University of Northern Iowa
UNI ScholarWorks

Presidential Scholars Theses (1990 – 2006) University Honors Program

1995

Multiquadric interpolation: Surface fitting in three-dimensional space

Michelle Ruse
University of Northern Iowa

Follow this and additional works at: http://scholarworks.uni.edu/pst
Part of the Mathematics Commons

Let us know how access to this document benefits you

Recommended Citation
Ruse, Michelle, "Multiquadric interpolation: Surface fitting in three-dimensional space" (1995). *Presidential Scholars Theses (1990 – 2006)*. 15.
http://scholarworks.uni.edu/pst/15

This Open Access Presidential Scholars Thesis is brought to you for free and open access by the University Honors Program at UNI ScholarWorks. It has been accepted for inclusion in Presidential Scholars Theses (1990 – 2006) by an authorized administrator of UNI ScholarWorks. For more information, please contact scholarworks@uni.edu.

MULTIQUADRIC INTERPOLATION: SURFACE FITTING IN THREE-DIMENSIONAL SPACE

Michelle Ruse

May 10, 1995
Senior Thesis Project
Dr. Timothy Hardy, advisor

INTRODUCTION

Using mathematics to solve a problem does not always yield a perfect or absolute answer but may instead yield an approximate solution. We can try to approximate the solution as precisely as possible by using the mathematical tools and skills that are available to us or we could try to discover new methods which would enable us to find good approximations. It is important that we have precise approximating tools to begin with, so that we may preserve as much accuracy as possible.

We can find such problems in the world around us. For instance, if we try to construct a topographical map of a mountainous region, first we gather data by measuring some elevations and locations. The data is then used to contruct the map. We now realize that because measuring every dip and valley of the area would be an impossible task, the map must be constructed from a set of random points. The next step is either to guess about the elevations between the data points, if there are enough points close enough together, or to estimate these elevations mathematically.

Since we would like to finish constructing the map by taking small regions around the known data points and finding approximating functions which, when graphed, will represent as

---

This work was supported in part by the National Center for Supercomputing Applications under grant TRA930381N.

precisely as possible the surface of the region, an entirely new problem arises. These surfaces around the known points cannot be easily calculated by use of simple functions. We now need to use these few, random data point to find an approximating surface by means of an interpolation method.

INTERPOLATION METHODS

The scenario above leads to an interpolation problem in R^3. We expect that interpolation methods in R^3 can be developed by generalizing interpolation methods appropriate for R^2. Well known and commonly used methods in the plane include Lagrange interpolatory polynomials, Taylor polynomials and cubic splines. We shall outline a general interpolation method appropriate for R^n called Multiquadric Interpolation and we shall compare Multiquadric Interpolation (MQ) with the classic methods by examining a specific problem in the plane.

Since we do not know the true elevation function, but only some point values of the function; the mathematical model is derived from the behavior of the approximating function. Thus, the accuracy of the approximation is an important factor in selecting a method to derive the estimated function from the given data.

Many interpolatory methods exist, such as the classical ones named above, and each has its strengths and its weaknesses. Some of these strengths and weaknesses depend upon the function itself, therefore some functions are easily interpolated because they do not change much, while others are not so easily estimated because their behavior is unpredictable. The problem is to find interpolation methods which can predict such behavior even in difficult instances.

We offer an example of interpolation in R^2 using the previously mentioned methods. We shall consider a known function, obtain data points by evaluating the function at specified abscissas and graph both the interpolating function the true function. By considering the difference between these functions, we can measure the accuracy of our technique. We consider Runge's function and seven specified data points.

$$f(x) = \frac{1}{1 + x^2}.$$

Figure 1: Lagrange approximation of f(x).

Figure 2: Taylor approximation of f(x).

Figure 3: Cubic Spline approximation of f(x).

These figures were created on MATLAB using Neville's Algorithm for the Lagrange Polynomial, a simple program for the Taylor Polynomial and the Natural (Free) Cubic Spline algorithm found in Burden and Faires's Numerical Analysis, 5th ed. [2]. As we can see in these figures some methods are more accurate than others, but we want more precision than these methods allow. Thus, we shall need to look for a more precise interpolation method.

MULTIQUADRIC INTERPOLATION

The above three classical methods of interpolation use polynomials in the Cartesian plane to derive approximating functions, so it seems that we could use polynomials of degree one in the form of linear combinations calculated from the given ordered pairs for interpolation in the Cartesian plane. The following discussion leads to the formation of an interpolatory method called Multiquadric Interpolation, or MQ.

Suppose we are dealing with one independent variable and we are given the following data:
$f(x_1), f(x_2), \ldots, f(x_n)$. The problem is to find an approximate $F(x)$ such that $F(x_i)=f(x_i)$ for $i=1,2, \ldots, n$ and $F(x)$ accurately describes the behavior of $f(x)$ in between these points. Throughout this report we shall use F to denote our interpolatory approximation to the function f with given data points. A reasonable function to try is the following:

$$F(x)=\sum_{i=1}^{n} c_i |x - x_i| \qquad \text{for } i=1,2,\ldots,n$$

where the c_i's are constants.

These c_i's are calculated by solving the linear equation $Ac=b$, where c is the unknown vector, b is the vector of length n with $f(x_1)$, $f(x_2), \ldots, f(x_n)$ as its components, and $A=[a_{ij}]$ is a matrix of size nxn whose components are given by $a_{ij} = |x_i-x_j|$. From this definition we can easily observe that A is a symmetric matrix with a principal diagonal of zeroes.

In order to find the constant vector c of unknown coefficients, matrix A must be nonsingular; accordingly $Ac=b$ will have a unique solution. Example 1, along with Figure 4 shows such an interpolation with one independent variable. Here the matrix is invertible; therefore the c_i's may be calculated uniquely. Figure 4 depicts graphically the interpolation method of Example 1 which finds a single approximating function from the data given.

In Example 1 the matrix was nonsingular, we can prove that in general the multiquadric matrix A will be nonsingular. We shall do this by developing an explicit formula for each of the c_i. We shall need the following three cases: first we develop a formula for c_i where $1 < i < n$, next we develop a formula for c_n and finally we develop a formula for c_1. In each case we have ordered our data points so that $x_1 < x_2 < \ldots < x_n$ and $f(x)$ is the vector of known data points,

$$A = \begin{bmatrix} |x_1-x_1| & |x_1-x_2| & \cdots\cdots & |x_1-x_n| \\ |x_2-x_1| & |x_2-x_2| & \cdots\cdots & |x_2-x_n| \\ \vdots & \vdots & \vdots & \vdots \\ |x_{i-1}-x_1| & |x_{i-1}-x_2| & \cdots\cdots & |x_{i-1}-x_n| \\ |x_i-x_1| & |x_i-x_2| & \cdots\cdots & |x_i-x_n| \\ |x_{i+1}-x_1| & |x_{i+1}-x_2| & \cdots\cdots & |x_{i+1}-x_n| \\ \vdots & \vdots & \vdots & \vdots \\ |x_{n-1}-x_1| & |x_{n-1}-x_2| & \cdots\cdots & |x_{n-1}-x_n| \\ |x_n-x_1| & |x_n-x_2| & \cdots\cdots & |x_n-x_n| \end{bmatrix}$$

As a linear system the matrix equation $Ac=f(x)$ becomes:

$$c_1|x_1-x_1| + c_2|x_1-x_2| +\ldots+ c_{i-1}|x_1-x_{i-1}| + c_i|x_1-x_i| + c_{i+1}|x_1-x_{i+1}| +\ldots+ c_n|x_1-x_n| = f(x_1)$$

$$c_1|x_2-x_1| + c_2|x_2-x_2| +\ldots+ c_{i-1}|x_2-x_{i-1}| + c_i|x_2-x_i| + c_{i+1}|x_2-x_{i+1}| +\ldots+ c_n|x_2-x_n| = f(x_2)$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$c_1|x_{i-1}-x_1| + c_2|x_{i-1}-x_2| +\ldots+ c_{i-1}|x_{i-1}-x_{i-1}| + c_i|x_{i-1}-x_i| + c_{i+1}|x_{i-1}-x_{i+1}| +\ldots+ c_n|x_{i-1}-xn| = f(x_{i-1})$$

$$c_1|x_i-x_1| + c_2|x_i-x_2| +\ldots+ c_{i-1}|x_i-x_{i-1}| + c_i|x_i-x_i| + c_{i+1}|x_i-x_{i+1}| +\ldots+ c_n|x_i-x_n| = f(x_i)$$

$$c_1|x_{i+1}-x_1| + c_2|x_{i+1}-x_2| +\ldots+ c_{i-1}|x_{i+1}-x_{i-1}| + c_i|x_{i+1}-x_i| + c_{i+1}|x_{i+1}-x_{i+1}| +\ldots+ c_n|x_{i+1}-x_n| = f(x_{i+1})$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$c_1|x_{n-1}-x_1| + c_2|x_{n-1}-x_2| +\dots+ c_{i-1}|x_{n-1}-x_{i-1}| + c_i|x_{n-1}-x_i| + c_{i+1}|x_{n-1}-x_{i+1}| +\dots+ c_n|x_{n-1}-x_n| = f(x_{n-1})$$

$$c_1|x_n-x_1| + c_2|x_n-x_2| +\dots+ c_{i-1}|x_n-x_{i-1}| + c_i|x_n-x_i| + c_{i+1}|x_n-x_{i+1}| +\dots+ c_n|x_n-x_n| = f(x_n).$$

<u>Proof(Case 1)</u>: We shall prove that c_i have unique solutions for $1<i<n$. We subtract the $(i-1)^{st}$ equation from the i^{th} equation in $Ac=f(x)$, giving us

$$c_1(|x_i-x_1|-|x_{i-1}-x_1|) + c_2(|x_i-x_2|-|x_{i-1}-x_2|) +\dots+ c_{i-1}(|x_i-x_{i-1}|-|x_{i-1}-x_{i-1}|) + c_i(|x_i-x_i|-|x_{i-1}-x_i|) + c_{i+1}(|x_i-x_{i+1}|-|x_{i-1}-x_{i+1}|) +\dots+ c_n(|x_i-x_n|-|x_{i-1}-x_n|) = f(x_i)-f(x_{i-1}).$$

We simplify,

$$c_1|x_i-x_{i-1}| + c_2|x_i-x_{i-1}| +\dots+ c_{i-1}|x_i-x_{i-1}| - c_i|x_i-x_{i-1}| - c_{i+1}|x_i-x_{i-1}| -\dots- c_n|x_i-x_{i-1}| = f(x_i)-f(x_{i-1}).$$

Since $(|x_i-x_n| - |x_{i-1}-x_n|)$ is the distance between x_i and x_{i-1}, the difference $|x_i-x_n| - |x_{i-1}-x_n| > 0$ whenever $n < i$. If n-i the difference is zero; if $n > i$ the difference is negative, so the negative sign precedes the c_n for $n > i$.

We divide both sides by the common factor and obtain

$$* \quad c_1 + c_2 +\dots+ c_{i-1} - c_i - c_{i+1} -\dots- c_n = \frac{f(x_i) - f(x_{i-1})}{|x_i - x_{i-1}|}.$$

Similarly we subtract the $(i+1)^{st}$ equation from the i^{th}

$$-c_1|x_i-x_{i+1}| - c_2|x_i-x_{i+1}| -\dots- c_{i-1}|x_i-x_{i+1}| -c_i|x_i-x_{i+1}| + c_{i+1}|x_i-x_{i+1}| +\dots+ c_n|x_i-x_{i+1}| = f(x_i)-f(x_{i+1}).$$

Since $|x_n-x_i| - |x_n-x_{i+1}|$ is the distance from x_i to x_{i+1}, and for $n < i$ or $n=i$, this difference is negative, a negative sign precedes c_i and the distance is written as an absolute value. For $n > i$ the difference is positive, so c_i is positive.

We simplify

$$** \quad -c_1 - c_2 -\dots- c_{i-1} - c_i + c_{i+1} +\dots+ c_n = \frac{f(x_i) - f(x_{i+1})}{|x_i-x_{i+1}|}.$$

We add * and **

$$-2c_i = \frac{f(x_i) - f(x_{i-1})}{|x_i - x_{i-1}|} + \frac{f(x_i) - f(x_{i+1})}{|x_i - x_{i+1}|},$$

$$c_i = \frac{-1}{2} * \left(\frac{f(x_i) - f(x_{i-1})}{|x_i - x_{i-1}|} + \frac{f(x_i) - f(x_{i+1})}{|x_i - x_{i+1}|} \right),$$

and therefore each c_i is determined uniquely.

Proof (Case 2): We shall prove that c_n is determined uniquely.

We add the last equation (n^{th}) to the first equation for $Ac=f(x)$

$$c_1(|x_1-x_1| + |x_n-x_1|) + c_2(|x_1-x_2| + |x_n-x_2|) +\ldots+ c_n(|x_1-x_n| + |x_n-x_n|) = f(x_1) + f(x_n).$$

We simplify, disregard $|x_1-x_1|$ since it is zero, and get

$$c_1|x_1-x_n| + c_2|x_1-x_n| + \ldots + c_n|x_1-x_n| = f(x_1) + f(x_n)$$

We factor $|x_1-x_n|$ from the left and divide:

$$\# \quad c_1 + c_2 + \ldots + c_n = \frac{f(x_1) + f(x_n)}{|x_1 - x_n|}.$$

Next we subtract the $(n-1)^{st}$ equation from the n^{th} equation:

$$c_1(|x_n-x_1| - |x_{n-1}-x_1|) + c_2(|x_n-x_2| - |x_{n-1}-x_2|) +\ldots+ c_{n-1}(|x_n-x_{n-1}| - |x_{n-1}-x_{n-1}|) + c_n(|x_n-x_n| - |x_{n-1}-x_n|) = f(x)-f(x_{n-1}).$$

The difference of this equation represents the distance between x_n and x_{n-1}, $x_n > x_{n-1}$, the coefficient of c_n will be -1:

$$c_1|x_n-x_{n-1}| + c_2|x_n-x_{n-1}| +\ldots+ c_{n-1}|x_n-x_{n-1}| - c_n|x_n-x_{n-1}| = f(x_n)-f(x_{n-1}).$$

We factor $|x_n-x_{n-1}|$ from the left, then divide by this factor:

$$\#\# \quad c_1 + c_2 +\ldots+ c_{n-1} - c_n = \frac{f(x_n) - f(x_{n-1})}{|x_n - x_{n-1}|}.$$

Now we subtract the equation ## from # and obtain the following:

$$-2c_n = \frac{f(x_n) + f(x_1)}{|x_n - x_1|} - \frac{f(x_n) - f(x_{n-1})}{|x_n - x_{n-1}|},$$

$$c_n = -\frac{1}{2} * \left(\frac{f(x_n) + f(x_1)}{|x_n - x_1|} - \frac{f(x_n) - f(x_{n-1})}{|x_n - x_{n-1}|} \right).$$

Therefore c_n is also uniquely determined.

<u>Proof (Case 3)</u>: We shall prove that c_1 is determined uniquely.

