# INTERPOLATION FROM A CLOUD OF POINTS

Timothy J. Baker

*Dept. of MAE, Princeton University, Princeton, NJ 08544, U.S.A. baker@tornado.princeton.edu*

## ABSTRACT

Let $V = \{P_1, P_2, \ldots, P_n\}$ be a set of points in either 2D or 3D space and let $\{q_1, q_2, \ldots, q_n\}$ be scalar values associated with the points. This paper presents a method for interpolating values of the scalar variable $q$ at any position $X$ in the convex hull of $V$. The interpolant consists of the sum of the linear interpolant for a simplex $T$ that contains $X$ and a least squares estimate of the higher order terms. The least squares fit is made through the cloud of $m$ points in $V$ that are closest to $X$ and are not already vertices of $T$. Conditions that determine the invertibility of the least squares system are examined and related to geometric constraints on the positions of points in the cloud.

**Keywords: interpolation, solution projection, least squares estimation**

## 1. INTRODUCTION

Data interpolation from a discrete set of points in either 2D or 3D space is required in many situations that arise when solving partial differential equations on a mesh of points. Time dependent problems, for example, often require the projection of a solution at a given time instant from one mesh onto a new, modified mesh that has been adapted for the computation at the next time step. Another example arises with the overset method which uses several overlapping meshes to represent a given domain with a consequent need to transfer the solution data between the component meshes. Graphical interpretation and feature detection for large data sets such as flowfield measurements also relies on accurate interpolation from scattered data [1]. If the donor mesh possesses a high degree of regularity (*i.e.* with a locally well defined set of coordinate directions) then interpolation by tensor splines or transfinite interpolation is possible. If the donor mesh consists of arbitrarily placed points, however, it will be necessary to devise data interpolation or reconstruction schemes that do not make any *a priori* assumptions about the underlying mesh geometry.

This problem has received much attention among protagonists of meshless methods [2, 3, 4, 5, 6] (also known as *hp* clouds and partition of unity methods). Data estimation is typically carried out by a moving least squares method [7, 8] which assigns an influence function to each mesh point. The domain of influence, over which this function is non-zero, extends a finite distance from the mesh point. Data reconstruction at any given position in space is then achieved by summing the contributions from of all domains of influence that enclose the particular position.

The method proposed in this paper starts from a representation of the interpolated value as a linear interpolant over a simplex whose vertices are points in the donor mesh. The linear interpolant is then augmented by a higher order estimate that is obtained from nearest point neighbors outside this simplex. Data projection is specifically designed to interpolate the exact value at each donor mesh point.

## 2. PROBLEM STATEMENT AND FORMULATION

Given a set of randomly distributed points $V = \{P_1, P_2, \ldots, P_n\}$ with associated scalar values $\{q_1, q_2, \ldots, q_n\}$ interpolate a value $q(\mathbf{x})$ at any given position $\mathbf{x}$ within the convex hull of $V$. It is required that $q(\mathbf{x}_i) = q_i,\ i = 1, \cdots, n$ where $\mathbf{x}_i$ is the position vector associated with point $P_i$.

Let $T$ be a containing simplex whose vertices are points in $V$ and such that $T$ contains the point $\mathbf{x}$. If there is a triangulation associated with the donor mesh points $V$ then this can be searched to find the unique simplex containing $\mathbf{x}$. If no triangulation has been defined, a containing simplex can be constructed by searching through the point cloud until a suitable set of vertices has been found. In either case, a fast search procedure (*e.g.* use of an octree data structure [9]) will enable the closest point $P \varepsilon V$ to be found in $O(log\ n)$ time where $n = card\ V$. If a triangulation of $V$ has been defined then it is possible to find the containing simplex in an additional $O(1)$ time. If no *a priori* triangulation of the convex hull of $V$ is given then one may first create a tringulation of $V$, a procedure that is reasonable if the set $V$ of mesh points is not too large. If $n = card\ V$ is extremely large and the number of positions at which interpolated values are needed is relatively small, it may be preferable to create a containing simplex for each interpolated position $\mathbf{x}$ by a gift wrapping procedure.

Let

$$q(\mathbf{x}) = q_{lin}(\mathbf{x}) + f(\mathbf{x}) \tag{1}$$

where $q_{lin}(\mathbf{x})$ is the interpolant obtained by a linear fit through the vertices of the containing simplex $T$. The function $f(\mathbf{x})$ is a higher order estimate of the error between the true function value and the linear interpolant. This estimate is obtained from a least sqaures fit through the nearest neighbors among the point set $V$. The interpolation can be carried out to arbitrarily high order. Although continuity of the derivatives is not guaranteed, tests of the reconstruction procedure indicate that the requisite degree of smoothness is obtained in practice.

