Using Local Optimization in Surface Fitting

To appear in Mathematical Methods for Curves and Surfaces

Stephen Mann

Abstract. Local optimization is used to set the free parameters in a triangular surface fitting scheme, resulting in surfaces with better shape. While some of the free parameters can be set to match curvature information, other free parameters are independent of this information.

§1. Introduction

A large number of local parametric triangular surface schemes have been developed over the past fifteen years (see [8,9] for a survey of such schemes). These schemes are local in that changes to part of the data only affect portions of the surface near the changed data. Surprisingly, all of these schemes exhibit similar shape defects. On closer inspection, it is seen that these schemes all have a large number of free parameters that are set using simple heuristics. By manually adjusting these parameters, one can improve the shape of the surfaces [9].

In this paper, I will investigate using local optimization to set the free parameters in a local, triangular, split-domain, polynomial surface interpolation scheme. More precisely, given a triangle of data, this scheme constructs three triangular parametric polynomial patches, each of which interpolates two of the positions and normals at the three vertices. Further, the patches meet each other with G^1 continuity, and if used to fill a triangular polyhedron, the resulting surface will also be G^1. Local optimization is used to set the remaining free parameters so as to minimize the error in interpolating second order data at the corners. Thus, our data is a set of three vertices with normals and second fundamental forms, $(p_i, \hat{n}_i, \mathrm{II}_i)$, for $i = P, Q, R$.

§2. The Modified Shirman-Séquin Scheme

I will be working with a modified version of Shirman-Séquin's scheme [11,12]. Shirman and Séquin created a split domain scheme that fits three quartic patches to a triangle of data. These triangles meet each other G^1, and if used

Mathematical Methods for Curves and Surfaces
M. Dæhlen, T. Lyche, and L. L. Schumaker (eds.), pp. 1–3.

ISBN xxx.

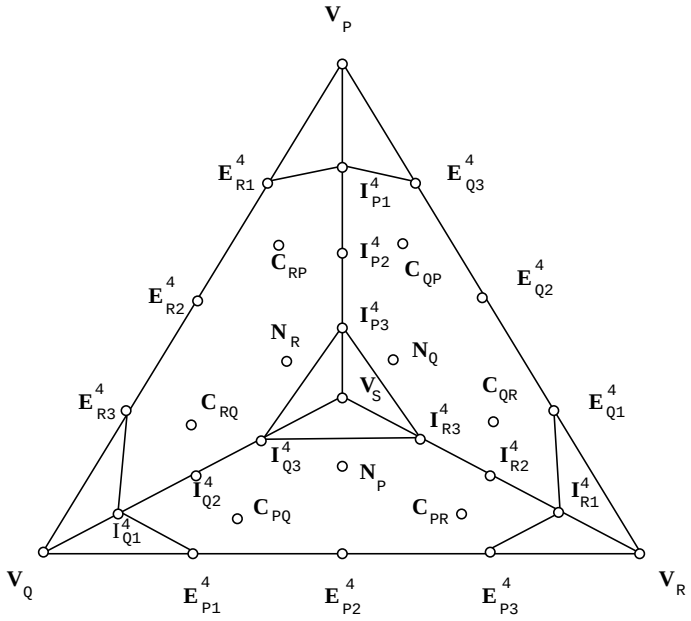

Fig. 1. Controls points in a split domain scheme.

to fill a triangular polyhedron, the entire surface will be G^1. Note that while the patches are quartic, the boundaries of the patches are only degree three.

The Bézier control points [3] for the three patches are shown in Figure 1. We will have use for both the cubic and quartic control points of the boundaries; thus, we shall denote the cubic control points with a superscript of 3 and the quartic with a superscript of 4. Using a modification of the Shirman-Séquin scheme, for $ijk \in \{PQR, QRP, RPQ\}$, we set these control points in the following manner:

$$V_i = p_i$$

$$E^3_{i1} = p_i + \alpha_{i1}\overrightarrow{t_{i1}}$$

$$E^3_{i2} = p_j - \alpha_{i2}\overrightarrow{t_{i2}}$$

$$I^3_{i1} = (1-\beta_i)p_i + \beta_i(E^4_{j2} + E^4_{k1})/2$$

$$\begin{aligned} C_{ij} = \frac{1}{3}\big\{ & c_i(k_{ij}+k_{ik})\widehat{C_{ij}} + k_{ij}\widehat{C_{ik}} \\ & + 2h_{ij}(E^3_{i2} - E^3_{i1}) + h_{ik}(E^3_{i1} - V_j)\big\} + E^3_{i1} \end{aligned} \tag{1}$$

