NONPARAMETRIC ESTIMATION OF MEAN FIELDS WITH APPLICATION TO PARTICLE METHODS FOR TURBULENT FLOWS

by

Thomas D. Dreeben and Stephen B. Pope

FDA 92-13 November 1992

Fluid Dynamics and Aerodynamics Program

Sibley School of
Mechanical and Aerospace Engineering

Cornell University Ithaca, New York 14853

1 Background and Objectives

In the application of pdf methods to turbulent flows, a particle method is used to solve a modeled equation for the evolution of a pdf. To determine the coefficients of the equation at each time step, it is necessary to use particle data to estimate mean quantities such as the mean velocity or the Reynolds stress as a function of position in physical space [1]. In this paper we consider three different methods to estimate mean quantities. Two of them have estimates calculated at evenly spaced points; the third one is performed on a grid with randomly generated spaces so that the results can be extended for use with unstructured grids in higher dimensions. We describe how the methods work and use a test function to determine how their accuracy depends upon relevant parameters.

2 Estimating Means for a Structured Grid

Suppose we are given data y_i' with $i = 1, 2, \ldots, N$ at evenly spaced x_i' over the interval $[a, b]$. As in the literature [2, 3, 4, 5, 6], the data is modeled as

$$y_i' = f(x_i') + \sigma\xi_i. \tag{1}$$

In our case, the random variables ξ_i are taken to be independent, identically distributed (iid), and standardized Gaussian. The data y_i' are then modeled as iid Gaussian random variables with mean $f(x_i')$ and variance σ^2. The task of the algorithm is to construct an estimate $\phi(x)$ for the function $f(x)$ from the data.

The domain $[a, b]$ is partitioned by $m + 1$ equally spaced points, called *knots*. Let $\Delta x = \frac{b-a}{m}$ denote the distance between any two adjacent knots. Then for $j = 0, 1, \ldots, m$, we define X_j to be the location of the j-th knot:

$$X_j = a + j\Delta x, \tag{2}$$

With this definition, the first and last knots (X_0,X_m) coincide with endpoints a and b. To construct the estimate, we make $\phi(x)$ a linear spline [3, 7]. This is a continuous piecewise-linear function where the slope is discontinuous at the knots. The spline is fully specified by its values at the knots. Hence, using a spline reduces the task of the algorithm to estimating the function f at the knots, X_j.

2.1 Two Estimation Algorithms

2.1.1 Cloud-in-Cell Method

The *cloud-in-cell* method is used to estimate $f(x)$ at the knots. This method is a variation of the approach proposed by Priestly and Chao [2]. We define a kernel K by

$$K(u) \equiv \begin{cases} 1 - |u| & \text{if } |u| \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Support for the kernel centered at the j-th knot consists of those x'_i for which

$$K\left(\frac{x'_i - X_j}{\Delta x}\right) > 0. \tag{4}$$

On the boundaries we extend the data periodically by defining

$$\begin{aligned} x'_i &= x'_{i-N} \text{ for } i > N & (5)\\ \text{and } x'_i &= x'_{i+N} \text{ for } i \leq 0. & (6) \end{aligned}$$

Then for $j = 1, \ldots, m$, the estimate $\phi(X_j)$ for f at the knot X_j is

$$\phi(X_j) \equiv c_j = \frac{\sum_i y'_i K\left(\frac{x'_i - X_j}{\Delta x}\right)}{\sum_i K\left(\frac{x'_i - X_j}{\Delta x}\right)}. \tag{7}$$

The coefficients c_j can be expressed as the solution to a matrix equation. Let $\underline{c}$ be the vector $(c_1, c_2, \ldots, c_m)^T$, $\underline{\underline{\mathbf{A}}}$ be the $m \times m$ diagonal matrix with

$$A_{jj} = \sum_i K\left(\frac{x'_i - X_j}{\Delta x}\right), \tag{8}$$

and $\underline{\mathbf{b}}$ be the vector $(b_1, b_2, \ldots, b_m)^T$ with

$$b_j = \sum_i y'_i K\left(\frac{x'_i - X_j}{\Delta x}\right), \tag{9}$$

then $\underline{c}$ is the solution to the matrix equation

$$\underline{\underline{\mathbf{A}}}\,\underline{\mathbf{c}} = \underline{\mathbf{b}}. \tag{10}$$

To construct the linear spline $\phi(x)$ from the vector $\underline{\mathbf{c}}$, we define basis functions $\tilde{b}_j(x)$. For $j = 1, 2, \ldots, m-1$, the basis function which is centered at the knot X_j is the linear spline

$$\tilde{b}_j(x) \equiv K\left(\frac{x - X_j}{\Delta x}\right) \tag{11}$$

where K is given by Eq. (3). The boundaries are treated with a special basis function

$$\tilde{b}_m(x) \equiv \begin{cases} K\left(\frac{x-X_0}{\Delta x}\right) & \text{for } X_0 < x \leq X_1 \\ K\left(\frac{x-X_m}{\Delta x}\right) & \text{for } X_{m-1} < x \leq X_m \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Then $\phi(x)$ is a linear combination of the basis functions with the estimates at the knots, $(c_j,\ j = 1, \ldots, m)$ used as the basis coefficients:

$$\phi(x) \equiv \sum_{j=1}^{m} c_j \tilde{b}_j(x). \tag{13}$$

2.1.2 Least Squares Method

The second estimation algorithm is *least squares linear splines* [3, 8]. In this application, least squares is identical to cloud-in-cell in everything except for the matrix $\underline{\underline{\mathbf{A}}}$ in Eq. (10). Here, $\underline{\underline{\mathbf{A}}}$ is determined by the normal equations which are described on pages 92-3 of [8]. The solution to these equations minimizes the expression

$$\sum_i [\phi(x'_i) - y'_i]^2. \tag{14}$$

The matrix is an $m \times m$ periodic tridiagonal matrix with elements

$$A_{jk} = \sum_i K\left(\frac{x'_i - X_j}{\Delta x}\right) K\left(\frac{x'_i - X_k}{\Delta x}\right). \tag{15}$$

Eq. (10) is solved for c_j; then the estimate $\phi(x)$ is the linear spline given in Eq. (13).

2.2 Testing the Methods

These two algorithms were implemented using data generated by the test function $f(x) = \sin x$ over the interval $[0, 2\pi]$. The error was characterized by the integrated mean squared error, which is

$$\epsilon \equiv \frac{1}{2\pi}\int_0^{2\pi} \langle [\phi(x) - f(x)]^2 \rangle \, dx. \tag{16}$$

This error is split into a deterministic part and a statistical part where

$$\begin{aligned} \epsilon &= \epsilon_d + \epsilon_s, & (17)\\ \epsilon_d &\equiv \frac{1}{2\pi}\int_0^{2\pi} [\langle \phi(x)\rangle - f(x)]^2 \, dx, & (18)\\ \text{and } \epsilon_s &\equiv \frac{1}{2\pi}\int_0^{2\pi} \langle [\phi(x) - \langle \phi(x)\rangle]^2 \rangle \, dx. & (19) \end{aligned}$$

The deterministic error (also known as the square of the bias) is a truncation error which occurs because the function f is being estimated at only a finite number of points X_j. The statistical error is caused by the random fluctuations.

The condition of *dense data* exists when each kernel is supported by a sufficiently large number of particles that the deterministic error is independent of N. Under dense data, the test is characterized by two parameters, γ and Δx. The quantity γ is defined to be σ^2/N. With the cloud-in-cell method it can be shown [5] that to leading order, the deterministic error varies as Δx^4 and that the statistical error varies as $\gamma/\Delta x$. So for sufficiently small Δx, there are constants α and β such that

$$\epsilon \approx \alpha \Delta x^4 + \beta \frac{\gamma}{\Delta x}. \tag{20}$$

For the test function $f(x) = \sin x$, Figure 1 shows how ϵ varies with Δx for different fixed values of γ with the cloud-in-cell method. For small Δx we see that the statistical error dominates the expression for ϵ because ϵ varies inversely with Δx and it depends upon γ. For large Δx the deterministic error dominates. Here all the curves collapse into one curve with positive slope to show the independence of ϵ upon γ and how ϵ scales with Δx^4.

This dependence upon Δx reflects a trade-off between deterministic and statistical error [5]. The deterministic error characterizes the inability of the estimate to detect variations in the underlying function f with position x. For a given kernel bandwidth Δx, the estimate $\phi(X_j)$ is determined by the particles which support the kernel centered at X_j. So $\phi(x)$ is sensitive to variations in f between kernels in different locations, but it is relatively insensitive to variations in f within the same kernel; those variations are mostly averaged out. As Δx increases, each estimate $\phi(X_j)$ depends upon a larger number of particles, so the estimate is less able to detect local variations in the underlying function f. This means that as Δx increases, so does the deterministic error. Statistical error works in the opposite way. As Δx increases, the larger number of sample particles reduces the sensitivity of $\phi(X_j)$ to random fluctuations in the data, thus reducing the statistical error. So overall, as Δx increases the deterministic error increases and the statistical error decreases. As Figure 1 shows, for each fixed γ there is an optimal Δx which minimizes ϵ by providing the best balance between deterministic and statistical error.

Figure 2 shows similar results for least squares. Here we see the same tradeoff between the deterministic and statistical error.

To compare the accuracy of the two methods, a single curve was sought to characterize the error for each method. The curve was derived by plotting a group of (ϵ, γ) against a group of $(\Delta x, \gamma)$. The groups were chosen so that as the independent variable approaches its extreme values, the family of curves approaches a single curve. If we assume that Eq. (20) applies to both methods, then it follows that

$$\frac{\epsilon}{\gamma^{0.8}} \approx \alpha \left(\frac{\Delta x}{\gamma^{0.2}} \right)^4 + \frac{\beta}{\left(\frac{\Delta x}{\gamma^{0.2}} \right)} \tag{21}$$

Figure 3 shows $\frac{\epsilon}{\gamma^{0.8}}$ plotted against $\frac{\Delta x}{\gamma^{0.2}}$ for both algorithms. This Figure shows that the optimum collapsed error is slightly smaller with cloud-in-cell, and it occurs at a Δx which is half as large as that with least squares. The optimum Δx's appear more clearly in Figure 4, which shows the same data on a linear plot. These results require that the estimation be performed on a structured regular grid, which is specified by the choice of Δx.

