Computer Aided Geometric Design 3 (1986) 53-66
North-Holland

A representation of a C^2 interpolant over triangles

Tracy WHELAN *

Department of Mathematics, University of Utah, Salt Lake City, UT 84112, USA

Received 30 April 1985
Revised 24 December 1985

Abstract. A C^2 piecewise nonic interpolant defined over triangles is derived using the Bernstein-Bézier method. The interpolant can be constructed to require only vertex data (including derivatives) and to have seventh degree polynomial precision.

Keywords. Scattered data, piecewise polynomial, bivariate interpolant, C^2, Bernstein-Bézier form, computer aided geometric design.

Introduction

The use of bivariate interpolants in the analysis, representation and design of 3-D surfaces is becoming increasingly common and necessary. In many applications, a representation of the surface to be produced or analysed is required. This paper will give an explicit representation of a bivariate interpolant which has been used for finite element analysis and which could be useful in more varied or general surface applications. The interpolant is a piecewise polynomial of degree 9 over triangles which has overall continuity class C^2. The representation will be given in Bernstein-Bézier form.

Surface schemes arise naturally in the use of the Finite Element Method. In the analysis of the stress and strain distributions in elastic continua, the distributions are modeled by initially separating the continuum into a number of 'finite elements', which are considered to be connected at nodes on their boundaries. The displacements of the nodes are parameters to be determined by the analysis, while a function within each element defines the state of displacement of the nodes (in turn defining the state of strain within the element). The purpose of the analysis is to determine the boundary stresses and any distributed loads [Zienkiewicz '77].

The elements used in a finite element analysis can be chosen to conform to different conditions. For example, there are rectangular and triangular elements, elements using only positional information and those also using derivative information, and so on. These schemes can be valuable in their own right for such applications as surface fitting (interpolation) and surface design. In these cases, it is necessary to have an explicit representation of the surface based on the data to which the surface will interpolate.

Barnhill and Farin [Barnhill, Farin '81] give representations of a piecewise quintic element defined over triangles which originated from the eighteen degrees-of-freedom element Q_{18} [Strang, Fix, '73]. To obtain the twenty-one data required to define a quintic, position, first and second derivatives at the vertices of each triangle and the normal derivative at the midpoint of each edge were used.

* Sponsored by the Department of Energy under Contract No. DE-AC02-82ER12046.

0167-8396/86/$3.50

In most applications, however, data which are not at the vertices of the triangles (elements) are not available. Therefore there is a well-known Finite Element Method technique called 'condensation of parameters' which can be applied to avoid the necessity of using such data. Instead of interpolating to data away from the vertices, one can constrain the solution to satisfy certain requirements, in the process obtaining the full number of equations to match the degrees of freedom for parameter evaluation. For example, Barnhill and Farin produce a condensed scheme by requiring that the normal derivatives along each edge be cubic polynomials rather than the quartics they would be without condensation.

It should be emphasized that condensed schemes are far more useful for purposes of interpolation than uncondensed schemes, since the condensed schemes do not rely on data away from vertices.

When using a surface scheme for interpolation, there are properties of the scheme which may be important or desirable. One such property is that the interpolating function be defined over *arbitrarily spaced* (i.e., not necessarily gridded) data. Another is that the interpolant be *local*; that is, the value of the surface at a point should depend only on data near that point. Taken together, these two properties suggest that the interpolant be defined piecewise over triangles, as any (nondegenerate) set of domain data can be triangulated, using a technique such as Little's method described in [Barnhill '77]. The piecewise definition of the interpolant can be formulated to give a local scheme. In our case, piecewise polynomial functions are chosen as they are especially easy to work with, and so may be more generally applicable than more complicated functions.

A quality that is generally desired is that the surface be *smooth*. The piecewise quintic interpolant described above is C^1 overall. However, curvature continuity is frequently desirable or essential, as in the representation of feature lines of automobiles or solutions of higher order partial differential equations. This leads to the need for a representation of an interpolant which is at least C^2.

Ženišek [Ženišek '73] was the first to give general theorems about triangular C^m finite elements. He shows that if a piecewise polynomial function defined over an arbitrarily triangulated domain is to be C^2, values through all fourth derivatives at the vertices of the triangles must be utilized. It should be noted that although such data are usually unavailable, techniques exist for the generation of derivative data based only on positional information. (See [Alfeld '84] for an example.)

Ženišek [Ženišek '73] states a theorem that the simplest piecewise polynomial function defined over triangles which will be C^2 is of degree 9. It is determined by position and derivatives up to and including all fourth derivatives at triangle vertices, first normal cross-boundary derivatives at midpoints of edges, second normal cross-boundary derivatives at points dividing the sides of triangles into 3 equal parts and position at the centroid of the triangle. We will first give a representation of this uncondensed scheme.

As we stated, uncondensed schemes are not as useful as condensed schemes which do not require data away from the vertices. Therefore we also derive a representation of a condensed piecewise nonic interpolant which requires only information through fourth derivatives at the vertices. Equations for determining the remaining parameters are derived by constraining first and second normal cross-boundary derivatives along edges to be of a certain polynomial degree, and then maintaining as high a degree of polynomial precision as possible.

In this paper, we begin by reviewing the notation used for the Bernstein–Bézier representation of a (bivariate) polynomial. We next derive a representation of the uncondensed scheme. This is followed by a derivation of the condensed interpolant, the piecewise nonic, overall C^2 interpolant dependent only on vertex data. Lastly, this interpolating function is applied to data drawn from a well-known bivariate function, and the resulting surface is displayed.

Representation of the interpolant

The interpolating function defined over each triangle will be described in terms of the barycentric coordinates of the triangle. Given a nondegenerate triangle with vertices $V_i = (x_i,\ y_i)$, $i = 1,2,3$, and a point $P = (x,\ y)$, one can solve the equations

$$x = x_1 b_1 + x_2 b_2 + x_3 b_3,$$
$$y = y_1 b_1 + y_2 b_2 + y_3 b_3,$$
$$1 = b_1 + b_2 + b_3$$

for the barycentric coordinates $\mathbf{b} = (b_1,\ b_2,\ b_3)$ of P.