$$\begin{aligned} C_{ik} = \frac{1}{3}\big\{ & k_{ik}\widehat{C_{ij}} + c_i(k_{ij}+k_{ik})\widehat{C_{ik}} \\ & + h_{ij}(V_k - E^3_{i2}) + 2h_{ik}(E^3_{i2} - E^3_{i1})\big\} + E^3_{i2} \end{aligned} \tag{2}$$

$$I^3_{i2} = \frac{3\beta_i}{16}V_i + \left(1 - \frac{27\beta_i}{16}\right)I^3_{i1} + \frac{12\beta_i}{16}(C_{ji} + C_{ki})$$

$$N_i = \frac{1}{4}I^3_{i1} - \frac{1}{4}(I^3_{j1} + I^3_{k1}) + \left(\frac{4}{9\beta_i} - \frac{5}{4}\right)I^3_{i2} + \left(\frac{-2}{9\beta_i} + \frac{5}{4}\right)(I^3_{k2} + I^3_{j2})$$

$$V_S = (I^3_{P2} + I^3_{Q2} + I^3_{R2})/3$$

Here, the $\overrightarrow{t_{in}}$ are vectors perpendicular to the normal $\hat{n}_i$. The α and β are free parameters; any positive setting of these free parameters will give us a G^1 surface scheme for most choices of the $\overrightarrow{t_{in}}$. The c_i are shape parameters used by Jensen [4] in constructing the crossboundary functions (Shirman-Séquin used Chiyokura-Kimura's crossboundary function [1], which is the special case of Jensen's function when $c_i = 1$).

The $\widehat{C}$s, ks, and hs are formed by Chiyokura-Kimura's method: $\widehat{C_{ij}}$ is perpendicular to both the normal $\hat{n}_j$ and to the vector $E^3_{i1} - V_j$ with the sign of $\widehat{C_{ij}}$ chosen to make k_{ij} positive. Similarly, $\widehat{C_{ik}}$ is perpendicular to the normal $\hat{n}_k$ and $E^3_{i2} - V_k$. The ks and hs are given by:

$$I^4_{i1} - V_j = k_{ij} \cdot \widehat{C_{ij}} + h_{ij} \cdot (E^3_{i1} - V_j) \tag{3}$$

$$I^4_{k1} - V_k = k_{ik} \cdot \widehat{C_{ik}} + h_{ik} \cdot (V_k - E^3_{i2}) \tag{4}$$

The above ks and hs actually depend linearly on the β_is. Since we want to optimize over the β_i, we will express the above equations in terms of the β_i by computing the ks and hs when the β_is are 1.0. Call these $\widehat{k_{ij}}$, $\widehat{h_{ij}}$, $\widehat{k_{ik}}$, and $\widehat{h_{ik}}$. The above two equations then become

$$I^4_{i1} - V_j = \beta_j \widehat{k_{ij}} \cdot \widehat{C_{ij}} + \beta_j \widehat{h_{ij}} \cdot (E^3_{i1} - V_j)$$

$$I^4_{k1} - V_k = \beta_k \widehat{k_{ik}} \cdot \widehat{C_{ik}} + \beta_k \widehat{h_{ik}} \cdot (V_k - E^3_{i2})$$

Many of the degrees of freedom occur in the boundaries of the three patches. In the discussion that follows, *exterior boundaries* refer to the boundaries given by the V_i and E^3_{in}. *Interior boundaries* refer to the boundaries given by the V_i, I^3_{in} and V_s, for $i \in \{P, Q, R\}$ $n \in \{1, 2, 3\}$.

Counting, we find a total of twelve shape parameters in the above equations. There are nine more shape parameters in the Shirman-Séquin scheme: we can vary the $\overrightarrow{t_{in}}$, and the I_{i1} could be allowed to vary in the tangent plane. We will set the $\overrightarrow{t_{in}}$ by placing each boundary curve in a plane. The choice of planar boundaries uses three degrees of freedom, while the particular planes chosen use another three. I chose the plane passing through the two data points and the average of the normals at the data points, which sets all six degrees of freedom. The I_{i1}s were restricted to a linear degree of freedom for reasons of symmetry.