3 Estimating Means for an Unstructured Grid

For applicability to pdf methods, we use the above ideas to construct an algorithm which can be extended to accommodate an irregular or unstructured grid. The problem is posed as follows: We are given data in one dimension which is modeled by Eq. (1). So that the model can be extended to unstructured grids in higher dimensions, the knots $X_j, j = 1, \ldots, m$, are given at irregular intervals over which the data is dense. As with previous methods, the estimate $\phi(x)$ is represented as a linear spline which is specified by its values at the given knots. However estimation for the values of ϕ at the knots X_j requires a different algorithm to allow for the balance of statistical and deterministic error. With previous methods the minimum error is achieved by a proper adjustment of the grid spacing. Here, the grid has been specified in the problem, so a new algorithm must allow for minimizing the error independently of the grid.

3.1 The Two Stage Estimation Algorithm

The method of local least squares [4] is used to estimate the function at the knots. To reduce computational cost, local least squares is preceded by a first stage of kernel estimates, similar to those with the cloud-in-cell method. Here the kernels are chosen to match the irregular grid, and the estimates are taken at the center of mass of the particles, weighted by the local kernels which they support. These first stage estimates are then used as weighted data for the second stage in which we use the local least squares algorithm.

3.1.1 Stage 1

The domain is partitioned into unequal subintervals, bounded by knots X_j. A kernel H_j is chosen to accommodate the irregularly spaced knots. We use the linear splines described as hat functions in [7]:

$$H_j(x) \equiv \begin{cases} \frac{x - X_{j-1}}{X_j - X_{j-1}} & \text{for } X_{j-1} \leq x < X_j \\ \frac{X_{j+1} - x}{X_{j+1} - X_j} & \text{for } X_j \leq x \leq X_{j+1} \\ 0 & \text{otherwise} \end{cases} . \tag{22}$$

The first stage generates the following three quantities: The weight of particles which support the kernel at the j-th knot,

$$w_j = \sum_i H_j(x_i); \tag{23}$$

the center of mass of particles which support the kernel at the j-th knot,

$$\overline{X_j} = \frac{\sum_i x_i H_j(x_i)}{w_j}; \tag{24}$$

and the estimate for f at the center of mass,

$$\overline{\phi_j} = \frac{\sum_i y_i H_j(x_i)}{w_j}. \tag{25}$$

The output w_j, $\overline{X_j}$, and $\overline{\phi_j}$ completes the first stage.

3.1.2 Stage 2

In stage 2, we use the local least squares algorithm [4] to estimate f at the knots, X_j. The points from stage 1, $(\overline{X_i}, \overline{\phi_i}), i = 1, \ldots, m$ are used as input data with weights w_i. The local least squares algorithm provides an estimate for f at X_j by locally fitting a polynomial to data which lies within a neighborhood of X_j. The size of the neighborhood is characterized by the bandwidth W, which is used as a parameter in the algorithm. For each estimate, the data is weighted with a kernel Q, where

$$Q(u) \equiv \begin{cases} (1-u^2)^2 & \text{if } |u| \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

Then $\tilde{\phi}_j(x)$ is a polynomial estimate for $f(x)$ in a neighborhood of the j-th knot which minimizes the expression

$$\sum_i Q\left(\frac{\overline{X_i} - X_j}{W}\right) w_i [\tilde{\phi}_j(X_i) - \overline{\phi_i}]^2. \tag{27}$$

A general method for choosing the order of the polynomial is described in [4]. In this application, first and second order polynomials are considered.

The *linear two stage algorithm* is implemented by fitting a first order polynomial to the points $(\overline{X_i}, \overline{\phi_i})$ in a neighborhood within a distance W centered at the knot X_j. We take a function of the form

$$\tilde{\phi}_j(x) = \hat{a} + \hat{b}(x - X_j), \tag{28}$$

where $\hat{a}$ and $\hat{b}$ are constants to be determined. If we introduce the notation

$$\hat{X}_{ij} \equiv \overline{X_i} - X_j \tag{29}$$

$$\hat{Q}_{ij} \equiv w_i Q\left(\frac{\hat{X}_{ij}}{W}\right) \tag{30}$$

then the constants $\hat{a}$ and $\hat{b}$ which minimize Eq. (27) are determined by solving the matrix equation

$$\begin{bmatrix} \sum_i \hat{Q}_{ij} & \sum_i \hat{Q}_{ij}\hat{X}_{ij} \\ \sum_i \hat{Q}_{ij}\hat{X}_{ij} & \sum_i \hat{Q}_{ij}\hat{X}_{ij}^2 \end{bmatrix} \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = \begin{bmatrix} \sum_i \hat{Q}_{ij}\overline{\phi_i} \\ \sum_i \hat{Q}_{ij}\overline{\phi_i}\hat{X}_{ij} \end{bmatrix}. \tag{31}$$

The *quadratic two stage algorithm* is implemented by fitting a second order polynomial; here we take a function of the form

$$\tilde{\phi}_j(x) = \hat{a} + \hat{b}(x - X_j) + \hat{c}(x - X_j)^2 \tag{32}$$

in place of Eq. (28). Then the constants $\hat{a}$, $\hat{b}$, and $\hat{c}$ which minimize Eq. (27) are determined by solving the quadratic version of Eq. (31),

$$\begin{bmatrix} \sum_i \hat{Q}_{ij} & \sum_i \hat{Q}_{ij}\hat{X}_{ij} & \sum_i \hat{Q}_{ij}\hat{X}_{ij}^2 \\ \sum_i \hat{Q}_{ij}\hat{X}_{ij} & \sum_i \hat{Q}_{ij}\hat{X}_{ij}^2 & \sum_i \hat{Q}_{ij}\hat{X}_{ij}^3 \\ \sum_i \hat{Q}_{ij}\hat{X}_{ij}^2 & \sum_i \hat{Q}_{ij}\hat{X}_{ij}^3 & \sum_i \hat{Q}_{ij}\hat{X}_{ij}^4 \end{bmatrix} \begin{bmatrix} \hat{a} \\ \hat{b} \\ \hat{c} \end{bmatrix} = \begin{bmatrix} \sum_i \hat{Q}_{ij}\overline{\phi_i} \\ \sum_i \hat{Q}_{ij}\overline{\phi_i}\hat{X}_{ij} \\ \sum_i \hat{Q}_{ij}\overline{\phi_i}\hat{X}_{ij}^2 \end{bmatrix}. \tag{33}$$

For either order polynomial fit (1, 2, or higher if desired), the estimate for f at X_j is

$$\phi(X_j) \equiv \tilde{\phi}_j(X_j) = \hat{a}. \tag{34}$$

The above procedure is repeated for each X_j to generate the estimates $\phi(X_j) = c_j$. Then the estimate at any x is represented as a linear spline. Here, the basis functions $\tilde{b}_j(x)$ are defined using the hat functions of Eq. (22) for $j = 0, \ldots, m$:

$$\tilde{b}_j(x) \equiv H_j(x). \tag{35}$$

Then the estimate $\phi(x)$ is the linear spline given by Eq. (13) as with previous methods.

3.2 Testing the Two Stage Method

The two stage algorithm was implemented using data generated by the same test function as before, $y = \sin x$ over the interval $[0, 2\pi]$, with a specified irregular grid. The grid specification is based on a reference length Δx_{ref}. A uniform $[0.1, 1.1]$ random number is multiplied by Δx_{ref} to generate each grid space. Then all grid spaces are scaled down slightly, so that the grid fits exactly into the domain $[0, 2\pi]$. The length Δx_{ref} is chosen so that the data $(x_i, y_i), i = 1, \ldots, N$ is dense for the first stage, and the output of the first stage $(\overline{X}_j, \overline{\phi_j}), j = 1, \ldots, m$ is dense data for the second stage. For purposes of evaluating the error, the final estimates for the function f were taken not on the irregular grid (as they would be in a real application), but at $M = 65$ evenly spaced intervals, bounded by $\tilde{X}_j, j = 0, \ldots, M$. The error is then characterized as a summed mean squared error:

$$\epsilon \equiv \frac{1}{M} \sum_{j=1}^{M} \langle [\phi(\tilde{X}_j) - f(\tilde{X}_j)]^2 \rangle. \tag{36}$$

Again ϵ is decomposed into a deterministic and statistical part where

$$\begin{aligned} \epsilon &= \epsilon_d + \epsilon_s, && (37) \\ \epsilon_d &\equiv \frac{1}{M} \sum_{j=1}^{M} [\langle \phi(\tilde{X}_j) \rangle - f(\tilde{X}_j)]^2, && (38) \\ \text{and } \epsilon_s &\equiv \frac{1}{M} \sum_{j=1}^{M} \langle [\phi(\tilde{X}_j) - \langle \phi(\tilde{X}_j) \rangle]^2 \rangle. && (39) \end{aligned}$$

With dense data, the test here is characterized by the parameters γ and W. Recall that γ is σ^2/N and W is the bandwidth of the kernel used in the second stage (local least squares) of the algorithm.

For the local least squares algorithm, the behavior of the error is described in Ruppert [6]. Ruppert's results apply to the cases considered here because the output from the first stage algorithm is dense data for the local least squares stage. These results imply that the behavior of the two stage error is similar to that with cloud-in-cell described in Eq. (20). With the linear fit, there are constants α_l and β_l such that

$$\epsilon \approx \alpha_l W^4 + \beta_l \frac{\gamma}{W}, \tag{40}$$

and with the quadratic fit, there are constants α_q and β_q such that

$$\epsilon \approx \alpha_q W^6 + \beta_q \frac{\gamma}{W}. \tag{41}$$

Figures 5 and 6 show how ϵ varies with W for different fixed values of γ for the two stage method. Figure 5 gives results for the linear fit; Figure 6 is for the quadratic fit. By comparing Figures 5 and 6 with Figures 1 and 2, we see that the dependence of ϵ upon γ and the bandwidth with the two stage algorithm is qualitatively similar to this dependence with the cloud-in-cell and least squares methods. At small bandwidths the statistical error is the dominant term and at large bandwidths the deterministic error is the dominant term. For each value of γ there is an optimal choice of the bandwidth W which provides the best balance between statistical and deterministic error.

Because the error associated with the linear two stage algorithm varies with W and γ in the same way that the cloud-in-cell error varies with Δx and γ, it is useful to derive a family of curves for the two stage algorithm which are collapsed onto one curve, like those in Figure 3. We use Eq. (40) to find an expression which is analogous to Eq. (21), for a group of (ϵ, γ) as a function of a group of (W, γ):

$$\frac{\epsilon}{\gamma^{0.8}} \approx \alpha_l \left(\frac{W}{\gamma^{0.2}}\right)^4 + \frac{\beta_l}{\left(\frac{W}{\gamma^{0.2}}\right)}. \tag{42}$$

Figure 7 shows $\frac{\epsilon}{\gamma^{0.8}}$ plotted against $\frac{W}{\gamma^{0.2}}$ for the two stage algorithm using linear local least squares. Referring to Figure 7 and Figure 3 we see that the minimum error with the linear two stage algorithm is comparable to the minimum error with cloud-in-cell, and it occurs at an optimum bandwidth which is between those with cloud-in-cell and with least squares.

