Understanding Errors in Approximating Principal Direction Vectors

By Jack Goldfeather and Victoria Interrante

Introduction

Suppose we are given only a surface mesh of vertices and polygons approximating some unknown smooth surface. There have been many methods proposed for approximating principal directions of the underlying surface [1,2,4,8,12,13]. In this paper we will examine a few of the known methods, showing how well they can work in some cases and how miserably they can fail in others. In particular we will show how very tiny normal curvature approximation errors can be magnified into large errors in the estimated principal directions. We also introduce a new method that we believe performs significantly better under certain conditions than many other proposed methods. In section I, we briefly describe the motivation for this work and its significance to applications in computer graphics. In section II, we outline the basic mathematics behind computing principal directions, stating the necessary formulas for the Weingarten curvature matrix and for its use in computing normal curvature in a given direction. In section III we describe in detail three methods each of which approximates the Weingarten curvature matrix at a vertex of the mesh. In section IV we apply each method to a test surface using a number of different mesh schemes and then examine the direction errors. In section V we develop some mathematical relationships between surface approximation errors and errors in principal directions and illustrate how similar-appearing approximation errors can lead to vastly different principal direction errors. In particular, we illustrate that while the chance for large errors increases near umbilical points, they can still occur at points on the surface where there is significant difference between the two principal curvatures. We conclude by summarizing the conditions under which principal direction estimation errors are most likely to occur, suggesting some steps that can be taken to improve the situation, and outlining promising directions for future work in this area.
I. Motivation

There are many applications for which it is useful to be able to calculate accurate estimates of the principal directions at points on a given surface. Principal direction vector fields have been widely used in shape analysis and surface interrogation in computer-aided manufacturing and design [7,11]. In these applications the surfaces are analytically defined, so the principal directions can be solved for directly and there are few problems with errors in the estimations. Principal direction vector fields have also been successfully used in conjunction with volumetrically-defined data to enhance the visual representation of surface shape for applications in molecular [3] and medical [10] visualization. In these applications the principal directions can be obtained by diagonalizing the Second Fundamental Form, whose entries can be fairly well approximated using first and second directional derivatives of the sampled 3D distribution.

Recently, there has been increasing interest expressed in the possibility of using principal direction vector fields over polygonal meshes for such purposes as guiding the direction of hatching strokes in pen-and-ink style renderings [5,9] or adaptively guiding the orientation of synthesized anisotropic texture patterns for enhanced surface shape representation [6]. However such efforts have been complicated by the lack of a robust and reliably accurate method for estimating the principal directions at points on an underlying smooth surface that is represented solely by a polygonal mesh. Although a number of methods for principal direction estimation have been previously published [13,8,2,4,1,12], inevitably what we have found in practice is that the computed vector field appears ”noisy”, requiring smoothing that can result in many of the vectors falling out of alignment with the true principal directions and complicating our efforts to generate surface markings that closely follow the form.

In the research described in this paper we sought to gain insight into the sources of errors in principal direction estimation on surfaces defined by point samples organized in a polygonal mesh, in order to determine what the worst potential pitfalls are and how they might be overcome. Our goal was to enable an approach in which instead of accepting large principal direction estimation errors and then working to hide them, we could strive to obtain more accurate initial principal direction estimates while flagging the points at which the probability of obtaining erroneous estimates is high.
II. A Quick Review of Surface Curvature

Let $p$ be a point on a smooth surface $S$, let $N_p$ be the unit normal to $S$ at $p$, and suppose $X(u,v)$ is a local parametrization of $S$ in a neighborhood of $p$. Then using $X_u(p)$, $X_v(p)$, $N_p$ as a local coordinate system, we can compute principal curvatures and principal directions as follows: Let $\lambda_1$ and $\lambda_2$ ($\lambda_1 \geq \lambda_2$) be the eigenvalues, and $p_1$, $p_2$ the associated unit eigenvectors of the Weingarten curvature matrix

$$W = \begin{pmatrix}
e G - f F \\
E G - F^2 \\
f G - g F \\
E G - F^2
\end{pmatrix}$$

where

$$e = N_p \circ X_{uu}(p) \quad E = X_u(p) \circ X_u(p)$$
$$f = N_p \circ X_{uv}(p) \quad F = X_u(p) \circ X_v(p)$$
$$g = N_p \circ X_{vv}(p) \quad G = X_v(p) \circ X_v(p)$$

Note that in the special case that $X_u$ and $X_v$ are orthogonal unit vectors, this becomes the symmetric matrix

$$W = \begin{pmatrix}
e & f \\
f & g
\end{pmatrix}.$$ 

If $u$ is a unit vector in the tangent plane to $S$ at $p$, then

$$\kappa_u = u^T W u$$

is the normal curvature of the surface in the direction of $u$.

It follows that $\lambda_1$ and $\lambda_2$ are the maximum and minimum normal curvatures of the surface at $p$, and $p_1 = \begin{pmatrix} p_{11} \\ p_{12} \end{pmatrix}$ and $p_2 = \begin{pmatrix} p_{21} \\ p_{22} \end{pmatrix}$ are the principal curvature vectors expressed in local coordinates. That is,

$$v_1 = p_{11} X_u + p_{12} X_v$$
$$v_2 = p_{21} X_u + p_{22} X_v$$

are the principal direction vectors in $\mathbb{R}^3$.

An important observation is that at points where $\lambda_1 = \lambda_2$, the notion of principal direction is not defined, since all vectors are eigenvectors. Such a point is called an umbilical point on the surface.
III. Three Principal Direction Approximation Methods

The first step in computing principal directions on a surface mesh is to compute at each vertex $p$ a vector $N_p$ that approximates the true unit surface normal $N_p$ at $p$. Most methods compute a “normalized average” - i.e., a set of vectors is summed and the resulting vector is normalized to length 1. As several different surface normal approximation methods have been proposed, we decided to take a closer look at three:

1. The normalized average of the surrounding triangle unit normals.
2. The normalized average of the surrounding triangle unit normals, each weighted by the angle of the triangle at $p$.
3. The normalized average of the adjacent edges, each weighted by the sum of the cotangents of the angles opposite the edge in the two triangles sharing the edge [2]. This weighting only makes sense if each pair of triangles is acute, so another method must be used at vertices where a triangle is not acute.

We experimented with each of these schemes on a complicated surface, using a variety of local meshes and a large number of random points on the surface. Figure 1 shows our findings.

In all cases, the vertices that we used lay in the analytically-defined surface. Under these conditions, all of the normal approximation methods worked very well. The average error in the approximated normal over all mesh types was 0.90 degrees for the unweighted method, 0.97 degrees for the angle-weighted method, and 1.04 degrees for the edgelength-weighted method. The median error over all mesh types was 0.41 degrees for the unweighted method, 0.50 degrees for the angle-weighted method, and 0.54 degrees for the edgelength weighted method.

Although the normal vector approximation errors can appear from Figure 1 to be superficially slightly smaller for the angle-weighted method in the cases of the global meshes and for the unweighted method in the cases of the local meshes, these differences are not significant either practically or statistically. Of course, using an incorrect surface normal will introduce error into a principal direction calculation. However, as we will show, even if the exact surface normal is used, there are other sources of significant potential error in principal direction estimation. For this reason, we decided to use exact surface normals in the subsequent steps of our investigation.
Figure 1. An experimental comparison of three normal vector approximation methods. Upper left: Median errors, per method and mesh type. Upper right: Error histogram, unweighted averaging method. Lower left: Error histogram, angle-weighted averaging method. Lower right: Error histogram, edge length-weighted averaging method.