§3. Cubic Curve Optimization

In Figure 5, we see five surfaces. The surface on the left is an S-patch surface [6]. The second surface on the left is the same S-patch surface subdivided into triangular pieces. The corners of these pieces have been sampled for position, normal, and second fundamental form. The surface in the center is a

Shirman-Séquin surface fit to this data. Our goal is to improve the quality of this center surface. We will begin by improving the exterior boundary curves.

The left column of Figure 2 shows two of the exterior boundary curves constructed for this ring data set by the standard Shirman-Séquin scheme (these are the Bézier curves described by the control points V_i and E_{in}). Below each curve is a curvature plot of the curve (the curved line is the curvature plot; the straight line connects the curvature of the S-patch surface at the endpoints of the curve). While the curvature of the top curve is distributed relatively uniformly over the curve, the curvature of the bottom curve is seen to concentrate at the endpoints of the curve, leaving a flat region in the middle.

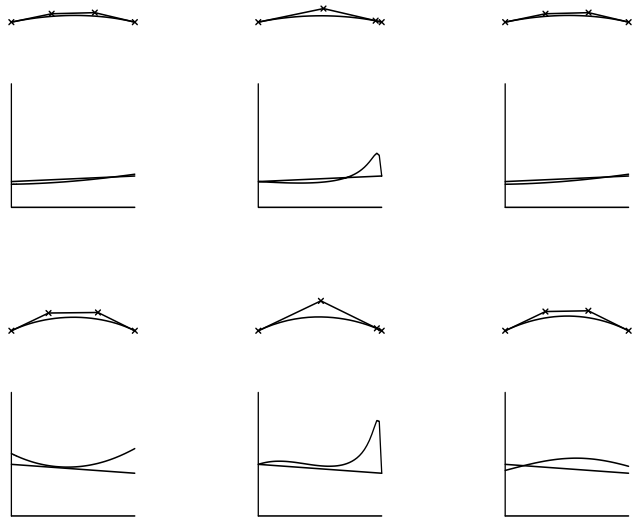

Fig. 2. Boundary curves and their curvature plots.

As discussed in [8], we can expect improved surface shape if we match second order data with the boundary curves. Thus, we would like a solution to the problem of interpolating the position, tangent direction, and curvature at two points with a cubic curve. Klass solved this problem numerically [5], while de Boor et. al. found an analytic solution to the problem [2]. Note that a solution may not exist, and when it does exist, it may not be unique.

I implemented the de Boor-Hollig-Sabin technique and integrated it into Shirman-Séquin's scheme. While some areas of the surface improve, there are regions where "lumps" appear in the surfaces as illustrated in the fourth surface of Figure 5. Looking at the boundary curves again reveals the cause of the shape defect.

The center column of Figure 2 shows the two curves constructed by the deBoor-Hollig-Sabin method for the same data used in the first column. As can be seen in the figure, both curves have a short first derivative at the right end point and a spike in curvature near that end of the curve. This short derivative causes a problem for the surface construction scheme because short endpoint first derivatives lead to large values of the h_{ij} (Equations 3 and 4). These scalars are then used to weight vector quantities in the construction of the C_{ij} (Equations 1 and 2).

Thus, the result of short derivatives at the end of the exterior boundaries is that the C_{ij} and any interior control points dependent on the C_{ij} may be positioned far from the triangle of data, as illustrated in Figure 3. In this figure, there are two views of two patches constructed for the same three data points. Two of the exterior boundary curves are identical in both patches. However, the data along the third boundary has two cubic curves that interpolate the specified position, tangent directions, and curvatures. In the patch on the left, we see that both tangents at the ends of this boundary curve are of reasonable length, and that the interior control points are uniformly distributed. However, in the patch on the right, we see that one of the tangents at the ends of this curve is short. The result is that the interior control points have a more chaotic placement, leading to the lumps seen in Figure 5.