Collapsed error curves can also be derived for the quadratic two stage case. Based on Eq. (41), we have

$$\frac{\epsilon}{\gamma^{0.86}} \approx \alpha_q \left(\frac{W}{\gamma^{0.14}}\right)^6 + \frac{\beta_q}{\left(\frac{W}{\gamma^{0.14}}\right)} \tag{43}$$

Figure 8 shows $\frac{\epsilon}{\gamma^{0.86}}$ plotted against $\frac{W}{\gamma^{0.14}}$ for the quadratic two stage algorithm.

4 Summarizing the Results

The errors for all of the methods can be expressed in a compact form so that the optimum bandwidth and corresponding error can be estimated. Consider the data modeled by Eq. (1) with the sinusoidal test function

$$f(x) = a \sin \frac{2\pi x}{L} \tag{44}$$

taken over the domain $[0, L]$. Let λ be a non-dimensional bandwidth: for the cases considered here,

$$\lambda \equiv \begin{cases} \frac{2\pi \Delta x}{L} & \text{for cloud-in-cell and least squares} \\ \frac{2\pi W}{L} & \text{for the two stage algorithms.} \end{cases} \tag{45}$$

and let p be an integer power. For the algorithms considered here, each of the error curves are well approximated by the expression

$$\epsilon \approx \hat{\epsilon} \equiv a^2[\alpha\lambda^p + \beta\frac{\gamma}{\lambda}], \tag{46}$$

where α and β depend upon which algorithm is used. Here, $p = 6$ for the quadratic two stage method, and $p = 4$ for the other methods. In each case, the optimum bandwidth and the corresponding error can be estimated by setting

$$\frac{d\hat{\epsilon}}{d\lambda} = 0 \tag{47}$$

and solving for λ and $\hat{\epsilon}$ in terms of α, β, γ, and p. If we let $\hat{\lambda}_o$ and $\hat{\epsilon}_o$ denote the estimates for the optimum bandwidth and minimum error respectively, then Eqs. (46) and (47) give

$$\hat{\lambda}_o \equiv \left(\frac{\beta\gamma}{p\alpha}\right)^{\frac{1}{p+1}} \tag{48}$$

and

$$\hat{\epsilon}_o \equiv a^2\alpha(p+1)\hat{\lambda}_o^p. \tag{49}$$

Now Eqs. (46), (48), and (49) can be used to find $\hat{\epsilon}(\lambda)$, $\hat{\lambda}_o$, and $\hat{\epsilon}_o$, if α and β are known. The error results shown in Figures 1, 2, 5, and 6 were used to construct estimates $\hat{\alpha}$ and $\hat{\beta}$, for α and β, using a parametric least squares

algorithm	$\hat{\alpha}$	$\hat{\beta}$
cloud-in-cell	1.30×10^{-2}	3.17
least squares	8.32×10^{-4}	6.60
linear two stage	2.07×10^{-3}	4.92
quadratic two stage	4.24×10^{-6}	9.73

Table 1: Estimates for α and β

algorithm. The deterministic portions were fit to a polynomial of order p in λ where $p = 6$ for the quadratic two stage algorithm and $p = 4$ for each of the others. The statistical portions were fitted to vary inversely with λ. These estimates for α and β are shown in Table 1. With these values, Eq. (46) becomes a good approximation for the results in Figures 1, 2, 5, and 6. Figures 9 and 10 show estimates $\hat{\lambda}_o$ and $\hat{\epsilon}_o$ as a function of γ, based on Eqs. (48) and (49). Empirical results are shown as points on the plot. Discrepancies between the data and the estimates are most pronounced with the plots of optimum bandwidth in Figure 9. These occur because the tests were run at discrete values of λ whereas the model values are determined by minimizing the value of a smooth function $\hat{\epsilon}$ of λ. In some cases, most notably the linear two stage method, the optimum bandwidth *of those bandwidths which were tested* is far enough from the true optimum bandwidth to cause appreciable error in the experimental results shown in Figure 9. It should be emphasized that the estimates for α and β are determined using the fact that $f(x)$ is sinusoidal for the cases considered here. In general $f(x)$ is unknown, so other means must be employed to estimate α and β in order to make use of Eqs. (46), (48), and (49).

5 Conclusions

5.1 Estimation Over Adjustable Grids

The cloud-in-cell and least squares methods are well suited to applications in which the grid spacing can be chosen to give the near minimum error. It is clear from Figures 3, 4 and 10 that the minimum error is comparable for both of these methods. Because least squares involves solving a periodic tridiagonal matrix equation, it is computationally more costly than cloud-in-cell. This cost increase is significant in 2 and 3 dimensions.

5.2 Estimation Over Specified Grids

The two stage method is better suited to applications in which a grid has been specified; there the error is optimized by adjusting the bandwidth independently of the grid. Referring to Figure 10, the quadratic two stage algorithm has a significantly smaller minimum error than does the linear two stage algorithm for the values of γ considered. However the slope $\frac{d\hat{\epsilon}_o}{d\gamma}$ is steeper with the quadratic two stage algorithm than with any of the others. This occurs because of the dependence of $\hat{\epsilon}_o$ upon p in Eq. (48) through Eq. (49), and the fact that $p = 6$ for the quadratic two stage method and $p = 4$ for the other methods. When $\gamma = 6.5 \times 10^{-4}$ the quadratic two stage minimum error line in Figure 10 crosses the lines for the other methods. So for sufficiently large γ, the quadratic two stage algorithm offers no advantage over the other algorithms because it incurs an excessively large statistical error. The quadratic two stage algorithm is also more costly than the linear two stage algorithm because it involves solving a 3×3 matrix equation rather than a 2×2 equation.

5.3 Optimum Bandwidth

Figure 9 shows how the estimated optimum bandwidths vary with γ for each of the different methods. This is important for computational cost with the two stage methods. As the bandwidth increases, the amount of data which supports each kernel also increases while the number of kernels remains fixed. So a larger bandwidth requires more calculations on the data in the second stage. If the error is minimized there is added computational expense involved with the quadratic two stage algorithm because its minimum error occurs at the largest bandwidth for the values of γ used here.

5.4 Sensitivity to Changes in Bandwidth

We can approximate sensitivity of the error to deviations of λ from λ_o using Eq. (46). Sensitivity here is taken to mean a fractional change in a quantity. For λ close to λ_o the error can be expressed as a Taylor Series expansion about λ_o:

$$\delta\epsilon = \left(\frac{\partial\epsilon}{\partial\lambda}\right)_{\lambda_o}\delta\lambda + \frac{1}{2}\left(\frac{\partial^2\epsilon}{\partial\lambda^2}\right)_{\lambda_o}\delta\lambda^2 + \ldots \tag{50}$$

Using Eqs. (46), (48), and (49), and dropping terms above second order, this becomes

$$\frac{\delta\hat{\epsilon}}{\hat{\epsilon}_o} \approx \frac{p}{2}\left(\frac{\delta\hat{\lambda}}{\hat{\lambda}_o}\right)^2 \tag{51}$$

With the sensitivity expressed in this way, it is clear that when the bandwidth is changed by a given factor θ, then the corresponding change in error depends primarily on p, and is directly proportional to θ^2. So the cloud-in-cell, least squares, and linear two stage methods are all equally sensitive to changes in bandwidth, and the quadratic two stage method is more sensitive because it has a larger value of p. Figures 11 through 14 show estimates of $\frac{\delta\epsilon}{\epsilon_o}$ and $\frac{\delta\lambda}{\lambda_o}$ based on Eq. (51). Each such estimate is shown as a dashed curve; the empirical results are shown as points on the plot. The data is based the empirical values of optimum bandwidth and minimum error shown in Figures 9, and 10; hence they are subject to the same error caused by the sampling of bandwidths at discrete points in the testing. This is the primary reason for the scatter in the data found in Figures 11 through 14.

References

[1] S.B. Pope. Pdf Methods for Turbulent Reactive Flows. *Progress in Energy and Combustion Science*, 11:119–192, 1985.

[2] M. B. Priestly and M. T. Chao. Non-parametric Function Fitting. *Journal of the Royal Statistical Society*, 34:385–392, 1972.

[3] R. L. Eubank. *Spline Smoothing and Nonparametric Regression.* Marcel Dekker, inc., 1988.

[4] W. S. Cleveland, S. J. Devlin, and E. Grosse. Regression by Local Fitting: Methods, Properties, and Computational Algorithms. *Journal of Econometrics*, 37:87–114, 1988.

[5] W. Hardle. *Applied Nonparametric Regression.* Cambridge University Press, 1990.

[6] D. Ruppert and M. P. Wand. Multivariate Locally Weighted Least Squares Regression. Technical report, Cornell University, Ithaca, N.Y., January 1992.

[7] C. de Boor. *A Practical Guide to Splines.* Springer-Verlag, 1978.

[8] G. Dahlquist and A. Bjorck. *Numerical Methods.* Prentice Hall, 1974.

