5
Minimum Variation Networks

Many surface modeling and data interpolation schemes use a mesh or network of curves as a key component in the construction of a smooth surface [126]. The minimum variation network (MVN) is a G^2 network composed of fair curves that provides an excellent frame on which to build a smoothly curved surface. The MVN is computed using the MVC functional (4.2) and MVC optimization techniques with curve specifications and G^2 curve continuity constructions replaced by surface specifications and G^2 surface continuity constructions. This chapter describes the calculation of minimum variation networks.

5.1 MVN Representation and Continuity

MVN are constructed to exactly match second order geometric specifications at the vertices of the network. Minimally, a network may be specified by vertex positions. A network specification may also include surface normal and curvature information to be interpolated. Finally, it is possible to specify the tangent direction at the end points of individual curves in the network. The network is constrained to meet at its vertices such that at each vertex a second order continuous surface containing the curves incident to the vertex can be constructed. The specification permits the assignment of a fixed or initial geometric characterization to each vertex of the network. This geometric description consists of the vertex position, and optionally, tangent plane, and principal directions and principal curvatures. In addition to the geometric specification, a list of incident curves is also provided. The curves making up the network are defined by pairs of vertices. It is also possible to fix and/or initialize the tangent directions associated with the end points of each curve. Last, in the case when the resulting MVN is to be input to our surface optimization system, patches are also identified by the vertices at their corners, and curves around their perimeter.

Internal to the optimization, the network of curves is defined via the second order parameters of a surface description at each vertex of the network and via a description of how each curve segment emerges from within the osculating paraboloid specified at its endpoints. Each such paraboloid is defined by the vertex position $\vec{p}$, a pair of conjugate directions, $\hat{w}_1, \hat{w}_2$, and the normal curvatures in those directions $\kappa_{w_1}, \kappa_{w_2}$. Conjugate directions are equivalent to principal directions in that, coupled with the associated curvatures, they fully characterize the curvature of a surface at a point [52]. Conjugate directions are more amenable to optimization because they do not have to be constrained to mutual orthogonality. The network is represented by quintic Hermite curves. These curves are defined by the positions and the first two parametric derivatives at their endpoints: $\vec{P}, \vec{P}', \vec{P}''$. Each curve in the network is defined by the position $\vec{p}$, tangent direction $\hat{t}$, and three scalar parameters, m, α, c, at each endpoint. The mapping from these values to the parameters defining the corresponding Hermite curve is

$$\vec{P} = \vec{p} \qquad \vec{P}' = m^2\hat{t} \qquad \vec{P}'' = m^4\vec{\kappa} + \alpha m^2\hat{t}$$

$$\vec{\kappa} = \kappa_n\hat{n} + c\hat{b} \qquad \hat{b} = \hat{n} \times \hat{t}$$

$$\kappa_n = \hat{t} \cdot \left[K\right] \cdot \hat{t}$$

$$\left[K\right] = \begin{bmatrix} \hat{w}_1 \\ \hat{w}_2 \\ \hat{n} \end{bmatrix}^{-1} \cdot \begin{bmatrix} \kappa_{w_1} & 0 & 0 \\ 0 & \kappa_{w_2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \left(\begin{bmatrix} \hat{w}_1 \\ \hat{w}_2 \\ \hat{n} \end{bmatrix}^{-1} \right)^T \tag{5.1}$$

Note that the curvature of the curve $\vec{\kappa}$ is the sum of two orthogonal components; κ_n, the component in the normal direction is a function of the surface curvature at the vertex and the tangent direction of the curve at its end point; c, the component in the binormal direction is independent of the surface curvature at the vertex and represents the curvature of the curve "within the surface."

During the optimization process, those variables not fixed by constraints are iteratively adjusted to minimize the MVC functional (4.2) summed over all curves of the network. At each iteration step, $\hat{w}_1$ and $\hat{w}_2$ are renormalized, and $\hat{t}$ is projected onto the plane spanned by $\hat{w}_1, \hat{w}_2$ and also renormalized. It is this normalization step in combination with the construction outlined in equation (5.1) that guarantees G^2 continuity is maintained.

5.2 Network Initialization

The curve network must be initialized to some reasonable shape before optimization can proceed. Because the starting point of an optimization strongly affects the rate of convergence, it is important that the starting point actually be near the optimal network. At each interpolation point a vertex normal vector is first initialized, then the tangent vectors of the incident curves are computed, next the principal directions and curvatures are defined, and finally each curve's scalar coefficients are initialized. The initialization techniques used are similar to the techniques used for G^1 network construction described in section 3.2. In order to initialize the second order parameters of the vertices, we have developed an new method for estimating the principal curvatures of a polyhedral structure.

The vertex normal is initialized as an average of the incident face normals weighted inversely proportional to the area of the incident face, i.e. the smaller the face the greater its influence on the vertex normal [28]. This technique contrasts with that of Shirman and Séquin and is based on the fact that as the area of a facet decreases it becomes a better and better local approximation of the surface interpolating its vertices. It is a conceptual equivalent to the weighting scheme described in section 4.9. The tangent vectors of curves incident to a vertex are set to the direction of the incident chords projected onto the plane defined by the vertex position $\vec{p}_i$ and the normal $\hat{n}$ (Fig. 5.1).

Once vertex normal vectors and incident tangent directions have been computed, the principal curvatures and principal directions at a vertex are calculated. Both Calladine [28] and Todd and McLeod [204] describe approaches for estimating the curvature of polyhedral surfaces. Calladine derives a formula for the Gaussian curvature at a vertex in a mesh:

$$\text{Gaussian curvature} = \frac{\text{angular defect at a vertex}}{\text{area associated with the vertex}}.$$

Where the angular defect is defined as $2\pi -$ the sum of the interior angles of the faces meeting at the vertex. The area associated with the vertex is 1/3 of the area of the triangles meeting at the vertex. This method only provides an estimate of Gaussian curvature and is inadequate in our application. Todd and McLeod compute a least squares fit to the Dupin indicatrix[1] [52:149] and require that a pairing be established among the vertices neighboring a vertex; this is not possible at vertices of odd order. At even order vertices, it remains problematic since the results vary greatly depending on the pairing chosen; logically opposite curves are not always appropriate partners.

Our approach uses a least squares fit of sample tangent directions and normal curvatures to compute the principal directions and curvatures. The initialization of these values is very important to the speed of convergence. First consider the situation shown in Figure 5.1. A vertex is shown with a number of incident edges. For each edge we calculate the curvature implied by that edge emanating from the vertex. Starting with edge $\vec{p}_i, \vec{p}_n$ we reflect $\vec{p}_n$ through the normal and fit a circle through $\vec{p}'_n, \vec{p}_i$ and $\vec{p}_n$. The radius of the resulting circle is the radius of curvature (Fig. 5.2). The reciprocal of this radius of curvature is the

[1] The Dupin indicatrix is an alternative representation for surface curvature $\kappa_1\xi^2 + \kappa_2\eta^2 = \pm 1$, where (ξ, η) are cartesian coordinates in the orthonormal basis $\{\hat{e}_1, \hat{e}_2\}$.

