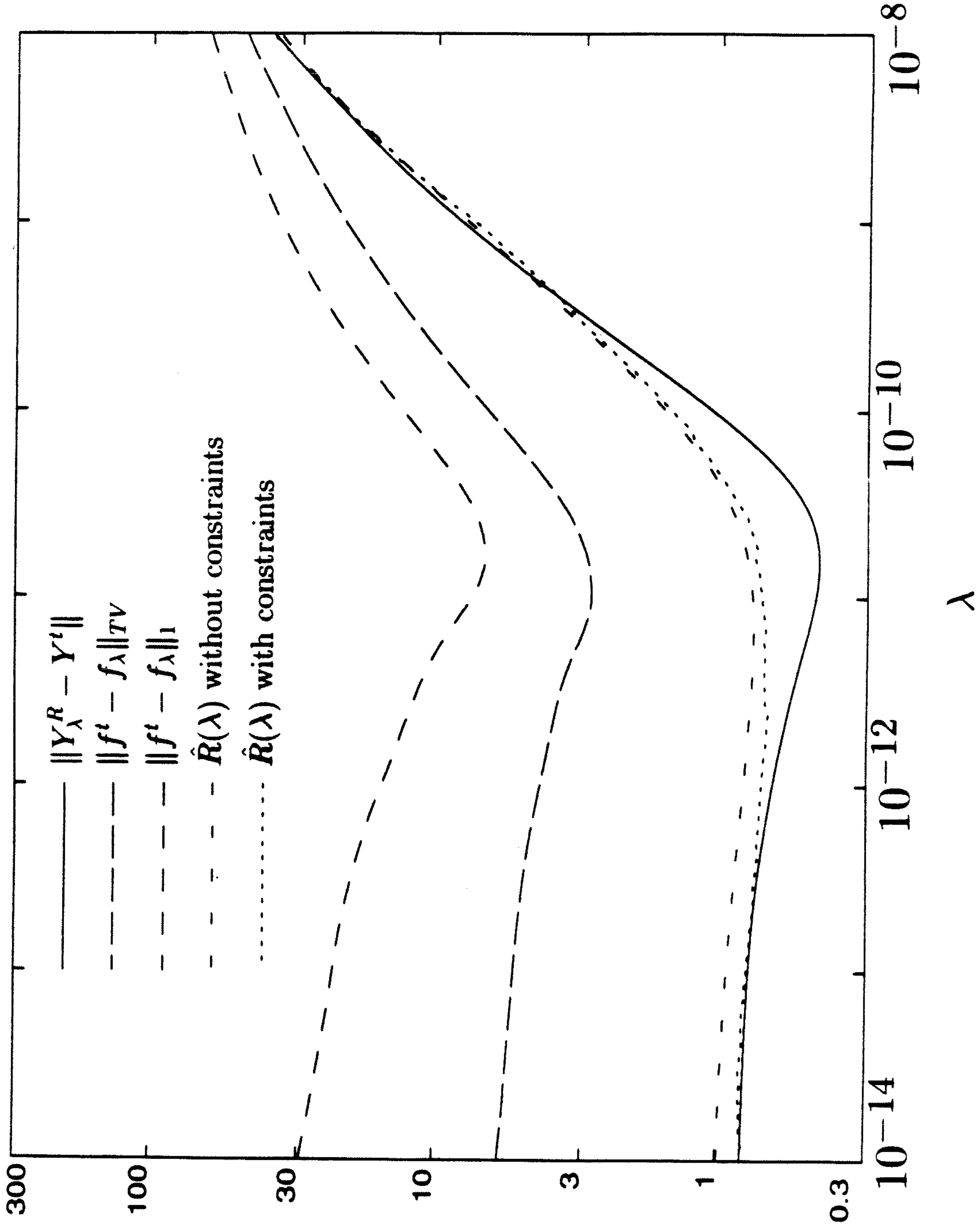

True distribution
Minimize R
Discrepancy
400
300
200
100
0
0.003
0.01
0.03
0.1
0.3
1
3
$\frac{dN}{d\log D_p}$
$f(D_p)$

Figure 1c



$\|Y_\lambda^R - Y^t\|$
$\|f^t - f_\lambda\|_{TV}$
$\|f^t - f_\lambda\|_1$
$\hat{R}(\lambda)$ without constraints
$\hat{R}(\lambda)$ with constraints
300
100
30
10
3
1
0.3
10−14
10−12
10−10
10−8
λ

Figure 1d

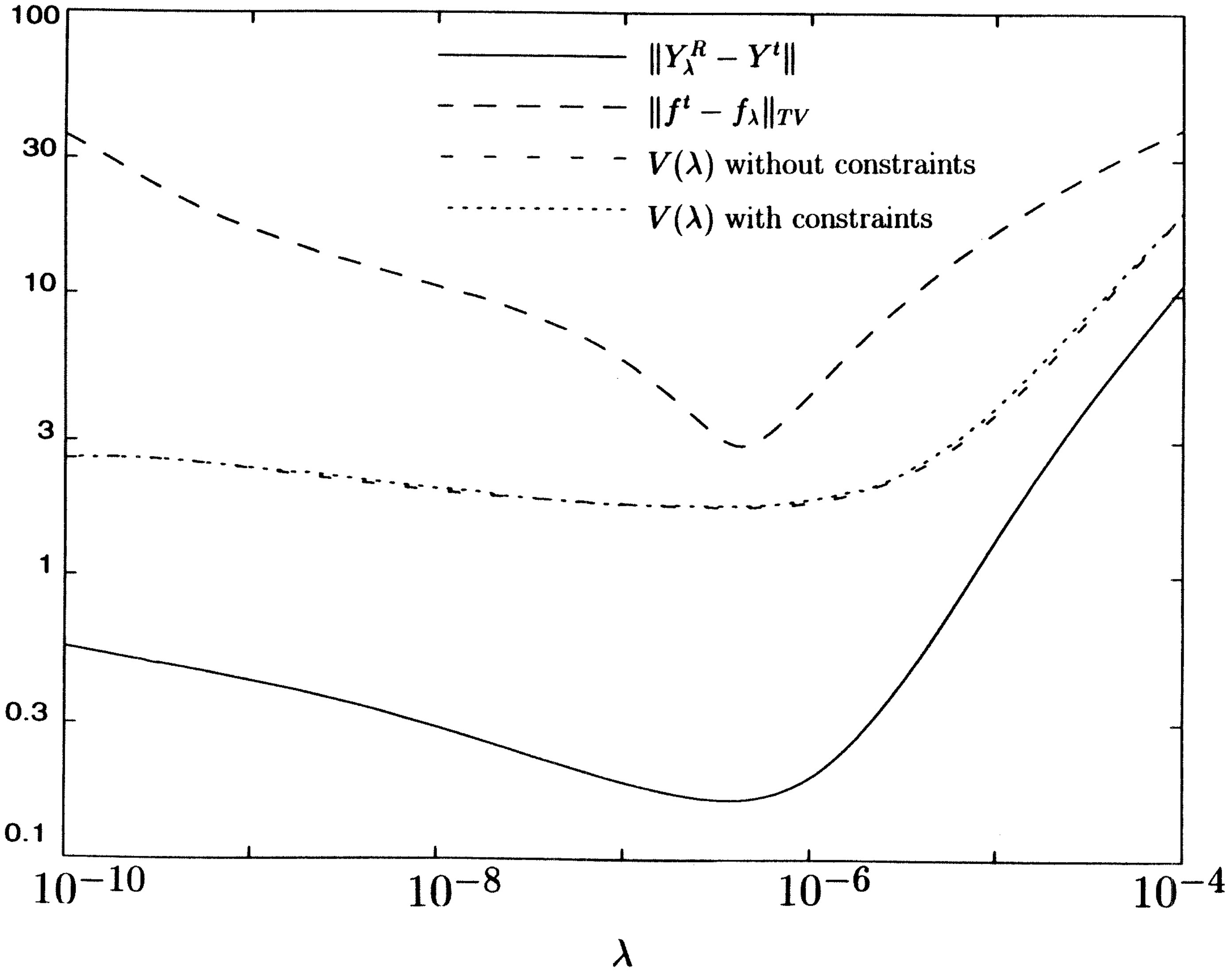

100
30
10
3
1
0.3
0.1
10^{-10}
10^{-8}
10^{-6}
10^{-4}
λ
$\|Y_\lambda^R - Y^t\|$
$\|f^t - f_\lambda\|_{TV}$
$V(\lambda)$ without constraints
$V(\lambda)$ with constraints

Figure 2a

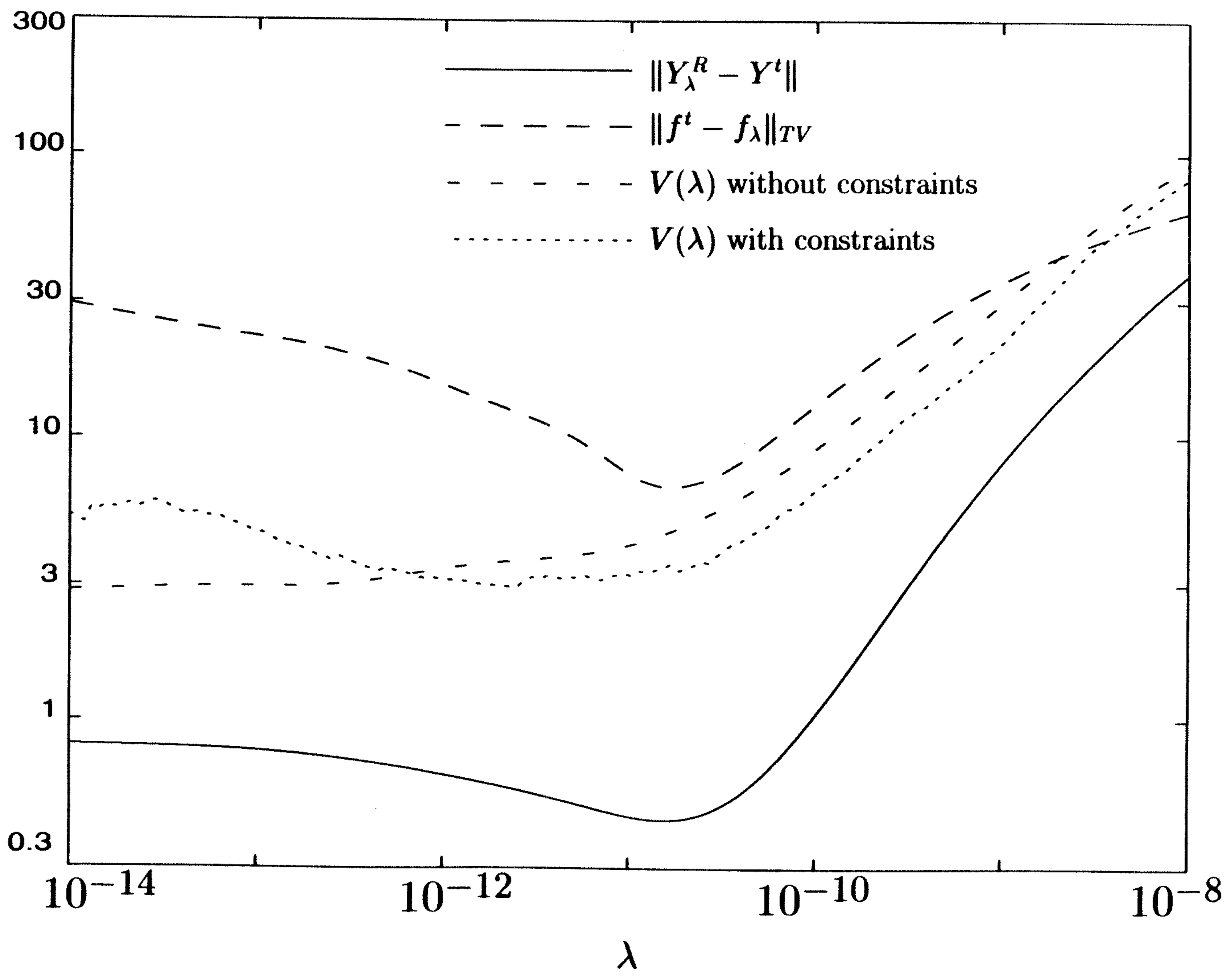

300
100
30
10
3
1
0.3
10^{-14}
10^{-12}
10^{-10}
10^{-8}
λ
$\|Y_\lambda^R - Y^t\|$
$\|f^t - f_\lambda\|_{TV}$
$V(\lambda)$ without constraints
$V(\lambda)$ with constraints

Figure 2b

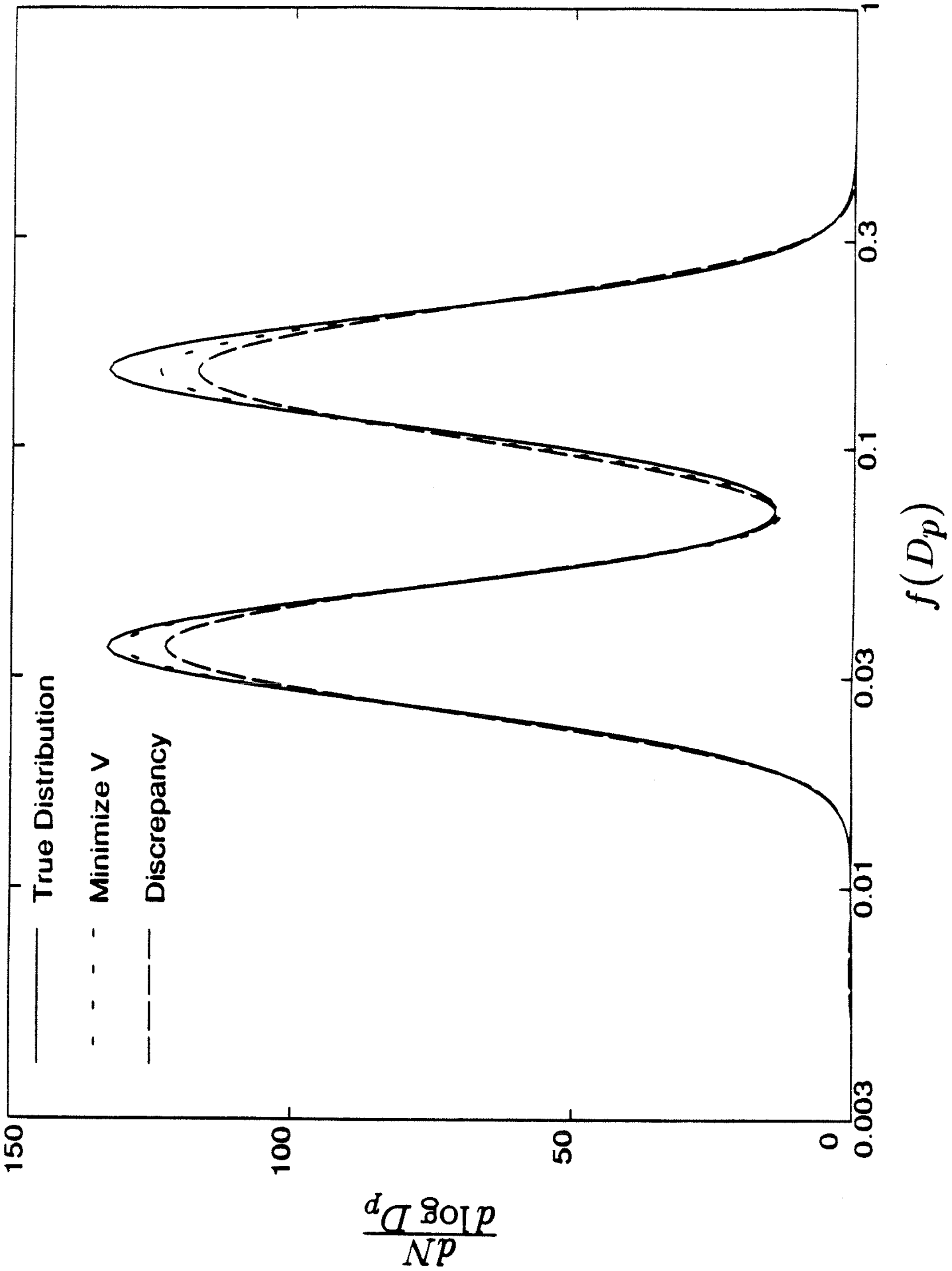

True Distribution
Minimize V
Discrepancy
150
100
50
0
0.003
0.01
0.03
0.1
0.3
1
$\frac{dN}{d\log D_p}$
$f(D_p)$

Figure 3a

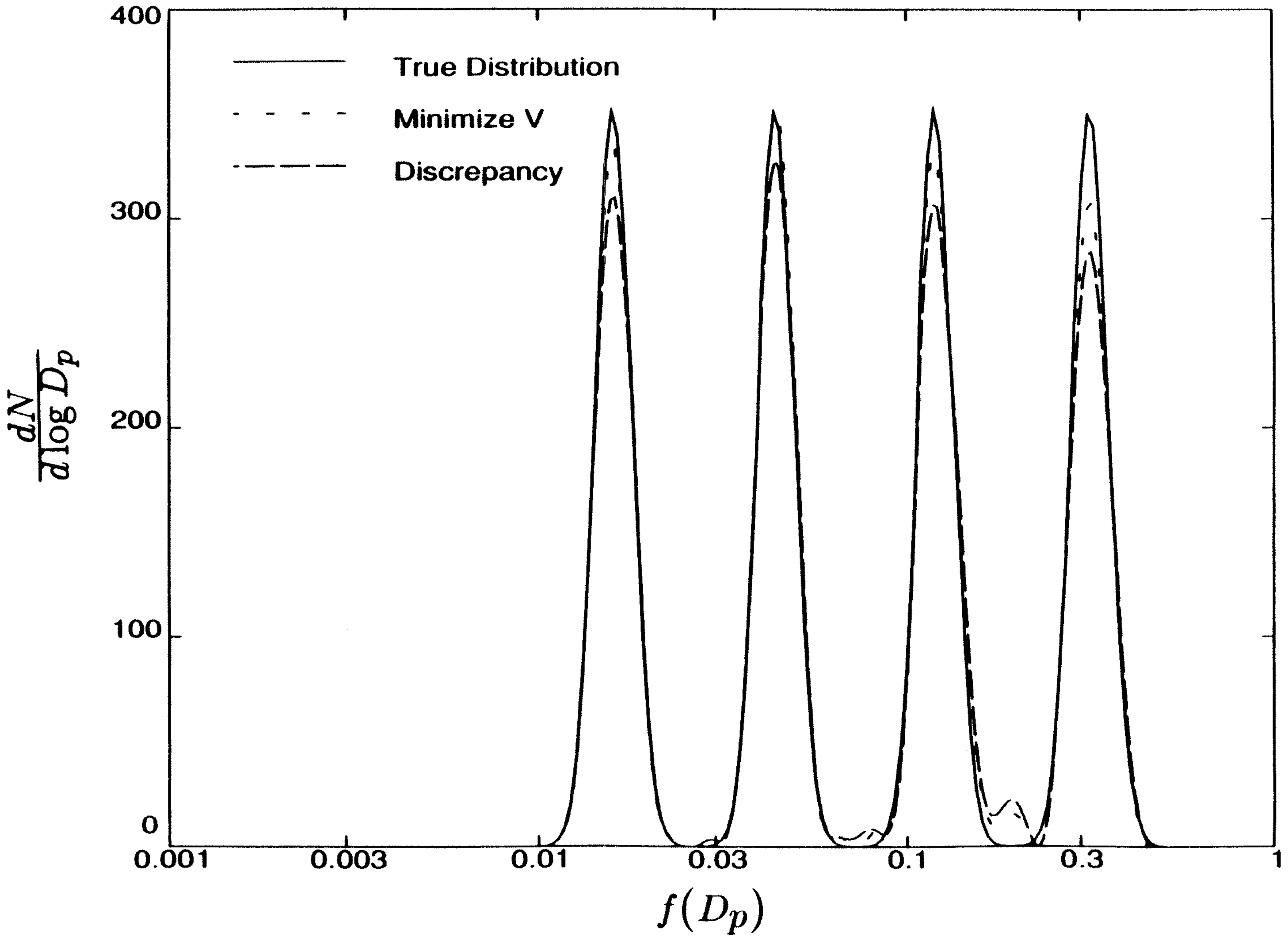

True Distribution
Minimize V
Discrepancy
400
300
200
100
0
0.001
0.003
0.01
0.03
0.1
0.3
1
$\frac{dN}{d\log D_p}$
$f(D_p)$

Figure 3b

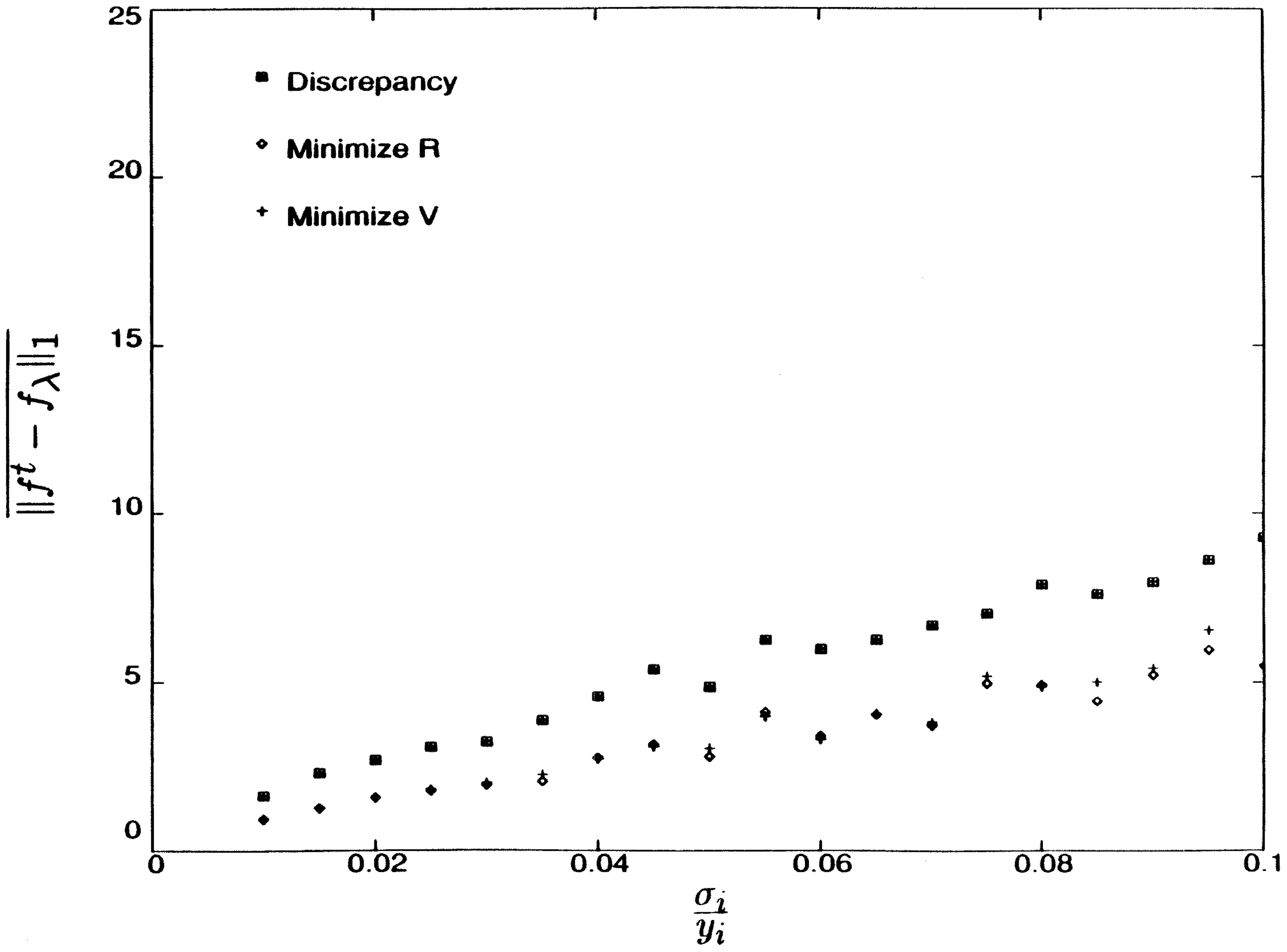

25
20
15
10
5
0
0
0.02
0.04
0.06
0.08
0.1
■ Discrepancy
◇ Minimize R
+ Minimize V
$\overline{\|f^t - f_\lambda\|_1}$
$\frac{\sigma_i}{y_i}$