First we add the n^{th} equation to the first equation:

$$c_1(|x_1-x_1| + |x_n-x_1|) + c_2(|x_1-x_2| + |x_n-x_2|) +\ldots+ c_n(|x_1-x_n| + |x_n-x_n|) = f(x_1) + f(x_n)$$

We simplify the sums since the x_i are linearly ordered:

$$c_1|x_1-x_n| + c_2|x_1-x_n| + \ldots + c_n|x_1-x_n| = f(x_1) + f(x_n).$$

Next we factor $|x_1-x_n|$ from the left and divide:

$$@ \quad c_1 + c_2 + \ldots + c_n = \frac{f(x_1) + f(x_n)}{| x_1 - x_n |}.$$

We subtract the 2^{nd} equation from the 1^{st} equation as follows:

$$c_1(|x_1-x_1| - |x_2-x_1|) + c_2(|x_1-x_2| - |x_2-x_2|) +\ldots+ c_{n-1}(|x_1-x_{n-1}| - |x_2-x_{n-1}|) + c_n(|x_1-x_n| - |x_2-x_n|) = f(x_1)-f(x_2)$$

So, the differences of this equation represent the distance between x_1 and x_2, and since x_2 is greater than x_1, c_1 will have a coefficient of -1:

$$-c_1|x_1-x_2| + c_2|x_1-x_2| +\ldots+ c_{n-1}|x_1-x_2| + c_n|x_1-x_2| = f(x_1)-f(x_2).$$

We factor $|x_1-x_2|$ from the left side, then divide by it:

$$@@ \quad -c_1 + c_2 + \ldots + c_{n-1} + c_n = \frac{f(x_1) - f(x_2)}{| x_1 - x_2 |}.$$

Finally we subtract @@ from @

$$-2c_1 = \frac{f(x_n) + f(x_1)}{| x_n - x_1 |} - \frac{f(x_1) - f(x_2)}{| x_1 - x_2 |},$$

$$c_1 = -\frac{1}{2} * \left(\frac{f(x_n) + f(x_1)}{| x_n - x_1 |} - \frac{f(x_1) - f(x_2)}{| x_1 - x_2 |} \right).$$

Therefore, c_1 is uniquely determined.

Since we obtained explicit formulae for each of the c_i, we know that the matrix equation Ac=f has a unique solution, and therefore A is a nonsingular matrix.

It can be noted that in the interpolatory equation

$$* \ F(x) = \sum_{i=1}^{n} c_i |x-x_i| ,$$

$|x-x_i|$ is the distance between x and x_i. Thus, $|x-x_i| = [(x-x_i)^2]^{1/2}$. The definition of Euclidean distance in higher dimensional spaces can be represented similarly, only more variables are added. Knowing this fact, now we can generalize the interpolation method (*) in R^n by arguing in the following manner:

Let x_i be any given data point in R^n, then

$$F(x) = \sum_{i=1}^{n} c_i \, d(x,x_i) \quad \text{where d is the distance.}$$

Beginning in three dimensional space, the equation has two independent variables, the data is ordered triples, and the distance equation is $d((x,y),(x_i,y_i)) = \sqrt{(x-x_i)^2 + (y-y_i)^2}$ for $i=1,2,...,n$. $F(x) = \sum_{i=1}^{n} c_i \, d((x,y),(x_i,y_i))$ is the approximating function. So the nxn matrix A is

$$\begin{bmatrix} \sqrt{(x_1-x_1)^2+(y_1-y_1)^2} & \sqrt{(x_1-x_2)^2+(y_1-y_2)^2} & \cdots & \sqrt{(x_1-x_n)^2+(y_1-y_n)^2} \\ \sqrt{(x_2-x_1)^2+(y_2-y_1)^2} & \sqrt{(x_2-x_2)^2+(y_2-y_2)^2} & \cdots & \sqrt{(x_2-x_n)^2+(y_2-y_n)^2} \\ \vdots & \vdots & & \vdots \\ \sqrt{(x_n-x_1)^2+(y_n-y_1)^2} & \sqrt{(x_n-x_2)^2+(y_n-y_2)^2} & \cdots & \sqrt{(x_n-x_n)^2+(y_n-y_n)^2} \end{bmatrix}$$

In order to solve $Ac=f(x,y)$, A must be nonsingular. Since it has a principal zero diagonal and for all other entries $a_{ij}=a_{ji}>0$, the matrix A is nonsingular as Blumenthal apparently knew in the 1930's when he obtained this result by use of Cayley-Menger Determinants [1]. It was some fifty years later, in the 1980's, that this fact was deemed important in approximation theory.

Now, A is invertible, so the unknown c_i's may be calculated. In Figure 4 the of addition of absolute value functions that yielded a piecewise linear approximating function was shown, the interpolatory function maybe visualized as a linear combinations of frustrums of right circular cones which are added pointwise to form the approximating surface.

If these frustrums lie on the x-axis, then projection of interpolation using one independent variable in the x-z plane may be visualized as sliding a sheet through the x-z axis. We see the result is a plane of absolute value functions as was shown in Example 1 and the accompanying Figure 4. So, in the general case when the frustrums do not lie on the x-axis and we slide a sheet along the x-z plane, the result is a plane of half-hyperbolas. It follows that an approximating function may be calculated using a linear combination of half-hyperbolas.

Now we generalize in R^3 to construct the approximating surface using two independent variables. This can be visualized in Figure 5 which shows two cones with origins x_1 and x_2. A random x,y pair is selected, then the corresponding z values from the frustrum of each cone are added to obtain the z values, z_1 and z_2, of the approximating function. This is similar to the pointwise addition of functions as was seen in Example 1.

Here the resulting approximation will be a surface which estimates the surface of the given data. Since the equation of absolute value was the first interpolation method that we evaluated and in examination of its natural generalization in R^n resulted in graph of half-hyperbolas, we now examine the equation of a hyperbola:

$$y^2 - (x-a)^2 = r^2,$$

can be written

$$y^2 = (x-a)^2 + r^2,$$

finally,

$$y = \sqrt{(x-a)^2 + r^2}.$$

We may use this to form each term of our approximating function. Thus the term $(x-x_i)^2 + r^2$ can substitute for $(x-x_i)^2$ or $|x-x_i|$ in our previous development. Thus, the new function for the interpolation will be the following:

$$** \quad F(x) = \sum_{i=1}^{n} c_i \sqrt{(x-x_i)^2 + r^2}$$

In solving the system $Ac=f(x)$, we must first know that there is a unique solution. Micchelli proved the following result:

> Given any distinct points $x_1,\ldots,x_n$ in the plane
> $(-1)^n \quad \det\sqrt{1 + |x_i-x_j|^2} \quad > 0$
> This theorem says, in particular, that there is a unique surface $f(x) = c_1 \; 1 + |x-x_1|^2 + \ldots + c_n \; 1 + |x-x_n|^2$ which interpolates (data) $y_1,\ldots,y_n$ at $x_1,\ldots,x_n$. [4]

Thus, the function (**) will work as an interpolatory function which will provide us with an approximating function for the known data. This approximating function, when graphed, will give an approximation to the true graph as the addition of hyperbolas, given that the parameter r^2 is greater than zero.

We were able to move from one variable to two variables and even n variables with the distance as the means of interpolation, so we may now move from one to two to n variables using absolute values, hyperbolas, then hyperboloids.

The equation of a hyperboloid of two sheets is the following: $w^2-(x-a)^2-(y-b)^2 = c^2$,

Therefore $w^2 = (x-a)^2 + (y-b)^2 + c^2$

finally, $w = \sqrt{(x-a)^2 + (y-b)^2 + c^2}$.

By restricting our attention to one case, we recognize this to be the equation for one sheet of a hyperboloid of two sheets; $w = (x-a)^2 + (y-b)^2 + c^2$ is one sheet of this hyperboloid.

We substitue this equation to form a new interpolatory function which is the following equation:

$$F(x) = \sum_{i=1}^{n} c_i \sqrt{(x-x_i)^2 + (y-y_i)^2 + r^2}.$$

This interpolatory function will find an approximation function for two independent variable which may be represented graphically as a surface. Now the linear equation $Ac=f(x,y)$ must have a unique solution so that c is a vector of constants and interpolation using $F(x)$ will be possible. Micchelli's theorem also guarentees the uniqueness of the c_i coefficients for $i=1,2,\ldots,n$. Therefore, we have an interpolatory function. Micchelli's theorem says that interpolation is also possible for any finite number of independent variables,

$$F(x) = \sum_{i=1}^{n} c_i \sqrt{(x-x_i)^2 + (y-y_i)^2 + \ldots + (z-z_i)^2 + r^2} .$$

Although we certainly do not expect to be able to visualize the resulting hypersurface in any space of greater than dimension three.

ACCURACY OF MULTIQUADRIC INTERPOLATION

As we can see in Figures 1-3, interpolation methods are not perfect. But we hope to demonstrate that the MQ method proves to interpolate Runge's function far better than the classical methods, although not infallibly. Richard Franke realized a need to evaluate the accuracy as well as other factors of known methods of interpolation of scattered data [3]. In his evaluation he rated these methods with letter grades A,A-,...,F, on the basis of many characteristics that he considered important for analyzing the techniques. The method developed above, called MQ, or Multiquadric Interpolation, received A's in Complexity, Accuracy, and Visual and received B-/C- in time evaluation. Still, this is merely Franke's idea of what criteria are necessary to receive an A grade.

We find it necessary as well as worthwhile to test our newly found interpolatory method to discover its advantages and limitations. Taking a closer look at our MQ formula, we would like to find reasonable values for the parameter r^2.

Although a graph of an MQ approximation function may closely resemble that of the actual function, it does not necessarily mean it is the best or most precise interpolation; especially since we do not fully understand the unkown r^2 parameter in the equation.

Even if we find a value for r^2 which yields an approximating function whose graph is close to the graph of the true function, it does not mean that we have found an optimal r2. In fact, it is a current topic of mathematical research in the area of interpolatory methods, but there is currently no single theory for determining an r^2 parameter for all cases.

OPTIMIZING r^2

Now we direct our examination of MQ to the search for an optimal parameter, r^2. Referring to Franke's work with interpolation, Audry Ellen Tarwater points out that Franke's evaluation clearly states that MQ is far better than all other methods evaluated, but "by optimizing r^2, the results obtained are significantly improved, indicating that MQ can be far better than previously expected" [5]. A few possibilities for optimizing the r^2 parameter include finding a set numerical value from the given data, finding a variable r^2 such that r^2 is some function or optimizing r^2 with information other than the data points.

We can try to optimize r^2 as a constant in R^2 for Runge's function, which was exhibited in figures 1,2 and 3 by changing the value of r^2 for different trials of approximating this function. In Appendix A, figures 6-13, there are some examples of varying r^2 between zero and ten. This is a simple function in R^2, so it is easy to substitute different values for r^2, find the L_1 error, and graph each approximation on the same graph with the true function, all in a reasonable amount of time and coding in MATLAB. We discover that Figure 6 with r^2 as zero, that we do not the have smoothness as the original function. Now we can refer to Tarwater's investigation which states that a larger r^2 increases the waves, and smaller r^2 decreases the smoothness of the graph [5]. In Figure 7 further investigation, with r^2 as ten, shows an undulating graph which resembles Lagrange Polynomial interpolation method.

Trying several values between zero and ten, we find by a visual analysis, as well as numerical analysis of the errors, that r^2 appears to be between zero and one. In Figure 8, where r^2 is two, and Figure 9, where r^2 is one, we see that our approximation is becoming more accurate, but we know that our approximation with r^2 parameter equal to zero is not better than the parameter. Further investigation, as seen in Figures 10 and 11, leaves the optimal parameter between 0.01 and 0.02. We further refined r^2 to 0.013, Figure 12, and 0.0133, Figure 13. But we can see very little difference between Figures 12 and 13, so it would seem that we optimized the constant parameter as far as was possible.

But this example oversimplifies the problem of optimizing the parameter, since all functions do not behave like Runge's function. Multiquadric approximations in R^3 present an even bigger problem. First of all, the immense amount of computing time necessary to run trials of the program and to graph it, as well as the vast possibilities of parameter values make trial and error methods inappropriate for finding precise results in a reasonable amount of time.

Although we do not expect to find the optimal r^2 by means of trial and error, we can explore the behavior of a function in R^3 and we might also gain some insight into the parameter using this method.

We investigated the parameter r^2 as it pertains to Franke's surface:

```
f(x)=9*(.75*exp(-.25*((x-3)²+(y-3)²))+.75*exp((-x/49)-(y/100))
    +.5*exp(-.25*((x-8)²+(y-4)²))-.2*exp(-1*(x-5)²-(y-8)²)).
```

On the 4-processor Cray Y-MP at NCSA, we ran a series of Fortran programs designed to solve the multiquadric matrix equation with various values for r^2. We not only varied r^2, but we also varied the number of data points used, which was anywhere from 20 up to 300 randomly selected data points. These programs yielded the unknown coefficients of the vector c, which were then used to construct the multiquadric interpolatory function. We then generated a graphical representation of the surface within MATLAB running on a 4-processor Sun 670-MP. We found it convenient to dilate the domain uniformly so as to present the surface on the domain [0,100]x[0,100]. A representative sample of the resulting surfaces are included in Figures 14a, 14b and 14c through Figures 20a, 20b and 20c.

In the figures shown, 20 random data points were used, while in Figures 14a-20a the surfaces are graphed on using the domain of [0,35]x[0,35], and Figures 14b-20b are graphed using the domain of [0,100]x[0,100]. Figures 14c-20c are the same surface as those in 14a-20c, expect they are shaded to aid visualization of the elevations in different regions of the graph. Here we look at the l_1 error. In Figures 14a and 14b the parameter is zero, we can see the decreased smoothness, not only due to the nature of MQ with small parameter values, but also due to the small number of data points. The two peaks in Figure 14b demonstrate this especially; both would appear to be cones if there were less known data points on the lower peak. But we can see that the higher peak and the value appear to be conical.

Trying r^2 as 40, yield the results be would expect, as Figures 15a and 15b show, the peaks and valley are smooth, but the edges which should be flat are wavy. Figures 16 and 17 show the parameter as one and two, respectively, which helps us determine that the optimal r^2 must be between these values. Furthermore, we find as shown in Figures 18-20 that the optimal parameter is near 1.3. The further refinements in Figures 19 and 20 with the parameter equal to 1.33 and 1.339, show little improvement even though r^2 has more precision. Again, we have gone as far as possible with this investigation of a constant parameter for this surface.

CONCLUSIONS

We found that varying r^2 gave us different graphs in R^2 and different surfaces in R^3. We consistently found that large values for r^2 resulted in poor interpolatory graphs and surfaces compared to the smaller r^2 values. In neither case did we find an optimal parameter value.

In constructing the Franke surface, not only did we change r^2, but we also varied the number of data points. Since we observe that the smaller the parameter value, the better the approximated surface; we also consider the number of known data points to explain the surfaces constructed with a small r^2 which are not smooth. The construction of these surfaces involved less data points than the smooth surface interpolation.

Therefore we can conclude that not only does r^2 seem to be a small number in both of our cases, but the accuracy of the interpolation also depends upon the number of data points given.