## 2.1 Linear Interpolant

Considering first the planar case, let the containing triangle $T$ be defined by the vertices $R_1, R_2, R_3 \ \varepsilon \ V$ and let $X$ be the point with coordinates $(x, y)$ at which an interpolated value $q(x, y)$ is required. Let $R_j$ have coordinates $(x_j, y_j)$, $j = 1, 2, 3$ and define the linear basis functions $\phi_1, \phi_2, \phi_3$ such that $\phi_i(x_j, y_j) = \delta_{ij}$, $i, j = 1, 2, 3$ where $\delta_{ij}$ is the Kronecker delta

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{2}$$

The basis functions correspond to the barycentric coordinates associated with the triangle $T$. Thus $\phi_1(x, y) = A_1/A$ where (see figure 1) $A_1$ is the area of triangle $XR_3R_2$, with $A_2$ and $A_3$ defined in a similar way, and $A = A_1 + A_2 + A_3$ is the area of triangle $R_1R_3R_2$. The linear interpolant is then given by

$$q_{lin}(x, y) = q_1\phi_1(x, y) + q_2\phi_2(x, y) + q_3\phi_3(x, y) \tag{3}$$

Since the linear interpolant must be exact if $q(x, y)$ is a constant, it follows that

$$\phi_1(x, y) + \phi_2(x, y) + \phi_3(x, y) = 1 \tag{4}$$

Similarly, the requirement that $q(x, y) = x$ and $q(x, y) = y$ be represented exactly by the linear interpolant leads to the equations

$$x_1\phi_1(x, y) + x_2\phi_2(x, y) + x_3\phi_3(x, y) = x \tag{5}$$

and

$$y_1\phi_1(x, y) + y_2\phi_2(x, y) + y_3\phi_3(x, y) = y \tag{6}$$

It follows that explicit expressions for the basis functions can be determined by inverting the system of equations

$$\begin{pmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{pmatrix} = \begin{pmatrix} 1 \\ x \\ y \end{pmatrix} \tag{7}$$

The determinant of the above matrix equals twice the area of triangle $R_1R_3R_2$. Thus, the basis functions are well defined provided the area of the triangle is nonzero.

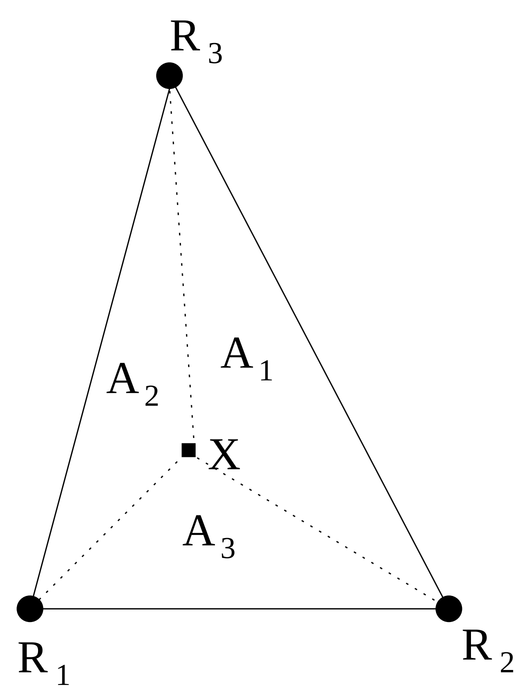


**Figure 1**: Definition of areas $A_1, A_2$ and $A_3$ associated with the point $X$.

## 2.2 Estimation of Quadratic Error Terms

In order for the interpolant to remain exact at the vertices of $T$, it is necessary to obtain an estimate of the quadratic error $f(x,y)$ that is zero at each vertex of $T$ (*i.e.* such that $f(x_j,y_j)=0,\ j=1,2,3$). This can be achieved in terms of the linear basis functions by representing the error as

$$f(x,y)=a\phi_1\phi_2+b\phi_2\phi_3+c\phi_3\phi_1 \tag{8}$$

Since each basis function is a linear function of $x$ and $y$ it follows that any product of two basis functions must be a quadratic function of $x$ and $y$. The three pairs of basis functions that appear in equation (8) represent the three distinct pairs which are identically zero at each vertex of $T$. To see this, note that $\phi_1=0$ on the extended edge $R_3R_2$ while $\phi_2=0$ on the extended edge $R_1R_3$ so that $\phi_1\phi_2$ is zero at each vertex of $T$. In a similar way, it can be seen that $\phi_2\phi_3$ and $\phi_3\phi_1$ are zero at each vertex of $T$. It follows that $f(x_j,y_j)=0,\ j=1,2,3$.

Now let $S_j\ \varepsilon\ V-\{R_1,R_2,R_3\},\ j=1,\ldots,m$ be the next $m$ donor mesh points that are closest to $X$. Let $\phi_i(j)$ represent the value of $\phi_i$ at the data point $S_j$. Similarly, let $q(j)$, respectively $q_{lin}(j)$, be the values of the data, respectively linear interpolant, at the data points $S_j,\ j=1,\ldots,m$. Define the matrix

$$B=\begin{pmatrix} \phi_1(1)\phi_2(1) & \phi_2(1)\phi_3(1) & \phi_3(1)\phi_1(1) \\ \phi_1(2)\phi_2(2) & \phi_2(2)\phi_3(2) & \phi_3(2)\phi_1(2) \\ \vdots & \vdots & \vdots \\ \phi_1(m)\phi_2(m) & \phi_2(m)\phi_3(m) & \phi_3(m)\phi_1(m) \end{pmatrix} \tag{9}$$

The coefficients $a,b,c$ are determined by computing the least squares approximation of the error terms for the $m$ extra points. Thus, the coefficients are found by inverting the $3\times 3$ system [10]

$$B^TB\mathbf{a}=B^T\mathbf{w} \tag{10}$$

where $\mathbf{a}=(a,b,c)^T$ and

$$\mathbf{w}=\begin{pmatrix} q(1)-q_{lin}(1) \\ q(2)-q_{lin}(2) \\ \vdots \\ q(m)-q_{lin}(m) \end{pmatrix} \tag{11}$$