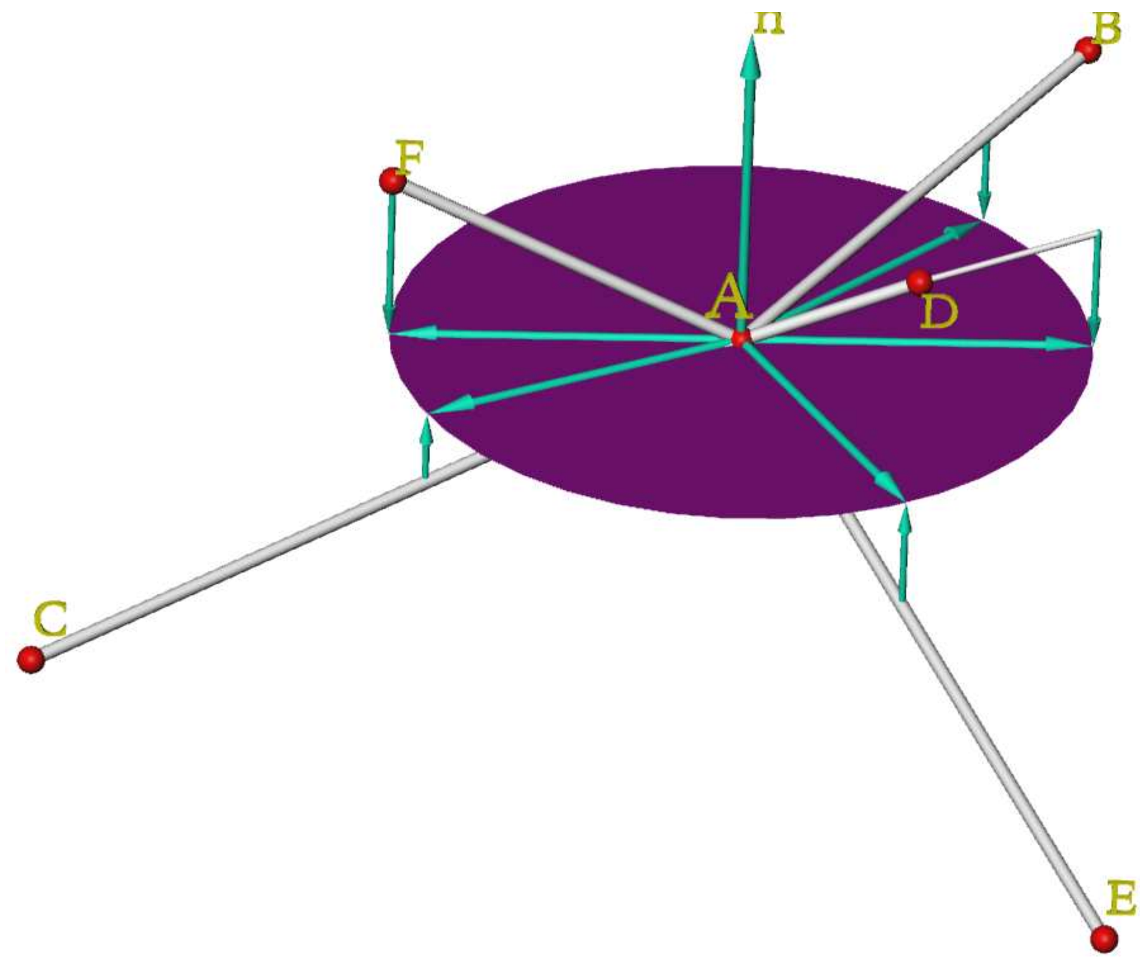

Figure 5.1. Tangent initialization.
The projection of incident chords onto the plane defined by the point and normal.

approximate normal curvature in the direction of $\hat{t}_n$. Repeating this procedure for each of the incident edges provides a set of sample tangent directions and approximate normal curvatures (Fig. 5.3). The set of tangent directions and approximate normal curvatures is used to compute a least squares fit for the principal directions $\hat{e}_1, \hat{e}_2$ and principal curvatures κ_1, κ_2 of the surface at the vertex as follows. We start with the expression for normal curvature expressed with respect to any convenient orthonormal basis in the plane defined by the normal,

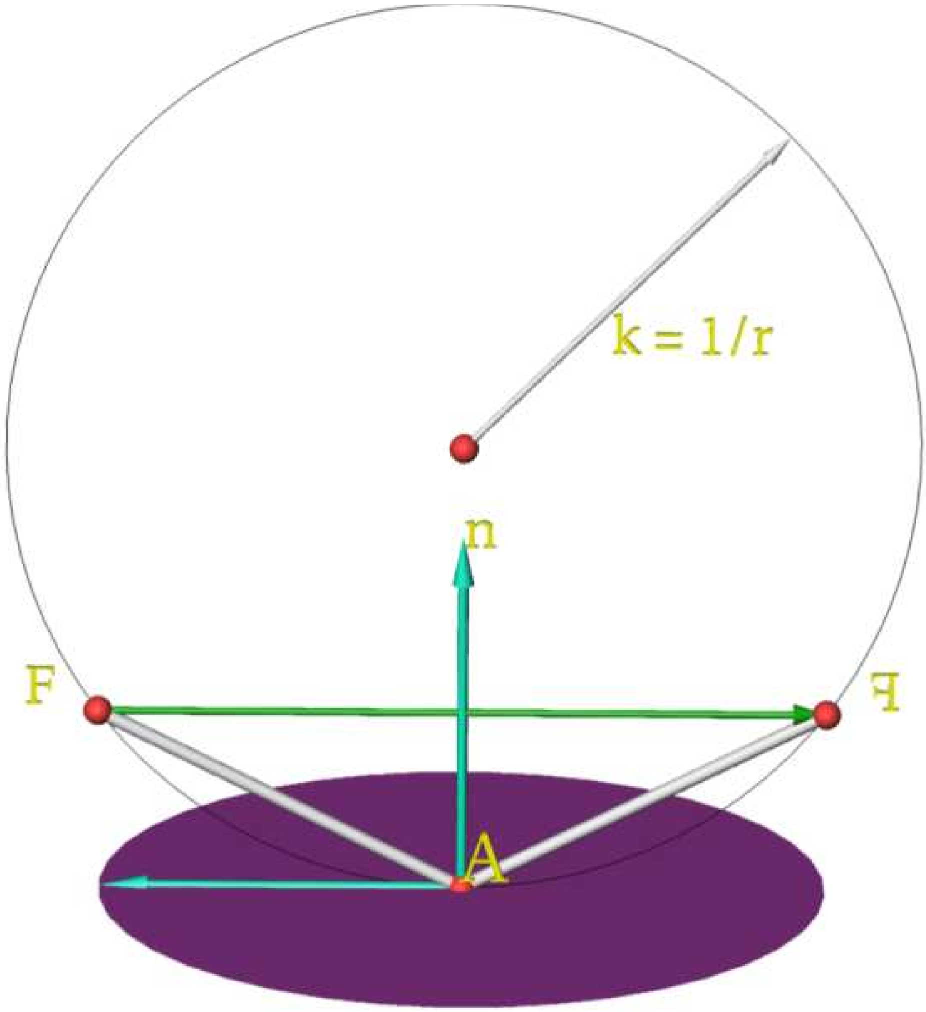

Figure 5.2. An Approximate Radius of Curvature.