Furthermore, the location of these points of these data points are important since some regions of the surface are accurate, while other regions are not. We can see this specifically at the corners of the surface, which indicates that there are too few points, especially along the edges. We can also see this in the Figures 14a and 14b and where r^2 was zero and the surface was smoother when specific regions contained more data points.

Although our MQ approximations proved to be accurate when r^2 was small, they were not always efficient. The Franke surface construction required the use of a supercomputer for data point computation and the use of MATLAB for generating a graphic representation. Although the supercomputer allows us to compute in a few minutes what takes two or more hours on our usual computer, we do not always have access to such technology. So, MQ is not as efficient as it is accurate.

As Tarwater concludes in her study of the parameter r^2, "It has been shown that to optimize r^2, i.e. to minimize the error of the approximation, more factors are involved than the data locations" [5], we can also conclude that we are missing some information necessary to optimize MQ interpolation. An interpolatory method may work well, but it cannot perfectly determine every function, so many different techniques are necessary tools for the mathematician. Also, analyzing the techniques for sensitivity, accuracy or other important traits of the interpolatory function cannot be overlooked. In conclusion, Multiquadric Interpolation seems to be the best interpolation method available to us today.

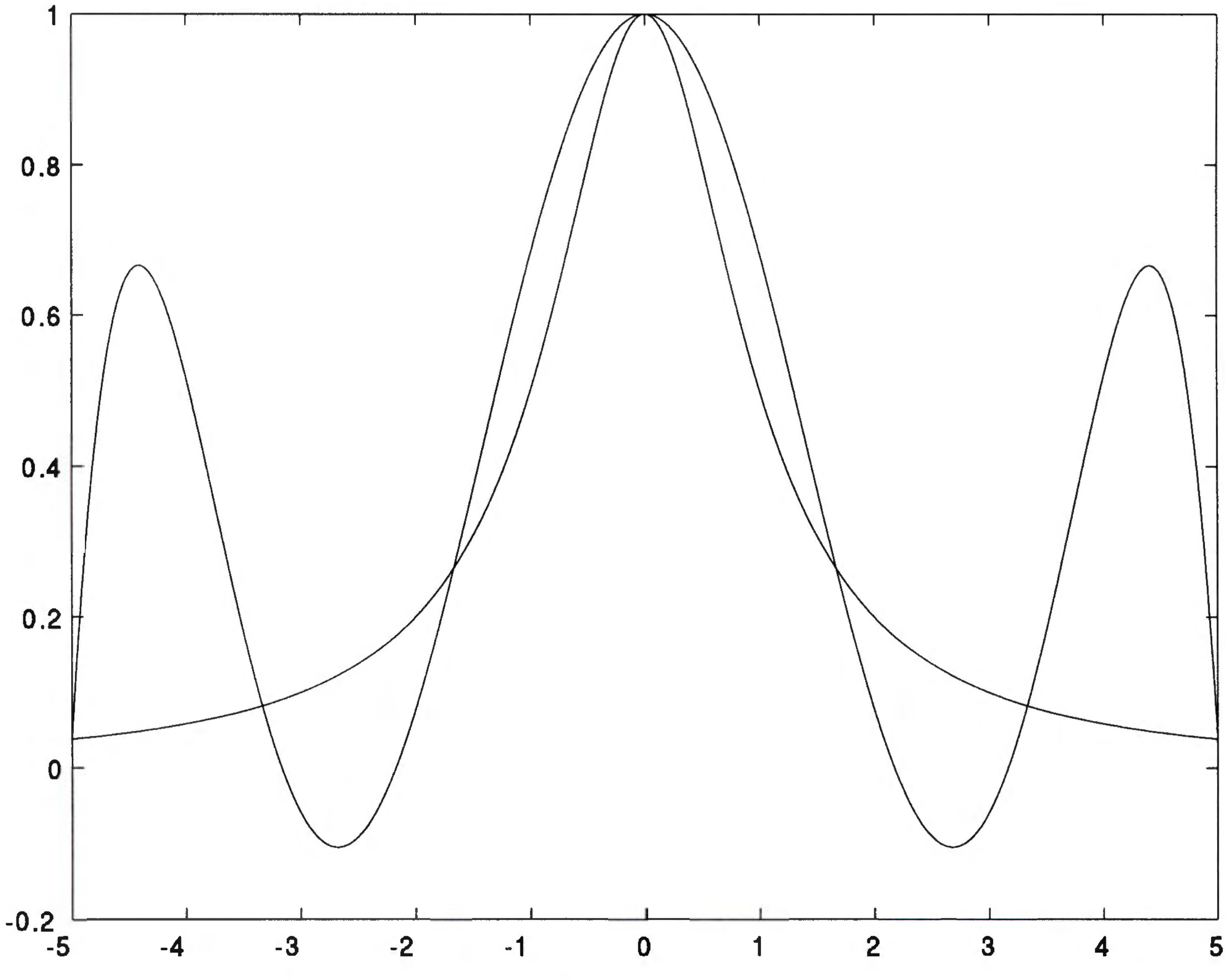


Figure 1.

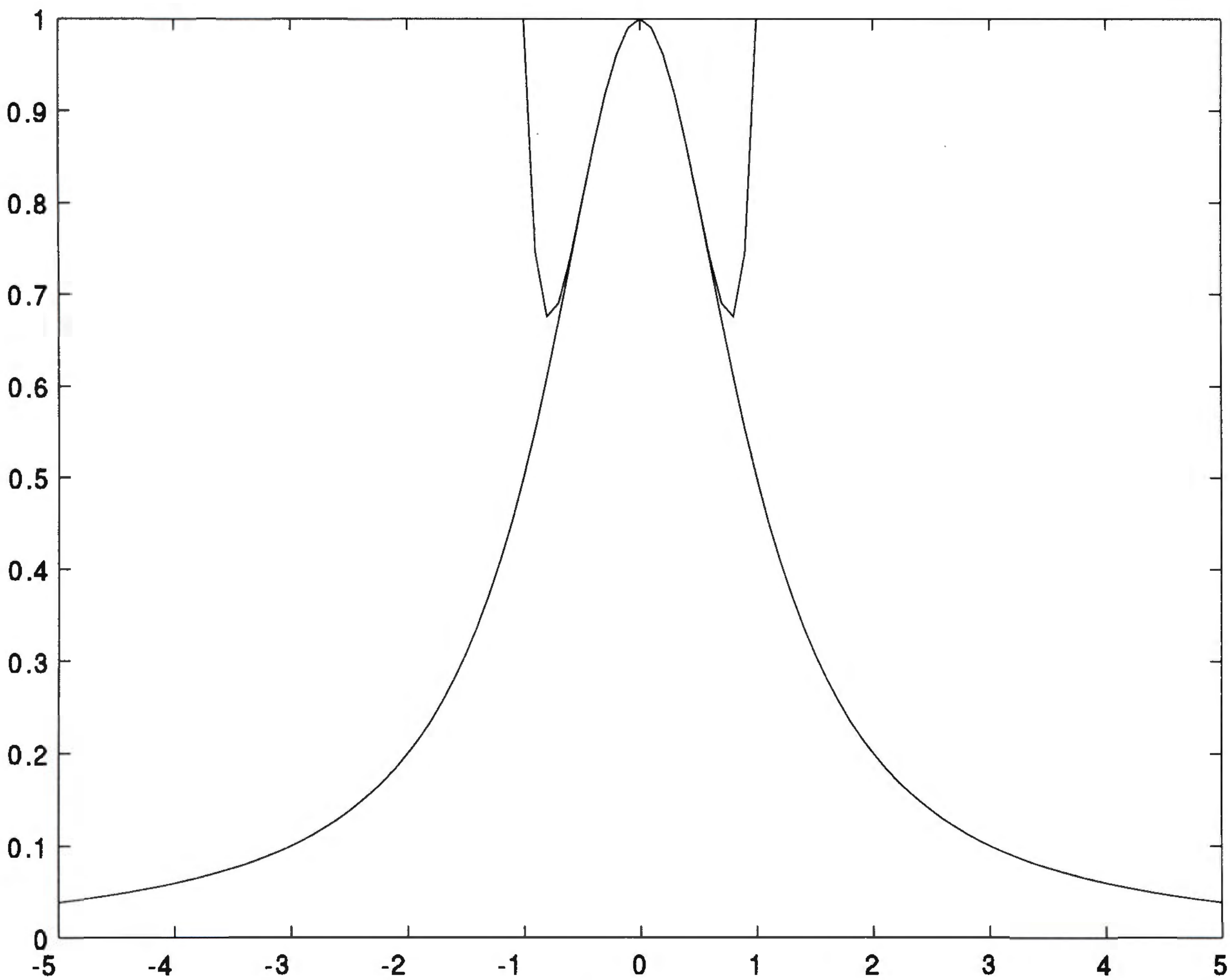


Figure 2.

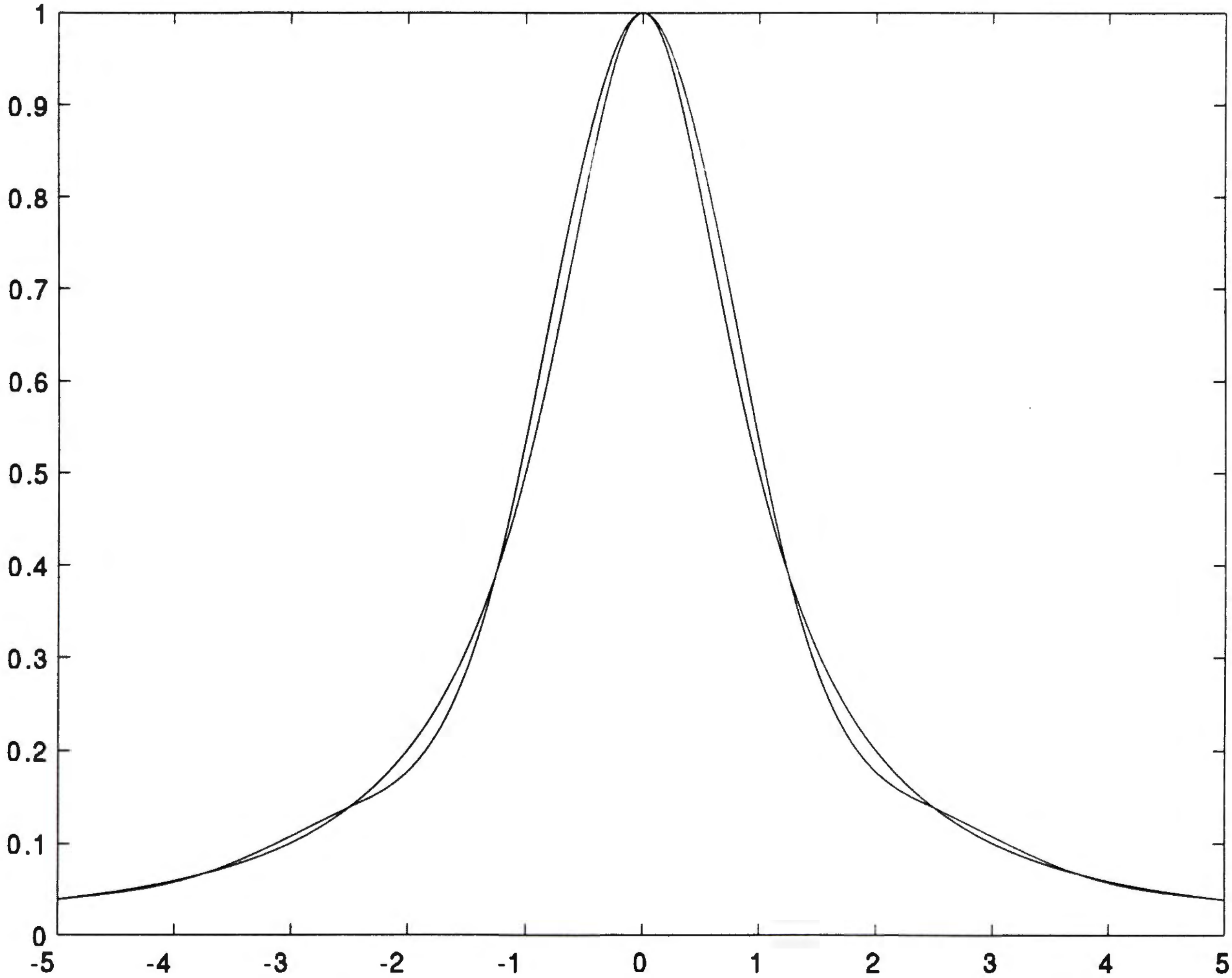


Figure 3.

Example 1

Interpolate the function f(x), given the following data:

$f(1)=4,\ f(2)=3,\ f(4)=7$

So n=3, $F(x)=\sum_{i=1}^{3} c_i|x-x_i|$, and $F(x_i)=f(x_i)$ for i=1,2,3.