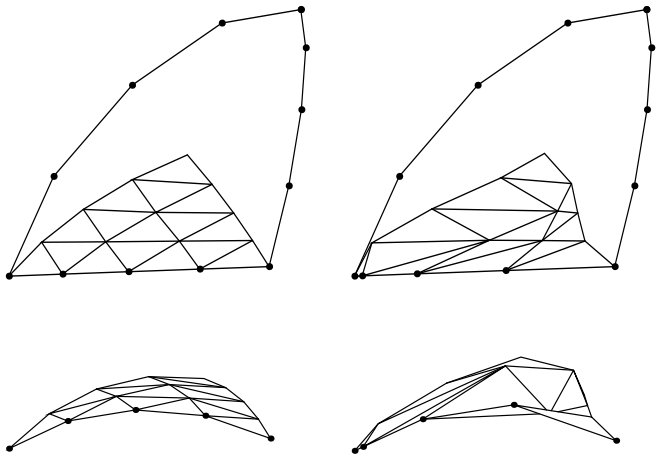

Fig. 3. Effect of short end tangents.

Thus, it is insufficient to construct exterior boundaries that match curvature at their endpoints. We must ensure that the derivatives at the ends of the curves are not too short. Therefore, we will solve a variation of the above curve construction problem. We will find a cubic curve that interpolates the position and tangent direction, and trades off interpolating the curvature with interpolating prescribed first derivatives at the endpoints.

We want our construction to be scale independent. To achieve this goal, we will always scale our problem to a "unit" problem by mapping the endpoints to be separated by unit distance. The first derivatives will scale "down" linearly, while the curvatures scale "up" linearly. So, if we wish to find f such that f interpolates our data at its endpoints as follows,

$$f(0) = P_0\,, \quad f'(0) = \overrightarrow{t_0}\,, \quad k(0) = k_0$$

$$f(1) = P_1\,, \quad f'(1) = \overrightarrow{t_1}\,, \quad k(1) = k_1$$

then we will map the data to

$$\bar{P_0} = P_0\,, \quad \bar{\overrightarrow{t_0}} = \overrightarrow{t_0}/|P_0 - P_1|, \quad \bar{k_0} = k_0 \cdot |P_0 - P_1|$$

$$\bar{P_1} = P_0 + (P_1 - P_0)/|P_1 - P_0|, \quad \bar{\overrightarrow{t_1}} = \overrightarrow{t_1}/|P_0 - P_1|, \quad \bar{k_1} = k_1 \cdot |P_0 - P_1|.$$

After constructing $\bar{f}$, we map the control points of $\bar{f}$ through the inverse transformation, giving the control points of f. For notational simplicity, I will use the unbarred symbols to denote the scaled data in the remainder of this section.

Our goal is to satisfy the following set of equations:

$$|f'(0) - \overrightarrow{t_0}| = 0, \qquad |f'(1) - \overrightarrow{t_1}| = 0, \tag{5}$$

$$\frac{f'(0) \times f''(0)}{|f'(0)|^3} - k(0) = 0, \qquad \frac{f'(1) \times f''(1)}{|f'(1)|^3} - k(1) = 0. \tag{6}$$

In general we cannot expect to satisfy all four equations simultaneously. Thus, we will rephrase this as a minimization problem. First, we square all four equations to remove the derivative discontinuity. Then, we will rewrite these equations in terms of the control points. From Figure 4, we see that the derivatives f at the end points are:

$$\begin{aligned}
f(0) &= P_0 \\
f'(0) &= 3(P_1 - P_0) = \alpha_0 \overrightarrow{t_0} \\
f''(0) &= 6(P_2 - 2P_1 + P_0) \\
&= 6\left[(P_3 - P_0) - \alpha_1 \overrightarrow{t_1}/3 - 2\alpha_0 \overrightarrow{t_0}/3\right] \\
f(1) &= P_3 \\
f'(1) &= 3(P_3 - P_2) = \alpha_1 \overrightarrow{t_1} \\
f''(1) &= 6\left[(P_0 - P_3) + \alpha_0 \overrightarrow{t_0}/3 + 2\alpha_2 \overrightarrow{t_1}/3\right]
\end{aligned}$$

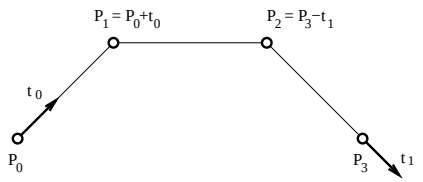

Fig. 4. Control points of derivatives.