The next step in estimating the principal directions is to choose a pair of orthonormal vectors $x_1$ and $x_2$ in the plane tangent to $N'_p$ to form a local orthonormal coordinate system $L = \{x_1, x_2, N'_p\}$ in $R^3$. All subsequent calculations are done with respect to this local coordinate system. We examined three principal direction estimation methods, outlined below, each of which approximates the Weingarten curvature matrix expressed in this local coordinate system.
III.a The Normal Curvature Approximation Method

Suppose $W$ is the unknown Weingarten matrix with respect to local coordinates $L$ at vertex $p$. Suppose there are $n$ vertices adjacent to $p$ and let $q_i$ denote the $i^{th}$ adjacent vertex. We denote by $y_i$ the unit vector obtained by projecting the vector $\overrightarrow{pq_i}$ (expressed in local coordinates $L$) onto the plane tangent to $N_p'$ and normalizing the result. Then using the result from section I, the normal curvature in the direction $y_i$ is given by $\kappa_{y_i} = y_i^T W y_i$. An approximation to this normal curvature is given by:

$$\kappa'_{y_i} = 2 \frac{(p-q_i) \circ N_p'}{(p-q_i) \circ (p-q_i)}$$

which is the curvature of the unique osculating circle passing through $p$ and $q_i$ with normal $N_p'$ at $p$ [2]. This produces a system of equations:

$$y_i^T W y_i = \kappa'_{y_i}, \quad i = 1, 2, \ldots, n$$

that we wish to solve for $W$. In [2], these equations are weighted in the same manner as the weighting in the third normal approximation method mentioned at the beginning of this section, but it is not clear that this improves the result much. The fact is that the osculating circle only produces a second-order approximation to the true normal curvature and second-order approximations can introduce significant error in many cases.

The first step in solving for $W$ is to reorganize (1) as follows. Let $y_i = (u_i, v_i)$ and

$$W = \begin{pmatrix} A & B \\ B & C \end{pmatrix}.$$

Then

$$y_i^T W y_i = (u_i \quad v_i) \begin{pmatrix} A & B \\ B & C \end{pmatrix} \begin{pmatrix} u_i \\ v_i \end{pmatrix} = (u_i^2 \ 2u_i v_i \ v_i^2) \begin{pmatrix} A \\ B \\ C \end{pmatrix}.$$

If we let $U$ be the $n \times 3$ matrix with rows $(u_i^2 \ 2u_i v_i \ v_i^2), x = (A \ B \ C)^T$, and $d$ be the $n$-vector whose $i^{th}$ entry is $\kappa'_{y_i}$ the entire system can be written as the matrix equation

$$U x = d.$$

If $d$ was the vector of true normal curvatures, this linear system would have an exact solution (assuming the adjacent vertices do not have some degenerate pattern, like all lying on the same line.) In practice, the
best we can hope to find is a least squares fit, i.e., a vector \( x \) that minimizes \( \| Ux - d \| \). There are standard numerical methods for finding this least squares solution. The resulting matrix

\[
W' = \begin{pmatrix}
A' & B' \\
B' & C'
\end{pmatrix}
\]

is used to approximate the principal directions. It is worth noting that the matrix \( U \) depends only on the projection of the adjacent vertices onto the local tangent plane, i.e. it is fixed by the choice of the mesh on the surface. The vector \( d \) depends on measurements we make at these adjacent vertices - i.e. it varies according to the amount of error we make. As we will see in a later section, it is possible to separate the mesh from the measurements so that we can understand how measurement error interacts with the local mesh geometry.

### III.b The Quadratic Surface Approximation Method

In this method we try to best-fit a quadratic surface to the adjacent vertices. We begin by transforming each adjacent vertex \( q_i \) to local coordinates \((x_i, y_i, z_i)\). In these local coordinates, \( p \) becomes \((0, 0, 0)\), \( N_p \) lies along the positive \( z \)-axis, and the quadratic surface looks like:

\[
z = f(x, y) = \frac{A}{2}x^2 + Bxy + \frac{C}{2}y^2.
\]

It is easy to show that the Weingarten matrix for such a surface is

\[
W = \begin{pmatrix}
A & B \\
B & C
\end{pmatrix}.
\]

As in the Normal Curvature method, we plug in the adjacent vertices to get a system of equations

\[
\begin{pmatrix}
\frac{1}{2}x_i^2 & x_i y_i & \frac{1}{2}y_i^2 \end{pmatrix} \begin{pmatrix}
x_i \\
y_i \\
z_i
\end{pmatrix} = z_i \quad i = 1, \ldots, n.
\]

As before, we can find a least-squares fit to this system. It order to be able to compare the result to the normal curvature method, we scale the \( i^{th} \) equation by \( \frac{2}{k_i^2} \), where \( k_i = \sqrt{x_i^2 + y_i^2} \). Then \((x_i, y_i) = k_i(u_i, v_i)\) so that

\[
\begin{pmatrix}
\frac{1}{2}x_i^2 & x_i y_i & \frac{1}{2}y_i^2 \\
\end{pmatrix} \begin{pmatrix}
u_i^2 \\
2u_i v_i \\
v_i^2
\end{pmatrix} = \frac{k_i^2}{2} \begin{pmatrix}
u_i^2 \\
2u_i v_i \\
v_i^2
\end{pmatrix}.
\]
Then we can write each scaled equation in the system as:

\[
\begin{pmatrix}
  u_i^2 & 2u_iv_i & v_i^2
\end{pmatrix}
\begin{pmatrix}
  x
\end{pmatrix}
= d_i
\]

where \( d_i = \frac{2}{k_i^2}z_i \). There is a nice geometric interpretation of \( d_i \). Consider the parabola with equation \( y = \frac{d_i}{2}x^2 \). Plugging in \( x = k_i \) we get

\[
y = \frac{d_i}{2}k_i^2 = z_i
\]

so that this parabola passes through the origin and the point \((k_i, z_i)\). Since \( k_i = \sqrt{x_i^2 + y_i^2} \), we can think of this parabola in three dimensions as passing through the local coordinate origin and the point \((x_i, y_i, z_i)\).

The one variable, planar formula for computing curvature is:

\[
\kappa = \frac{|y''|}{(1 + y'^2)^{3/2}}.
\]

Using this formula for the parabola, at \((0,0)\) we get \( \kappa = |d_i| \). In other words, \( d_i \) is the normal curvature of a parabola passing through \((x_i, y_i, z_i)\) and the origin. We can now see that the Quadratic Surface method is identical to the Normal Curvature method, except that curvature approximations are done using parabolas rather than circles. This suggests that these two methods will produce similar results, with one performing slightly better than the other depending on whether circles or parabolas better approximate the surface locally in some direction.

### III.c The Adjacent-Normal Cubic Approximation Method

In both of the proceeding methods, we did not use all of the information available to us. Namely, we did not use the known (approximated) normal vectors at adjacent vertices \( q_i \). We can use this information to create a third-order approximation method that we believe has not yet appeared in the literature. As we will see in the next section, this method seems to perform significantly better than the first two in many cases.

As in the Quadratic method we try to fit a surface to the adjacent vertex data. Let

\[
f(x, y) = \frac{A}{2}x^2 + Bxy + \frac{C}{2}y^2 + Dx^3 + Ex^2y + Fxy^2 + Gy^3.
\]
The Weingarten matrix for this surface is still $W = \begin{pmatrix} A & B \\ B & C \end{pmatrix}$, because in the local coordinate system the curvature only depends on the second degree terms. However, using the third degree terms in the least-squares fit will produce values for $(A, B, C)$ different from the ones we get in the quadratic method. The normal to this surface is given by:

$$N(x, y) = (f_x(x, y), f_y(x, y), -1)$$

$$= (Ax + By + 3Dx^2 + 2Exy + Fy^2, Bx + Cy + Ex^2 + 2Fxy + 3Gy^2, -1).$$

Let $(a_i, b_i, c_i)$ denote the normal at the data point $(x_i, y_i, z_i)$ (both normal and point must be transformed to the local coordinates), and let $x = (A \ B \ C \ D \ E \ F \ G)^T$. Rewrite the normal as $(-\frac{a_i}{c_i}, -\frac{b_i}{c_i}, -1)$. Then for each point, we have an equation

$$(\frac{1}{2}x_i^2 \ x_iy_i \ \frac{1}{2}y_i^2 \ x_i^2 \ x_i^2y_i \ x_iy_i^2 \ y_i^2) \ x = z_i$$

and for each normal we have two equations:

$$(x_i \ y_i \ 0 \ 3x_i^2 \ 2x_iy_i \ y_i^2 \ 0) \ x = -\frac{a_i}{c_i}$$

$$(0 \ x_i \ y_i \ 0 \ x_i^2 \ 2x_iy_i \ 3y_i^2) \ x = -\frac{b_i}{c_i}.$$ 

As in the preceding method, we scale these equations by the same scale factor $\frac{2}{k_i}$, to obtain a system

$$U \ x = d$$

where $U$ is a $3n \times 7$ matrix and $d$ is a $3n$-vector. Again, we find a least-squares fit, but use only $A$, $B$, and $C$ from the result. Note that $n$ has to be at least 3 for there to be at least as many equations as unknowns.
III.d Higher Order Methods