The equation Ac=f(x) is

$$\begin{bmatrix} |1-1| & |1-2| & |1-4| \\ |2-1| & |2-2| & |2-4| \\ |4-1| & |4-2| & |4-4| \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \\ 7 \end{bmatrix}$$

Solve using linear algebra, Gaussian elimination

$$\left[\begin{array}{ccc|c} 0 & 1 & 3 & 4 \\ 1 & 0 & 2 & 3 \\ 3 & 2 & 0 & 7 \end{array}\right] \text{R1<-> R2} \left[\begin{array}{ccc|c} 1 & 0 & 2 & 3 \\ 0 & 1 & 3 & 4 \\ 3 & 2 & 0 & 7 \end{array}\right] \text{R3 -> R3-3R1}$$

$$\left[\begin{array}{ccc|c} 1 & 0 & 2 & 3 \\ 0 & 1 & 3 & 4 \\ 0 & 2 & -6 & -2 \end{array}\right] \text{R3 -> R3-2R2} \left[\begin{array}{ccc|c} 1 & 0 & 2 & 3 \\ 0 & 1 & 3 & 4 \\ 0 & 0 & -12 & -10 \end{array}\right]$$

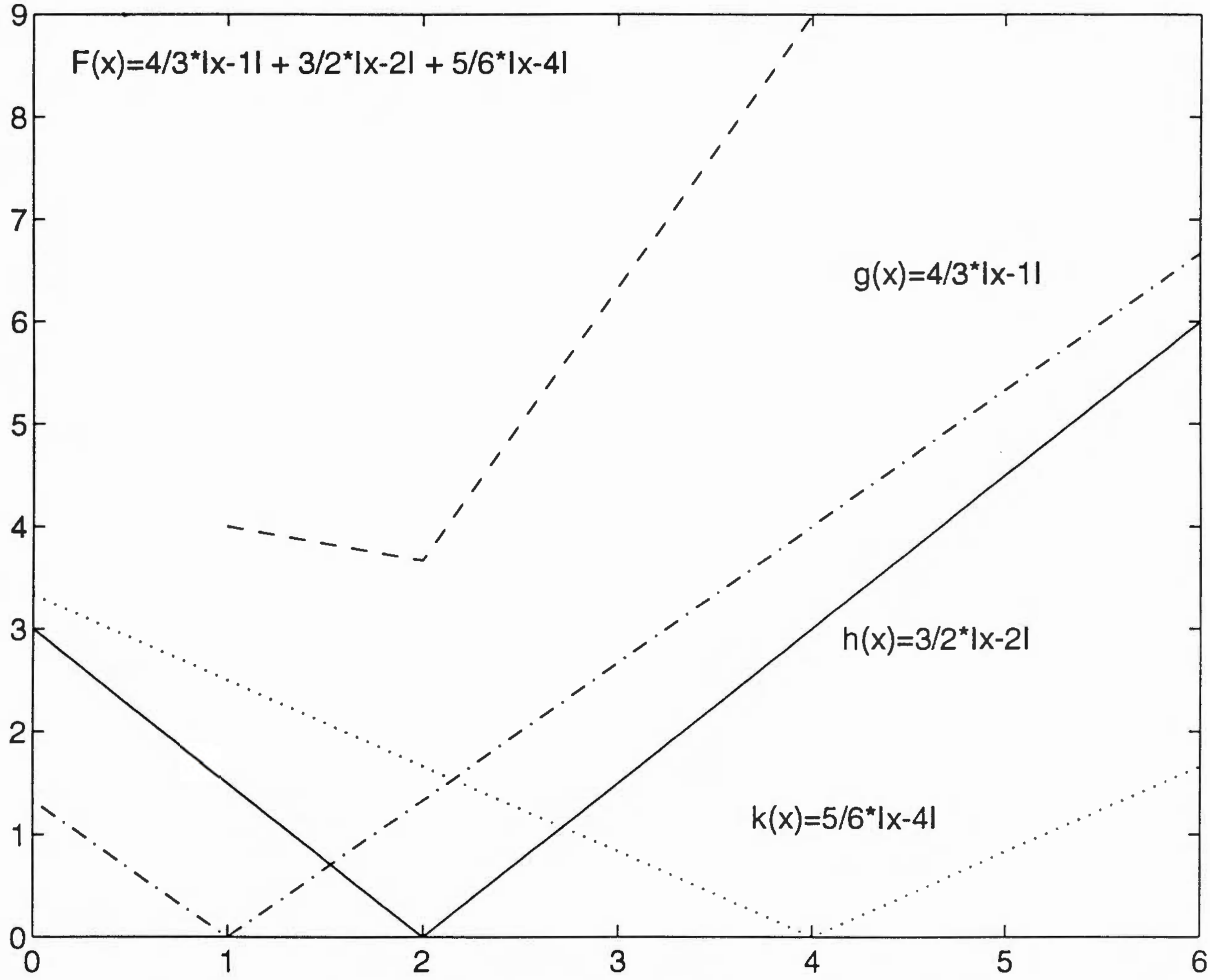
Now

$$\begin{aligned} -12c_3 &= -10 & \qquad c_3 &= 5/6 \\ c_2 + 3c_3 &= 4 & c_2 &= 3/2 \\ c_1 + 0c_2 + 2c_3 &= 3 & c_1 &= 4/3 \end{aligned}$$

$$F(x) = 4/3|x-1| + 3/2|x-2| + 5/6|x-4|$$

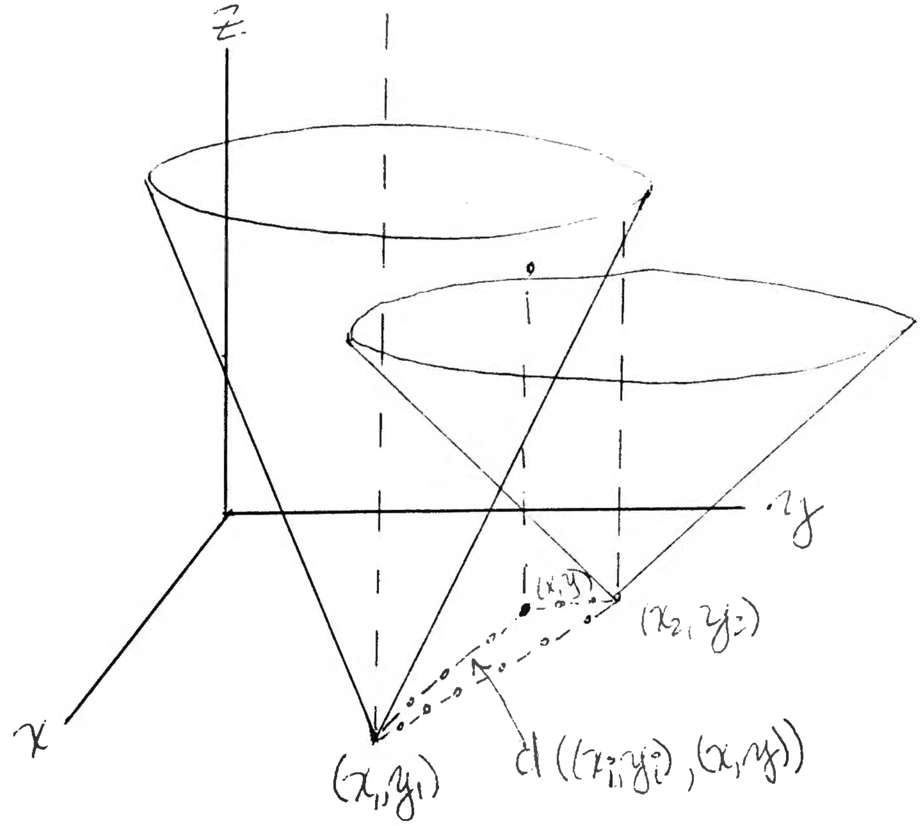
Let $g(x) = 4/3|x-1|$, $h(x) = 3/2|x-2|$, $k(x) = 5/6|x-4|$.

We graph each separate term gives a physical representation of the interpolation. Graphing F(x) along with the terms illustrates that it piecewise linear. We expect this result since the piecewise linear functions are a subspace of C[1,4] and the addition of the lines, as shown in the graph, forms a line. The function is not continuous at the data points since we only require that F(x)=f(x) for the given data.



Example 1. pointwise addition to get linearly piecewise approximation

Figure 4.



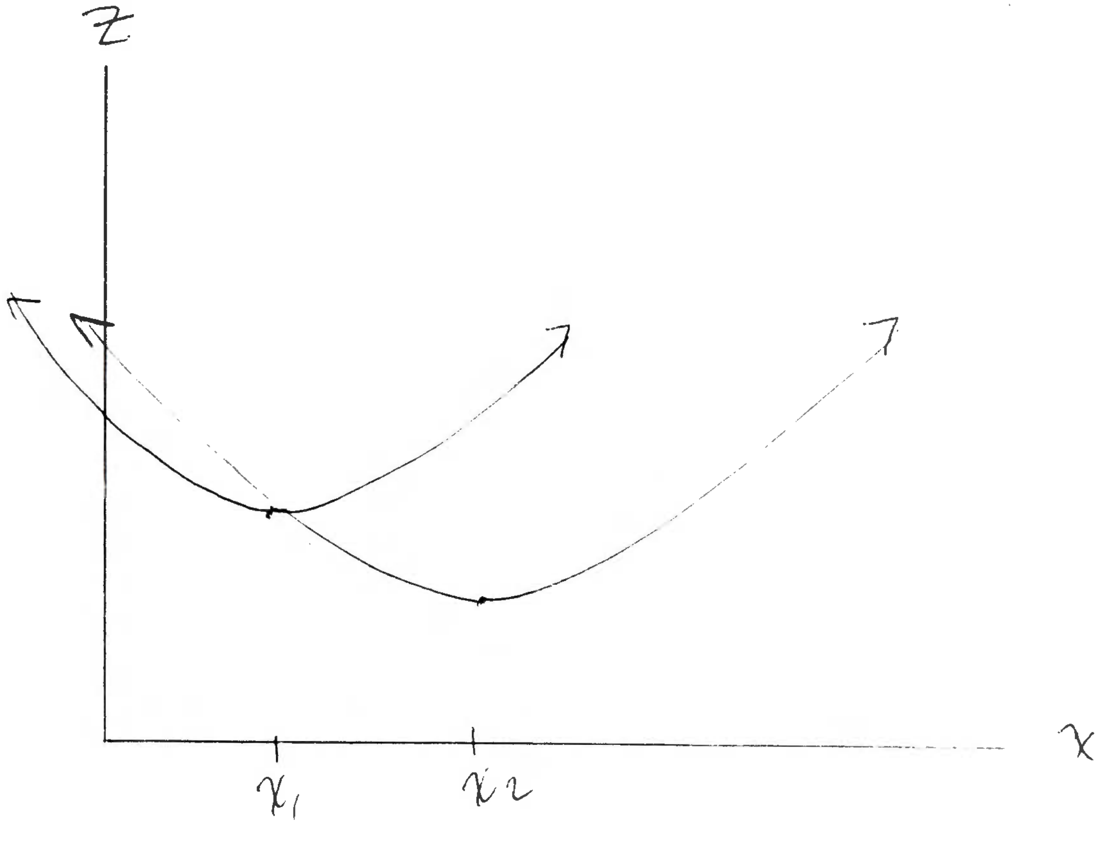


Figure 5.

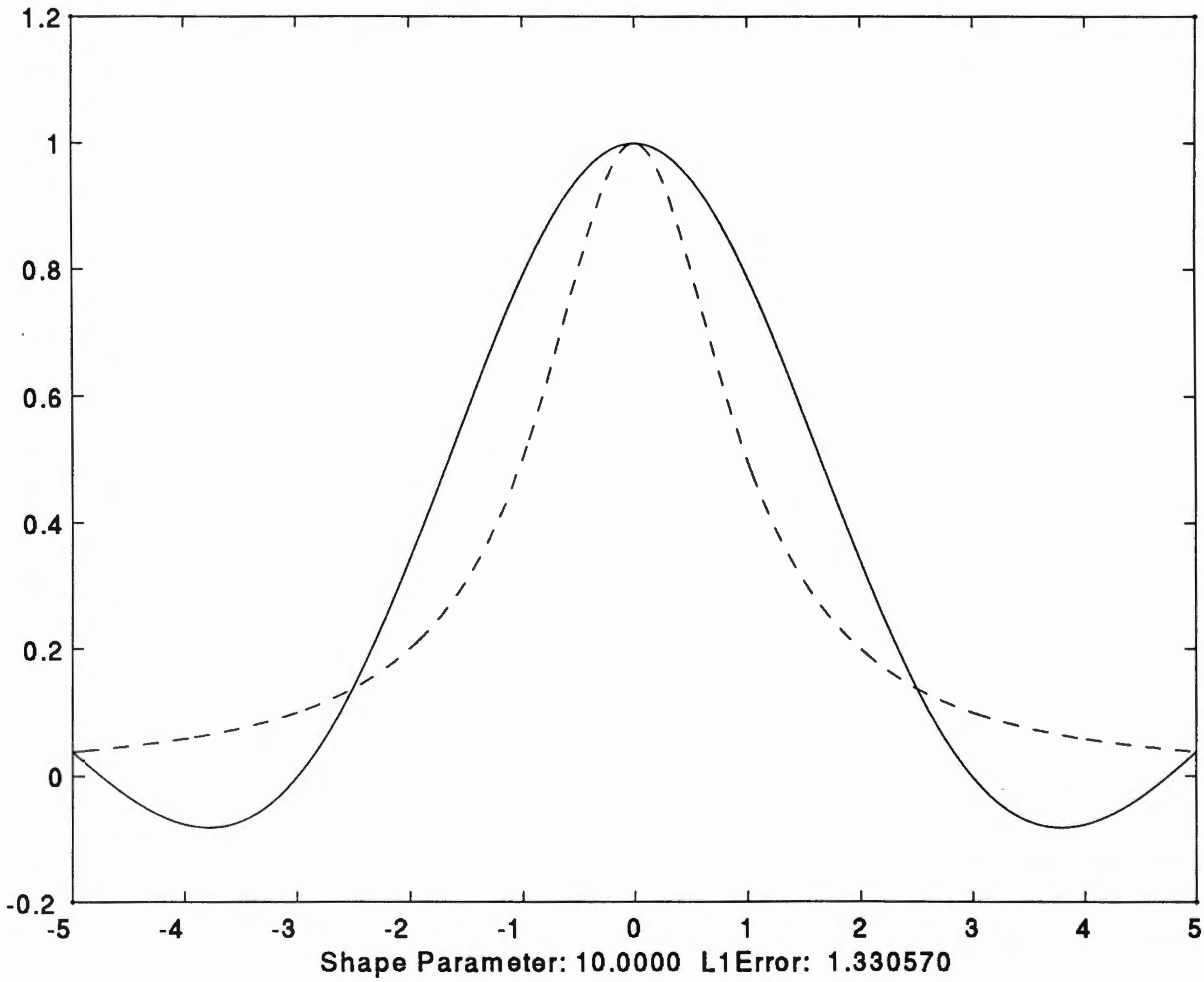

1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-5
-4
-3
-2
-1
0
1
2
3
4
5
Shape Parameter: 10.0000 L1Error: 1.330570

Figure 6.