The interpolation procedure generalizes in a straightforward manner to 3D. In this case,

$$q(x,y,z)=q_{lin}(x,y,z)+f(x,y,z) \tag{12}$$

and

$$\begin{aligned} q_{lin}(x,y,z) &= q_1\phi_1(x,y,z)+q_2\phi_2(x,y,z) \\ &+q_3\phi_3(x,y,z)+q_4\phi_4(x,y,z) \end{aligned} \tag{13}$$

where $\phi_i(x,y,z),\ i=1,2,3,4$ are the linear basis functions associated with the tetrahedron $T$ that contains the point $X$. In an analogous manner, explicit expressions for these four linear basis functions can be found by inverting the system of equations

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{pmatrix}\begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{pmatrix}=\begin{pmatrix} 1 \\ x \\ y \\ z \end{pmatrix} \tag{14}$$

The quadratic error is now represented as

$$\begin{aligned} f(x,y,z) &= a\phi_1\phi_2+b\phi_1\phi_3+c\phi_1\phi_4 \\ &+d\phi_2\phi_3+e\phi_2\phi_4+f\phi_3\phi_4 \end{aligned} \tag{15}$$

By a similar argument to that given earlier, it can be seen that the six pairs of basis functions appearing in equation (15) are all zero at each of the four vertices of the tetrahedron $T$. The least squares approximation leads to the system,

$$C^TC\mathbf{a}=C^T\mathbf{w} \tag{16}$$

where $C$ is an $m\times 6$ matrix, each of whose rows is formed from the six distinct products of two basis functions evaluated at one of the $m$ points $S_j$. The vector $\mathbf{a}$ is the column vector containing the 6 coefficients $a,b,c,d,e,f$ and $\mathbf{w}$ is defined as before. Inversion of the $6\times 6$ matrix $C^TC$ is required to obtain the coefficients.

### 2.3 Higher Order Interpolants

The procedure generalizes in a natural way to permit cubic and higher order estimation of the error terms. In the planar case, the error up to and including third order terms is given by

$$f(x,y)=\begin{array}{c} a\phi_1\phi_2^2+b\phi_1\phi_3^2+c\phi_2\phi_1^2+d\phi_2\phi_3^2 \\ +e\phi_3\phi_1^2+f\phi_3\phi_2^2+g\phi_1\phi_2\phi_3 \end{array} \tag{17}$$

where the seven triple products of basis functions are the set of distinct products that are cubic in $x$ and $y$ and are identically zero at the vertices of the triangle $T$. Determination of the coefficients by least squares leads to a $7\times 7$ system of equations to invert. In a similar way one can estimate the error terms up to fourth order accuracy with an expression for $f(x,y)$ that is formed as a linear combination of the 12 distinct quadruple products of the basis functions that are zero at each vertex of $T$. It follows in this case that there are 12 coefficients to be computed by solving a $12\times 12$ system of equations. Third order and fourth order accuracy in 3D requires the inversion of a $16\times 16$ and a $31\times 31$ system respectively.

## 3. SIZE OF POINT CLOUD

The number $m$ of mesh points $S_j, j=1,\cdots,m$ that are used to determine the least squares estimate should not be too large in order to maintain a compact support for the evaluation of the error term. Too few points, on the other hand, will result in a covariance matrix $B^TB$ (or $C^TC$ in 3D) that is singular. Although the least squares system of equations is consistent and therefore always has a solution, a non-singular covariance matrix ensures that the least squares solution is unique.

In principle, one could handle the singular case by choosing the minimum length solution, or which is equivalent, by taking the pseudo inverse $\mathbf{a} = B^+\mathbf{w}$ [11]. Evidently, $m$ cannot be larger than $n = card\ V$ so that the pseudo inverse should be used if the point set $V$ is extremely small and one is, in effect, trying to interpolate through an insufficiently large set of data points. In general, however, the size of $V$ will not be a limitation and the number $m$ of data points used in the least squares fit should be chosen to be sufficiently large to ensure invertibility of the covariance matrix.

In order to determine conditions under which the covariance matrix $B^TB$ will be singular it should first be noted that $B^TB$ has the same null-space as $B$ [10]. This follows from the observation the nullspace of $B$ is contained in the nullspace of $B^TB$ and *vice versa*. First $B\mathbf{x} = \mathbf{0} \Rightarrow B^TB\mathbf{x} = \mathbf{0} \Rightarrow N(B) \subset N(B^TB)$. Conversely, $B^TB\mathbf{x} = \mathbf{0} \Rightarrow \mathbf{x}^TB^TB\mathbf{x} = 0$ so that $\|B\mathbf{x}\|^2 = 0 \Rightarrow B\mathbf{x} = \mathbf{0}$. Hence $N(B^TB) \subset N(B)$ and these two inclusions show that $N(B^TB) = N(B)$.

It follows that $rank\ B^TB = rank\ B$ and the number of linearly independent columns of $B^TB$ is therefore the same as the number of linearly independent columns of $B$. For the planar case with quadratic error estimation, the matrix $B$ is given by equation (9) and hence invertibility of $B^TB$ requires $m \geq rankB = 3$.