The preceding method can be extended to higher orders:

\[ f_4(x, y) = \frac{A}{2}x^2 + Bxy + \frac{C}{2}y^2 + \sum_{n=0}^{n=3} D_n x^{3-n} y^n + \sum_{n=0}^{n=4} E_n x^{4-n} y^n \]

\[ f_5(x, y) = \frac{A}{2}x^2 + Bxy + \frac{C}{2}y^2 + \sum_{n=0}^{n=3} D_n x^{3-n} y^n + \sum_{n=0}^{n=4} E_n x^{4-n} y^n + \sum_{n=0}^{n=5} F_n x^{5-n} y^n \]

and so on. However, in order that there be at least as many equations as unknowns, the maximum degree of the approximation that we can compute is limited by the number of adjacent vertices in the mesh. For example, if we use 6 adjacent vertices, then there will be 18 equations. The highest order method that we can use is then degree 5, since such an approximation will have the 18 unknowns \((A, B, C, D_0, \ldots, D_3, E_0, \ldots, E_4, F_0, \ldots, F_5)\).

As we will see in what follows, higher order methods seem to make small errors smaller, but can also make large errors larger.

IV. Testing the Methods

We have noted that principal direction approximation methods are often tested using very simple models, such as surfaces of revolution triangulated using a regular pattern. Most methods seem to work well in these circumstances. However we have found that these previously proposed methods can fail for more complex surfaces with irregular mesh patterns. By fail we mean that the approximated principal directions can be wildly off from the true ones, even at non-umbilical points. We sought to gain deeper insight into the source of these errors, in order to better determine how it might be possible to attempt to overcome them.

We set out to design a test surface that had significant areas where the curvature was “interesting”, and we devised a number of different ways to create local triangular meshes on this surface.
Our surface, \( S(u,v) \), is defined by:

\[
\begin{align*}
  f(u) &= -2u^4 + 2u^2 + u/6 + 0.3 \\
  g(u) &= \begin{cases} 
  .9 + u + f(-.9), & -9 - f(-.9) \leq u \leq -9; \\
  f(u), & -9 \leq u \leq 1; \\
  1 - u - f(1), & 1 \leq u \leq 1 + f(1); 
\end{cases} \\
  h(u) &= \begin{cases} 
  -9, & -9 - f(-.9) \leq u \leq -9; \\
  u, & -9 \leq u \leq 1; \\
  1, & 1 \leq u \leq 1 + f(1); 
\end{cases} \\
  x(u,v) &= g(u) \cos(v) \\
  y(u,v) &= g(u) \sin(v) \\
  z(u,v) &= h(u) + 0.2 \sin(2r(u,v)) + 0.15 \cos(3r(u,v)y(u,v)) \\
  S(u,v) &= (x(u,v), y(u,v), z(u,v)) \\ .9 - f(-.9) \leq u \leq 1 + f(1), & 0 \leq v \leq 2\pi.
\end{align*}
\]

\[ \text{Figure 2. Test Surface} \]

We selected 575 points on this surface, none of which were umbilical points. Table 1 shows the distribution of principal curvature differences for these points.

\begin{center}
\begin{tabular}{ |c|c| } 
\hline
\( \lambda_1 - \lambda_2 \) & Points with \( \lambda_1 - \lambda_2 \) in indicated range \\
\hline
0-0.1 & 17 \\
0.1-0.2 & 47 \\
0.2-0.3 & 37 \\
0.3-0.4 & 29 \\
0.4-0.5 & 30 \\
> 0.5 & 415 \\
\hline
\end{tabular}
\end{center}

\[ \text{TABLE 1. Principal Curvature Differences on Test Surface} \]
In devising mesh schemes, we had a number of hypotheses we wished to test:

1. Does regularity of the mesh improve accuracy and randomness reduce it?
2. Does enforcing equidistance to adjacent vertices improve accuracy?
3. Does enforcing equiangularity between adjacent vertices improve accuracy?
4. Does choosing adjacent vertices along the true principal directions matter?

With these in mind, we used two approaches to creating meshes.

**IV.a Global Parameter Space Triangulations**

First we created a Global Regular triangulation of parameter space. Figure 3a shows a 4 by 4 global regular triangulation. The actual triangulation we used was 50 by 50. Then we created a Global Random triangulation by randomly moving the interior points. Figure 3b shows a 4 by 4 global random triangulation.

![Figure 3. Left: Global Regular Triangulation, Right: Global Random Triangulation.](image)

**IV.b Local Tangent Space Projections**

Next we created six local meshing patterns, which we defined independently around each of 575 individual vertices from the global random mesh. We used these patterns to investigate the hypotheses listed in the preceding section. We began by defining the six planar mesh patterns shown in Figure 4. These patterns were formed by using various rotations of a single radial edge about a given centerpoint.
In all cases six edges were used, with the first edge defined to be aligned with the analytically-determined first principal direction. In Local Mesh 1, the angle of rotation between all of the edges was defined to be equal (60°). In Local Mesh 2, the angles between the edges were defined in a regular pattern with 3-way symmetry, alternating 30° with 90°. Local Meshes 3-6 were generated by choosing the five successive angles of rotation randomly in [1° .. 90°].

Figure 4. Six Local Mesh patterns.

Figure 5. Surface mesh obtained from local mesh in tangent plane.
For each of these patterns and at each of the points on the surface created by the Global Random triangulation we then determined the points on the surface that projected onto the pattern in the tangent plane at that point (Figure 5). Finally, we rotated the patterns in the tangent plane 5 times (72 degrees each time). The initial orientation of each local mesh was configured to have one edge lying along a principal direction. Subsequent rotations moved this edge away from the principal direction. We did this in order to see if having mesh edges coincidentally aligned with or counter-balanced against the principal directions affected the accuracy of the approximation.

Figure 6 shows the estimated principal directions computed using each of the three methods in the case where the data is represented by the Global Regular mesh. All of the methods performed very well under these conditions. Figure 7 shows the estimated principal directions computed using the same three methods at the vertices of the Global Random mesh. Note the increase in the prevalence of large angle errors, especially with the two second order methods; estimated directions more than 10° out of alignment with the true principal directions are highlighted in red.

More results for our surface are summarized in Figures 8-9 and Tables 2-5 in the appendix. Figure 8 shows the median angle error, in degrees, for each of our methods, using each of our meshes for 575 points on the surface, across all rotations. The size and prevalence of errors varied according to the mesh type, with the smallest errors occurring in the local equi-angle case and the greatest errors occurring in the cases of the global and local random meshes and the local 30°-90° pattern. Performance was best, overall, when the Cubic method was used, and was not as good in the cases of the Quadratic and Normal Curvature methods. The performance in the cases of these two second order methods was very similar.

Figure 9 provides a more detailed look at the data, showing the percent of the mesh points having angle errors below certain thresholds, for each method. Figure 7a shows, for example, that for the Cubic method performance was fairly consistent across all mesh types, with approximately 22-27% of the mesh points having angle errors above 1°, and with that number rapidly dropping to below 5% for errors above 10°-15°. Figure 7b shows that performance was more varied by mesh type in the case of the Normal Curvature method, with between 24% to over 40% of the mesh points having angle errors above 1°, and with that number dropping below 5% at thresholds between 7°-22°, depending on mesh type.
Figure 6. Approximated principal directions, Global Regular Mesh. Directions marked in red have errors above $10^5$. Surface coloration highlights areas in which the difference between the two principal curvatures becomes small (near-umbilic points). Upper left: Exact directions. Upper right: Approximated directions, Quadratic Method. Lower left: Approximated directions, Normal Curvature Method. Lower right: Approximated directions, Cubic Method.
Figure 7. Approximated principal directions, Global Random Mesh. Directions marked in red have errors above $10^5$. Surface coloration highlights areas in which the difference between the two principal curvatures becomes small (near-umbilic points). Upper left: Exact directions. Upper right: Approximated directions, Quadratic Method. Lower left: Approximated directions, Normal Curvature Method. Lower right: approximated directions, Cubic Method.
Figure 8. Median error in the approximated principal directions, per mesh type and method.