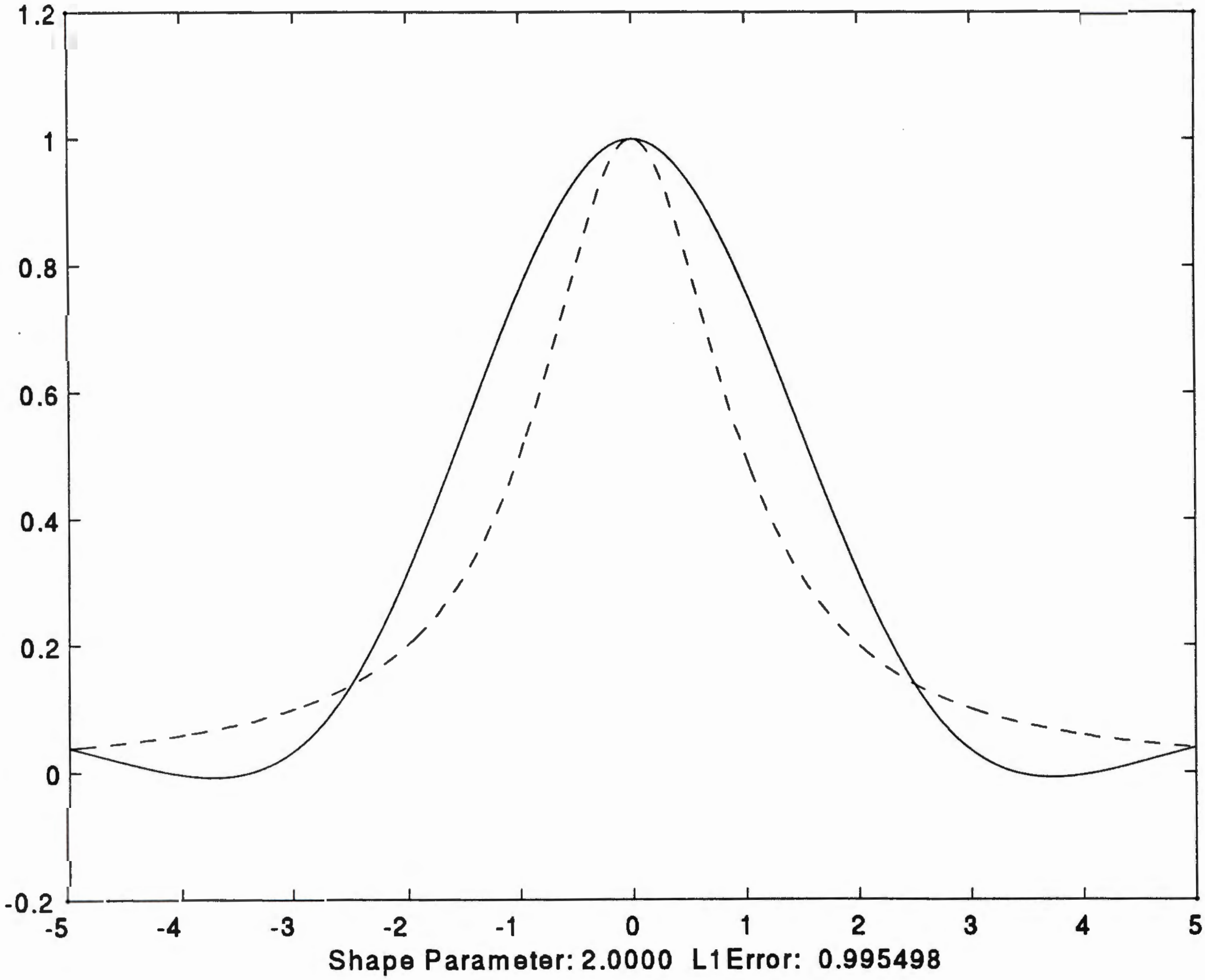
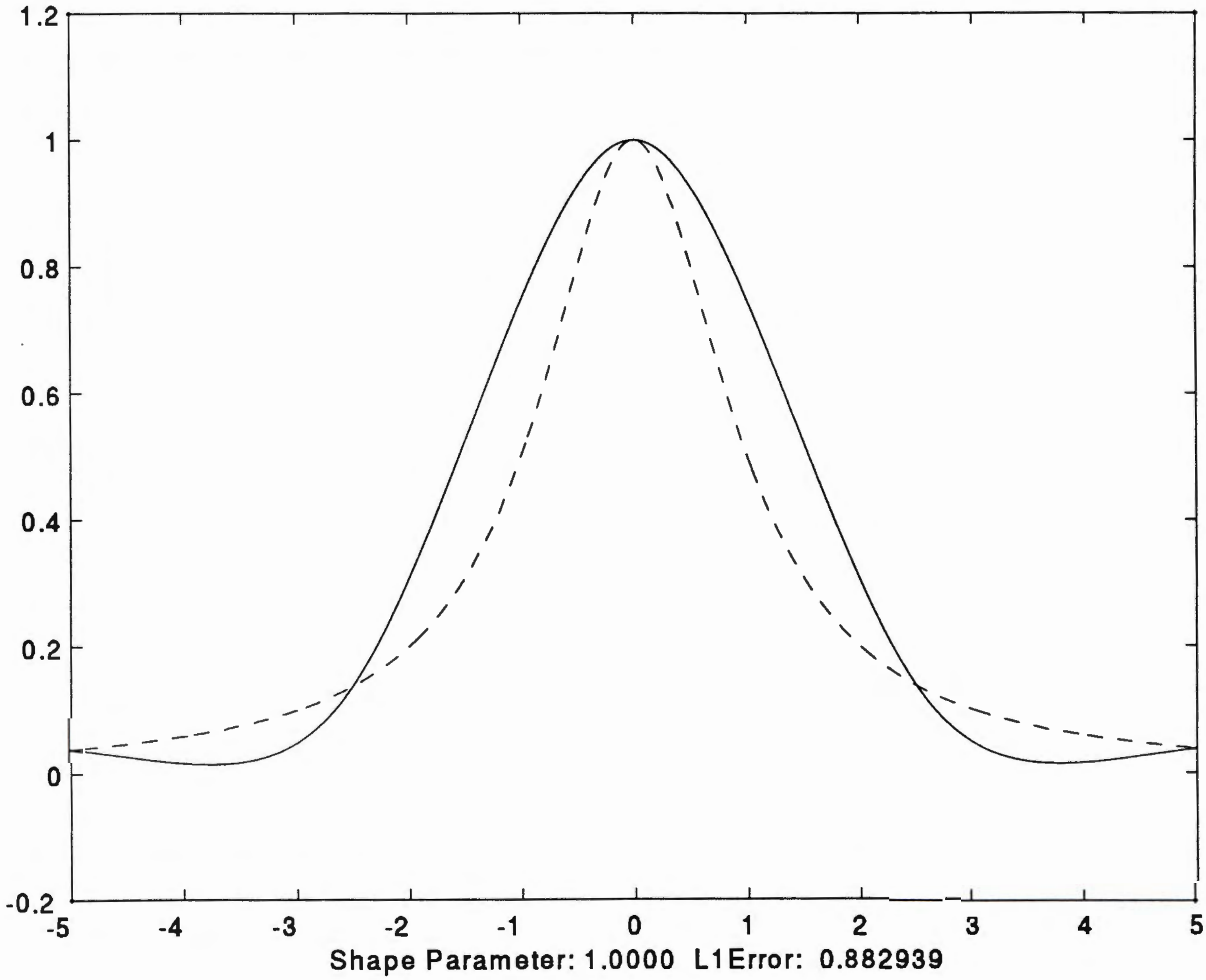
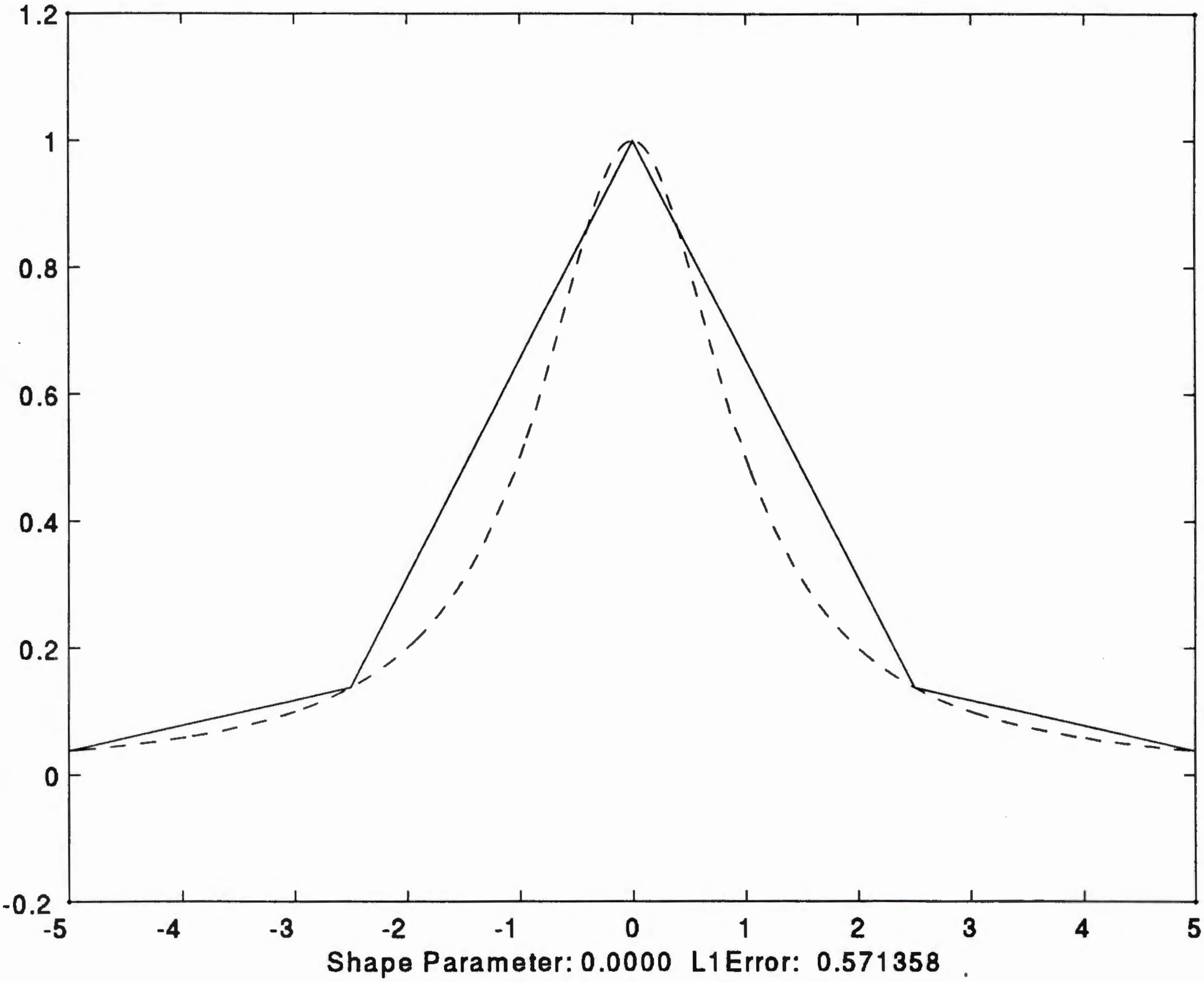

1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-5
-4
-3
-2
-1
0
1
2
3
4
5
Shape Parameter: 2.0000 L1Error: 0.995498

Figure 7



1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-5
-4
-3
-2
-1
0
1
2
3
4
5
Shape Parameter: 1.0000 L1Error: 0.882939

Figure 8.


1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-5
-4
-3
-2
-1
0
1
2
3
4
5
Shape Parameter: 0.0000 L1Error: 0.571358

Fiqure 9.

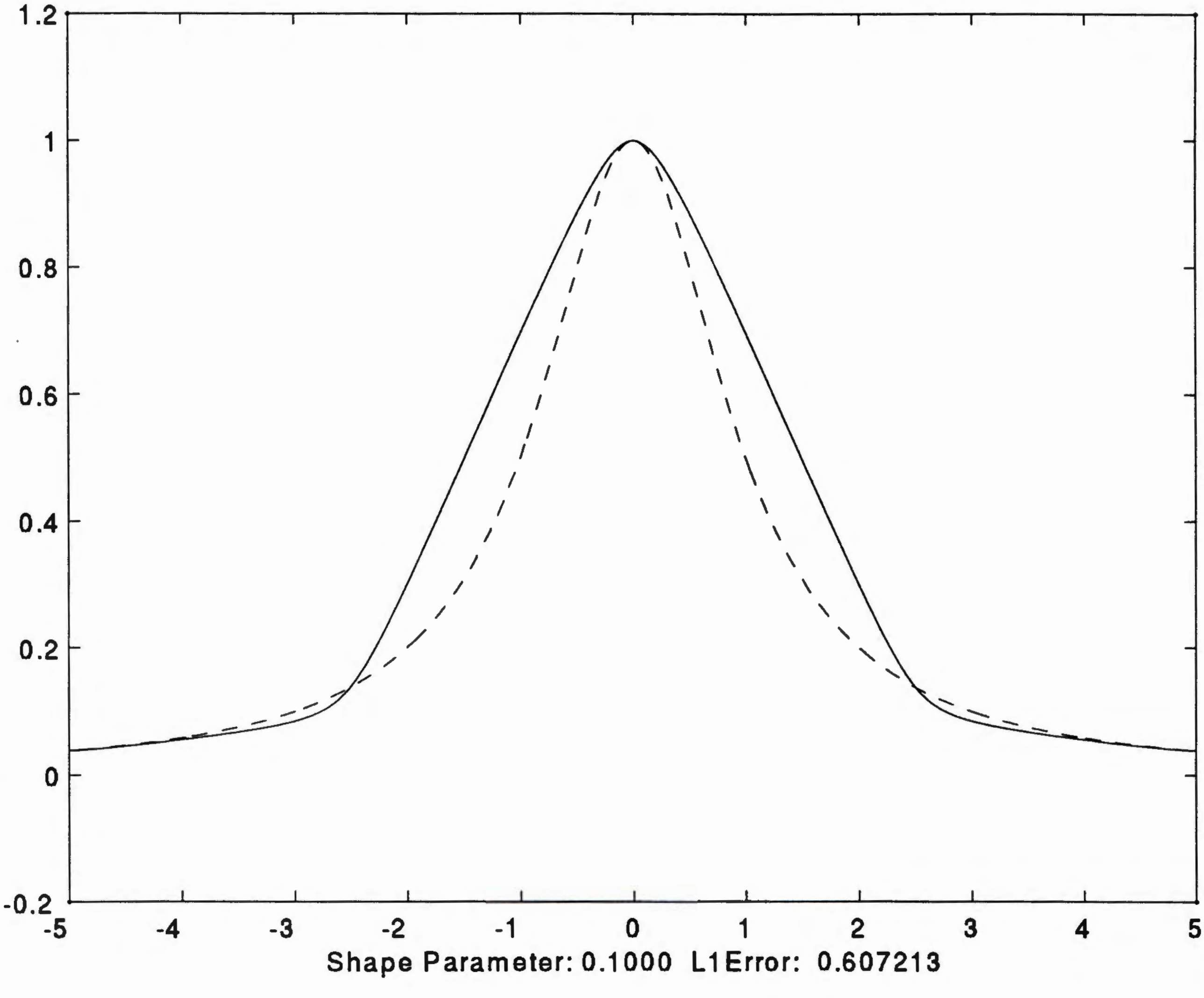

1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-5
-4
-3
-2
-1
0
1
2
3
4
5
Shape Parameter: 0.1000 L1Error: 0.607213

Figure 10.