Figure 3c

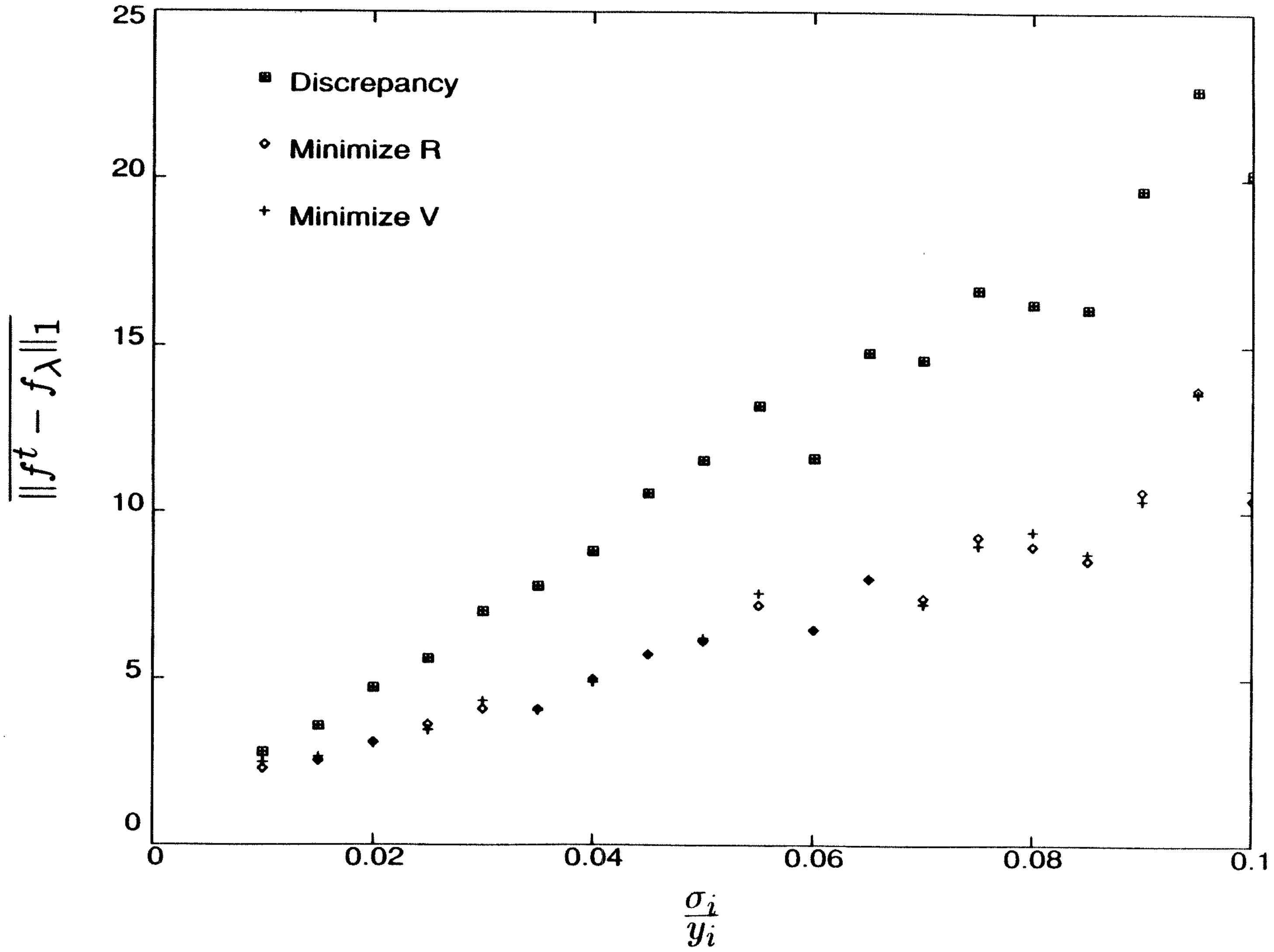

Discrepancy
Minimize R
Minimize V
25
20
15
10
5
0
0
0.02
0.04
0.06
0.08
0.1
$\|f^t - f_\lambda\|_1$
$\frac{\sigma_i}{y_i}$

Figure 3d

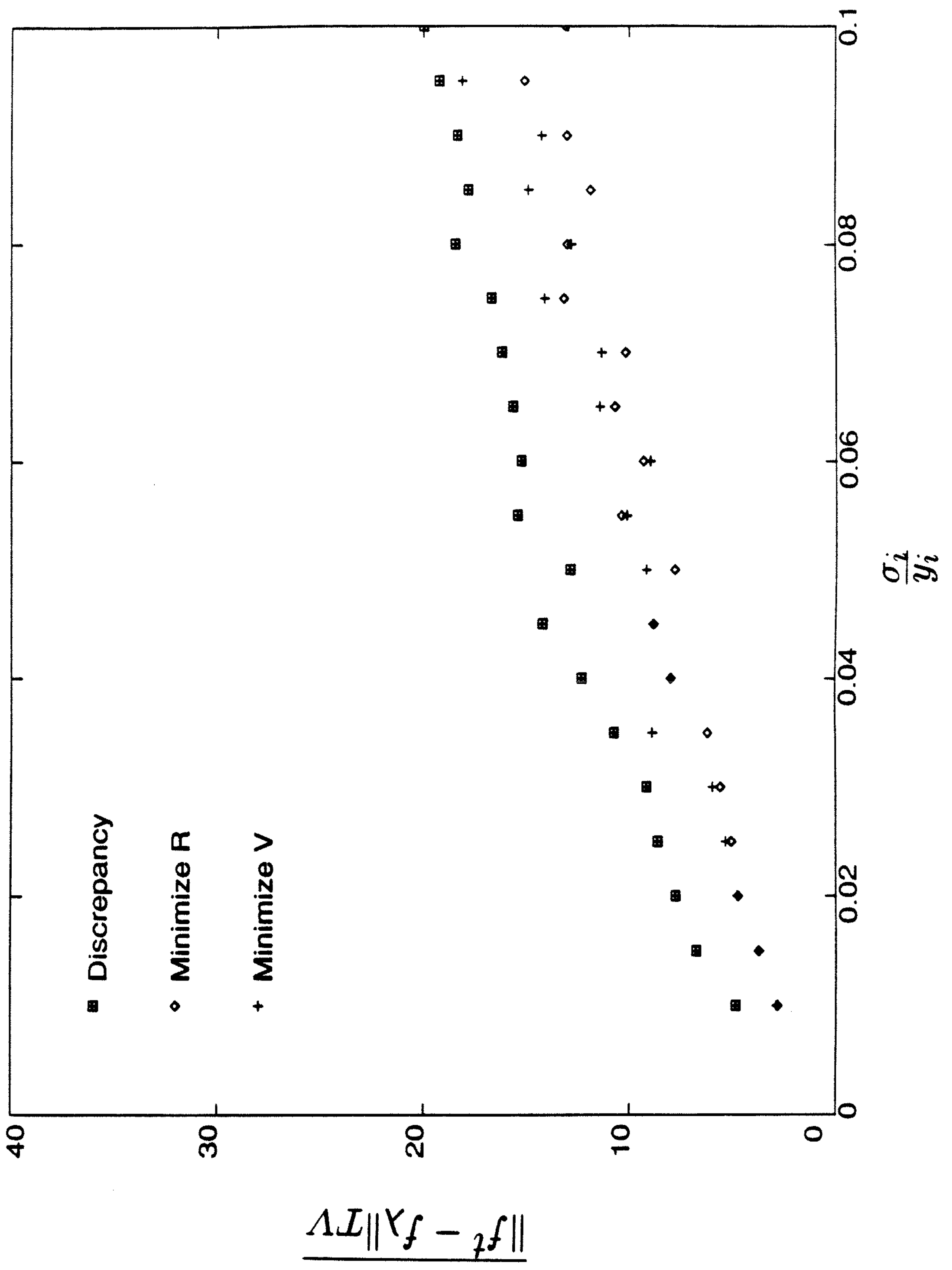

■ Discrepancy
◇ Minimize R
+ Minimize V
0
0.02
0.04
0.06
0.08
0.1
0
10
20
30
40
$\frac{\sigma_i}{y_i}$
$\|f^t - f_\lambda\|_{TV}$

Figure 3e

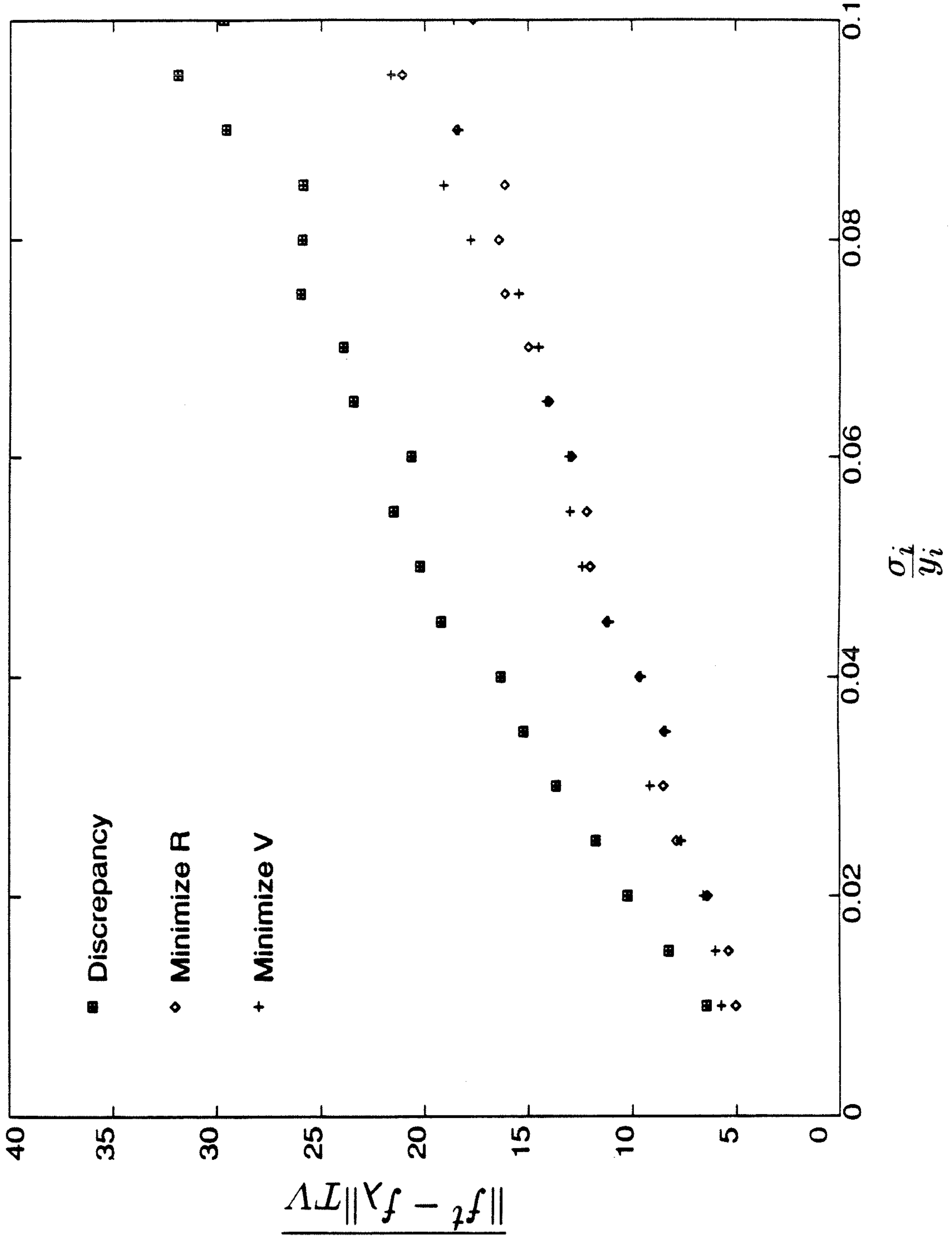
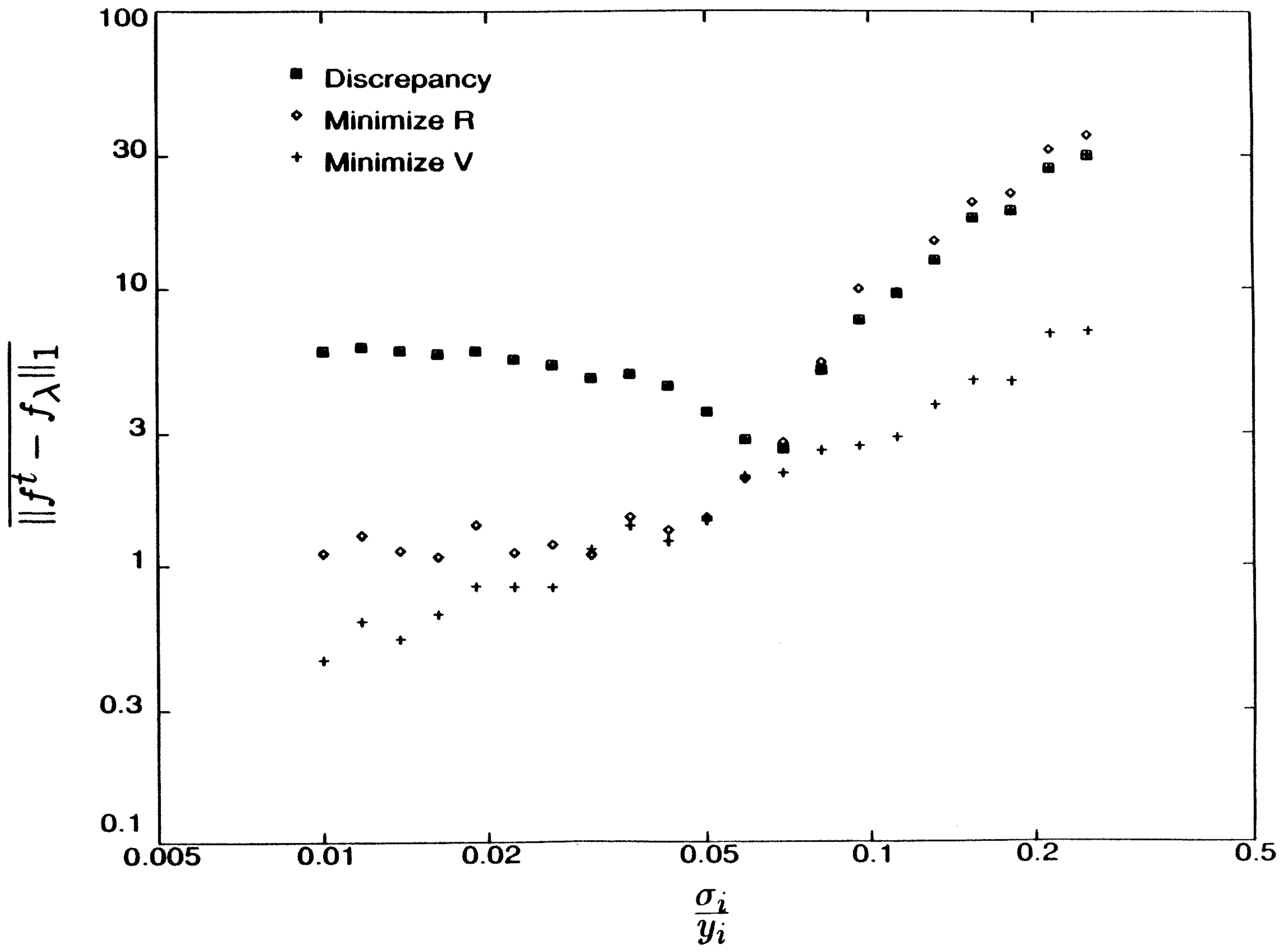

Discrepancy
Minimize R
Minimize V
0
5
10
15
20
25
30
35
40
$\frac{\|f^t - f_\lambda\|_{TV}}{}$
0
0.02
0.04
0.06
0.08
0.1
$\frac{\sigma_i}{y_i}$

Figure 3f



Discrepancy
Minimize R
Minimize V
100
30
10
3
1
0.3
0.1
0.005
0.01
0.02
0.05
0.1
0.2
0.5
$\overline{\|f^t - f_\lambda\|_1}$
$\frac{\sigma_i}{y_i}$

Figure 4a

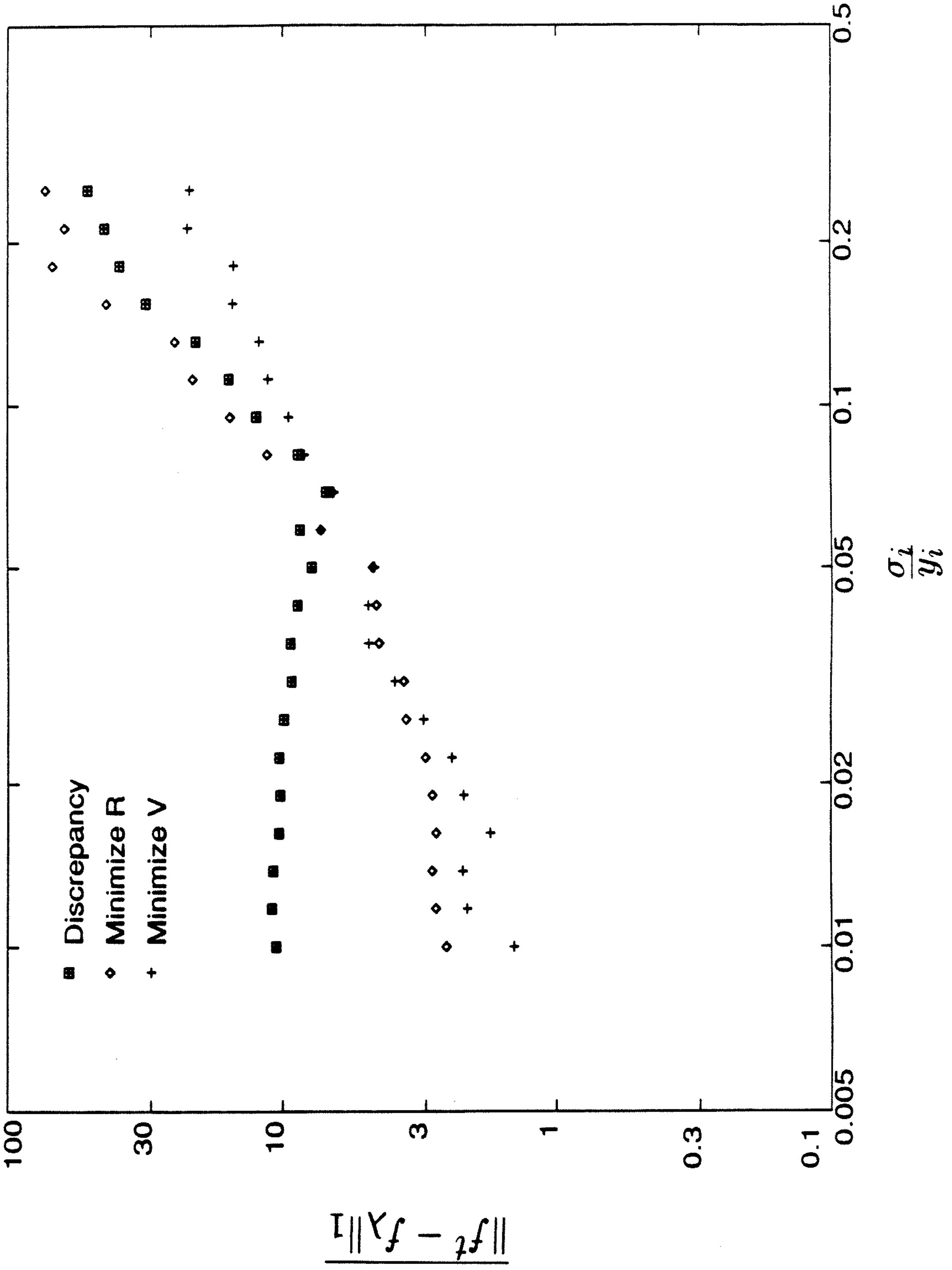

100
30
10
3
1
0.3
0.1
0.005
0.01
0.02
0.05
0.1
0.2
0.5
Discrepancy
Minimize R
Minimize V
$\frac{\sigma_i}{y_i}$
$\|f^t - f_\lambda\|_1$

Figure 4b

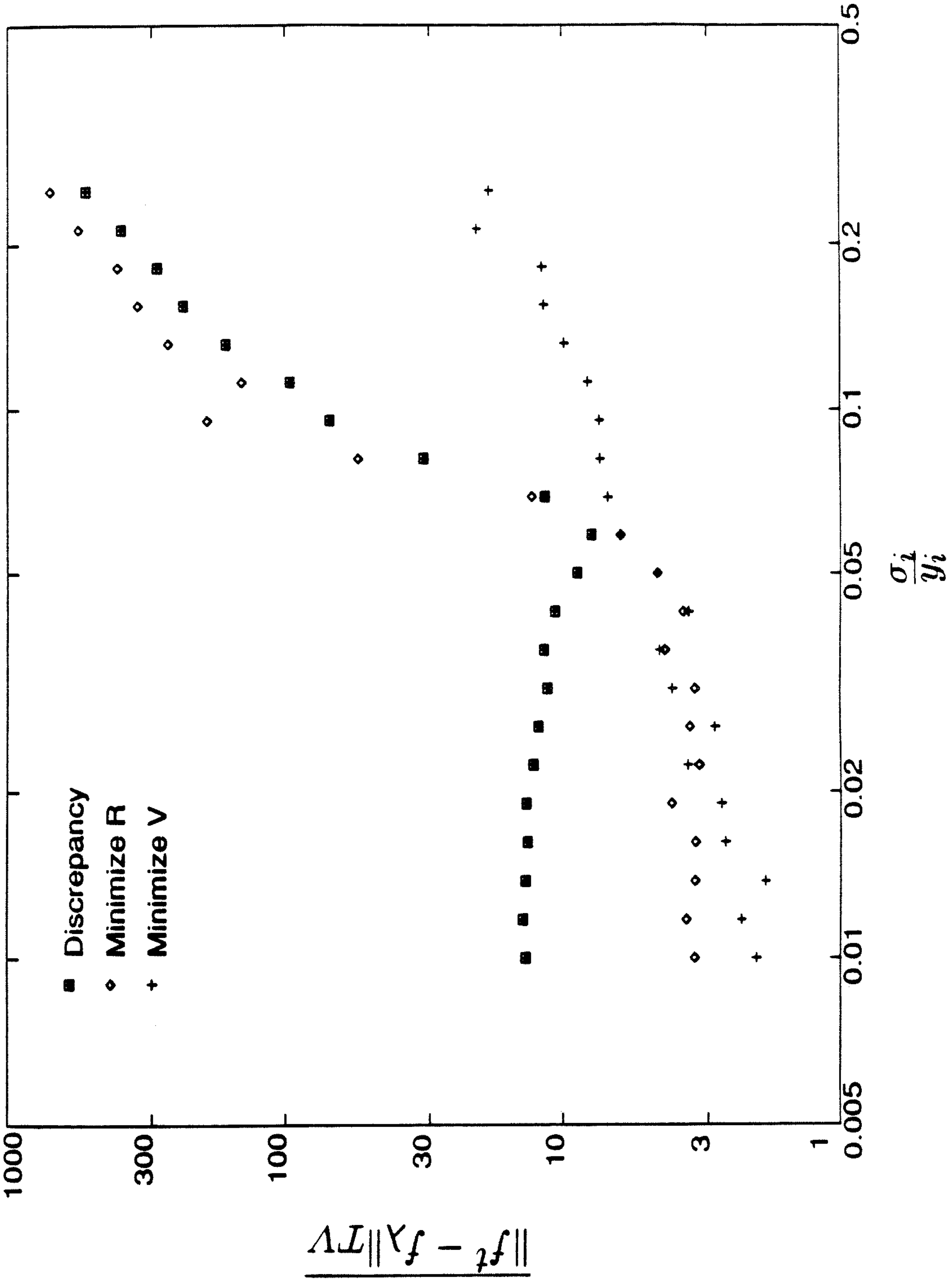

■ Discrepancy
◇ Minimize R
+ Minimize V
1000
300
100
30
10
3
1
0.005
0.01
0.02
0.05
0.1
0.2
0.5
$\|f^t - f_\lambda\|_{TV}$
$\frac{\sigma_i}{y_i}$

Figure 4c

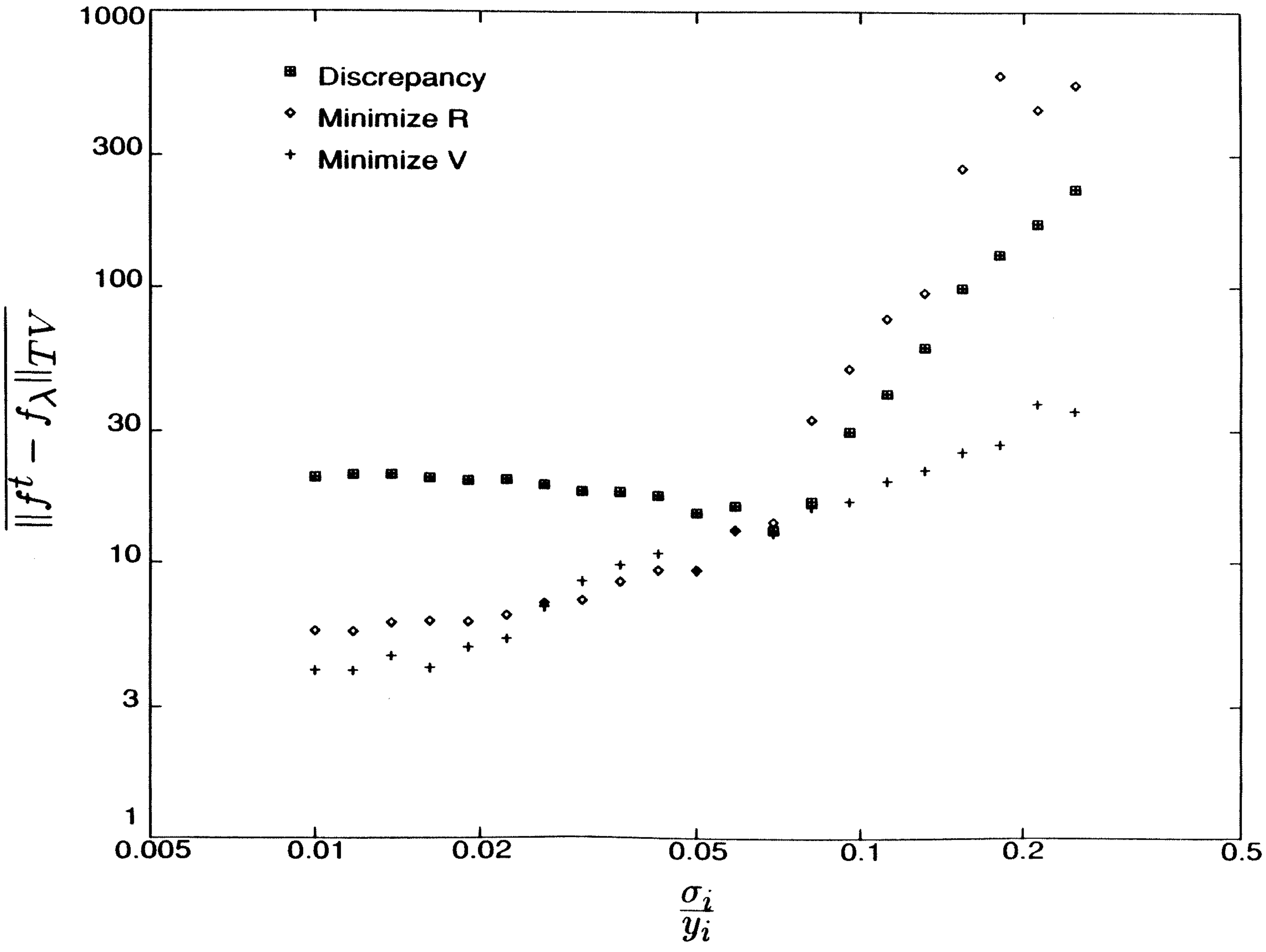

Discrepancy
Minimize R
Minimize V
1000
300
100
30
10
3
1
0.005
0.01
0.02
0.05
0.1
0.2
0.5
$\|f^t - f_\lambda\|_{TV}$
$\frac{\sigma_i}{y_i}$

