DEVICE MODELLING BY RADIAL BASIS FUNCTIONS

by

A. I. Mees, M. F. Jackson, and L. O. Chua

Memorandum No. UCB/ERL M90/70

10 August 1990

DEVICE MODELLING BY RADIAL BASIS FUNCTIONS

by

A. I. Mees, M. F. Jackson, and L. O. Chua

Memorandum No. UCB/ERL M90/70

10 August 1990

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

Device modelling by radial basis functions

A.I.Mees[1], M.F. Jackson[2] and L.O. Chua[3]

Abstract

Electronic devices are often modelled either by piecewise-linear functions or by empirically-derived combinations of nonlinear elementary functions. We present an alternative modelling technique which works directly from data and provides a straightforward and relatively automatic method of interpolating smoothly from measurements. Since the results are non-explicit (that is, the models are algorithmic rather than analytic), our approach is likely to be most appropriate in situations where it is difficult to derive an explicit functional form analogous to the Ebers-Moll equation for bipolar transistors. A good example is in the modelling of submicron devices in VLSI circuits, where the relevant device physics are currently poorly understood.

[1] A.I. Mees is with the Department of Mathematics, The University of Western Australia, Nedlands, Perth, Western Australia 6009.

[2] M.F. Jackson is with the Mathematics Department, Adelaide University, Adelaide, South Australia 5000.

[3] L.O. Chua is with the Electronics Research Laboratory, University of California, Berkeley, CA 94720.

1. Introduction.

Often in engineering it is necessary to treat one physical variable as a function of several others [1]. For example, a bipolar junction transistor is often modelled by treating the emitter current I_e and the collector current I_c as functions of the quantities base-collector voltage V_{bc} and base-emitter voltage V_{be}. The well-known Ebers-Moll model [2] gives an explicit functional form for I_e and I_c which are measured for different values of the applied voltages V_{bc} and V_{be} [3]. But in many device modelling problems it is not possible to determine such a function, either because the inherent nonlinearities can not be approximated accurately by elementary functions, or because the internal device physics is not well understood, as in many submicron devices.

If no suitable function is known then it is necessary to determine one by experiment, and the resulting model is called a *dc black box model* [3]. One way to do this is to observe the values of the dependent variable at certain fixed values of the independent variables, then use the observations in a model that estimates the dependent variable for all situations. Radial Basis Interpolation (RBI) is one method for doing this. There are others. One is piecewise-linear fitting [4, 5], which we shall refer to several times in this paper. Another involves making a local linear fit and then globally smoothing to give an approximate fit that is smooth and that is hoped to be good in spite of the disturbance caused by the separate smoothing process [6].

RBI is attractive for nonlinear device modelling via the black box approach for a number of reasons. Firstly, we shall demonstrate that it can be seen as a very natural generalization of the canonical piecewise-linear modelling process [5]. Secondly, it proves to be very flexible and powerful for a wide range of different kinds of relationships: unlike smoothed linear fit, for example, it is not restricted in the types of behaviour it can model, and it can give an exact fit at all points in some

desired set. Thirdly, it can be used to improve the fit of an existing model. Finally, it can be used in an automatic adaptive process which takes experimental data and makes a black-box model that will fit the data well and be cheap to calculate.

2. Radial Basis Interpolation.

Consider the determination of a dependent variable, y, from an independent variable, $x \in \mathbb{R}^n$, by an unknown function g, that is $y = g(x)$. We are going to treat g as scalar-valued since if the quantity in the problem is vector-valued, then we have $g(x) = (g_1(x), \ldots, g_n(x))$ where g_i are scalars, and we can consider each g_i individually without any loss of generality.

There are infinitely many ways to produce a function f that fits a finite amount of data, but for many purposes it is reasonable to restrict the class of models by making assumptions about smoothness. Even then, one has to choose some class of functions. It is well-known that global polynomial fits cause difficulties even with scalar problems, and one of the advantages of RBI is that it allows fits which are C^k smooth for any desired k, but which do not share the wildly oscillating nature of global polynomial approximations. Indeed, for some purposes, radial basis functions can be used for extrapolation as well as interpolation, although, like other interpolative methods, RBI is unlikely to be appropriate for extrapolation to large distances.

2.1 Standard radial basis interpolation

Suppose that, by experiment say, values $y_1, ..., y_m$ of y have been found at $x_1, ..., x_m$. Then we have $y_i = g(x_i)$ for $i = 1, \ldots, m$. The radial basis approximation to g is the function f defined by

$$f(x) = \sum_{i=1}^{m} \lambda_i \, \phi(|x - x_i|) \tag{1}$$

where the radial basis function ϕ is almost any scalar function of one variable we care to choose (for example, the identity: $\phi(r) = r$), and the λ_i's are computed so that all the known values fit exactly; that is, $y_i = f(x_i)$ for $i = 1, \ldots, m$.

Notice that in the case where ϕ is the identity function, equation (1) is

$$f(x) = \sum_{i=1}^{m} \lambda_i |x - x_i| \tag{2}$$

which when x is scalar is a slightly specialised version of the canonical piecewise-linear representation [7]. To come even closer to that model, we define the *affine plus radial basis interpolation (ARBI)* as

$$f(x) = \alpha.x + \beta + \sum_{i=1}^{m} \lambda_i \phi(|x - x_i|) \tag{3}$$

where $\alpha \in \mathbb{R}^n$ and $\beta \in \mathbb{R}$ are to be found along with $\lambda \in \mathbb{R}^m$. In the case where $n = 1$ and ϕ is the identity, (3) is exactly the canonical piecewise-linear form.

For the moment we shall concentrate on RBI, which is equivalent to choosing $\alpha = 0$ and $\beta = 0$ in ARBI. Later we shall discuss the determination of α and β in the more general model (3).

We require that the interpolation be exact at the known data points. Thus

$$y_i = \sum_{j=1}^{m} \lambda_j \phi(|x_i - x_j|) \text{ for } i = 1, \ldots, m \tag{4}$$

Writing the matrix Φ with elements

$$\Phi_{ij} = \phi(|x_i - x_j|)$$

we can rewrite (4) as m linear equations in m unknowns:

$$\Phi \lambda = y \tag{5}$$

where y and λ are the vectors with elements y_i and λ_i, $i = 1, \ldots, m$. Everything is known except $\lambda_1, \ldots, \lambda_m$. Solving (5) therefore determines f completely.

Notice that the radial basis approximation is linear in g. If

$$g(x) = \rho p(x) + \sigma q(x) + \tau$$

and we write p and q for the vectors with components $p(x_i)$ and $q(x_i)$, and 1 for the vector with all components equal to 1, then

$$\lambda(y) = \rho\lambda(p) + \sigma\lambda(q) + \tau\lambda(1),$$

where

$$\lambda(p) = \Phi^{-1}p$$

and so on.

With some choices of ϕ, RBI can be seen as an extension of natural splines to multivariable interpolation [8, 9, 10]. The choices $\phi(r) = r^k$ where k is an odd positive integer [8] and $\phi(r) = r^k \log(r)$ where k is an even positive integer [9, 10] are particularly natural with the latter corresponding to thin-plate spline fits when $x \in \mathbb{R}^2$. Not all choices of ϕ are appropriate, however: $\phi(r) = r^2$ is a bad one, at least when the norm $|x - x_i|$ in (1) is the Euclidean norm, since then it just makes a global quadratic fit.

We are going to use the spline-like interpolations. The effect of varying the exponent is to alter the "stiffness" of the interpolation. (We are using a metaphor here, since the physical idea of stiffness corresponds more closely to the size of a multiplier on the third derivative.) In general, the higher the exponent of r, the stiffer the fitted surface will be, in the sense that more derivatives are forced to vary slowly at the data points. The $\log(r)$ term can be regarded as giving a small increase in exponent. The case $\phi(r) = r$ in this sense is infinitely unstiff, and so not smooth at the data points. Thus $r^2 \log(r)$ is the least stiff function that is likely to work well and it is the one we shall usually employ.

Computationally the significant part of the problem is that of solving the linear equations (5) for λ. The size of the matrix Φ is the number m of data points and so the computational effort, which is of order m^3, may be large. Fortunately, this calculation is only performed once for a particular set of data points and ϕ. The work involved to interpolate for any given point is then considerably less, of order m.

As well as the difficulty of long computation time, there is a risk that as m grows, Φ will become ill-conditioned. Dyn and Levin [9] show that well-conditioned Φ results from choices of ϕ including

$$r^{2(k+1)}\log(r) \quad k \geq 0,$$

$$|r|^{2k+1} \quad k \geq 1,$$

$$(r^2+d^2)^{\pm 1/2} \quad 0 < d \ll 1.$$

In the latter case, d is often chosen to be of the order of the separation of the data points. Notice that any desired degree of smoothness can be obtained. For example, $r^k\log(r)$ is a C^{k-1} function of $r \geq 0$.

The choice $r^2\log(r)$ can give singular systems of equations in some (fairly contrived) situations, even in one dimension. This is because $r^2\log(r) = 0$ at $r = 1$ as well as at $r = 0$. The problem is unlikely to arise in practice, and in this respect at least, r^3 is probably completely safe.

For our purposes, we shall assume that the number of data points required is sufficiently small (up to a few hundred with current workstations) that numerical and computational difficulties do not dominate the problem. In Section 2.3 we shall discuss an automatic method of keeping under control the number of data points required by the interpolation, and therefore the computational effort, even when there are many measurements to fit.

2.2 Affine plus radial basis interpolation.

We shall see later that RBI on its own can perform relatively badly. Comparison with the one-dimensional canonical piecewise linear form suggests, as we remarked earlier, that linear and constant terms be added to the RBI function. Although it is important to realise that a set of data points does not on its own define a unique interpolating function, it is natural in the absence of any other information to require that constant and linear functions, at least, should be interpolated exactly, and that linear trends be identified. In making such a requirement we are imposing our own prejudices on the data, but since these prejudices are usually well-informed, this is a proper thing to do.

The naturalness of using ARBI rather than RBI is easily seen in the one dimensional case, where ARBI is a general piecewise linear fit while RBI is a piecewise linear fit which is constrained *outside* the fitting region. To prove this, observe that whenever $x > \max_{1\le i\le m} x_i$,

$$\begin{aligned} f(x) &= \sum_{i=1}^{m} \lambda_i |x - x_i| \\ &= \sum_{i=1}^{m} \lambda_i (x - x_i) \\ &= x \sum_{i=1}^{m} \lambda_i - \text{constant}. \end{aligned}$$

Similarly, whenever $x < \min_{1\le i\le m} x_i$,

$$f(x) = \text{constant} - x \sum_{i=1}^{m} \lambda_i.$$

That is, there is a symmetry in the RBI fit outside the range of the given points, so RBI certainly cannot fit a general piecewise-linear function. (And it will clearly be very bad at extrapolation in most cases.) It is easy to show that ARBI does not suffer from this deficiency. The point is that unadorned RBI tends to be best at

dealing with approximations of functions that are, in a suitable sense, symmetrical around the origin.

The parameters of an ARBI are not specified uniquely by requiring exact fit at the data points, since the addition of the $n+1$ parameters $\alpha_1, \ldots, \alpha_n$ and β means that there are more unknowns than equations.

One way to remove the indeterminacy is to find the best affine fit first and then fit a RBI model; this amounts to subtracting the best linear model before doing the radial basis interpolation. (The same can, of course, be done with *any* desired model; that is, any set of prejudices can be imposed at the start.) This approach is conceptually simple and may often be preferred, but it is possible to do better. We can require that the interpolation method be designed so that all affine functions will be fitted exactly, a reasonable requirement as we have already discussed.

