Polynomial Precision Triangular Interpolants

Stephen Mann

Abstract. Piecewise polynomial surfaces constructed to interpolate triangulated data using triangular Bézier patches typically construct patches to meet with C^1 or higher continuity. However, the continuity conditions are not enough to set all the degrees of freedom in these patches. The remaining degrees of freedom may be used as shape parameters, but good settings of these parameters are important for achieving good shape. In this paper, I investigate another approach to constructing triangular interpolants. Rather than building continuity into the initial construction, I first construct patches that interpolate derivative information and achieve polynomial precision, setting all the degrees of freedom in the process. While these interpolants are only C^0, I then adjust the control points to increase the continuity without decreasing the polynomial precision.

§1. Introduction

There are many solutions to the following problem: Given a triangulated set of points in the plane, with height values and derivatives at the points, find a smooth function that interpolates the values and derivatives at the data points. Here smooth means for the surface to be C^k, with $k \geq 0$. See one of several survey papers for an overview of schemes to solve this problem [8,9]. In this paper, I will investigate piecewise polynomial schemes for interpolating such data. The standard method for interpolating this data with polynomials is to split the data triangles into three or more pieces [4,14], which allows for a C^1 construction with compatible twist vectors at the corners of the patches.

I am concerned with constructions that achieve polynomial precision. Similar constructions exists for (and this work is based on) split domain schemes [1,10] and for non-polynomial schemes [7]. The schemes I present will fit a single polynomial per data triangle without splitting.

Through out I will assume that we have a non-degenerate triangulation of the data with no overlapping triangles. Further, I will not worry about boundary conditions. I will use the multi-variate Bernstein-Bézier representation for polynomials with barycentric coordinates, and index the control

Mathematical Methods in CAGD: Oslo 2000
Tom Lyche and Larry L. Schumaker (eds.), pp. 1–3.

ISBN 0-8265-xxxx-x.

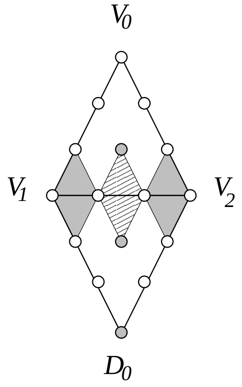

Fig. 1. Two cubics meeting along a common boundary..

points using standard multiindexing notation; for details, see any introductory text on CAGD, such as Farin's [6]. Many of the figures in this paper will be of schematic form; although the control points are placed regularly in a plane in the diagram, they represent points in three-space.

In my diagrams, when a group of control points are set using the derivative information at one of the V_i, I will circle those points with a dashed circle, as in Figure 5. Conversely, these dashed regions also indicate the number of derivatives needed at the corresponding V_i (i.e., at each data point, we need the position and the appropriate derivatives for setting the circled control points). Control points covered by more than one dashed circle will be set by averaging the values computed for each set of derivatives.

The approach in this paper is to use the derivative information to obtain polynomial precision with as low of degree patch as possible. Once the degree is determined, the maximum continuity with which the patches meet is also determined.

§2. Continuity

The continuity conditions for functional triangle Bézier patches are easily described in terms of the control points. Here I will just describe the C^0 and C^1 conditions; higher order conditions can be found elsewhere [11], with a bit of discussion of C^2 conditions appearing below.

For two patches to meet with C^0 continuity, they must have identical control points along their shared boundary. For the two patches to meet with C^1 continuity, adjacent panels of the control nets must be coplanar. For example, as illustrated schematically in Figure 1, the two patches must share boundary control points. To meet with C^1 continuity, each of the three panels of four control points must be coplanar.

The observation of Foley and Opitz [7] is that if the two center panels are not coplanar, we may make them coplanar by extending each panel over the other and averaging (Figure 2). Mann later generalized this averaging scheme to higher order continuity (Figure 3 illustrates the C^2 adjustment) [11,13]. One advantage of using such an averaging scheme is that if both patches have degree p polynomial precision before averaging, then they will continue to

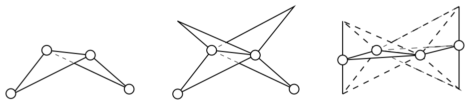

Fig. 2. Adjusting the panels to meet C^1..