The nonic interpolant $N(\mathbf{b})$ is of the form

$$N(\mathbf{b}) = \sum_{\substack{r,s,t \geq 0\\ r+s+t=9}} c_{r,s,t} B^9_{rst}(\mathbf{b})$$

where

$$B^9_{rst}(\mathbf{b}) = \frac{9!}{r!s!t!} b_1^r b_2^s b_3^t,$$
$$r + s + t = 9,$$
$$b_1 + b_2 + b_3 = 1;\ \ b_i \geq 0$$

is a Bernstein polynomial of degree 9. The $c_{\mathrm{r,s,t}}$ are parameters to be determined by the interpolation conditions. With this notation, the subscripts of each parameter correspond to the exponents of the barycentric coordinates in the term which the parameter multiplies.

The Bernstein polynomials $B^n_{rst}(\mathbf{b})$,

$$B^n_{rst}(\mathbf{b}) = \frac{n!}{r!s!t!} b_1^r b_2^s b_3^t,$$
$$r + s + t = n,$$
$$b_1 + b_2 + b_3 = 1;\ \ b_i \geq 0$$

form a basis for all bivariate polynomials of degree n. $N(\mathbf{b})$ is called a Bézier, or Bernstein–Bézier polynomial of degree 9. It is always assumed that 0^0 as well as 0! equals 1. For example, evaluation of $N(\mathbf{b})$ for $\mathbf{b} = (0, 0, 1)$ gives just one non-zero term, $c_{0,0,9}$.

The equation $\mathbf{b}(t) = \mathbf{u}t + \mathbf{d}$ defines a straight line with direction $\mathbf{b}'(t) = \mathbf{u}$. Taking the derivative of $N(\mathbf{b})$ with respect to the direction $\mathbf{u} = (u_1,\ u_2,\ u_3)$ gives

$$\frac{\partial N(\mathbf{b})}{\partial \mathbf{u}} = \frac{\partial N}{\partial b_1} u_1 + \frac{\partial N}{\partial b_2} u_2 + \frac{\partial N}{\partial b_3} u_3$$

where

$$\frac{\partial N(\mathbf{b})}{\partial b_1} = 9 \sum_{r+s+t=8} c_{r+1,s,t} B^8_{rst}(\mathbf{b}),$$

$$\frac{\partial N(\mathbf{b})}{\partial b_2} = 9 \sum_{r+s+t=8} c_{r,s+1,t} B^8_{rst}(\mathbf{b}),$$

$$\frac{\partial N(\mathbf{b})}{\partial b_3} = 9 \sum_{r+s+t=8} c_{r,s,t+1} B^8_{rst}(\mathbf{b}),$$

and thus

$$\frac{\partial N(\mathbf{b})}{\partial \mathbf{u}} = 9 \sum_{r+s+t=8} \left(u_1 c_{r+1,s,t} + u_2 c_{r,s+1,t} + u_3 c_{r,s,t+1}\right) B^8_{rst}(\mathbf{b}).$$

In general,

$$\frac{\partial^l N(\mathbf{b})}{\partial \mathbf{u}^l} = \frac{9!}{(9-l)!} \sum_{r+s+t=9-l} \sum_{\rho_1+\rho_2+\rho_3=l} c_{r+\rho_1,s+\rho_2,t+\rho_3} B^l_{\rho_1\rho_2\rho_3}(\mathbf{u}) B^{9-l}_{rst}(\mathbf{b}).$$

Mixed derivatives are computed similarly; for example, if $\mathbf{u}$ is in the direction of edge 3 ($\mathbf{u} = (-1, 1, 0)$) and $\mathbf{v}$ is in the direction of minus edge 2 ($\mathbf{v} = (-1, 0, 1)$), then

$$\frac{\partial^2 N(\mathbf{b})}{\partial \mathbf{u} \partial \mathbf{v}} = 72\Bigg(\sum_{r+s+t=7} (c_{r+2,s,t} - c_{r+1,s+1,t} - c_{r+1,s,t+1} + c_{r,s+1,t+1}) B^7_{rst}(\mathbf{b})\Bigg).$$

A general bivariate nonic polynomial requires 55 parameters for its definition, and in the context of interpolation this amounts to solving a 55 by 55 system of linear equations. Using the Bernstein–Bézier representation simplifies the derivation of the interpolant to a large degree. By properly ordering the defining equations, almost all of the coefficients $c_{r,s,t}$ can be expressed in terms of one of the data along with combinations of previously computed coefficients. In the worst cases, the remaining parameters occur in pairs of equations to be solved simultaneously.

Derivation of the interpolant. Uncondensed scheme

As we stated in the introduction, the data used to define the uncondensed scheme will be position and all derivatives through fourth-order derivatives at the vertices of each triangle, first normal cross-boundary derivatives at midpoints of the triangle edges, second normal cross-boundary derivatives at points dividing the edges into 3 equal parts, and position at the centroid of the triangle.

It will be assumed that all data are given with respect to the triangle over which $N(\mathbf{b})$ is to be defined. For example, if

$$\left.\frac{\partial F}{\partial x}\right|_{V_1} \text{ and } \left.\frac{\partial F}{\partial y}\right|_{V_1}$$

are given, one can determine

$$\left.\frac{\partial F}{\partial (V_2 - V_1)}\right|_{V_1} \text{ and } \left.\frac{\partial F}{\partial (V_3 - V_1)}\right|_{V_1}$$

from their values. This assumption not only simplifies the expressions which result for the coefficients $c_{r,s,t}$, but also aids in making full use of the symmetry available.

Data at the vertices of the triangle determine the first 45 coefficients of $N(\mathbf{b})$. Some examples follow.

Example 1. *Positional data at vertices.* Evaluation of $N(\mathbf{b})$ at vertex 1 ($\mathbf{b} = (1, 0, 0)$) gives $c_{9,0,0} = F(1, 0, 0) = F(V_1)$. Similarly, $c_{0,9,0} = F(V_2)$ and $c_{0,0,9} = F(V_3)$.