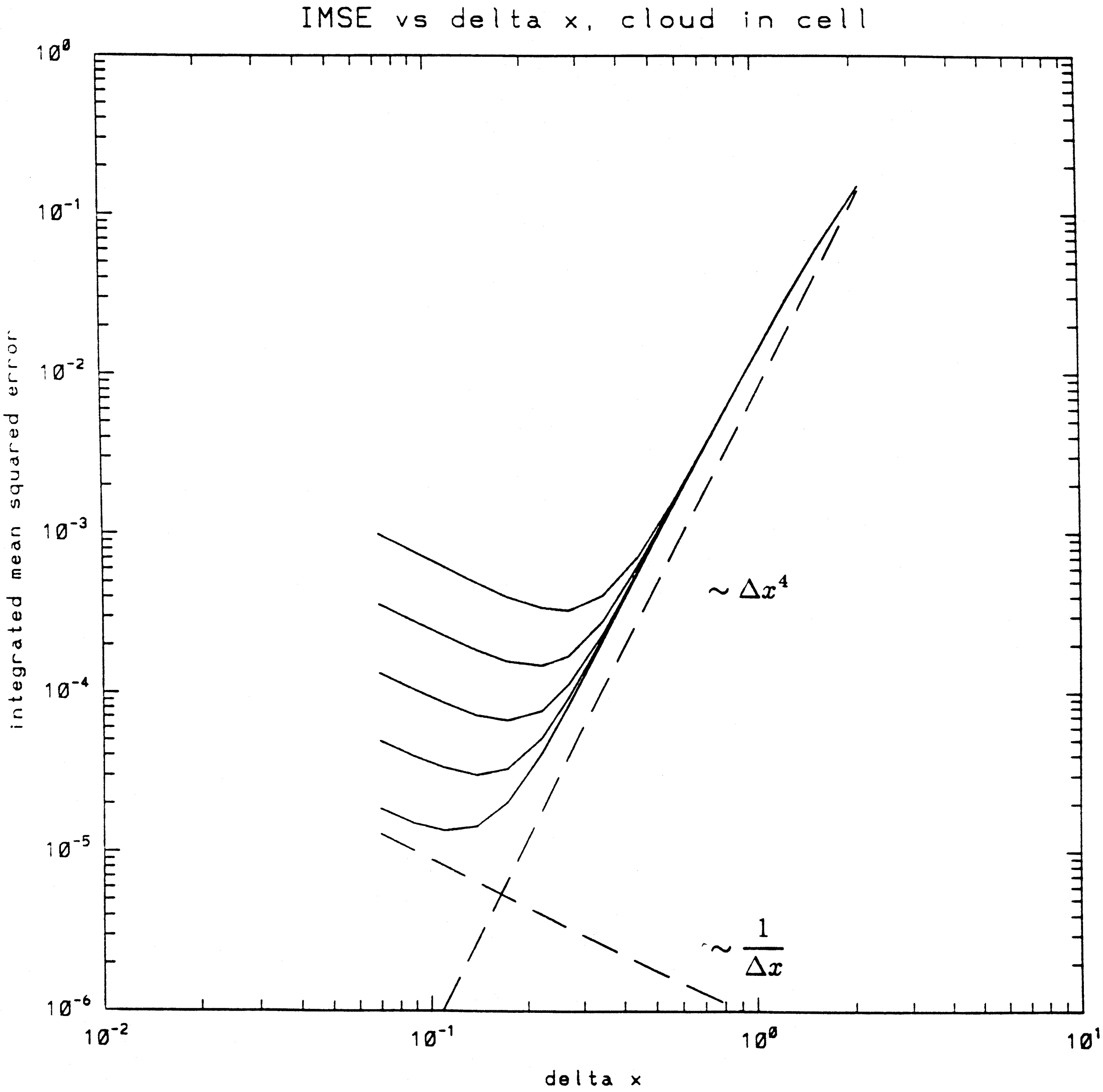

Figure 1: Error for the cloud in cell method

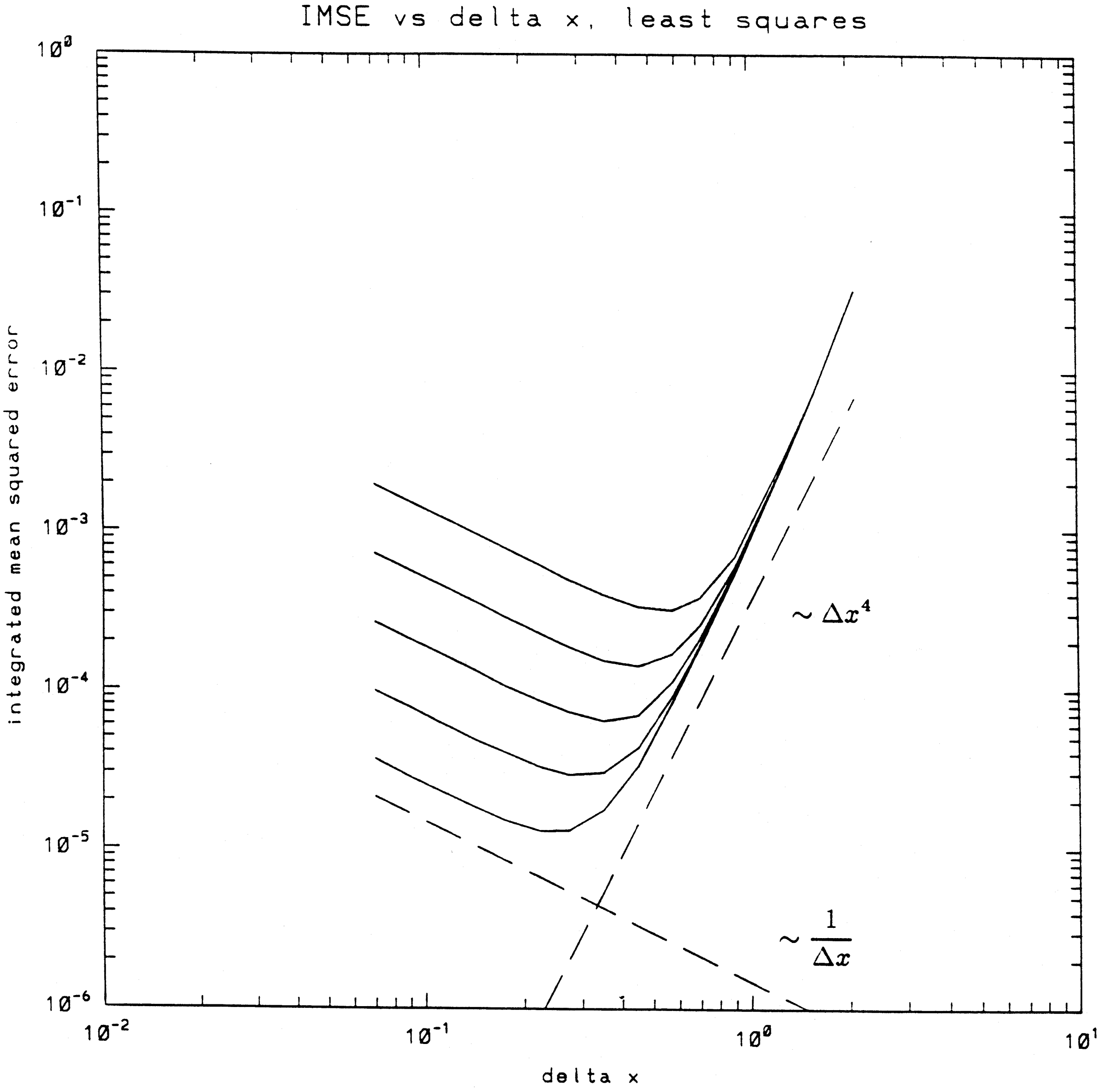

Figure 2: Error for the least squares method

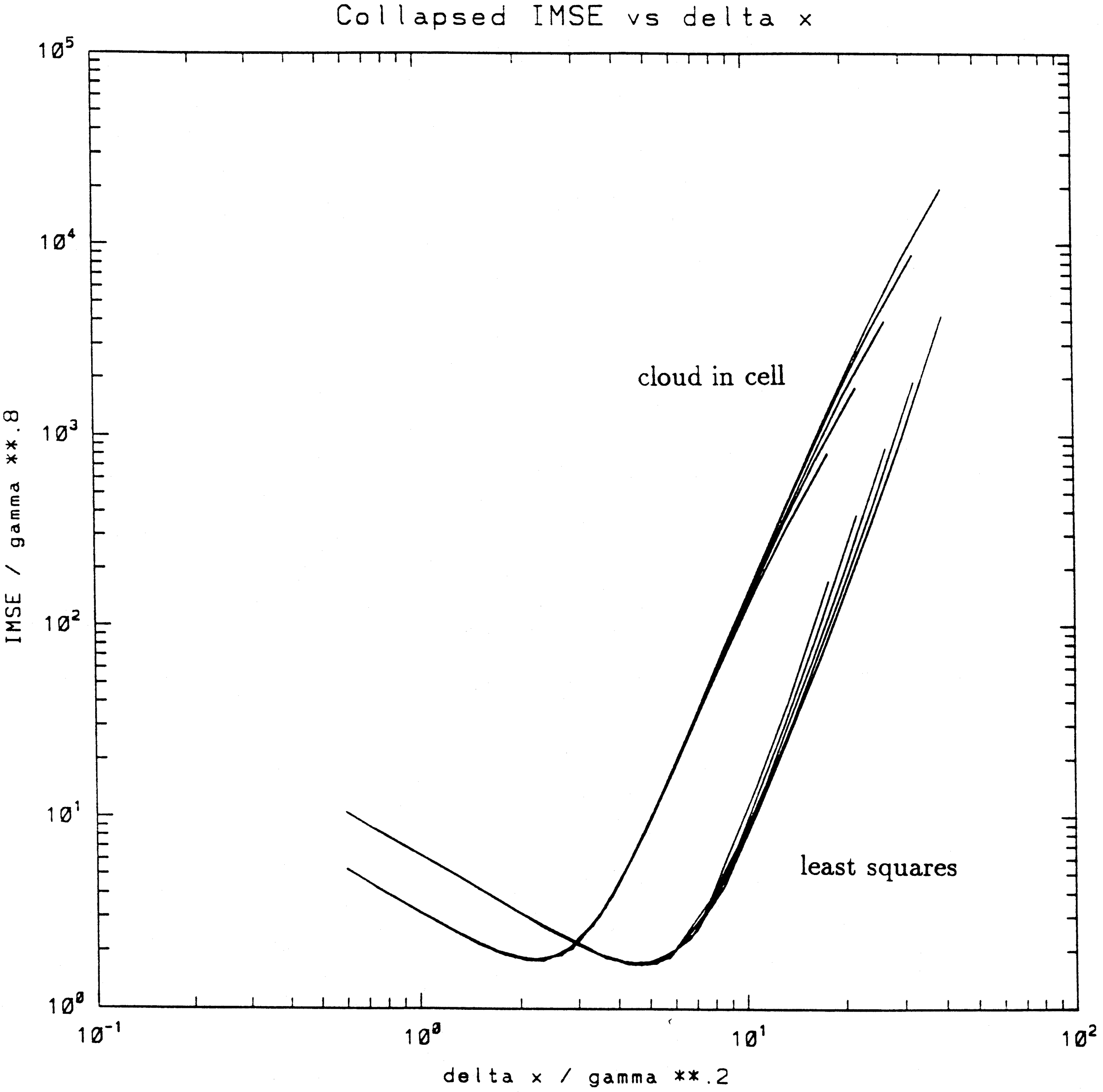

Figure 3: Collapsed error for cloud in cell and least squares methods