Figure 4d

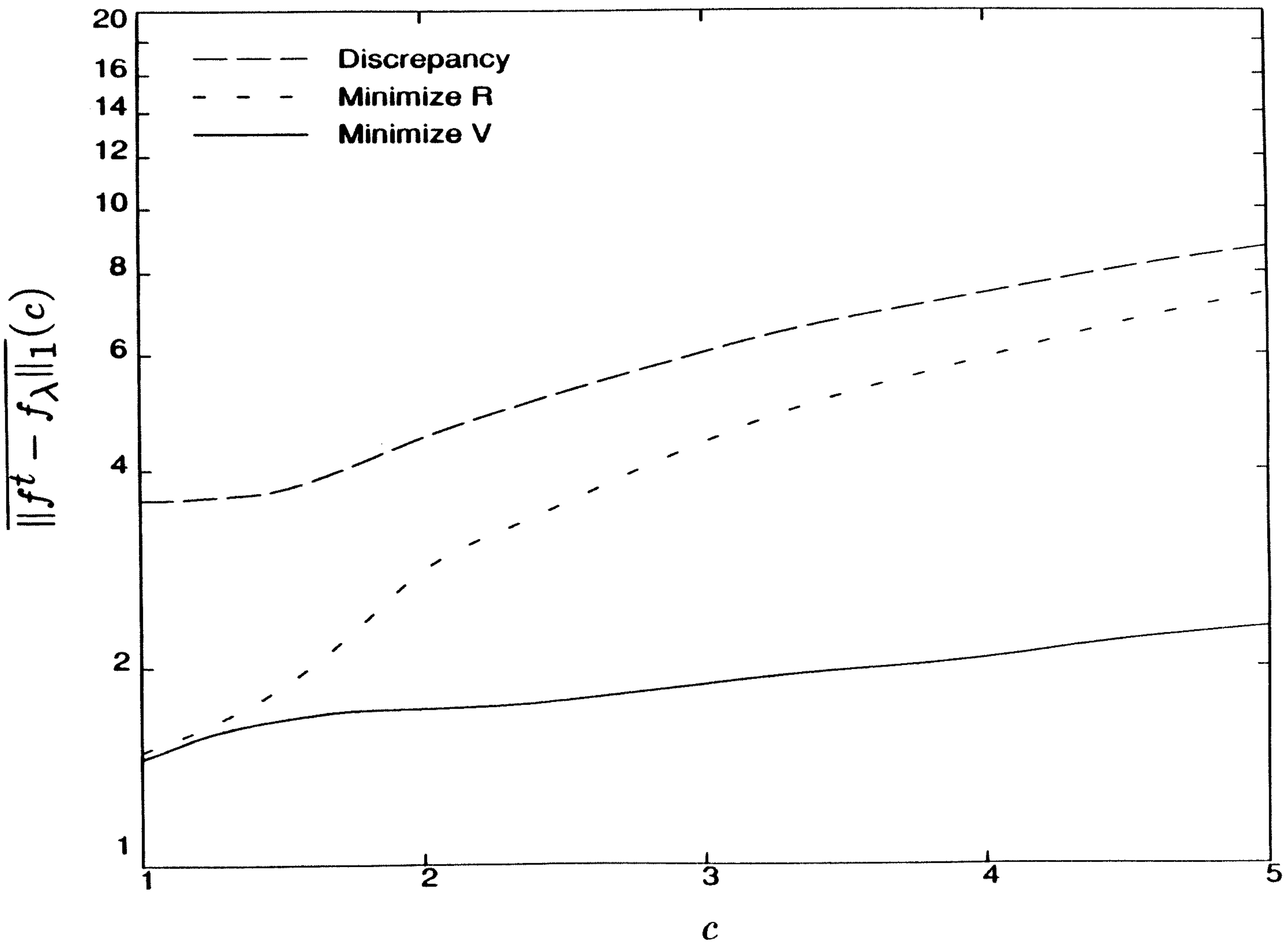

Discrepancy
Minimize R
Minimize V
$\overline{\|f^t - f_\lambda\|_1}(c)$
20
16
14
12
10
8
6
4
2
1
1
2
3
4
5
c

Figure 5a

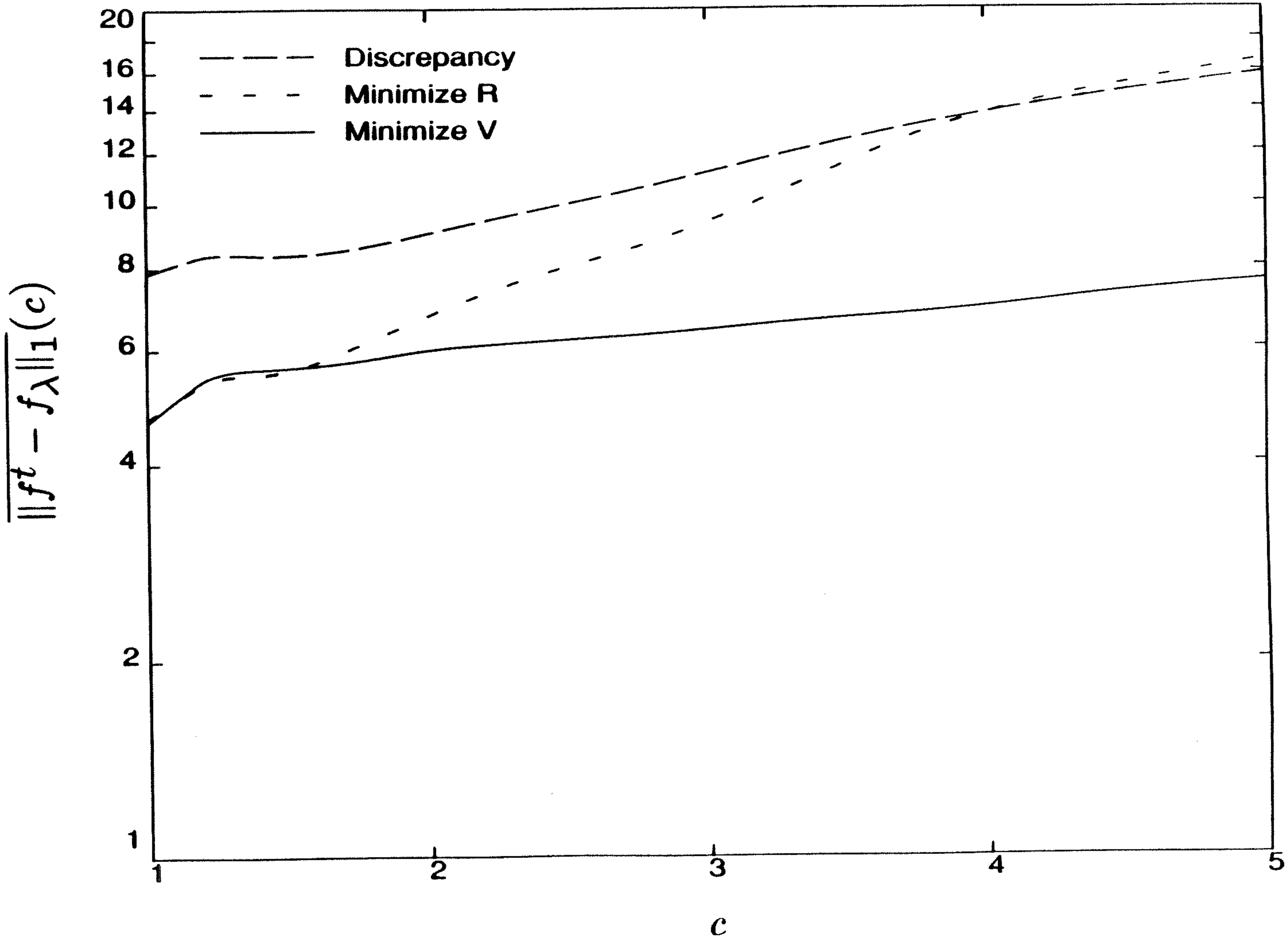

20
16
14
12
10
8
6
4
2
1
1
2
3
4
5
c
$\overline{\|f^t - f_\lambda\|_1}(c)$
Discrepancy
Minimize R
Minimize V

Figure 5b

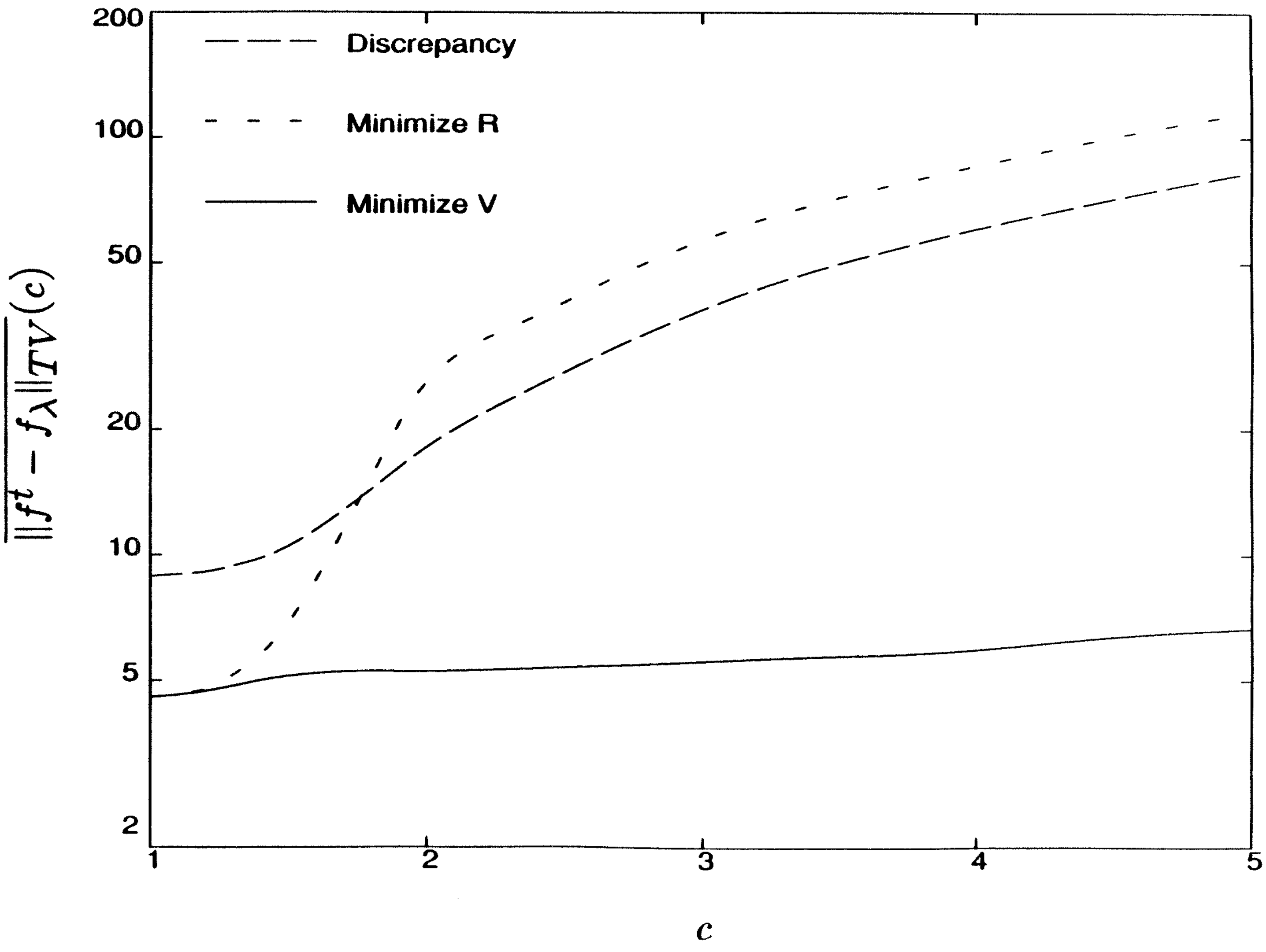

Discrepancy
Minimize R
Minimize V
200
100
50
20
10
5
2
$\overline{\|f^t - f_\lambda\|_{TV}}(c)$
1
2
3
4
5
c

Figure 5c

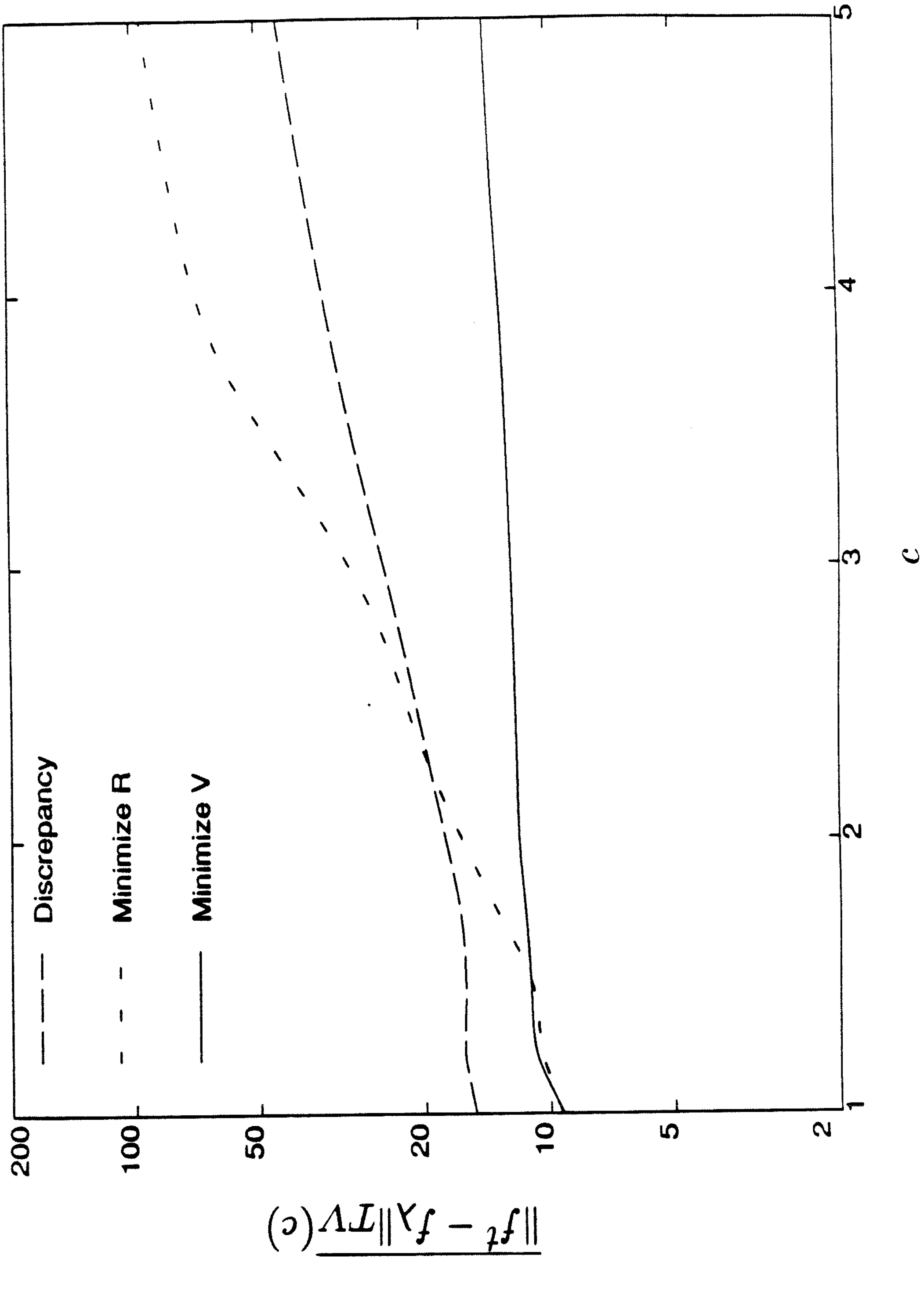

Discrepancy
Minimize R
Minimize V
200
100
50
20
10
5
2
1
2
3
4
5
c
$\underline{\|f^t - f_\lambda\|_{TV}}(c)$

Figure 5d

Chapter 4

Estimating the Variance in Solutions to the Aerosol Data Inversion Problem

Abstract

Regularization has been successfully used for solving a wide variety of ill-posed problems such as the inversion of aerosol size distribution data. The solutions are well characterized and converge nicely to the true distribution as the number of data increases.

If there are few data, then there can be many reasonable distributions that are consistent with the measurements. Here, in addition to knowing an optimal solution, one should also have an estimate of the variance of the solution or a characterization of the size of the solution set.

We set up the necessary machinery to allow one to estimate the variance of linear functionals of the size distribution, e.g., the concentration of particles in a given size interval. This estimate depends on the form of the weighted average, the variance in one's *a priori* estimate of the size distribution, the instrument's response, and the uncertainty in the data.

There are many applications. We demonstrate, for example, how to determine which of two instruments will better allow one to estimate the concentration of particles in a given size interval. Also, we determine the number of measurements necessary to ensure the variance of the estimated concentration is less than a specified value.

4.1 Introduction

The inversion of aerosol size distribution data amounts to solving the integral equation: find $f(x)$ given a set of measurements $\mathbf{y}$ and the relation

$$\int f(x)k_i(x)\,dx = y_i + \varepsilon_i \quad i = 1, \ldots, p \tag{4.1}$$

subject to inequality constraints, where $k_i(x)$ is the known instrument response, and ε is the measurement error.

This inversion problem is ill-posed [40]:

- solutions will become unstable when the number of data are large
- solutions are not unique
- exact solutions often do not exist

Regularization has been successfully used to find realistic solutions to this inverse problem [47,48]. Here, an optimal element in the feasible set is identified based on the fidelity of the solution to the data and the stability of the solution. Regularization is a powerful technique because often the regularized solution will remain stable and converge quickly to the true solution as the number of data increases [21] .

In many aerosol data inversion problems, however, there are limited data; for example, a diffusion battery may provide only 12 data. In such a case, many realistic solutions may agree with the data exactly, and the regularized solution is only one of these. To illustrate this, we inverted two log-normal test distributions from simulated data for the diffusion battery [7,8], with the diffusion battery response functions shown in Figure 1. If the test distribution has a log-mean diameter of 0.020 μm, then the regularized solution is reasonably accurate as shown in Figure 1. On the other hand, there is a large disparity between the true and inverted solution when the test distribution has log-mean diameter of 0.008 μm.

The difficultly behind inverting the distribution with the 0.008 μm peak is clarified by examining the diffusion battery kernel functions in Figure 2a; the diffusion battery does not response adequately to the smaller particles. No inversion algorithm can expect to overcome this inherent deficiency of the instrument and select the true solution when the true size distribution falls significantly in this size range. As a result, one will always have more uncertainty in the inverted solution for the distribution with the 0.008 μm peak in comparison to that with the 0.020 μm peak.

In addition to determining a single optimal distribution, it would also be valuable, therefore, to characterize the uncertainty in the distribution. For example, instead of stating that there are 100 particles/cm^3 in the size interval 0.010–0.020 μm and 100 particles/cm^3 in the size interval 0.006–0.012 μm, one would like to include error estimates and state that there are 100±10 particles/cm^3 in the size interval 0.010–0.020 μm and 100±80 particles/cm^3 in the size interval 0.006–0.012 μm. Thus, one goal of this paper is to determine the variance of properties of the size distribution, such as the concentration, or the integral over a certain interval. Along with this, we can benefit by determining which linear functionals of the size distribution have the most or least variance. This information can help characterize an instrument's performance as well as provide bounds on a linear functional's variance.

Additionally, it will be valuable to quantify the concept of the amount of information provided by an instrument, and add rigor to the statement "the diffusion battery provides insufficient information about the number of particles in the size interval 0.006–0.012 μm". With a clear definition of information we can easily determine which combination of measurements is best to employ in a given experiment, and we can even examine questions of instrument design. For example, we can investigate the modifications that one can make to an instrument's response function to increase the amount of information that the instrument provides.

In Section 2 we describe the main concepts that are necessary to calculate the variance of linear functionals of the size distribution. In Section 3 we quantify the information provided by a set of measurements and show how the information is affected by the errors in the data and the choice of autocorrelation functions. In Section 4 we point out some of the advantages of this approach over a recently proposed alternative and also discuss shortcomings in our variance analysis. In Section 5 we examine the information content of several instruments. We show, for example, how the performance of the diffusion battery can be improved. Finally we demonstrate that dependent errors in an instrument can place an upper bound on the amount of information that is provided by the instrument, and that dependent errors point to an advantage in using more than one instrument to determine a size distribution.

4.2 Variance of the size distribution

We assume that the aerosol size distribution, $f(x)$, can be represented as a histogram, linear spline, or more generally as a finite linear combination of basis functions $g_i(x)$, and we write

$$f(x) = \sum_{i=1}^{n} f_i g_i(x) \tag{4.2}$$

where n is a sufficiently large integer. Using Eq. (4.2), Eq. (4.1) has the vector representation

$$K\mathbf{f} = \mathbf{y} + \boldsymbol{\varepsilon} \tag{4.3}$$

where the elements of K are

$$K_{ij} = \int_0^1 k_i(x) g_j(x)\, dx \tag{4.4}$$

In this paper we will assume that $f(x)$ is a sample from a multivariate normal random process with the covariance matrix ρ whose elements represent

$$\rho_{ij} = \mathbf{E}\{(f_i - \bar{f}_i)(f_j - \bar{f}_j)\} \tag{4.5}$$

where $\mathbf{E}$ is the expectation operator, and the overbar denotes the expected value. In Section 4 we explain why it is necessary to assume that ρ is known, and we attempt to justify this assumption. We assume that ρ is positive definite, or equivalently that all nonzero linear functionals of $f(x)$ have a positive variance. *a priori.* A linear functional is denoted by ℓ, and

$$\ell(f(x)) = \mathbf{l}^T\mathbf{f} = \sum_{i=1}^{n} l_i f_i \tag{4.6}$$

where l_i is an element of the column vector $\mathbf{l}$. A simple calculation shows that $\mathbf{V}\{\ell(f(x))\} = \mathbf{l}^T\rho\mathbf{l}$, where $\mathbf{V}$ denotes variance.

After the measurements are obtained, a straightforward calculation shows that the new covariance matrix of $f(x)$, $\hat{\rho}$, is given by

$$\begin{aligned} \hat{\rho} &= (\rho^{-1} + K^T N^{-1} K)^{-1} & (4.7) \\ &= \rho - \rho_D & (4.8) \end{aligned}$$

where

$$\rho_D = \rho K^T (N + K\rho K^T)^{-1} K\rho \tag{4.9}$$

and where N is the covariance matrix of the errors in the measurements. Note that ρ_D is positive semi-definite, or if the matrix defined by the measurements is orthogonal to $\rho\mathbf{l}$, then the variance of the linear functional ℓ is unchanged by the measurement; otherwise the variance must decrease as expected.

Some linear functionals of $f(x)$ with covariance matrix $\hat{\rho}$ are worth noting. We refer to a linear functional, ℓ_a, as a weighted average if $\sum_{i=1}^{n} l_{ai} = 1$ and as a nonnegative weighted average if additionally all the l_{ai}s are nonnegative. For example, the concentration in a given size interval is proportional to a nonnegative weighted average. The weighted average of $f(x)$ with the minimum variance solves

$$\begin{array}{ll} \text{minimize} & \mathbf{l_a}^T \hat{\rho} \mathbf{l_a} \\ \text{subject to} & \sum_{i=1}^{n} l_{ai} = 1 \end{array}$$

or,

$$\mathbf{l_a} = \frac{\hat{\rho}^{-1}\mathbf{u}}{\mathbf{u}^T \hat{\rho}^{-1}\mathbf{u}} \tag{4.10}$$

where $\mathbf{u}$ is a column vector with every element equal to 1. Also, note that $\mathbf{l_a}^T \hat{\rho} \mathbf{l_a}$ is strictly convex. Thus, if i satisfies $\hat{\rho}_{ii} \geq \hat{\rho}_{jj}$, then $l_{ai} = 1$ defines a nonnegative weighted average with the maximum variance. The weighted average with the maximum variance does not exist since the variance of weighted averages is not bounded.

Additionally, we can look at the linear functionals, ℓ_s, that lie on the sphere $||\mathbf{l_s}||_2 = 1$. Note that

$$\hat{\rho} = U_{\hat{\rho}} S_{\hat{\rho}}^2 U_{\hat{\rho}}^T \tag{4.11}$$

where $U_{\hat{\rho}}$ is an orthonormal matrix of eigenvectors of $\hat{\rho}$, and $S_{\hat{\rho}}^2$ is the diagonal matrix of positive eigenvalues. Thus, ℓ_s with the largest variance is represented by an eigenvector of $\hat{\rho}$ with the largest eigenvalue, and the variance is equal to the largest eigenvalue. Similarly, ℓ_s with the least variance is represented by an eigenvector of $\hat{\rho}$ with the smallest eigenvalue, and the variance is equal to the smallest eigenvalue. These maximal linear functionals are important, not only because they bound the variance that one will obtain for a given linear functional, but also they can characterize the weak and strong points in a measurement system.

4.3 Information

Information theory provides the useful concept of the amount of information provided by a set of measurements. The entropy or uncertainty in the random variable x, $H(x)$, is [5]

$$H(x) = \int_{-\infty}^{+\infty} p_x(y) \log p_x(y)\, dy \tag{4.12}$$

If x has variance σ, then the uncertainty of x cannot be greater than that of a normal random variable with standard deviation σ [5].