The approximate radius of curvature in the direction of $\hat{t}_n$ is calculated by reflecting the chord through the normal and fitting an osculating circle through the vertex and through the end points of a chord and its image.

$$\kappa_n = \hat{t} \cdot [K] \cdot \hat{t}$$

$$[K] = \begin{bmatrix} \hat{e}_{1,x} & \hat{e}_{1,y} \\ -\hat{e}_{1,y} & \hat{e}_{1,x} \end{bmatrix}^{-1} \cdot \begin{bmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{bmatrix} \cdot \left(\begin{bmatrix} \hat{e}_{1,x} & \hat{e}_{1,y} \\ -\hat{e}_{1,y} & \hat{e}_{1,x} \end{bmatrix}^{-1} \right)^{\mathrm{T}} .$$

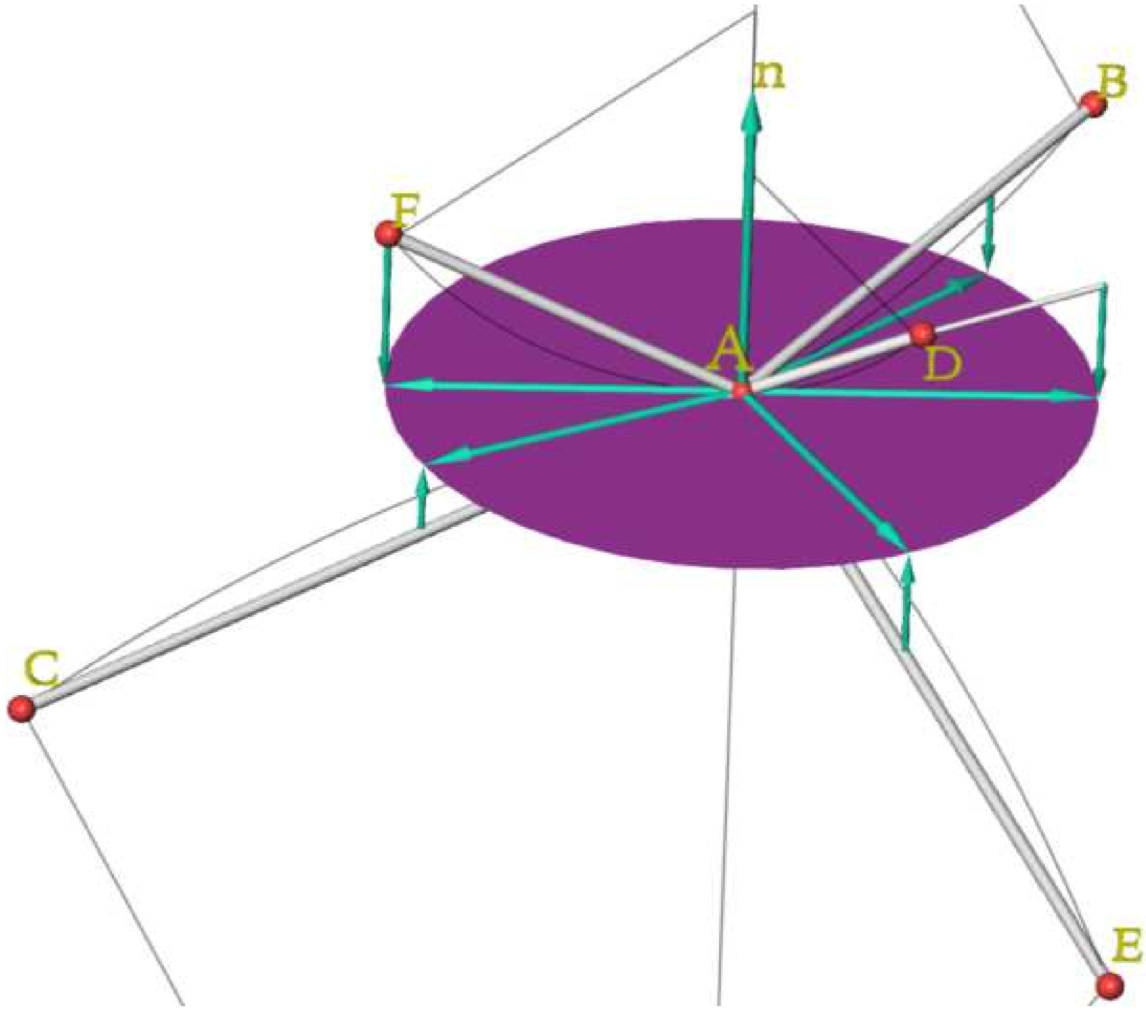

Figure 5.3. A Set of Tangent Directions and Approximate Normal Curvatures.
Using the technique illustrated in Figure 5.2, a set of normal curvatures are displayed as the outlines of "pie wedges". Note that only the wedges associated with vertices *D* and *F* are completely visible.

The normal curvature is found by multiplying the curvature tensor K twice by the desired tangent direction defined relative to the local basis. From this expression we extract the tangent components, to produce an over determined set of $m+1$ linear equations:

$$\begin{bmatrix} \hat{t}_{0,x}^2 & \hat{t}_{0,x}\hat{t}_{0,y} & \hat{t}_{0,y}^2 \\ \hat{t}_{1,x}^2 & \hat{t}_{1,x}\hat{t}_{1,y} & \hat{t}_{1,y}^2 \\ \cdot & \cdot & \cdot \\ \hat{t}_{m,x}^2 & \hat{t}_{m,x}\hat{t}_{m,y} & \hat{t}_{m,y}^2 \end{bmatrix} \cdot \begin{bmatrix} \hat{e}_{1,x}^2\kappa_{\hat{e}_1} + \hat{e}_{1,y}^2\kappa_{\hat{e}_2} \\ 2\hat{e}_{1,x}\hat{e}_{1,y}(\kappa_{\hat{e}_1} - \kappa_{\hat{e}_2}) \\ \hat{e}_{1,x}^2\kappa_{\hat{e}_2} + \hat{e}_{1,y}^2\kappa_{\hat{e}_1} \end{bmatrix} = \begin{bmatrix} \kappa_{n,0} \\ \kappa_{n,1} \\ \cdot \\ \kappa_{n,m} \end{bmatrix} \equiv Ax = b$$

where the unknowns, x, are expressions involving the principal directions and principal curvatures. The general formula for computing the least squares solution to this type of system is $A^{\mathrm{T}}A\bar{x} = A^{\mathrm{T}}b$, where $\bar{x}$ is the least squares solution for "x" in equation . Having solved for x, we have three equations and four unknowns

$$\begin{bmatrix} \hat{e}_{1,x}^2\kappa_1 + \hat{e}_{1,y}^2\kappa_2 \\ 2\hat{e}_{1,x}\hat{e}_{1,y}(\kappa_1 - \kappa_2) \\ \hat{e}_{1,x}^2\kappa_2 + \hat{e}_{1,y}^2\kappa_1 \end{bmatrix} = \begin{bmatrix} \bar{x}_0 \\ \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}.$$

Adding the fact that $\hat{e}_{1,x}^2 + \hat{e}_{1,y}^2 = 1$, allows us to solve for the principal directions and principal curvatures.