## 3.1 Condition for a Diagonal Covariance Matrix

An example for which $m = 3$ is sufficient is shown in figure 2. The point $S_1$ lies on the extended edge $R_2R_1$ so that $\phi_3(1) = 0$. Similarly, $\phi_1(2) = 0$ since $S_2$ lies on the extended edge $R_3R_2$ and $\phi_2(3) = 0$ since $S_3$ lies on the extended edge $R_1R_3$. Hence

$$B = \begin{pmatrix} \phi_1(1)\phi_2(1) & 0 & 0 \\ 0 & \phi_2(2)\phi_3(2) & 0 \\ 0 & 0 & \phi_3(3)\phi_1(3) \end{pmatrix} \tag{18}$$

and

$$B^TB = \begin{pmatrix} \phi_1^2(1)\phi_2^2(1) & 0 & 0 \\ 0 & \phi_2^2(2)\phi_3^2(2) & 0 \\ 0 & 0 & \phi_3^2(3)\phi_1^2(3) \end{pmatrix} \tag{19}$$

In this particular case the matrix $B^TB$ will be diagonal, a property that does not hold unless the three extra data points $S_1, S_2, S_3$ all lie on extended edges of the containing triangle $T$. More generally, if the point $S_j,\ j = 1, \cdots, m$ lies on the extended edge of the containing triangle $T$ opposite vertex $R_i$ where $i$ = 1, 2 or 3 then $\phi_i(j) = 0$. Hence each row of $B$ has two zero entries and just one non-zero entry. It follows that the columns of $B$ are orthogonal so that $B^TB$ will be diagonal.

**Remark 1**: If every point $S_j,\ j = 1, \cdots, m$ lies on an extended edge of the containing triangle $T$ then $B^TB$ will be diagonal. If, in addition, there is at least one point $S_j$ on each of the three extended edges then $B^TB$ will be invertible.

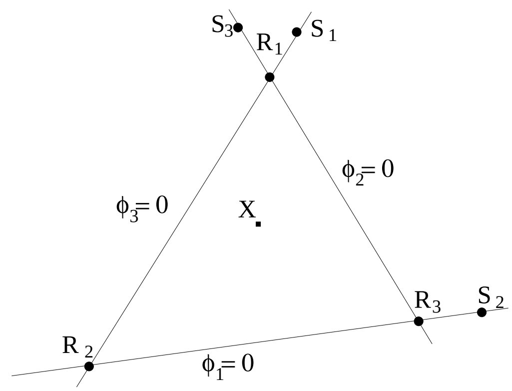


**Figure 2**: A case when three points suffice for invertibility.

## 3.2 A Condition when Three Extra Points is Insufficient

The more interesting situation, or at least the situation of greater concern, is associated with $B^TB$ being singular. Figure 3 illustrates a situation when three extra points $S_1, S_2, S_3$ are not sufficient to make $B$ and hence $B^TB$ nonsingular. If two data points, say $S_1$ and $S_2$, lie on the same extended side, say $R_2R_1$, then $\phi_3(1) = 0$ and $\phi_3(2) = 0$. It follows that

$$B = \begin{pmatrix} \phi_1(1)\phi_2(1) & 0 & 0 \\ \phi_1(2)\phi_2(2) & 0 & 0 \\ \phi_1(3)\phi_2(3) & \phi_2(3)\phi_3(3) & \phi_3(3)\phi_1(3) \end{pmatrix} \tag{20}$$

Hence $B$ and therefore $B^TB$ have rank 2. The deficiency in rank occurs as a result of the fact that the linear basis function $\phi_i$ is zero on the extended edge of $T$ that is opposite vertex $R_i$.

In general, if there are $m$ extra data points of which the first $m-1$ lie on an extended edge of $T$, say $R_2R_1$

so that $\phi_3(j) = 0,\ j = 1, \cdots, m-1$ then

$$B = \begin{pmatrix} \phi_1(1)\phi_2(1) & 0 & 0 \\ \phi_1(2)\phi_2(2) & 0 & 0 \\ \vdots & \vdots & \vdots \\ \phi_1(m-1)\phi_2(m-1) & 0 & 0 \\ \phi_1(m)\phi_2(m) & \phi_2(m)\phi_3(m) & \phi_3(m)\phi_1(m) \end{pmatrix} \tag{21}$$

and

$$B^T B = \begin{pmatrix} \mathbf{c_1} & \mathbf{c_2} & \mathbf{c_3} \end{pmatrix} \tag{22}$$

where the columns are given by

$$\mathbf{c_1} = \begin{pmatrix} \sum \phi_1^2 \phi_2^2 \\ \phi_1(m)\phi_2(m)^2\phi_3(m) \\ \phi_1(m)^2\phi_2(m)\phi_3(m) \end{pmatrix} \tag{23}$$

$$\mathbf{c_2} = \begin{pmatrix} \phi_1(m)\phi_2(m)^2\phi_3(m) \\ \phi_2(m)^2\phi_3(m)^2 \\ \phi_1(m)\phi_2(m)\phi_3(m)^2 \end{pmatrix} \tag{24}$$

$$\mathbf{c_3} = \begin{pmatrix} \phi_1(m)^2\phi_2(m)\phi_3(m) \\ \phi_1(m)\phi_2(m)\phi_3(m)^2 \\ \phi_1(m)^2\phi_3(m)^2 \end{pmatrix} \tag{25}$$

As expected the covariance matrix $B^T B$ has rank 2 since columns 2 and 3 are linearly dependent.