If a measurement, y_1, provides new information about x, then the density function of x becomes $p_{x/y_1}(z)$, and the uncertainty of x becomes $H(x/y_1)$. The information conveyed about x by y_1 is denoted as $I(x|y_1)$,

$$I(x|y_1) = H(x) - H(x/y_1) \tag{4.13}$$

and simply measures a reduction in the uncertainty of x. In general we denote the information added by the measurements $\bigcup_{i=1}^{n_z} z_i$ to the random variable x that has been previously characterized by the data $\bigcup_{i=1}^{n_y} y_i$ as $I(x/\bigcup_{i=1}^{n_y} y_i|\bigcup_{i=1}^{n_z} z_i)$. This measure of information satisfies the following intuitive requirement:

$$\begin{aligned} I(\ell(f)|\bigcup_{i=1}^{2} y_i) &= I(\ell(f)|y_1) + I(\ell(f)/y_1|y_2) && (4.14)\\ &= I(\ell(f)|y_2) + I(\ell(f)/y_2|y_1) && (4.15) \end{aligned}$$

or information is additive, and the sum is independent of the order in which it is added. I can be thought of as a state function in much the same way that ΔS, entropy change, is a state function in thermodynamics.

For example, assume an unknown constant x^t is characterized as a normal random variable with variance σ_x^2, and that an independent datum can be used alone to characterize x^t as a normal random variable with variance σ_y^2. Then one can show by substituting the normal density function into Eq. (4.13) that

$$\begin{aligned} I(x|y_1) &= \frac{1}{2}\log(1+\frac{\sigma_x^2}{\sigma_y^2}) \\ &\approx \log\frac{\sigma_x}{\sigma_y} \qquad \sigma_x \gg \sigma_y \\ &\approx \frac{\sigma_x^2}{2\sigma_y^2} \qquad \sigma_x \ll \sigma_y \end{aligned}$$

It is interesting to note that here the information is independent of the mean values of the random variables.

One can extend these calculations to apply to the n-dimensional $f(x)$ to look at the entropy reduction of the random discretized size distribution; however, more useful information can be obtained by examining the information added to linear functionals of the size distribution. Here, one is able to determine where information decrease is (or is not) occurring. If Eq. (4.7) is substituted into Eq. (4.13), then we find the amount of information added by the p measurements with response matrix K to $\ell(f)$ is given by

$$I(\ell(f)|K) = -\tfrac{1}{2}\log[1-\frac{\mathbf{1}^T\rho_D\mathbf{1}}{\mathbf{1}\rho\mathbf{1}}] \qquad (4.16)$$

which will always be finite since one can assume N is positive definite. Since $\ell(f)$ is a normal random variable and the measurements are linear, the information added by a set of measurements is independent of the measurements, $\mathbf{y}$. Also, the information that is added is always nonnegative.

An important tool in the analysis of information content is the singular value decomposition (SVD):

$$K = USV^T \tag{4.17}$$

Here, S is a diagonal matrix of positive generalized eigenvectors and U is a $p \times p$ rotation matrix of left-hand generalized eigenvectors, and similarly V is the $n \times n$ rotation matrix of right-hand eigenvectors. If ρ can be approximated as rI where r is a positive scalar, and $N \approx \eta I$ where η is a positive scalar then substituting Eq. (4.17) into Eq. (4.16) yields

$$I(\ell(f)|K) \approx -\tfrac{1}{2}\log[1 - \frac{\sum_{i=1}^{p} \frac{\hat{l}_i^2 r S_{ii}}{\eta + r S_{ii}}}{\sum_{i=1}^{n} \hat{l}_i^2}] \tag{4.18}$$

where $\hat{\mathbf{l}}$ is the rotated linear functional $\mathbf{l}V$. This expression highlights the importance of the SVD in determining the information content of an instrument. The instrument's information contribution depends on how well the linear functional is represented along the principal axes that are defined by V.

In most cases of interest in aerosol science, we can assume $\eta \ll rS_{ii}$, or roughly speaking the variance in our estimate of a measurement is much smaller after the measurement has been observed, or

$$\frac{r^2 S_{ii}^2}{\eta^2 + r^2 S_{ii}^2} \approx 1 - \frac{\eta^2}{r^2 S_{ii}^2} \tag{4.19}$$

This may not hold for all S_{ii}, and if not, we also assume the corresponding $\hat{l}_i$'s are not disproportionately large. With these assumptions, we find

$$I(\ell(f)|K) \approx \log\frac{r}{\eta} - \tfrac{1}{2}\log\frac{\sum_{i=1}^{n} \frac{\hat{l}_i^2}{S_{ii}^2}}{\sum_{i=1}^{n} \hat{l}_i^2} \tag{4.20}$$

Eq. (4.20) points out an approximate relationship between the magnitude of ρ and N and the amount of information provided by an instrument. This motivates us to define a measure of information that is less sensitive to the magnitude of ρ and N, namely

$$i(\ell(f)|K) = I(\ell(f)|K) + \log\frac{p\,\mathbf{tr}[\rho]}{n\,\mathbf{tr}[N]} \tag{4.21}$$

where $\mathbf{tr}\,[\;\;]$ is the trace operator that equals the sum of the diagonals, the sum of the eigenvalues, and the Frobenius norm of the square root of the matrix argument. Thus, $i(\ell(f)|K)$ should provide knowledge of the information provided by an instrument over a large range of scale changes of ρ and N.

4.4 Application to instrument analysis and design

We will examine the information content of three instruments: the scanning electrical mobility spectrometer (SEMS) [46], the impactor, and the diffusion battery (DB) through some illustrative numerical experiments. In all cases, we assume $\rho_{ii} = 1$, and that the off-diagonal elements decrease linearly so that f_i and f_j are uncorrelated if the corresponding diameter ratio is less than $\sqrt{10}$ or half of a decade. The covariance matrix ensures that the sample size distributions are reasonable and not "white noise", and the covariance matrix used here is only illustrative. As the order of $\rho \longrightarrow \infty$, this linear decay will become undesirable since the sample functions will not be continuous in the expectation sense [10]. If the errors in the data are assumed to be independent, then N_{ii} is assumed to equal the integral of the corresponding kernel function; thus, $I(\ell(f)|K)$ is an approximation to $i(\ell(f)|K)$.

If the errors from an instrument are dependent, then all of the channels are assumed to have a common error source that has the same standard deviation as the corresponding independent error source. For example the errors, ε, in the data of a 3 channel instrument would satisfy

$$\varepsilon = \begin{pmatrix} c_1 & & & c_1 \\ & c_2 & & c_2 \\ & & c_3 & c_3 \end{pmatrix} \mathcal{N} \tag{4.22}$$

where $\sqrt{2}c_i$ equals $\sqrt{N_{ii}}$ in the independent case and $\mathcal{N}$ is a vector of independent normal random vectors with mean = 0, and variance = 1. Note that the constant defined in Eq. (4.21) is the same for both the dependent and independent errors.

In particular, we will look at $I(\ell_i(f)|K)$ where $\ell_i(f) = f_i$. This characterizes the information that is provided about particles in a narrow size range and highlights the particle sizes at which the instrument provides or fails to provide information. Note that a plot of $i(\ell_i(f)|K)$ is not as useful since some $\hat{l}_j$ will be large where S_{jj} is small. Also we look at $I(\ell_{I_1}(f)|K)$ where I_1 denotes the integral of the size distribution over the first one-third section of the inversion interval. Here, the inversion interval coincides with the support of the kernel functions. Similar calculations are performed for I_2, I_3, and I_{total}, where I_{total} represents the integral of the size distribution.

The kernel functions for the three instruments are shown in Figure 2 along with the amount of information provided by the instruments about particles with diameter d. Note, for example, that for particles in the 0.03–0.1 μm size range, $I(\ell_i(f)|K)$ is 2-3 times times larger for the SEMS than the impactor; the interpretation is that the reduction in variance provided by the SEMS is approximately 10 times larger that the variance reduction provided by the impactor. The signal to noise ratio is assumed to be roughly the same for all channels and instruments. It is also interesting to note that one can explain the sudden decrease in $I(\ell_i(f)|K)$ for the SEMS at the smaller particle sizes and the more gradual decrease at the larger end of the size range simply by looking at the kernel functions.

Application: diffusion battery

The diffusion battery is of interest because of the difficulty that has been noted by some when inverting diffusion battery data [49]. This difficulty is sometimes incorrectly attributed to the inversion algorithm instead of the diffusion battery's inherent lack of information.

One interesting observation is that we can make two different assumptions about the diffusion battery measurements: in the first, the data represent the number of particles collected by the i^{th} stage of screens, and in the second the data represent the number of particle that penetrate the i^{th} stage of screens. The two points of view lead to equivalent inversion problems as long as N is modified when we change from one point of view to the other. If N is unchanged, then we will obtain different conclusions on the information content of the diffusion battery as shown in Figure 3. Specifically, the diffusion battery has noticeably larger information content for the smaller particles if one can assume the fraction of particles collected have independent errors. Also in Figure 3, we show $I(\ell_i(f)|K)$ when a dependent error source corrupts the data.

We next examine the affect of certain modifications on information content of the diffusion battery. A straightforward approach that is guaranteed to improve the information content is simply to add a single SEMS measurement to the data. We calculated the information that was added by the diffusion battery at various SEMS inner rod voltages to the 4 integrals of the size distribution described above. The results are shown in Figure 4 and illustrate the anticipated results. For example, since the diffusion battery provides less information about smaller particles, the SEMS-DB combination adds more information to the total integral as the mean diameter of the SEMS kernel is decreased.

The integral of the smaller particles is difficult to estimate with a diffusion battery because the contribution to the signal by the larger particles adds uncertainty to the estimation. Therefore, it seems reasonable to expect that a filter could eliminate the larger particles and increase the amount of information provided by the diffusion battery about the smaller particles. The effect of the filter can be more clearly seen by examining Eq. (4.7). First, the filter decreases the magnitude of N by decreasing the signal, and second redirects the right-hand eigenvectors of K to lie in the same direction as ℓ_I.

To test this, we simulated a filter with a geometric standard deviation of 1.4 and looked at the amount of information added by the diffusion battery to the various integrals. As expected, the information added by the diffusion battery increased for the smaller particles as shown in Figure 5. For the center section, the information increased as the larger particles were eliminated until the filter started to remove particles in the center section. Additionally, as one should expect, the information about the total integral decreased because components of K parallel to ℓ_{It} were eliminated.

Application: multiple impactors

A single impactor provides little information about the size distribution as shown in Figure 2. However, if multiple impactors are used, then an unlimited amount of information can be added as suggested by Figure 6a. Notice that it is not necessary for the kernel functions to respond over a narrow size interval, as does the SEMS, for a set of measurements to provide a large amount of information. The increase in information is not as rapid as one could obtain by increasing SEMS channels because, roughly speaking, the broad response of the impactor leads more quickly to approximate linear dependence, and therefore the magnitude of the corresponding eigenvalues of K is less. If all of the impactors have a dependent error source, then the variance of a linear functional is substantially increased. This simply reflects that all of the data are affected in the same way by the dependent source, and making more measurements does not reduce the effect of this error source.

The most realistic case is that each impactor has its own dependent error source, or the errors from one impactor to another are uncorrelated. The value of having multiple instruments for measuring linear functionals of the size distribution is clearly pointed out in the difference between Figures 6b and 6c.

Application: SEMS

Finally, we examine the effect of dependent error sources on the SEMS by increasing the variance of the independent error source from 0.1 % of the total variance to 100 % of the total variance. The information $I(\ell(f)|K)$ added by the SEMS to the various integrals is shown in Figure 7, where a in the Figure represents the ratio of the variance of the independent source to the total variance. Note the large reduction in the amount of information provided by the SEMS to these linear functionals in the presense of dependent error sources. Again, the explanation is that none of the data are able to detect the errors generated by the dependent source since all of the data are affected in the same manner.

4.5 The covariance matrix

One could take this analysis a step further and use the expected distribution with the covariance matrix to determine the most likely solution as in [16]. This approach is undesirable, however, because the most likely solution is sensitive to the estimated mean distribution, and a poor estimate can make convergence to the true solution painfully slow. Additionally, it is expensive to estimate the mean distribution and covariance matrix. Therefore, we prefer regularization to find the optimal solution, and the techniques described in this paper are best left to estimating the variance of the solution.

A weakness one can anticipate with this approach is that information about the linear inequality constraints cannot be conveyed by the covariance matrix alone. We anticipate that often this will not be a serious problem because

- often the information provided by the constraints is secondary to the information provided by the data
- the results in this paper are used only to obtain an estimate of the variance, and this estimate should be reasonable for general random processes, even those with inequality constraints.

A second problem is that the experimentalist may have difficulty deciding on a best ρ and N. Note, though, that the covariance matrix is simply a method of providing subjective information about the size distribution that enables one to exclude unrealistic size distributions, e.g., dirac-delta functions. When we say "this distribution simply cannot be correct", then we have available some idea of what is realistic or unrealistic, and this information should be conveyed by the covariance matrix. An example of specifying a realistic covariance matrix can be found in [16] along with some discussion. Finally, the results obtained in this paper can always be prefaced with the statement, "if the covariance matrix is given by ρ, then one obtains the following results." For this reason, $i(\ell(f)|K)$ is a useful measure since it is less sensitive to the magnitude of ρ and N.

An alternative method for examining the size of the solution set, Extreme-Value Estimation (EVE), has been presented recently by Paatero and co-workers [36,37]. The technique is to

1. discretize the integral equation
2. define a maximum acceptable error in the data, δ_{max}
3. look at the maximum and minimum of selected linear functionals of the size distribution in the feasible set defined by the inequality constraints and δ_{max}

Here, the inequality constraints are usually nonnegativity of the size distribution. This approach is undesirable, however, not only because the results depend upon the specified δ_{max}, but also because the convex set is not bounded. As a result, the extremes represent the unrealistic cases. Another problem is that all solutions in the convex set are considered equally important when in fact some are next to impossible. A simple example can point out the underlying shortcomings of this technique. Suppose that one only knows the integral of the unknown size distribution. What conclusions could be made about $f(\hat{x})$ for example? EVE can only conclude that $0 \leq f(\hat{x}) < +\infty$ for any value of δ_{max}, even though we may know *a priori*, for example, that $f(\hat{x}) > 100$ is likely, $f(\hat{x}) > 10^4$ is unlikely, and that $f(\hat{x}) > 10^{10}$ is impossible. This problem can exist even when there are a large number of measurements. The resulting conclusions thus depend on a fortuitous discretization and choice of linear functionals.

Equal weighting to all solutions is undesirable because it produces a misleading measure of the size of the solution set. For example if one applies EVE reasoning to estimate the size of the sample space of a normal random variable, one would conclude the set ranged from $-\infty$ to $+\infty$. Here, as in our case, a more useful description of the size of the sample space is the variance.

4.6 Conclusions

The estimation of the variance of linear functionals of the size distribution is a useful tool in the interpretation of aerosol size distribution data. If there is insufficient data to determine a single "best" solution, then this tool can be used to reflect one's uncertainty in an inverted solution. In addition to estimating the variance of important properties, like moments, one can compare measurement systems before an experiment is actually carried out and decide on the best design without expending a large and possibly misleading effort on inverting simulated data.

The examples demonstrate why, for example, it is difficult to resolve size distributions with particles less than 8 nm in diameter from diffusion battery data. Additionally, some possibilities for remedying this problem were presented. A filter, for example, can be useful because particles of the distribution that contribute to the uncertainty instead of the linear functional are removed.

Finally, dependent errors in an instrument can significantly affect the amount of information provided about linear functionals. For this reason, multiple instruments can be valuable when attempting to estimate linear functionals of the size distribution because the likelihood of dependent error sources is drastically reduced.