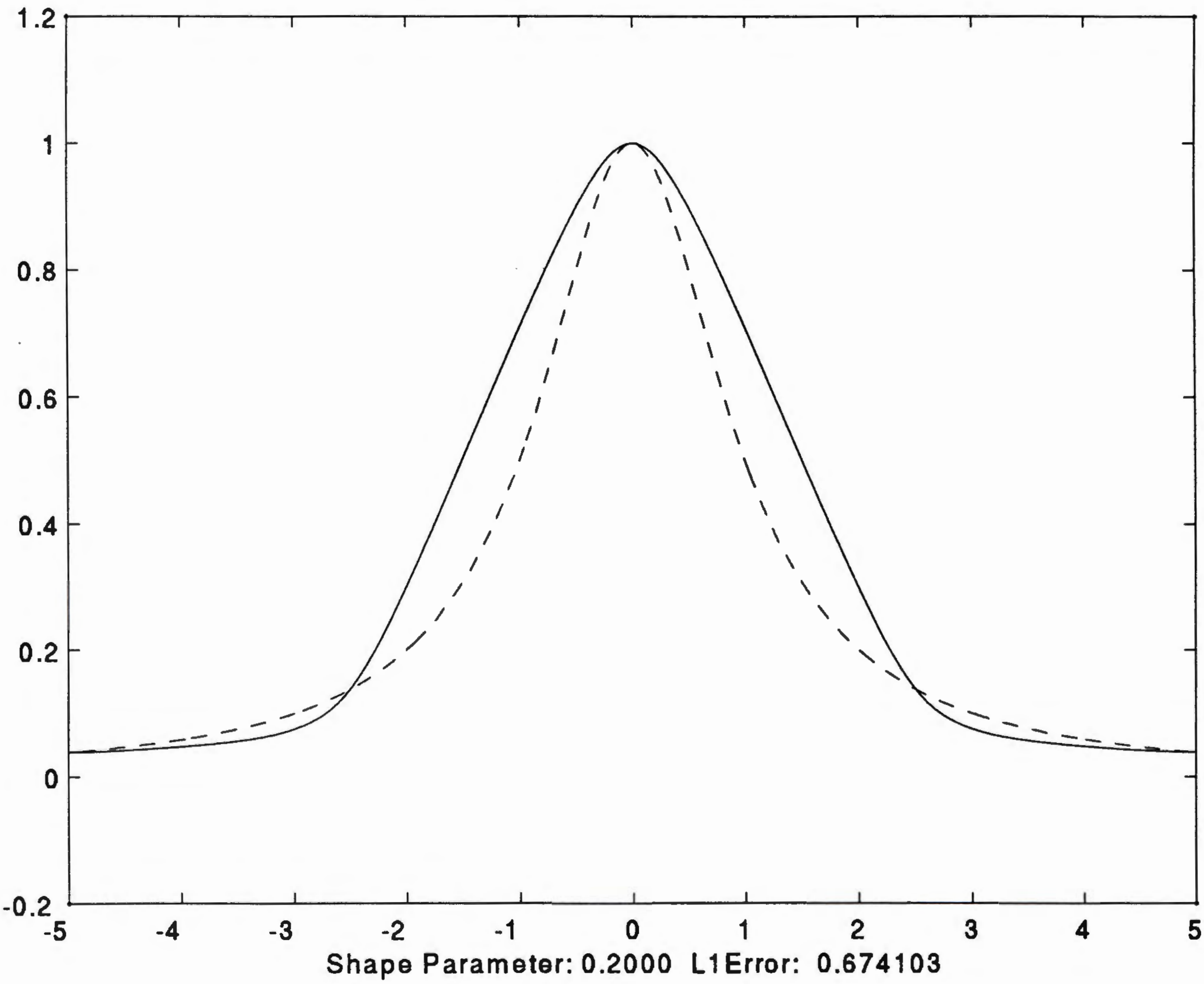

1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-5
-4
-3
-2
-1
0
1
2
3
4
5
Shape Parameter: 0.2000 L1Error: 0.674103

Figure 11.

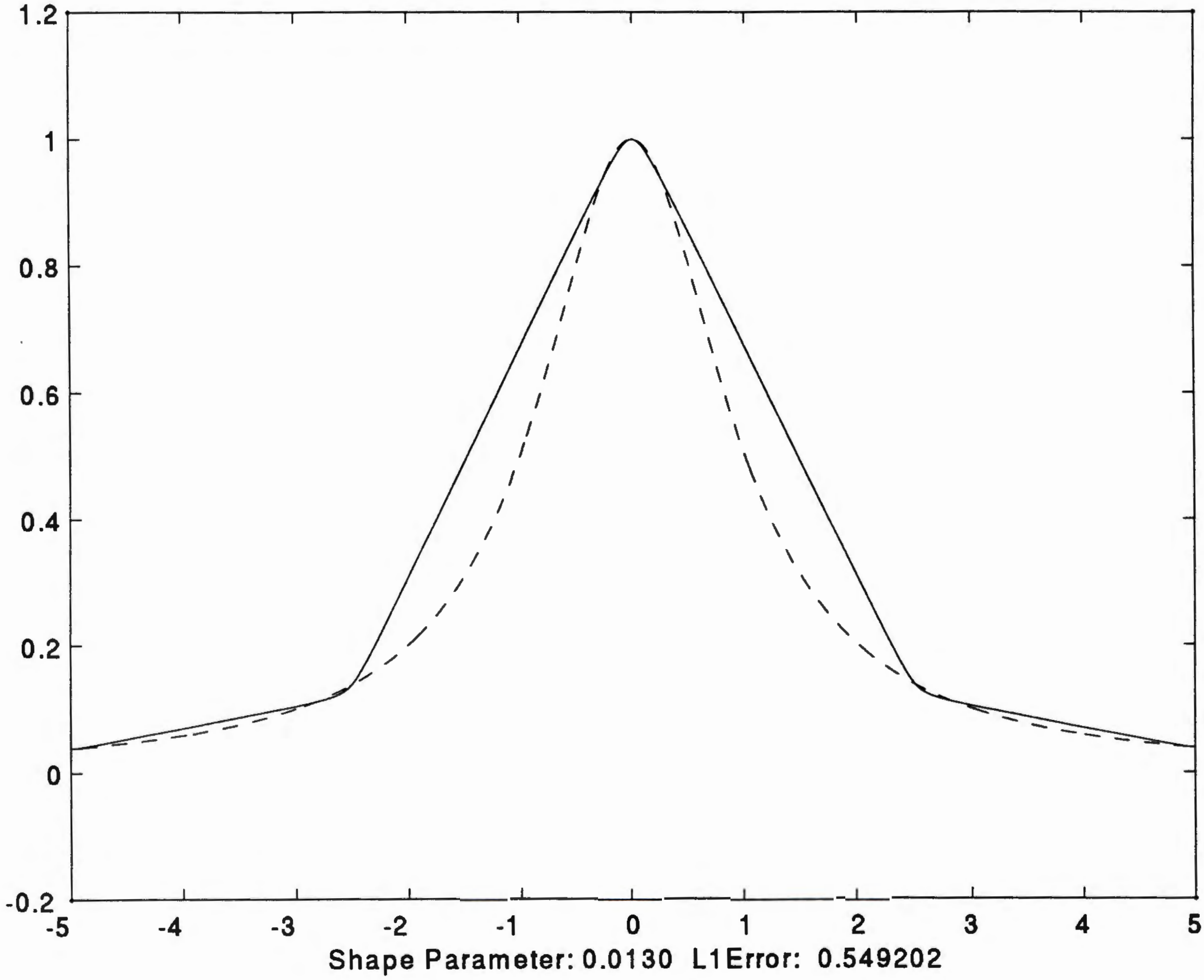

1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-5
-4
-3
-2
-1
0
1
2
3
4
5
Shape Parameter: 0.0130 L1Error: 0.549202

Figure 12.

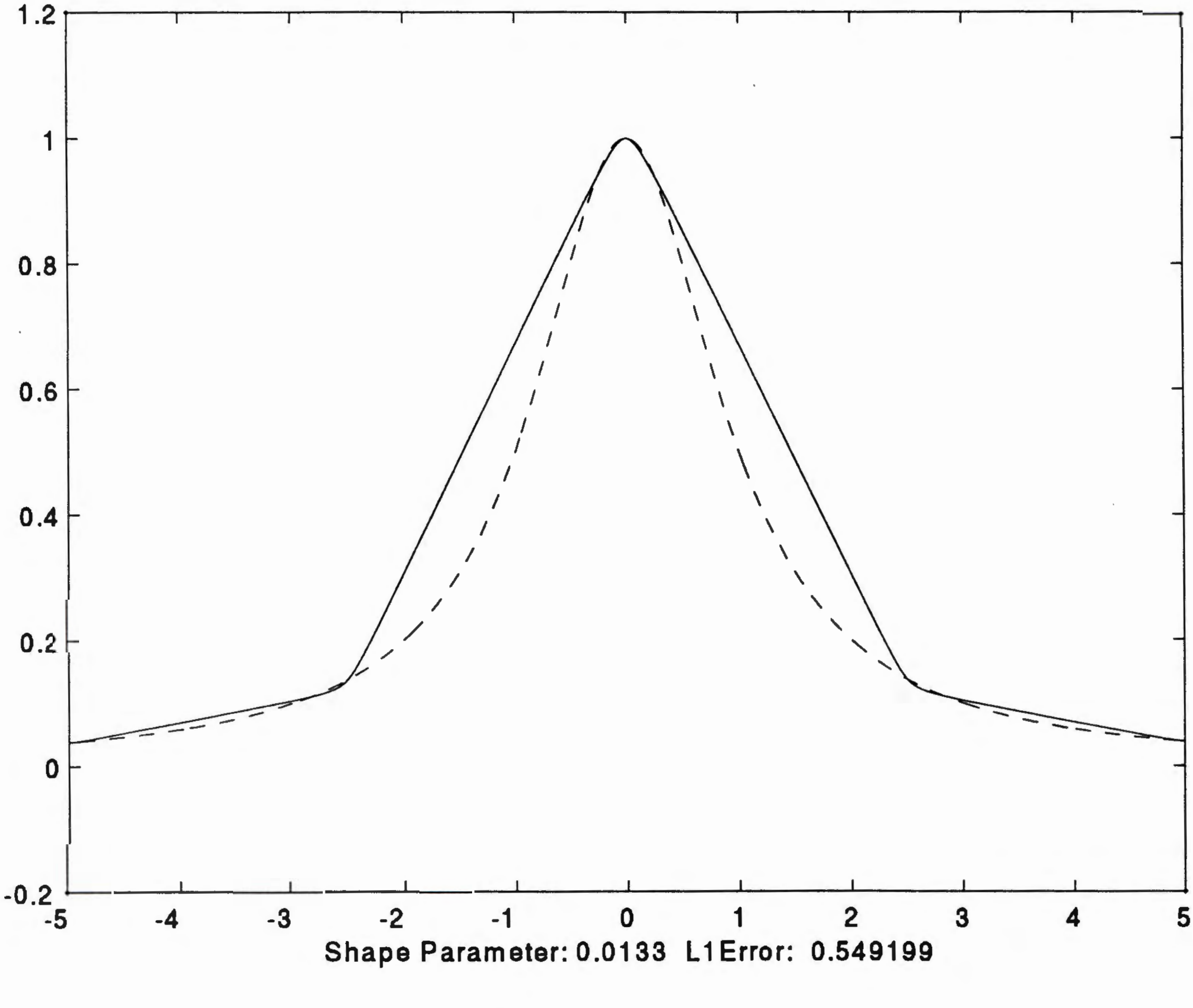

1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-5
-4
-3
-2
-1
0
1
2
3
4
5
Shape Parameter: 0.0133 L1Error: 0.549199

Figure 13.

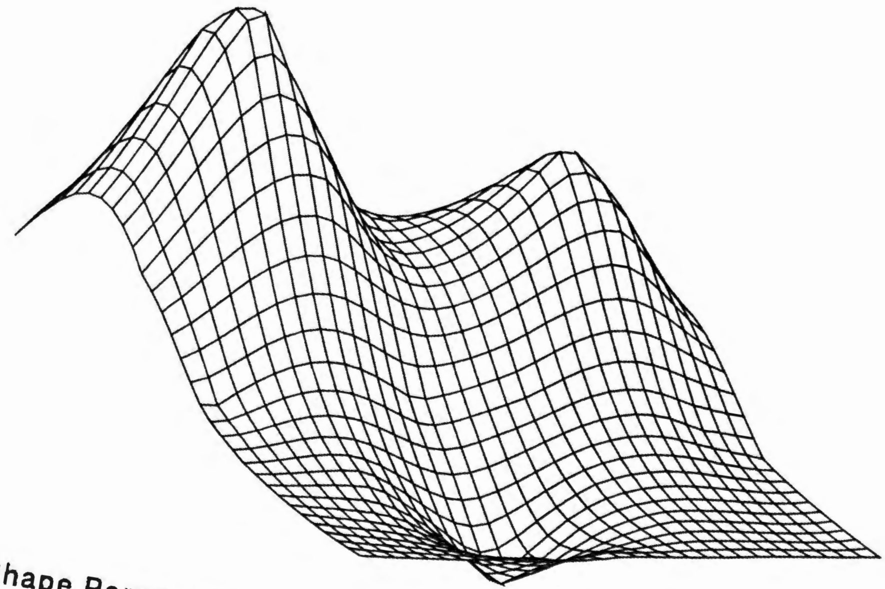
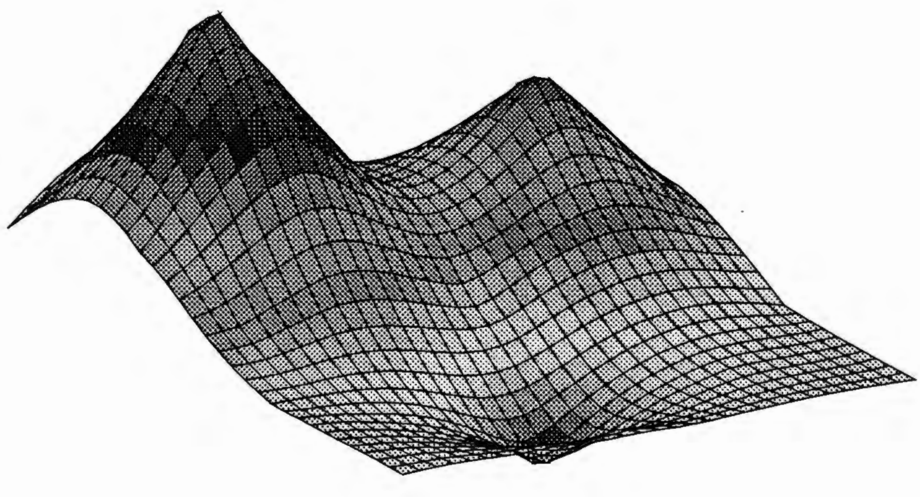
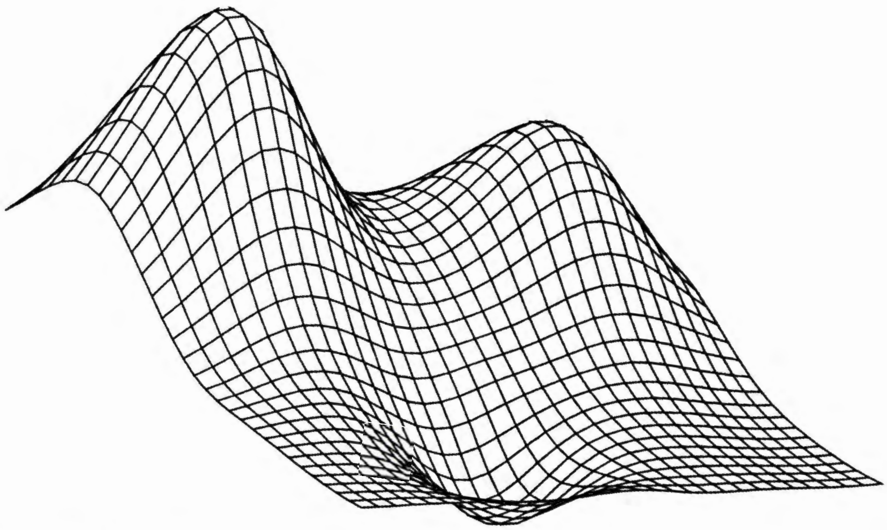


Figure 14a.