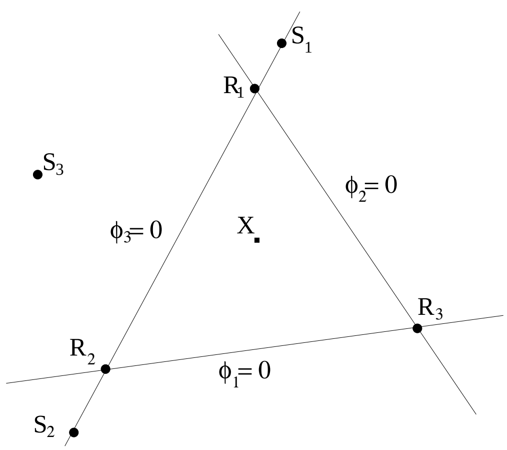


**Figure 3**: A example when three points do not suffice for invertibility.

We summarize this result as

**Remark 2**: The covariance matrix $B^T B$ will be singular if more than $m-2$ of the $m$ extra data points $S_j, j = 1, \cdots, m$ lie on one extended edge $e$ of the containing triangle $T$.

## 3.3 A Condition for Matrix $B$ to have linearly dependent columns

It is also possible for $B$ and hence $B^T B$ to be singular if any two columns of $B$ are linearly dependent. This can only arise if all $m$ points lie on a line through a vertex of the containing triangle $T$. This possibility is illustrated in figure 4. Suppose, for example, that the extra data points all lie on a straight line $L$ through a vertex of the containing triangle $T$. In particular, as shown in figure 4, let $R_3$ be the vertex through which $L$ passes. Then, if $S$ is any data point on $L$, it follows that

$$\phi_1(S) = \frac{A_1}{A}, \qquad \phi_2(S) = \frac{A_2}{A} \tag{26}$$

with

$$A_1 = \frac{1}{2} h_1 l_1, \qquad A_2 = \frac{1}{2} h_2 l_2 \tag{27}$$

Here, $A_1$ is the area of triangle $R_2 S R_3$, $A_2$ is the area of triangle $R_3 S R_1$, $l_1$ is the length of edge $R_2 R_3$, $l_2$ is the length of edge $R_3 R_1$ and $h_1$, respectively $h_2$, is the length of the perpendicular from $S$ to the extended edge $R_2 R_3$, respectively $R_3 R_1$. Now let $\theta$ be the angle between line $L$ and the extended edge $R_3 R_1$ and let $\psi$ be the angle between $L$ and the extended edge $R_2 R_3$. It follows that

$$h_1 = d\, sin\psi \qquad \text{and} \qquad h_2 = d\, sin\theta \tag{28}$$

Hence

$$\phi_2(S) = \alpha \phi_1(S) \ \text{ where } \ \alpha = \frac{l_2 sin\theta}{l_1 sin\psi} \tag{29}$$

Since $\alpha$ does not depend on the distance $d$ of $S$ from the vertex $R_3$, this relation must be true for any position $S$ on $L$. It follows that the second and third columns of $B$ are linearly dependent so that $B^T B$ will be singular.

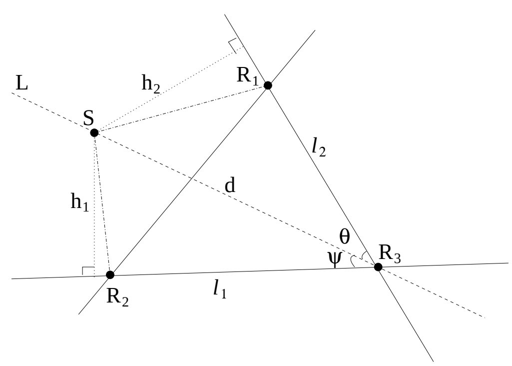


**Figure 4**: Colinearity condition for a rank deficient matrix.

**Remark 3**: The matrix $B^T B$ will be singular if the $m$ points $S_j,\ j = 1, \cdots, m$ are colinear and lie on a line passing through a vertex of the containing triangle $T$.

For a general arrangement of points it is unlikely that these pathological situations will arise. It is possible, however, that the covariance matrix $B^T B$ will be badly conditioned if the number $m$ of extra data points is small and the arrangement of donor mesh points has a lattice organization as illustrated in figure 9. In practice, a well conditioned covariance matrix is usually assured by taking $m$ equal to twice the number of columns in the matrix $B$. Since the system of normal equations (eqns. (10) or (16)) is non-negative definite, inversion can be accomplished by a Cholesky decomposition and the determinant and/or condition number of $B^T B$ monitored to detect singular behavior. If this does occur, further data points can be acquired until the system of equations does become well conditioned.

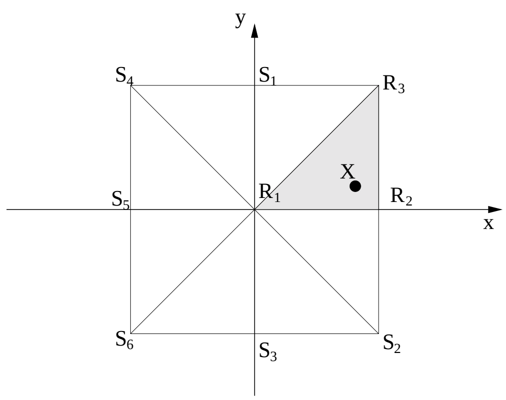


**Figure 5**: Mesh for the nine point test example.

## 4. QUADRATIC INTERPOLANTS FOR A NINE POINT MESH

The mesh shown in figure 5 contains nine points, eight placed around the perimeter of a square of side length 2. The remaining point $R_1$ lies at the center of the square which is taken to be the origin. Let $q$ take the value 1 at the origin and the value zero at each of the eight perimeter points. For the interpolation position $X$ the containing triangle has vertices $R_1(0,0), R_2(1,0)$ and $R_3(1,1)$ with the data values $q_1 = 1, q_2 = q_3 = 0$. The basis functions are

$$\phi_1(x,y) = 1-x, \;\; \phi_2(x,y) = x-y, \;\; \phi_3(x,y) = y \quad (30)$$

whence

$$q_{lin}(x,y) = 1-x \quad (31)$$

The linear interpolant over the other seven triangles is easily obtained and is displayed for the entire mesh in figure 6.