Figure 9. Prevalence of errors in the approximated principal directions, by magnitude.

Table 2, given in the appendix, provides numeric data listing the number of points at which the degree error fell into each indicated range. For example, the first entry in the table says that out of 575 vertices tested using the Global Regular mesh, 498 had a principal direction error of less than 3°, 56 had an error between 3°-6°, and 21 had an error of more than 6°. The rotations only make sense for the six Local Mesh schemes. We found no apparent advantage for the principal-direction aligned rotations (see, for example, the equiangular case).
Table 3 shows that the results are virtually identical when the constraint of equal edge length on the surface (as opposed to on the local parameterization in the tangent plane) is enforced. At each vertex we recomputed the local mesh to have edges of length equal to the average of the 6 local edges from Table 2. In Table 4 we enforced a global equal length constraint. The average edge length in Table 2 is 0.08. Table 4 shows the result for all edges constrained to be of lengths 0.05 and 0.1. We show only the Rotation 1 column, as the other columns are similar.

Table 5 separates out those points with a principal curvature difference of less than 0.2. Each entry of the form n(m) is read as n points with curvature difference greater than 0.2 and m points with curvature difference less than 0.2, so for example, the first entry 464(34) 33(23) 18(3) says that of the 21 points where there was an error of more than 6 degrees, only 3 of them were even close to being umbilical points. Table 5 is perhaps the most revealing. It reveals that large principal direction errors can occur away from umbilical points.

The results suggest several conclusions:

1. The degree ≥ 3 methods nearly consistently outperform both of the degree 2 methods.
2. The regular meshes tend to produce better results than the random ones.
3. The equiangular mesh (Local Mesh 1) produces good results for all three methods.
4. Aligning edges with principal directions does not seem to matter.
5. Enforcing equality of adjacent edge length does not seem to matter, but the shorter all of the edges are the better.
6. Large errors can occur away from umbilical points.

V. A Unified Approach to Error Analysis

The previous section illuminates how well or badly the three methods that we examined can work. We saw that some mesh schemes seem to produce fewer large errors than others, but that in nearly every case there are non-umbilical points where the error is large. In this section we will develop the mathematics necessary to understand why this is so.
From a mathematical point of view, all three of the methods outlined in Section III have the same two steps:

1) Find a least-squares fit to a linear system

\[ Ux = d \]

where \( U \) is computed from the mesh geometry and \( d \) is computed from measurements at the adjacent vertices in the surface mesh.

2) Use the first three entries, \[ \begin{pmatrix} A' \\ B' \\ C' \end{pmatrix} \], in the least-squares solution \( x \) to form a symmetric matrix

\[ W' = \begin{pmatrix} A' & B' \\ B' & C' \end{pmatrix} \]

and then find the eigenvalues and eigenvectors of \( W' \).

We seek to understand how the local mesh geometry and the measurement error interact to create error in principal direction.

We begin with a standard result from linear algebra:

**Theorem 1.** A least squares solution to \( Ux = d \) is the exact solution to \( U^TUx = U^Td \).

In our situation, \( U \) is known exactly and is computed from the projection of the adjacent vertices onto the local tangent plane, but \( d \) is known only approximately. If we let \( d_T \) denote the (unknown) vector that \( d \) approximates and \( x_T \) the exact solution to \( U^TUx_T = U^Td_T \), then

\[ U^TU(x - x_T) = U^T(d - d_T). \]

Then \( e = x - x_T = \begin{pmatrix} e_A \\ e_B \\ e_C \end{pmatrix} \) is what creates the error in our approximation to the Weingarten matrix. More precisely,

\[ W' = \begin{pmatrix} A' & B' \\ B' & C' \end{pmatrix} = \begin{pmatrix} A & B \\ B & C \end{pmatrix} + \begin{pmatrix} e_A & e_B \\ e_B & e_C \end{pmatrix}. \]

We will see in what follows that it is not so much the magnitude of \( e \) that creates principal direction errors, but rather the relationships between these errors that really determines the magnitude of the approximation errors.

The first step in our analysis is to show that direction errors are independent of the choice of local coordinates. It might seem that since the entries in the matrix \( U \) depend on the choice of local coordinates
in the tangent plane, the entries in the matrix $W'$ will also depend on the local coordinates and this, in turn, will affect errors in direction. However, we can prove the following theorem:

**Theorem 2.** Suppose $L$ and $L'$ are two different orthonormal coordinate systems in the tangent plane and $U$ and $U'$ are the coefficient matrices found in any of our approximation methods using $L$ and $L'$, respectively. Then the approximated Weingarten matrices have the same eigenvalues and eigenvectors.

**Proof:** We prove the theorem for the Normal Curvature method, as the other methods have a similar proof. If $(u, v)$ and $(u', v')$ are the coordinates of a point in $L$ and $L'$, respectively, then

$$
\begin{pmatrix}
  u' \\
  v'
\end{pmatrix} =
\begin{pmatrix}
  \cos \theta & \sin \theta \\
  -\sin \theta & \cos \theta
\end{pmatrix}
\begin{pmatrix}
  u \\
  v
\end{pmatrix} = R
\begin{pmatrix}
  u \\
  v
\end{pmatrix}.
$$

Using this, it is not hard to show that $U' = US$ where

$$S =
\begin{pmatrix}
  \cos^2 \theta & 2\cos\theta\sin\theta & \sin^2 \theta \\
  -\sin\theta\cos\theta & \cos^2 \theta - \sin^2 \theta & \sin\theta\cos\theta \\
  \sin^2 \theta & -2\sin\theta\cos\theta & \cos^2 \theta
\end{pmatrix}.
$$

Let $U^T U x = U^T d$ and $U'^T U' x' = U'^T d$. Replacing $U'$ by $US$ in the second equation produces

$$S^T U^T U S x' = S^T U'^T d.$$

Since $S$ is invertible, we get $U^T U S x' = U'^T d$. That is, $x = S x'$. Now if $x = (A, B, C)$ and $x' = (A', B', C')$, it is not hard to show that

$$
\begin{pmatrix}
  A & B \\
  B & C
\end{pmatrix} = R^{-1}
\begin{pmatrix}
  A & B \\
  B & C
\end{pmatrix} R.
$$

That is, the approximate Weingarten matrices are orthogonally similar, so they have the same eigenvalues and eigenvectors. This completes the proof.