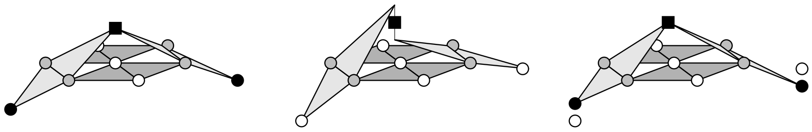

Fig. 3. Left: C^2 condition. Middle, right: C^2 adjustment..

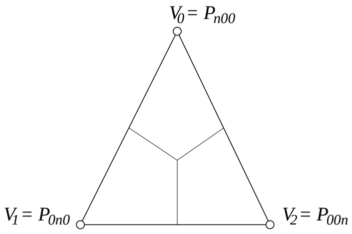

Fig. 4. Control point averaging for first scheme..

have degree p polynomial precision after averaging. This is the idea of the three schemes discussed in the next three sections.

§3. Scheme 1

This interpolant requires $\lfloor 2n/3 \rfloor$ complete derivatives at each data point, although I will not use all of the higher order derivatives. For P_{ijk}, with $i > j$, $i > k$, I use the data at V_0 to set the value of P_{ijk}. If $i < j = k$, then I compute two values using the data at V_0 and V_1 and average them. If $i = j = k$, then I compute three values, using the data at V_0, V_1, and V_2 and average the result. The remaining cases are handled in a similar fashion. This divides the data triangle into three regions as illustrated in Figure 4.

With this scheme, when constructing a degree n patch, if the data at all three corners comes from a single degree n polynomial, then this scheme reproduces that polynomial. In cases where a control vertex is the average of two or three values computed from the corner data, these values will be identical when the corner data come from a single polynomial.

The piecewise interpolant filling an arbitrary triangular network will create a C^0 piecewise polynomial surface. The C^0 continuity conditions are met because the boundary control points between two adjacent patches are computed using the same data.

Two neighboring patches built with this construction will only meet with C^0 continuity. However, note that only one or two panels along the boundary are out of C^1 alignment. Thus, if we apply the C^1 adjustment to these panels (a single panel when the degree is odd, two panels when the degree is even), then the two patches will meet with C^1 continuity.

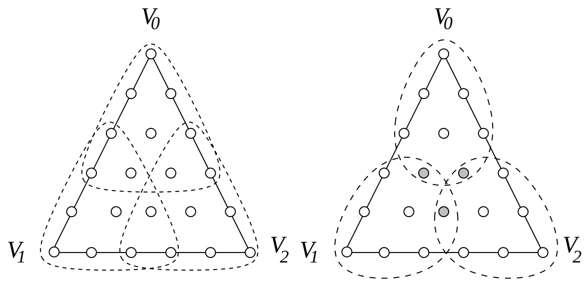

Fig. 5. Derivatives needed for the quintic version of first scheme..

We must be careful, however, that the vertices adjusted for one boundary do not affect the C^1 connection along another boundary. While we can apply the averaging construction given in this paper to pairs of patches, to apply it to a network of patches the degree of patch required by my constructions is $4k + 1$, where k is the level of continuity desired [15]. This requirement is needed so that the vertices adjusted to achieve C^k continuity along one boundary are not involved in the C^k conditions along another boundary.

As an example, Figure 5 indicates the derivatives needed for the quintic case. On the left I have an illustration of the full derivatives at the data points. However, I do not need nor use all of these derivatives. The figure on the right indicates the needed derivatives. In this latter figure, we set the white vertices using the derivative information at data point within the same region. The shaded vertices are set to the average of the two values computed from the derivative information at the data vertices in the two regions in which they lie. Note that we only have an averaged center value for even degree patches.

Having set the control points of the patch using the corner data results in a C^0 patch network. To construct C^1, we take this patch network, and adjust the shaded points of Figure 5 using the Foley-Optiz scheme.