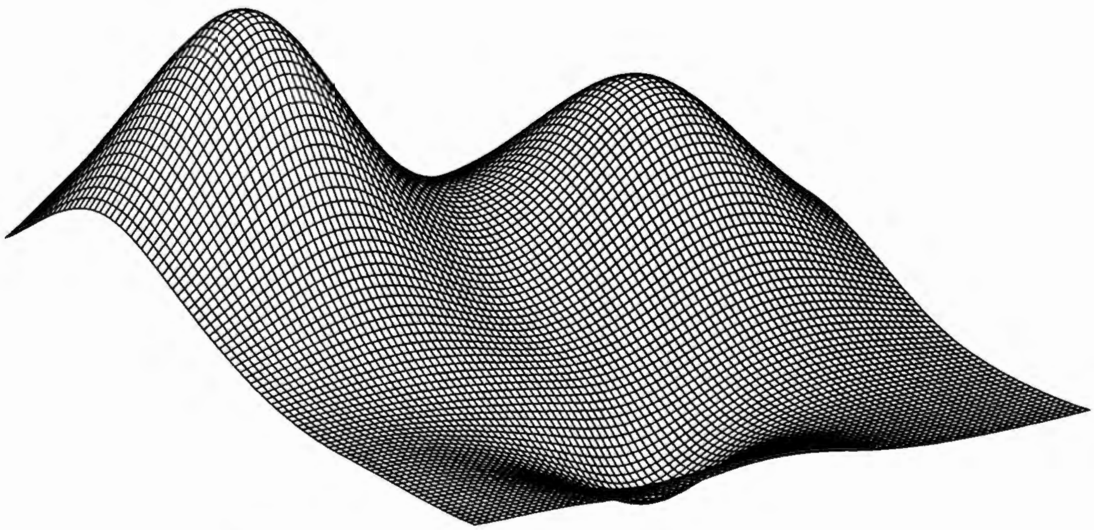
Shape Parameter: 0   L1 Error: 224.6

Figure 14c.



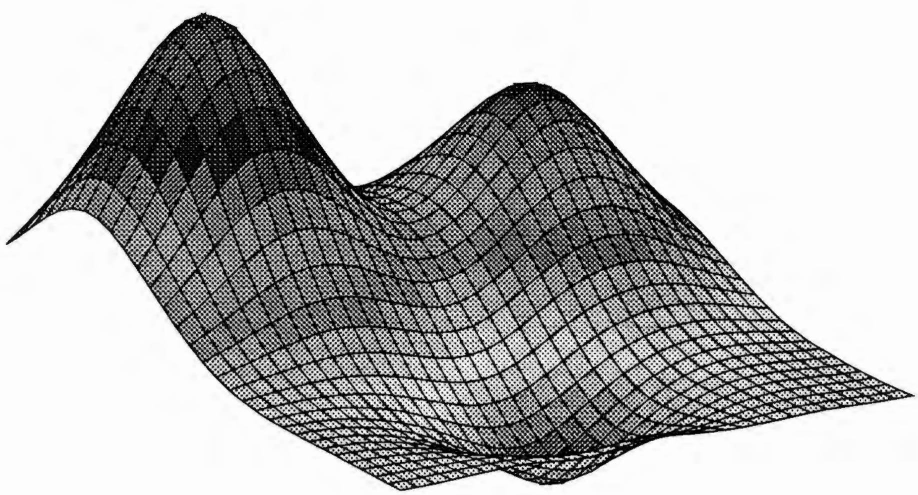
Shape Parameter: 1   L1 Error: 118

Figure 15a.



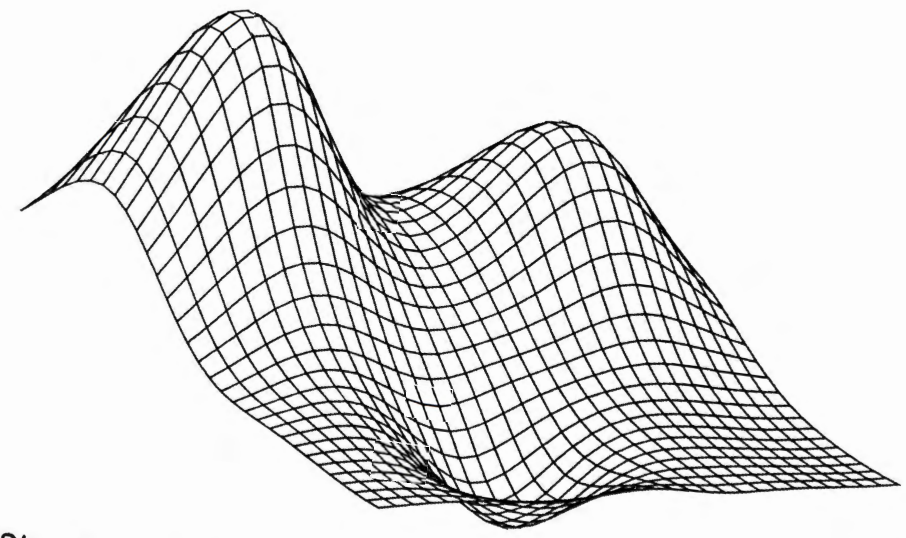
Shape Parameter: 1   L1 Error: 1071

Figure 15b.



Shape Parameter: 1 L1 Error: 133

Figure 15c.



Shape Parameter: 2   L1 Error: 119.6

Figure 16a.

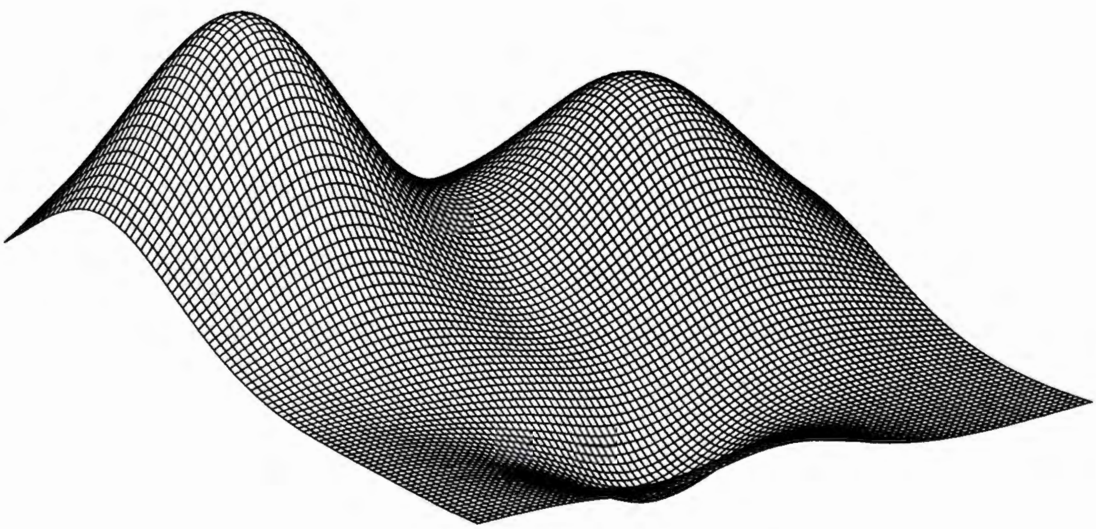
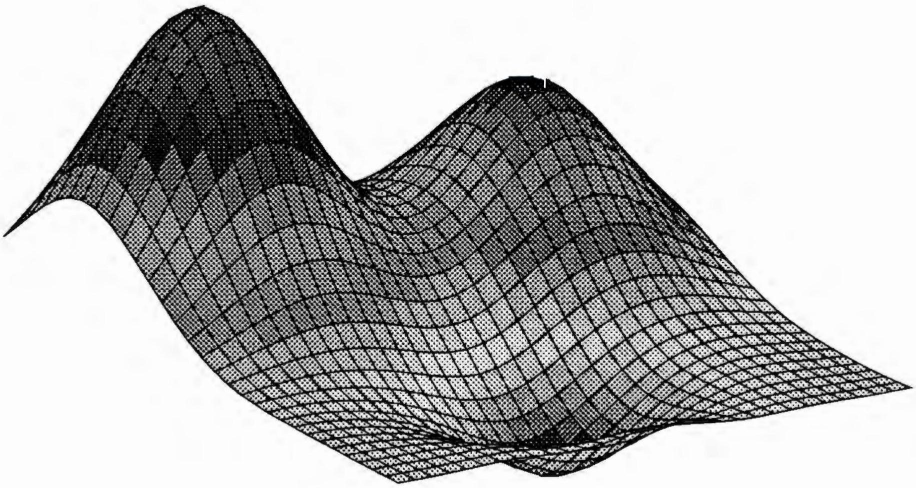
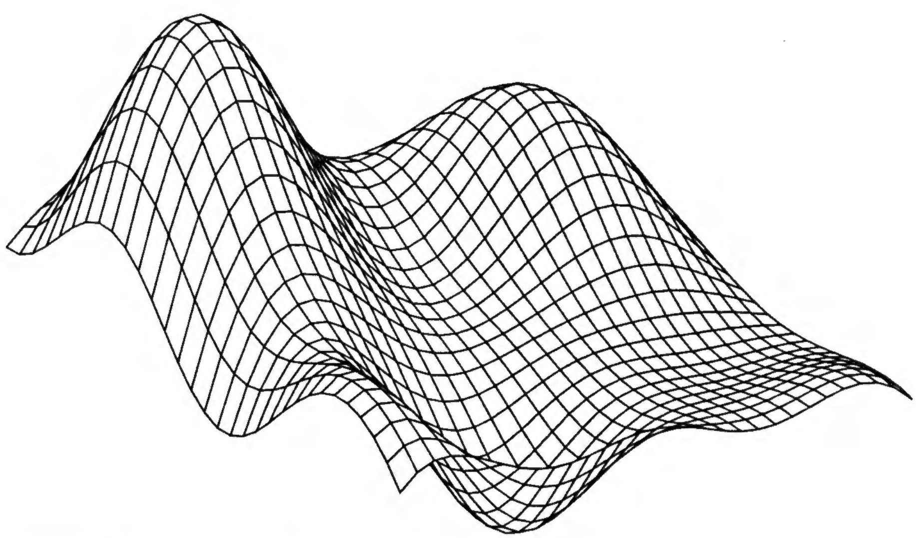


Figure 16b.



Shape Parameter: 2 L1 Error: 136.2

Figure 16 c.



Shape Parameter: 40 L1 Error: 396

Figure 17a.

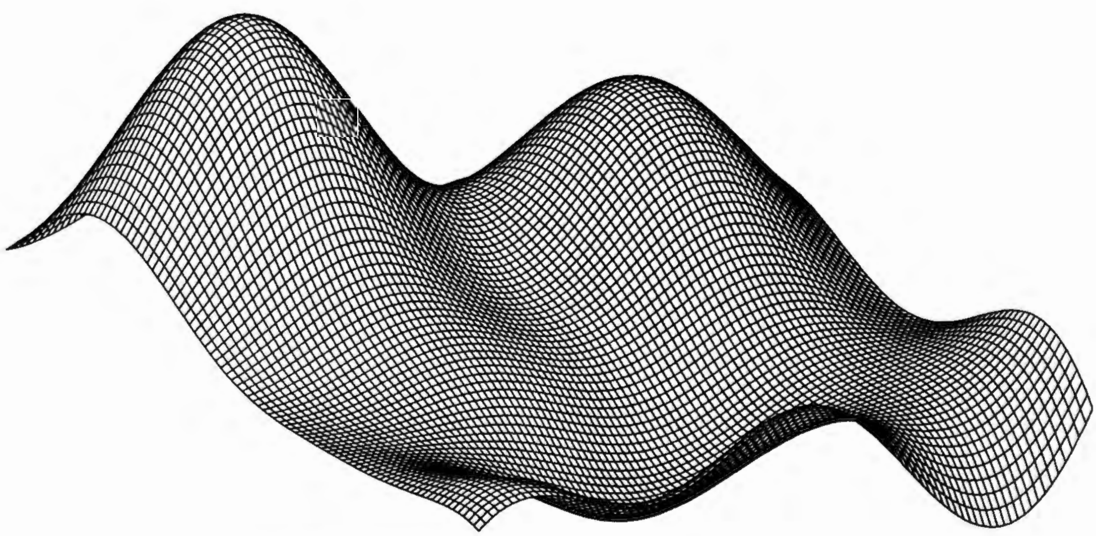
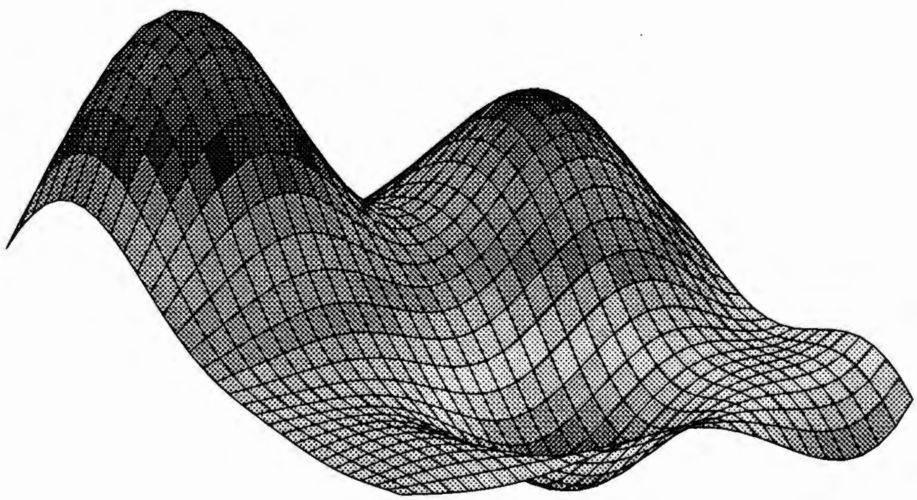
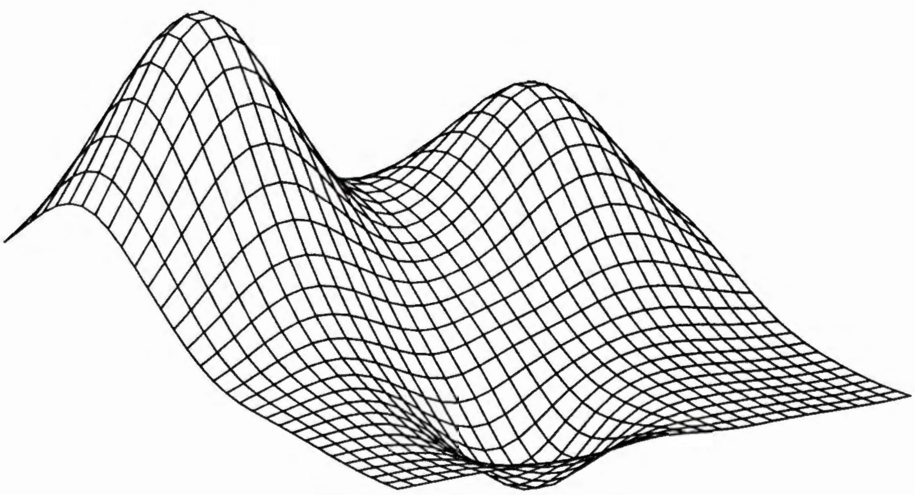


Figure 17b.