Lemma *ARBI is exact for all functions of the form $p.x + q$ whenever α, β and λ are the unique solution to the equations*

$$y_i = \alpha.x_i + \beta + \sum_{j=1}^{m} \Phi_{ij}\lambda_j, \qquad i = 1, \ldots, m, \tag{6}$$

$$\sum_{i=1}^{m} \lambda_i = 0, \tag{7}$$

$$\sum_{i=1}^{m} \lambda_i x_i = 0. \tag{8}$$

Proof: Suppose $y_i = p.x_i + q$ for all i. Then $\alpha = p$, $\beta = q$, $\lambda = 0$ satisfies (6-8), and if (6-8) have a unique solution then solving them gives α, β and λ these values.

Remarks: Equations (6-8) specify $m+1+n$ linear equations, which can be solved for the $n+1+m$ unknowns in α, β and λ. The rank condition needed for uniqueness holds generically: that is, if it were false then almost all arbitrarily small perturbations of the basis points x_i and the data y_i would make it true.

2.3 Adaptive RBI and ARBI.

Often it will be possible to use all of the measured data in a RBI or ARBI model, but in many cases a more economical model will be useful, or even essential if using all of the data would give a matrix too large to invert in reasonable time. An automatic method of selecting a subset of the data is desirable. The idea is to choose some points from the original data in such a way that good approximations are obtained for the other points. We describe a simple approach here, for the RBI case. The ARBI case differs only in minor points of notation.

Begin with an RBI based on a small number of points, having indices in some set $\mathcal{R}$. (In the examples given later, we started with two diagonally opposite corner points, but in practice one might choose a larger initial set, chosen either randomly or from knowledge of important features of the system being modelled.) Let $f_{\mathcal{R}}$ be the radial basis interpolation function defined by the points in $\mathcal{R}$.

Examine the rest of the points and select the point which at this stage is being interpolated with the least accuracy. That is, find the index $i_0 \notin \mathcal{R}$ such that

$$|f_{\mathcal{R}}(x_{i_0}) - y_{i_0}| \geq |f_{\mathcal{R}}(x_i) - y_i|$$

for all $i \notin \mathcal{R}$. Bring the badly interpolated point x_{i_0} into $\mathcal{R}$

Using the new $f_{\mathcal{R}}$ so defined, re-examine the remaining points, and again select the worst interpolated to bring into $\mathcal{R}$. Continue this process until a predefined number of points have been used, or until no point is fitted with worse than some given accuracy: that is, the worst interpolated point satisfies $|f_{\mathcal{R}}(x_{i_0}) - y_{i_0}| < \epsilon$ for some given ϵ.

The addition of one point at a time need not involve re-inverting Φ at each stage since standard updating methods [11] can be used to make the calculation more efficient.

2.4 Best fit RBI

We now briefly discuss an approach due to Broomhead (private communication), who suggests using singular value decomposition to provide a best least squares fit to the data using some subset of the data points. It is the method of choice when the data is noisy. Although it could in principle be used for data reduction, it is not clear how to define an adaptive algorithm analogous to the one we have just described.

Suppose we have somehow selected a subset $\mathcal{R}$ of the data indices. We describe the building of a best-fit RBI model; an ARBI model could perhaps be built by subtracting off the best affine fit first, or simply by including the affine parameters α and β in the fitting calculation.

Using the independent variables x_i, $i \in \mathcal{R}$ as basis points, form the matrix $\Phi_{\mathcal{R}}$ with elements

$$\Phi_{\mathcal{R}ij} = \phi(|x_i - x_j|), \;\; i = 1, \ldots, m, \quad j \in \mathcal{R}.$$

Now solve the equations

$$\Phi_{\mathcal{R}} \lambda_{\mathcal{R}} = y \tag{9}$$

for $\lambda_{\mathcal{R}i}, i \in \mathcal{R}$. Assuming $\mathcal{R}$ is a strict subset of $1, \ldots, m$, there are more unknowns than equations. By calculating the pseudo-inverse of $\Phi_{\mathcal{R}}$ in the standard way by singular value decomposition [12], or in any other way, we obtain a least-squares solution $\hat{\lambda}_{\mathcal{R}}$ of (9).

Define the approximate interpolating function

$$f_{\mathcal{R}}(x) = \sum_{i \in \mathcal{R}} \hat{\lambda}_{\mathcal{R}i} \, \phi(|x - x_i|). \tag{10}$$

Equation (10) defines a function that fits the data in a best least squares sense for the given subset $\mathcal{R}$.

Selection of the subset $\mathcal{R}$ remains the main problem with this method. It may be possible to select suitable principal components by singular value decomposition of all of the data, but this assumes the data set is small enough to decompose, in which case it is certainly small enough for Φ to be inverted. Our recommendation is that this method be borne in mind for cases where the data is noisy; since this is not usually the case with circuit element data, we do not consider this technique further in the present paper.

3. Examples of RBI.

To be accepted by the electronic device modelling community, RBI will have to do well in some cases that can already be modelled satisfactorily. Here and later in this paper, we have taken some widely-used explicit models and used them as software circuit elements, to generate pseudo-experimental data for use by RBI. Then we have compared surface plots of the RBI results with surface plots of the original functions. Throughout we have used $\phi(r) = r^2 \log r$ unless stated otherwise.

Ebers-Moll model: complete fit

The Ebers-Moll equation for the emitter current I_e of a BJT is

$$I_e = (e^{V_{bc}/V_t} - 1)I_s - (e^{V_{bc}/V_t} - 1)I_s/\alpha_f.$$

Fig 1 shows this function when $I_s = 10^{-14}$, $V_t = 0.026$ and $\alpha_f = 0.99$. Here, the vertical coordinate denotes I_e, the horizontal coordinate in front denotes V_{bc}, and the horizontal coordinate at the back denotes V_{be}. Reconstruction of this surface was attempted using just the knowledge of a few points on it.

A 25 point grid was chosen to give more points in the rapidly varying regions: in fact, the grid was $I \times I$ where $I = \{0.4, 0.56, 0.72, 0.76, 0.8\}$. The result of RBI using this set is shown in Fig 2. This grid was chosen because it gave the RBI

a lot of information only where it was likely to be needed: that is, only where the function was far from being locally flat. Further accuracy could of course be obtained by using more data points. On the region shown in the graph, the greatest error is 0.043. With only 25 data points the RBI model is remarkably good, and there are no problems with either computational time or numerical errors. This would appear to be a practically feasible way of modelling a circuit element: take more observations in areas where the output is varying rapidly and fewer where it is varying slowly.

A piecewise-linear example

It might seem that cases where piecewise-linear functions are traditionally used will not be easy to manage using RBI, because RBI uses smooth functions. That this need not be so is shown in Fig 3a which shows a radial basis model of the function $f(x_1, x_2) = |x_1|$ on the set $[-1,1] \times [-1,1]$ using only 30 points. The points were chosen using the automatic method of Section 2.3, which selected the points shown in Fig 3b. The original 400 data points were from a grid $I \times I$ where $I = \{-1, -7/9, -5/9, \ldots, -1/9, 1/9, \ldots, 5/9, 7/9, 1\}$. Except very close to the fold the model is very good; close to the fold, there was never any data given to the model, and the real system is probably not modelled well by the piecewise-linear function in any case. It is remarkable how well RBI has performed. This is not to say that RBI is always more appropriate than piecewise-linear functions; the latter have advantages when one is interested in an explicit qualitative analysis of the circuit dynamics [13] and in many other problems, though as it happens, RBI is also extremely powerful in dynamical systems problems ([14]; see also Mees and Chua, in preparation).

Ebers-Moll model: automatic subset selection

Sometimes the shape of the function is totally unknown before interpolation, but many data points are available or can be measured easily. An interpolation fitting all of these points is desirable, but is generally computationally expensive.

To simulate this situation, values of the emitter current I_s were calculated from the Ebers-Moll equation on a grid of 400 data points. (Actually, this number of data points could be fitted completely on a workstation but our purpose here is to illustrate data compression.) The algorithm of Section 2.3 was applied to select subsets of the points.

Figs 4(a)-(c) show interpolations of the 400 point Ebers-Moll data based on subsets of 9,25 and 49 points respectively. The 49 point automatic interpolation has a maximum absolute error of 0.02 and a mean absolute error of 0.005.

MOSFET model

The Shichman-Hodges model [15] for a MOSFET is,

$$I_d = \begin{cases} k(V_{gs} - V_t - 0.5V_{ds})V_{ds} & \text{if } V_{gs} - V_t \geq V_{ds}, \\ 0.5k(V_{gs} - V_t)^2(1 + \lambda(V_{ds} - V_{gs} + V_t)) & \text{otherwise,} \end{cases}$$

which is plotted as Fig 5 for $k = 5 \times 10^{-5}, V_t = 1.0$ and $\lambda = 0.02$. Figs 6(a)-(c) show RBI approximations for 9, 25 and 49 points using the automatic version of the algorithm. The greatest error in the 49 point case is 2×10^{-7}, with a mean absolute error of 6×10^{-8}.

RBI versus ARBI

RBI can fit badly if the surface to be fitted lies far from the origin. Suppose we were interplating, say, $(I_d + 5)$, where I_d is from the Shichman-Hodges model. Because RBI is linear in the function being fitted, the result for a fixed set of basis points will just be the surface of Fig 5 added to the RBI fit for the constant function with value 5. Fig 7 shows an $r^2 \log r$ RBI approximation to this constant function, using 11 randomly selected points on the square $[-1, 1] \times [-1, 1]$. (This was deliberately selected to show exceptionally bad performance.)

The explanation for the poor performance of radial basis interpolation without affine functions is the same here as we discussed in Section 2.3 in the one-dimensional

case: the RBI model is incomplete since it enforces symmetry about the origin, and is also biased to functions which are zero there. In the present case, for large $|x|$ in the RBI model

$$f(x) \approx \left(\sum_{i=1}^{m} \lambda_i \right) \phi(|x|).$$

Since we have been using basis functions ϕ which are increasing, $f(x)$ will tend either to $+\infty$ or to $-\infty$ depending on the sign of $\sum \lambda_i$.

Think of the RBI surface as a flexible sheet fixed at a number of points. If the surface is fixed as $|r| \to \infty$, then moving all of the points up will stretch it in a way that may have little to do with the function that is being approximated. This is very clear in Fig 7. It happens that there, $\sum \lambda_i < 0$, so the surface is fixed at $-\infty$ as can be seen from the picture. The cure would be to make $\sum \lambda_i = 0$ which is exactly condition (7) in the definition of the ARBI model. This makes the surface flat at infinity. In the same way, condition (8) allow the surface to tilt if necessary.

We conclude that if increasing basis functions are being used, ARBI should always be used in preference to RBI; at worst, it can do no harm beyond slightly increasing the computational effort. (It could be argued that one should instead use decreasing radial basis functions, but these have their own problems, and in our experience require many more points to fit the sort of functions we are considering in this paper.)

Volterra kernel

Fig 8 shows the graph for a 2^{nd}-Order Volterra kernel given by

$$|H_2(x_1, x_2)| = |x_1 + x_2| / \sqrt{(10^6 + x_1^2)(10^6 + x_2^2)},$$

and interpolations using the above algorithm are shown in Figs 9(a)-(c) for 9, 25 and 49 points. The greatest error in the 49 point case is 5×10^{-5}, with a mean absolute error of 10^{-5}.