Table 5.1 provides the results of a least squares fit to the sample directions and curvatures shown in Figure 5.3. The resulting principal directions and curvatures are

$$\begin{aligned} \kappa_1 &= 0.9548 & \hat{e}_1 &= \{0.1266, 0.0, -0.9919\} \\ \kappa_2 &= -0.6647 & \hat{e}_2 &= \{0.9919, 0.0, 0.1266\} \end{aligned} . \tag{5.2}$$

A set of principal directions and principal curvatures locally define the geometry of a surface. Given principal directions and curvatures the normal curvature in a given tangent $\hat{t}$ direction is computed:

$$\kappa_n = \kappa_1(\hat{e}_1 \cdot \hat{t})^2 + \kappa_2(e_2 \cdot \hat{t})^2. \tag{5.3}$$

We illustrate our example least squares solution in two ways. First, in Figure 5.4 with a surface (①) swept out by the osculating circles as rotated about the normal, changing radius according to (5.3) with curvature values (5.2); the arcs of the approximate osculating circles are also shown. Second, in Figure 5.5 we plot normal curvature versus tangent direction. Vertical lines are drawn connecting the approximate normal curvatures with the least squares fit curvatures.

To complete the initialization of the network, the scalars associated with each curve are set as follows: m, the magnitude of the first derivative, is set to the chord length; α, which relates the first and second derivatives, is set to zero so that the derivatives are orthogonal and the curves are arc length parameterized at their end points; c is set to zero so that the curve's only curvature component is the normal curvature in the tangent direction.

Vertex ID	$\vec{P}$	Approximate Radius of Curvature	Approximate Normal Curvature	Least Squares Fit Normal Curvature
B	{1.0, 0.4, 1.0}	2.7	0.370	-0.0584
C	{-1.0, -0.5, -2.0}	-5.25	-0.190	0.452
D	{0.0, 0.2, 0.5}	0.725	1.379	0.929
E	{-2.0, -0.75, 1.0}	-3.70833	-0.269	-0.162
F	{0.0, 0.5, -1.0}	1.25	0.8	0.928

Table 5.1. Least Squares Fit Curvature Example.

This table lists the inputs and solutions to finding the least squares fit curvature for the arrangement of vertices shown in Figure 5.4. Vertex *A* is located at the origin, and the surface normal is along the *y*-axis.

5.3 Optional Network Constraints

Since the quality of the network directly impacts the quality of the resulting surface, we present an optional heuristic constraint. A very successful method for improving network quality is to force suitable opposing pairs of curve segments incident to a common vertex of the network to join with G^2 continuity; this is an extension of Shirman and Séquin's opposite edge method [200]. Pairs of curves are made G^1 continuous by forcing them to share tangent vectors. G^2 continuity is imposed by forcing the curves to also share the binormal component, c, from (5.1). Curves may be paired up automatically at even order vertices. At odd order vertices and at vertices where custom or partial pairings are desired, matches may be specified manually.

During initialization, shared tangent vectors are set to the average of the individual tangents computed by chord projection. Figure 5.6 illustrates curve continuity applied to 16 regularly spaced points on the surface of a torus. Figure 5.6② illustrates the improvement to the network when G^2 continuity is imposed. In a second example, we illustrate a case where there is an appreciable difference between a network of curves that are matched up with G^1 continuity versus a network of curves where paired up curves are

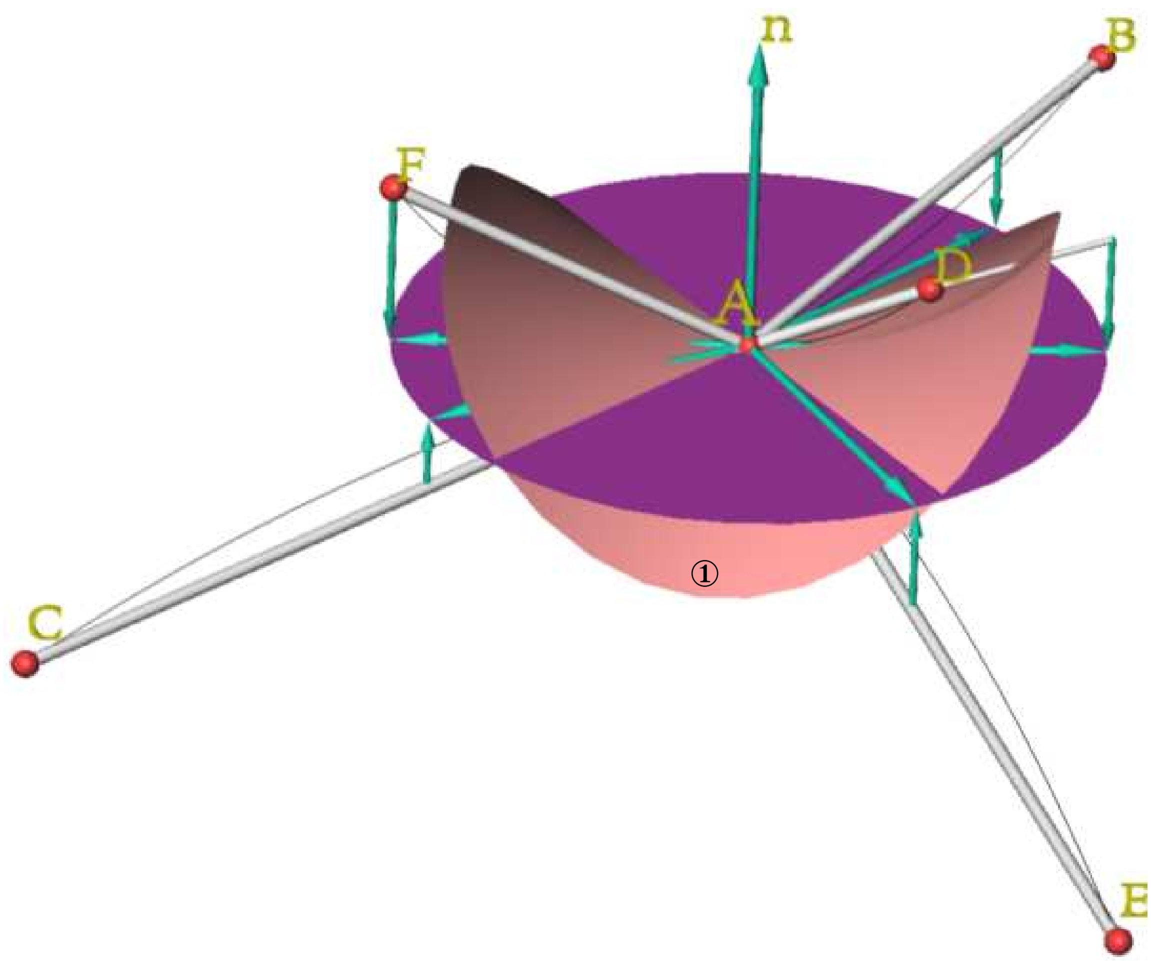

Figure 5.4. A Least Squares Curvature Solution.