List of Figures

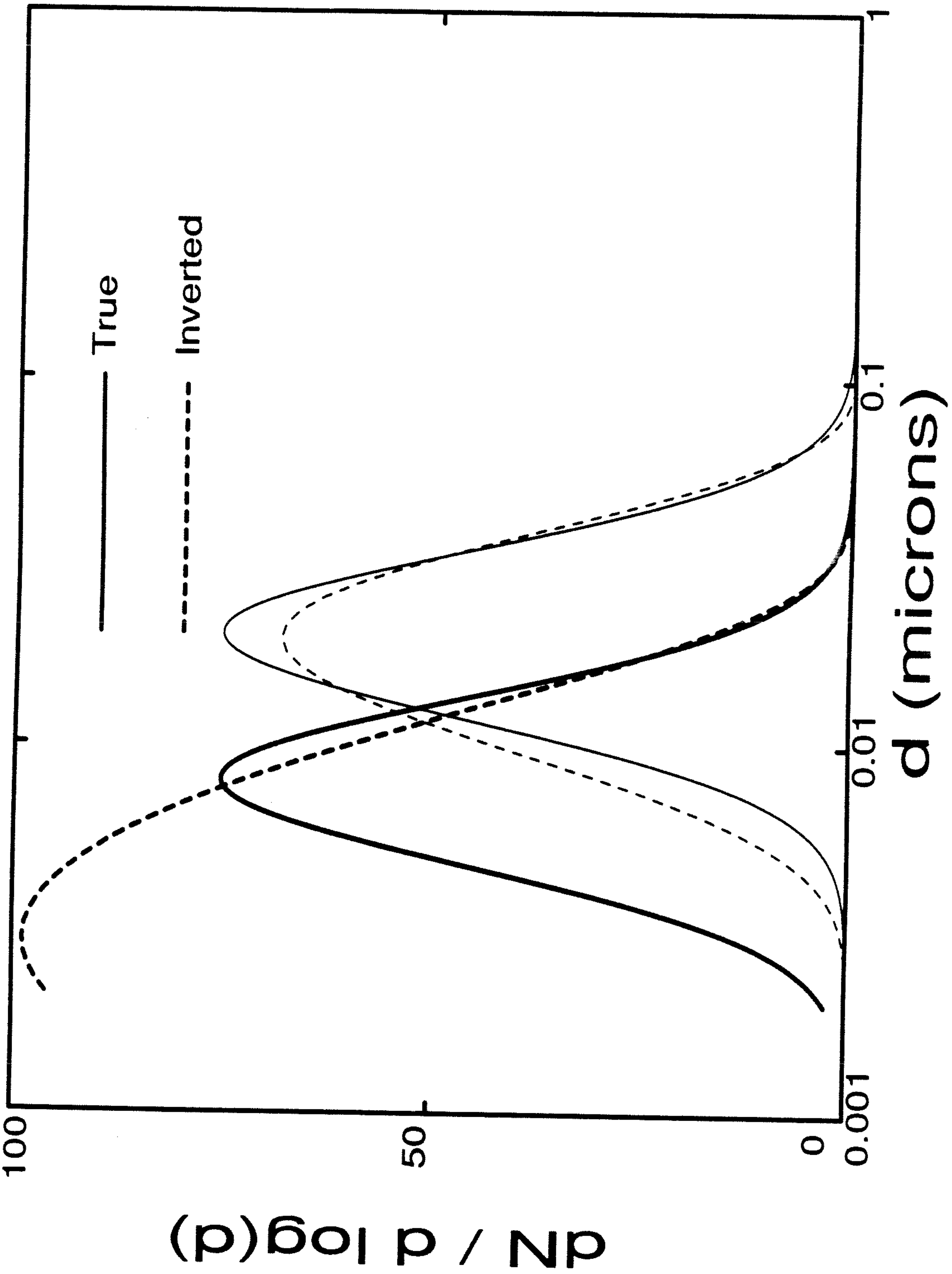


Figure 1

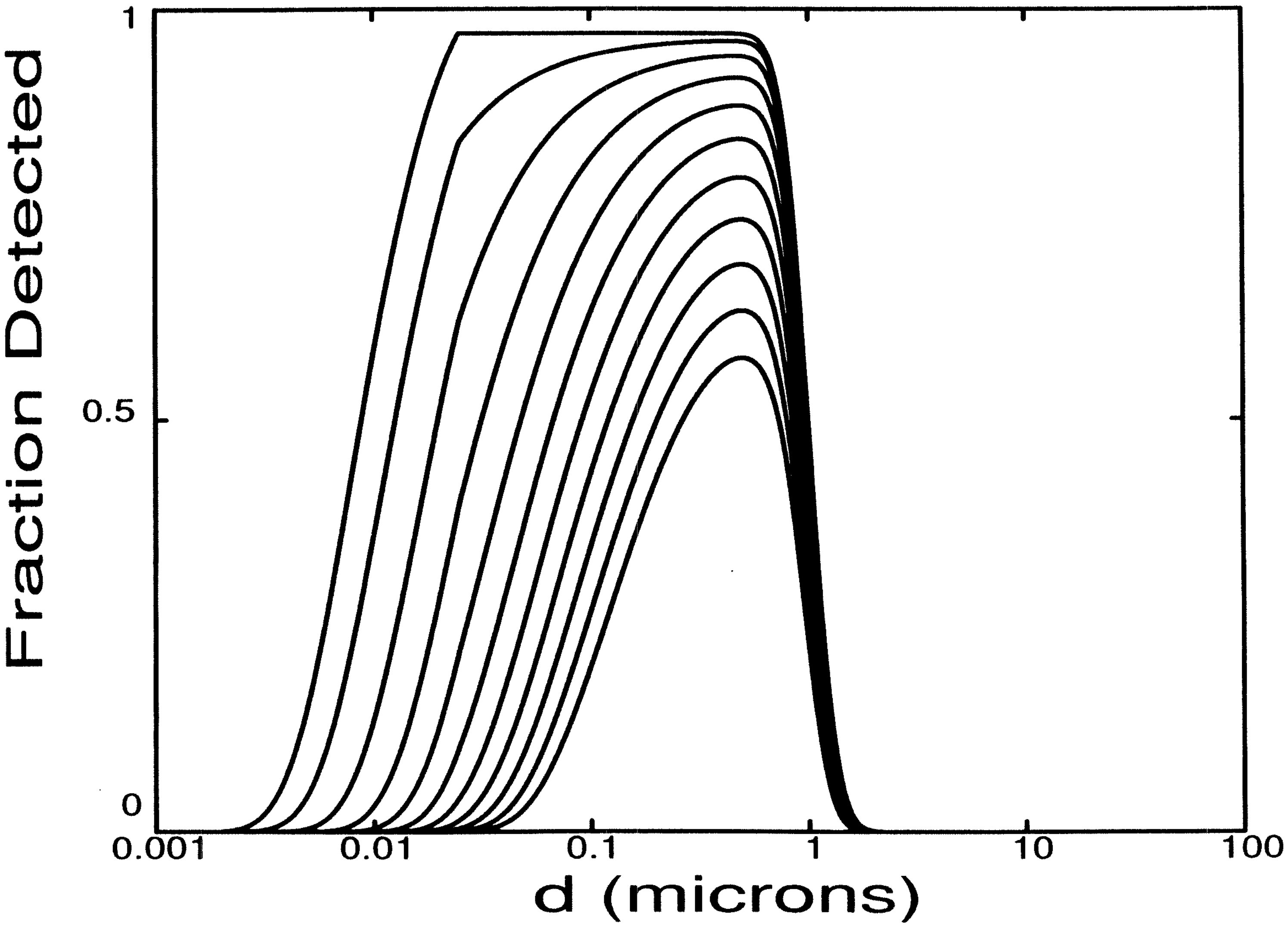


Figure 2a

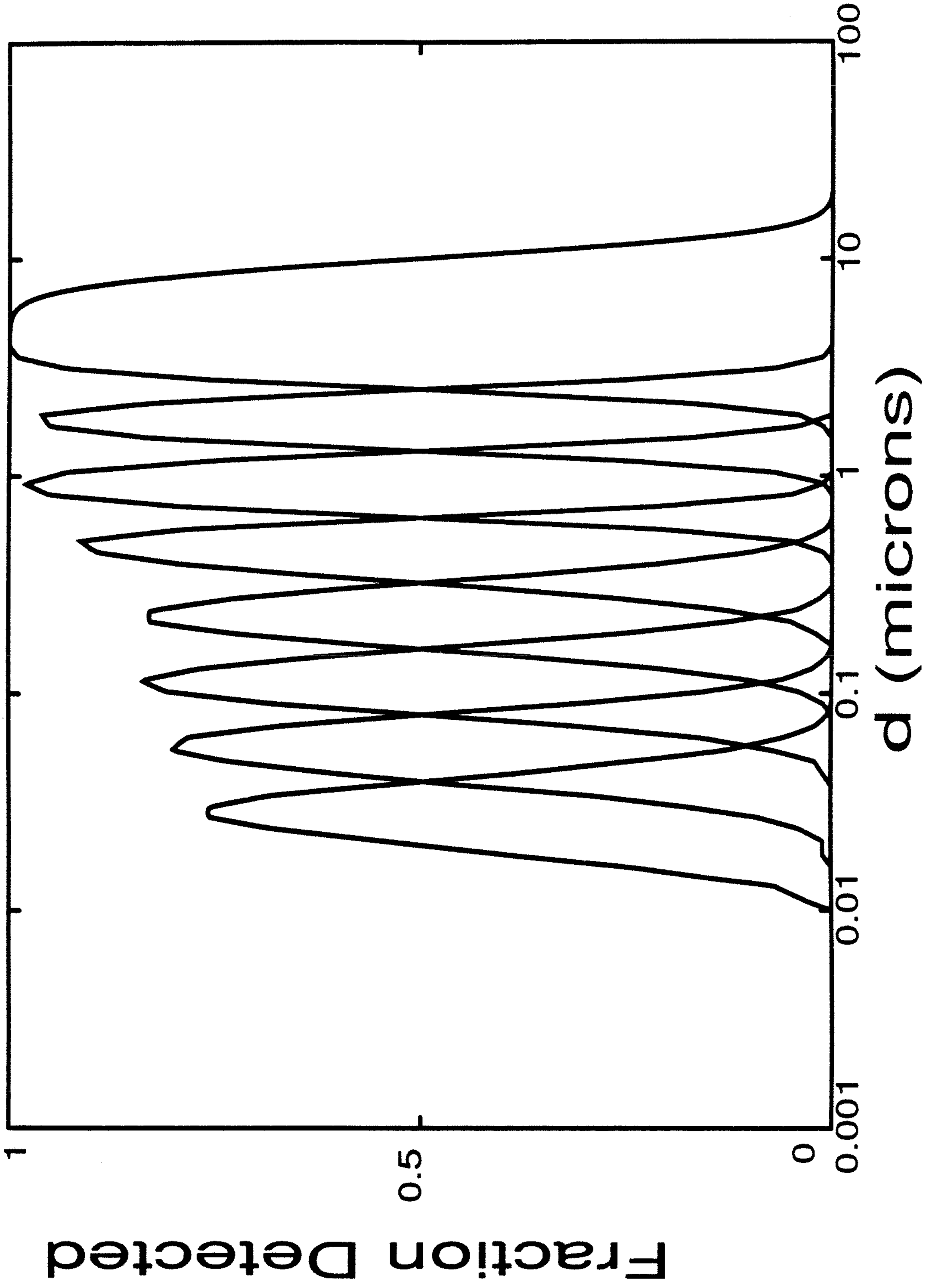


Figure 2b

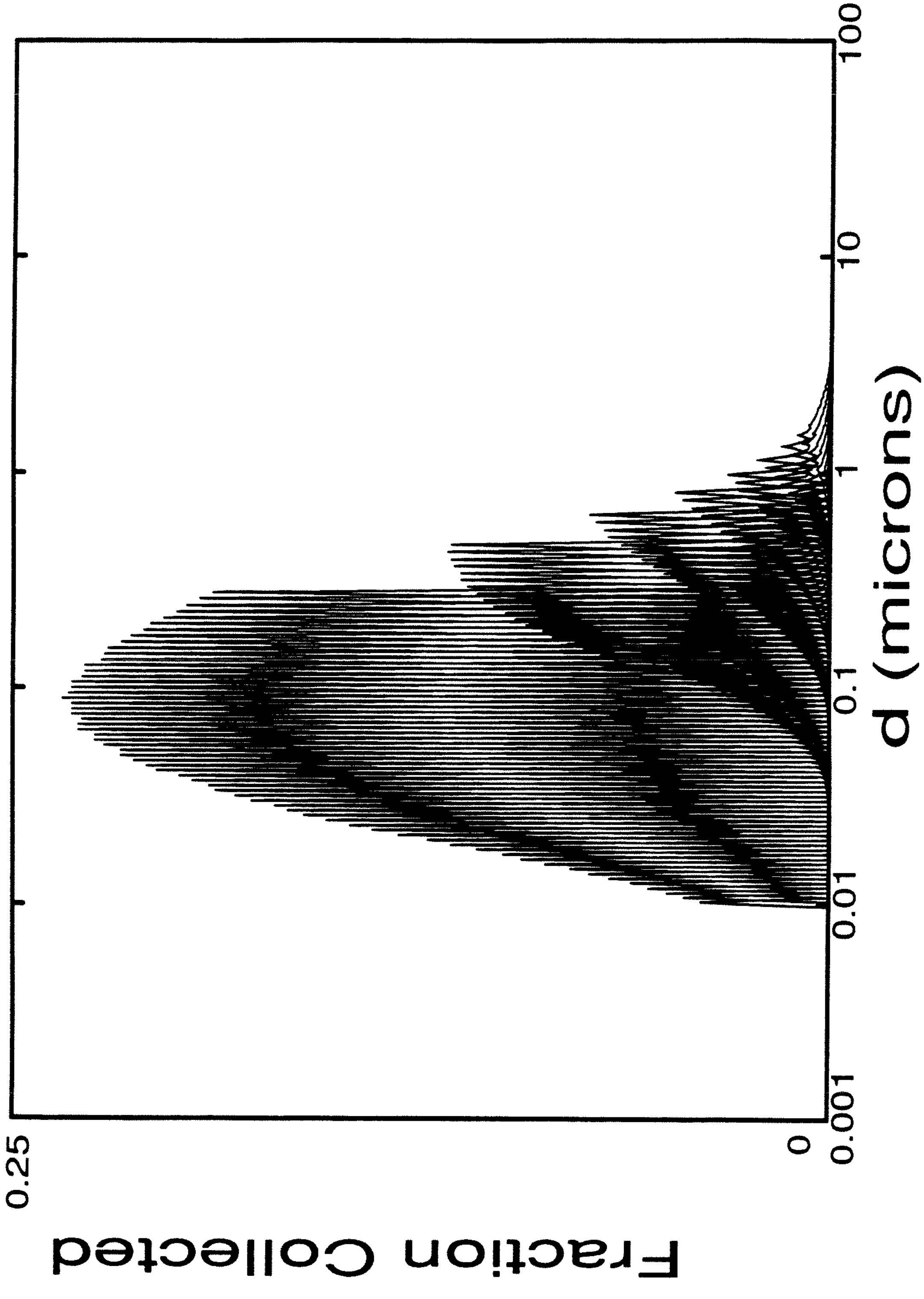


Figure 2c

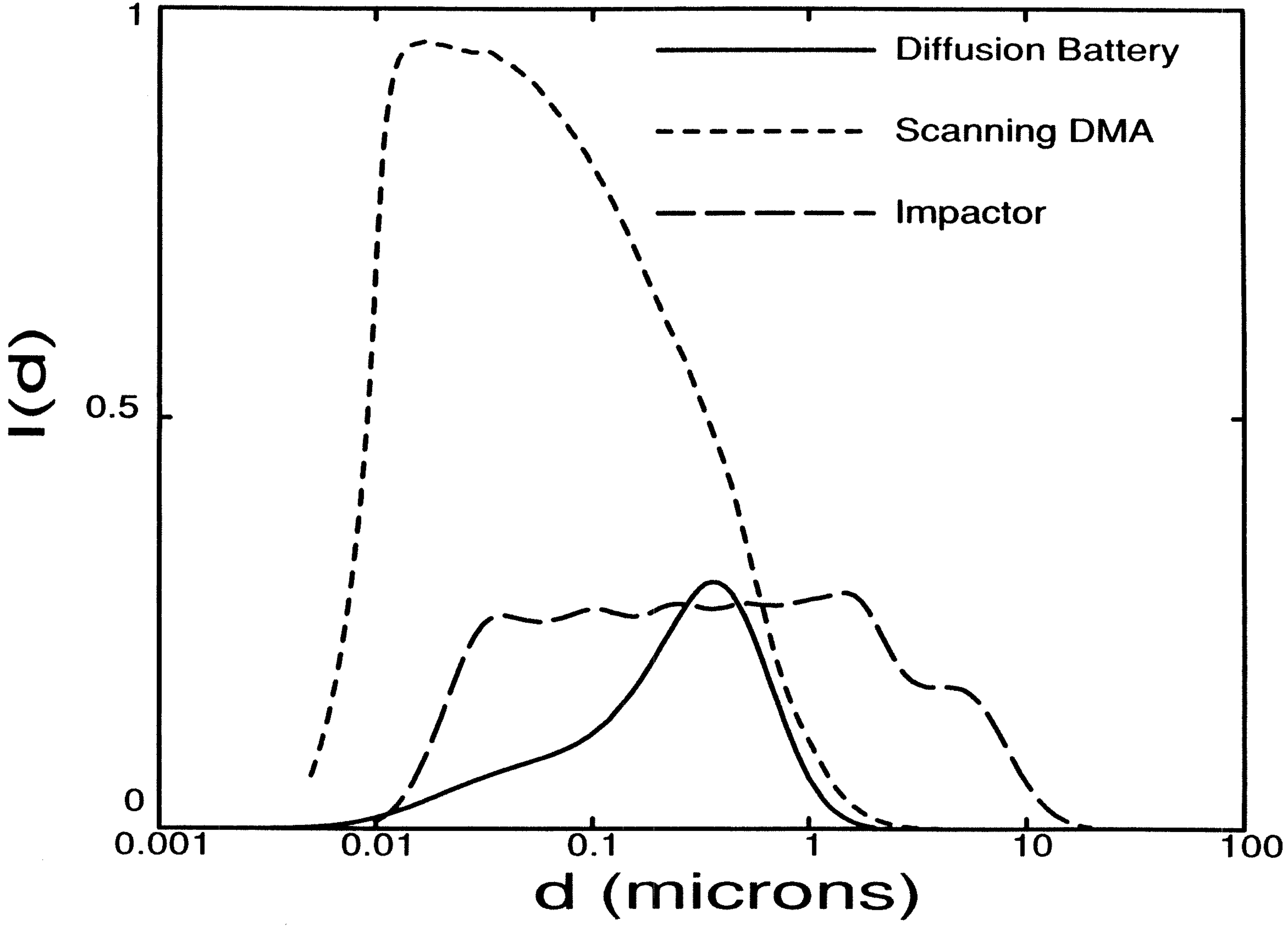

1
0.5
0
I(d)
0.001
0.01
0.1
1
10
100
d (microns)
Diffusion Battery
Scanning DMA
Impactor