Example 2. *First derivative data at vertices.* Let $\mathbf{u} = (-1, 1, 0)$ (the direction of $V_2 - V_1$). Then

$$\left.\frac{\partial N}{\partial \mathbf{u}}\right|_{V_1} = 9(-c_{9,0,0} + c_{8,1,0})$$

which we want to equal $\partial F/\partial(V_2 - V_1)\,|_{V_1}$. Thus,

$$c_{8,1,0} = \frac{1}{9} \left.\frac{\partial F}{\partial (V_2 - V_1)}\right|_{V_1} + c_{9,0,0}.$$

By symmetry, one can then obtain $c_{8,0,1}$, $c_{1,8,0}$, $c_{0,8,1}$, $c_{0,1,8}$ and $c_{1,0,8}$. For instance,

$$c_{0,8,1}=\frac{1}{9}\frac{\partial F}{\partial(V_3-V_2)}\bigg|_{V_2}+c_{0,9,0}.$$

Example 3. *Mixed partial second order derivative data at vertices.* Let $\mathbf{u}=(-1,1,0)$ and $\mathbf{v}=(-1,0,1)$. Then

$$\frac{\partial^2 N}{\partial\mathbf{u}\partial\mathbf{v}}\bigg|_{V_1}=72(c_{9,0,0}-c_{8,1,0}-c_{8,0,1}+c_{7,1,1}).$$

This should equal $\partial^2F/\partial(V_2-V_1)\partial(V_3-V_1)\,|_{V_1}$. Therefore,

$$c_{7,1,1}=\frac{1}{72}\frac{\partial^2F}{\partial(V_2-V_1)\partial(V_3-V_1)}\bigg|_{V_1}+c_{8,0,1}+c_{8,1,0}-c_{9,0,0}.$$

Table 1 supplies equations for nine of the $c_{r,s,t}$; symmetry applied to these gives all coefficients except $c_{1,4,4}$, $c_{4,1,4}$, $c_{4,4,1}$, $c_{4,2,3}$, $c_{4,3,2}$, $c_{3,4,2}$, $c_{2,4,3}$, $c_{3,2,4}$, $c_{2,3,4}$ and $c_{3,3,3}$.

The first nine of the remaining coefficients are determined by the first and second normal cross-boundary derivative information at specified points on the edges; an example is supplied below (Example 4). Finally, $c_{3,3,3}$ is determined from the position at the centroid. Sample equations for the values of the remaining coefficients are given in Table 2.

In future, commas within the subscripts on the coefficients $c_{r,s,t}$ will be dropped whenever this will not lead to any possible ambiguity in the definitions. This practice will also be followed in Tables 1 through 4.

Example 4. *First normal cross-boundary derivative data at the midpoint of edge 1.* Let e_i represent edge i of the triangle, $i=1,2,3$. Let

$$h_i=\frac{-e_{i-1}\cdot e_i}{|e_i|^2},\quad e_0\equiv e_3,\quad i=1,2,3.$$

The normal directions to the edges of the triangles are given by:

$$\mathbf{n}_1=(1,\ h_1-1,\ -h_1),$$
$$\mathbf{n}_2=(-h_2,\ 1,\ h_2-1),$$
$$\mathbf{n}_3=(h_3-1,\ -h_3,\ 1),$$

where $\mathbf{n}_i$ is the normal to edge e_i. Thus, on edge 1,

$$\begin{aligned}\frac{\partial N}{\partial\mathbf{n}_1}(0,\tfrac{1}{2},\tfrac{1}{2})=\tfrac{9}{256}(&c_{180}+(h_1-1)c_{090}-h_1c_{081}\\&+8(c_{171}+(h_1-1)c_{081}-h_1c_{072})+28(c_{162}+(h_1-1)c_{072}-h_1c_{063})\\&+56(c_{153}+(h_1-1)c_{063}-h_1c_{054})+70(c_{144}+(h_1-1)c_{054}-h_1c_{045})\\&+56(c_{135}+(h_1-1)c_{045}-h_1c_{036})+28(c_{126}+(h_1-1)c_{036}-h_1c_{027})\\&+8(c_{117}+(h_1-1)c_{027}-h_1c_{018})+c_{108}+(h_1-1)c_{018}-h_1c_{009})\end{aligned}$$

which should equal $(\partial F/\partial\mathbf{n}_1)(0,\frac{1}{2},\frac{1}{2})$.

This produces the first equation given in Table 2.

The uncondensed scheme interpolates to 55 data which correspond to linearly independent functionals with respect to nonic polynomials. Since the number of terms in a general bivariate nonic polynomial is 55, the uncondensed scheme has polynomial precision 9.

Table 1
Examples of coefficients based on vertex data.

Coefficient	Previously unused data required	Defining equation
c_{900}	$F(V_1)$	$c_{900}=F(V_1)$
c_{810}	$\frac{\partial F(V_1)}{\partial(V_2-V_1)}$	$c_{810}=\frac{1}{9}\frac{\partial F(V_1)}{\partial(V_2-V_1)}+c_{900}$
c_{720}	$\frac{\partial^2 F(V_1)}{\partial(V_2-V_1)^2}$	$c_{720}=\frac{1}{72}\frac{\partial^2 F(V_1)}{\partial(V_2-V_1)^2}+2c_{810}-c_{900}$
c_{711}	$\frac{\partial^2 F(V_1)}{\partial(V_2-V_1)\partial(V_3-V_1)}$	$c_{711}=\frac{1}{72}\frac{\partial^2 F(V_1)}{\partial(V_2-V_1)\partial(V_3-V_1)}+c_{801}+c_{810}-c_{900}$
c_{630}	$\frac{\partial^3 F(V_1)}{\partial(V_2-V_1)^3}$	$c_{630}=\frac{1}{504}\frac{\partial^3 F(V_1)}{\partial(V_2-V_1)^3}+3c_{720}-3c_{810}+c_{900}$
c_{621}	$\frac{\partial^3 F(V_1)}{\partial(V_2-V_1)^2\partial(V_3-V_1)}$	$c_{621}=\frac{1}{504}\frac{\partial^3 F(V_1)}{\partial(V_2-V_1)^2\partial(V_3-V_1)}+c_{720}+2c_{711}-c_{801}-2c_{810}+c_{900}$
c_{540}	$\frac{\partial^4 F(V_1)}{\partial(V_2-V_1)^4}$	$c_{540}=\frac{1}{3024}\frac{\partial^4 F(V_1)}{\partial(V_2-V_1)^4}+4c_{630}-6c_{720}+4c_{810}-c_{900}$
c_{531}	$\frac{\partial^4 F(V_1)}{\partial(V_2-V_1)^3\partial(V_3-V_1)}$	$c_{531}=\frac{1}{3024}\frac{\partial^4 F(V_1)}{\partial(V_2-V_1)^3\partial(V_3-V_1)}+3c_{621}+c_{630}$ $-3c_{711}-3c_{720}+c_{801}+3c_{810}-c_{900}$
c_{522}	$\frac{\partial^4 F(V_1)}{\partial(V_2-V_1)^2\partial(V_3-V_1)^2}$	$c_{522}=\frac{1}{3024}\frac{\partial^4 F(V_1)}{\partial(V_2-V_1)^2\partial(V_3-V_1)^2}+2c_{612}+2c_{621}$ $-c_{702}-4c_{711}-c_{720}+2c_{801}+2c_{810}-c_{900}$

Derivation of the interpolant. Condensed scheme

We now derive the Bézier representation of the condensed version of the interpolant. The condensed scheme utilizes the same vertex information as the uncondensed scheme. Consequently, the derivation of the first 45 coefficients of the condensed scheme is identical to that of the first 45 coefficients of the uncondensed interpolant.

