COMMUNICATIONS IN APPLIED NUMERICAL METHODS, Vol. 7, 479–485 (1991)

A C^1 TRIANGULAR INTERPOLANT SUITABLE FOR SCATTERED DATA INTERPOLATION

T. N. T. GOODMAN

Department of Mathematics and Computer Science, The University, Dundee DD1 4HN, U.K.

AND

H. B. SAID

School of Mathematical and Computer Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia

SUMMARY

We present here a method of constructing a triangle interpolant which interpolates position and partial derivatives specified at the three vertices of the triangle. The method employs the cubic Bézier triangular patch technique. The data given enable us to determine the appropriate Bézier control points so that adjacent patches meet with C^1 continuity. However, the interior control point for the patch is replaced by three separate points, due to the implementation of three local schemes, each of which satisfies the boundary conditions on only one side of the triangle. Convex combination is used to blend these three local schemes.

INTRODUCTION

Among the many methods for scattered data interpolation (see Franke[1]), a class of methods triangulate the domain with the data points at the vertices and then define a local interpolant over each triangle. These methods have the advantage of being local, i.e. the surface at any point depends only on the data 'close' to that point. In order to gain a smooth surface it is usual to employ certain derivative values on the boundaries of the triangles. If these are not given, they can be estimated from the data values and can be used as 'shape parameters' to modify the surface to give a more pleasing result.

In the method presented in this paper we gain a C^1 surface by prescribing first-order derivatives at the vertices. The interpolant on each triangle is a rational function gained by blending three cubic polynomials. It might seem redundant to propose yet another such scheme, but to our surprise our simple method involves considerably less computation than others we have seen in the literature and on the functions tested so far yield results at least as accurate. We start by giving a brief review of the Bézier representation of a cubic polynomial on a triangle, which is then used to describe our local interpolant. Finally, this is applied to scattered data interpolation and results are given for two test functions.

THE CUBIC BÉZIER TRIANGULAR PATCH

Consider a triangle T with vertices V_1, V_2, V_3 and barycentric co-ordinates u, v, w such that any point on the triangle can be expressed as

$$V = uV_1 + vV_2 + wV_3, \qquad u + v + w = 1$$

0748–8025/91/060479–07\$05.00
© 1991 by John Wiley & Sons, Ltd.

Received 28 June 1990
Revised 30 October 1990

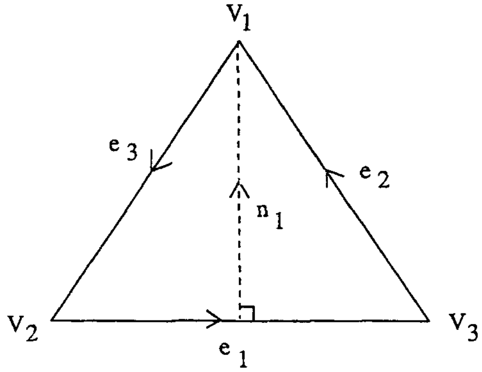


Figure 1. Notations on the triangle T

We denote by e_i the side opposite the vertex V_i, from V_{i+1} to V_{i-1} (where indices are always taken modulo 3) (see Figure 1). Then a cubic Bézier triangular patch is defined as

$$P(u, v, w) = u^3 b_{3,0,0} + 3u^2 v b_{2,1,0} + 3u^2 w b_{2,0,1} + 3uv^2 b_{1,2,0} + 3uw^2 b_{1,0,2} + v^3 b_{0,3,0} + 3v^2 w b_{0,2,1} + 3vw^2 b_{0,1,2} + w^3 b_{0,0,3} + 6uvw b_{1,1,1} \quad (1)$$

where $b_{r,s,t}$ are the Bézier ordinates or control points of P.