Theorem 2 tells us that our direction errors will not depend on how we chose the local coordinates. As a consequence of this theorem, it suffices to choose the true principal directions as our local coordinates. This will simplify some of our later calculations. In particular, with respect to these local coordinates, $W = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$ where $\lambda_1$ and $\lambda_2$ are the principal curvatures. Suppose $W'$ is our approximated matrix with respect to these local coordinates. We express $W'$ in two ways:
(1) Since $W'$ approximates $W$, 

$$\begin{pmatrix} \lambda_1 + e_A & e_B \\ e_B & \lambda_2 + e_C \end{pmatrix}.$$ 

(2) Since $W'$ is a symmetric 2 by 2 matrix, it can be orthogonally diagonalized. That is, it can be written

$$W' = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} \tau_1 & 0 \\ 0 & \tau_2 \end{pmatrix} \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix},$$

where $\tau_1$ and $\tau_2$ are the eigenvalues of $W'$ and $\begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix}$ and $\begin{pmatrix} -\sin \theta \\ \cos \theta \end{pmatrix}$ are orthonormal eigenvectors. Note also that since we have chosen the true eigenvectors as our local coordinate system, $\theta$ is precisely the angle error between the true directions and the approximated ones. We can find $\theta$ by equating these two ways of writing $W'$. First, multiply out the three matrices in (2) to obtain:

$$W' = \begin{pmatrix} \tau_1 \cos^2 \theta + \tau_2 \sin^2 \theta & \tau_1 \sin \theta \cos \theta - \tau_2 \sin \theta \cos \theta \\ \tau_1 \sin \theta \cos \theta - \tau_2 \sin \theta \cos \theta & \tau_1 \sin^2 \theta + \tau_2 \cos^2 \theta \end{pmatrix}.$$ 

Equating corresponding entries in the two ways to express $W'$, we obtain three equations:

1. $\tau_1 \cos^2 \theta + \tau_2 \sin^2 \theta = \lambda_1 + e_A$;
2. $\tau_1 \sin^2 \theta + \tau_2 \cos^2 \theta = \lambda_2 + e_C$;
3. $\tau_1 \sin \theta \cos \theta - \tau_2 \sin \theta \cos \theta = e_B$.

Subtracting equation (2) from equation (1), and multiplying equation (3) by 2, we obtain, respectively:

4. $(\tau_1 - \tau_2) \cos(2\theta) = e_A - e_C + \lambda_1 - \lambda_2$;
5. $(\tau_1 - \tau_2) \sin(2\theta) = 2e_B$.

Finally, dividing equation (5) by equation (4) we get the result:

6. $\tan(2\theta) = \frac{2e_B}{e_A - e_C + \lambda_1 - \lambda_2}$.

We summarize this in the following theorem:
**Principal Error Theorem**

Suppose \( W' = W + E \) where

\[
W = \begin{pmatrix}
  \lambda_1 & 0 \\
  0 & \lambda_2
\end{pmatrix}, \quad E = \begin{pmatrix}
  e_A & e_B \\
  e_B & e_C
\end{pmatrix}
\]

Then the angle \( \theta \) between the first eigenvectors of \( W \) and \( W' \) satisfies

\[
\tan(2\theta) = \frac{2e_B}{e_A - e_C + \lambda_1 - \lambda_2}.
\]

Since \( \tan \) is an increasing function, angle errors will be large when

\[
\frac{2e_B}{e_A - e_C + \lambda_1 - \lambda_2}
\]

is large.

Many simulations have convinced us that the size of the numerator in this expression is the most significant culprit in producing angle errors. Let’s examine more carefully how it is computed.

From our earlier work in this section, the total error vector \( e = (e_A, e_B, e_C, \ldots) \) is the solution to

\[
U^T U e = U^T \delta d
\]

where \( U \) is computed from the local mesh projected onto the local tangent plane, and \( \delta d = d - d_T \) is the vector of errors we made because we did not compute the exact normal curvatures, (or the surface is not exactly quadratic or cubic, depending which method we are using). Hence

\[
e = (U^T U)^{-1} U^T \delta d.
\]

If we let \( Y = (U^T U)^{-1} U^T \), and let \( Y_i \) denote the \( i \)‘th row of \( Y \), then

\[
e_B = Y_2 \circ \delta d = \|Y_2\| \|\delta d\| \cos \theta_{dY_2},
\]

where \( \theta_{dY_2} \) is the angle between \( Y_2 \) and \( \delta d \). \( Y_2 \) is fixed by the choice of local mesh, so we see that \( e_B \) can vary widely depending on \( \theta_{dY_2} \). In the worst case \( (\theta_{dY_2} = 0) \), \( e_B = \|Y_2\| \|\delta d\| \) and in the best case \( (\theta_{dY_2} = \frac{\pi}{2}) \), even if \( \delta d \) has large magnitude, \( e_B = 0 \).
We illustrate with an example how volatile the situation can be. Suppose we are using Local Mesh 3 at a point where the difference between the principal curvatures is 0.5, that is, away from an umbilical point.

This mesh’s local $uv$ coordinates are:

\[
\begin{pmatrix}
  u_1 & v_1 \\
  u_2 & v_2 \\
  u_3 & v_3 \\
  u_4 & v_4 \\
  u_5 & v_5 \\
  u_6 & v_6
\end{pmatrix} = \begin{pmatrix}
  1 & 0 \\
  0.0334296 & 0.999441 \\
  -0.0698233 & 0.997559 \\
  -0.999741 & 0.0227421 \\
  -0.0945938 & -0.995516 \\
  0.360022 & -0.932944
\end{pmatrix}
\]

and

\[
Y = (U^T U)^{-1} U^T
\]

\[
= \begin{pmatrix}
  0.504502 & 0.00796006 & -0.0165789 & 0.498407 & 0.0275063 & -0.0217962 \\
  0.128266 & 0.427104 & -0.0347749 & 0.026751 & 0.697892 & -1.24506 \\
  -0.00117602 & 0.321524 & 0.263133 & -0.0136473 & 0.352888 & 0.0772777
\end{pmatrix}
\]

First we choose an error vector in the same direction as the second row of $Y$:

\[
\delta d_1 = (0.0256332, 0.0854208, -0.00695498, 0.00533501, 0.139578, -0.249012)
\]

To do this, we scaled down the second row to produce errors of a magnitude consistent with using a circle to approximate normal curvatures along 6 adjacent vertices. Using $\delta d_1$ produces the error vector:

\[
e_1 = Y \delta d_1 = (0.0256332, 0.447604, 0.0555445)
\]

We then apply the Principal Error Theorem:

\[
\tan(2\theta) = \frac{2e_B}{e_A - e_C + \lambda_1 - \lambda_2} = \frac{2(0.447604)}{0.0256332 - 0.0555445 + 0.5} = 1.904.
\]

producing an error of $\theta = 31.1471$ degrees.

Now we choose an error vector whose entries are close to those in $\delta d_1$, but which is perpendicular to the second row of $Y$:

\[
\delta d_2 = (0.02, 0.28, 0.1, -0.05, 0.1, 0.150304)
\]

In this case, $e_2 = Y \delta d_2 = (-0.0147848, 0.0163903)$ which produces an error of 0 degrees!
Figure 10 shows the correlation between the magnitude of the error and the each of the components in (6) for two of the methods examined, at 575 points on our test surface and using six different rotations of each local mesh at each point.

Note that the size of the numerator, $2e_B$, depends upon $\|Y_2\|$, which is determined by the mesh, $\|d\|$, which reflects the magnitude of the errors we have made in approximating the normal curvatures, and $\theta_{dY_2}$, the angle between $Y_2$ and $\delta d$. Note also that the error vectors for the degree 3 method have 18 components, not 6, so although the magnitude of $Y_2$ in these cases is significantly smaller, the magnitude of an 18 component $\delta d$ is likely to be greater. The size of the denominator in (6) depends both upon $e_A - e_C$ and upon the difference between the two principal curvatures, $\lambda_1 - \lambda_2$. By definition, $\lambda_1 > \lambda_2$. The magnitude of $e_A - e_C$ in turn depends upon $\|Y_1 - Y_3\|$, $\|d\|$, and the angle between $Y_1 - Y_3$ and $\delta d$. When both $\lambda_1 - \lambda_2$ and $e_A - e_C$ are small, larger errors become more likely.

However, it seems that it is the size of the numerator that has the most significant effect. In the case of the cubic method, the sizes of the errors in the estimates of the normal curvatures seems to be the factor most significantly influencing the size of the error in the principal direction estimates; in the case of the normal curvature method, the direction in which the errors were made seems for some meshes to be nearly as significant.