There are three steps to determining the remaining coefficients. Steps 1 and 2 use the fact that N restricted to an edge is a univariate Bézier polynomial. For example, on edge e_1, points have coordinates of the form $(0, 1-b_3, b_3)$, so

$$\begin{aligned} N(\mathbf{b})\,|_{e_1} &= \sum_{k=0}^{9} c_{0,9-k,k}\frac{9!}{(9-k)!k!}(1-b_3)^{9-k}b_3^k \\ &= \sum_{k=0}^{9} c_{0,9-k,k}B_k^9(b_3) \end{aligned}$$

where

$$B_k^9(b_3)=\frac{9!}{k!(9-k)!}b_3^k(1-b_3)^{9-k}$$

Table 2
Examples of coefficients based on edge data, uncondensed scheme.

$$c_{144}=\frac{1}{70}\bigg(\frac{256}{9}\frac{\partial F}{\partial \mathbf{n}_1}(0,\tfrac{1}{2},\tfrac{1}{2})-(c_{180}+c_{108}+8(c_{171}+c_{117})+28(c_{162}+c_{126})+56(c_{153}+c_{135})+(h_1-1)c_{090}-h_1c_{009}$$
$$+(-h_1+8(h_1-1))c_{081}+((h_1-1)-8h_1)c_{018}+(-8h_1+28(h_1-1))c_{072}+(8(h_1-1)-28h_1)c_{027}$$
$$+(-28h_1+56(h_1-1))c_{063}+(28(h_1-1)-56h_1)c_{036}+(-56h_1+70(h_1-1))c_{054}+(56(h_1-1)-70h_1)c_{045})\bigg)$$

$$c_{234}=\frac{(2A_1-A_2)}{3},\quad c_{243}=A_1-2c_{234}$$

where

$$A_1=\left(\frac{\partial^2 F}{\partial \mathbf{n}_1^2}(0,\tfrac{1}{3},\tfrac{2}{3})-S\right)\Big/C,\quad A_2=\left(\frac{\partial^2 F}{\partial \mathbf{n}_1^2}(0,\tfrac{2}{3},\tfrac{1}{3})-S\right)\Big/C,\quad C=\tfrac{2240}{243}$$

and

$$S=\sum_{\substack{k=0\\k\neq 3,4}}^{7}72\left(\sum_{\rho_1+\rho_2+\rho_3=2}c_{\rho_1,7-k+\rho_2,k+\rho_3}B^2_{\rho_1\rho_2\rho_3}(\mathbf{n}_1)\binom{7}{k}\frac{2^k}{3^7}\right)$$
$$+\sum_{k=3}^{4}72\left(\sum_{\substack{\rho_1+\rho_2+\rho_3=2\\(\rho_1,\rho_2,\rho_3)\neq(2,0,0)}}c_{\rho_1,7-k+\rho_2,k+\rho_3}B^2_{\rho_1\rho_2\rho_3}(\mathbf{n}_1)\binom{7}{k}\frac{2^k}{3^7}\right)$$

$$c_{333}=\tfrac{6561}{560}\left(F(\tfrac{1}{3},\tfrac{1}{3},\tfrac{1}{3})-\sum_{\substack{r+s+t=9\\(r,s,t)\neq(3,3,3)}}c_{rst}B^9_{rst}(\tfrac{1}{3},\tfrac{1}{3},\tfrac{1}{3})\right)$$

is a univariate Bernstein polynomial of degree 9. Step 3 is to determine the final coefficient so as to preserve as high a degree of polynomial precision as possible.

Step 1. Require the normal derivative of N along each edge to be seventh degree. We have

$$\left.\frac{\partial N}{\partial \mathbf{n}_i}\right|_{e_i}=9\sum_{k=0}^{8}s_k^{(i)}B_k^8(b_{i-1}),\quad i=1,2,3;\quad b_0\equiv b_3$$

where

$$s_k^{(1)}=c_{1,8-k,k}+(h_1-1)c_{0,9-k,k}-h_1c_{0,8-k,k+1},$$

$$s_k^{(2)}=-h_2c_{k+1,0,8-k}+c_{k,1,8-k}+(h_2-1)c_{k,0,9-k},$$

$$s_k^{(3)}=(h_3-1)c_{9-k,k,0}-h_3c_{8-k,k+1,0}+c_{8-k,k,1}.$$

Clearly, $(\partial N/\partial \mathbf{n}_i)|_{e_i}$ is generally a univariate polynomial of degree 8. The above polynomials will actually be of degree 7 if

$$s_0^{(i)}-8s_1^{(i)}+28s_2^{(i)}-56s_3^{(i)}+70s_4^{(i)}-56s_5^{(i)}+28s_6^{(i)}-8s_7^{(i)}+s_8^{(i)}=0,\quad i=1,2,3.$$

These three equations can be used to solve for c_{144}, c_{414}, and c_{441}. The expression for c_{144} is given in Table 3. The others are easily obtained by symmetry.

Table 3
Examples of coefficients based on condensation of parameters, condensed scheme.

$c_{144} = \frac{1}{70}s_1 + (1-h_1)c_{054} + h_1 c_{045}$
$s_1 = -s_0^{(1)} + 8s_1^{(1)} - 28s_2^{(1)} + 56s_3^{(1)} + 56s_5^{(1)} - 28s_6^{(1)} + 8s_7^{(1)} - s_8^{(1)}$
$s_k^{(1)} = c_{1,8-k,k} + (h_1 - 1)c_{0,9-k,k} - h_1 c_{0,8-k,k+1}$, $k = 0,1,2,3,5,6,7,8$
$c_{432} = r_3 - (h_3-1)^2 c_{630} + 2h_3(h_3-1)c_{540} - 2(h_3-1)c_{531} - h_3^2 c_{450} + 2h_3 c_{441}$
$r_3 = \frac{1}{35}(4r_0^{(3)} - 21r_1^{(3)} + 42r_2^{(3)} + 21r_5^{(3)} - 14r_6^{(3)} + 3r_7^{(3)})$
$r_k^{(3)} = (h_3-1)^2 c_{9-k,k,0} - 2h_3(h_3-1)c_{8-k,k+1,0} + 2(h_3-1)c_{8-k,k,1} + h_3^2 c_{7-k,k+2,0} - 2h_3 c_{7-k,k+1,1} + c_{7-k,k,2}$, $k = 0,1,2,5,6,7$