§4. Scheme 2

With any interpolant that achieves polynomial precision, you need to set the degrees of freedom in a patch to match enough data to specify a polynomial of the desired degree. In many constructions, some of that data (position and derivatives) comes from the corner points, and the remaining data is specified as cross-boundary derivatives along the edges of the data triangle.

The interesting feature of the Foley-Optiz interpolant is that while it needs position and first derivatives at the corners, it needs no cross-boundary derivatives. Instead, it uses data stored at other vertices in the triangulation to completely specify the degrees of freedom in the patch and achieve polynomial precision. This is the basis for my second interpolant; some of the degrees of freedom in the patch will be set using the data at the triangle corners, and the remaining degrees of freedom will be set using data at other vertices in the triangulation. The advantages of this method are (a) it requires less data overall and (b) it does not need the cross-boundary derivative data. The disadvantage is a loss of locality in the construction.

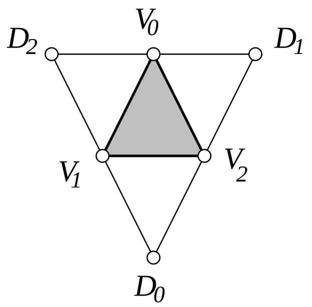

Fig. 6. Interpolant to be constructed over shaded triangle..

The construction is slightly different for even and odd degree. I give here the construction for odd degree $n = 2d+1$, for which I will construct a degree n patch over the triangle $V_0V_1V_2$ and I require the position and d complete derivatives at each of the six vertices shown in Figure 6.

The construction works as follows:

1) For P_{ijk} where $i \geq d$, set P_{ijk} to match the relevant derivative data at V_0 (similarly, we will match the derivative data at V_1 and V_2 when $j \geq d$ and $k \geq d$).
2) Construct three degree n patches. Each patch will use the settings from the previous step and set the remaining control points so that the patch interpolates the data at one of the D_i (only position and $d-1$ derivatives are needed at D_i). This step requires solving a linear system involving the unknown control points.
3) Average the interior control points from the three patches as described below.

The goal of the averaging scheme is to blend in a symmetric manner the vertices for which I have computed multiple values while maintaining polynomial precision. In my construction, I have a set of vertices corresponding to the shaded region of Figure 7 for which I have computed multiple values. The control points are divided into three regions; within a region, the values computed for one of the D_i's is used. On the boundaries between regions, an average of values is used. I.e., for $i < n/3$ (the vertices in the shaded regions), use those values computed for the closest D_0. For $i > j = k$ (the vertices along the boundary between two regions), average the two values computed for D_1 and D_2 (and not use the values computed for D_0). For $i = j = k = n/3$ (the center point), average all three values computed for the D_i Figure 7 illustrates the scheme I used. The values in the other two regions are computed in a similar fashion.

For degree $n = 2d$, if we have only $d-1$ derivatives at the data points, then do not have enough constraints to set the interior control points. However, if we have d derivatives, then the data at the corners gives two settings for the middle control point along each boundary. For our purposes, it is sufficient to use the latter approach, and take the average of the two values for each of these three boundary vertices, and then continue the construction as outlined for the odd degree. Note that for degrees 2 and 4, the derivatives constraints

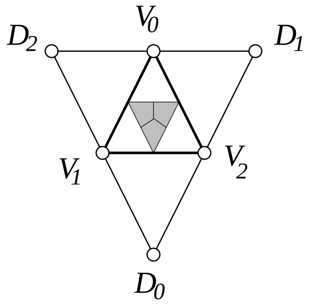

Fig. 7. Averaging scheme for second scheme..

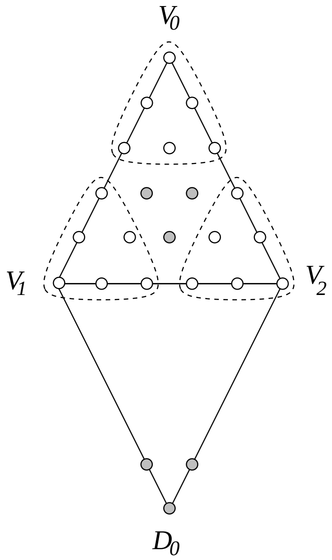

Fig. 8. Construction for second scheme illustrated for $n = 5$..

at the V_i fully specify the control points of the patch. It is not until degree 6 that we have control points left unset by the derivatives at V_i.