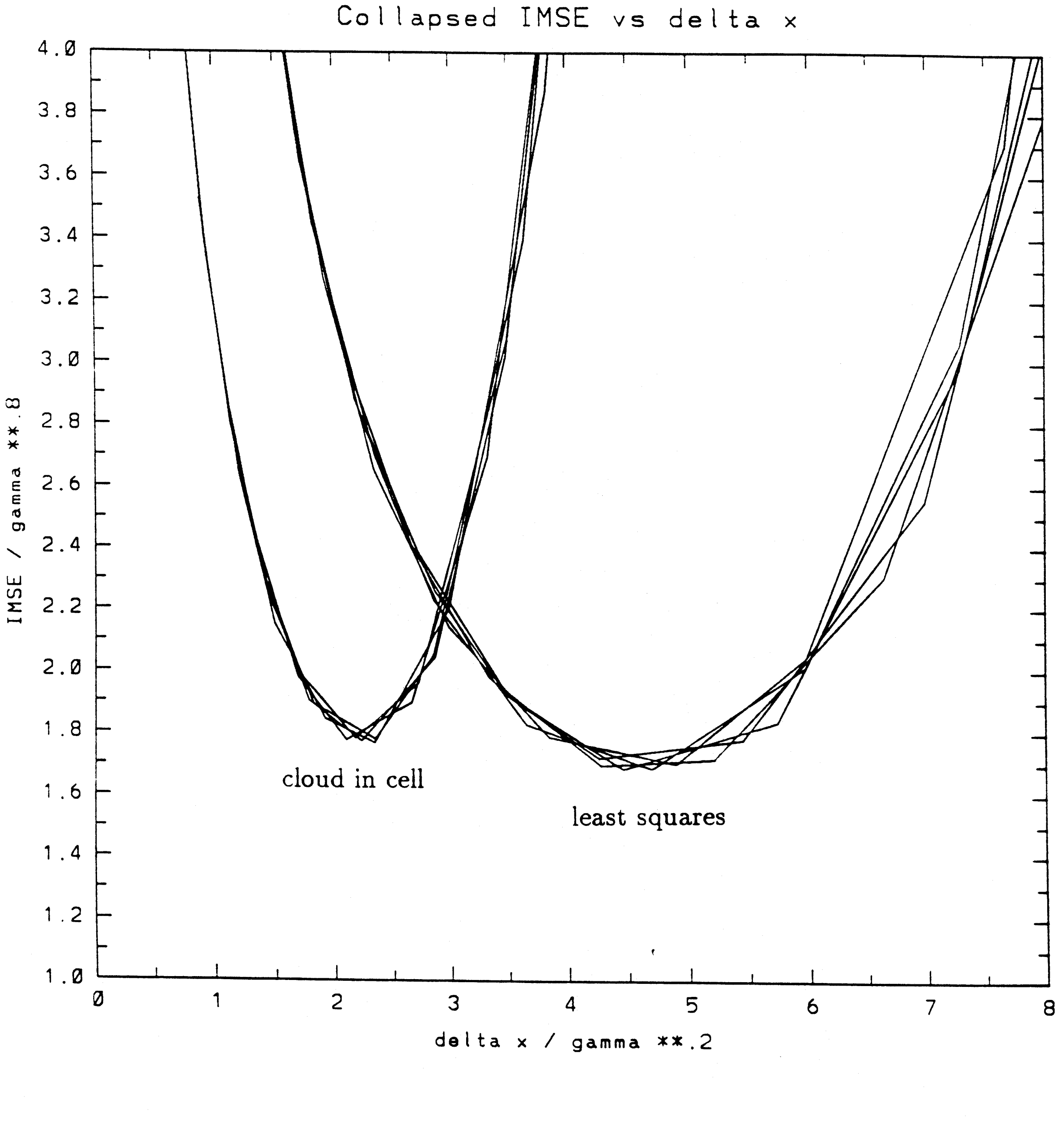

Figure 4: Collapsed error for cloud in cell and least squares methods

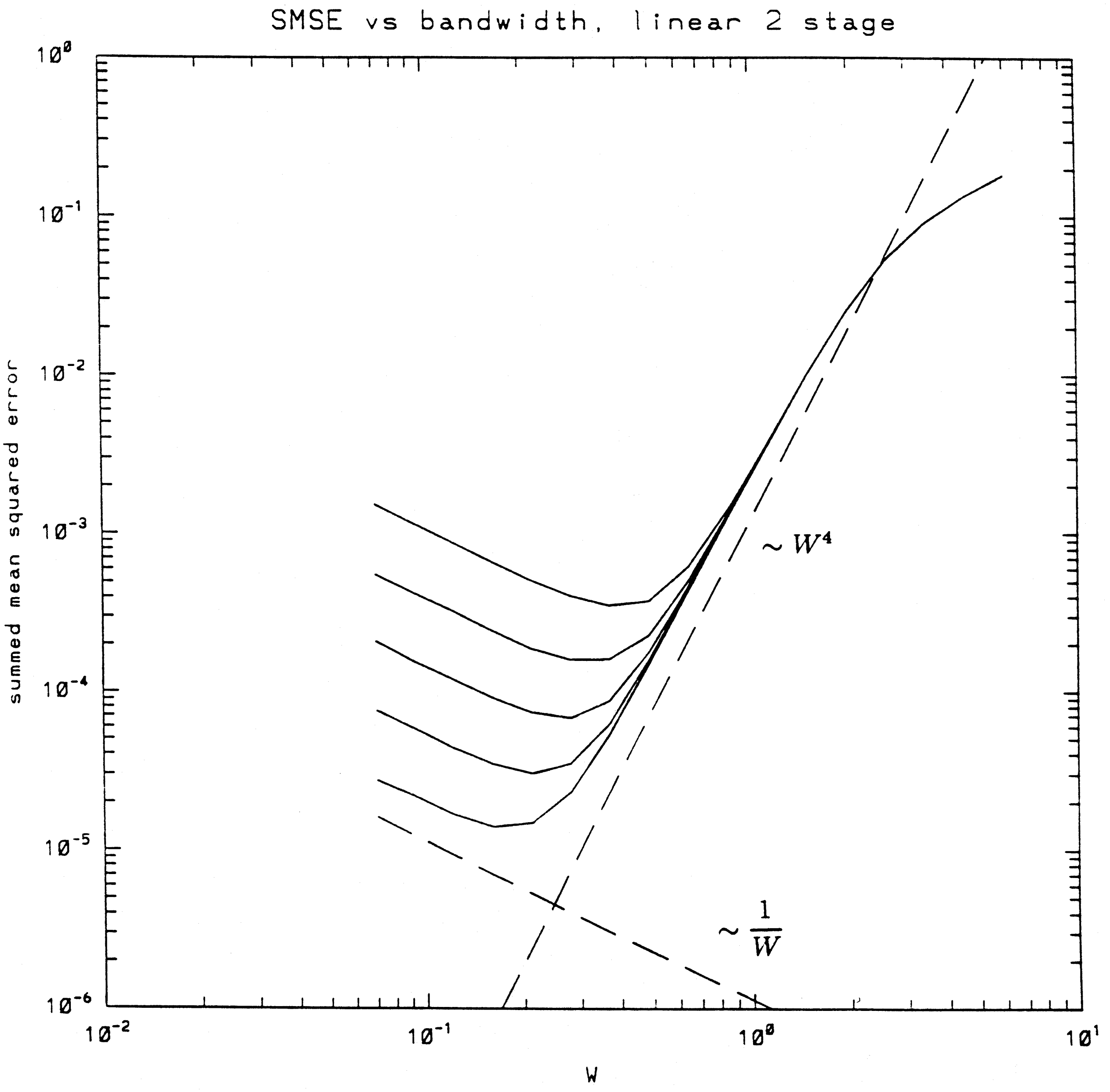

Figure 5: Error for the linear two stage method

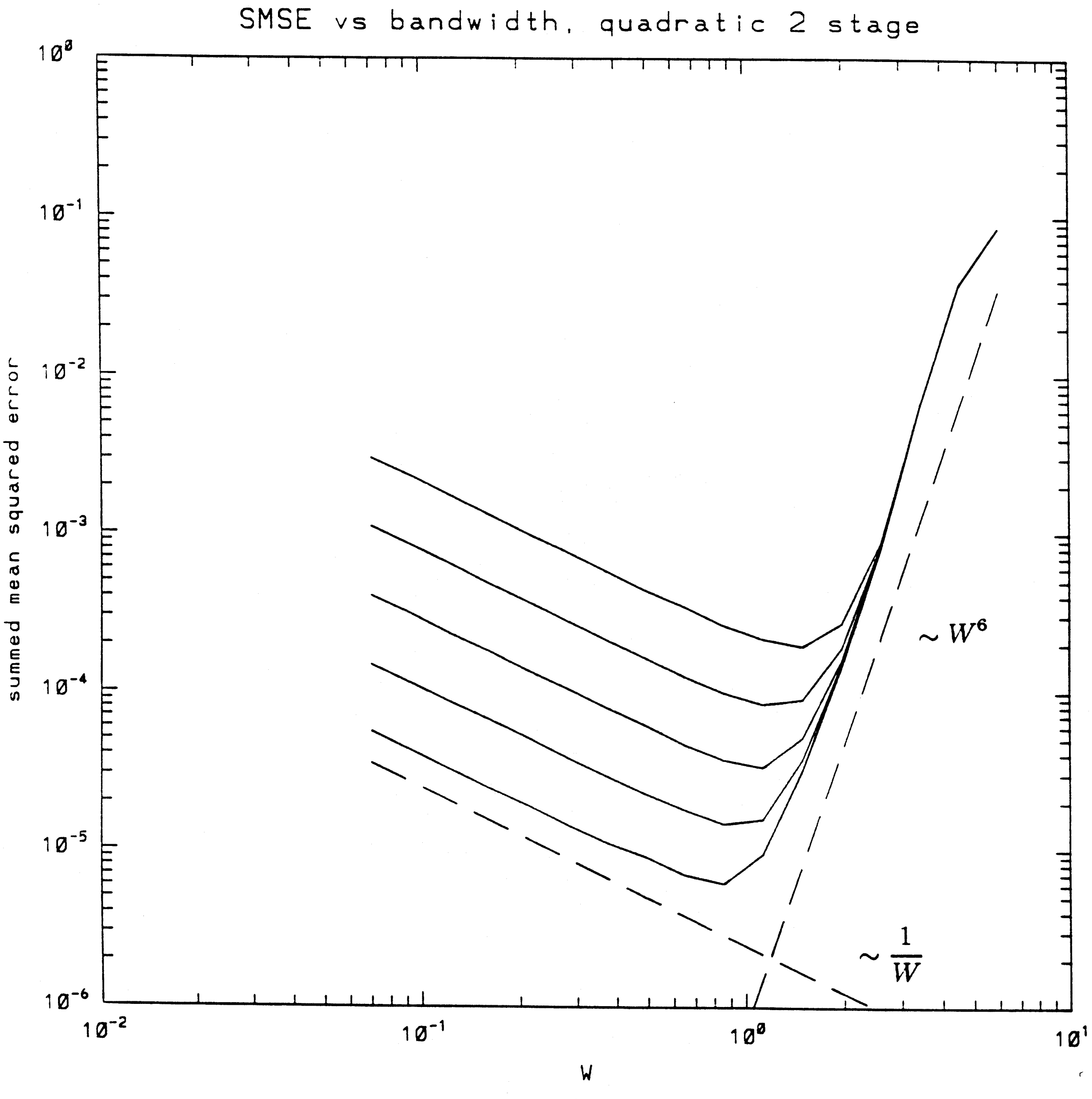