4. Discussion

The examples make it clear that ARBI is a useful method for modelling nonlinear functions for which only experimental data is available, or for which data is available and there is a model which provides a partial fit.

The first thing to bear in mind is that any modelling method which takes all of its knowledge of the function from the data points requires there to be enough data points to represent all features of interest.

If there is already some favoured model, it can be subtracted off or factored out before the process begins, as was done in the discussion of best affine fit. In that case, far fewer data points are likely to be required and the computational effort will be enormously diminished.

If there are more than 2 or 3 parameters, a check should first be made to discover whether some of them have simple roles, for example as scale factors. This is not because ARBI is inherently incapable of dealing with such cases, but is to reduce the need for a very large number of measurements and the consequent need for a large radial basis model. For example, an additive constant will be dealt with correctly and completely by the affine fit stage but if the additivity is not recognised by the experimenter, many unnecessary measurements may be made. Similarly, an output scale factor will also be dealt with by the affine fit, but once more at the expense of a requirement for many measurements: had the experimenter used logarithmic plots to discover that the parameter was a scale factor, there would have been a reduction in the dimension of the space of measurements. In other words, good interpolation methods do not do away with the need for good science.

Whether or not other simplifications have been achieved, it may often be possible to choose to make measurements more densely in regions of interest and less densely elsewhere; in such cases, a direct fit to all of the data will probably suffice.

If it is necessary to use a method that is as automatic as possible, the adaptive procedure of Section 2.3 is recommended, *after* possible subtraction of a preferred model and identification and removal of variables with simple roles such as scaling. It assumes that enough measurements can be made to give information about all important features. In the case of noisy data the method of Section 2.4 is likely to be better, but has the disadvantage that there is no very clear way to choose a subset of points efficiently.

Finally, to put things in perspective, it may be useful to compare the pros and cons of ARBI with the canonical piecewise-linear representation and its recent generalization [4]. The main advantage of the latter seems to be in situations where it is advantageous to exploit the linearity property in qualitative analysis, and in piecewise-linear circuit equilibrium analysis where many useful techniques and algorithms are available, including one for finding *all* equilibrium solutions via combinatorial techniques [5, 16, 17]. No other presently available representation can make such guarantees.

The ARBI method is, however, more general in certain respects, since a piecewise linear fit to a surface will involve many components if the surface has a complex shape, and combinatorial methods will eventually fail because of their exponential dependence on the number of segments. As we have pointed out, the main uses of ARBI are likely to be in cases where the device physics is poorly understood, where a black box model is satisfactory, and where either the extensive investigation needed for a full piecewise-linear analysis is too costly, or the system does not yield to piecewise-linear representation.

Acknowledgements

AIM wishes to thank King's College, Cambridge, The University of California and The University of Arizona for their hospitality, and all of these institutions together with The University of Western Australia and ONR grant N00014-14-86K-0351 for financial aid. AIM also wishes to thank M.J. Powell for introducing him to the possibilities of RBI. MFJ wishes to thank the Mathematics Department of the University of Western Australia for a Vacation Scholarship. The surface plots were all made with Mathematica.

Figure Captions

Fig 1 Surface defined by the Ebers-Moll model of a BJT.

Fig 2 Radial basis approximation to the BJT surface of Fig 1 using the basis points defined in the text, chosen so that more data is available near the back and right edges where the function is varying more rapidly.

Fig 3 Radial basis approximation to a piecewise-linear function. The points were chosen from a 20×20 grid by the adaptive algorithm of Section 2.3. (a) The approximation to the surface. (b) The subset of points selected by the algorithm from the uniform grid. Note that the algorithm had no information very close to the fold and has made a smooth approximation there; the important feature is that the flat sections have been approximated well.

Fig 4 Radial basis approximation to the BJT surface of Fig 1 using the adaptive model, choosing data from a 20×20 grid. (a) 9 point subset; (b) 25 point subset; (c) 49 point subset.

Fig 5 Surface defined by Shichman-Hodges model of a MOSFET.

Fig 6 Radial basis approximations to the MOSFET surface of Fig 5 using the adaptive algorithm and a 20×20 grid. (a) 9 point subset; (b) 25 point subset; (c) 49 point subset.

Fig 7 Poor approximation by unmodified radial basis algorithm without affine part. Shown is an RBI fit to the constant function with value 5, using 11 randomly selected basis points. The resulting surface may be regarded as a flexible sheet fixed at $-\infty$ at its edges, and stretched over the data points.

Fig 8 Surface defined by a Volterra kernel.

Fig 9 Adaptive radial basis fits to the Volterra surface of Fig 8. (a) 9 point subset; (b) 25 point subset; (c) 49 point subset.

References

[1] L. O. Chua, "Device modelling via basic nonlinear circuit elements," *IEEE Transactions on Circuits and Systems,* vol. CAS-27, pp. 1059-1087, 1980.

[2] J. J. Ebers and J. L. Moll, "Large-signal behaviour of junction transistors," *Proceedings of the IRE,* vol. 42, pp. 1761-1772, 1954.

[3] L. O. Chua and P. M. Lin, *Computer-aided analysis of Electronic Circuits: Algorithms and Computational Techniques,* Englewood Cliffs, New Jersey: Prentice Hall, 1975.

[4] C. Kahlert and L. O. Chua, "A generalized canonical piecewise-linear representation," *IEEE Transactions on Circuits and Systems,* vol. CAS-37 no. 3, pp. 373-383, 1990.

[5] L. O. Chua and A. C. Deng, "Canonical piecewise-linear analysis: generalized breakpoint hopping algorithm," *International Journal of Circuit Theory and Applications,* vol. 14, pp. 35-52, 1986.

[6] P. B. L. Meijer, "Fast and smooth highly nonlinear multidimensional table models for device modelling," *IEEE Transactions on Circuits and Systems,* vol. CAS-37 no. 3, pp. 335-346, 1990.

[7] L. O. Chua and S. M. Kang, "Section-wise piecewise-linear functions: canonical representation, properties and applications," *Proceedings of the IEEE,* vol. 65, pp. 915-929, 1977.

[8] M. J. D. Powell, *Radial basis functions for multivariable interpolation: a review,* Report 1985/NA12, Department of Applied Mathematics and Theoretical Physics, Cambridge University, 1985.

[9] N. Dyn and D. Levin, "Iterative solutions of systems originating from integral equations and surface interpolation," *SIAM Journal of Numerical Analysis,* vol. 20 no. 2, pp. 377-390, 1983.

[10] N. Dyn, D. Levin and S. Rippa, "Numerical procedures for surface fitting of scattered data by radial functions," *SIAM Journal of Scientific and Statistical Computing,* vol. 7 no. 2, pp. 639-659, 1986.

[11] P. E. Gill, W. Murray and M. H. Wright, *Practical Optimization,* New York: Academic Press, 1981.

[12] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes: the art of scientific computing,* Cambridge: Cambridge University Press, 1986.

[13] L. O. Chua, M. Komuro and T. Matsumoto, "The double scroll family," *IEEE Transactions on Circuits and Systems,* vol. CAS-33, pp. 1073-1118, 1986.

[14] M. Casdagli, "Nonlinear prediction of chaotic time series," *Physica D,* vol.

35 no. 3, pp. 335-356, 1989.

[15] H. Shichman and D. A. Hodges, "Modelling and simulation of insulated-gate field-effect transistor switching circuits," *IEEE Journal on Solid State Circuits,* vol. SC-3, pp. 285-289, 1968.

[16] L. O. Chua and R. Ying, "Canonical piecewise-linear analysis," *IEEE Transactions on Circuits and Systems,* vol. CAS-30, pp. 125-140, 1983.

[17] L. O. Chua and A. C. Deng, "Canonical piecewise-linear analysis, Part II: Tracing driving-point and transfer characteristics," *IEEE Transactions on Circuits and Systems,* vol. CAS-32, pp. 417-444, 1985.