The derivative of P with respect to the direction $z = (z_1, z_2, z_3) = z_1 V_1 + z_2 V_2 + z_3 V_3$, $z_1 + z_2 + z_3 = 0$ is given by

$$\frac{\partial P}{\partial z} = \frac{\partial P}{\partial u} z_1 + \frac{\partial P}{\partial v} z_2 + \frac{\partial P}{\partial w} z_3 \quad (2)$$

Further details of the triangular patch can be found in Farin.[2]

THE SCHEME

We assume that the data $F(V_i)$ and its first partial derivatives $F_x(V_i)$ and $F_y(V_i)$ for $i = 1, 2, 3$ are given at the vertices. Then the derivative along the side e_i is given by

$$\frac{\partial F}{\partial e_i} = (x_{i-1} - x_{i+1}) \frac{\partial F}{\partial x} + (y_{i-1} - y_{i+1}) \frac{\partial F}{\partial y}$$

From the data given we can now determine all the $b_{r,s,t}$ except $b_{1,1,1}$, e.g.

$$b_{3,0,0} = F(V_1)$$

$$b_{2,1,0} = F(V_1) + \frac{F_{e3}(V_1)}{3}$$

$$b_{2,0,1} = F(V_1) - \frac{F_{e2}(V_1)}{3}$$

By symmetry we can obtain the other six control points. Now it remains to determine $b_{1,1,1}$ so that the C^1 requirement on all sides of the triangle is satisfied. The scheme proceeds as follows:

(i) We determine $b^i_{1,1,1}$ so that the C^1 condition on the boundary e_i only is satisfied. We define a local scheme P_i by replacing $b_{1,1,1}$ in (1) with $b^i_{1,1,1}$.

(ii) Convex combination is then employed to blend these three local schemes so that conditions on all sides of the triangle are satisfied.

Local scheme

In the construction for the normal derivatives on e_i we will use linear interpolation of the normal derivative values at the vertices. For simplicity we shall consider e_1, and the rest follows by symmetry. The interior control point $b^1_{1,1,1}$ is used to replace $b_{1,1,1}$ in (1).

Let n_1 be the inward normal direction to the edge e_1 (see Figure 1). Then

$$\begin{aligned} n_1 &= -e_3 + \frac{(e_3 \cdot e_1)}{|e_1|^2} e_1 \\ &= (1, -1, 0) - h_1(0, -1, 1) \\ &= (1, h_1 - 1, -h_1) \end{aligned}$$

where

$$h_1 = -\frac{(e_3 \cdot e_1)}{|e_1|^2}$$

By using (1) and (2), the normal derivative on e_1 is given by

$$\begin{aligned} \frac{\partial P_1}{\partial n_1} &= 3(b_{1,2,0} - b_{0,3,0} - h_1(b_{0,2,1} - b_{0,3,0}))v^2 \\ &\quad + 6(b^1_{1,1,1} - b_{0,2,1} - h_1(b_{0,1,2} - b_{0,2,1}))vw \\ &\quad + 3(b_{1,0,2} - b_{0,1,2} - h_1(b_{0,0,3} - b_{0,1,2}))w^2 \end{aligned} \tag{3}$$

Since we need the normal derivative to be linear on e_1, then, from (3),

$$\begin{aligned} 2(b^1_{1,1,1} - b_{0,2,1} - h_1(b_{0,1,2} - b_{0,2,1})) &= (b_{1,2,0} - b_{0,3,0} - h_1(b_{0,2,1} - b_{0,3,0})) \\ &\quad + (b_{1,0,2} - b_{0,1,2} - h_1(b_{0,0,3} - b_{0,1,2})) \end{aligned}$$

Hence,

$$\begin{aligned} b^1_{1,1,1} &= \tfrac{1}{2}(b_{1,2,0} + b_{1,0,2} + h_1(2b_{0,1,2} - b_{0,2,1} - b_{0,0,3}) \\ &\quad + (1 - h_1)(2b_{0,2,1} - b_{0,3,0} - b_{0,1,2})) \end{aligned} \tag{4}$$

Similarly we define $b^2_{1,1,1}$ and $b^3_{1,1,1}$ for the local schemes P_2 and P_3.