Shape Parameter: 40 L1 Error: 235.5

Figure 17c.



Shape Parameter: 1.3   L1 Error: 116.9

Figure 18a.

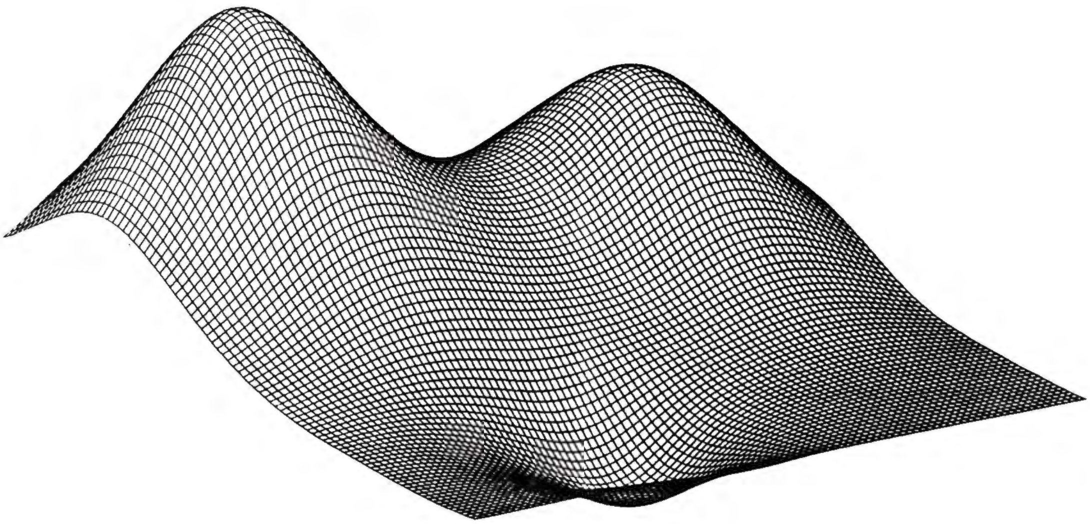
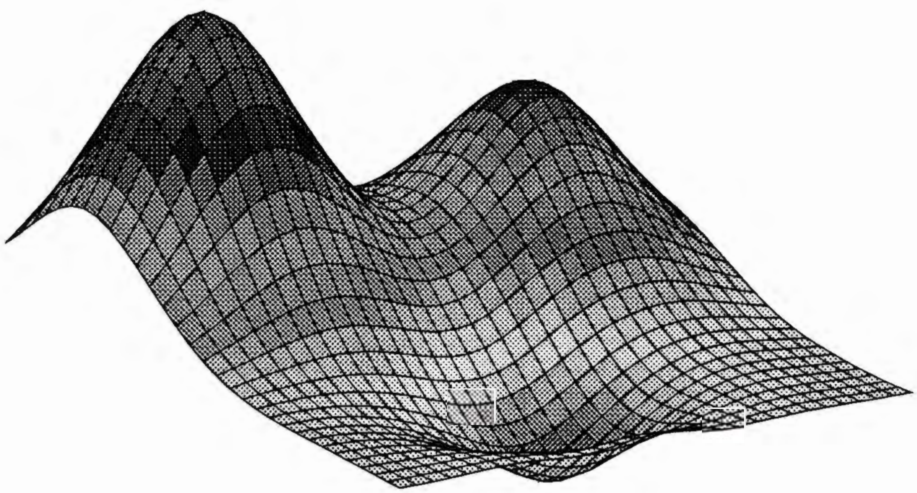
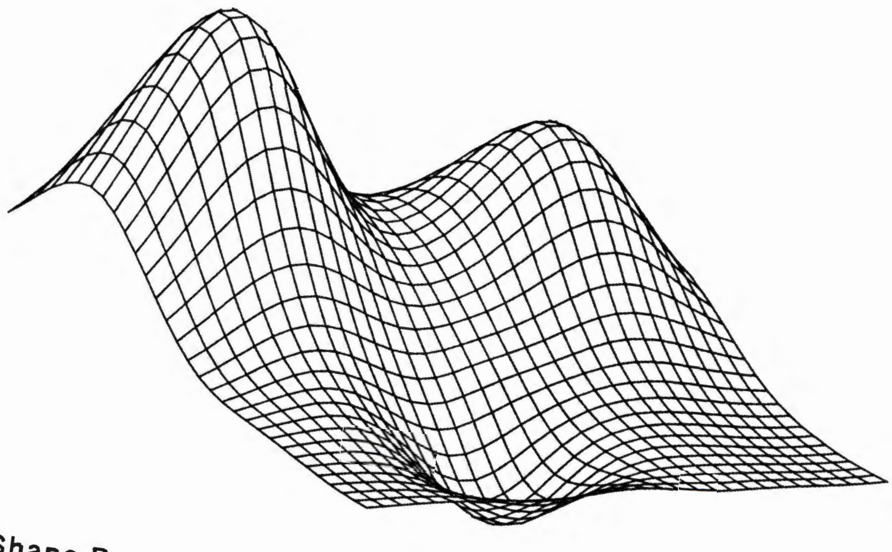


Figure 18b.



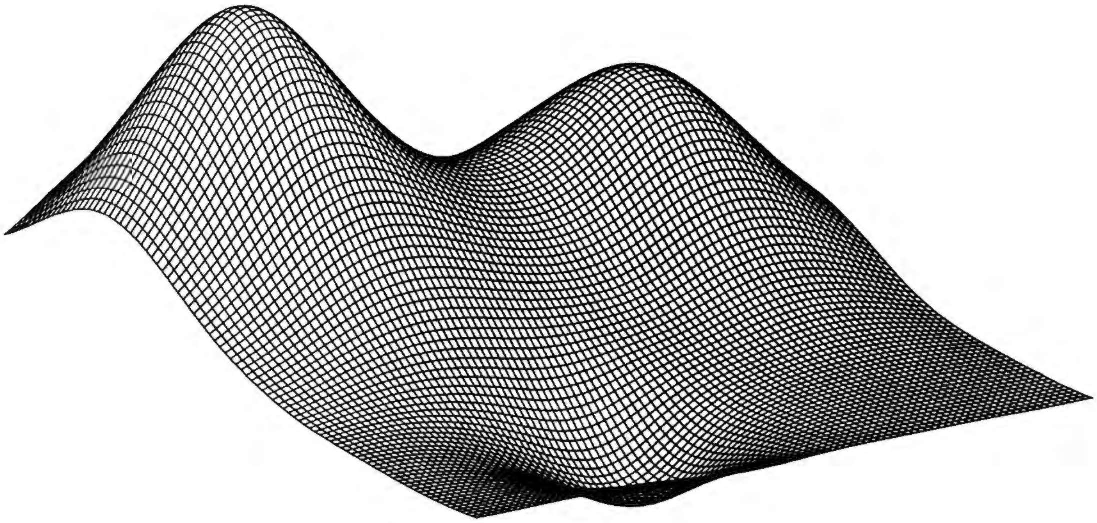
Shape Parameter: 1.3   L1 Error: 131.3

Figure 18c.



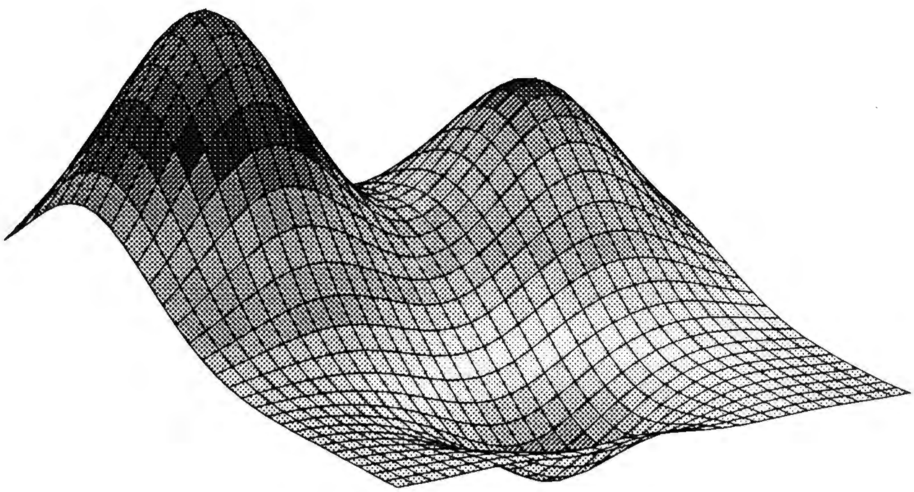
Shape Parameter: 1.33   L1 Error: 116.8

Figure 19a.



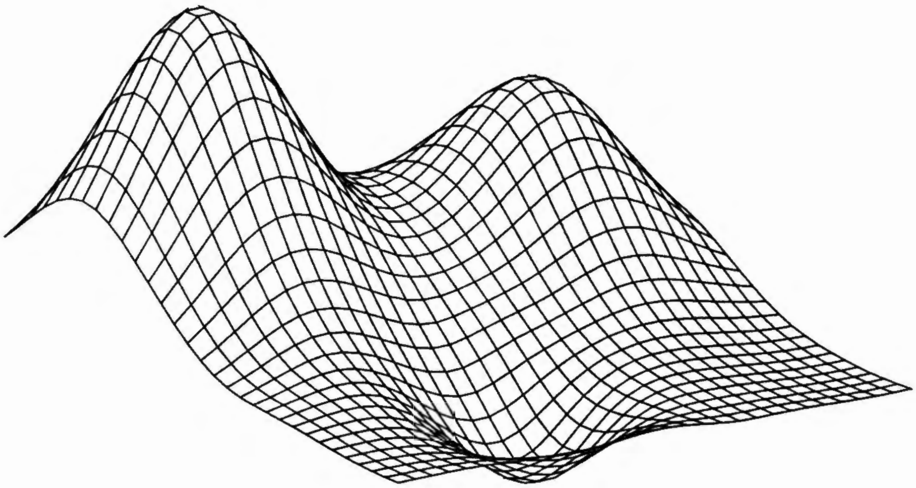
Shape Parameter: 1.33   L1 Error: 1716

Figure 19b.



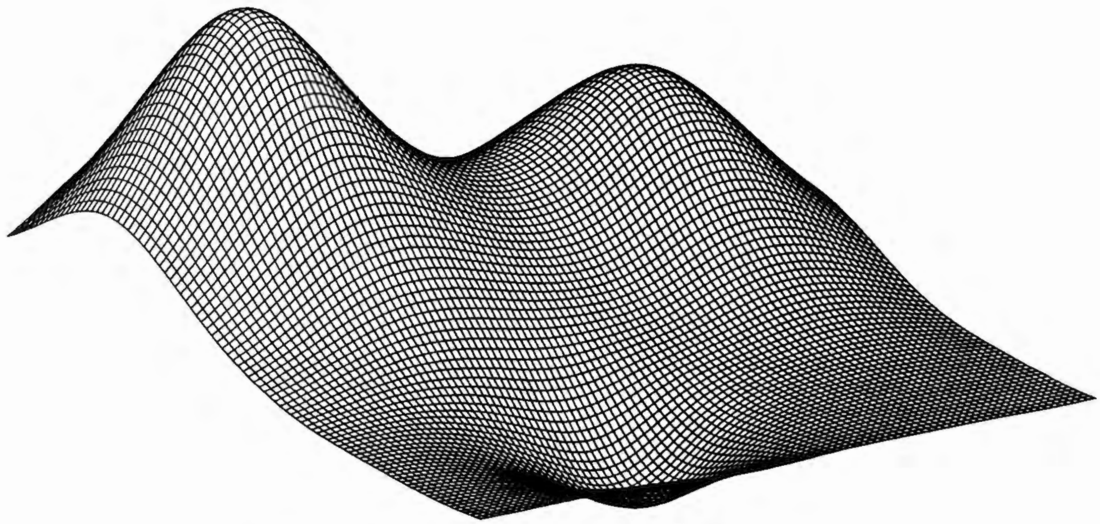
Shape Parameter: 1.33   L1 Error: 131.3

Figure 19c.



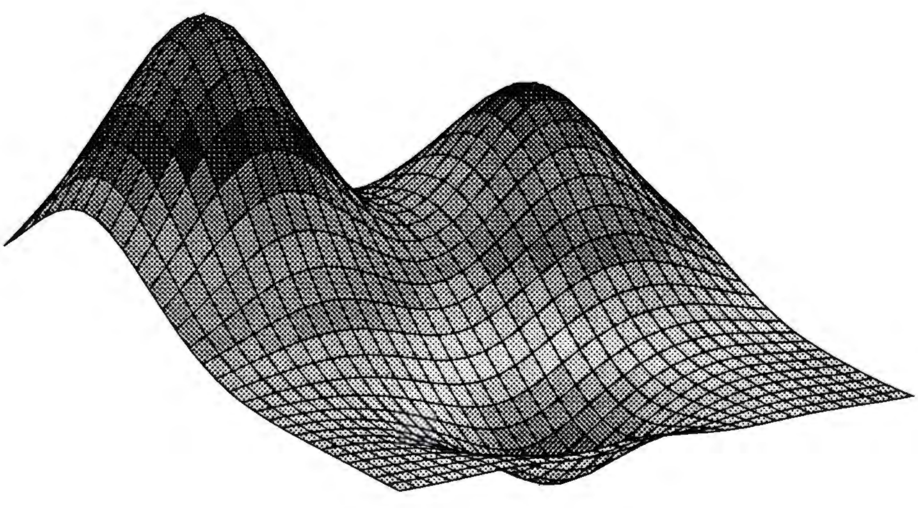
Shape Parameter: 1.339   L1 Error: 116.8

Figure 20a.



Shape Parameter: 1.339   L1 Error: 1716

Figure 20b.



Shape Parameter: 1.339   L1 Error: 131.3

Figure 20c.

Acknowledgments

I would like to thank the University of Northern Iowa and the Presidential Scholars Board for the opportunity to be a part of the Presidential Scholars Program, which I found challenging, as well as enjoyable. I would also like to thank the National Center for Supercomputing Applications for grant number TRA930381N which aided my research and data gathering. I cannot thank Dr. Timothy Hardy enough for all of his help, input, encouragement and patience. Thank you.

References

1. Leonard M. Blumenthal, Theory and Applications of Distance Geometry, 2nd edition Chelsea Publishing Company, New York (1970).

2. Richard L. Burden and J. Douglas Faires, Numerical Analysis, 5th edition PWS Publishing Company, Boston (1993).

3. Richard Franke, "Scattered Data Interpolation: Tests of Some Methods", Mathematics of Computation, vol. 38, pp181-200 (1982).

4. Charles A. Micchelli, "Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions", Constructive Approximate Theory, vol. 2, pp11-22 (1986).

5. Audry Ellen Tarwater, "A Parameter Study of Hardy's Multiquadric Method for Scattered Data Interpolation", M.S. Thesis University of California, Davis (1985).