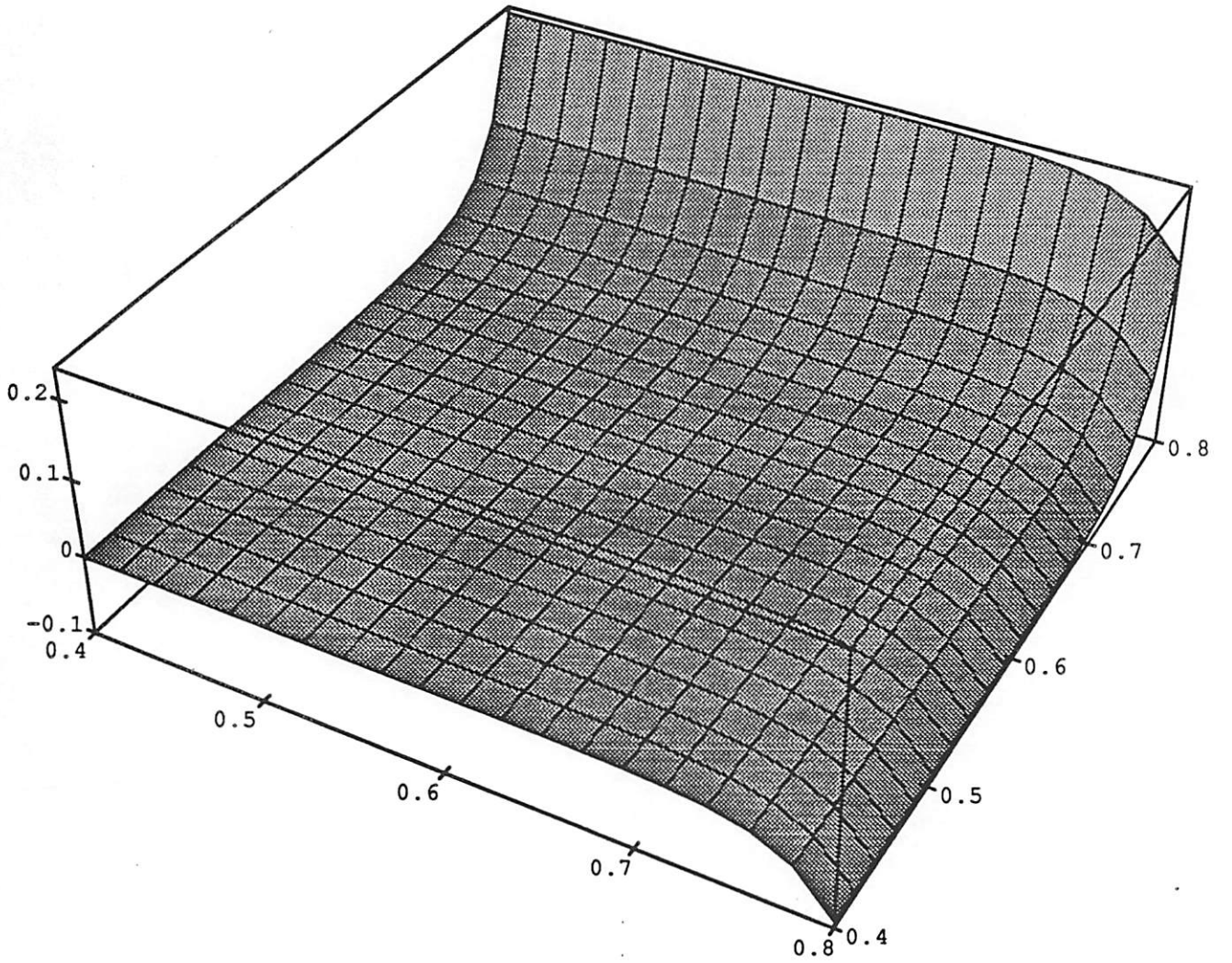

Figure 1

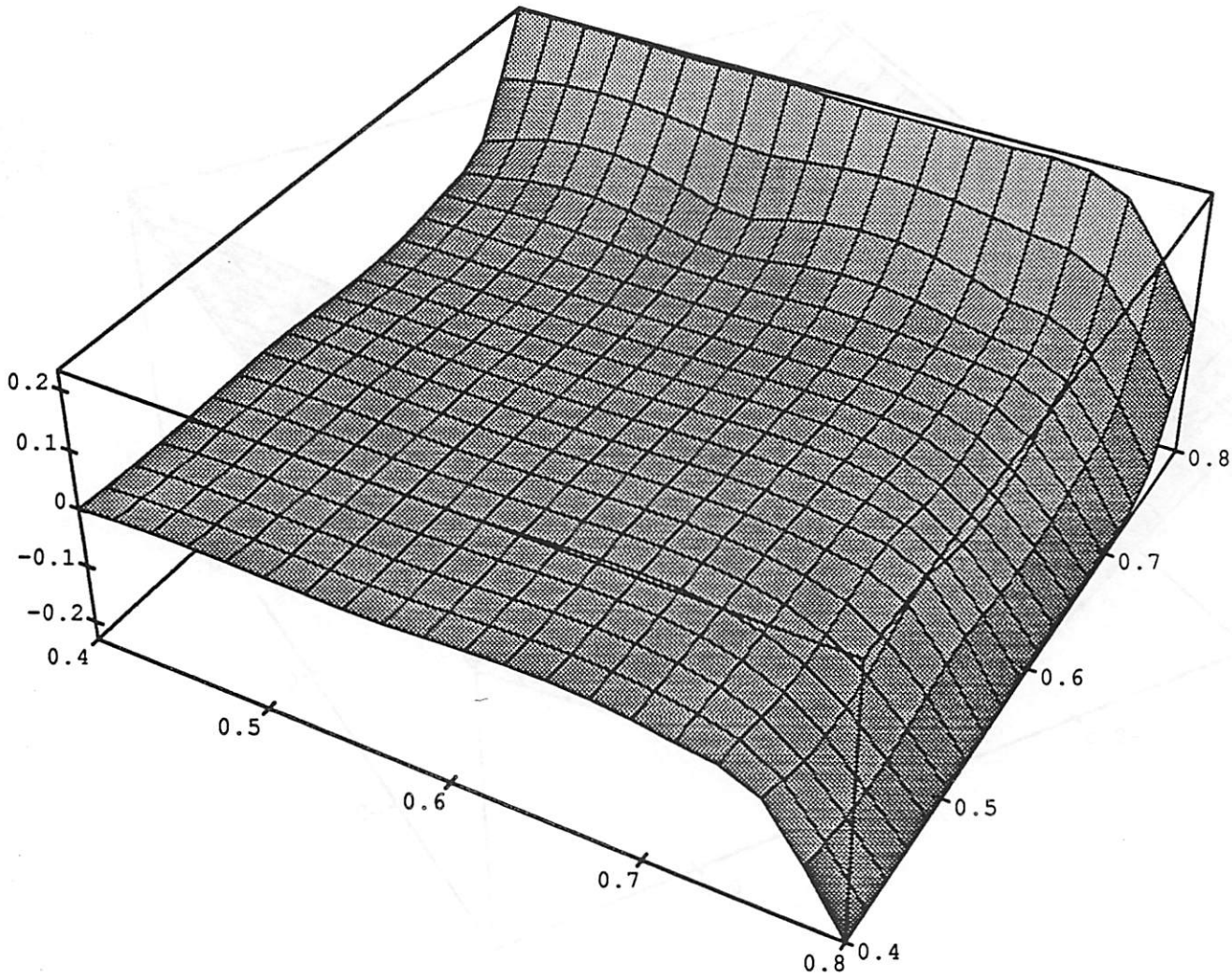

Figure 2

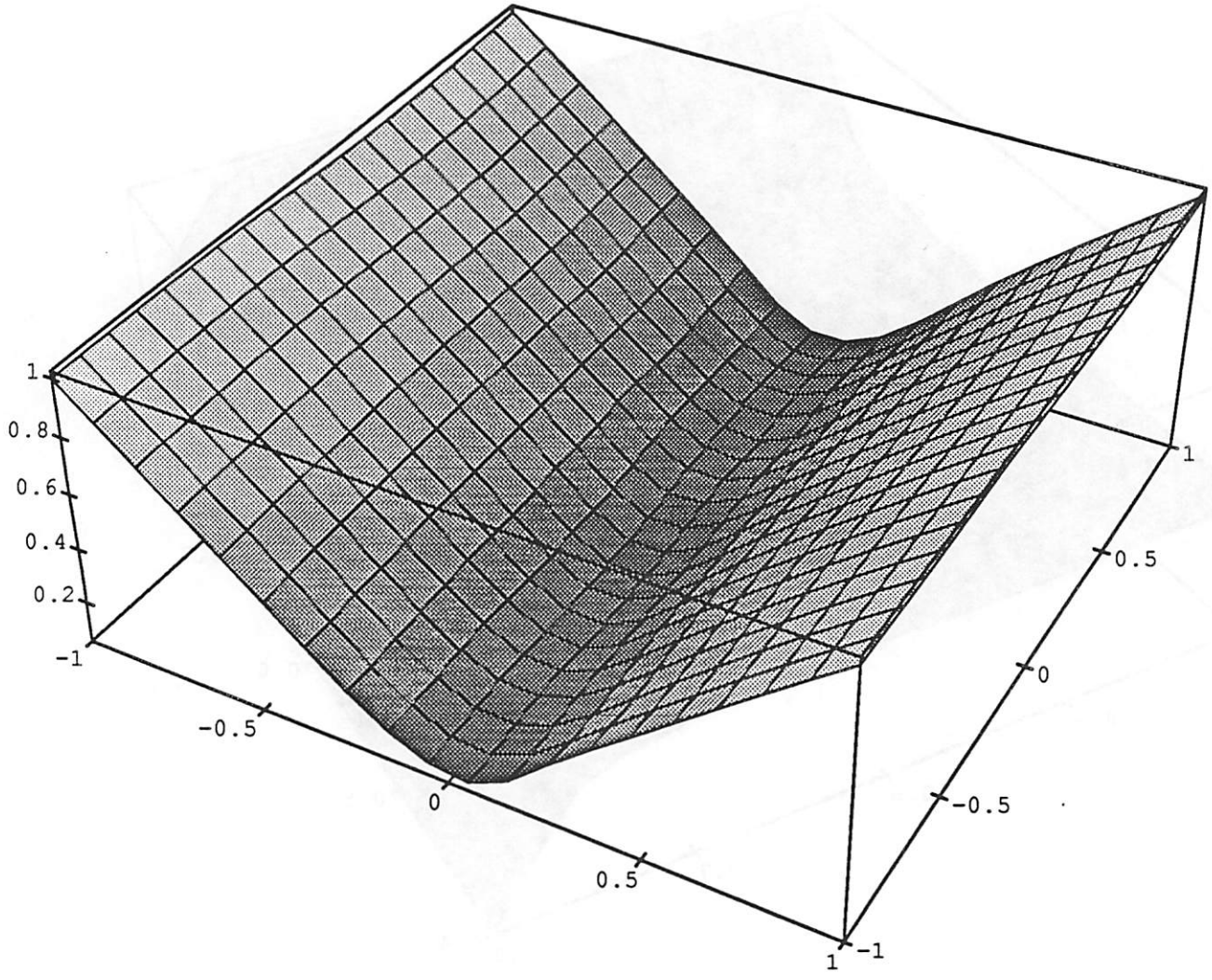

figure 3 (a)

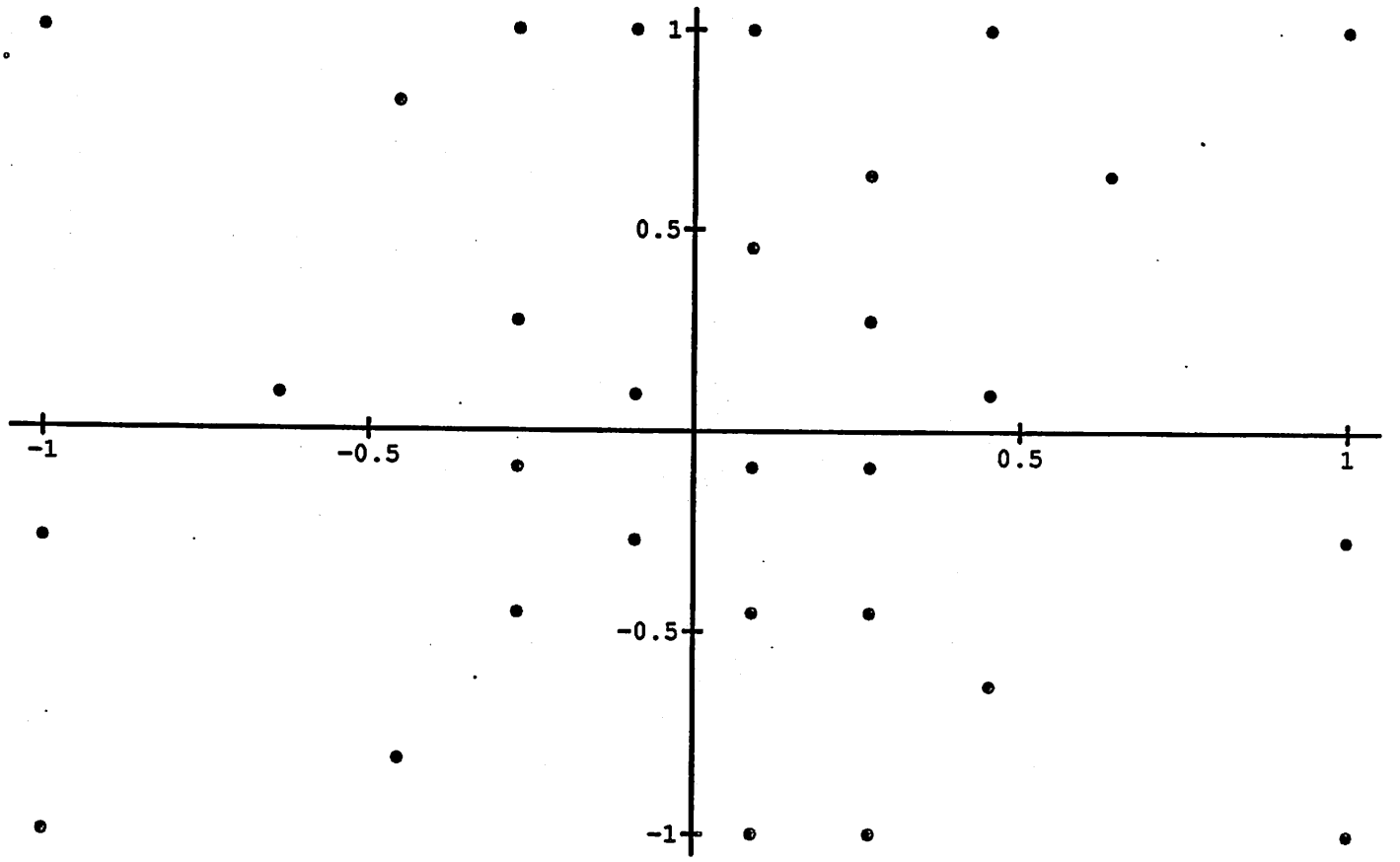

Figure 3 (b)