Here we see the surface (①, osculating paraboloid) swept out by the osculating circles rotated about the surface normal, and arcs of the sample osculating circles of which it is a least squares fit.

constrained to meet with G^2 continuity. In this case the network passes through 12 points on a symmetrical S-shaped tube. The network specification includes position, surface normal and curvature as well as curve tangent constraints.

In Figure 5.7① we see the curves of the network meeting with G^1 continuity; in the lower half of the figure the curves running lengthwise along the tube have curvature profiles attached, highlighting the distribution of curvature. Note that in Figure 5.7② the curves

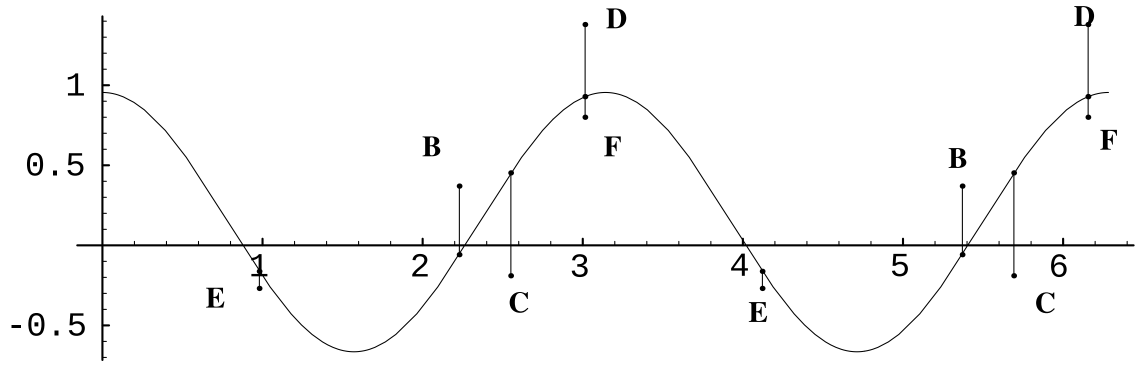

Figure 5.5. Normal Curvature vs. Tangent Direction.

The approximate normal curvature associated with each vertex is connected to the plot by a vertical line.

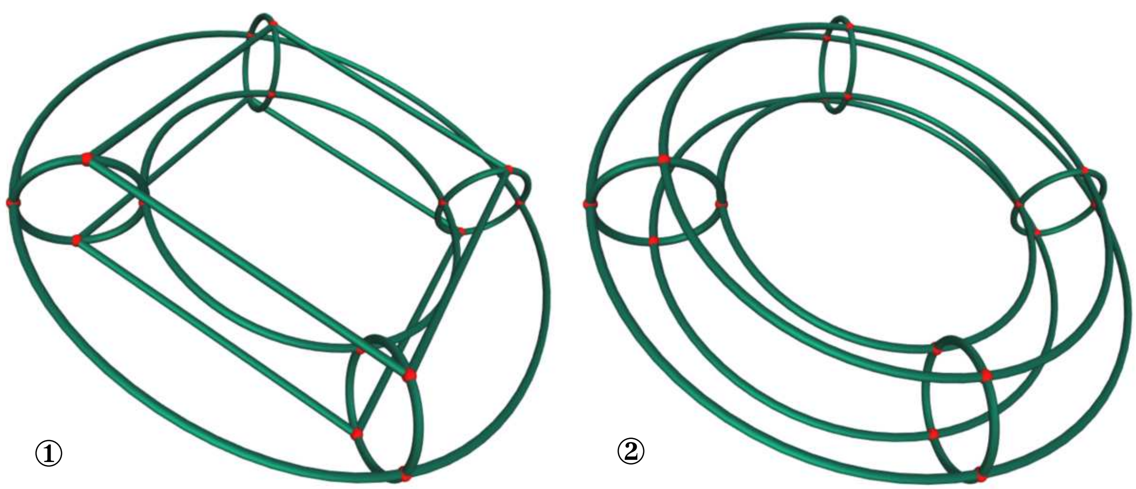

Figure 5.6. Optional Network Continuity Constraints, G^0 versus G^2.

A network through points on a torus ① with G^0 and ② with G^2 continuous curves through vertices.

are G^2 continuous. This latter G^2 continuous network is of superior quality, in part, because some of the curves of the network are naturally G^2 continuous and those that are only G^1 continuous create asymmetry unless constrained to G^2 continuity.