If the nearest three extra points $S_1, S_2$ and $S_3$ are used to obtain the quadratic correction we find that

$$B\mathbf{a} = \mathbf{w} \quad (32)$$

where

$$B = \begin{pmatrix} -1 & -1 & 1 \\ 0 & -2 & 0 \\ 1 & -1 & -1 \end{pmatrix}, \;\; \mathbf{w} = \begin{pmatrix} -1 \\ 0 \\ -1 \end{pmatrix} \quad (33)$$

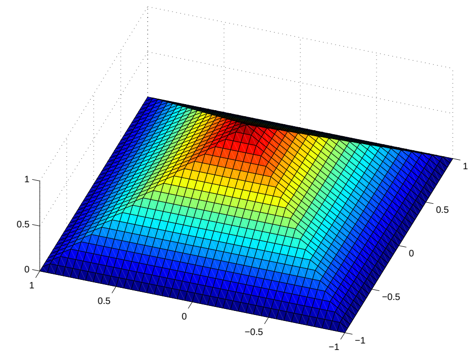


**Figure 6**: Linear interpolant for nine point test example.

In this case $rank\ B = 2$ so that the system of normal equations (10) is singular. Using Householder transformations [11] we obtain

$$R\mathbf{y} = \mathbf{g} \quad (34)$$

where $R = QBK$ and $R$ has the form

$$R = \begin{pmatrix} R_{11} & 0 \\ 0 & 0 \end{pmatrix} \quad (35)$$

where $R_{11}$ is a $2 \times 2$ upper triangular matrix. $Q$ and $K$ are orthogonal matrices and $\mathbf{y} = K^T\mathbf{a}, \;\; \mathbf{g} = Q\mathbf{w}$. For this particular example

$$R_{11} = \begin{pmatrix} -2 & 0 \\ 0 & \sqrt{6} \end{pmatrix}, \;\; \mathbf{g} = \begin{pmatrix} 0 \\ \frac{\sqrt{2}}{\sqrt{3}} \\ -\frac{2}{\sqrt{3}} \end{pmatrix} \quad (36)$$

and

$$K = \begin{pmatrix} -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{pmatrix} \quad (37)$$

Let $\mathbf{y}^T = (y_1, y_2, y_3)$. Solving the system

$$\begin{pmatrix} -2 & 0 \\ 0 & \sqrt{6} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{\sqrt{2}}{\sqrt{3}} \end{pmatrix} \quad (38)$$

gives $y_1 = 0, \;\; y_2 = \frac{1}{3}$. To obtain the pseudo inverse we set $y_3 = 0$ and then compute $\mathbf{a} = K\mathbf{y}$ to give

$$a = c = 0, \;\; b = \frac{1}{3} \quad (39)$$

Substituting the values of these coefficients into equation (8) and using the linear basis functions given in equations (30) we find that the quadratic correction based on the three nearest extra points is

$$f_3(x,y) = \frac{1}{3}y(x-y) \quad (40)$$

and the corresponding second order interpolant is

$$q_3(x,y) = 1 - x + \frac{1}{3}y(x-y) \qquad (41)$$

The corresponding interpolants for the remaining seven triangles can be easily found by symmetry considerations. Figure 7 shows this interpolant for the entire mesh.

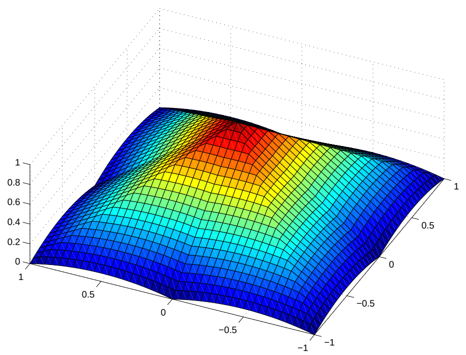

**Figure 7**: Quadratic interpolant using three extra points and the pseudo-inverse.

When four extra points $S_1, S_2, S_3$ and $S_4$ are used we find that

$$B = \begin{pmatrix} -1 & -1 & 1 \\ 0 & -2 & 0 \\ 1 & -1 & -1 \\ -4 & -2 & 2 \end{pmatrix} \qquad (42)$$

which has rank 3 so that $B^T B$ is invertible. In this case we have

$$a = c = \frac{2}{3},\ b = \frac{1}{3} \qquad (43)$$

leading to the quadratic interpolant over triangle $R_1R_2R_3$ given by

$$q_4(x,y) = 1 - x + \frac{2}{3}x(1-x) + \frac{1}{3}y(x-y) \qquad (44)$$

The corresponding interpolant $q_5(x,y)$ based on five extra points has the coefficients

$$a = \frac{13}{14},\ b = \frac{2}{7},\ c = \frac{15}{14} \qquad (45)$$

while the interpolant $q_6(x,y)$ based on all six extra points has the coefficients

$$a = \frac{48}{53},\ b = \frac{15}{53},\ c = \frac{54}{53} \qquad (46)$$

The interpolant $q_4(x,y)$ based on four extra points is displayed for the entire mesh in figure 8.