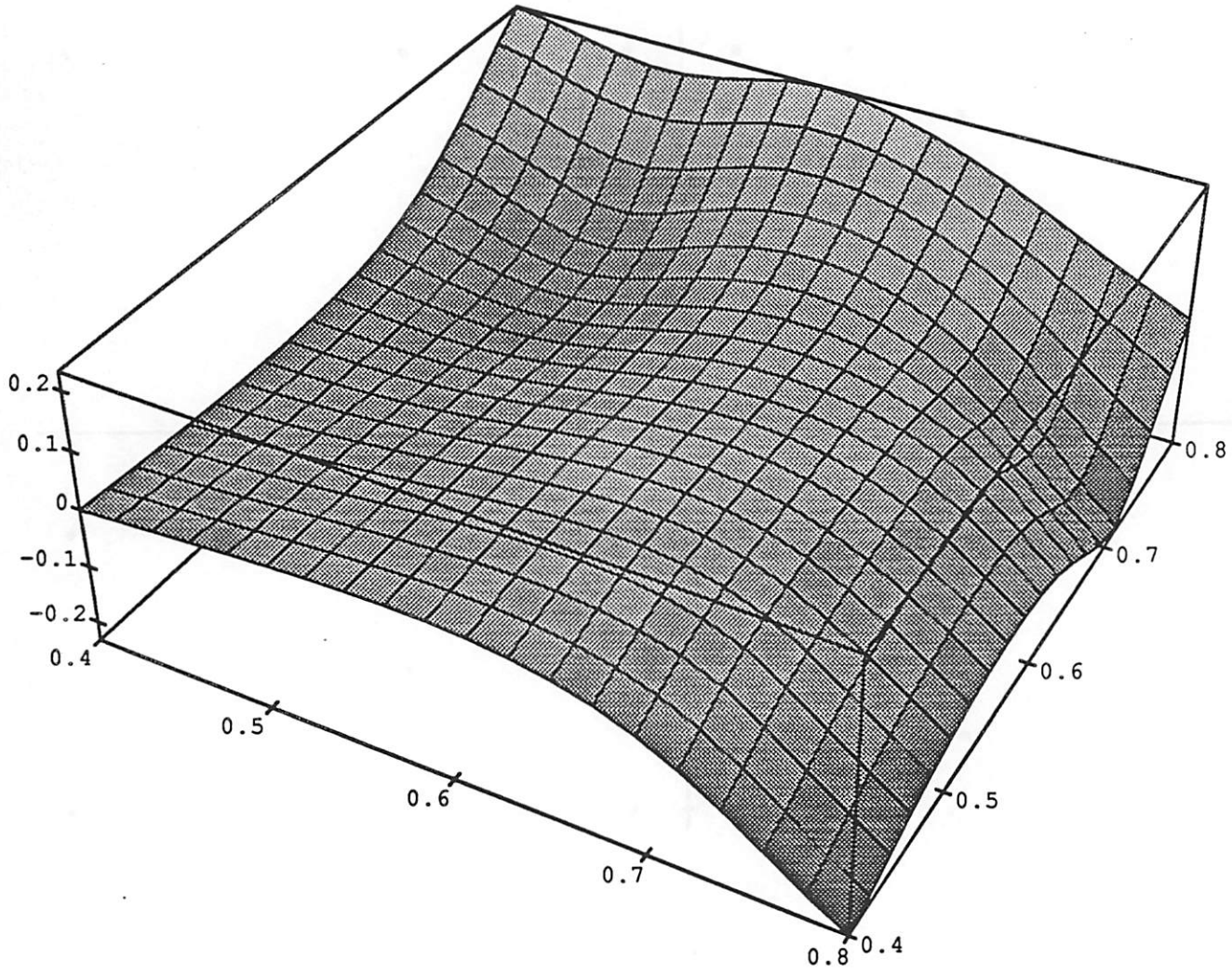

Figure 4 (a)

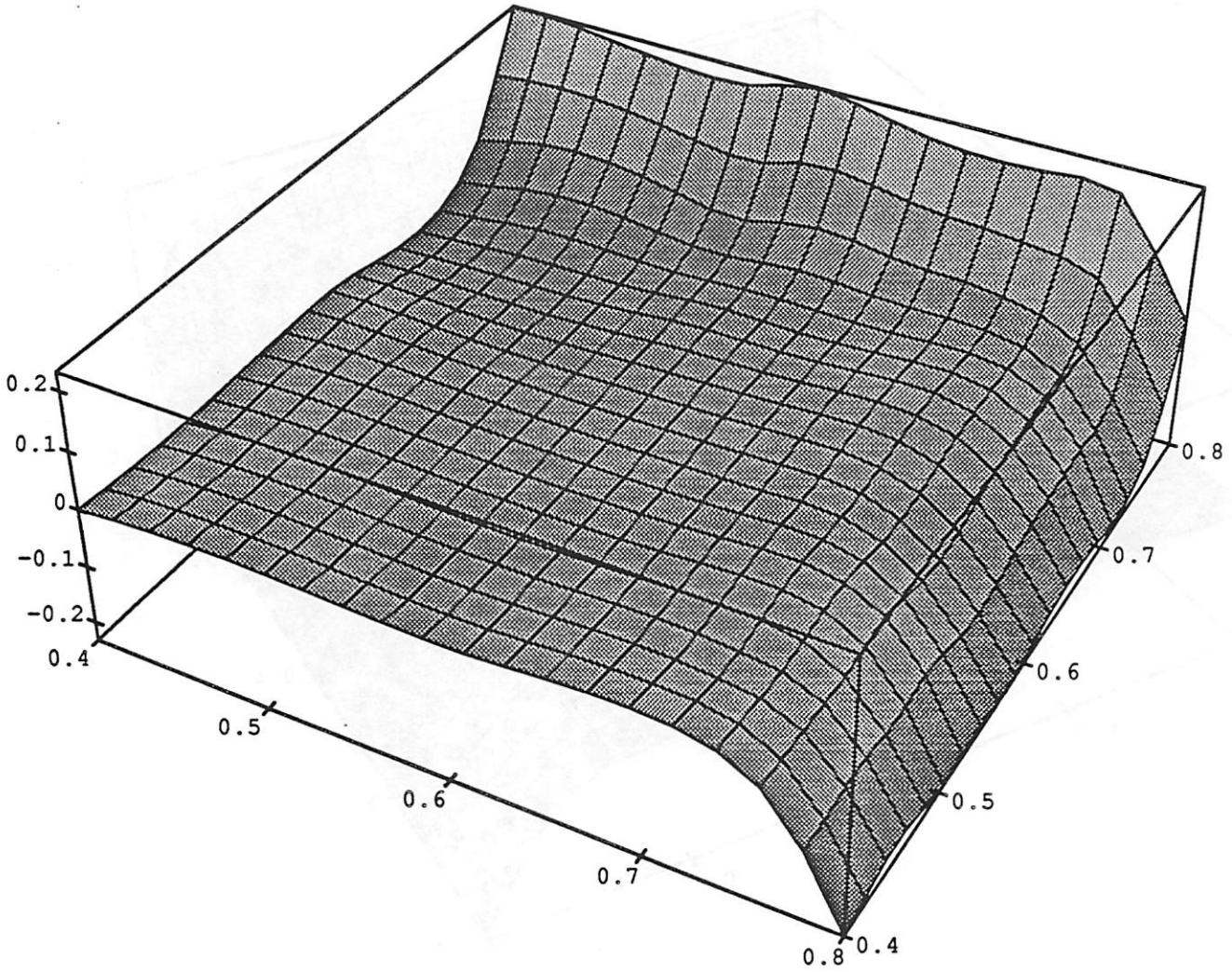

Figure 4 (b)

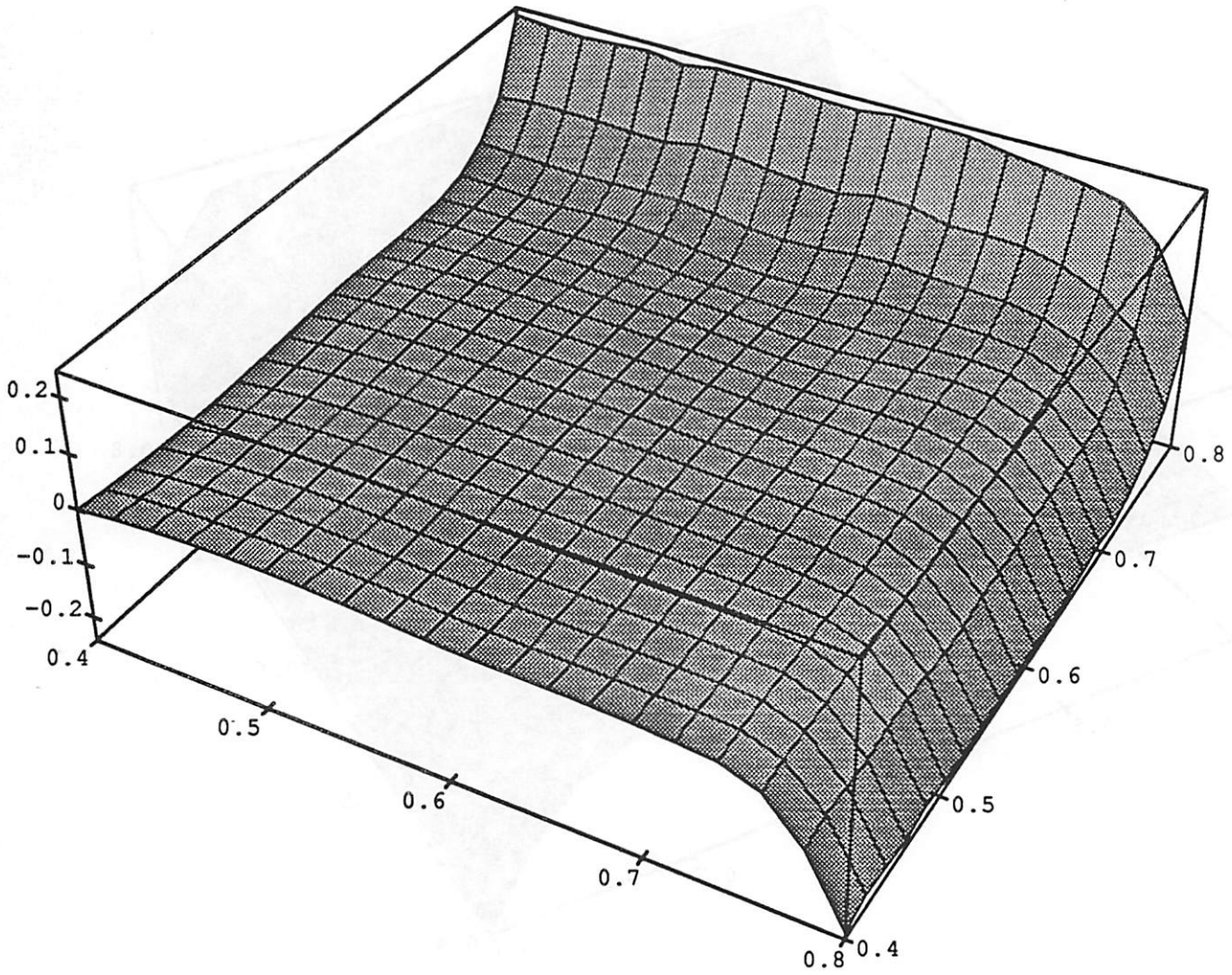

Figure 4 (c)