Step 2. Constrain the second normal derivative of N along each edge to be fifth degree.

$$\left.\frac{\partial^2 N}{\partial \mathbf{n}_i^2}\right|_{e_i} = 72\sum_{k=0}^{7} r_k^{(i)} B_k^7(b_{i-1}),\quad i = 1,2,3;\quad b_0 \equiv b_3$$

where

$$r_k^{(1)} = c_{2,7-k,k} + 2(h_1-1)c_{1,8-k,k} - 2h_1 c_{1,7-k,k+1} + (h_1-1)^2 c_{0,9-k,k} - 2h_1(h_1-1)c_{0,8-k,k+1} + h_1^2 c_{0,7-k,k+2},$$
$$r_k^{(2)} = h_2^2 c_{k+2,0,7-k} - 2h_2 c_{k+1,1,7-k} - 2h_2(h_2-1)c_{k+1,0,8-k} + c_{k,2,7-k} + 2(h_2-1)c_{k,1,8-k} + (h_2-1)^2 c_{k,0,9-k},$$
$$r_k^{(3)} = (h_3-1)^2 c_{9-k,k,0} - 2h_3(h_3-1)c_{8-k,k+1,0} + 2(h_3-1)c_{8-k,k,1} + h_3^2 c_{7-k,k+2,0} - 2h_3 c_{7-k,k+1,1} + c_{7-k,k,2}.$$

For these to be fifth degree rather than seventh degree it is necessary that:

$$-r_0^{(i)} + 7r_1^{(i)} - 21r_2^{(i)} + 35r_3^{(i)} - 35r_4^{(i)} + 21r_5^{(i)} - 7r_6^{(i)} + r_7^{(i)} = 0,$$
$$7r_0^{(i)} - 42r_1^{(i)} + 105r_2^{(i)} - 140r_3^{(i)} + 105r_4^{(i)} - 42r_5^{(i)} + 7r_6^{(i)} = 0,\quad i = 1,2,3.$$

The only terms which involve undetermined coefficients are $r_3^{(i)}$ and $r_4^{(i)}$. From the above equations,

$$r_3^{(i)} = \tfrac{1}{35}(4r_0^{(i)} - 21r_1^{(i)} + 42r_2^{(i)} + 21r_5^{(i)} - 14r_6^{(i)} + 3r_7^{(i)}),$$
$$r_4^{(i)} = \tfrac{1}{35}(3r_0^{(i)} - 14r_1^{(i)} + 21r_2^{(i)} + 42r_5^{(i)} - 21r_6^{(i)} + 4r_7^{(i)}).$$

These expressions can be used to solve for c_{432}, c_{423}, c_{342}, c_{243}, c_{324} and c_{234}. The expression for c_{432} is given in Table 3. Again, by symmetry, we can determine explicit expressions for the other coefficients.

We note that it is not actually necessary to constrain the normal derivative to the edge; we could constrain the derivatives in any direction not tangent to the edge. Normal derivatives have been used here and in the preceding section because they are dependent on the data in a symmetric fashion and are unambiguous.

Step 3. Maintain polynomial precision of degree 7.

Because the constraint in Step 2 is not generally satisfied by eighth degree polynomials, the polynomial precision of the condensed scheme cannot be higher than 7. Although there are

Table 4
Expression for c_{333}, based on minimization technique to maintain polynomial precision. Condensed scheme.

$$\begin{aligned}
c_{333} = (& -8c_{900} + 36c_{810} + 36c_{801} - 18c_{720} - 252c_{711} - 18c_{702} - 105c_{630} + 441c_{621} + 441c_{612} - 105c_{603} + 99c_{540} \\
& + 234c_{531} - 1674c_{522} + 234c_{513} + 99c_{504} + 99c_{450} - 990c_{441} + 1395c_{432} + 1395c_{423} - 990c_{414} + 99c_{405} - 105c_{360} \\
& + 234c_{351} + 1395c_{342} + 1395c_{324} + 234c_{315} - 105c_{306} - 18c_{270} + 441c_{261} - 1674c_{252} + 1395c_{243} + 1395c_{234} \\
& - 1674c_{225} + 441c_{216} - 18c_{207} + 36c_{180} - 252c_{171} + 441c_{162} + 234c_{153} - 990c_{144} + 234c_{135} + 441c_{126} - 252c_{117} \\
& + 36c_{108} - 8c_{090} + 36c_{081} - 18c_{072} - 105c_{063} + 99c_{054} + 99c_{045} - 105c_{036} - 18c_{027} + 35c_{018} - 8c_{009})/3720
\end{aligned}$$

many possible methods for determining the value of the final coefficient, c_{333}, we choose a technique which will retain as high a degree of polynomial precision as possible. By using a procedure for determining c_{333} which is itself precise for polynomials of degree 7 or more, we can maintain the seventh degree polynomial precision of the interpolant.

One way to do this is by a minimization technique: we minimize the sum of the squares of all ninth derivatives of $N(\mathbf{b})$ with respect to c_{333}. Since this procedure is precise for eighth degree polynomials, the seventh degree polynomial precision of the interpolant is maintained.

A program in REDUCE, a symbol manipulation language [Hearn '83], was written to solve for the corresponding value of c_{333} in terms of the other coefficients. The resulting expression for c_{333} is given in Table 4.

An example and pictures

In this section, the condensed version of the interpolant is applied to data drawn from a bivariate function first introduced in [Franke '82]. While in most applications an underlying primitive function does not exist, such an example does give an indication of how the interpolant works over a defined data set. The primitive function is given by

$$\begin{aligned}
f(x, y) = {} & 0.75\ \mathrm{e}^{-((9x-2)^2+(9y-2)^2)/4} + 0.75\ \mathrm{e}^{-((9x+1)^2/49+(9y+1)/10)} \\
& + 0.50\ \mathrm{e}^{-((9x-7)^2+(9y-3)^2)/4} - 0.20\ \mathrm{e}^{-((9x-4)^2+(9y-7)^2)}.
\end{aligned}$$

The surface from which the data are taken contains two maxima, one minimum and a saddle point in the domain of interest, the unit square.