Final scheme

We define our interpolant as a convex combination of all the local schemes, namely

$$\begin{aligned} P(u, v, w) &= \frac{(v^2w^2P_1 + w^2u^2P_2 + u^2v^2P_3}{(v^2w^2 + w^2u^2 + u^2w^2)} \\ &= u^3b_{3,0,0} + 3u^2vb_{2,1,0} + 3u^2wb_{2,0,1} + 3uv^2b_{1,2,0} + 3uw^2b_{1,0,2} \\ &\quad + v^3b_{0,3,0} + 3v^2wb_{0,2,1} + 3vw^2b_{0,1,2} + w^3b_{0,0,3} \\ &\quad + 6uvw\,\frac{v^2w^2b^1_{1,1,1} + w^2u^2b^2_{1,1,1} + u^2v^2b^3_{1,1,1}}{v^2w^2 + w^2u^2 + u^2v^2} \end{aligned} \tag{5}$$

The scheme satisfies the C^1 requirements on the three sides of the triangle. Although it appears that singularities occur at the vertices, it can be easily shown that these singularities are removable.

It should be noted that we have to compute the control points b only once. Once this has been done, then to compute a point in the triangle we need 38 multiplications, 1 division and 13 additions. However, in Lawson[3] it needs 55 multiplications, 4 divisions and 65 additions, while in Renka and Cline[4] it needs 54 multiplications, 8 divisions and 62 additions. This shows that our scheme is much more efficient than those two methods with respect to operation count. There are also methods which employ cubic Hermite functions to define the boundaries of the triangle, but this requires far more multiplications.

APPLICATION TO SCATTERED DATA INTERPOLATION

In scattered data interpolation we wish to construct a smooth bivariate function which interpolates given data $F(x_i, y_i), i = 1, \ldots, n$. An extensive survey on this problem was made by Franke.[1]

Our method of solution to this problem consists of the following three steps:

1. triangulate the nodes (x_i, y_i), see Lawson[3]
2. estimate the partial derivatives at the nodes, see Renka and Cline[4]
3. compute point in the triangle by the triangular scheme.

In this paper we shall not consider steps 1 and 2 above, and we assume that these have been implemented. We then use our proposed scheme in step 3. For the implementation of the scheme we shall adopt the 36 data nodes and triangulation given in Whelan.[5] We shall use two test functions which were considered in Renka and Cline:[4]

$$\begin{aligned} F1 &= 0\cdot75\exp(-((9x-2)^2+(9y-2)^2/4) \\ &\quad + 0\cdot75\exp(-(9x+1)^2/49-(9y+1)/10) \\ &\quad + 0\cdot5\exp(-((9x-7)^2+(9y-3)^{2)}/4) \\ &\quad - 0\cdot2\exp(-(9x-4)^2-(9y-7)^2) \\ F2 &= (1\cdot25+\cos(5\cdot4y))/(6+6(3x-1)^2) \end{aligned}$$

Values at the 36 data nodes were computed, and we take true derivatives for the implementation of the scheme. One could estimate the derivatives at these points by using method suggested in Renka and Cline.[4] Values at 725 grid points of a 25×25 uniform mesh in the unit square were computed, and maximum absolute and mean errors were considered. These errors are listed in Table I. Renka and Cline[4] listed these errors for various methods implemented using 33 data nodes and 100 data nodes. It is seen that our method has better accuracy when compared with the 33 data nodes and equal merit when compared with the 100 nodes.