We can rewrite Equations 5 in terms of the length of the first derivative, i.e., setting $\overrightarrow{t_{i1}}$ and $\overrightarrow{t_{i2}}$ to be the desired length. We set our goal first derivative to be of unit length for simplicity:

$$(\alpha_0 - 1.0)^2, \qquad (\alpha_1 - 1.0)^2. \tag{7}$$

For the curvature equations (Equations 6), we square them to make them strictly non-negative:

$$\frac{(f'(0) \times f''(0))^2}{|f'(0)|^6} - \frac{2k_0(f'(0) \times f''(0))}{|f'(0)|^3} + k_0^2 \tag{8}$$

$$\frac{(f'(1)\times f''(1))^2}{|f'(1)|^6} - \frac{2k_1(f'(1)\times f''(1))}{|f'(1)|^3} + k_1^2 \tag{9}$$

I used an optimizer that needs the derivatives of the equations we optimize [7]. The two partials of Equations 7 are given by

$$\frac{\partial(\alpha_i - 1)^2}{\partial\alpha_j} = [2\alpha_i - 2]\delta_{ij},$$

where δ is the Kronecker delta function.

We will be optimizing Equations 7, 8, and 9 over α_0 and α_1. To expand these equations in terms of the αs, we will rewrite their terms with respect to the αs:

$$\begin{aligned}
f'(0)\times f''(0) &= \alpha_0\overrightarrow{t_0}\times 6[(P_3 - P_0) - \alpha_1\overrightarrow{t_1}/3 - 2\alpha_0\overrightarrow{t_0}/3] \\
&= 2\alpha_0[3\overrightarrow{t_0}\times(P_3 - P_0) - \alpha_1\overrightarrow{t_0}\times\overrightarrow{t_1}] \\
\frac{\partial}{\partial\alpha_0}(f'(0)\times f''(0)) &= 2[3\overrightarrow{t_0}\times(P_3 - P_0) - \alpha_1\overrightarrow{t_0}\times\overrightarrow{t_1}] \\
\frac{\partial}{\partial\alpha_1}(f'(0)\times f''(0)) &= -2\alpha_0\overrightarrow{t_0}\times\overrightarrow{t_1} \\
f'(1)\times f''(1) &= 2\alpha_1[3\overrightarrow{t_1}\times(P_0 - P_3) + \alpha_0\overrightarrow{t_1}\times\overrightarrow{t_0}] \\
\frac{\partial}{\partial\alpha_0}(f'(1)\times f''(1)) &= 2\alpha_1\overrightarrow{t_1}\times\overrightarrow{t_0} \\
\frac{\partial}{\partial\alpha_1}(f'(1)\times f''(1)) &= 2[3\overrightarrow{t_1}\times(P_0 - P_3) + \alpha_0\overrightarrow{t_1}\times\overrightarrow{t_0}]
\end{aligned}$$

Now we will take the partials of Equations 8 and 9, and expand in terms of our control points and free parameters. I will just state the result here; a derivation may be found in [10]:

$$\begin{aligned}
&\frac{\partial}{\partial\alpha_0}\left(\frac{(f'(0)\times f''(0))^2}{|f'(0)|^6} - \frac{2k_0(f'(0)\times f''(0))}{|f'(0)|^3} + k_0^2\right) = \\
&\qquad\frac{-16[3\overrightarrow{t_0}\times(P_3 - P_0) - \alpha_1\overrightarrow{t_0}\times\overrightarrow{t_1}]^2}{\alpha_0^5} \\
&\qquad + \frac{8k_0[3\overrightarrow{t_0}\times(P_3 - P_0) - \alpha_1\overrightarrow{t_0}\times\overrightarrow{t_1}]}{\alpha_0^3}
\end{aligned} \tag{A}$$

$$\begin{aligned}
&\frac{\partial}{\partial\alpha_1}\left(\frac{(f'(0)\times f''(0))^2}{\alpha_0^6} - \frac{2k_0(f'(0)\times f''(0))}{\alpha_0^3} + k_0^2\right) = \\
&\quad\frac{-8[3\overrightarrow{t_0}\times(P_3 - P_0) - \alpha_1\overrightarrow{t_0}\times\overrightarrow{t_1}][\overrightarrow{t_0}\times\overrightarrow{t_1}]}{\alpha_0^4} + \frac{4k_0[\overrightarrow{t_0}\times\overrightarrow{t_1}]}{\alpha_0^2}
\end{aligned} \tag{A}$$