Figure 6: Error for the quadratic two stage method

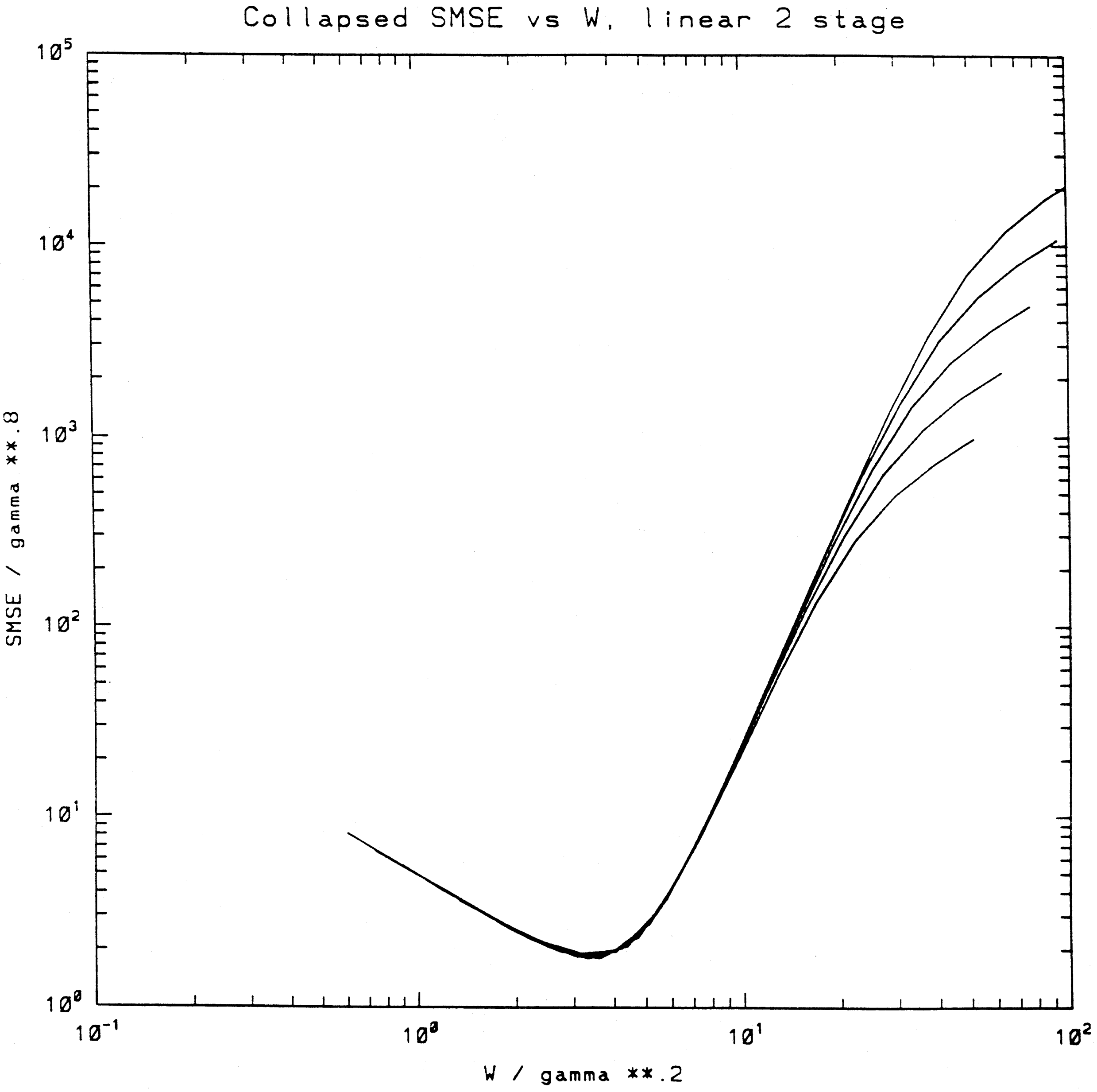

Figure 7: Collapsed error for the linear two stage method

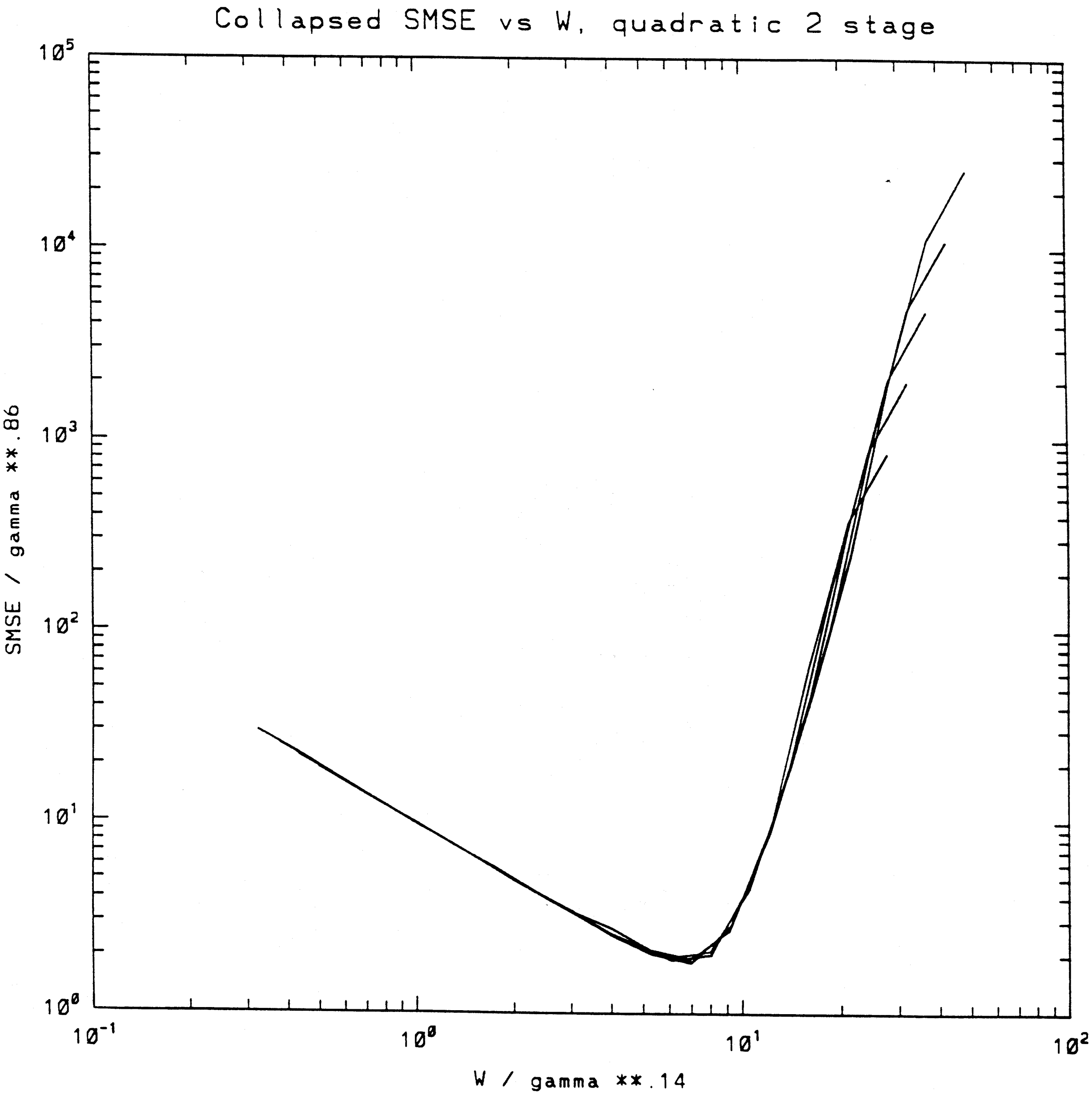

Figure 8: Collapsed error for the quadratic two stage method