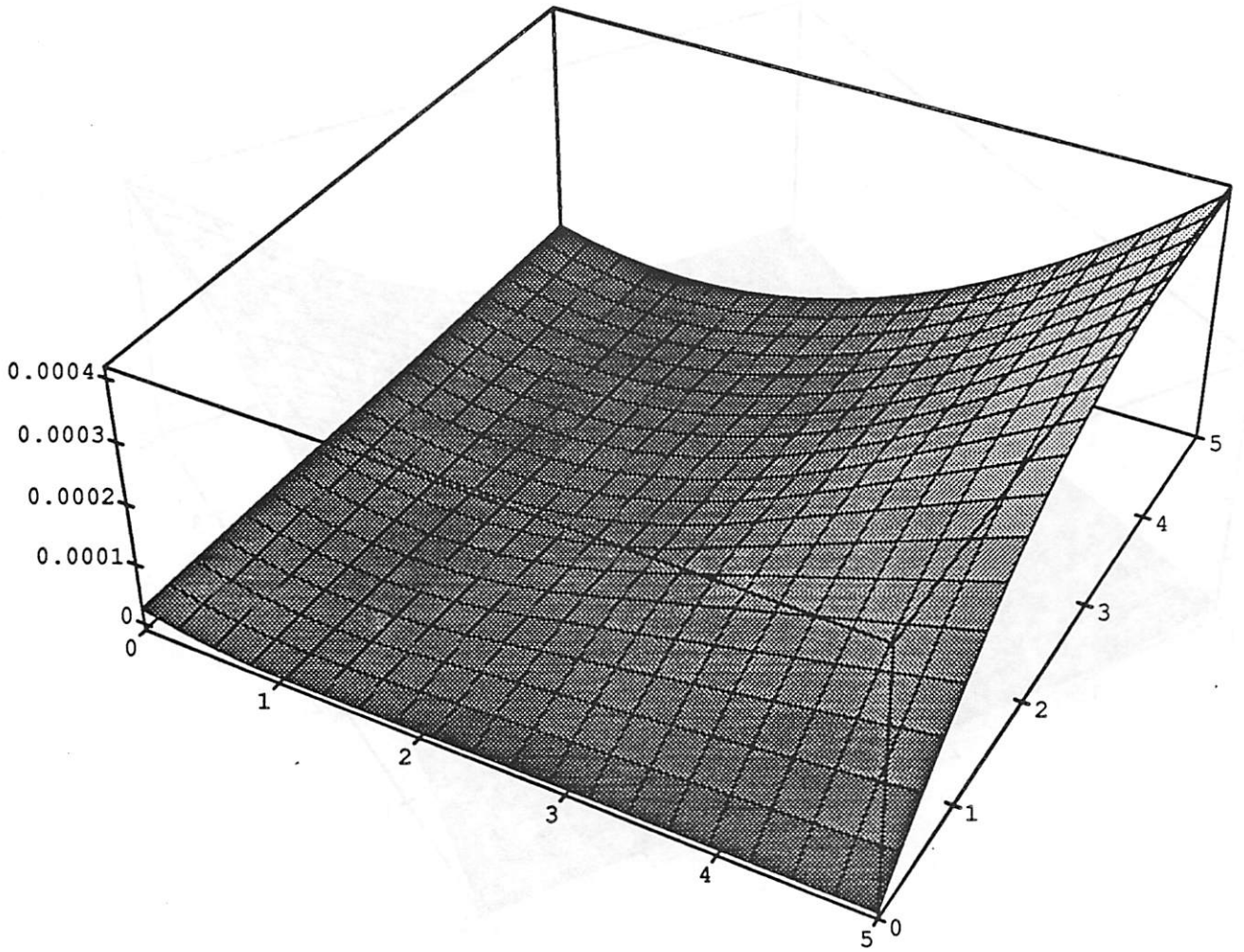

Figure 5

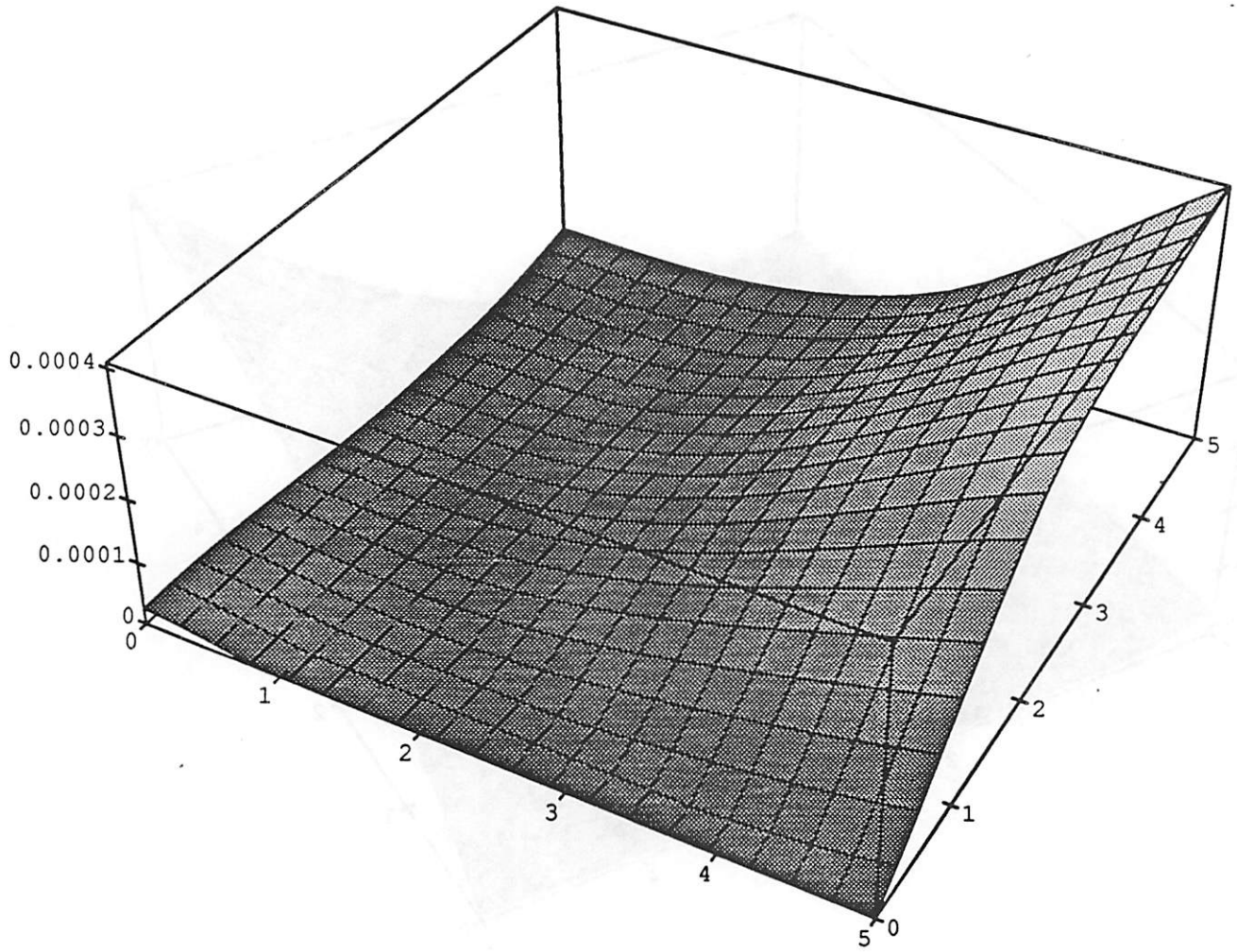

Figure 6 (a)

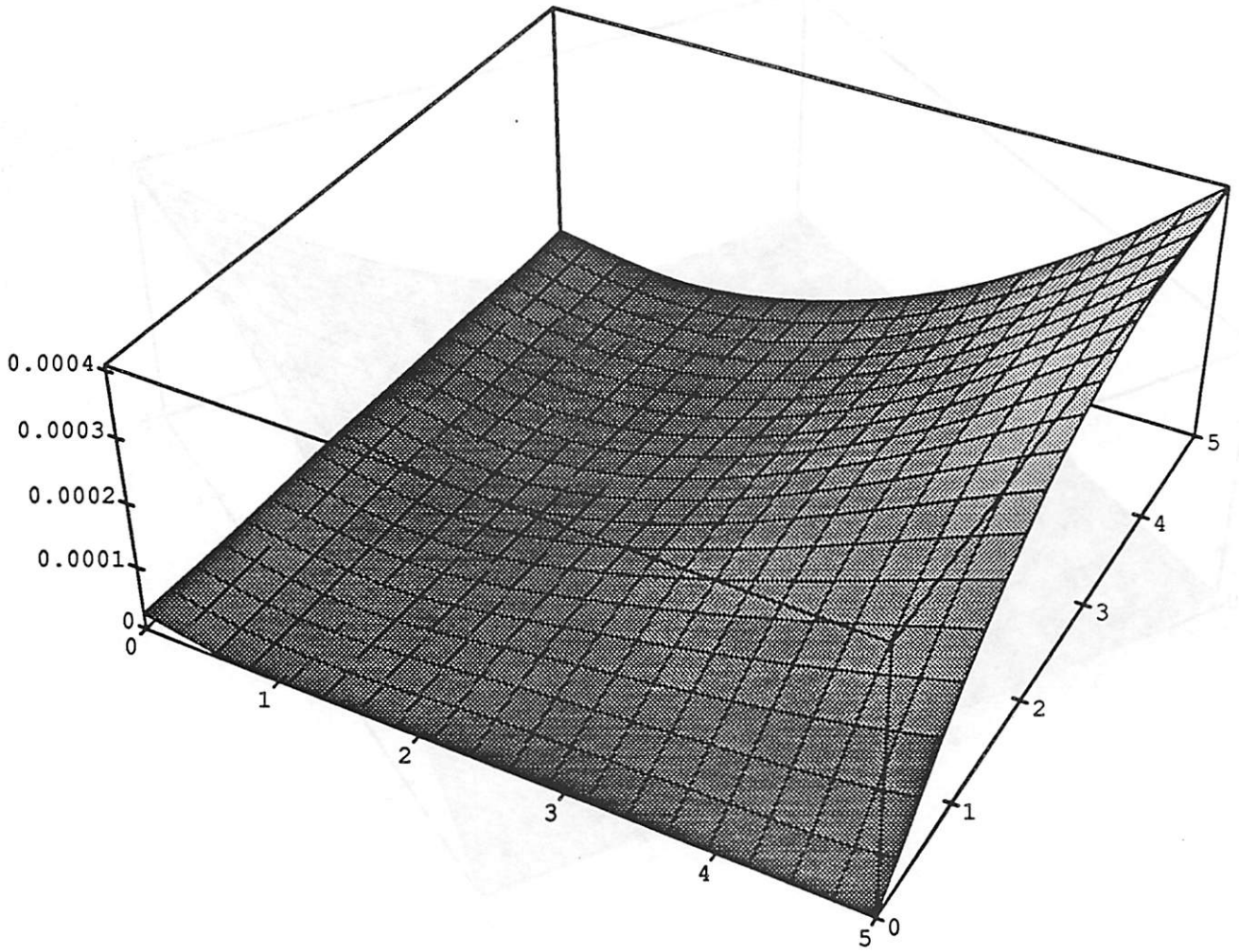

Figure 6 (b)