Thirty-six points on the unit square were chosen and triangulated using Little's method given in [Barnhill '77]. The points and triangulation are given in Table 5, and a plot of the triangulation in Fig. 1. Data through fourth derivatives were then evaluated at each point. FORTRAN code for the definition of the condensed interpolant was written on the University of Utah College of Science DEC-20. (A working version is available from the Department of Mathematics, University of Utah.) The code for the condensed interpolant was tested using MICROSCOPE, a software system for analysing multivariate functions. (See [Alfed, Harris '84].) MICROSCOPE was used to demonstrate that the interpolant obtained by applying the code to the data drawn from Franke's function is in fact C^2 by testing continuity at selected points across triangle edges and at vertices.

The pictures were drawn on an ADAGE monitor and photographed using a Dunn Instruments 632 Camera System.

Fig. 2 depicts Franke's primitive function. Fig. 3 shows the condensed nonic interpolant based on data from 36 points. The two surfaces are visually indistinguishable from each other within the resolution of the graphics equipment used.

Table 5
Points and triangulation used in computing interpolant applied to Franke's function.

Points			Points			Triangulation				Triangulation				Triangulation			
#	x	y	#	x	y	#	points			#	points			#	points		
1	0.00	0.00	19	0.80	0.85	1	11	15	14	19	36	21	20	37	20	33	19
2	0.50	0.00	20	0.85	0.65	2	14	8	11	20	7	29	34	38	28	35	17
3	1.00	0.00	21	1.00	0.50	3	15	9	36	21	34	8	7	39	29	13	12
4	0.15	0.15	22	1.00	1.00	4	11	6	9	22	6	2	5	40	4	1	26
5	0.70	0.15	23	0.50	1.00	5	6	11	8	23	10	9	5	41	26	34	4
6	0.50	0.20	24	0.10	0.85	6	14	16	13	24	18	23	17	42	31	16	24
7	0.25	0.30	25	0.00	1.00	7	14	7	8	25	20	19	18	43	27	5	2
8	0.40	0.30	26	0.25	0.00	8	8	26	6	26	34	29	4	44	10	32	21
9	0.75	0.40	27	0.75	0.00	9	15	20	18	27	6	26	2	45	28	17	23
10	0.85	0.25	28	0.25	1.00	10	14	17	35	28	16	12	13	46	10	3	32
11	0.55	0.45	29	0.00	0.25	11	9	15	11	29	29	7	13	47	19	22	30
12	0.00	0.50	30	0.75	1.00	12	17	14	15	30	26	8	34	48	30	18	19
13	0.20	0.45	31	0.00	0.75	13	7	14	13	31	5	27	10	49	33	20	21
14	0.45	0.55	32	1.00	0.25	14	16	14	35	32	18	30	23	50	1	4	29
15	0.60	0.65	33	1.00	0.75	15	17	15	18	33	35	28	24	51	28	25	24
16	0.25	0.70	34	0.19	0.19	16	20	15	36	34	35	24	16	52	24	25	31
17	0.40	0.80	35	0.32	0.75	17	6	5	9	35	16	31	12	53	3	10	27
18	0.65	0.75	36	0.79	0.46	18	9	10	36	36	21	36	10	54	22	19	33

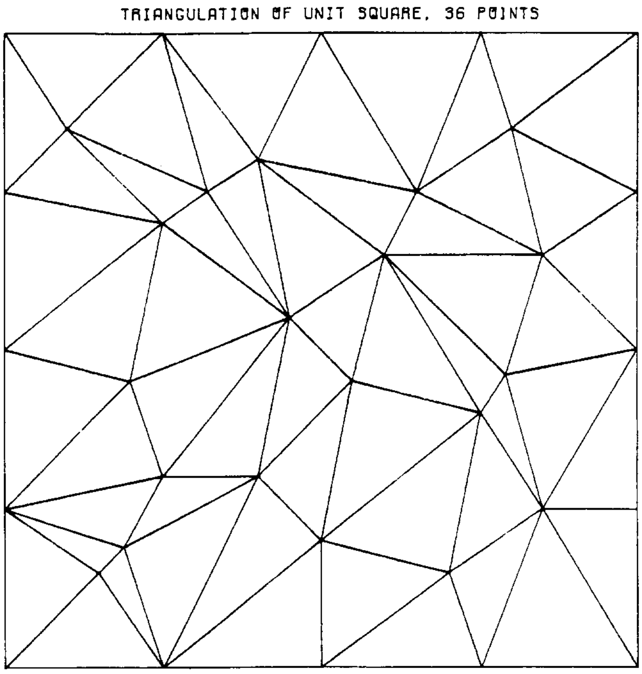

Fig. 1. Triangulation of the data given in Table 5.

Fig. 4 is a multiple (200 times) of the absolute error. (The surface in Figs. 2,3, and 5 are oriented in the same manner, while Fig. 4 has been rotated upwards by 10 degrees to aid in viewing the irregularities.) None of the data points is placed directly at the extrema, and so the

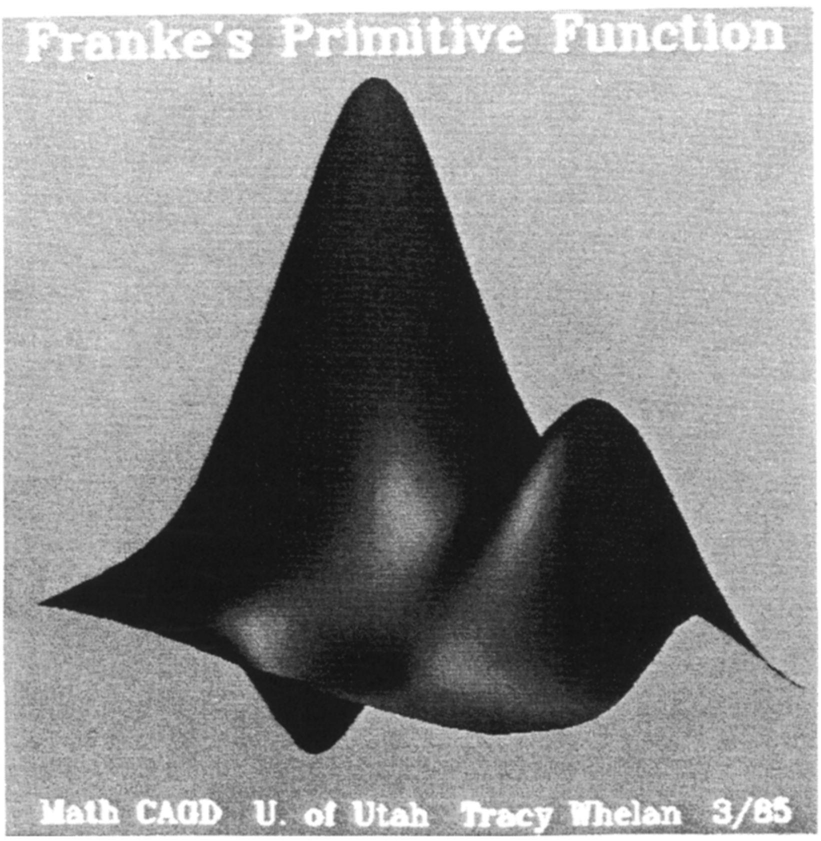

Fig. 2. Franke's primitive function.