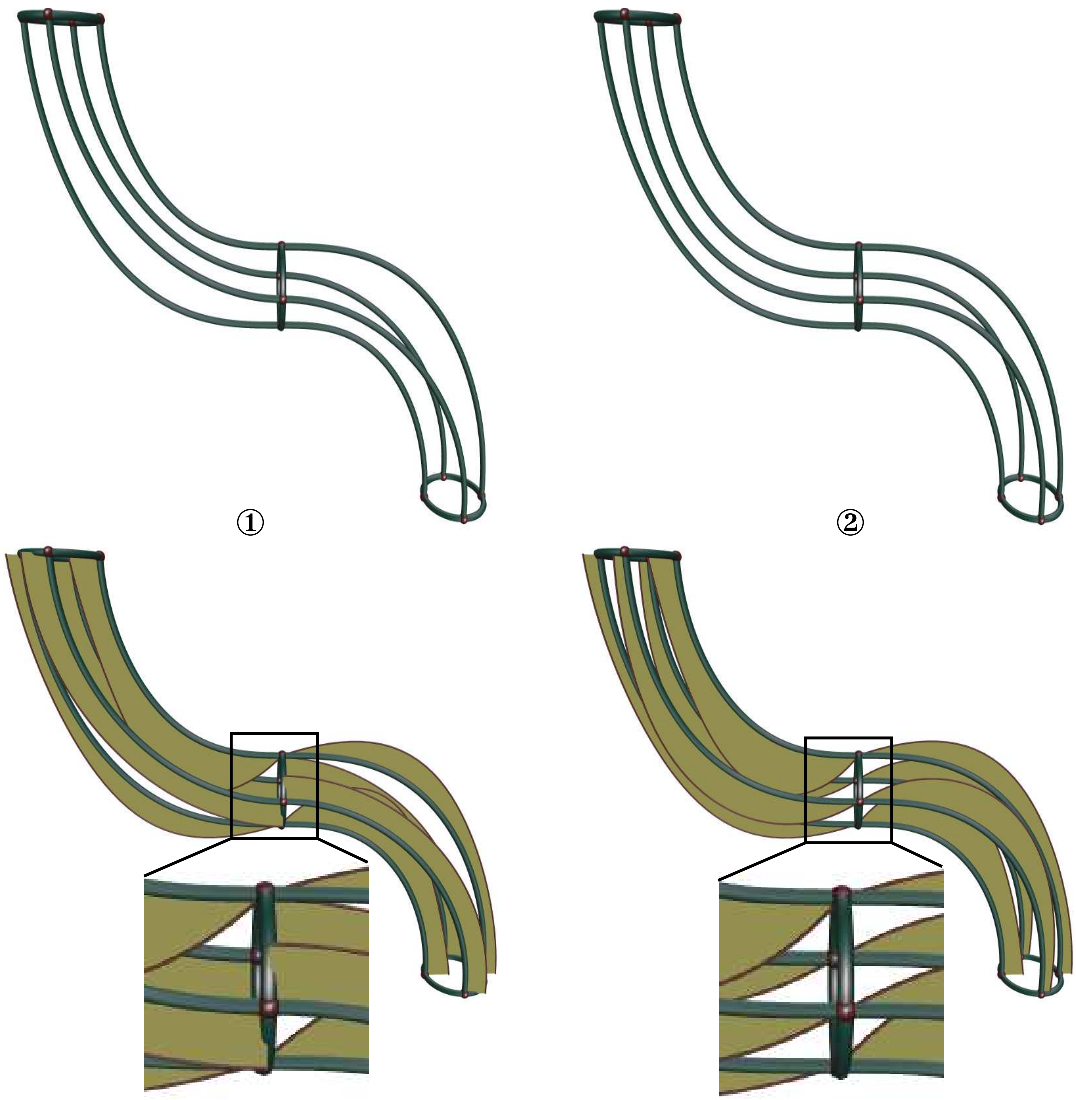

Figure 5.7. Optional Network Continuity Constraints, G^1 versus G^2.
A network through points on an S-shaped tube ① with G^1 and ② with G^2 continuous curves through vertices.

5.4 Comparisons

To assess the quality of MVN, we compare them with other methods and examine the distribution of curvature along the network. Since there are no other published methods for creating G^2 networks of curves, we compare MVN with the networks resulting from Shirman's method [198] and from that of Mann et al [126]; these methods are reviewed in section 3.2. The latter approach is an adaptation of deBoor-Höllig-Sabin [44], thus we use the initials “DBHS” to identify these networks. For our comparisons, we use the data sets from the study of Mann et al. [126] (Fig. 5.8); the corresponding input files are provided in appendix B. These data sets are samplings of existing surfaces: both *octahedron* and *sphere6* take their data from the surface of a sphere, *capsule* is a cylinder with hemispherical caps, *franke4* is a sampling of the Franke function [84]

$$z = \frac{1}{3} e^{\left(-\frac{81}{16}\right)\left[(x-0.5)^2 + (y-0.5)^2\right]},$$

and *torus* takes its data from the surface of a torus with major and minor radii 1.5 and 0.5 respectively. The networks calculated by the three methods in the comparison are shown in figures 5.9-5.12. In the figures the networks are rendered with a curvature fin attached to their component curves. The fin follows the curve normal and its width is proportional to the curvature of the curve at the point where it is attached. This is the same method as was used in the evaluation of MVC in section 4.11. In terms of the subjective quality of the network, the Shirman-Séquin approach is outperformed by the other two methods in all cases. Examining *octahedron* and *sphere6*, we see a subtle difference between the

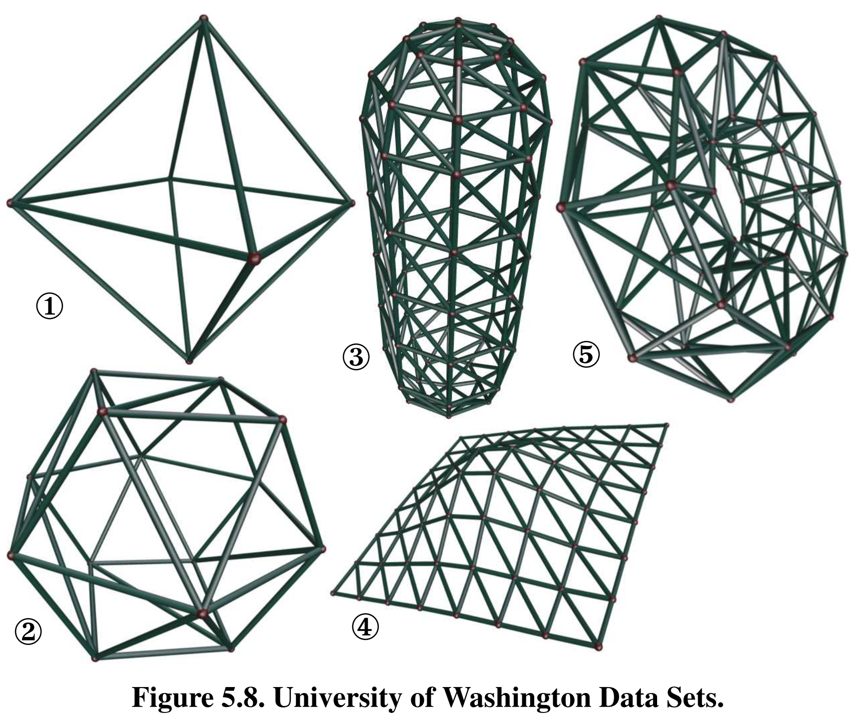

Figure 5.8. University of Washington Data Sets.
① *octahedron,* ② *sphere6,* ③ *capsule,* ④ *franke4*, ⑤ *torus.*