Figure 2d

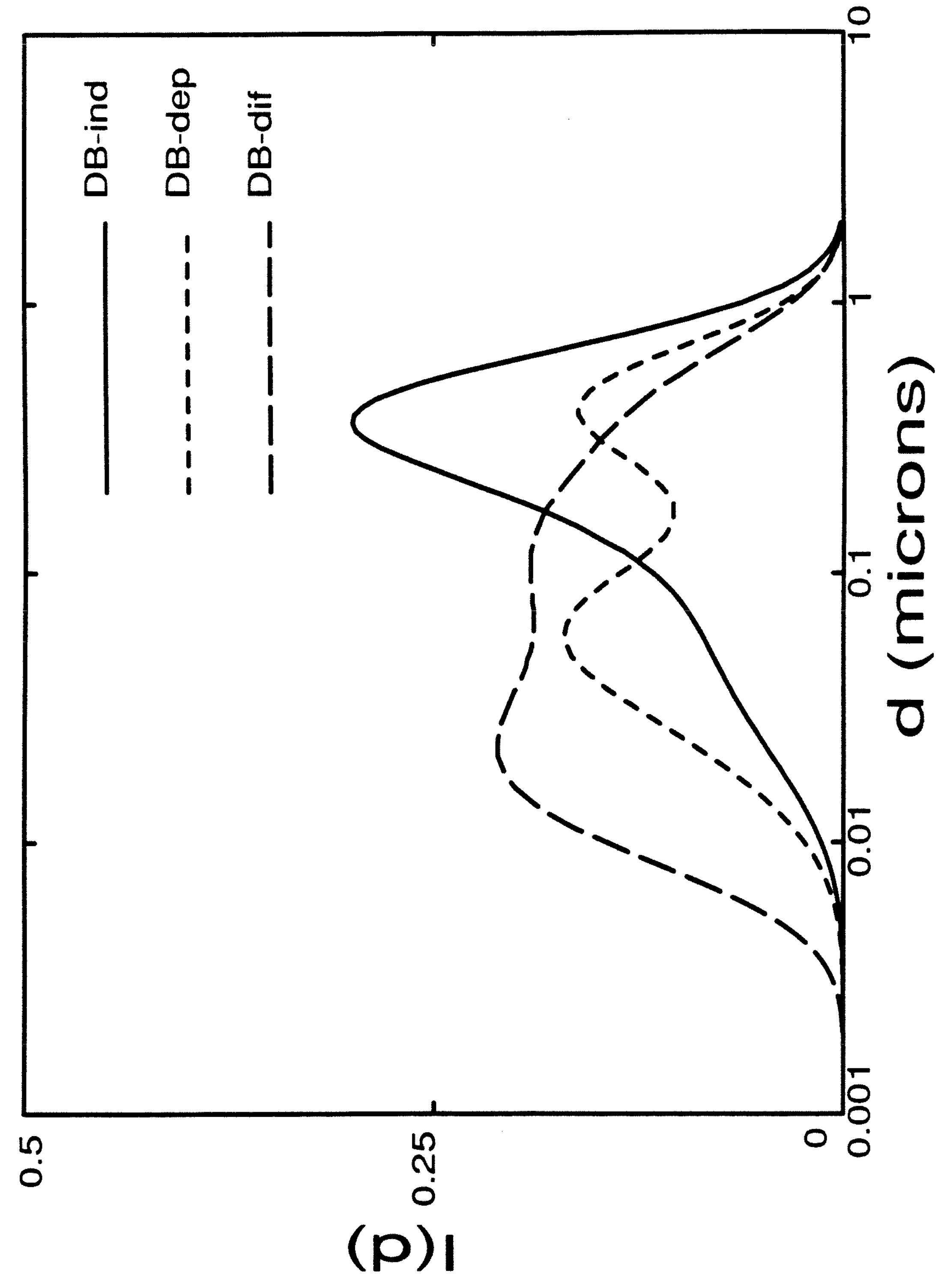


Figure 3

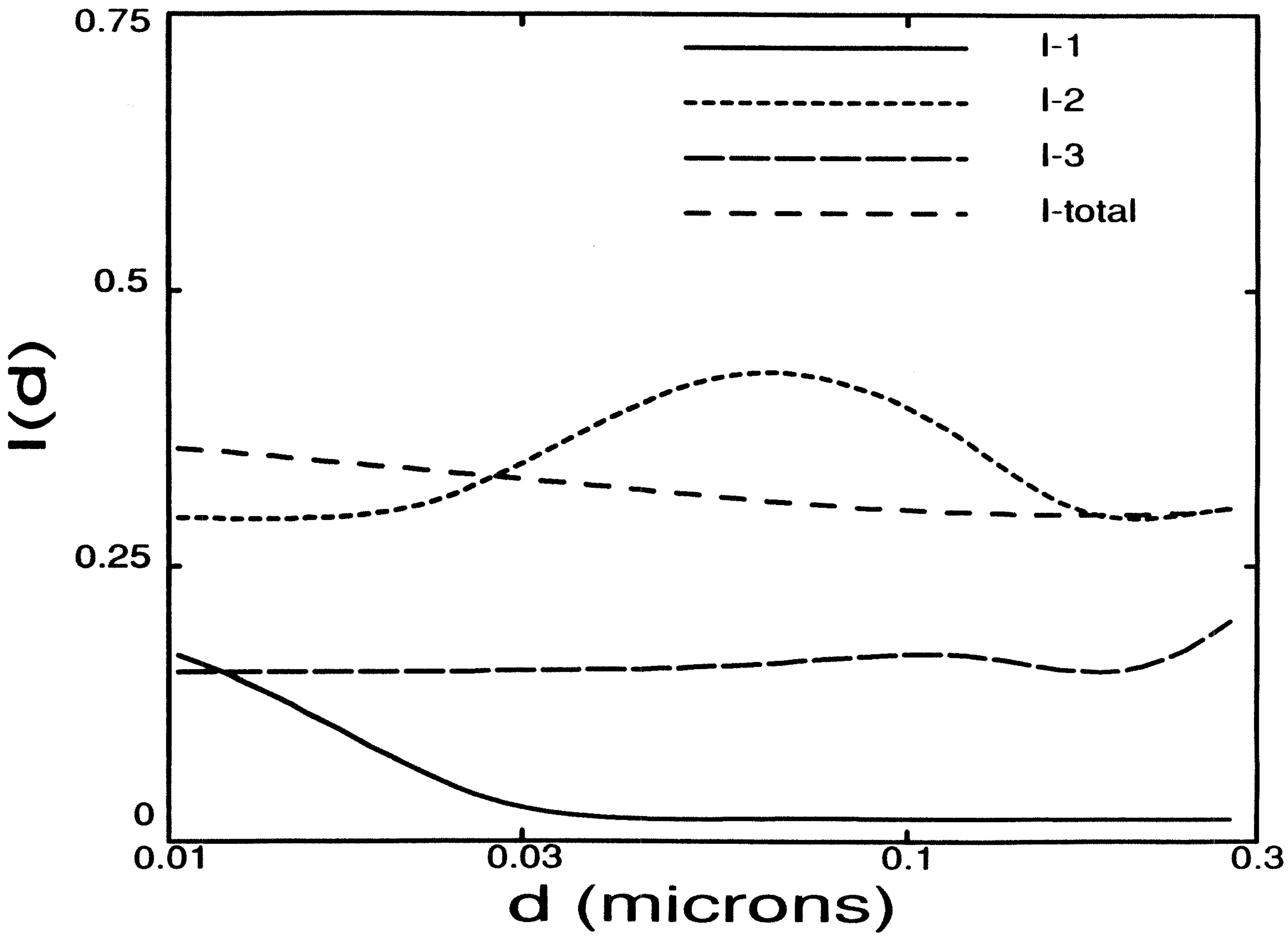


Figure 4

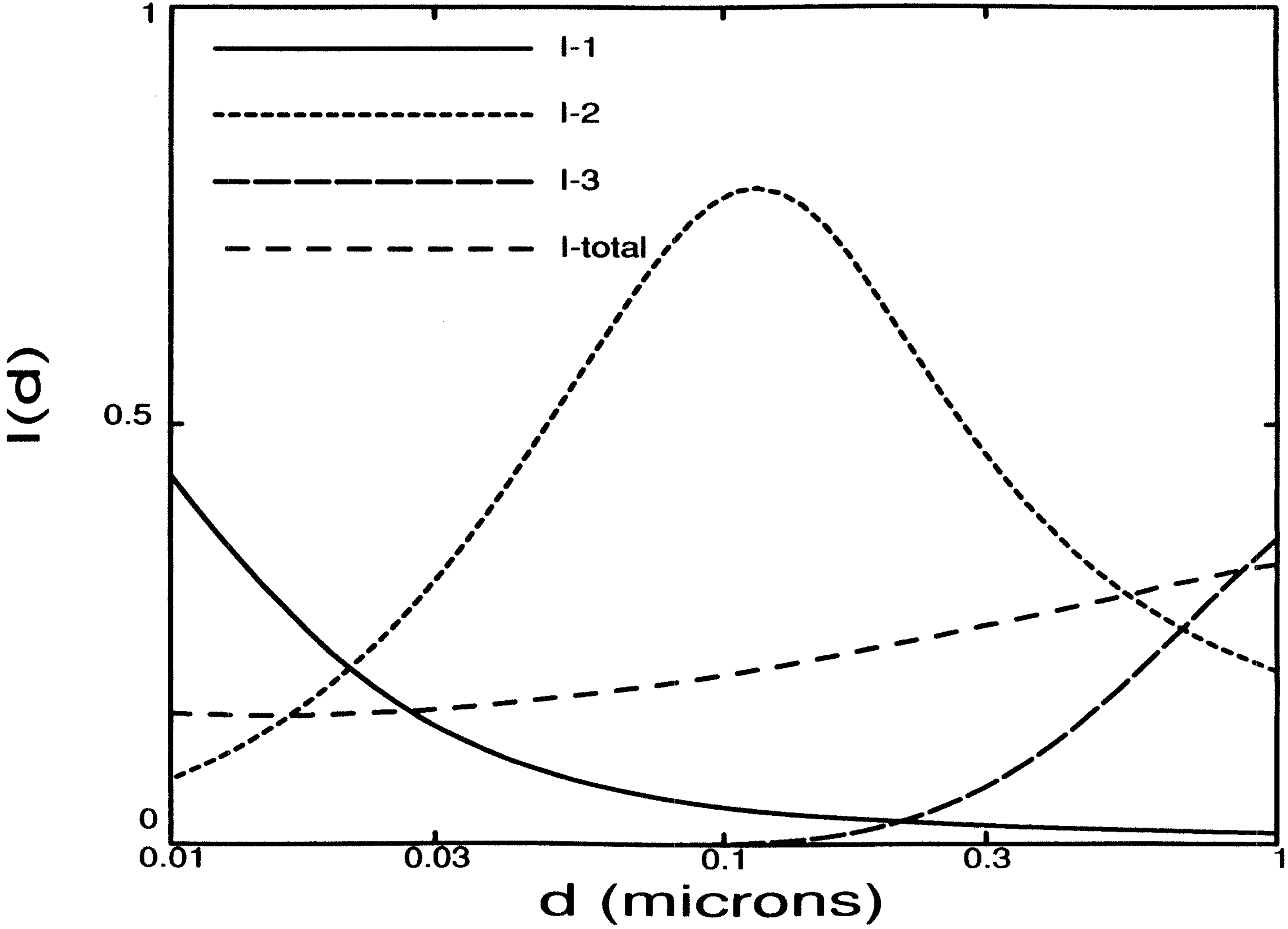


Figure 5

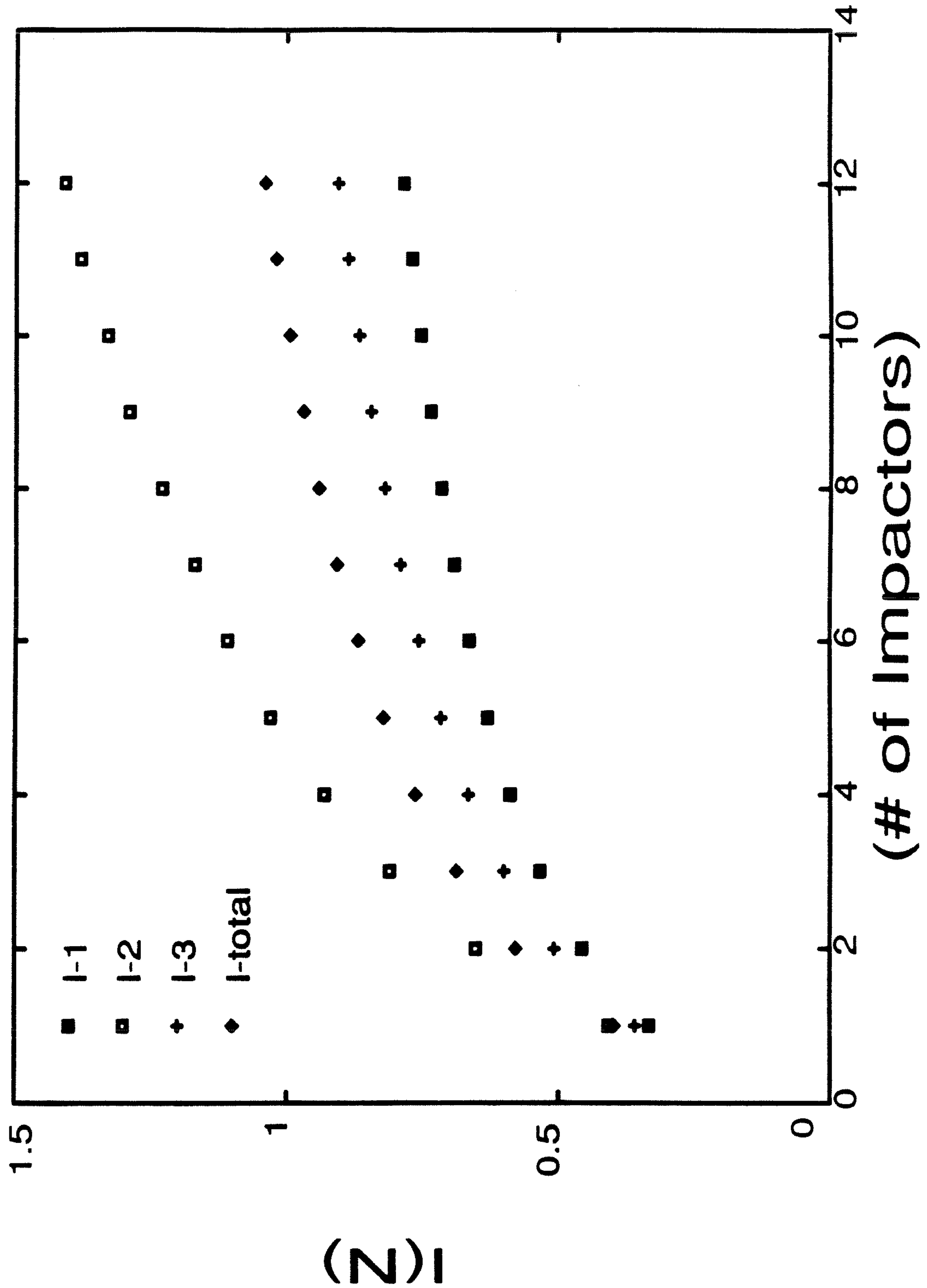


Figure 6a

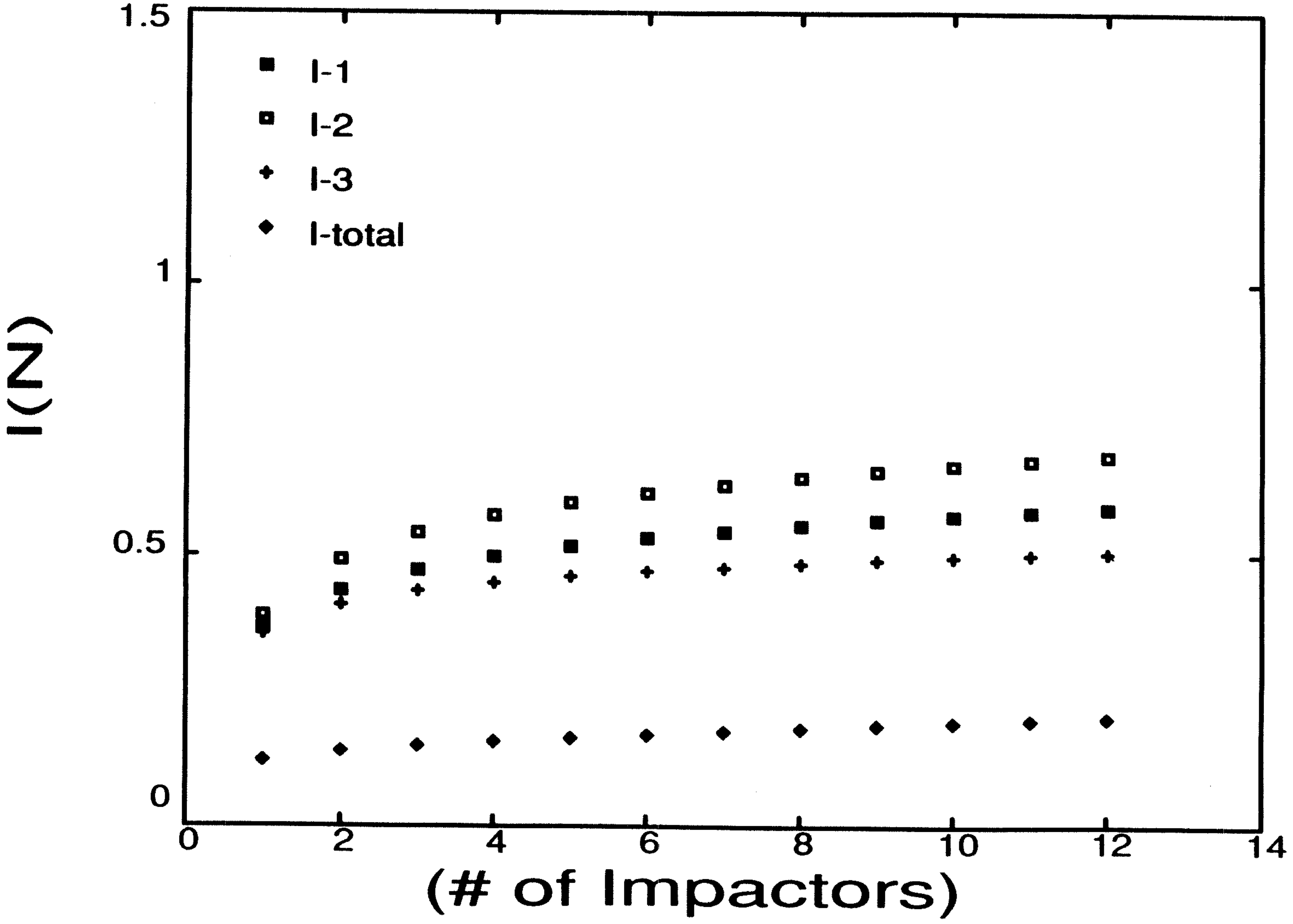


Figure 6b

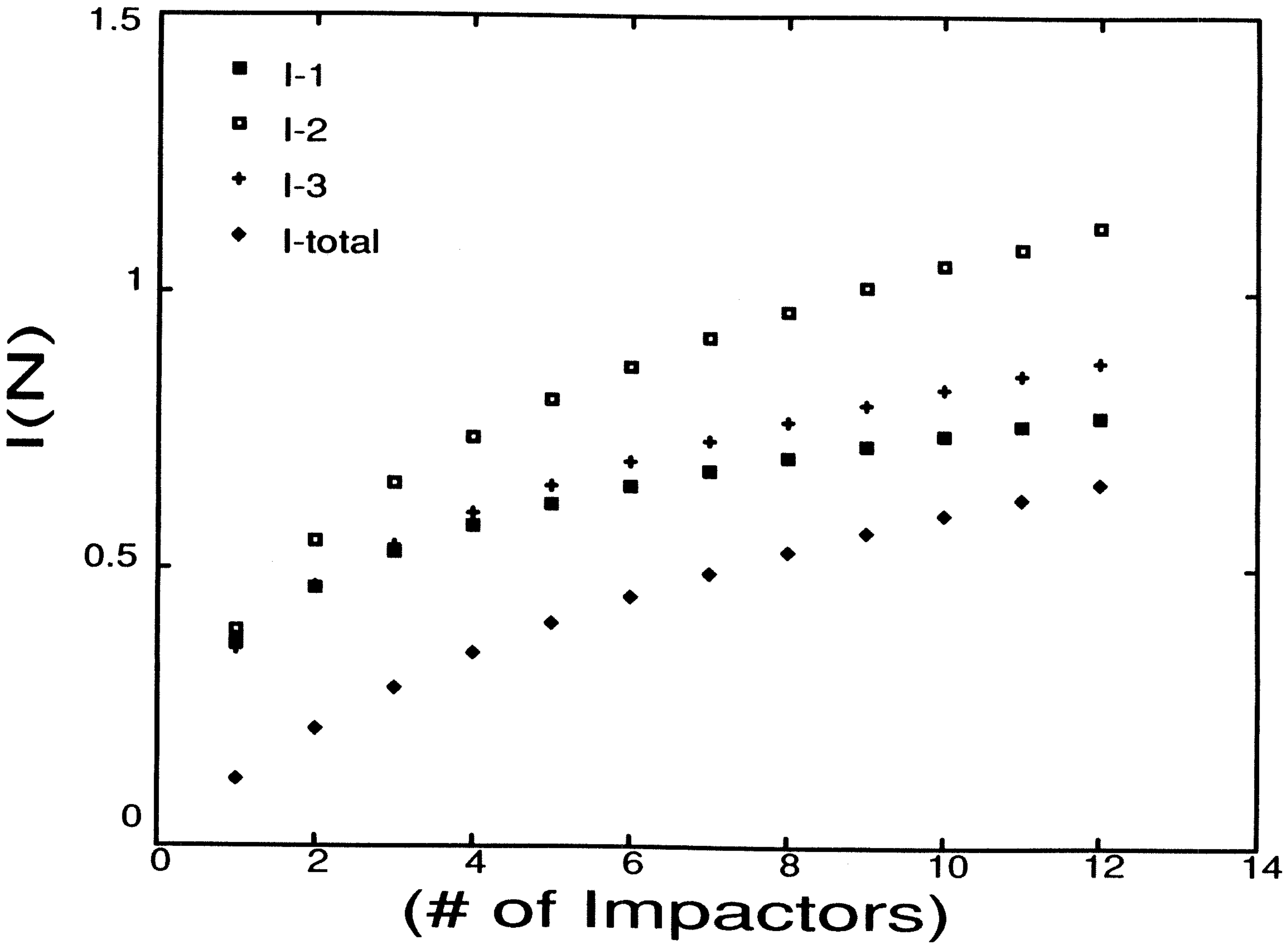


Figure 6c

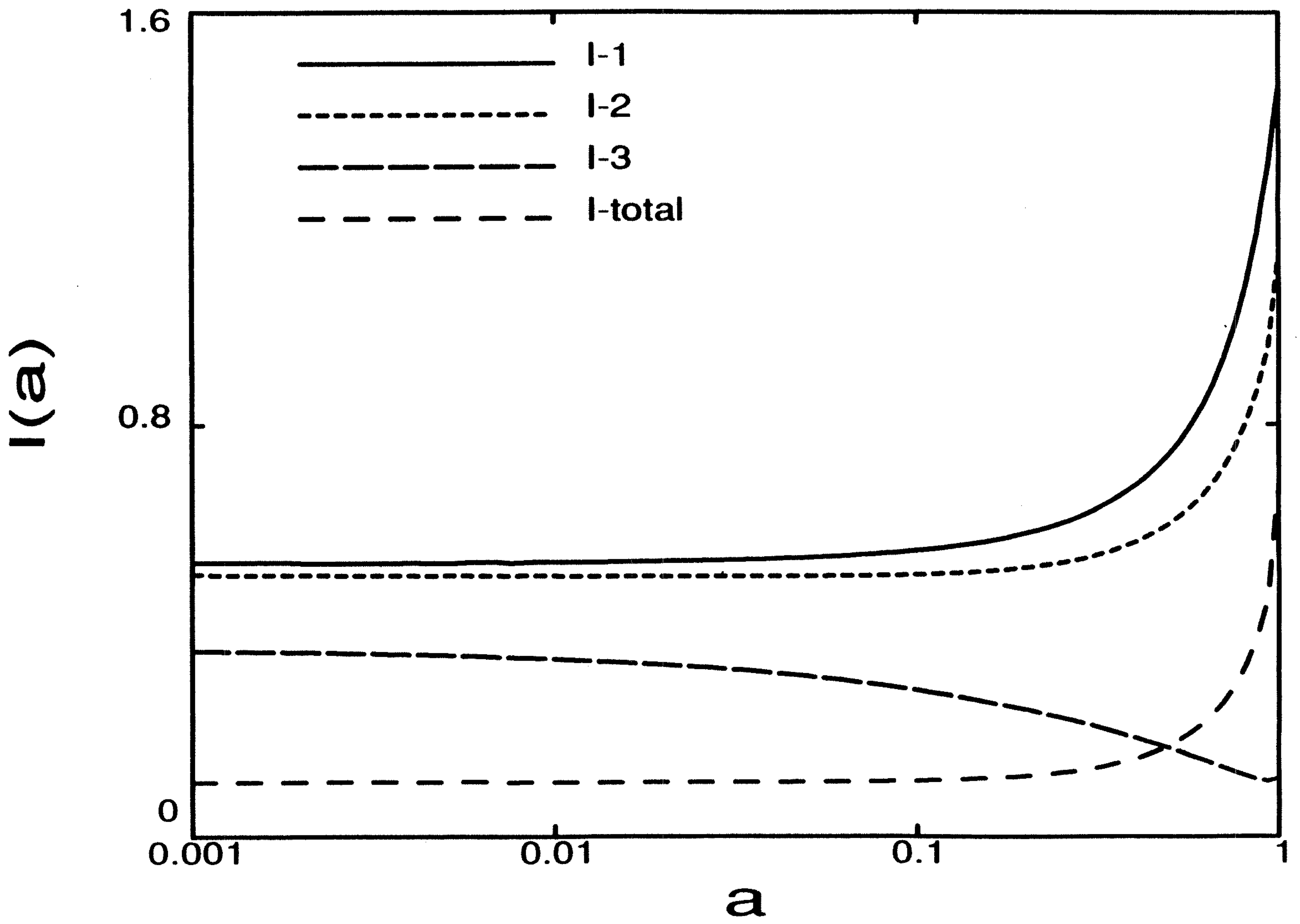


Figure 7

Chapter 5

Conclusions

5.1 Summary

Aerosol data inversion is an ill-posed problem, and thus solution techniques must address the lack of a unique solution, increasing instability as the number of data increase, and the lack of a solution that agrees with the data. The issues can be effectively addressed with the regularization algorithm described in Chapter 2; an optimal solution is identified in the feasible set that is both faithful to the data and stable with respect to fluctuations in the data.

The method of discrepancy that was used in Chapter 2 to choose the regularization parameter, can be improved upon in most cases. Two techniques that generally choose better regularization parameters are (1) minimizing an unbiased estimate of the inverted errors if the magnitude of the errors is known, and (2) generalized cross validation if many data are available. If inequality constraints are important to the solution, then the calculations presented in [11] must be modified since changes in the regularization parameter can affect the set of active constraints and alter the linear approximation.

If there are limited data, then the data define a large solution set, and one can expect a sizable difference between the true and inverted distribution. Here, in addition to an optimal solution, one should also know the variance in the solution. A simple approach to this problem is to view the size distribution as a sample of a random process with a specified autocorrelation function. From this, one can define the amount of information provided by a set of measurements. One can show, as in Chapter 4, that dependent errors can place an upper bound on the amount of information provided by an instrument, and that filters can improve the ability of an instrument to determine a linear functional by eliminating components of the distribution that are orthogonal to the linear functional.

5.2 Project impact on the aerosol community

The results of this research should have a significant impact on the aerosol community. First, when compared to previously developed inversion algorithms, the algorithms developed in this thesis provide better resolution of the size distribution and are successful under a much larger set of conditions. Second, the similarity of aerosol inversion problems, irrespective of the instrument response functions, has been emphasized along with a solution that is independent of the particular instrument. The benefit here is an elimination of the tide of proposed ad-hoc inversion techniques that are being developed for specific instruments under specific conditions. Finally, this research has demonstrated the importance as well as the ease of including all of the available information into the inversion process. This includes information on errors in the data, information from multiple instruments, and information provided by inequality constraints.

5.3 Recommendations for future research

5.3.1 Justification for additional work

There are several exciting possibilities for future research on the inversion of aerosol size distribution data. One must first answer the following: Is further research in this direction justified, especially given that improved instrument design will probably have larger impact on this particular inversion problem? The answer, in my opinion, is an unconditional yes:

1. This research has opened and can continue to open new opportunities in experimental development.

2. The inversion problem will always exist, especially in relation to measuring properties of sub-micron aerosol samples, even after the ultimate particle sizing instrument has been developed.

3. A common problem in all fields of science is data interpretation, and this often leads to ill-posed problems. The techniques learned in aerosol data inversion are directly applicable to many ill-posed problems, and this can open many doors for the young researcher.

5.3.2 Measuring $f(x,t)$

The goal of the first project is to extend the work in this thesis to the determination of the size distribution as a function of time, $f(x,t)$. This distribution is the desired result of many experimental studies. Part of the motivation for this project is the realization that the size distribution at time t provides information about the size distribution at other times. One can make better use of information provided by the data if these related sub-inversions are treated as a single inversion problem. This expected result could readily be confirmed by calculating the information provided by realistic cross-correlation functions of the random process that generates $f(x,t)$.

The extension of regularization to higher dimensions is straightforward [45]. One detail that needs to be resolved is how to combine derivatives with respect to time and size to form a regularization functional. The resolution of this detail may lie in the use of multiple regularization parameters [22]. Also, one can expect to encounter some numerical difficulties because of the size of the inversion problem.

This project of determining $f(x,t)$ leads nicely to the problem of choosing a best set of measurements. A choice of measurements is often necessary because one can only make a single measurement in the time interval Δt. The choice of subsequent measurements depends on how $f(x,t)$ is evolving, which must be inferred from previous measurements, and on the linear functionals available to apply to $f(x,t)$. The final result of this project would be, for example, a SEMS [46] that optimally altered the scan rate as a function of voltage and time to obtain the best resolution of an evolving size distribution. I feel this project has scientific as well as commercial value.

5.3.3 Extrapolation to $f_{\lambda,\infty}$

The second research project attempts to answer the following: If $4n$ data are available for a single size distribution, then is it possible to use the regularized solutions $f_{\lambda,n}$, $f_{\lambda,2n}$, and $f_{\lambda,4n}$ to estimate $f_{\lambda,\infty}$? Here $f_{\lambda,i}$ is the regularized solution generated with a subset of i data. This question has current application, for example, in the case of epiphaniometer [17], where potentially large data sets can be generated. In the similar problem of numerical integration, extrapolation techniques are commonly exploited. The potential benefits are

- improved accuracy for a given computational effort.
- an estimate of the error in the regularized solution.

Appendix A

MICRON User's Guide and Reference

A.1 Introduction

Purpose

MICRON (**M**ulti-**I**nstrument inversion using **C**onstrained **R**egularizati**ON**) is a batch FORTRAN 77 program that reconstructs an aerosol size distribution from conventional aerosol size distribution data. MICRON will use data from any combination of instruments if the user supplies FORTRAN subroutines that model the instruments' response.

Installation

MICRON has been successfully tested on VAX and SUN workstations. The source code is provided on a diskette and divided into 5 files:

mainmc.f	main program and primary subroutines
ioutmc.f	input and output subroutines
utilmc.f	basic math and string handling utilities
qpslmc.f	quadratic program solver
usrfmc.f	user supplied instrument subroutines

The user must transfer these files from a PC to a larger computer (e.g., VAX or SUN) and then compile and link the files to generate an executable file.

Notation

Throughout this manual we will use the following notation:

- placeholders for input that should be provided by the user are printed in *italics*.

- literal input and output is printed in `typewriter type`. Also, FORTRAN variables and source code are printed in `typewriter type`.
- file names are printed in this type.

A.2 An example

One should take the following steps to invert data with MICRON:

Step A write subroutines that will calculate the instruments' response and link these with MICRON.

Step B write a data file, micron.dat, that contains a description of the instruments and MICRON's operating parameters

Step C check the instrument response functions.

Step D test MICRON on artificial data.

Step 1 prepare the files that will contain the measured data.

Step 2 write a data file, micron.jbs, that contains the list of jobs that MICRON will execute.

Steps A-D need to be performed only when MICRON is first set up for a specific set of instruments. All of these steps are briefly illustrated in the following example.

We assume that a differential mobility analyzer has generated four sets of data. Each set of data contains 50 measurements that correspond to 50 voltage settings. The same voltages were used to generate each set of data and are contained in the file volts.dat.

Step A

First we must provide a subroutine that calculates the instrument's response to a monodispersed aerosol source of unit concentration at a given diameter and channel (voltage). The instrument response is determined by the experimentalist through calibrations and/or theoretical models. The units of the data, size distribution, and instrument response function must be consistent. The following section of code gives an idea of how to start:

```
      double precision function INST1(dp)
************************************************************
* INST1 calculates the differential mobility analyzer
* (DMA) response described in * Aerosol Sci. and Tech.,
* 2:465-475.
*
* Description of variables:
* dp     - particle diameter in microns
* iinst  - instrument's index = 1 for INST1
* iicha  - the channel (voltage) of interest
* ncha   - the number of channels for this instrument
*          (= 50)
* ndma   - unit number used to open data file of voltages
* temp   - air temperature
* volts  - the array of 50 channel voltages
* FRAC   - a function that calculates the fraction of
*          particles detected by the DMA
* FUNIT  - finds a valid unit number
* SENS   - a function that calculates the response of
*          the detector
************************************************************
      integer           i,     iinst, iicha, ncha
      double precision  dp,    FRAC,  par,    SENS,    temp,
     &                  volt,  volts(50)
      logical           first
      parameter        (minst = 10,    mpar = 6)
      common /cmpar/    par(minst,mpar)
      common /cmkerf/   iinst,  iicha,  ncha
      save   /cmpar/
      save   /cmkerf/
      save              first
      data              first /.true./
*-----------------------------------------------------------
      if (first) then
        first = .false.
        call FUNIT(ndma)
        open(ndma, file = 'volts.dat', status = 'old')
        read(ndma,*) (volts(i), i = 1, ncha)
        close(ndma)
      end if
      temp  = par(iinst,1)
      volt  = volts(iicha)
      INST1 = FRAC(volt, temp, dp) * SENS(dp, temp)
      return
      end
```

Note the following:

1. the functions must be named INST1, INST2, ...,

2. the channel number iicha, must be passed to the instrument subroutine as shown. INST10.

3. parameters, temperature in this case, can be passed to the instrument functions. The parameters are stored in par(iinst,*) and are accessed as shown.

4. use FUNIT to find a legal unit number if files are opened.

More examples of instrument functions are given in Chapters A.4, Chapters A.6, and the source code in the file usrfmc.f.

Step B

Next we must provide a data file, micron.dat, that lists the instruments and any operating instructions. The following micron.dat corresponds to the DMA listed above, and informs MICRON that INST1 has 50 channels and will be referenced by the word DMA.

```
# MICRON ignores lines beginning with '#'
# the case of the letters is unimportant
# sample micron.dat
#
Instrument 1 = DMA, number of channels = 50
temp = 298.0

#Have MICRON compute the least squares solutions
Least squares solution: yes
```

The first line informs MICRON that INST1 has 50 channels and will be referenced by the word DMA. The second states that MICRON has a parameter, TEMP, that has a default value of 298.0.

There are several options that can be specified in this file, and additional information is given in Chapter A.3.

Step C

The instrument response subroutine should be checked before data are inverted. A data file of DMA response values for channel 5 (for example) can be generated by writing the following line in the job file, micron.jbs:

```
PLOT Channel 5
```

After executing MICRON, the data file dma5.plt should be checked to insure that the subroutine INST1 is working.

Step D

A data file that contains artificial data can be generated and inverted by placing the following lines in micron.jbs and executing MICRON:

```
  simulate sample
  invert   sample
```

The instructions that MICRON needs to make sample.inp are in the file sample.sim that is created by the user:

```
#
# sample.sim
#
# add 5% random error to the data
add error
error parameters = 0.0 0.05

# simulate data for the dma
instrument       = dma

# assume the true distribution is lognormal
test function    = log_normal
integral = 100.0
geo_sdev = 1.5
geo_mean = 0.1
```

Any convenient root name can be used in place of sample. More information on simulating data is in Chapter A.3.

Step 1

The user must write formatted data files that contain the measurements for each distribution. The data in each file corresponds to a single distribution. For example: dataset1.inp

```
instrument = dma
# list the data for channel 1, 2, ...
# one datum per line, the second number is the standard deviation.
# negative data are ignored
1.02  0.05
1.11  0.06
  etc.
```

The user can specify several options in the *root*.inp file in addition to the measurements, and these are described in Chapter A.3.

Step 2

The user needs to put a list of jobs and commands in **micron.jbs** and then execute MICRON. The following file requests all of the jobs mentioned in this chapter:

```
#
# sample micron.jbs
#
instrument = dma
plot         channel 5
simulate     sample
invert       sample
invert       dataset1
invert       dataset2
invert       dataset3
```

The output that corresponds to **dataset1.inp** is placed in the following files when MICRON is executed if the `message` levels are large enough (see Chapter A.3) :

dataset1.log: the warnings and intermediate results.

dataset1.out: the solution vector and solution properties.

dataset1.plt: the solution vector.

dataset1.ech: the echoed input data.

A list of valid commands for **micron.jbs** is listed in Chapter A.3.

An additional example is described in detail in Chapter A.4, and output from the example is provided. The example problem in Chapter A.4 should be solved to insure MICRON is working properly.

A.3 MICRON's data files

This chapter describes the syntax of each of the data files that the user needs to write for MICRON. Each data file has a list of valid keywords. MICRON expects (in most cases) to see entries of the form

keyword = parameters

The parameters are usually on the same line as the keywords. We will write

`nothing` $= i_1, c_1, c_2, r_1$

to mean that the keyword `nothing` is followed by one integer, two character variables, and one real number. The parameters should follow the keyword in the order specified. MICRON usually ignores words following the keyword if they are not expected or recognized. Brackets, [ ], are placed around optional parameters.

When writing data files for MICRON remember:

- lines beginning with # are ignored
- the case of the letters used in keywords is not important
- some keywords have aliases
- keywords may be abbreviated (truncated) as long as the truncated portion uniquely defines the keyword. The underlined portion of each keyword listed below represents the minimum truncation.

Examples of all the data files mentioned in this chapter are given in Chapter A.4.

micron.dat

micron.dat is read when MICRON begins executing and contains a list of the instrument names, parameter names, default parameter values, and special operating instructions.

Keywords and syntax:

```
constrain  instrument  largest  machine
maximum    message     norm     smoothing
tolerance  weak
```

`constrain` the sum of the errors: c_1
where c_1 is either **yes** or **no**. If c_1 is **yes**, then MICRON constrains the sum of the errors to be nonnegative.
(Default = `no`)

`instrument` $i_1 = c_1$, number of channels $= i_2$
where i_1 is the instrument index, and c_1 is the name the user wants to use to refer to the instrument, and i_2 is the number of data that the instrument provides. i_1 must immediately follow `instrument`, and c_1 must immediately follow i_1. On the lines that follow the keyword `Instrument`, the parameter names and their default values must be listed. *Do not* choose a parameter name that can be confused with a keyword. For example, the entry

```
instrument 4 = impactor, 8 channels
temperature   = 298.0    +/- 10.0
```

informs MICRON that the subroutine `INST4` will be referred to by `impactor`, and that it has 8 channels. Also the `impactor` has a single parameter referred to by `temperature`; `temperature` has a default value of 298. The second numeric value after `temperature` is optional and informs MICRON that the data base micron.kri, should be updated whenever the parameter value is reset outside the range (288,308). The default value of the second parameter is two percent of the first parameter.

`largest` inversion interval $= r_1, r_2$
where r_1 and r_2 are the endpoints of the largest inversion interval (d_{min}, d_{max}) that MICRON needs to consider. Larger intervals can require more computer time. The contribution to the instrument readings must be negligible for particles outside the

largest inversion interval.
(Default = 0.001, 11.0 (microns))

`machine` tolerances = r_1, r_2, r_3, r_4
allows the user to specify the following constants associated with the computer being used:

$r_1 =$ the machine floating point precision (e.g., 1.d-15)

$r_2 =$ the minimum positive number of the machine (e.g., 1.0d-38).

$r_3 =$ the largest number on the machine (e.g., 1.0d+38).

$r_4 =$ the default output unit number. (e.g., 6)

(Default: MICRON computes these tolerances if they are not set in **micron.dat**. These calculations will cause underflows on some machines.)

`maximum` failures = i_1
resets the number of successive inversion failures that will cause MICRON to quit. (Default = 5)

`message` level for $c_1 = i_1$
where c_1 specifies an output data file and i_1 is an integer that helps MICRON decide how much output to place in the output data files. c_1 must be one of the following:

`.ech` the amount of input data that is echoed.

`.log` the amount of information that is included with the calculation log.

`.out` the amount of information provided with the solution.

`.plt` the amount of information sent to the plot file.

`.scr` the amount of output that is written to the screen.

The integer parameter is interpreted as follows:

`.ech` $\leq$ 00	the **.ech** file is not opened.
$\geq$ 10	the **.inp** file is echoed if a reading failure occurs.
$\geq$ 20	same as 10 but the **micron.dat** file is echoed.
$\geq$ 30	same as 20 but the **.inp** file is always echoed.

(Default = 20)

`.log` ≤ 00 the **.log** file is not opened.

≥ 20 completion status of each inversion is written along with any errors to the **.log** file.

≥ 30 same as 20 with any warning messages.

≥ 40 same as 30 with the results of calculating the solution dimension, minimum error in the transformed data, the target of the norm squared recovered error.

≥ 50 same as 40 with the recovered error discrepancy, solution roughness, and quadratic functional for each smoothing parameter.

≥ 60 same as 50 with the matrix representation of the kernel functions.

≥ 70 same as 60 with the inverse of the error array.

(Default = 40)

`.out` ≤ 00 the **.out** file is not opened.

≥ 10 header plus solution vector is written to the **.out** file.

≥ 20 same as 10 with some solution properties (total number, variance, mean diameter, roughness, discrepancy, and value of the functional minimized).

≥ 30 same as 20 with an echo of the input instrument reading, bounds on the inverted data, the inverted readings, and percent differences.

≥ 40 same as 30 with corresponding solution diameter, and dV/dlog(diam).

≥ 50 same as 40 with the computation level, dimension of solution vector, inversion interval.

(Default = 100)

`.plt`; this tells MICRON how much output to generate to the **.plt** file. The following rules are used:

.plt ≤ 0 the .plt file is not opened.

≥ 10 the solution vector is written to the .plt file. The solution is reported in the interval defined by the variances of the kernel functions.

≥ 20 same as 10 but including the diameters to which the solution vector corresponds.

≥ 30 same as 20 but including $dV/dln(diam)$, where V is aerosol volume.

(Default = 0)

.scr ≤ 0 no output.

≥ 10 report any program terminating errors.

≥ 20 same as 10 plus the success/failure of each inversion or other task.

≥ 30 same as 20 plus the intermediate progress of each inversion.

≥ 40 same as 30 plus the solution statistics for each smoothing parameter.