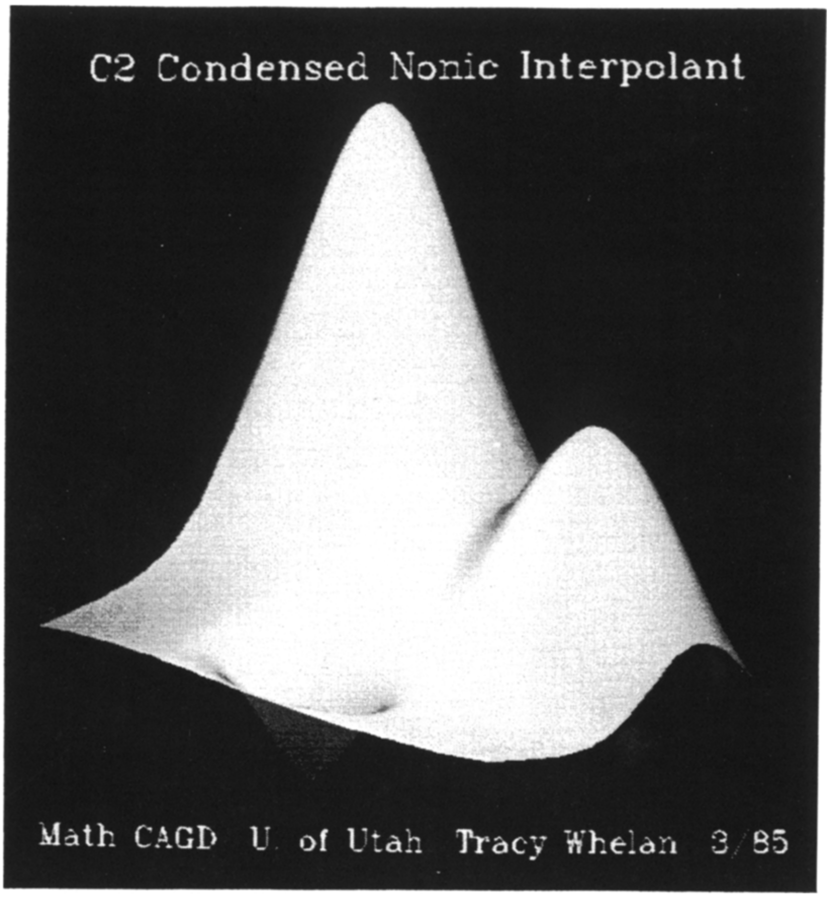

Fig. 3. C^2 condensed nonic interpolant, based on exact data.

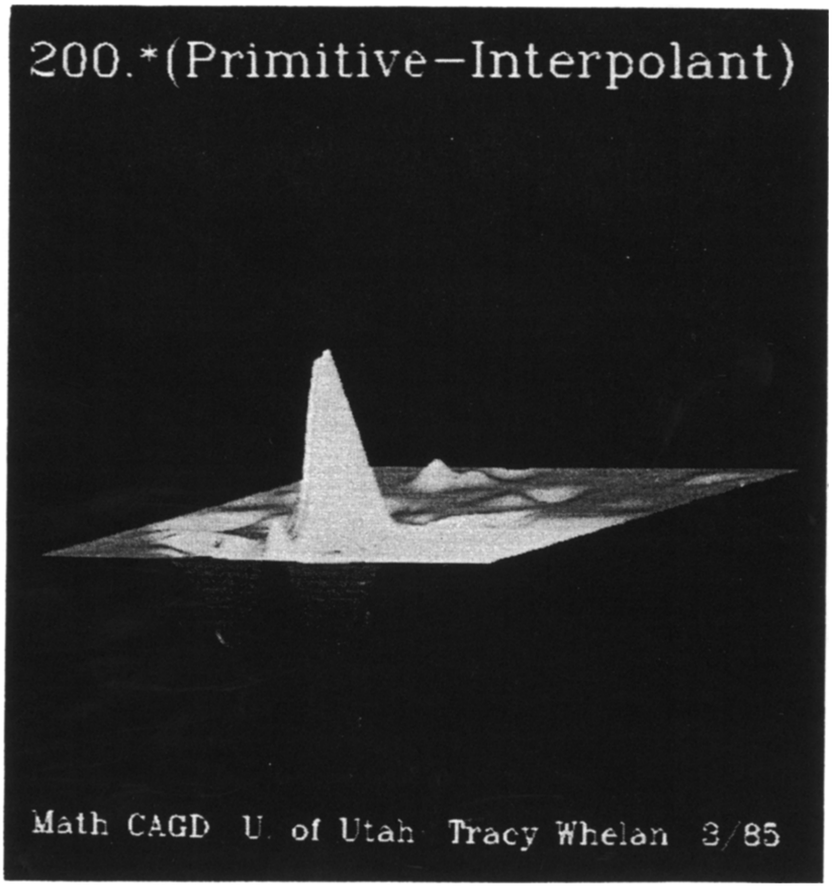

Fig. 4. A multiple of the error surface produced by interpolating to data drawn from Franke's function by C^2 nonic interpolant, condensed scheme.