![Factors Correlating with Error](image1.png)

**Factors Correlating with Error**

**Cubic Method**

- $\|d\|$
- $\lambda_1 - \lambda_2$
- $\|Y_2\|$
- $\|Y_1 - Y_3\|$
- $\theta_{dY_2}$
- $\theta_{Y_1Y_3}$
- $e_A - e_C$
- $\|Y_1 - Y_3\|$
- $\|d\|$  

**Normal Curvature Method**

- $\|d\|$
- $\lambda_1 - \lambda_2$
- $\|Y_2\|$
- $\|Y_1 - Y_3\|$
- $\theta_{dY_2}$
- $\theta_{Y_1Y_3}$
- $e_A - e_C$
- $\|Y_1 - Y_3\|$
- $\|d\|$  

*Figure 10. Correlation of the various components of equation (6) with error.*
Table 6 in the appendix shows the median values of each of the components of (6) by mesh type and method. From this table it can be seen that the median errors appear to be smallest when the median angle between $\delta d$ and $Y_2$ is closest to 90°, indicating the advantage of situations in which a higher proportion of errors can cancel out.

V. Conclusions

We have shown that approximating principal directions on a surface mesh can be a tricky business. Although it is important to minimize the error in approximating various quantities such as normal curvature, an unfortunate pattern of small errors can produce large angle errors. Using a non-trivial test surface, we have shown that principal direction approximation methods are particularly prone to error when working with irregularly sampled data, and that these errors appear to be more severe for second order methods than for higher order methods that take advantage of more of the known information about the surface.

There are many important directions for future work in this area. A major practical concern that we did not touch upon at all in our work is the problem of surface smoothing. In many cases, approximate meshes are obtained from a sampling process that is prone to a certain amount of intrinsic error. How might we best recover the mesh that describes the continuous smooth surface that is approximated by a collection of noisy samples? If we have only a mesh that is highly irregular as well as noisy, how could we improve our chances of obtaining a smooth principal direction vector field and avoiding the introduction of large principal direction estimation errors? Would resampling to obtain a mesh in which the edge distribution is more equiangular help? Finally, it might be interesting to see how well the findings we have made hold up under further testing with a wider variety of mesh types and surface shapes.
<table>
<thead>
<tr>
<th>Mesh Type</th>
<th>Rotation 1</th>
<th>Rotation 2</th>
<th>Rotation 3</th>
<th>Rotation 4</th>
<th>Rotation 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree Error</td>
<td>0-3</td>
<td>3-6 &gt;6</td>
<td>0-3</td>
<td>3-6 &gt;6</td>
<td>0-3</td>
</tr>
</tbody>
</table>

**Normal Curvature Method**

<table>
<thead>
<tr>
<th>Mesh Type</th>
<th>Global Regular</th>
<th>Global Random</th>
<th>Local Mesh 1</th>
<th>Local Mesh 2</th>
<th>Local Mesh 3</th>
<th>Local Mesh 4</th>
<th>Local Mesh 5</th>
<th>Local Mesh 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree Error</td>
<td>498 56 21</td>
<td>374 84 117</td>
<td>514 28 33</td>
<td>327 101 147</td>
<td>252 128 195</td>
<td>316 124 135</td>
<td>324 153 98</td>
<td>466 58 51</td>
</tr>
</tbody>
</table>

**Quadratic Approximation**

<table>
<thead>
<tr>
<th>Mesh Type</th>
<th>Global Regular</th>
<th>Global Random</th>
<th>Local Mesh 1</th>
<th>Local Mesh 2</th>
<th>Local Mesh 3</th>
<th>Local Mesh 4</th>
<th>Local Mesh 5</th>
<th>Local Mesh 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree Error</td>
<td>501 54 20</td>
<td>372 87 116</td>
<td>512 30 33</td>
<td>322 104 149</td>
<td>243 134 198</td>
<td>316 130 129</td>
<td>320 157 98</td>
<td>458 64 53</td>
</tr>
</tbody>
</table>

**Cubic Approximation**

<table>
<thead>
<tr>
<th>Mesh Type</th>
<th>Global Regular</th>
<th>Global Random</th>
<th>Local Mesh 1</th>
<th>Local Mesh 2</th>
<th>Local Mesh 3</th>
<th>Local Mesh 4</th>
<th>Local Mesh 5</th>
<th>Local Mesh 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree Error</td>
<td>514 24 37</td>
<td>305 22 48</td>
<td>500 25 50</td>
<td>507 26 42</td>
<td>522 22 31</td>
<td>504 27 44</td>
<td>519 24 32</td>
<td>491 35 49</td>
</tr>
</tbody>
</table>

**Degree 4 Approximation**

<table>
<thead>
<tr>
<th>Mesh Type</th>
<th>Global Regular</th>
<th>Global Random</th>
<th>Local Mesh 1</th>
<th>Local Mesh 2</th>
<th>Local Mesh 3</th>
<th>Local Mesh 4</th>
<th>Local Mesh 5</th>
<th>Local Mesh 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree Error</td>
<td>511 25 39</td>
<td>495 30 60</td>
<td>525 12 38</td>
<td>483 28 64</td>
<td>513 13 49</td>
<td>500 17 58</td>
<td>507 25 43</td>
<td>501 24 50</td>
</tr>
</tbody>
</table>

**Degree 5 Approximation**

<table>
<thead>
<tr>
<th>Mesh Type</th>
<th>Global Regular</th>
<th>Global Random</th>
<th>Local Mesh 1</th>
<th>Local Mesh 2</th>
<th>Local Mesh 3</th>
<th>Local Mesh 4</th>
<th>Local Mesh 5</th>
<th>Local Mesh 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree Error</td>
<td>523 4 48</td>
<td>518 3 54</td>
<td>527 6 42</td>
<td>529 7 39</td>
<td>521 12 42</td>
<td>528 10 37</td>
<td>523 11 41</td>
<td>333 3 39</td>
</tr>
<tr>
<td>Mesh Type</td>
<td>Rotation 1</td>
<td>Rotation 2</td>
<td>Rotation 3</td>
<td>Rotation 4</td>
<td>Rotation 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Global Regular</td>
<td>518 25 32</td>
<td>518 25 32</td>
<td>318 117 140</td>
<td>376 89 110</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local Mesh 1</td>
<td>472 48 55</td>
<td>442 71 62</td>
<td>492 36 47</td>
<td>458 61 56</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local Mesh 2</td>
<td>509 32 33</td>
<td>509 32 33</td>
<td>509 32 33</td>
<td>509 32 33</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local Mesh 3</td>
<td>481 45 46</td>
<td>481 45 46</td>
<td>481 45 46</td>
<td>481 45 46</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local Mesh 4</td>
<td>492 36 47</td>
<td>492 36 47</td>
<td>492 36 47</td>
<td>492 36 47</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local Mesh 5</td>
<td>509 32 33</td>
<td>509 32 33</td>
<td>509 32 33</td>
<td>509 32 33</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local Mesh 6</td>
<td>469 52 54</td>
<td>498 39 38</td>
<td>498 39 38</td>
<td>498 39 38</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE 3. Degree Errors for 575 Points on Test Surface — Local Equal Edge Length Enforced

Degree 2 Approximation

<table>
<thead>
<tr>
<th>Mesh Type</th>
<th>Rotation 1</th>
<th>Rotation 2</th>
<th>Rotation 3</th>
<th>Rotation 4</th>
<th>Rotation 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Regular</td>
<td>376 89 110</td>
<td>376 89 110</td>
<td>376 89 110</td>
<td>376 89 110</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 1</td>
<td>321 103 151</td>
<td>417 55 103</td>
<td>350 112 113</td>
<td>321 86 168</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 2</td>
<td>321 103 151</td>
<td>417 55 103</td>
<td>350 112 113</td>
<td>321 86 168</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 3</td>
<td>243 135 197</td>
<td>308 181 86</td>
<td>383 90 102</td>
<td>502 35 38</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 4</td>
<td>318 117 140</td>
<td>432 68 75</td>
<td>439 70 66</td>
<td>472 48 55</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 5</td>
<td>324 163 88</td>
<td>442 73 60</td>
<td>522 26 27</td>
<td>495 44 36</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 6</td>
<td>469 52 54</td>
<td>498 39 38</td>
<td>498 39 38</td>
<td>498 39 38</td>
<td></td>
</tr>
</tbody>
</table>