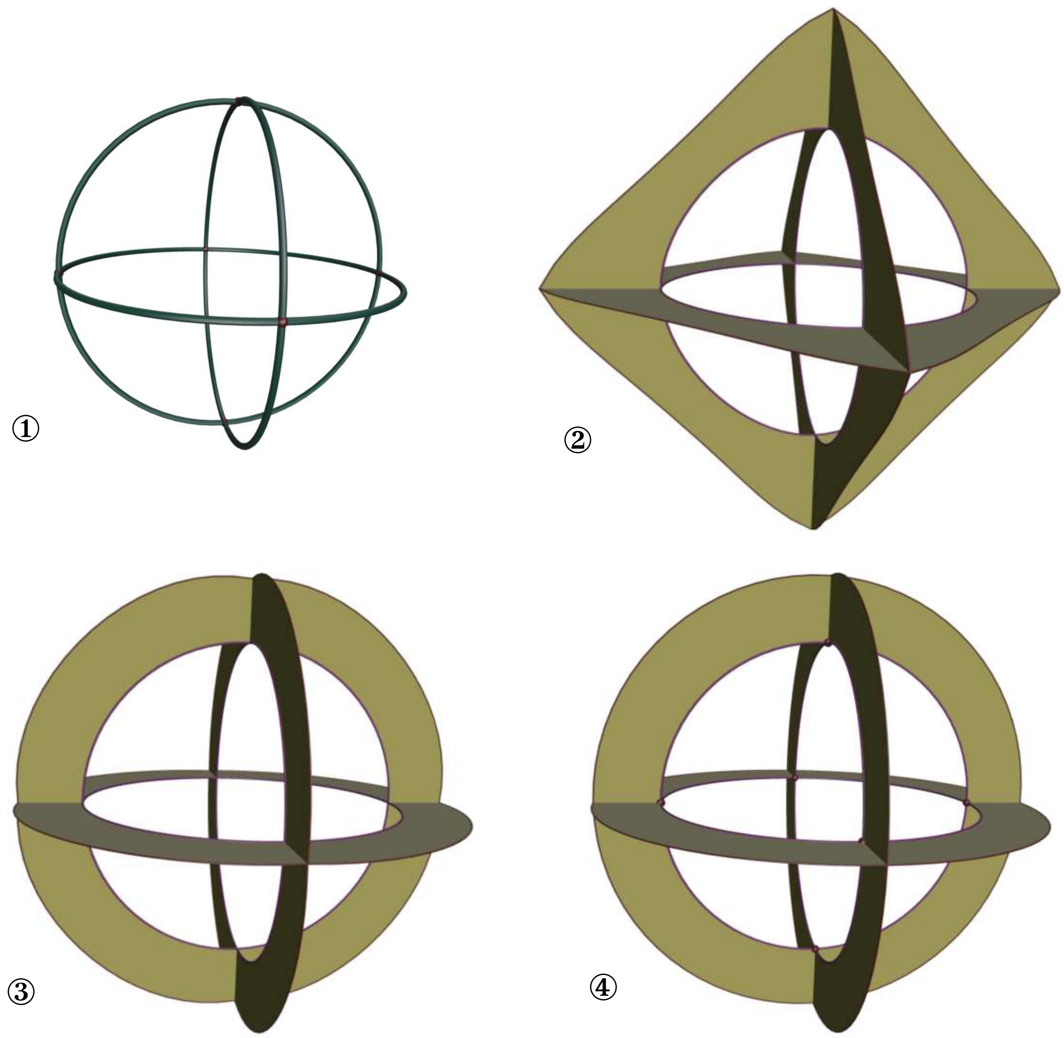

Figure 5.9. ***Octahedron.***

① Minimum variation network. ②,③, and ④ are drawn with curvature profiles. ② Shirman-Séquin. ③ DBHS. ④ MVN.

curvature distribution of the MVN and DBHS, the latter network has slightly lower curvature near the interpolation points. Examining *Franke4* and finally *torus*, we see a marked difference in the respective networks. The MVN *torus* is very regular, with smoothly varying curvature. The DBHS *torus* has large discontinuities when passing through interpolation points, and curvature is not distributed evenly along the arcs of the

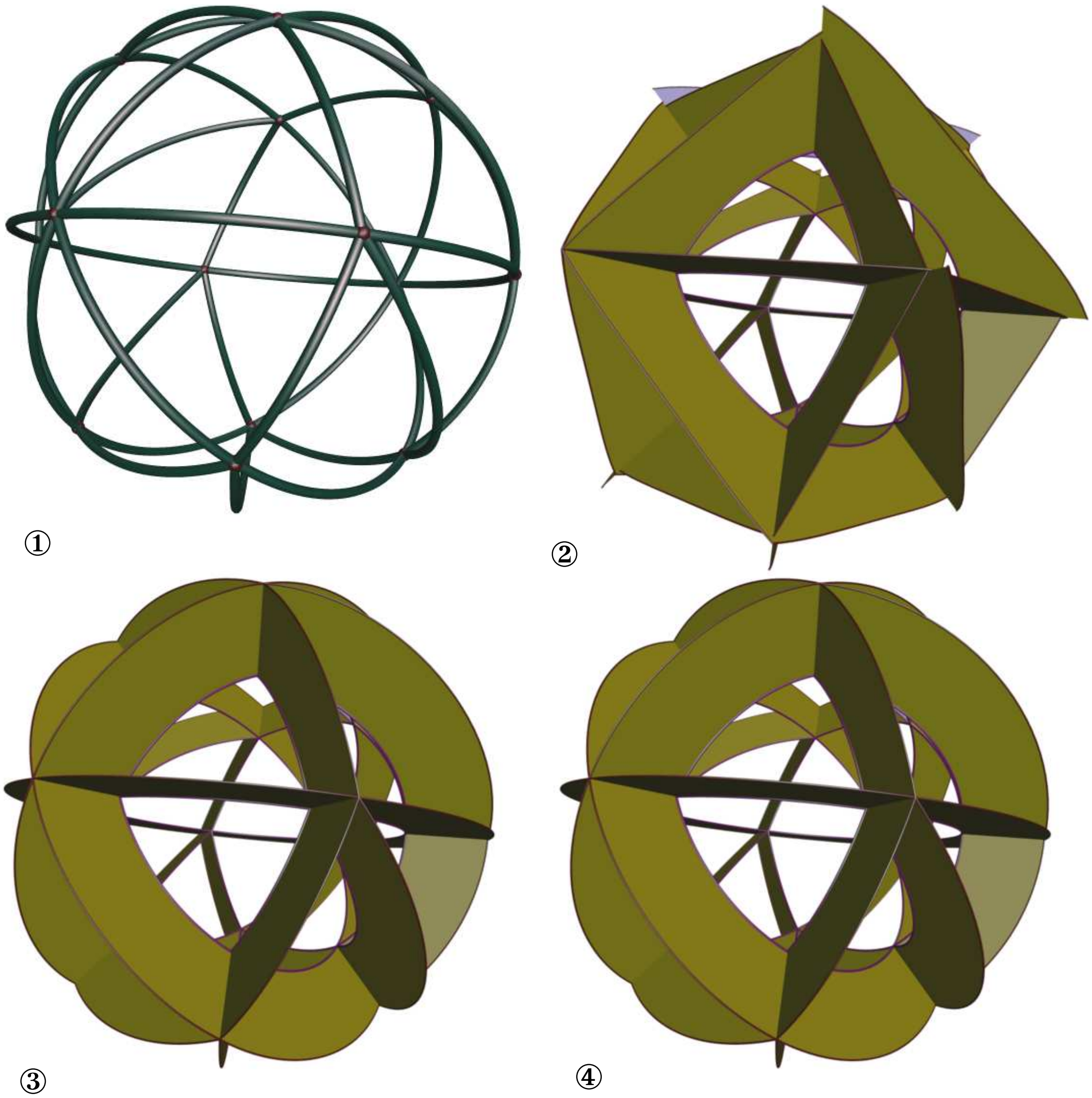

Figure 5.10. ***Sphere6*.**

① Minimum variation network. ②,③, and ④ are drawn with curvature profiles. ② Shirman-Séquin. ③ DBHS. ④ MVN.

network. Much of the regularity and all of the arc-to-arc continuity of the MVN may be attributed to imposing G^2 continuity among the arcs of the network. The superior distribution of curvature is due to the curvature variation functional.