This second construction also reproduces degree n polynomials when the data at all six vertices of Figure 6 come from a polynomial, and is also only C^0, since in general two neighboring patches will not meet with C^1 continuity because the center panel of control points will not be coplanar. Again, we can increase the continuity to C^k if our patch is of degree $4k + 1$ using the averaging schemes described in this paper.

Figure 8 illustrates steps one and two of the this scheme for quintics. In this figure, the white circles represent control points set using the data at the corresponding V_i. The remaining shaded points are constructed by fitting a patch to the white points and the position and derivative information at D_0. This process is repeated for D_1 and D_2, giving three values for each of the shaded control points.

Having set the control points of the patch using the corner data results in a C^0 patch network. To get C^1 continuity, we take this patch network, and adjust the shaded points of Figure 5 using the Foley-Optiz scheme.

The most difficult part of this scheme is the interpolation of the neighbor-

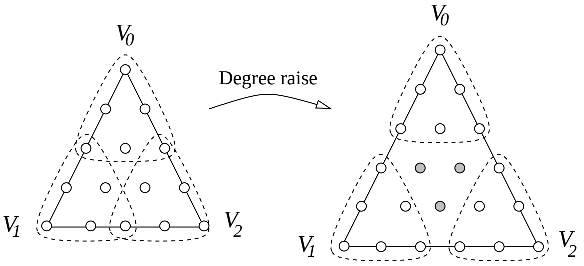

Fig. 9. Construction for new scheme illustrated to achieve C^1 continuity..

ing data. This step requires setting up and solving a system of linear equations. I used LAPACK [2] to solve the system of equations. The difficulty is in finding the coefficients of the control points in the derivative expressions. However, these coefficients are readily obtained by running a reverse de Casteljau construction, similar to the work of Barry and Goldman [3]. See the technical report [12] for details.

§5. Scheme 3

The problem with the first of the schemes described in the previous two sections is that it requires a large number of derivatives at the data points. The second scheme requires fewer derivatives at the data points, but uses data at neighboring triangles (i.e., it is less "local"); in the next section we will see that this use of data at neighboring triangles results in poor surface quality.

We can reduce the number of derivatives from what is required by the first of those schemes without borrowing from the neighboring triangles with the following construction for a C^k patch network:

1) Find the minimum number of derivatives needed to avoid solving the vertex consistency problem (i.e., to get a C^k join, you need degree $4k+1$ patches and $2k$ derivatives and position at the data points).
2) Fit a degree $\lfloor (3k+2)/2 \rfloor$ patch to this data using the scheme described in Section 3.
3) Degree raise this patch to degree $4k+1$.
4) Reset the corner groups of control points using the derivative information at the corners.
5) Adjust the interior control points to obtain C^k continuity.

Figure 9 illustrates this scheme for constructing C^1 quintics. On the left, we first construct a quartic patch using the scheme described in Section 3. Next, we degree raise this quartic patch to quintic, giving the patch on the right. We then reset the circled control points to match the derivatives at the corners, and finally we adjust the gray points to meet the neighboring patches with C^1 continuity.

An obvious variation on this scheme is to start with a degree $4k+1$ patch, set the corner control points using the derivatives, then set the remaining interior control points by assuming that patch is a degree raised patch (similar to Farin's construction for the quadratic precision point of a cubic [5]) and then adjusting for continuity.

§6. Results

I ran several tests on the three schemes, which are detailed in [12]; here I will summarize the results. As a first test, I verified the polynomial reproduction powers. The results of these tests matched the theory: all the schemes had the level of polynomial precision and continuity that was expected.