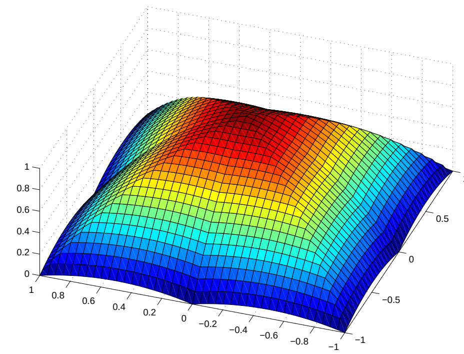

**Figure 8**: Quadratic interpolant using four extra points for the least sqares fit.

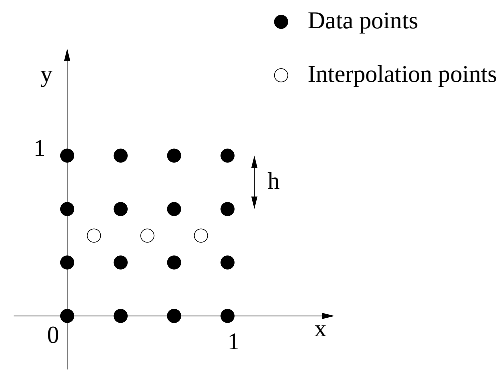


**Figure 9**: Point set $V$ (filled in circles) and interpolation positions (open circles).

## 5. A SMOOTHLY VARYING TEST CASE

A simple test case, using a smoothly varying function to define the data values, illustrates the efficacy of the procedure for different orders of interpolation. A donor mesh (see figure 9) was defined by a regular lattice of points covering a square whose sides have unit length. The minimum spacing, or lattice width, $h$ between mesh points was varied in order to investigate how well the accuracy of the interpolation schemes improved as $h$ was reduced in size. The data values assigned to the mesh points were given by

$$q(x,y) = (sin\frac{\pi}{2}x\, sin\frac{\pi}{2}y)^2 \qquad (47)$$

a function that varies smoothly between 0 and 1.

Interpolated values were obtained at a series of points along a line that ran across the mesh and such that the interplation positions were as far as possible from the donor mesh points. The interpolated values were compared with the exact values to determine the interpolation error at each sample point and the root mean

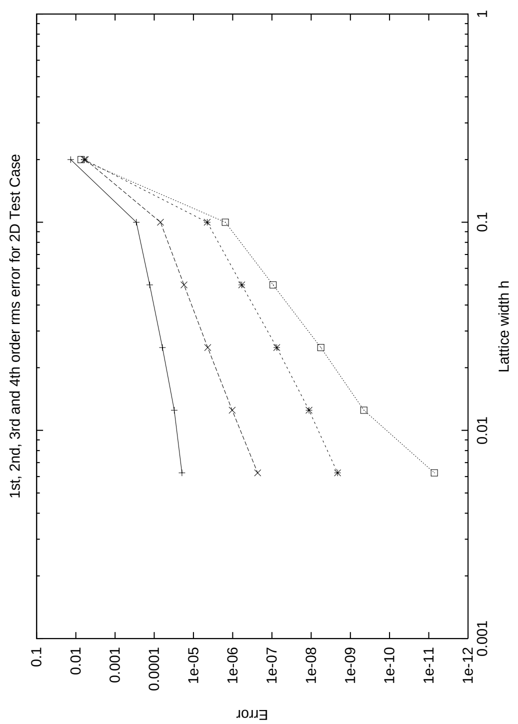


**Figure 10**: Error versus mesh width $h$ for the 2D test case.

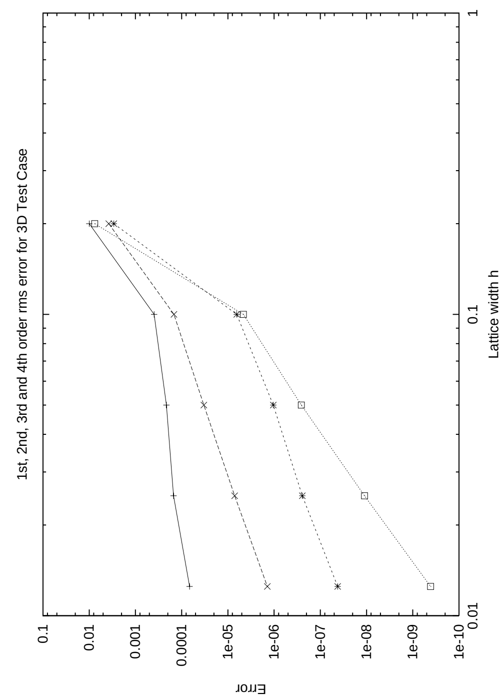


**Figure 11**: Error versus mesh width $h$ for the 3D test case.

square (rms) value of the error at all sample points was computed. Figure 10 shows the rms error versus mesh width $h$ for the linear interpolation as well as for second order, third order and fourth order interpolation. The rate at which the error diminishes as $h$ becomes smaller shows clearly how the higher order interpolation schemes provide superior performance albeit by requiring the inversion of larger matrix systems than the lower order schemes.

A similar comparison in 3D was made for a three dimensional lattice of points on the unit cube whose data values were given by

$$q(x,y) = (sin\frac{\pi}{2}x\, sin\frac{\pi}{2}y\, sin\frac{\pi}{2}z)^2 \tag{48}$$

The root mean square error versus mesh width $h$ for the linear interpolation as well as for second order, third order and fourth order interpolation is shown in figure 11. The trend of error reduction versus lattice spacing $h$ is similar to that demonstrated for the 2D case in figure 10.