$$\begin{aligned}
&\frac{\partial}{\partial\alpha_0}\left(\frac{(f'(1)\times f''(1))^2}{\alpha_1^6} - \frac{2k_1(f'(1)\times f''(1))}{\alpha_1^3} + k_1^2\right) = \\
&\quad\frac{-8[3\overrightarrow{t_1}\times(P_0 - P_3) - \alpha_0\overrightarrow{t_0}\times\overrightarrow{t_1}][\overrightarrow{t_0}\times\overrightarrow{t_1}]}{\alpha_1^4} + \frac{4k_1[\overrightarrow{t_0}\times\overrightarrow{t_1}]}{\alpha_1^2}
\end{aligned} \tag{A}$$

$$\frac{\partial}{\partial\alpha_1}\left(\frac{(f'(1)\times f''(1))^2}{\alpha_1^6}-\frac{2k_1(f'(1)\times f''(1))}{\alpha_1^3}+k_1^2\right)=$$
$$\frac{-16[3\overrightarrow{t_1}\times(P_0-P_3)-\alpha_0\overrightarrow{t_0}\times\overrightarrow{t_1}]^2}{\alpha_1^5} \qquad (A)$$
$$+\frac{8k_1[3\overrightarrow{t_1}\times(P_0-P_3)-\alpha_0\overrightarrow{t_0}\times\overrightarrow{t_1}]}{\alpha_1^3}$$

To summarize, I used the optimizer of Mahdavi-Amiri and Bartels [7] to build cubic curves by optimizing Equations 7, 8, and 9 using the first derivatives of these equations with respect to α_0 and α_1 (Equations A) to drive the optimizer. Examples of the resulting curves can be seen in the right hand column of Figure 2. While these curves do not match the end point curvatures exactly, their end point curvatures are close to the desired curvatures, and none of the derivatives at the ends of the curves is short. Further, the curvature distribution throughout the curves is more uniform.

I integrated this curve construction into the surface construction technique as follows: For each $ijk \in \{PQR, QRP, RPQ\}$,

1. Pick a plane through V_j and V_k.
2. Construct the boundary $V_j, E^3_{i1}, E^3_{i2}, V_k$ using the curve construction technique detailed in this section.

Once we have constructed all the E_{ij}, we construct the interior control points using the remaining formulas given in Section 2.

The right most surface in Figure 5 show the surface constructed by using these optimized curves in the Shirman-Séquin scheme. Several of the shape defects have disappeared, and no lumps have appeared. However, while this is a significant improvement, there are still several "wrinkles" visible in the surface. To attempt to remove these shape defects, we will look at adjusting other shape parameters in the next section.

§4. Internal Degrees of Freedom

Internally, there are several degrees of freedom. There are degrees of freedom in constructing the cross boundary derivatives and in constructing the interior boundary curves. We would like to vary the degrees of freedom influencing the cross boundary derivatives to improve the shape of the surface patch. One choice is to use the degrees of freedom to improve the order of approximation. Another idea is to vary these degrees of freedom and try to further match the second fundamental forms at the data points. As discussed in this section, neither of these approaches works in this setting.

Using the Shirman-Séquin settings of the αs, βs, and cs, the construction discussed in Section 2 has linear precision. If we construct boundary curves with quadratic precision, we could hope to achieve quadratic precision. However, using the Shirman-Séquin setting of the βs and cs does not have quadratic precision. Further, even if we use Jensen's generalization of the crossboundaries, we are unable to achieve quadratic precision [10].

A second approach to setting the internal degrees of freedom would be to match more data contained in the second fundamental forms at the data points. In the previous section, we matched curvature at the ends of the exterior boundary curves. There are two more ways in which we can match the second fundamental forms at our data points: we can match curvature at the ends of the interior boundaries, and we can match mixed partial derivatives at the patch corners.

There are three curvature equations and six mixed partial equations. Ideally, we would expand these nine equations in terms of our βs and cs, re-express these equations as minimization problems, and vary the βs and cs to minimize a weighted sum of the equations. We might also want additional equations to prevent the βs and cs from becoming too small or too large. However, the last is a moot point: as detailed in [10], if we expand our curvature equations in terms of the control points and unknowns (the βs and the cs), then we find the equations are independent of all the βs and the cs. Thus, we are unable to vary the βs and cs to match curvature on the interior of our patches.