As a second test, I empirically checked the polynomial convergence. As a first test, I ran my methods on the Franke function, However, as I was only using doubles, the sampling level required to see the expected convergence rate was dense enough that I ran out of floating point precision just as the theoretical convergence was being achieved.

As a better test of convergence, I sampled the function

$$z(x, y) = \sin(\pi x/2 + \pi/4)\sin(\pi y/2 + \pi/4)$$

over the $[0, 1]$ square. This function is between 0.5 and 1 in this region, and since it is a single peak we should be able to observe the asymptotic behavior earlier than with the Franke function. Further, since the values of this function over this region are close to 1, the absolute error is approximately the relative error, and we can readily tell when we have run out of floating point precision.

The convergence rates were almost exactly as predicted, with a few expectations. First, the degree 9 interpolants slightly worse than expected. This is only notable because the lower degree interpolants had the theoretical convergence rate to 2 decimals of precision (the degree 9 interpolants converged about 20% slower than expected).

The other surprise is scheme 3, which performed far better than expected. It is unclear from this single data set if the scheme was converging more rapidly because it had not reached its asymptotic behavior, or if it actually has better convergence properties than might otherwise be expected.

I also generated isophote plots, shaded images, and curvature plots of the surfaces fit to samplings of the Franke function. Figure 10 shows degree five surfaces fit to an 8×8 sampling of the Franke function (no patches were fit to the boundary triangles). Scheme 3, on the right of the figure, clearly constructed a better surface than scheme 2 (in the middle), but it is not as good of quality as the surface constructed by scheme 1 (on the left). While it is not surprising that scheme 1 did a better job than scheme 3 (since scheme 1 uses more derivatives than scheme 3 and has higher polynomial precision than scheme 3), it is a bit surprising that scheme 3 did better than scheme 2, since scheme 2 reproduces fifth degree polynomials while scheme 3 only reproduces fourth degree polynomials.

§16. Conclusions

The purpose of this work was to investigate several data fitting schemes devised to test averaging as a means of achieving continuity between two patches

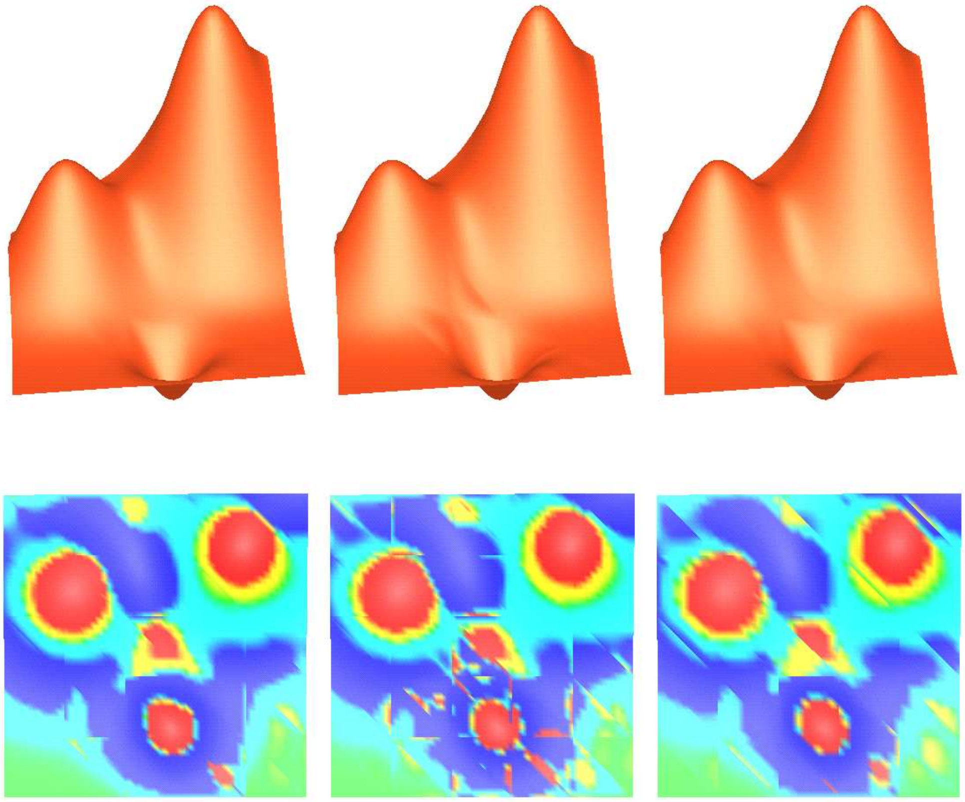

Fig. 10. Caption.