Degree 3 Approximation

<table>
<thead>
<tr>
<th>Mesh Type</th>
<th>Rotation 1</th>
<th>Rotation 2</th>
<th>Rotation 3</th>
<th>Rotation 4</th>
<th>Rotation 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Regular</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 1</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 2</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 3</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 4</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 5</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 6</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td>301 54 20</td>
<td></td>
</tr>
</tbody>
</table>

Degree 4 Approximation

<table>
<thead>
<tr>
<th>Mesh Type</th>
<th>Rotation 1</th>
<th>Rotation 2</th>
<th>Rotation 3</th>
<th>Rotation 4</th>
<th>Rotation 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Mesh 1</td>
<td>305 14 56</td>
<td>494 23 58</td>
<td>500 19 56</td>
<td>496 22 57</td>
<td>493 22 60</td>
</tr>
<tr>
<td>Local Mesh 2</td>
<td>305 14 56</td>
<td>494 23 58</td>
<td>500 19 56</td>
<td>496 22 57</td>
<td>493 22 60</td>
</tr>
<tr>
<td>Local Mesh 3</td>
<td>524 9 42</td>
<td>526 16 33</td>
<td>454 41 80</td>
<td>444 39 92</td>
<td>514 18 43</td>
</tr>
<tr>
<td>Local Mesh 4</td>
<td>512 15 48</td>
<td>530 16 29</td>
<td>533 10 32</td>
<td>509 18 48</td>
<td>495 25 55</td>
</tr>
<tr>
<td>Local Mesh 5</td>
<td>519 18 38</td>
<td>503 20 52</td>
<td>435 34 106</td>
<td>421 33 121</td>
<td>454 39 82</td>
</tr>
<tr>
<td>Local Mesh 6</td>
<td>508 19 48</td>
<td>525 17 43</td>
<td>501 18 56</td>
<td>526 13 36</td>
<td>496 30 49</td>
</tr>
</tbody>
</table>

Degree 5 Approximation

<table>
<thead>
<tr>
<th>Mesh Type</th>
<th>Rotation 1</th>
<th>Rotation 2</th>
<th>Rotation 3</th>
<th>Rotation 4</th>
<th>Rotation 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Regular</td>
<td>523 4 48</td>
<td>523 4 48</td>
<td>523 4 48</td>
<td>523 4 48</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 1</td>
<td>530 14 31</td>
<td>520 13 42</td>
<td>533 7 35</td>
<td>532 2 41</td>
<td>525 14 36</td>
</tr>
<tr>
<td>Local Mesh 2</td>
<td>533 2 40</td>
<td>543 4 28</td>
<td>534 10 31</td>
<td>537 1 37</td>
<td>540 3 32</td>
</tr>
<tr>
<td>Local Mesh 3</td>
<td>530 10 35</td>
<td>537 5 33</td>
<td>540 3 32</td>
<td>540 3 32</td>
<td>532 5 38</td>
</tr>
<tr>
<td>Local Mesh 4</td>
<td>535 5 35</td>
<td>544 9 22</td>
<td>531 11 33</td>
<td>539 9 27</td>
<td>521 12 42</td>
</tr>
<tr>
<td>Local Mesh 5</td>
<td>528 12 35</td>
<td>536 5 34</td>
<td>539 1 35</td>
<td>539 2 34</td>
<td>531 5 39</td>
</tr>
<tr>
<td>Local Mesh 6</td>
<td>534 4 37</td>
<td>528 8 39</td>
<td>525 11 39</td>
<td>533 3 39</td>
<td>529 4 42</td>
</tr>
</tbody>
</table>
## TABLE 4. Degree Errors for 575 Points on Test Surface — Global Equal Length Edges

<table>
<thead>
<tr>
<th></th>
<th>Length = 0.05</th>
<th>Length = 0.10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree Error</td>
<td>0-3 3-6 &gt;6</td>
<td>0-3 3-6 &gt;6</td>
</tr>
</tbody>
</table>

### Normal Curvature Method

<table>
<thead>
<tr>
<th>Local Mesh</th>
<th>Degree Error</th>
<th>Degree Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>369 5 1</td>
<td>489 29 57</td>
</tr>
<tr>
<td>2</td>
<td>397 111 67</td>
<td>267 105 203</td>
</tr>
<tr>
<td>3</td>
<td>345 98 132</td>
<td>145 177 253</td>
</tr>
<tr>
<td>4</td>
<td>408 119 48</td>
<td>246 124 205</td>
</tr>
<tr>
<td>5</td>
<td>457 92 26</td>
<td>197 223 155</td>
</tr>
<tr>
<td>6</td>
<td>538 24 13</td>
<td>418 71 86</td>
</tr>
</tbody>
</table>

### Degree 2 Approximation

<table>
<thead>
<tr>
<th>Local Mesh</th>
<th>Degree Error</th>
<th>Degree Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>569 5 1</td>
<td>486 32 57</td>
</tr>
<tr>
<td>2</td>
<td>396 112 67</td>
<td>266 103 206</td>
</tr>
<tr>
<td>3</td>
<td>347 96 132</td>
<td>138 186 251</td>
</tr>
<tr>
<td>4</td>
<td>407 120 48</td>
<td>250 122 203</td>
</tr>
<tr>
<td>5</td>
<td>456 93 26</td>
<td>184 230 161</td>
</tr>
<tr>
<td>6</td>
<td>538 24 13</td>
<td>404 84 87</td>
</tr>
</tbody>
</table>

### Degree 3 Approximation

<table>
<thead>
<tr>
<th>Local Mesh</th>
<th>Degree Error</th>
<th>Degree Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>570 0 5</td>
<td>473 34 68</td>
</tr>
<tr>
<td>2</td>
<td>571 0 4</td>
<td>475 35 65</td>
</tr>
<tr>
<td>3</td>
<td>571 1 3</td>
<td>497 27 51</td>
</tr>
<tr>
<td>4</td>
<td>571 0 4</td>
<td>470 40 65</td>
</tr>
<tr>
<td>5</td>
<td>571 1 3</td>
<td>488 32 55</td>
</tr>
<tr>
<td>6</td>
<td>568 1 6</td>
<td>462 40 73</td>
</tr>
</tbody>
</table>

### Degree 4 Approximation

<table>
<thead>
<tr>
<th>Local Mesh</th>
<th>Degree Error</th>
<th>Degree Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>574 0 1</td>
<td>463 27 85</td>
</tr>
<tr>
<td>2</td>
<td>574 0 1</td>
<td>438 47 90</td>
</tr>
<tr>
<td>3</td>
<td>575 0 0</td>
<td>499 21 55</td>
</tr>
<tr>
<td>4</td>
<td>574 0 1</td>
<td>470 28 77</td>
</tr>
<tr>
<td>5</td>
<td>574 0 1</td>
<td>497 27 51</td>
</tr>
<tr>
<td>6</td>
<td>572 0 3</td>
<td>484 29 62</td>
</tr>
</tbody>
</table>

### Degree 5 Approximation

<table>
<thead>
<tr>
<th>Local Mesh</th>
<th>Degree Error</th>
<th>Degree Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>574 0 1</td>
<td>509 10 56</td>
</tr>
<tr>
<td>2</td>
<td>574 0 1</td>
<td>515 6 54</td>
</tr>
<tr>
<td>3</td>
<td>574 0 1</td>
<td>519 7 49</td>
</tr>
<tr>
<td>4</td>
<td>574 0 1</td>
<td>518 6 51</td>
</tr>
<tr>
<td>5</td>
<td>574 0 1</td>
<td>513 10 52</td>
</tr>
<tr>
<td>6</td>
<td>572 0 3</td>
<td>514 4 57</td>
</tr>
</tbody>
</table>
### TABLE 5. Degree Errors for 575 Points Separated by Principal Curvature Difference of 0.2 on Test Surface