## 6. INTERPOLATION THROUGH A STEP FUNCTION

A more severe test is provided by data values which represent a discontinuous jump. In the 2D case the following step function was therefore used to define the data values $q$ at each lattice point on the unit square.

$$q(x,y) = \begin{cases} 1 & \text{if } x < \frac{1}{2} \\ 0 & \text{if } x \geq \frac{1}{2} \end{cases} \tag{49}$$

A cut through the interpolating surface along the line $y = 0.5$ is shown in figure 12 over part of the $x$ axis for data defined on the donor mesh at a lattice spacing of $h = 0.025$. The linear interpolant decreases from a value of 1 at $x = 0.475$ to zero at $x = 0.5$. The second order interpolant displays an overshoot ahead of the step jump and an undershoot after the step jump whose magnitude is around 10% of the step height. The overshoots extend roughly one half of the lattice spacing $h$ on either side of the discontinuity. (Note that the symbols shown on the curves in figures 12 and 13 do not represent actual mesh positions which are much more widely spaced and at which the function values $q(x.y)$ are, of course, interpolated exactly). The cubic interpolant is smoother with an overshoot and undershoot that is about half the amplitude of that for the quadratic interpolant but which extends about twice as far (*i.e.* a whole lattice spacing $h$ before and after the discontinuity). Figure 13 shows the corresponding result for a lattice width $h = 0.00625$. The results, as one would expect, are similar to the previous comparison showing overshoots and undershoots

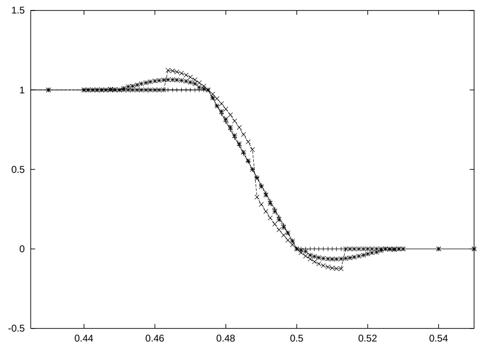


**Figure 12**: Interpolated function values of $q(x, y)$ at $y = 0.5$ for a lattice width of $h = 0.025$

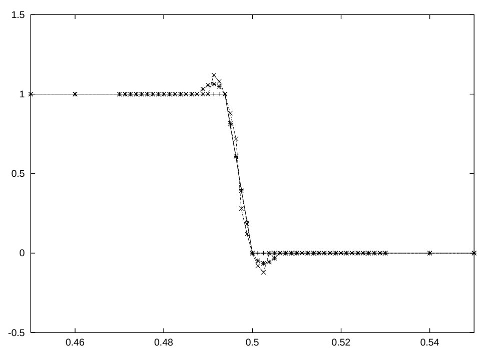


**Figure 13**: Interpolated function values of $q(x, y)$ at $y = 0.5$ for a lattice width of $h = 0.00625$

of comparable magnitude and extending the same distance when scaled by the lattice spacing.

## 7. ACKNOWLEDGMENT


Part of this research was carried out in the course of consulting activity for Combustion Research and Flow Technology, Inc. and was funded by the Air Force Research Laboratory, Wright-Patterson AFB, OH under contract F33615-02-C-3215, Phase II SBIR (technical monitors Frank Witzeman and Matthew Grismer). The support is gratefully acknowledged.


## References


[1] N.A. Malik and Th. Dracos Interpolation Schemes for Three-Dimensional Velocity Fields from Scattered Data Using Taylor Expansions, *J. Comp. Phys.*, 231-243, 1995.

[2] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming and P. Krysl Meshless methods: An overview and recent developments, *Comput. Methods Appl. Mech. Engrg.*, **139**, 3-47, 1996.

[3] C.A. Duarte and J.T.. Oden H-p Clouds - an h-p Meshless Method, *Numerical Methods for Partial Differential Equations*, 1-34, 1996.

[4] T.J. Liszka, C.A.M.. Duarte and W.W. Tworzydlo *hp*-Meshless cloud method, *Comput. Methods Appl. Mech. Engrg.*, **139**, 263-288, 1996.

[5] B. Nayroles, G. Touzot and P. Villon Generalizing the Finite Element Method: Diffuse Approximation and Diffuse Elements, *Computational Mechanics*, 307-318, 1992.

[6] A. Rassineux, P. Villon, J-M. Savignat, and O. Stab Surface Remeshing by Local Hermite Diffuse Interpolation, *Int. J. Num. Meth. Eng.*, 31-49, 2000.

[7] P. Lancaster and K. Salkauskas Surfaces Generated by Moving Least Squares Methods, *Math. Comp.*, **37**, 141-158, 1981.

[8] T. Liszka An Interpolation Method for an Irregular Net of Nodes, *Int. J. Num. Meth. Eng.*, **20**, 1599-1612, 1984.

[9] T.J. Baker Automatic Mesh Generation for Complex Three-Dimensional Regions using a Constrained Delaunay Triangulation. *Engineering with Computers.*, **5**, 161-175, 1989

[10] G. Strang Linear Algebra and its Applications. pub. Harcourt Brace Jovanovich, 1988.

[11] C.L. Lawson and R.J. Hanson *Solving Least Squares Problems*, pub. Prentice-Hall, 1974.