Table I. Errors of the test functions

Test functions	Max. error	Mean error
$F1$	0·064142	0·005479
$F2$	0·006305	0·000743

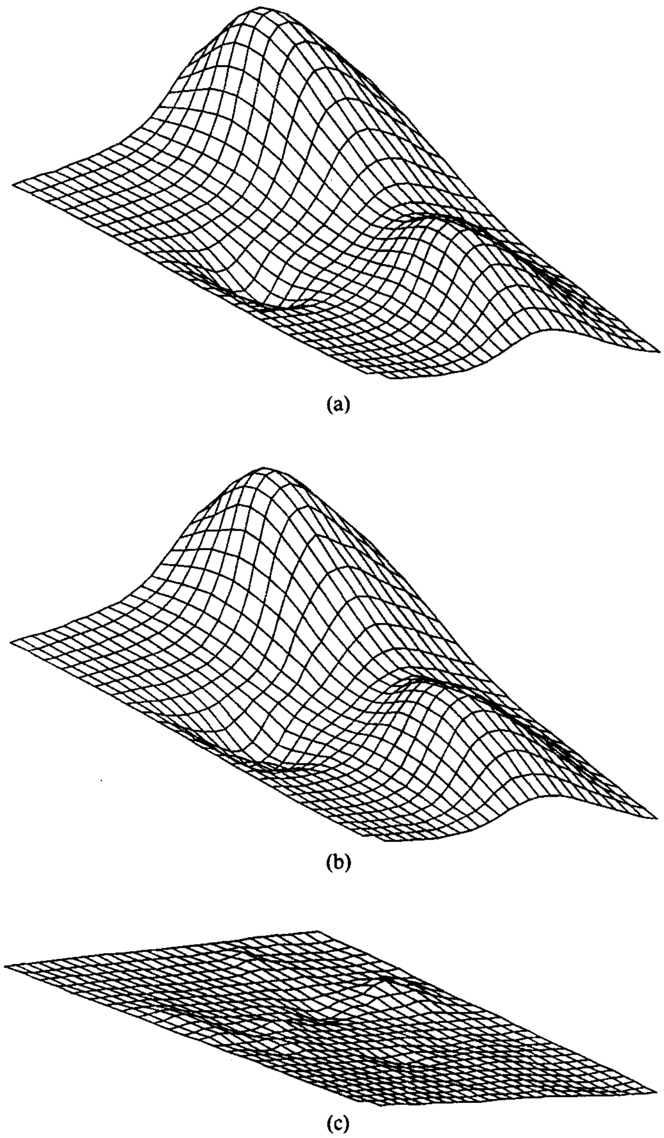
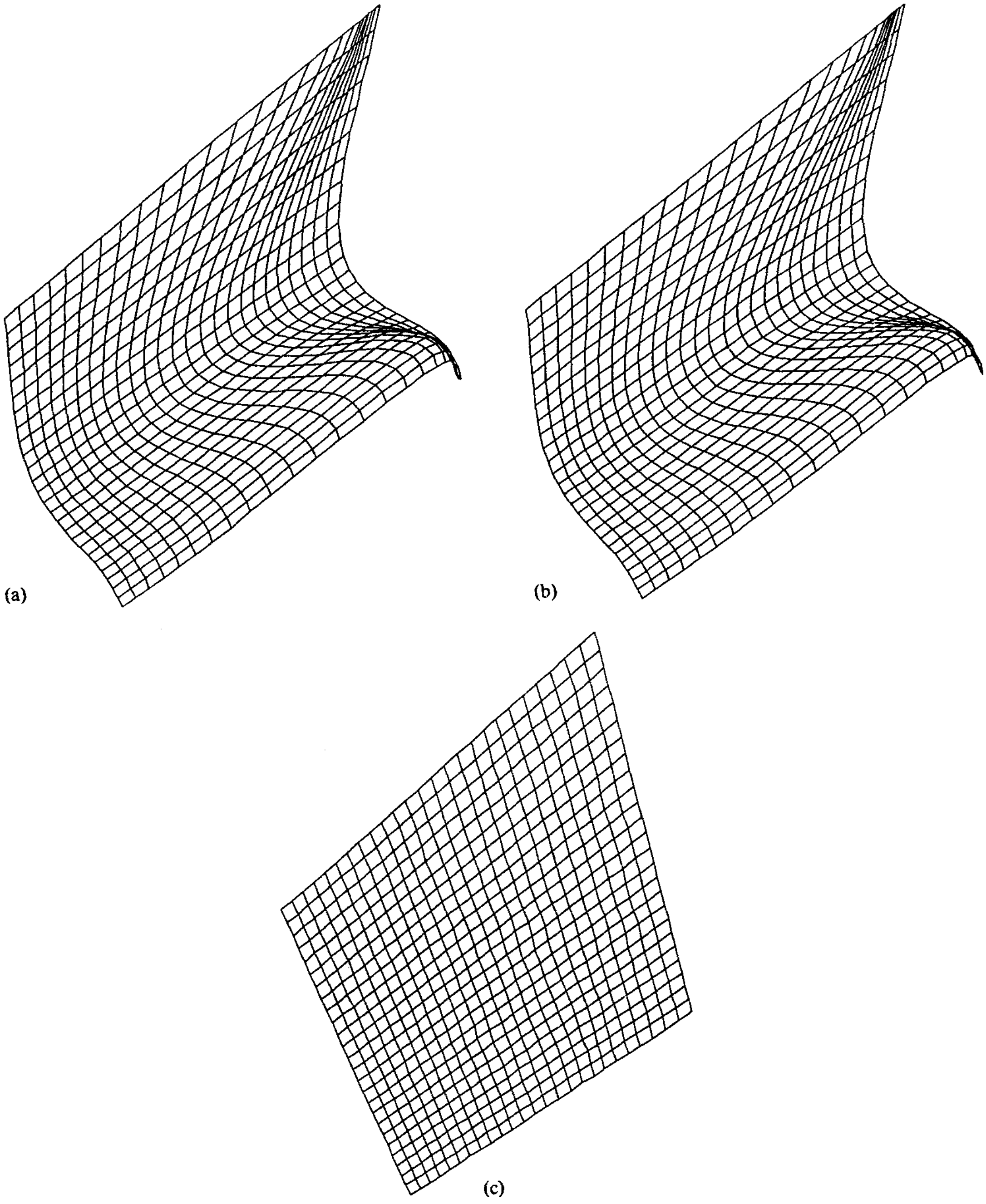