(Default = 30)

`norm` of penalty term = i_1
Here $i_1 = 1, 2, 3, 4,$ or 5 and represents the derivative in the term $\int(f^{(i_1)}(x))^2\,dx$ used to define "smooth" in MICRON's solution.
(Default = 3)

`smoothing` chosen by c_1
defines the algorithm that MICRON uses to choose the smoothing parameter. c_1 is one of the following:

CGCV same as UGCV except the nonnegativity constraints are taken into account. Recommended over UGCV when constraints are an important part of the solution. MICRON may have difficulty choosing the initial smoothing parameter when CGCV is used, and the user may need to specify an initial smoothing parameter in *root*.inp.

`CRR` Same as `URR` except that the nonnegativity constraints are taken into account. May work better than `URR` for sharp distributions.

`DISC` MICRON will choose the smoothing parameter so that the sum of the squares of the inverted errors is equal to the expected value of the sum of the squares of the errors in the measured data. This method is not recommended because the solution can become unrealistic if the standard deviations of the errors in the data are not accurately estimated.

`UGCV` MICRON will use generalized cross validation without taking the constraints into consideration. This method is recommended whenever there is a large number of data relative to the amount of structure in the distribution.

`URR` MICRON will find the smoothing parameter that minimizes an unbiased estimate of the inverted error. Recommended if there are only a few data (≈ 15) , and the amount of error in the data can be accurately estimated.

(Default = `cgcv` if the number of data is greater than 15 and `crr` otherwise.)

`tolerance` for choosing the smoothing parameter = r_1
where r_1 represents $\log(\lambda_{max}/\lambda_{min})$. MICRON will make sure the optimal smoothing parameter lies inside the interval (λ_{min}, λ_{max}), and find a smoothing parameter that lies inside this interval.
(Default = computed for the method of discrepancy, 0.25 otherwise.)

`weak` solutions are acceptable: c_1
where c_1 is `yes` or `no`. Small smoothing parameters may cause the minimization routine that finds the size distribution to find weak minimums. Sometimes this will cause the solution to include unrealistic bumps.
(Default = `yes`)

micron.jbs

micron.jbs contains a list of jobs and instructions for MICRON and is read while MICRON is executing.

Keywords and syntax:

```
calculate  copy        create     drop
echo       instrument  invert     least
make       message     plot       points
recover    simulate    smoothing  tolerance
update     weak
```

`calculate` channel i_1
instructs MICRON to create a data file that contains the instrument response for channel i_1. The output file name is a combination of the instrument name and channel number. If i_1 is not a valid channel number, then the instrument response for all of the channels will be computed for each diameter. The `instrument` and `number` commands will often be used prior to the `calculate` command.

`copy`
MICRON will make a formatted copy of the data base micron.kri. The formatted copy is named micron.cpy.

`create`
instructs MICRON to create micron.kri. This command is not normally used since MICRON will automatically create micron.kri if needed.

`drop` c_1
instructs MICRON to create an unformatted data file named c_1. The data file will contain the instrument response matrices that are stored in MICRON. This option is useful if the instrument response matrix is expensive to calculate.

`echo`
alias for `copy`.

`instrument` c_1
where c_1 is an instrument name in micron.dat. Following `instrument`, the instrument parameters may be reset.

`invert` c_1
Instructs MICRON to invert the data contained in the file with root name c_1 and suffix .inp.

`least` squares solution : c_1
where c_1 is either `yes` or `no`. If `yes`, then MICRON will compute the least-squares solution.

`make`
alias for `create`.

`message` level $c_1 = i_1$
same as the `message` command in micron.dat.

`number` of points $= i_1$
resets the number of function evaluations made by MICRON in the `calculate` command.

`plot`
alias for `calculate`.

`recover` c_1
instructs MICRON to recover the unformatted file named c_1 that has been previously dropped by MICRON.

`simulate` c_1
Instructs MICRON to create a data file with root name c_1 and suffix .inp that contains simulated data. The instructions for simulating the data are contained in a file with root name c_1 and suffix .sim.

`smoothing` parameter chosen by c_1
same as the `smoothing` command in micron.dat.

tolerance
same as the tolerance command in micron.dat.

update c_1
where c_1 is the name of an instrument listed in micron.dat. This command instructs MICRON to recalculate the information in micron.kri for the instrument c_1.

weak solution: c_1
Same as the **weak** command in micron.dat.

micron.sim

MICRON reads micron.sim once per run to find out the test function names and parameters that are needed to simulate data.

Keywords and syntax:

funtion test

function $i_1 = c_1$
where i_1 is the test function's index, and c_1 is the name the user wants to use to refer to the function. c_1 must immediately follow i_1, and i_1 must immediately follow function. On the lines that follow the keyword function, the parameter names and their default values must be listed. *Do not* choose a parameter name that can be confused with a keyword. For example, the entry

```
function 1 = log_normal
integral = 100.0
geo_mean = 1.0
geo_sdev = 1.8
```

informs MICRON that the subroutine TF1 will be referred to by LOG_NORM, and that the LOG_NORM has three parameters.

test
alias for function.

root.inp

root.inp contains the instrument readings that correspond to a single distribution. Special inversion instructions may also be placed in this file. MICRON can be instructed to read the *root*.inp file (for example sample.inp) by placing the following entry in micron.jbs:

```
invert sample
```

Keywords and syntax:

```
concentration  dimension  endpoint    ends
error          header     instrument  interval
intvl          inversion  iterations  itns
ndim           skip       smoothing   smp
solution       title
```

`concentration` constraints = r_1, r_2
constrains the integral of the size distribution to lie between r_1 and r_2. A negative value is interpreted as no constraint. For example the entry

```
conc = -1.0  1.0
```

constrains the integral of size distribution to be less than 1. This option should be used cautiously.
(Default = no constraints)

`dimension` of the solution vector = i_1
defines the number of linear splines that are used to represent the solution vector. Larger values of i_1 require more computing time. i_1 must be less than MICRON's internal parameter `mdim = 151`.
(Default: dimension is computed)

`endpoint` constraints = c_1
instructs MICRON to constrain the ends to equal zero. c_1 must be one of the following:

both - both ends are constrained.

lower - the lower endpoint is constrained.

neither - neither end is constrained.

none - same as **neither**.

`upper` - the upper endpoint is constrained.

(Default = `neither`)

`ends`
alias for `endpoint`.

`error` description = c_1
instructs MICRON to begin reading in the elements of the error array. Off-diagonal entries in the error array can be used to describe dependent errors in the data. The magnitude of the entry in the i^{th} row and j^{th} column represents the standard deviation of the j^{th} error source. See [47] for more details. c_1 must be one of the following:

`dense`: MICRON will attempt to read the error array row by row. For example:

```
error array is dense
row 3
0.0 0.1 1.0
row 1
1.0 0.2 0.0
```

MICRON will stop reading a row of the error array when the first entry is non-numeric. MICRON will stop reading in a dense error array when the first word is not `row`.

`diagonal`: MICRON will assume the errors are independent and that the diagonal elements have already been entered; MICRON takes no action when `error = diagonal`.

`sparse`: MICRON assumes that the lines that immediately follow this option will contain the row index, column index, and the off-diagonal element. For example, the previous `dense` error array could have been described by

```
error array is sparse
3 2 0.1
3 3 1.0
1 1 1.0
1 2 0.2
```

Combinations of the `dense` and `sparse` option may be used.
(Default = `diagonal`)

`header` $= c_1$
where c_1 is a line of text that will be placed at the top of the output file. Two lines of text may be specified with two `header` commands. If `blank` is the first word of the header, then the line of text is erased.
(Default = `blank`)

`instrument` $= c_1$
where c_1 is one of the instrument names listed in micron.dat. This informs MICRON that data will follow for the instrument c_1. Parameters that correspond to the instrument may be reset on the lines that immediately follow the keyword `instrument`. Following the parameters, the data for *all* the channels must be entered. Each line can contain only 1 datum, and the format is

y(i) sigma(i) [bl(i)] [bu(i)]

where *y(i)* is the datum for the i^{th} channel, *sigma(i)* is the standard deviation, and *bl(i)* and *bu(i)* are the lower and upper bounds on the inverted data. If the user wants MICRON to ignore the datum from a channel, then the word `skip` or a negative number may be entered in place of the datum.

`interval`
alias for `inversion`

`intvl`
alias for `inversion`

`inversion` interval $= r_1, r_2$
informs MICRON that the solution distribution needs to be computed for diameters lying between r_1 and r_2; the size distribution must not contribute to the data outside this interval. Larger intervals require more computer time.
(Default = the smallest interval such that the kernel functions are zero outside the interval.)

`iterations`
alias for `smoothing iterations`

`itns`
alias for `smoothing iterations`.

ndim
alias for dimension.

smoothing iterations $= i_1$
defines the maximum number of iterations that MICRON will use to find the optimal smoothing parameter. Note $i_1 = 0$ can be used to find the least squares solution, and $i_1 = 1$ can be used to specify the final smoothing parameter.
(Default = MICRON's internal parameter: miter = 20)

smoothing parameter $= r_1$
MICRON attempts to minimize a function of the smoothing parameter, and sometimes MICRON's initial smoothing parameter guess needs to be chosen by the user. This option should not be used when the smoothing parameter is chosen by URR and UGCV; (see section micron.dat).
(Default = computed)

smp
alias for smoothing parameter.

solution
alias for dimension.

title
alias for header.

root.sim

root.sim contains instructions that MICRON needs to simulate data for the file *root*.inp. MICRON can be instructed to read the *root*.sim file (for example sample.sim) and create sample.inp by placing the following entry in micron.jbs:

```
simulate sample
```

Keywords and syntax:

```
add_error  data    function  instrument
name       number  seed      test
```

`add_error` : c_1
where c_1 is **yes** or `no`. If c_1 is yes, then MICRON will add normal random error to the data. The random error is generated by a random number generator. The standard deviation is set by the `error` command and is reported to the *root*.inp file.
(Default = `no`)

`function` $= c_1$
alias for `test`.

`instrument` $= c_1$
where c_1 is the name of an instrument listed in micron.dat. This instructs MICRON to simulate data for the instrument indicated. The entries that follow `instrument` may be used to reset the parameters (e.g., `temperature`) that correspond to the specified instrument. For example:

```
instrument  = dma
temperature = 298.0 K
```

`name` = of the test distribution is c_1
This entry resets the name of the test distribution. c_1 must be the last word on the line.
(Default = `test_dis`)

`number` of function evaluations $= i_1$
MICRON will create a file of test distribution values, and i_1 is the number of function values that are reported. The function values are logarithmically spaced between the d_{max} and d_{min} specified in micron.dat, and the test distribution values are placed in a file named *name*.plt, where *name* is the name of the test distribution.
(Default = 100)

`seed` $= i_1$
re-seeds the random number generator that is used to generate random error.

`test` function $= c_1$
where c_1 is the name of one of the test functions listed in micron.sim. This entry instructs MICRON to add the specified test function evaluation to the test distribution. The entries that follow `test` may be used to reset the test function parameters.

For example, if log_norm corresponds to a log-normal distribution, then a tri-modal distribution with three modes of equal size could be specified as follows:

```
name = tri_modal
test function = log_normal
integral      = 100.0
geo_mean      = 0.03
geo_sdev      = 1.8

test function = log_normal
geo_mean      = 0.06

test function = log_normal
geo_mean      = 0.12
```

A.4 Another example

Below we describe the input and output of an example in detail. This example should be used as a test when MICRON is first installed.

We assume that two impactors have provided us size distribution data. The two impactors have slightly different responses, and they are used side by side to measure a size distribution. Calibrated collection efficiency data is available for each stage of the impactors, and the efficiency data is in the data files impcta.dat, and impctb.dat. The first column of these files is the particle diameter in microns at which the efficiency is calibrated, and the subsequent columns are the efficiency of stage 1, stage 2, etc. The file impcta.dat is shown here, and impctb.dat is on the diskette with the source code.

impcta.dat:

```
0.010  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
0.011  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.03
0.013  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.07
0.014  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.14
0.016  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.25
0.018  0.00  0.00  0.00  0.00  0.00  0.00  0.01  0.39
0.021  0.00  0.00  0.00  0.00  0.00  0.00  0.01  0.55
0.024  0.00  0.00  0.00  0.00  0.00  0.00  0.04  0.71
0.027  0.00  0.00  0.00  0.00  0.00  0.00  0.09  0.83
0.030  0.00  0.00  0.00  0.00  0.00  0.00  0.17  0.91
0.034  0.00  0.00  0.00  0.00  0.00  0.00  0.29  0.96
0.038  0.00  0.00  0.00  0.00  0.00  0.00  0.45  0.99
0.043  0.00  0.00  0.00  0.00  0.00  0.01  0.61  0.99
0.049  0.00  0.00  0.00  0.00  0.00  0.02  0.75  1.00
0.055  0.00  0.00  0.00  0.00  0.00  0.07  0.86  1.00
0.063  0.00  0.00  0.00  0.00  0.00  0.16  0.93  1.00
0.071  0.00  0.00  0.00  0.00  0.00  0.31  0.97  1.00
0.080  0.00  0.00  0.00  0.00  0.00  0.50  0.99  1.00
0.090  0.00  0.00  0.00  0.00  0.01  0.69  1.00  1.00
0.102  0.00  0.00  0.00  0.00  0.04  0.84  1.00  1.00
```

```
0.115  0.00  0.00  0.00  0.00  0.10  0.93  1.00  1.00
0.130  0.00  0.00  0.00  0.00  0.21  0.97  1.00  1.00
0.147  0.00  0.00  0.00  0.00  0.37  0.99  1.00  1.00
0.166  0.00  0.00  0.00  0.00  0.56  1.00  1.00  1.00
0.188  0.00  0.00  0.00  0.02  0.74  1.00  1.00  1.00
0.213  0.00  0.00  0.00  0.05  0.87  1.00  1.00  1.00
0.240  0.00  0.00  0.00  0.13  0.95  1.00  1.00  1.00
0.272  0.00  0.00  0.00  0.26  0.98  1.00  1.00  1.00
0.307  0.00  0.00  0.00  0.43  1.00  1.00  1.00  1.00
0.347  0.00  0.00  0.00  0.63  1.00  1.00  1.00  1.00
0.392  0.00  0.00  0.00  0.79  1.00  1.00  1.00  1.00
0.443  0.00  0.00  0.01  0.90  1.00  1.00  1.00  1.00
0.500  0.00  0.00  0.05  0.96  1.00  1.00  1.00  1.00
0.565  0.00  0.00  0.20  0.99  1.00  1.00  1.00  1.00
0.639  0.00  0.00  0.50  1.00  1.00  1.00  1.00  1.00
0.722  0.00  0.00  0.79  1.00  1.00  1.00  1.00  1.00
0.816  0.00  0.00  0.95  1.00  1.00  1.00  1.00  1.00
0.922  0.00  0.01  0.99  1.00  1.00  1.00  1.00  1.00
1.042  0.00  0.07  1.00  1.00  1.00  1.00  1.00  1.00
1.178  0.00  0.25  1.00  1.00  1.00  1.00  1.00  1.00
1.331  0.00  0.56  1.00  1.00  1.00  1.00  1.00  1.00
1.504  0.00  0.83  1.00  1.00  1.00  1.00  1.00  1.00
1.700  0.01  0.96  1.00  1.00  1.00  1.00  1.00  1.00
1.921  0.04  1.00  1.00  1.00  1.00  1.00  1.00  1.00
2.170  0.17  1.00  1.00  1.00  1.00  1.00  1.00  1.00
2.453  0.45  1.00  1.00  1.00  1.00  1.00  1.00  1.00
2.772  0.75  1.00  1.00  1.00  1.00  1.00  1.00  1.00
3.132  0.93  1.00  1.00  1.00  1.00  1.00  1.00  1.00
3.540  0.99  1.00  1.00  1.00  1.00  1.00  1.00  1.00
4.000  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
```

Step A

The measured data and the size distribution have units of mass/volume. Here, the instrument response subroutine represents the fraction of particles collected.

```
      double precision function INST1(diam)
************************************************************************
*
* INST1 computes the response for an 8 stage impactor.
* Calibrated stage efficiency data is contained in impcta.dat.
* The size distribution and data must the same units.
*
* VARIABLES:
*
* i,j     - counters
* ihigh   - higher index used in the binary search routine
* iinst   - the instrument index for INST1 = 1
* ilow    - lower index used in the binary search routine
* mcha    - the variable used to dimension the calibration data array
* ncha    - the number of LPI channels = mcha = 8
* mdia    - the number of diameters for which calibration data is avail.
* nlpi    - the unit number used to read in the calibration data.
* dia     - the array which holds the particle diameters at each impactor
*          stage for which calibration data exists.
* diam   - the diameter for which the kernel function is to be evaluated.
* transm - for stage i, it is the fraction of particles which are not
*          collected by stages 1 through ichan-1.
* xivec  - the vector containing the calibration data for particle
*          diameter "diam" for channels 1 through "ncha"
*        - the vector containing the kernel function for each stage
*          at diameter diam (upon exit).
* xker   - the matrix used to contain the calibration data for
*          diameters contained in the array "dia".
*
* Description of called subroutines:
*
* FUNIT  - finds a unit number which can be used to connect to the file
*          containing calibration data.
*
* MATINT - interpolates the calibration data contained in the
*          array xker with a binary search routine.
*
* LAST MODIFIED: 9 NOV 1988
************************************************************************
      integer           i,      ihigh,  iinst,  ilow,   j,      jcha,
     &                  mcha,   mdia,   ncha,   nlpi
      double precision  dia,    diam,   transm, xivec,  xker
      logical           first
      parameter        (mcha   = 8)
      parameter        (mdia   = 50)
      common /cmkerf/   iinst,  jcha,   ncha
      save   /cmkerf/
      dimension         dia(mdia), xker(mcha,mdia), xivec(mcha)
      save              xker,   dia,    first
      data              first  /.true./
*-----------------------------------------------------------------------
c
c **READ CALIBRATION DATA
c
      if (first) then
        first = .false.
        call FUNIT(nlpi)
        open(nlpi, file = 'impcta.dat', status = 'old')
        rewind(nlpi)
        read(nlpi, *) (dia(j),(xker(i,j),i=1,mcha),j=1,mdia)
        close(nlpi)
      end if
c
c **INTERPOLATE THE CALIBRATED EFFICIENCY DATA FOR EACH STAGE
c
      do 100 i = 1, jcha
        ilow  = 1
        ihigh = mdia
        call MATINT(i, 1, mcha, 1, mdia, xker, dia, xivec(i), diam)
100   continue
c
c **SINCE THE IMPACTOR STAGES ARE IN SERIES THE FRACTION DETECTED BY
c **STAGE "i" DEPENDS OF THE FRACTION WHICH HAVE NOT BEEN
```

```
c ##COLLECTED BY STAGES 1 THROUGH "i"-1
c
      transm   = 1.d00
      do 400 i = 2, jcha
        transm = (1.d00 - xivec(i - 1)) * transm
400   continue
      INST1 = xivec(jcha) * transm
      return
*
*___end of INST1___*
*
      end
```

The subroutine MATINT is a linear interpolation routine; the source code is contained and described in the diskette with the example instrument response functions. INST2 is similar to INST1 except that the calibration data is in the file impctb.dat.

Step B

Next we need to write micron.dat. The two impactors are named impcta and impctb.

```
# Sample micron.dat
Instrument 1 = impcta, number of channels = 8
Instrument 2 = impctb, number of channels = 8
message level .plt = 20
```

No parameters are passed to these instrument subroutines.

Step C

Files can be generated that contain the instrument response data for all of the channels. These are generated by writing the following micron.jbs data file and then executing MICRON:

```
points     = 500
instrument = impctA
plot channel 1
plot channel 2
plot channel 3
plot channel 4
plot channel 5
plot channel 6
plot channel 7
plot channel 8
instrument = impctB
plot channel 1
plot channel 2
plot channel 3
```

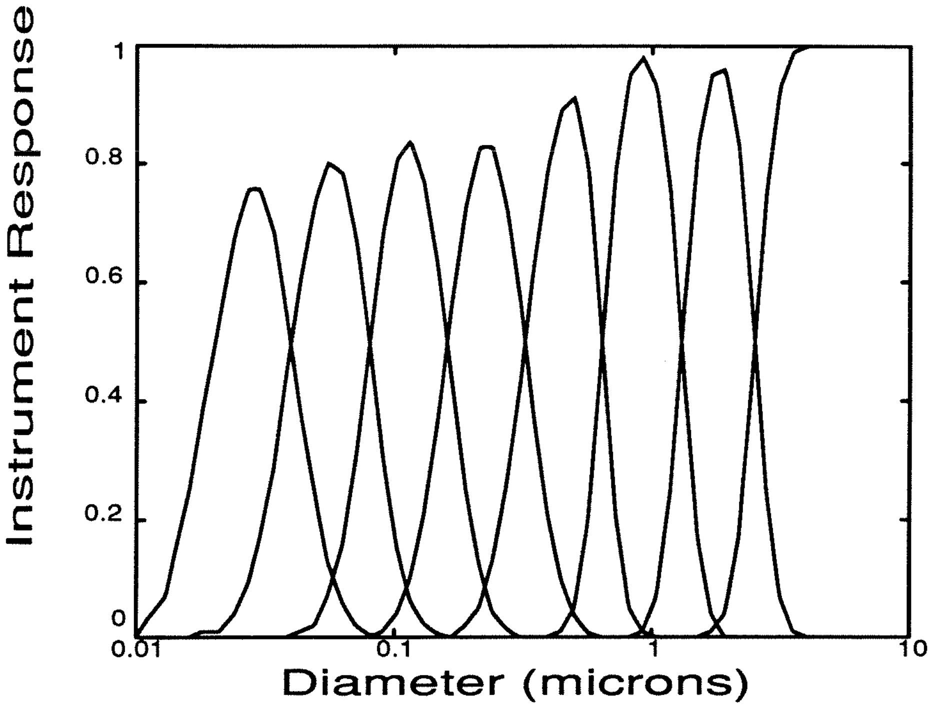


Figure A.1: Impactor A response curves

```
plot channel 4
plot channel 5
plot channel 6
plot channel 7
plot channel 8
```

The x,y data are contained in the files impcta1.plt, impcta2.plt, ..., impcta8.plt. The plots of these data files are shown in Figure A.1.

Step D

Next we should test MICRON on artificial data. We can instruct MICRON to generate three files that contain noisy data by creating the data files ab05.sim, a05.sim, b05.sim, and micron.jbs. ab05.sim:

```
# ab05.sim

# assume 5 percent error is in the data
error parameters = 0.0 0.05

# add the error
```

```
add error
seed = 100

# generate data for both impactors
instrument = impcta
instrument = impctb

# the test distribution is bimodal
test function = log_norm
integral      = 100.0
log_mean      = 1.0
geo_sdev      = 1.5
test function = log_norm
integral      = 20.0
log_mean      = 0.1
geo_sdev      = 1.5

# generate a file that contains the test distribution
name = bimodal
points = 500
```

micron.jbs:

```
simulate a05
simulate b05
simulate ab05
```

a05.sim and b05.sim are the same as ab05.sim except a05.sim does not reference impctb, and b05.sim does not reference impcta. Also, b05.sim should not contain the seed command. MICRON will create three input files that contain noisy data: a05.inp contains data for the impcta, b05.inp contains data for the impctb, and ab05.inp contains data for both impactors. The same bimodal distribution is assumed for all three files, and MICRON places the bimodal function values in the file bimodal.plt. Note that MICRON must have access to micron.sim.

Next we invert the data from all three files by writing the following micron.jbs and executing MICRON:

```
smoothing chosen by crr
message level .plt = 20
invert ab05
invert a05
invert b05
```

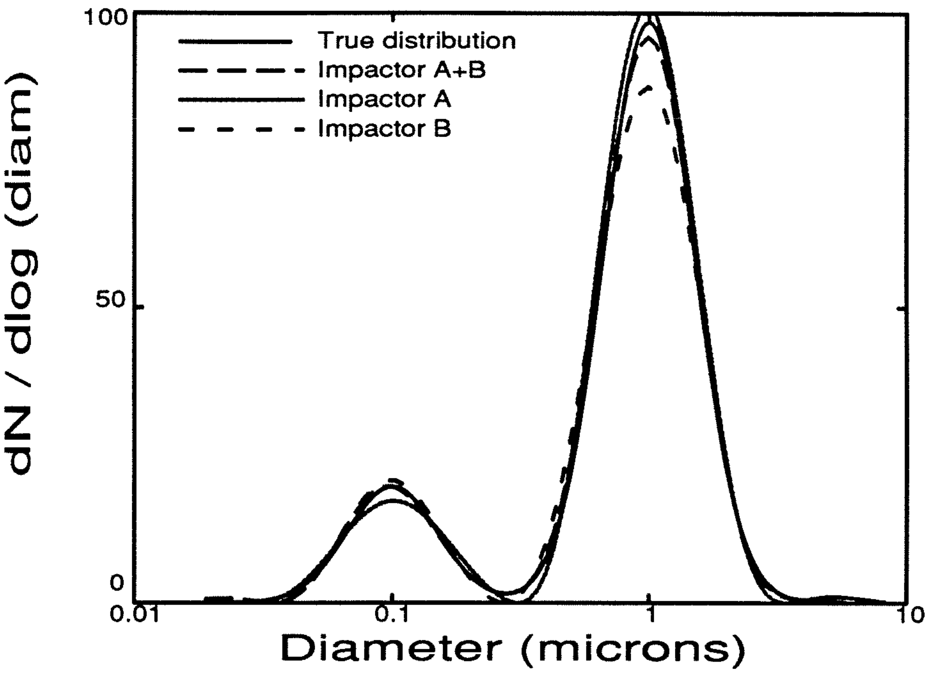


Figure A.2: Size distributions with smoothing defined by CRR.

MICRON will create the data files a05.plt, b05.plt, and ab05.plt. These files contain the inverted distributions and are shown in Figure A.2. Here, the distribution that was obtained by inverting the data from both impactors seems best.

We can also compare the inversion shown in Figure A.2 with the inversions obtained when CGCV is used to choose the smoothing parameter. The user should replace crr in the previous micron.jbs with cgcv and re-execute MICRON. The results are shown in Figure A.3 and demonstrate that CGCV does not work well when only a few data are available.