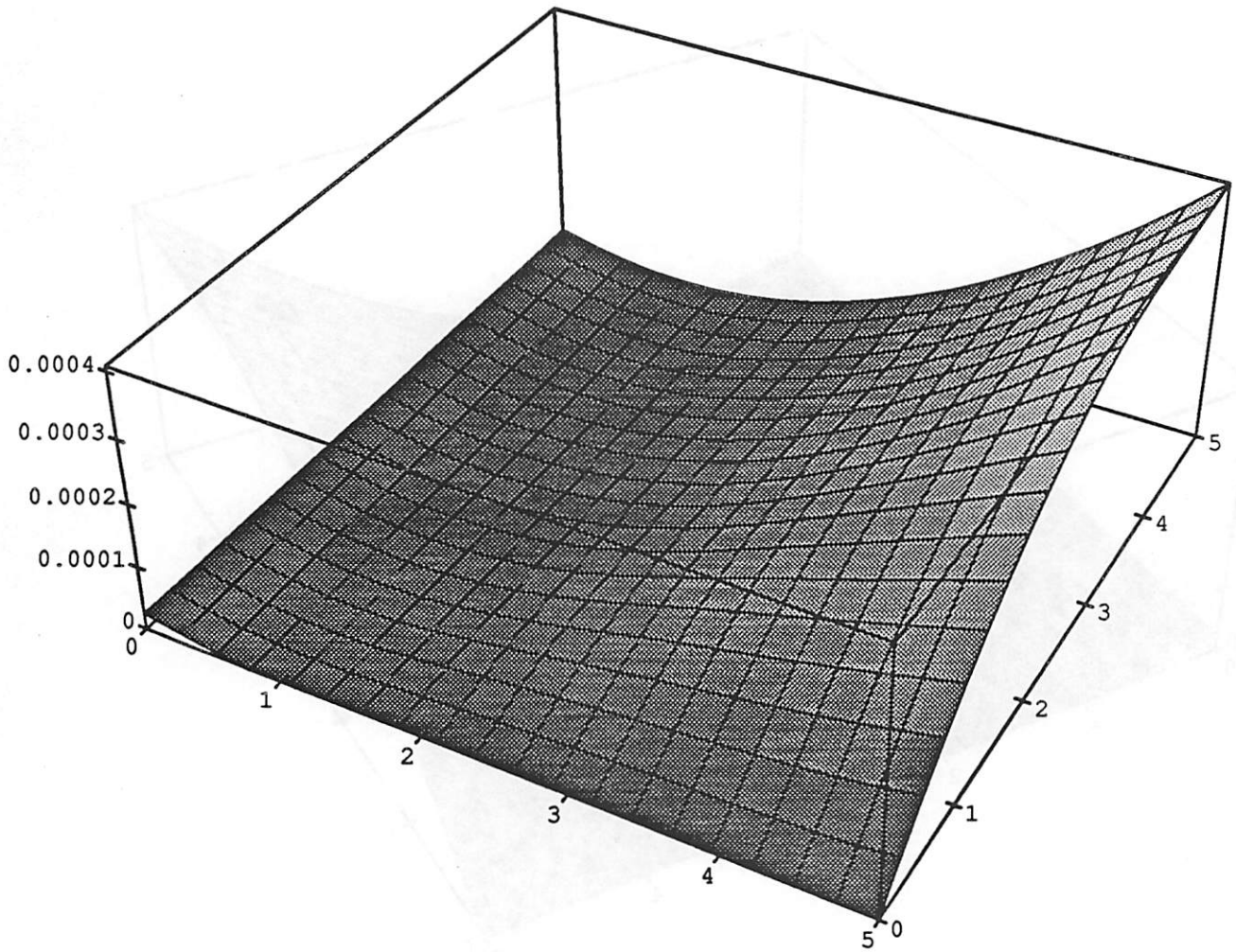

Figure 6(c)