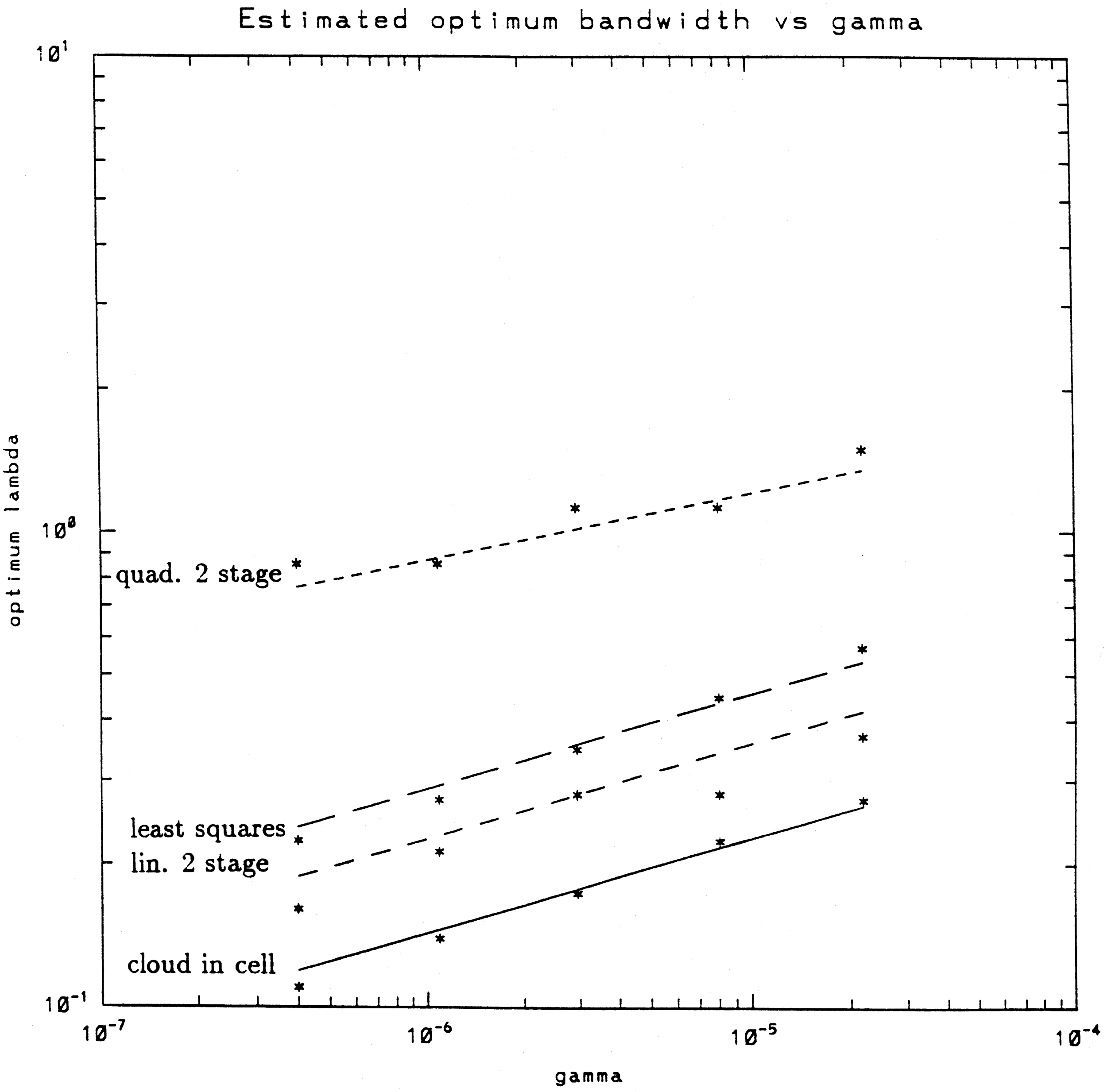

Figure 9: Optimum bandwidth

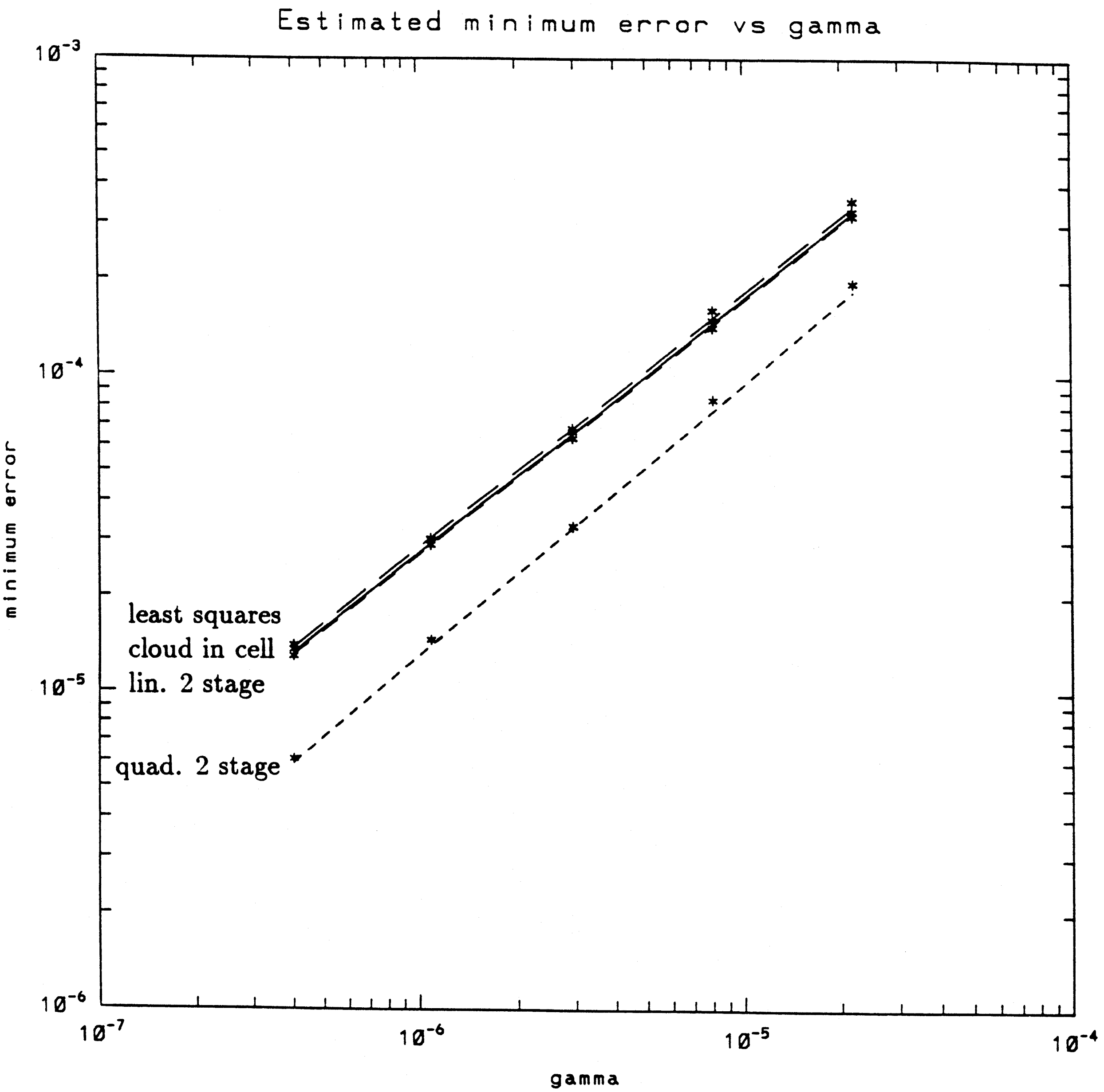

Figure 10: Minimum error

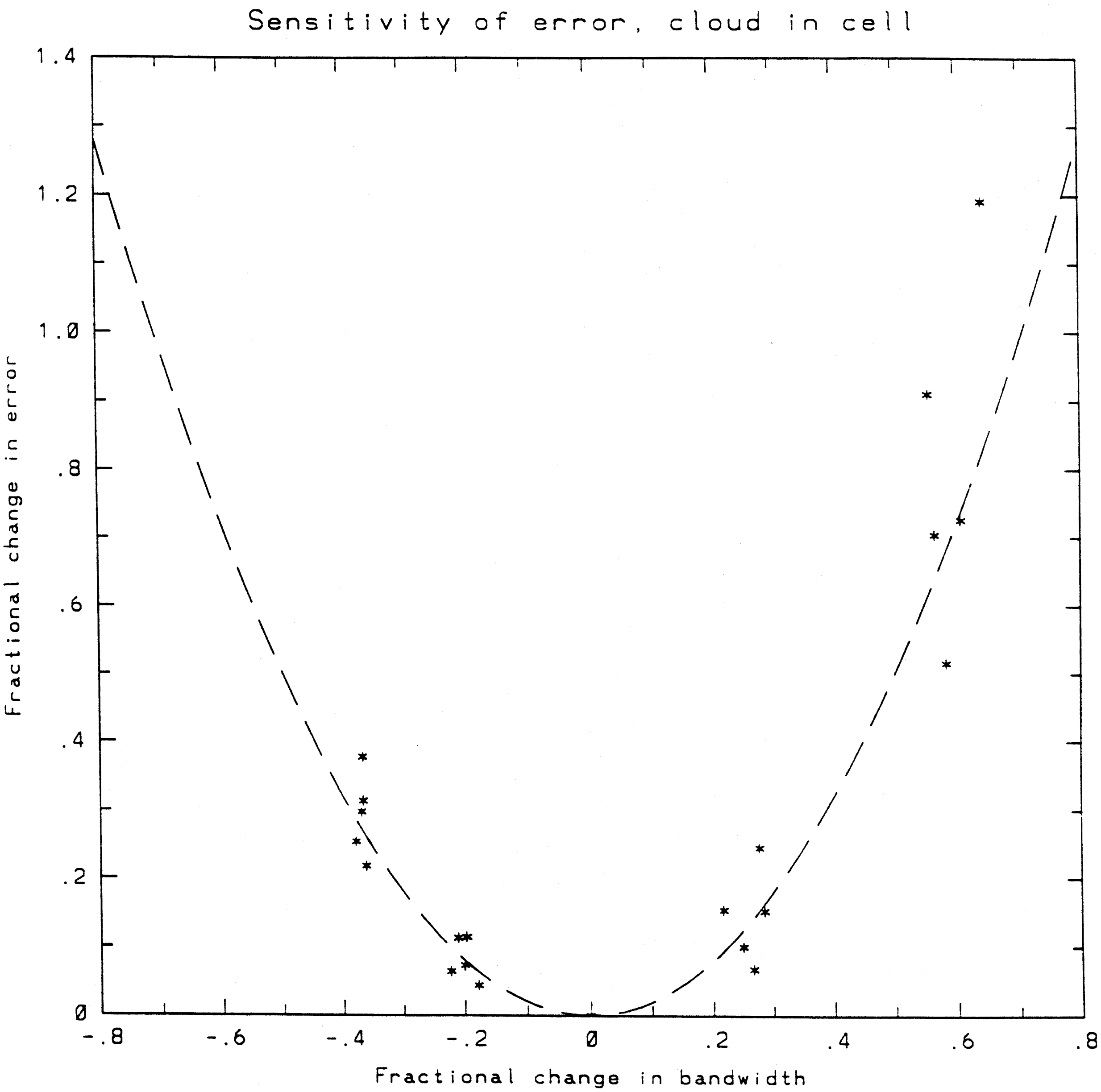

Figure 11: Error sensitivity, cloud in cell