§5. Conclusions and Future Work

We have seen that improved settings of degrees of freedom in a G^1 surface fitting scheme yields a surface of better quality. Thus far, we have only found improved settings for degrees of freedom on the boundaries of the surface patches. Improved settings for the remaining degrees of freedom should be found (the β_i and the c_i of the previous section).

The degrees of freedom along the boundary were set by matching second fundamental forms at the data points. We have shown that the internal degrees of freedoms are independent of such data. Thus, an alternative way of setting them needs to be found. Further, we would prefer to construct surfaces for a triangulated set of points without normals and second fundamental forms specified at these points. Techniques that do not require such data also deserve investigation.

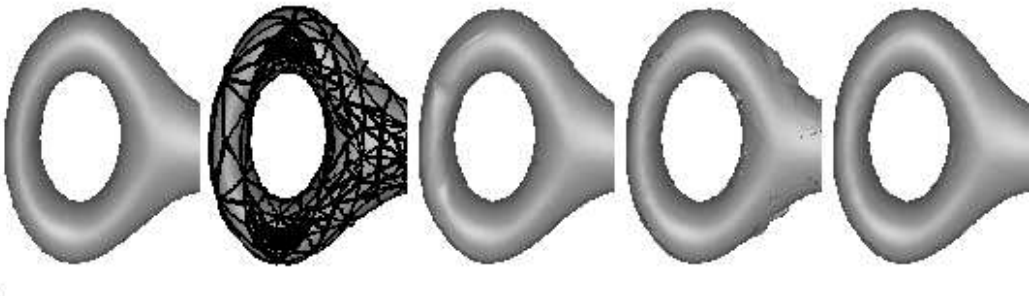

Fig. 5. Ring surface.

References

1. H. Chiyokura and F. Kimura, Design of solids with free-form surfaces, Computer Graphics, **17**(3), (1983), 289–298.
2. C. de Boor, K. Höllig, and M. Sabin, High accuracy geometric Hermite interpolation, Computer Aided Geometric Design, **4**(4), (1987), 269–278.
3. G. Farin, *Curves and Surfaces for CAGD*, Academic Press, 1993.
4. T. Jensen, Assembling triangular and rectangular patches and multivariate splines, in *Geometric Modeling: Algorithms and New Trends*, G. Farin (ed), SIAM, 1987, 203–220.
5. R. Klass, An offset spline approximation for plane cubic splines, Computer Aided Design, **15**(5), (1983), 297–299.
6. C. Loop and T. DeRose, A Multisided Generalization of Bézier Surfaces, ACM Transactions on Graphics, **8**(3), (1989), 204–234.
7. N. Mahdavi-Amiri and R. H. Bartels, Constrained Nonlinear Least Squares: An Exact Penalty Approach with Projected Structured Quasi-Newton Updates, ACM Transactions on Mathematical Software, **15**(3), (1989), 220–242.
8. S. Mann, C. Loop, M. Lounsbery, D. Meyers, J. Painter, T. DeRose, K. Sloan, A Survey of Parametric Scattered Data Fitting Using Triangular Interpolants, in *Curve and Surface Design*, H. Hagen (ed), SIAM, 1992, 145–172.
9. S. Mann, Surface approximation using geometric hermite patches, dissertation, University of Washington, 1992.
10. S. Mann, Using local optimization to improve surface fitting, Research Report CS-94-33, Computer Science Department, University of Waterloo, Waterloo, Ontario (August 1994).
11. L. A. Shirman and C. H. Séquin, Local surface interpolation with Bézier patches, Computer Aided Geometric Design, **4**(4), (1987), 279–295.
12. L. A. Shirman and C. H. Séquin, Local surface interpolation with Bézier patches: errata and improvements, Computer Aided Geometric Design, **8**(3), (1991), 217–221.

Acknowledgements. This work was funded in part by a grant from the National Science and Engineering Research Council of Canada

Stephen Mann
Computer Science Department
University Of Waterloo
200 University Ave. W.
Waterloo, Ontario CANADA N2L 3G1