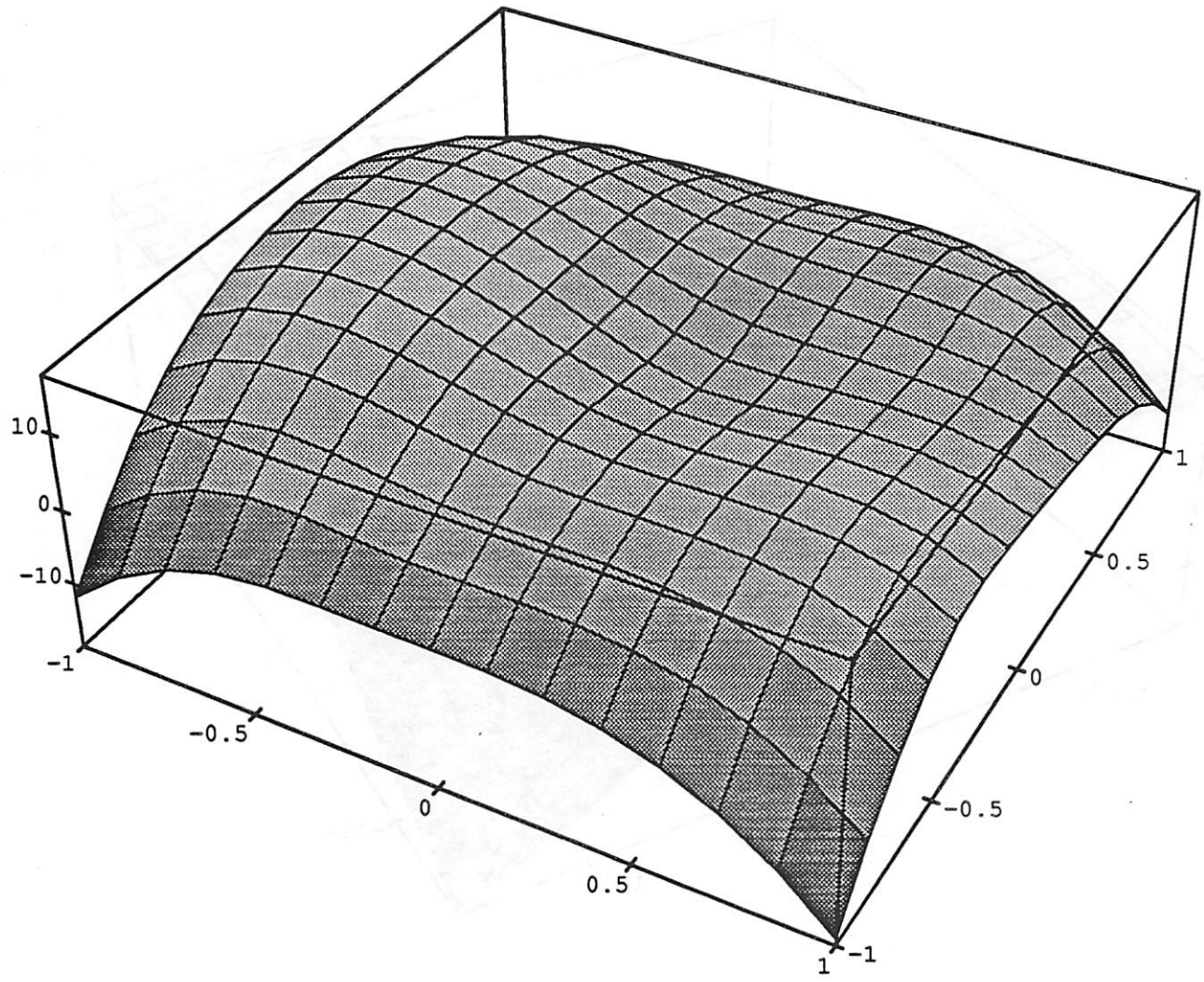

Figure 7

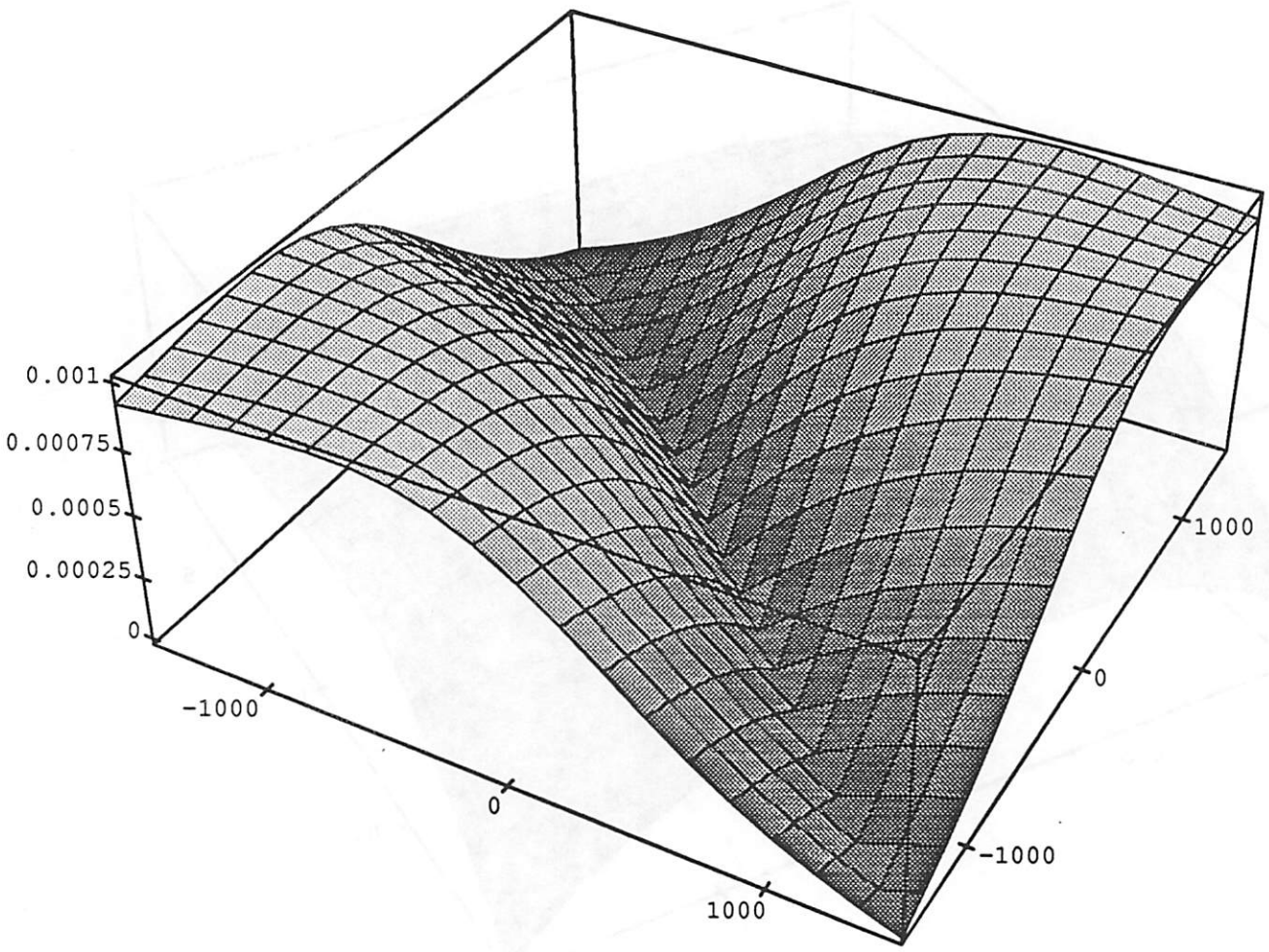

Figure 8

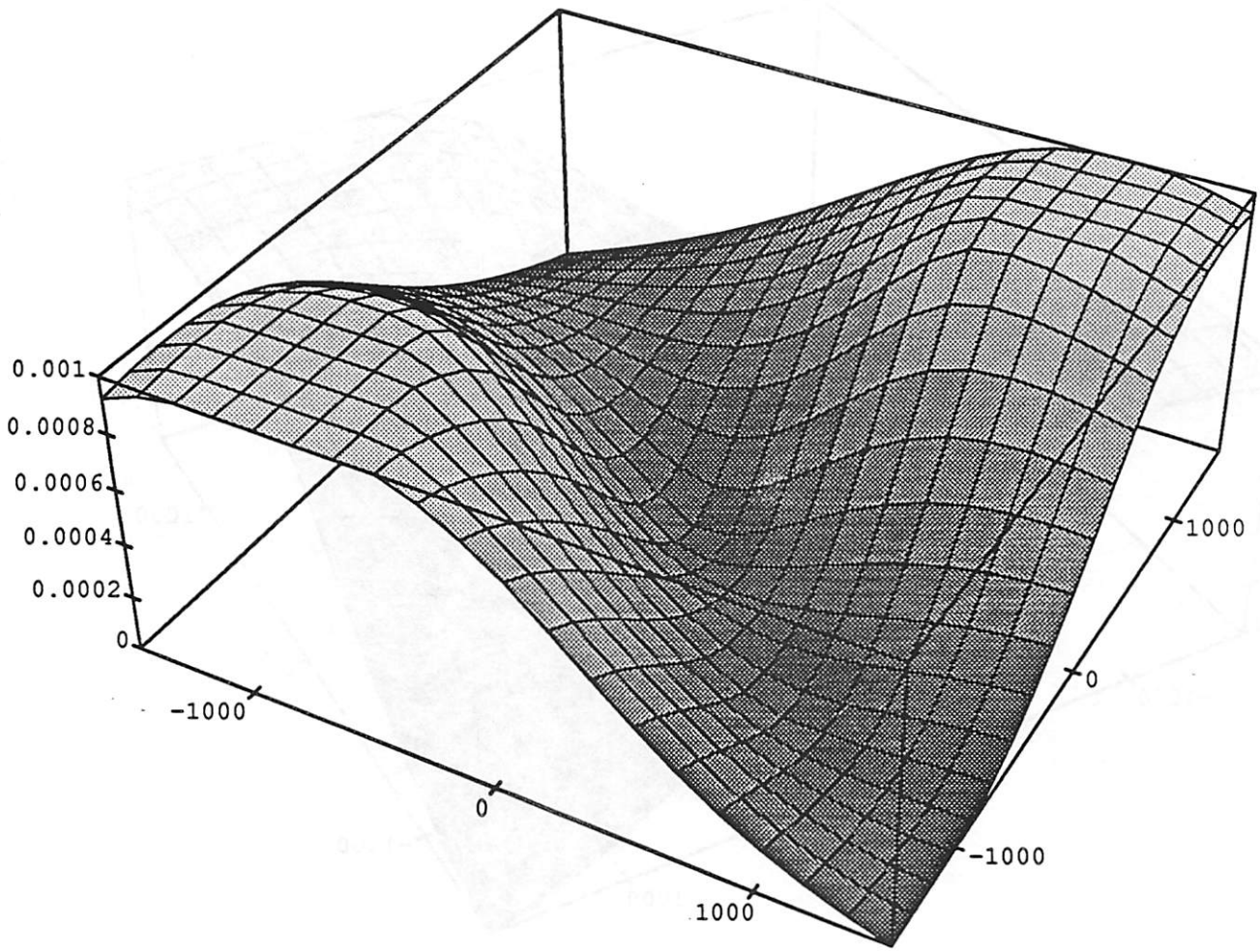

Figure 9(a)

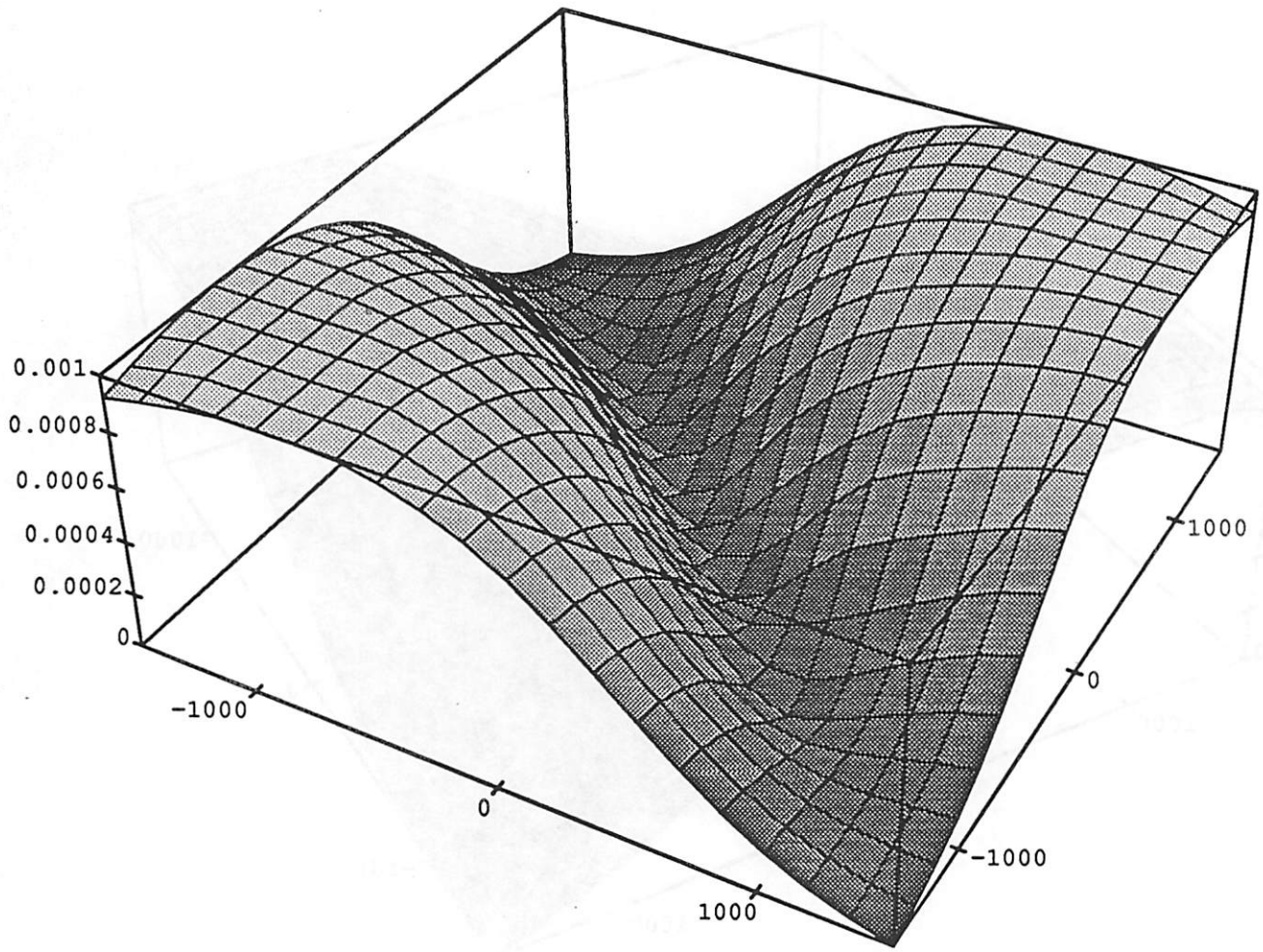

Figure 9(b)

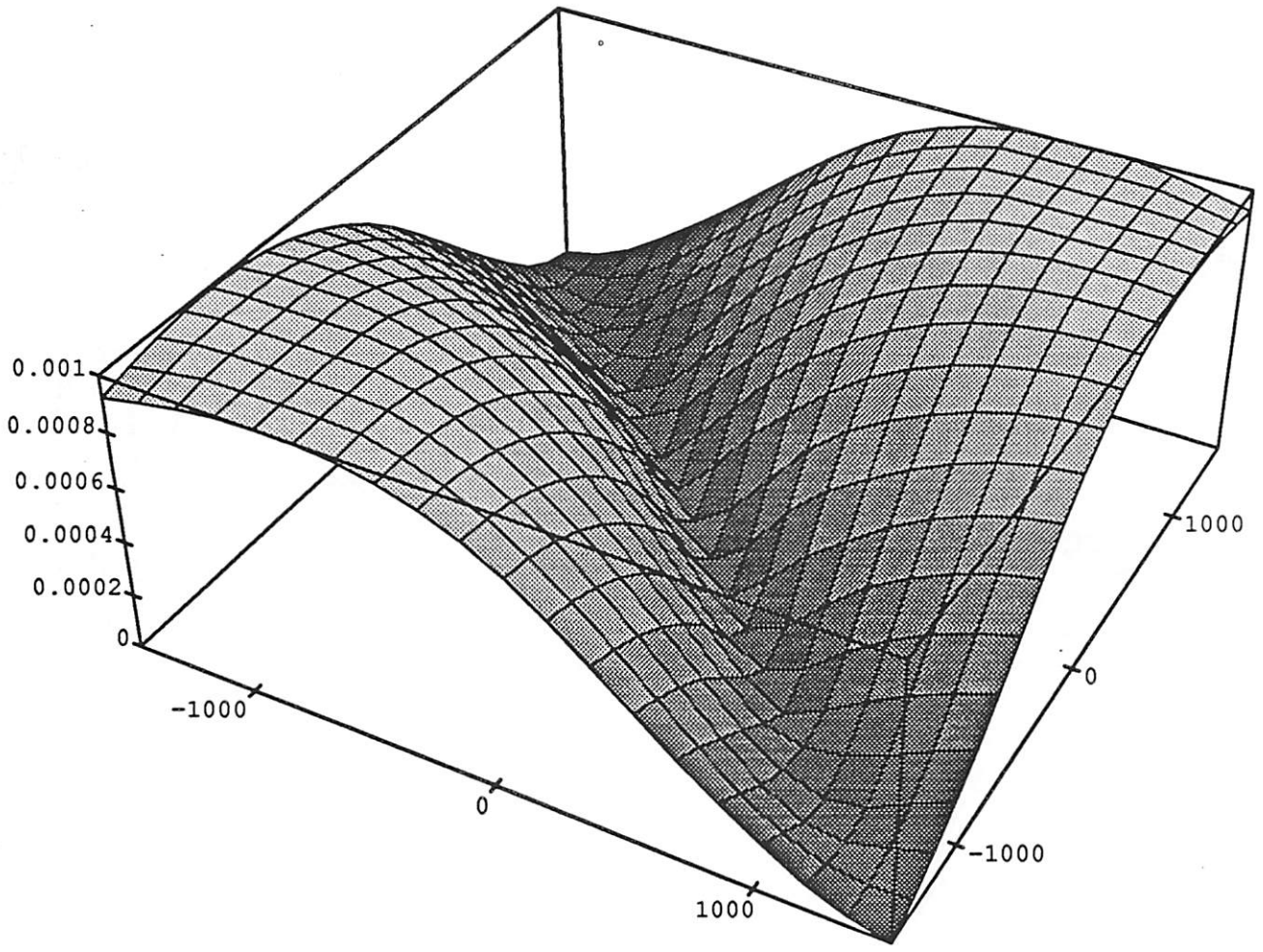

Figure 9(c)