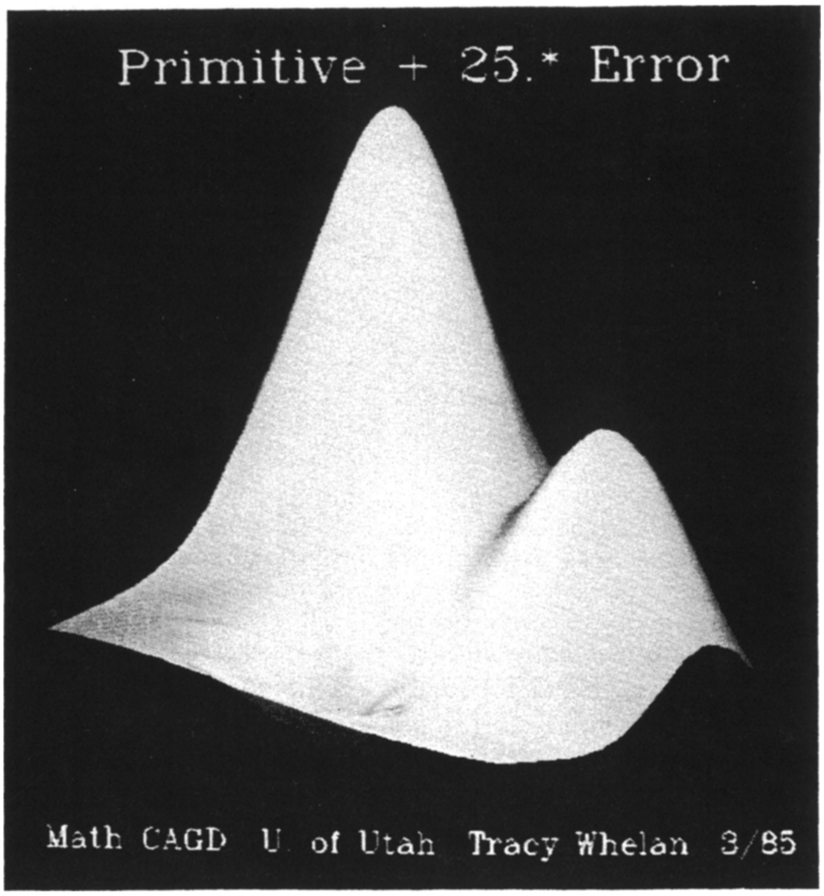

Fig. 5. Franke's primitive function plus a multiple of the error.

interpolating surface cannot be expected to represent the primitive function especially well near those points. Fig. 5 demonstrates this; it is Franke's primitive function plus a smaller multiple (25 times) of the error. The area where distortion in the primitive function first becomes apparent is in the region surrounding the minimum. This, together with Fig. 4 representing the error, suggests that the largest deviations from the values of the primitive function occur near areas of the most rapid change of the function.

Competing methods

We include references to a few of the many competing surface schemes. We restrict attention to interpolatory schemes which are locally defined over a triangulation, are at least C^1, and depend on discrete data. The references are not meant to be complete.

The quintic interpolant given in [Barnhill, Farin '81] is the C^1 analogue of the C^2 nonic interpolant whose representation is given in this paper. It uses C^2 data (derivative information through all second derivatives) and has cubic precision.

A short history of triangular patch interpolants can be found in [Barnhill '83], which includes a description of a rational C^1 scheme which uses C^1 data.

The element due to [Clough, Tocher '65] is a piecewise polynomial C^1 interpolant based on C^1 data. Each triangle in the triangulation formed from the data is split by connecting the vertices to the barycenter, and the interpolant is cubic over each of these resulting triangles. Implementation of this scheme is described by [Lawson '77]. [Farin '83] has produced a modified Clough–Tocher interpolant which is also a piecewise polynomial C^1 function.

[Alfeld, Barnhill '84] describes methods for producing both uncondensed and condensed C^2 interpolants which use C^2 data. The uncondensed scheme has quintic precision, while the precision of the condensed scheme is cubic.

Conclusion

We have given representations, in Bézier form, of both uncondensed and condensed versions of a piecewise nonic interpolant over triangles which is C^2. The polynomial precision of the uncondensed scheme is 9. The condensed scheme was defined using a minimization technique which ensured seventh degree polynomial precision.

Acknowledgments

The author wishes to express her gratitude to Professor R.E. Barnhill for his long-term support and encouragement, as well as for suggesting the perusal of the problem. Dr. P. Alfeld suggested the minimization technique used to compute the final parameter in the derivation of the condensed version of the interpolant, and gave generously of his time to aid in the production of the pictures in this article. The author would like to thank the referees of this article for their careful reading of the manuscript and the numerous helpful suggestions they made for its improvement.

This research was supported by the Department of Energy under contract DE-AC02-85ER12046 to the University of Utah.

References

Alfeld, P. (1984), Multivariate scattered data derivative generation by functional minimization, MRC Technical Summary Report #2703, Univ. Of Wisconsin.

Alfeld, P. and Barnhill, R.E. (1984), A transfinite C^2 interpolant over triangles, Rocky Mountain J. Math. 14, 17–40.

Alfeld, P. and Harris, W. (1984), MICROSCOPE: A software system for multivariate analysis, MRC Technical Summary Report #2701, Univ. of Wisconsin.

Barnhill, R.E. (1977), Representation and approximation of surfaces, in: J.R. Rice, ed., *Mathematical Software III*, Academic Press, New York, 68–119.

Barnhill, R.E. (1983), Computer aided surface representation and design, in: R.E. Barnhill and W. Boehm, eds., *Surfaces in Computer Aided Geometric Design*, North-Holland, Amsterdam, 1–24.

Barnhill, R.E. and Farin, G. (1981), C^1 quintic interpolation over triangles: two explicit representations, Int. J. For Num. Meth. in Eng. 17, 1763–1778.

Clough, R.W. and Tocher, J.L. (1965), Finite element stiffness matrices for analysis of plates in bending, Proc. Conf. Matrix Methods in Struct. Mech., Air Force Inst. of Tech., Wright-Patterson A.F.B., Ohio.

Farin, G. (1983), Smooth interpolation to scattered 3D data, in: R.E. Barnhill and W. Boehm, eds., *Surfaces in Computer Aided Geometric Design*, North-Holland, Amsterdam, 43–63.

Franke, R. (1982), Scattered data interpolation: tests of some methods, Math. Comp. 38, 181–200.

Hearn, A.C. (1983), *Reduce User's Manual*, Version 3.0, The Rand Corp., Santa Monica, CA 90406.

Lawson, C.L. (1977), Software for C^1 surface interpolation, in: J.R. Rice, ed., *Mathematical Software III*, Academic Press, New York, 161–194.

Strang, G. and Fix, G.J. (1973), *An Analysis of the Finite Element Method*, Prentice-Hall, Englewood Cliffs, NJ.

Ženišek, A. (1973), Hermite interpolation on simplices in the finite element method, Proc. Equadiff. III, 3rd Czech. Conf. Diff. Equa. Appl., Brno 1972, 271–277.

Zienkiewicz, O.C. (1977), *The Finite Element Method*, McGraw-Hill, London, UK.