Figure 2. Test function $F1$: (a) surface of true function; (b) surface given by our method; (c) error of (a) and (b)

For the illustration of our method we show the surfaces of the test functions. Figures 2 and 3 correspond with the test functions 1 and 2, repectively. Figures (a) are the surfaces of the true functions, (b) are the surfaces given by our method and (c) are of the errors of (a) and (b).

CONCLUSION

We have proposed a method of constructing a smooth surface through scattered data. The method is based on the Bézier triangular patch. It requires significantly less computation and produces comparable accuracy to some of the existing methods.



(a) (b) (c)

Figure 3. Test function F_2: (a) surface of true function; (b) surface given by our method; (c) error of (a) and (b)

ACKNOWLEDGEMENT

We should like to thank Dr. K. Unsworth for helping us with the graphics packages. This work was performed during the second author's sabbatical leave at the University of Dundee under the Association of Commonwealth Universities Fellowship.

REFERENCES

1. R. Franke, 'Scattered data interpolation: Test of some methods', *Math. Comput.*, **38**, 181–200 (1988).
2. G. Farin, 'Triangular Bernstein–Bézier patches', *Computer-Aided Geometric Design*, **3**, 83–127 (1986).
3. C. L. Lawson, 'Software for C^1 surface interpolation', in *Mathematical Software* III, J. R. Rice, (ed.), Academic Press, New York, 161–194 (1977).
4. R. J. Renka and K. Cline, 'A triangle-based C^1 interpoltion method', *The Rocky Mountain Journal of Maths.*, **14**, 223–237 (1984)
5. T. Whelan, 'A representation of a C^2 interpolant over triangles', *Computer-Aided Geometric Design*, **3**, 53–66 (1986).