without losing polynomial precision. The continuity construction itself worked as predicted: the desired level of continuity was achieved without losing polynomial precision. Further, the continuity scheme was easy to implement.

While the schemes mostly behaved as expected, scheme 2 performed worse than expected, and appears to perform worse than the scheme 3. This poor behaviour is likely due to poor stability properties of the method and to the non-locality of the data used.

As a final note, while the averaging construction for obtaining continuity has only been applied here for triangular Bézier patches, the idea is easily extended to curves, and in turn, to tensor product Bézier surfaces. In the latter case, we would want (as we did for triangles) enough derivatives at the data points to consistently set enough mixed partial derivatives for the continuity adjustments we are applying to the patches.

References

1. Alfeld, F., L., Schumaker, Smooth finite elements based on Clough-Tocher triangle splits, in preparation.
2. Anderson, E., Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen,

LAPACK Users' Guide, Society for Industrial and Applied Mathematics, Philadelphia, 1999.

3. Barry, P. J. and R. N. Goldman, Algorithms for progressive curves: Extending B-spline and blossoming techniques to the monomial, power, and Newton dual bases, in *Knot Insertion and Deletion Algorithms for B-spline Modeling,* R. N. Goldman and T. Lyche (eds), SIAM, Philadelphia, 1993, 11–63.
4. Clough, R. and J. Tocher, Finite element stiffness matrices for analysis of plates in bending, in *Proc. Conf. Matrix Methods in Structural Analysis,* xxx (ed), Wright-Patterson Air Force Base, Dayton OH, 1965, xxx–xxx.
5. Farin, G., Smooth interpolation to scattered 3D data, in *Surfaces in Computer Aided Geometric Design,* R. E. Barnhill and W. Boehm (eds), North Holland, Amsterdam, 1983, 43–64.
6. Farin, G., *Curves and Surfaces for Computer Aided Geometric Design,* Academic Press, NY, 1997.
7. Foley, T. A. and K. Opitz, Hybrid Cubic Bezier Triangle Patches, in *Mathematical Methods in Computer Aided Geometric Design II,* T. Lyche and L. Schumaker (eds), Academic Press, New York, 1992, 275–286.
8. Franke, Richard, Scattered data interpolation: tests of some methods, Math. Comp. **38(157)** (1982), 181–200.
9. Franke, R. and G. M. Nielson, Scattered data interpolation and applications: A tutorial and survey, in *Geometric Modeling,* H. Hagen and D. Roller (eds), Springer Verlag, Berlin, 1991, 131–160.
10. Mann, S., Cubic precision Clough-Tocher interpolation, Comput. Aided Geom. Design **16(2)** (1999), 85–88.
11. Mann, S., Triangular Interpolants to Scattered Data with High Degree Polynomial Precision, University of Waterloo Report CS-2000-01, 2000.
12. Mann, S., Implementation of Some Triangular Data Fitting Schemes Using Averaging To Get Continuity, University of Waterloo Report CS-2000-10, 2000.
13. Mann, S., Adjusting Control Points to Achieve Continuity, submitted for publication.
14. Powell, M. J. D. and M. A. Sabin, Piecewise quadratic approximations on triangles, ACM Trans. Math. Software **3** (1977), 316–325.
15. Ženišek, A., Interpolation polynomials on the triangle, Numer. Math. **15** (1970), 283–296.

Stephen Mann
Computer Science Department
University of Waterloo
Waterloo, Ontario, CANADA N2L 3G1
`smann@cgl.uwaterloo.ca`