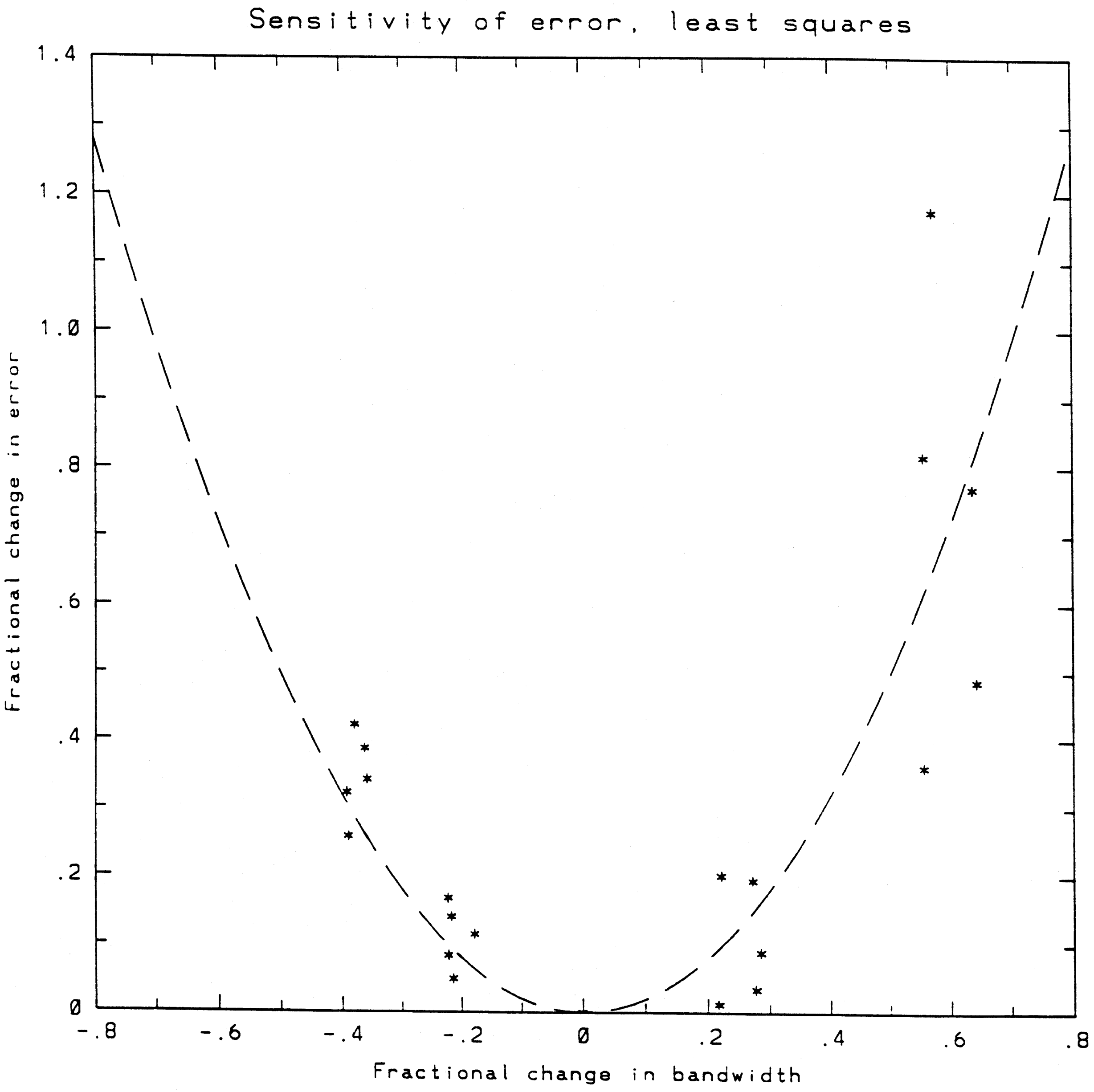

Figure 12: Error sensitivity, least squares

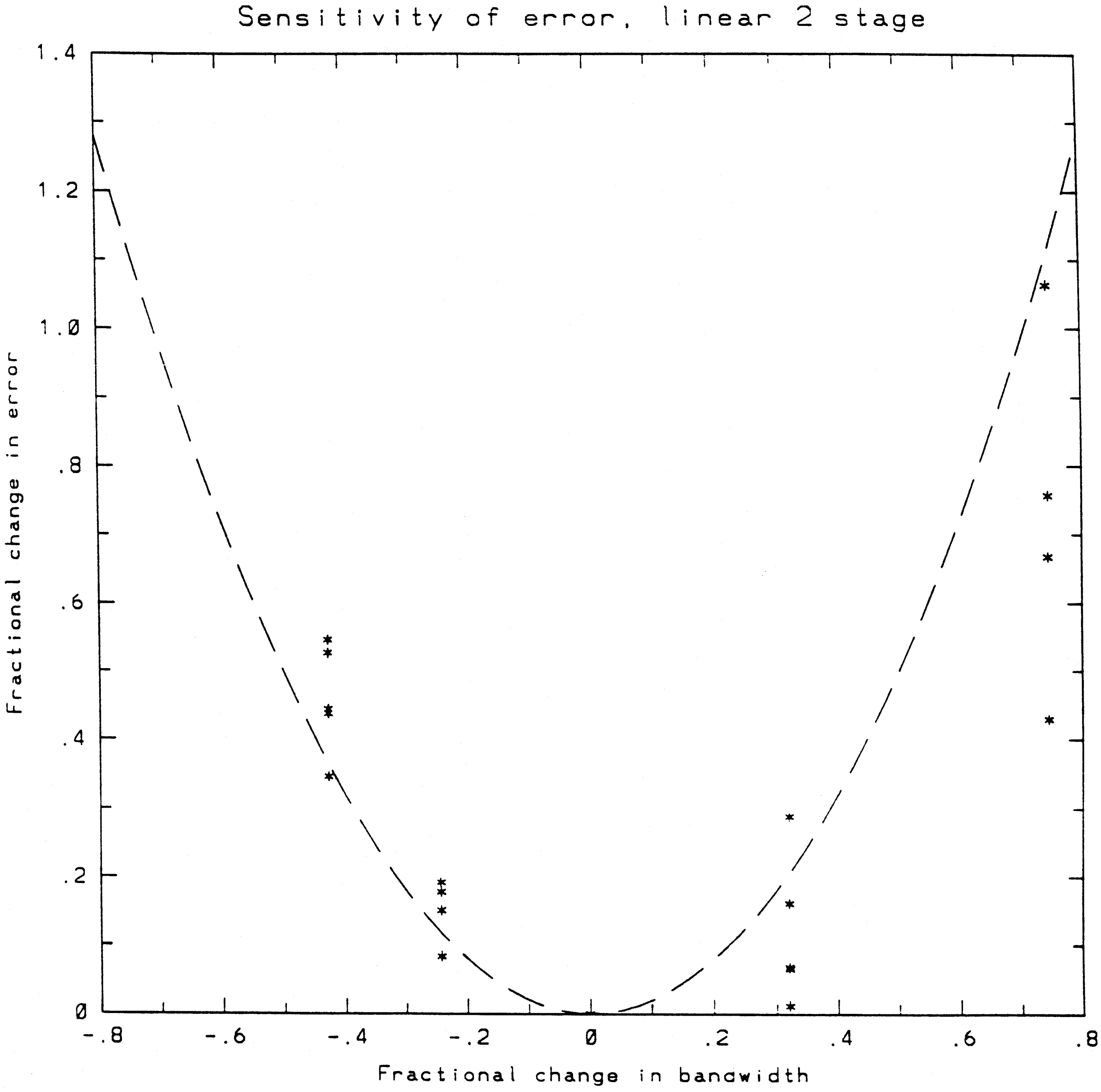

Figure 13: Error sensitivity, linear two stage

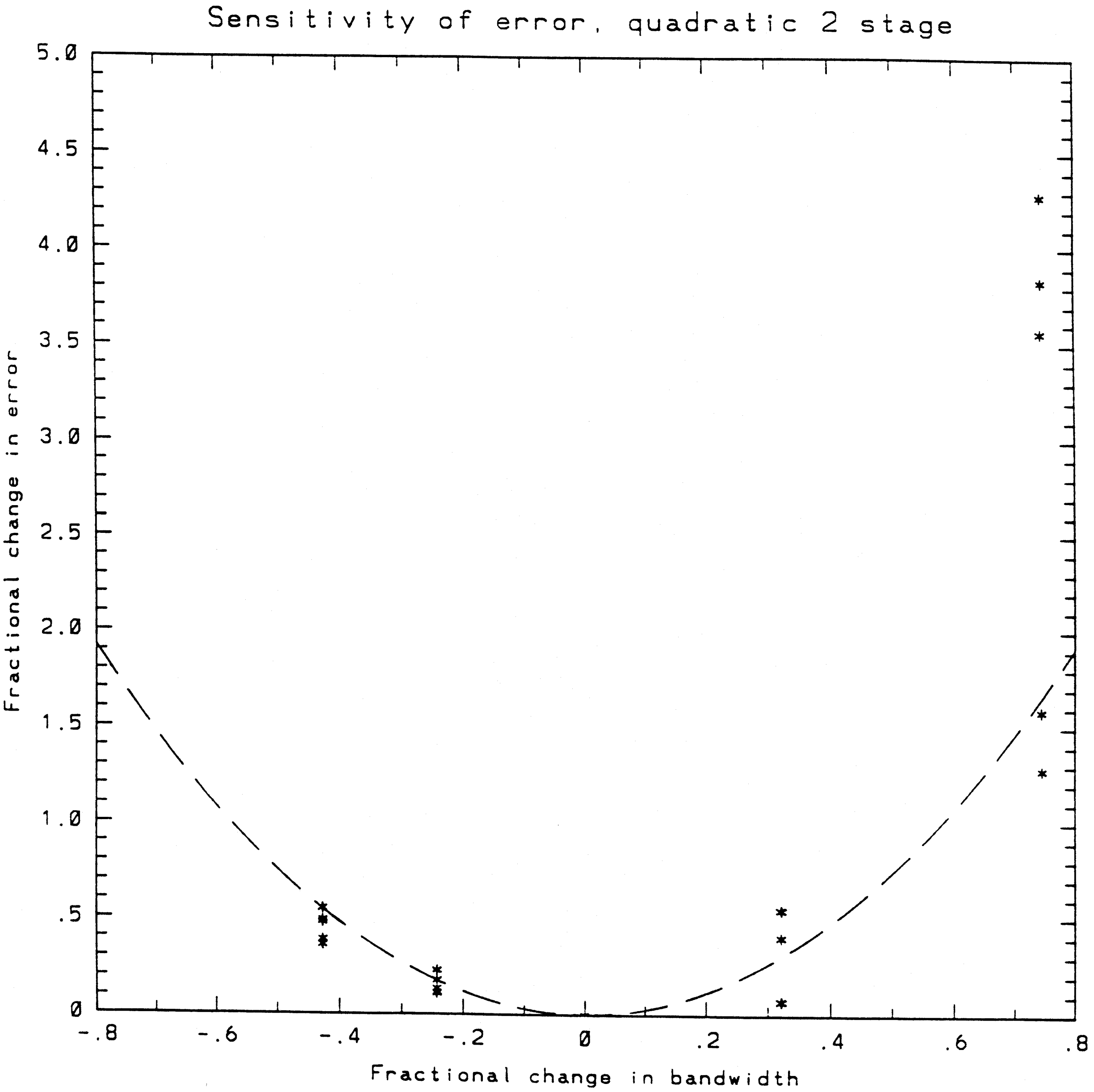

Figure 14: Error sensitivity, quadratic two stage