<table>
<thead>
<tr>
<th>Mesh Type</th>
<th>Rotation 1</th>
<th>0-3</th>
<th>3-6</th>
<th>&gt;6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal Curvature Method</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Global Regular</td>
<td>464(34)</td>
<td>33(23)</td>
<td>18(3)</td>
<td></td>
</tr>
<tr>
<td>Global Random</td>
<td>337(37)</td>
<td>71(13)</td>
<td>103(14)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 1</td>
<td>460(54)</td>
<td>25(3)</td>
<td>26(7)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 2</td>
<td>317(10)</td>
<td>64(37)</td>
<td>130(17)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 3</td>
<td>232(20)</td>
<td>116(12)</td>
<td>163(32)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 4</td>
<td>297(19)</td>
<td>98(26)</td>
<td>116(19)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 5</td>
<td>300(24)</td>
<td>130(23)</td>
<td>81(17)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 6</td>
<td>425(41)</td>
<td>46(12)</td>
<td>40(11)</td>
<td></td>
</tr>
</tbody>
</table>

### Degree 2 Approximation

<table>
<thead>
<tr>
<th>Mesh Type</th>
<th>Rotation 1</th>
<th>0-3</th>
<th>3-6</th>
<th>&gt;6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Regular</td>
<td>466(35)</td>
<td>32(22)</td>
<td>17(3)</td>
<td></td>
</tr>
<tr>
<td>Global Random</td>
<td>335(37)</td>
<td>74(13)</td>
<td>102(14)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 1</td>
<td>458(54)</td>
<td>27(3)</td>
<td>26(7)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 2</td>
<td>312(10)</td>
<td>67(37)</td>
<td>132(17)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 3</td>
<td>223(20)</td>
<td>122(12)</td>
<td>166(32)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 4</td>
<td>297(19)</td>
<td>104(26)</td>
<td>110(19)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 5</td>
<td>296(24)</td>
<td>135(22)</td>
<td>80(18)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 6</td>
<td>417(41)</td>
<td>52(12)</td>
<td>42(11)</td>
<td></td>
</tr>
</tbody>
</table>

### Degree 3 Approximation

<table>
<thead>
<tr>
<th>Mesh Type</th>
<th>Rotation 1</th>
<th>0-3</th>
<th>3-6</th>
<th>&gt;6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Regular</td>
<td>458(56)</td>
<td>24(0)</td>
<td>33(4)</td>
<td></td>
</tr>
<tr>
<td>Global Random</td>
<td>448(57)</td>
<td>19(3)</td>
<td>44(4)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 1</td>
<td>445(55)</td>
<td>23(2)</td>
<td>43(7)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 2</td>
<td>453(54)</td>
<td>22(4)</td>
<td>36(6)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 3</td>
<td>467(55)</td>
<td>18(4)</td>
<td>26(5)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 4</td>
<td>450(54)</td>
<td>23(4)</td>
<td>38(6)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 5</td>
<td>464(55)</td>
<td>20(4)</td>
<td>27(5)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 6</td>
<td>436(55)</td>
<td>35(0)</td>
<td>40(9)</td>
<td></td>
</tr>
</tbody>
</table>

### Degree 4 Approximation

<table>
<thead>
<tr>
<th>Mesh Type</th>
<th>Rotation 1</th>
<th>0-3</th>
<th>3-6</th>
<th>&gt;6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Regular</td>
<td>454(57)</td>
<td>24(1)</td>
<td>37(2)</td>
<td></td>
</tr>
<tr>
<td>Global Random</td>
<td>437(58)</td>
<td>18(2)</td>
<td>56(4)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 1</td>
<td>467(58)</td>
<td>10(2)</td>
<td>34(4)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 2</td>
<td>426(57)</td>
<td>27(1)</td>
<td>58(6)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 3</td>
<td>455(58)</td>
<td>12(1)</td>
<td>44(5)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 4</td>
<td>441(59)</td>
<td>16(1)</td>
<td>54(4)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 5</td>
<td>451(56)</td>
<td>21(4)</td>
<td>39(4)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 6</td>
<td>447(54)</td>
<td>20(4)</td>
<td>44(6)</td>
<td></td>
</tr>
</tbody>
</table>

### Degree 5 Approximation

<table>
<thead>
<tr>
<th>Mesh Type</th>
<th>Rotation 1</th>
<th>0-3</th>
<th>3-6</th>
<th>&gt;6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Regular</td>
<td>463(60)</td>
<td>4(0)</td>
<td>48(0)</td>
<td></td>
</tr>
<tr>
<td>Global Random</td>
<td>457(61)</td>
<td>3(0)</td>
<td>51(3)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 1</td>
<td>467(60)</td>
<td>6(0)</td>
<td>38(4)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 2</td>
<td>470(59)</td>
<td>7(0)</td>
<td>34(5)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 3</td>
<td>461(60)</td>
<td>11(1)</td>
<td>39(3)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 4</td>
<td>468(60)</td>
<td>10(0)</td>
<td>33(4)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 5</td>
<td>462(61)</td>
<td>11(0)</td>
<td>38(3)</td>
<td></td>
</tr>
<tr>
<td>Local Mesh 6</td>
<td>473(60)</td>
<td>3(0)</td>
<td>35(4)</td>
<td></td>
</tr>
</tbody>
</table>

29
### TABLE 6. Components of the Principal Direction Errors - Median Values

#### Normal Curvature Method

| Error Type       | Error | Num. | Denom. | $||\nu_2||$ | $||Y_1Y_3||$ | $||d||$ | $\theta d\nu_2$ | $\theta dY_1Y_3$ |
|------------------|-------|------|--------|-------------|-------------|--------|----------------|-----------------|
| Global Regular   | 0.5816| 0.0242| 1.5943 | 0.8866      | 1.0961      | 0.2871 | 85.7942        | 85.9385         |
| Global Random    | 1.6724| 0.0826| 1.5751 | 0.7080      | 1.2248      | 0.3196 | 76.7223        | 74.3489         |
| Local Equi-angle | 0.1918| 0.0167| 1.7288 | 0.5774      | 1.1547      | 0.3040 | 87.6794        | 86.1888         |
| Local 30-90      | 1.8221| 0.0837| 1.6099 | 0.5773      | 1.1547      | 0.3138 | 71.1061        | 69.5400         |
| Local Random 1   | 1.9897| 0.0948| 1.6284 | 1.2098      | 1.9767      | 0.2728 | 76.3860        | 78.1628         |
| Local Random 2   | 1.5147| 0.0737| 1.6646 | 0.6342      | 1.2462      | 0.3057 | 75.1857        | 68.6009         |
| Local Random 3   | 1.4092| 0.0798| 1.7039 | 1.0117      | 1.6519      | 0.2725 | 80.3466        | 81.3653         |
| Local Random 4   | 0.9644| 0.0460| 1.6908 | 0.5838      | 1.2010      | 0.3124 | 80.7168        | 76.7227         |

#### Cubic Method

| Error Type       | Error | Num. | Denom. | $||\nu_2||$ | $||Y_1Y_3||$ | $||d||$ | $\theta d\nu_2$ | $\theta dY_1Y_3$ |
|------------------|-------|------|--------|-------------|-------------|--------|----------------|-----------------|
| Global Regular   | 0.1647| 0.0127| 1.8344 | 0.0153      | 0.0303      | 2.5361 | 75.7809        | 63.8826         |
| Global Random    | 0.1990| 0.0152| 1.7970 | 0.0160      | 0.0304      | 2.8937 | 77.9822        | 70.3364         |
| Local Equi-angle | 0.0906| 0.0087| 1.7875 | 0.0149      | 0.0298      | 2.8991 | 84.8990        | 66.5182         |
| Local 30-90      | 0.1131| 0.0102| 1.7881 | 0.0157      | 0.0315      | 2.8181 | 83.1192        | 70.1950         |
| Local Random 1   | 0.1311| 0.0133| 1.8100 | 0.0133      | 0.0306      | 2.6616 | 80.1992        | 67.0239         |
| Local Random 2   | 0.1340| 0.0120| 1.7754 | 0.0136      | 0.0312      | 2.7878 | 81.2036        | 68.9842         |
| Local Random 3   | 0.1328| 0.0133| 1.8103 | 0.0133      | 0.0305      | 2.6916 | 80.2445        | 66.7793         |
| Local Random 4   | 0.1267| 0.0115| 1.7982 | 0.0155      | 0.0310      | 2.8677 | 82.4222        | 68.9718         |

### REFERENCES