One also can (and should) test the effects of the following:

- varying the magnitude of the error in the data
- adding dependent error to the data
- specifying an inaccurate error magnitude

The subroutines INPSPC and OUTSPC can simplify these tasks. See the accompanying *MICRON Programmer's Manual.*

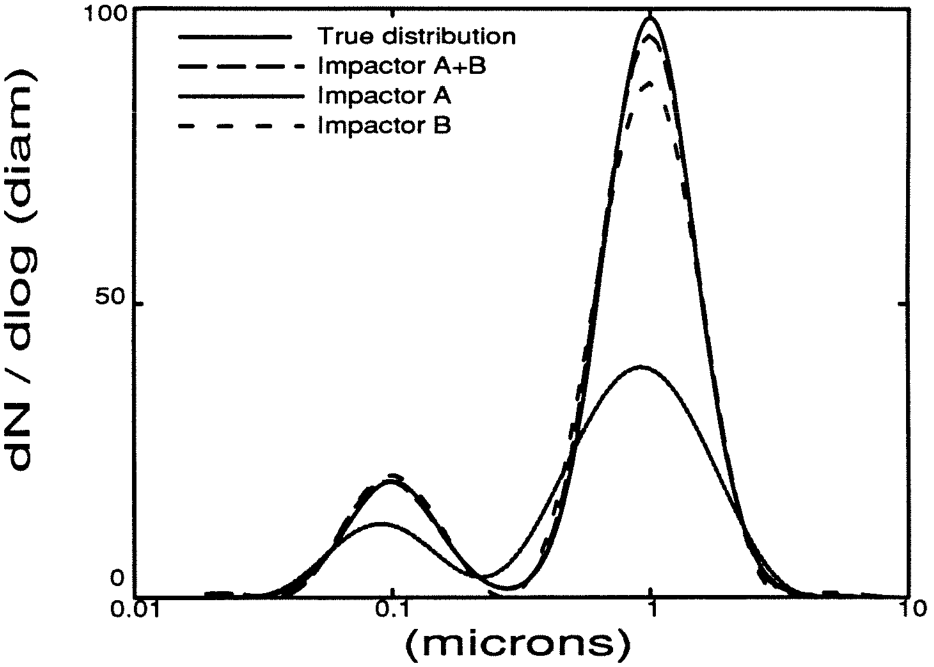


Figure A.3: Size distribution with smoothing defined by CGCV.

Step 1

A data file that contains instrument data will look like the following:
impab1.inp

```
instrument = impcta
2.1  0.3
5.2  0.3
6.3  0.4
11.6 0.5
10.1 0.5
6.0  0.4
2.8  0.3
1.1  0.1

# The first two channels of impctb are not reliable
instrument = impctb
skip
skip
5.5  0.1
8.2  0.1
10.5 0.3
8.1  0.2
3.0  0.1
1.7  0.1
```

```
# The distribution should be zero at the endpoints.
ends = both
```

Step 2

We can invert many sets of data (e.g. impab1.inp) by writing the following micron.jbs and executing MICRON:

```
message level .plt = 20
invert impab1
invert impab2
invert impab3
invert impab4
 etc.
```

A.5 A description of the output

MICRON will create up to 4 output files per inversion in addition to writing to the screen: *root*.ech, *root*.log, *root*.out, and *root*.plt. Additionally, MICRON can also create micron.dmp and micron.cpy while executing. Below we will describe some of the output that is generated by MICRON while inverting ab05.inp. Most of MICRON's output is self-explanatory and is not described below.

ab05.log

```
The minimum norm square of the transformed  error
vector is  0.4466D+01


Smoothing parameter iteration:   1
smoothing parameter                      =  0.1982880D-06
Smoothing choice function (CGCV)         =  0.5314418D+01
roughness of solution                    =  0.1689037D+06
objective function                       =  0.9590015D+00
```

1. The minimum norm square of the transformed error vector is the minimum achievable sum of the squares of the errors when the system of equations is rescaled so that the errors are independent with unit variance. If this number is on the same order of the number of data, then the magnitude of the errors that was specified by the user is probably incorrect.

2. the smoothing choice function is the value of the function that MICRON is trying to minimize as a function of the smoothing parameter. If MICRON is choosing the smoothing parameter poorly, then this number may give some insight to the problem.

3. the roughness of the solution is the integral of the square of the second derivative of the size distribution function.
4. the objective function is the sum of the matching term and the penalty term.

ab05.out

Most of the output in this file is self-explanatory. Only a few comments apply:

1. the size distribution is provided in log base e.
2. the column next to the inverted data titled **`rsdiff.`** represents the difference of the inverted and the measured data divided by the standard deviation.

ab05.plt

This file only contains the solution vector. Only a few comments apply:

1. the size distribution is provided in log base e.
2. the size distribution is provided only over the interval defined by the variance of the kernel functions.
3. the third column is $\pi/6d_p^3$ times the second column, where d_p is the diameter provided in the first column.

A.6 Example instrument response functions

Below are two thoroughly documented example instrument response functions. We assume the reader is familiar with writing FORTRAN subroutines. There are several utility routines in the file usrfmc.f that can ease the task of writing an instrument response function, and the use of these is illustrated below. Additional examples of response functions are in usrfmc.f. The user should attempt to modify an existing response subroutine before writing a new one from scratch.

The first subroutine is for a screen-type diffusion battery. MICRON passes four parameters to INST5.

```
      double precision  function INST5(diam)
************************************************************************
*
* INST5 models the TSI 3040 model screen type diffusion battery
* response described in J. Aerosol Sci. 11:549-556. The response is
* assumed to follow eq. 6 on page 551. It is assumed that a condensation
* nuclei counter is used to detect the particles, and that the channel
* response for stage i is the number of particles detected before
* stage i minus the number of particles detected after stage i.
*
* Description of variables:
* minstt - the declared value of "minst" in MICRON, used to check
*          minst is properly declared
* mpart  - the declared value of "mpar" in MICRON, used to check if
*          mpar is properly declared.
* msgscr - the MICRON's screen message level.
* ndiff  - the instrument index of the condensation nuclei counter (CNC)
* nchat  - the number of channels associated with the CNC
* nscr   - the default output unit number for the screen.
* jchat  - the channel number of the CNC for which a response is desired
* alpha  - solid volume fraction of filter
* cexp   = -log(Penetration) per screen
* df     - fluid flow cross section diameter of diffusion battery in cm
* fibrad - fiber radius in microns
* flow   - sampling flow rate (liters/min)
* frp    - eq. (4) J. Aero. Sci. 11:550
* interc - eq. (5) J. Aero. Sci. 11:550
* pe     - particle Peclet #
* premul - holds the product of FILTER and the CNC response
* rho    - particle density g/cc
* rmin   - the smallest positive number on the computer
* st     - particle Stokes #
* temp   - air temperature in Kelvin
* press  - air pressure in atmosphere
* uo     - undisturbed flow velocity cm/sec
*
* Description of called subroutines:
* AIRVSC - air viscosity in g/cm/s
* CC     - Cunningham slip correction
* DCPART - diffusion coefficient of particle in cm^2/s
* EZKERX - computes the condensation nuclei counter response that
*          is assumed to be computed by INST4.
* PARCHK - used to check if the value of a FORTRAN parameter declared
*          in the instrument subroutine is the same as the value
*          declared in MICRON.
*
* LAST MODIFIED:25 JAN 1988
************************************************************************
      integer          iinst,  jcha,   jchat,  ktscrn, minst,  minstt,
     &                 mpar,   mpart,  msg,    msgscr, ncha,   nchat,
     &                 ndiff,  nscr
      double precision AIRVSC, alpha,  artfil, CC,     cexp,   convcm,
     &                 diam,   DCPART, df,     dfbmax, EZKERX, fibrad,
     &                 FILTER, flow,   frp,    interc, one,    pe,
     &                 par,    pi,     premul, press,  r,      rho,
     &                 rmin,   rp,     st,     temp,   two,    uo,
     &                 xilow,  xiup,   zero
      logical          first
      parameter       (convcm = 1.d+04,       alpha  = 3.45d-01)
      parameter       (df     = 3.81d00,      dfbmax = 1.d+0)
      parameter       (fibrad = 10.d00,       pi     = 3.141592d00)
      parameter       (mpar   = 6,            minst  = 10)
      parameter       (one    = 1.d+0,        zero   = 0.d+0)
      parameter       (two    = 2.d+0)
      common /cmrmin/  rmin
      save   /cmrmin/
      common /cmpar/   par(minst, mpar)
      save   /cmpar/
      common /cmkerf/  iinst,  jcha,   ncha
      save   /cmkerf/
      common /cmscr/   msgscr, nscr
      save   /cmscr/
      common /cmparm/  minstt, mpart
      save   /cmparm/
      save             msg, first
      data msg, first  /0, .true./
      data             ndiff,  nchat,  jchat
     &                /4,      1,      1/
```

```
*---------------------------------------------------------------------
c
c ##check to make sure PAR is dimensioned correctly
c
      if (first) then
        first = .false.
        call PARCHK(minst, minstt, 'minst', 'INST5')
        call PARCHK(mpar,  mpart,  'mpar', 'INST5')
      end if
c
c ##WRITE A WARNING IF THE RESPONSE IS COMPUTED ABOVE THE
c ##CALIBRATION INTERVAL
c
      if (diam .gt. dfbmax) then
        if (msg .eq. 0) then
          if (msgscr .ge. 10) then
            write(nscr, 1) dfbmax
            msg = 1
          end if
        end if
      end if
c
c ##FIRST CHECK IF FILTER OR THE CNC RESPONSE WILL ALLOW A
c ##QUICK EXIT
c
      artfil = FILTER(diam, dfbmax)
      if (artfil .le. rmin) then
        premul = zero
      else
        premul = artfil * EZKERX(ndiff, jchat, nchat, diam)
      end if
      if (premul .le. rmin) then
        INST5 = zero
      else
c
c ##ELSE WE'RE FORCED TO COMPUTE THE RESPONSE
c
        temp   = par(iinst,1)
        flow   = par(iinst,2)
        rho    = par(iinst,3)
        press  = par(iinst,4)
        uo     = 200.d00 / 3.d00 * flow / pi / df**2
        pe     = two * fibrad * uo / DCPART(diam, press, temp) /
     &           convcm
        st     = rho * diam**2 * CC(diam, press, temp) * uo / 18.d00 /
     &           AIRVSC(temp) / fibrad / convcm
        r      = diam / fibrad / two
        rp     = r + one
        frp    = (one / rp) - rp + (two * rp * dlog(rp))
        interc = (29.6d00 - 28.d00*(alpha**0.62)) * r**2 -
     &           (27.5d00 * (r**2.8d00))
        cexp   = 1.96d00*pe**(-two/3.d+0) + 1.69d00*frp +
     &           3.91d00*interc*st +1.94d00/dsqrt(pe) *
     &           r**(two/3.d+0)
        ktscrn = (jcha*(jcha + 1))/2
        xiup   = dexp( - cexp*dble(ktscrn))
        xilow  = dexp( - cexp*dble(ktscrn - jcha))
        INST5  = (xilow - xiup) * premul
      end if
      return
1     format(/,' Warning:',/,
     &' The  instrument  response  for the diffusion  battery  is not',/,
     &' well known above ',d10.3, ' microns.')

*
*___end of INST5___*
*
      end
```

The next subroutine is for a differential mobility analyzer; MICRON passes six parameters to INST8.

```
      double precision  function INST8(diam)
************************************************************************
*
* INST8 returns the theoretical differential mobility analyzer (DMA) re-
* sponse as described in Aerosol Sci. and Tech., 2:465-475.  The
* calculations follow those outlined in AS&T 2:474.  The detector is
* the condensation nuclei counter.
*
* Description of variables:
* mcha   - the number of channels
* minstt - the declared value of "minst" in MICRON, used to check
*          minst is properly declared
* mpart  - the declared value of "mpar" in MICRON, used to check if
*          mpar is properly declared.
* ndiff  - the instrument index of the condensation nuclei counter (CNC)
* nchat  - the number of channels associated with the CNC
* jchat  - the channel number of the CNC for which a response is desired
* diam   - the particle diameter in microns
* e      - electron charge dyne cm/V
* geo    - length/ln(outer radius/inner radius) of the DMA
* par1   - air temperature in Kelvin
* par2   - clean air flow rate in cc/sec
* par3   - main exit flow rate in cc/sec
* par4   - sample aerosol flow rate in cc/sec
* par5   - inlet aerosol flow rate in cc/sec
* par6   - air pressure
* volt   - DMA rod voltage
*
* Description of called subroutines:
* AIRVSC - air viscosity in poise
* CC     - Cunningham slip correction
* CHDST  - computes the fraction of particles of size "diam"
*          carrying "ichrg" charges when exposed to bipolar charging
* DTXF   - computes the function omega in eq. 2 in Aero. Sci. & Tech.
*          2:466.
* EZKERX - computes the fraction of particles detected by the
*          condensation nuclei counter which is assumed to be
*          calculated by INST4.
* PARCHK - used to check if the value of a FORTRAN parameter declared
*          in the instrument subroutine is the same as the value
*          declared in MICRON.
* SRF    - computes sensory response function
*
* LAST MODIFIED:25 JAN 1988
************************************************************************
      integer           icha,   ichrg,  iinst,  jcha,   jchat,  lchrg,
     &                  mcha,   minst,  minstt, mpar,   mpart,  ncha,
     &                  nchat,  nchrg,  ndiff,  ndma
      double precision  AIRVSC, CC,     CHDST,  chdstt, dmatmp, diam,
     &                  DTXF,   dtxft,  e,      geo,    par,    pi,
     &                  pi4,    pkht,   pmf,    press,  rmin,   scale,
     &                  SRF,    srft,   stemp,  temp,   v,      v0,
     &                  v1,     v2,     v3,     v4,     visc,   volt
      logical           first
      parameter        (mcha   = 20,             pi     = 3.14159265d00)
      parameter        (pi4    = 4.d+0 * pi,     e      = 1.602d-12)
      parameter        (geo    = 60.246d00,      mpar   = 6)
      parameter        (minst  = 10)
      common /cmpar/    par(minst, mpar)
      save   /cmpar/
      common /cmkerf/   iinst,  jcha,   ncha
      save   /cmkerf/
      common /cmrmin/   rmin
      save   /cmrmin/
      common /cmparm/   minstt, mpart
      save   /cmparm/
      dimension         volt(mcha)
      save              scale,  stemp,  visc,   volt,   first
      data              first,  stemp,  visc
     &                 /.true., 0.d+0,  0.d+0/
*-----------------------------------------------------------------------
c
c **FIRST READ THE DATA FILE CONTAINING THE DMA VOLTAGE SETTINGS
c
      if (first) then
        first = .false.
        call PARCHK(minst, minstt, 'minst', 'INST8')
        call PARCHK(mpar, mpart, 'mpar', 'INST8')
        call FUNIT(ndma)
        open(ndma, file = 'dma1.dat', status = 'old')
        rewind(ndma)
        read(ndma, *) (volt(icha), icha = 1, ncha)
        close(ndma)
        scale = 1.d+04 * e * geo / 3.d+0 / pi
      end if
c
c
      ndiff =  4
      nchat  = 1
      jchat  = 1
      temp   = par(iinst, 1)
      press  = par(iinst,6)
      v0     = par(iinst, 2) + par(iinst, 3)
      v1     = (v0 - par(iinst, 4) - par(iinst, 5))/pi4
      v2     = (v0 - dabs(par(iinst, 4) - par(iinst, 5)))/pi4
      v3     = (v0 + dabs(par(iinst, 4) - par(iinst, 5)))/pi4
      v4     = (v0 + par(iinst, 4) + par(iinst, 5))/pi4
      pkht   = dmin1(1.d00, (par(iinst, 4)/par(iinst, 5)))
      dmatmp = rmin
      if (stemp .ne. temp) then
        stemp = temp
        visc  = scale / AIRVSC(temp)
      end if
      pmf    = visc * (CC(diam, press, temp) * volt(jcha)) / diam
      call FDCHGI(lchrg, nchrg, diam, pmf, temp, v1, v4)
c
c **THE PARTICLE MAY HAVE 0, 1, 2, ... CHARGES AND WE NEED SUM THE
c **DMA RESPONSE DUE TO THE PARTICLES CARRYING EACH NUMBER OF CHARGES.
c
      v      = dble(lchrg)*pmf
      do 100 ichrg = lchrg, nchrg
        chdstt = CHDST(diam, temp, ichrg)
        dtxft  = DTXF(pkht, v, v1, v2, v3, v4)
        srft   = SRF(ichrg, jchat, nchat, ndiff, par(iinst, 4), diam)
        dmatmp = dmatmp + chdstt * dtxft * srft
        v      = v + pmf
100   continue
      INST8 = dmatmp*par(iinst, 5)
      return
*
*___end of INST8___*
*
      end
```

Note that MICRON has open files, and FUNIT must be called to return a valid unit before opening any additional files. All of the subroutines referenced in INST8 are contained in the file usrfmc.f.

A.7 In case of trouble

Failed inversion will occur and have the following manifestations:

1. MICRON complains about an input error.
2. MICRON complains about a numerical error.
3. the solution is unrealistic.

If an error occurs the user should always

- check steps C and D described in this manual.
- check the output files to make sure that the input files were read correctly.
- check for warnings in the .**log** file.
- attempt to simulate data and construct a similar inversion problem with a known solution. If a similar inversion problem cannot be found, then the original problem may be unrealistic.

If the solution is unrealistic:

- use MICRON to find the least-squares solution. If the `rsdiff.` of any inverted datum in the .**out** file is much larger than 1, then the specified standard deviations may be unrealistic. Also, the minimum error should be less than the number of data.
- use an alternative smoothing algorithm. See below.

If a numerical error occurs:

- solve a simpler, related problem (e.g., reduce the solution dimension, eliminate constraints, choose the smoothing parameter, etc.)

Specific problems that may occur are the following

- the instrument response matrix can cause numerical difficulties due to scaling problems. For example, the number distribution response matrix for an impactor can cause difficulties; here, the problem can be avoided by working with the mass distribution.
- the algorithm that is used to choose the smoothing parameter can fail. If `RR` or `DISC` is used, then the specified standard deviations may be incorrect. If `GCV` is used, then the initial smoothing parameter may be poorly chosen, there may be too few data, or in some cases `GCV` can fail due to instabilities. This can be avoided in many cases by manually choosing the first smoothing parameter, or by using `RR` to choose the smoothing parameter.
- the user may have specified zero standard deviations; these place a severe constraint on the solution.

References

[1] R. A. Adams. *Sobolev Spaces.* Academic, 1975.

[2] J. K. Agarwal and G. J. Sem. Continuous flow, single-particle-counting condensation nucleus counter. *Journal of Aerosol Science*, 11:343–357, 1980.

[3] D. J. Alofs and P. Balakumar. Inversion to obtain aerosol size distributions from measurement with a differential mobility analyzer. *Journal of Aerosol Science*, 13(6):513–527, 1982.

[4] D. J. Alofs and P. Balakumar. Inversion to obtain aerosol size distributions from measurements with a differential mobility analyzer. *Journal of Aerosol Science*, 13:512–527, 1982.

[5] R. Ash. *Information Theory.* John Wiley and Sons, 1965.

[6] K. E. Atkinson. *An Introduction to Numerical Analysis.* John Wiley and Sons, 1978.

[7] Y. S. Cheng, J. A. Keating, and G. M. Kanapilly. Theory and calibration of a screen-type diffusion battery. *Journal of Aerosol Science*, 11:549–556, 1980.

[8] Y. S. Cheng and H. C. Yeh. Theory of a screen-type diffusion battery. *Journal of Aerosol Science*, 11:313–320, 1980.

[9] D. W. Copper and L. A. Spielman. Data inversion using nonlinear programming with physical constraints: aerosol size distribution measurement by impactors. *Atmospheric Environment*, 10:723–729, 1976.

[10] D. R. Cox and H. D. Miller. *The Theory of Stochastic Processes.* Chapman and Hall, 1965.

[11] P. Craven and G. Wahba. Smoothing noisy data with spline functions. *Numer. Math*, 31:377–403, 1979.

[12] J. G. Crump and J. H. Seinfeld. Further results on inversion of aerosol size distribution data: higher-order Sobolev spaces and constraints. *Aerosol Science and Technology*, 1:363–369, 1982.

[13] J. G. Crump and J. H. Seinfeld. A new algorithm for inversion of aerosol size distribution data. *Aerosol Science and Technology*, 1:15–34, 1982.

[14] E. de Doncker and I. Robinson. TRIEX: integration over a triangle using nonlinear extrapolation, algorithm no. 612. *ACM Trans. Math. Softw.*, 10:17–22, 1984.

[15] J. Franklin. *Methods of Mathematical Economics.* Springer-Verlag, 1980.

[16] J. N. Franklin. Well–posed stochastic extensions of ill-posed linear problems. *Journal of Mathematical Analysis and Applications*, 31:682–716, 1970.

[17] H. W. Gaggeler, U. Baltensperger, M. Emmenegger, D. T. Jost, A. Schmidt–Ott, P. Haller, and M. Hofmann. The epiphaniometer, a new device for continuous aerosol monitoring. *Journal of Aerosol Science*, 20:557–564, 1989.

[18] P. E. Gill, S. J. Hammarling, W. Murray, M. A. Saunders, and M. H. Wright. *User's guide for LSSOL (Version 1.0): A Fortran Package for Constrained Linear Least-Squares and Convex Quadratic Programming.* Technical Report SOL 86–1, Systems Optimization Laboratory, Department of Operations Research, Stanford University, January 1986.

[19] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. Procedures for optimization problems with a mixture of bounds and general linear constraints. *ACM Transactions on Mathematical Software*, 10:282–298, 1984.

[20] G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21:215–223, 1979.

[21] C. W. Groetsch. *The Theory of Tikhonov Regularization for Fredholm equations of the first kind.* Pitman Publishing Limited, 1984.

[22] C. Gu and G. Wahba. *Minimizing GCV/GML scores with multiple smoothing parameters via the Newton method.* Technical Report 847, Dept. of Statistics, University of Wisconsin, 1988.

[23] D. E. Hagen and D. J. Alofs. Linear inversion method to obtain aerosol size distributions from measurements with a differential mobility analyzer. *Aerosol Science and Technology*, 2:465–475, 1983.

[24] C. Helsper, H. Fissan, A. Kapadia, and B. Y. H. Liu. Data inversion by simplex minimization for the electrical aerosol analyzer. *Aerosol Science and Technology*, 1:135–146, 1982.

[25] D. K. Kahaner and O. W. Rechard. TWODQD an adaptive routine for two-dimensional integration. *Journal of Computational and Applied Mathematics*, 17:215–234, 1987.

[26] G. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33:82–95, 1971.

[27] Y. Kousaka, K. Okuyama, and M. Adachi. Determination of particle size distribution of ultra-fine aerosols using a differential mobility analyzer. *Aerosol Science and Technology*, 4:209–225, 1985.

[28] C. Lanczos. *Linear Differential Operators.* D. Van Nostrand Co., London, 1961.

[29] D. P. Laurie. CUBTRI:, automatic cubature over a triangle. *ACM Trans. Math. Softw.*, 8:210–218, 1982.

[30] K. Li. Asymptotic optimality for c_p, c_L, cross validation and generalized cross-validation: discrete index set. *Annals of Statistics*, 15:958–975, 1987.

[31] K. Li. Asymptotic optimality of c_L and generalized cross-validation in ridge regression with application to spline smoothing. *Annals of Statistics*, 14:1101–1112, 1986.

[32] E. F. Maher and N. M. Laird. EM algorithm reconstruction of particle size distributions from diffusion battery data. *Journal of Aerosol Science*, 16:557–570, 1985.

[33] G. R. Markowski. Improving Twomey's algorithm for inversion of aerosol measurement data. *Aerosol Science and Technology*, 7:127–141, 1987.

[34] V. A. Morozov. *Methods for Solving Incorrectly Posed Problems.* Springer-Verlag New York Inc., 1984.

[35] V. A. Morozov. On the solution of functional equations by the method of regularization. *Soviet Math. Doklady*, 7:414–417, 1966.

[36] P. Paatero and T. Raunemaa. Analysis of CO_2 thermograms by the new extreme-value estimation (eve) deconvolution principal. *Aerosol Science and Technology*, 10:365–369, 1989.

[37] P. Paatero, T. Raunemaa, and R. L. Dod. Composition characteristics of carbonaceous particle samples, analyzed by eve deconvolution method. *Journal of Aerosol Science*, 19:1223–1226, 1988.

[38] D. L. Phillips. A technique for the numerical solution of certain integral equations of the first kind. *J. Assn. Comp. Mach.*, 9:84–97, 1962.

[39] C. H. Reinsch. Smoothing by spline functions. II. *Numer. Math*, 16:451–454, 1971.

[40] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems.* V. H. Winston and Sons, Washington D. C., 1977.

[41] S. Twomey. Comparison of constrained linear inversion and an iterative nonlinear algorithm applied to the indirect estimation of particle size distributions. *Journal of Computational Physics*, 18:188–200, 1975.

[42] S. Twomey. On the numerical solution of Fredholm integral equations of the first kind by the inversion of the linear system produced by quadrature. *J. Assn. Comp. Mach.*, 10:97–101, 1963.

[43] M. Villalobos and G. Wahba. Inequality-constrained multivariate smoothing splines with application to the estimation of posterior probabilities. *Journal of the American Statistical Association*, 82:239–248, 1987.

[44] G. Wahba. Smoothing noisy data with spline functions. *Numer. Math.*, 24:383–393, 1975.

[45] G. Wahba and J. Wendelberger. Some new mathematical methods for variational objective analysis using splines and cross validation. *Monthly Weather Review*, 108:1122–1143, 1980.

[46] S. C. Wang and R. C. Flagan. Scanning electrical mobility spectrometer. *Aerosol Science and Technology*, 1990. in press.

[47] J. K. Wolfenbarger and J. H. Seinfeld. Inversion of aerosol size distribution data. *Journal of Aerosol Science*, 1989. in press.

[48] J. K. Wolfenbarger and J. H. Seinfeld. Regularized solutions to the aerosol data inversion problem. *SIAM Journal on Scientific and Statistical Computing*, 1990. in press.

[49] J. J. Wu, D. W. Cooper, and R. J. Miller. Evaluation of aerosol deconvolution algorithms for determining submicron particle size distributions with the diffusion battery and condensation nucleus counter. *Journal of Aerosol Science*, 20:477–482, 1989.