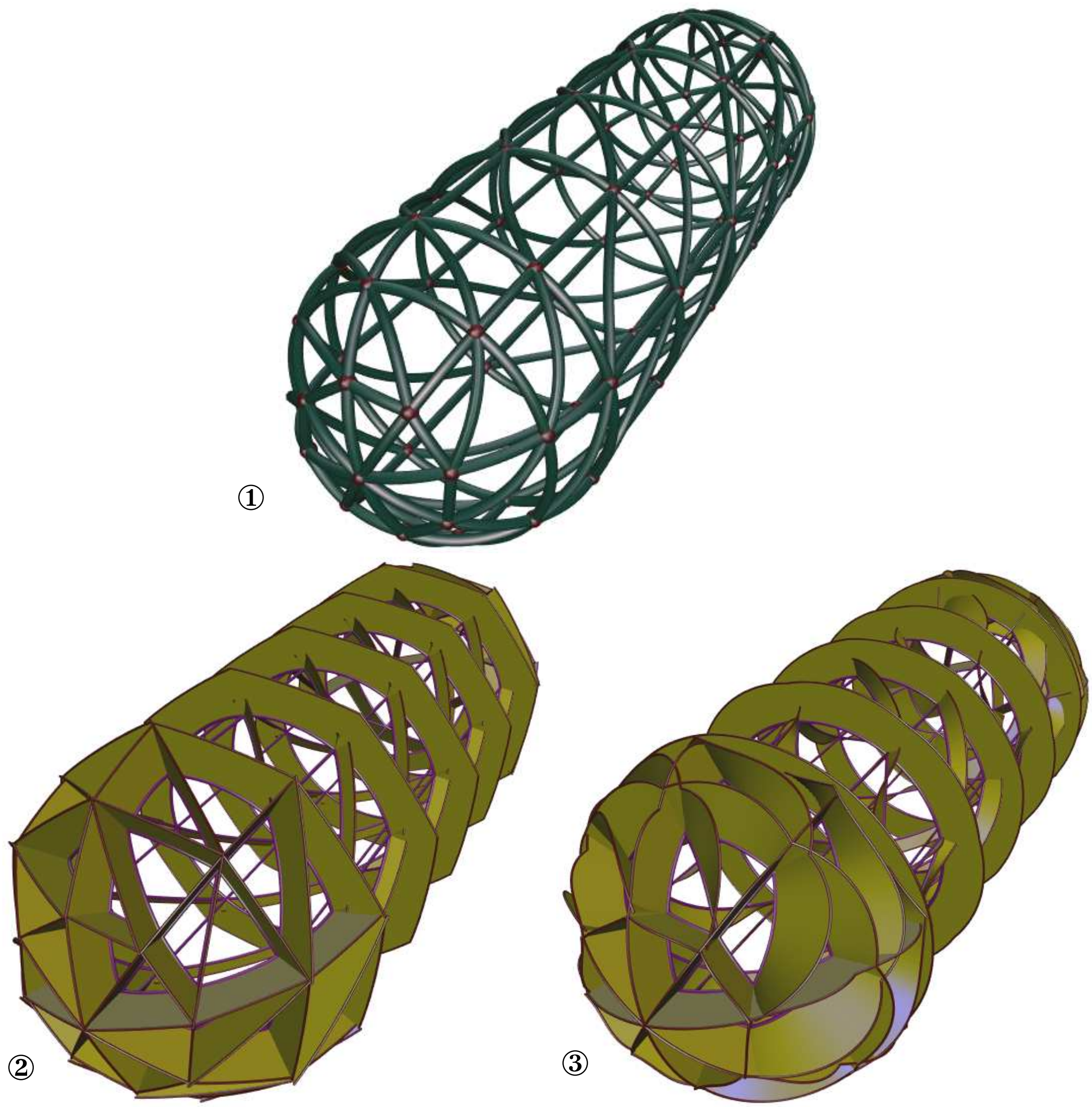

Figure 5.11. ***Capsule*.**

① Minimum variation network. ②, and ③ are drawn with curvature profiles. ② Shirman-Séquin. ③ MVN. The DBHS example is not included because there is no consistent curvature info where the cylinder and hemisphere meet.

Next we examine the network generated for points on a prismatic tetrahedral frame (Fig. 5.14), also see Figure 6.6 on page 139. We will revisit all of these examples in section 6.5 after the networks have been used to create minimum variation surfaces. As a

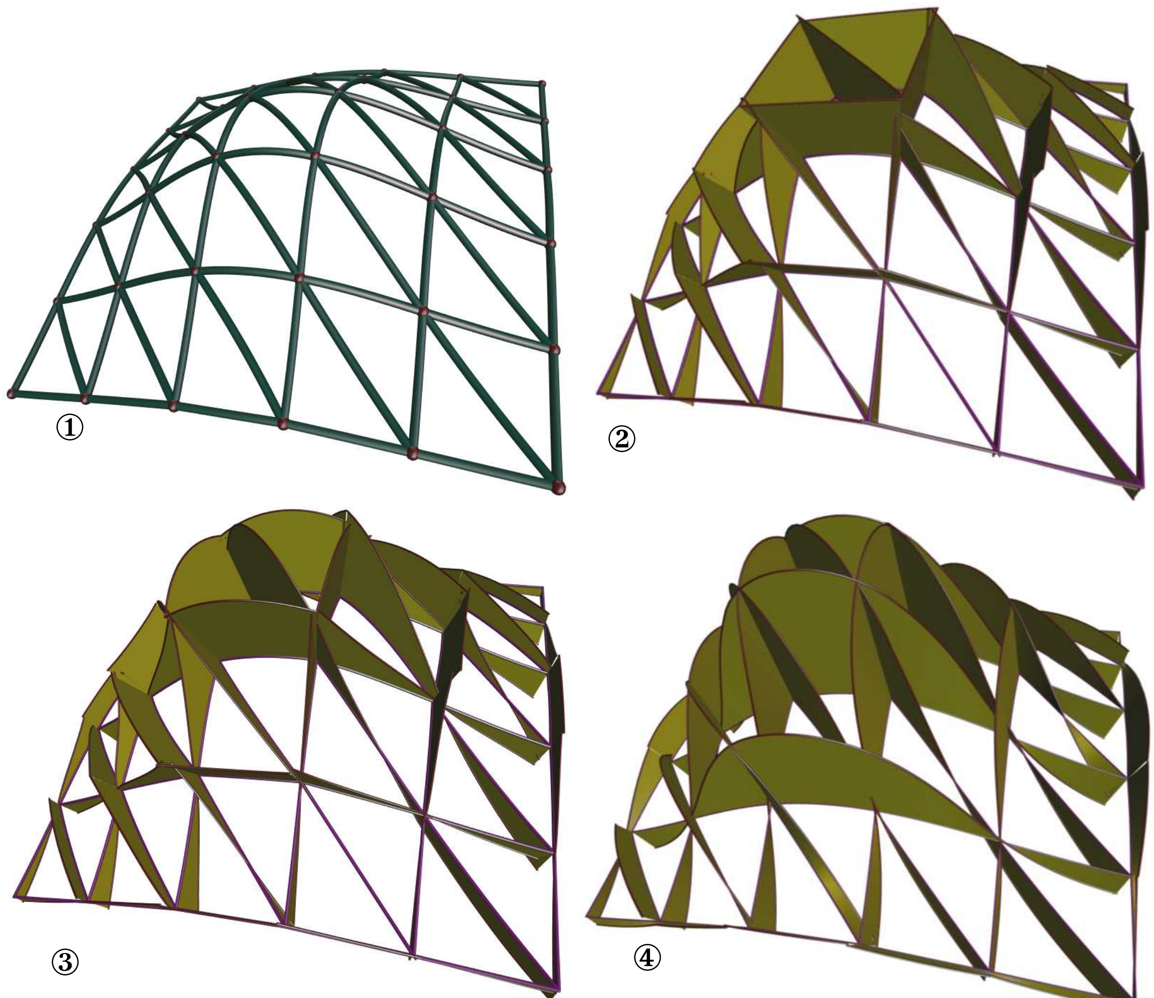

Figure 5.12. ***Franke4.***
① Minimum variation network. ②,③, and ④ are drawn with curvature profiles. ② Shirman-Séquin. ③ DBHS. ④ MVN.

final example we compute the network for a common CAGD blending problem, the joining of two cylinders of differing radii meeting at a right angle (Fig. 5.15).

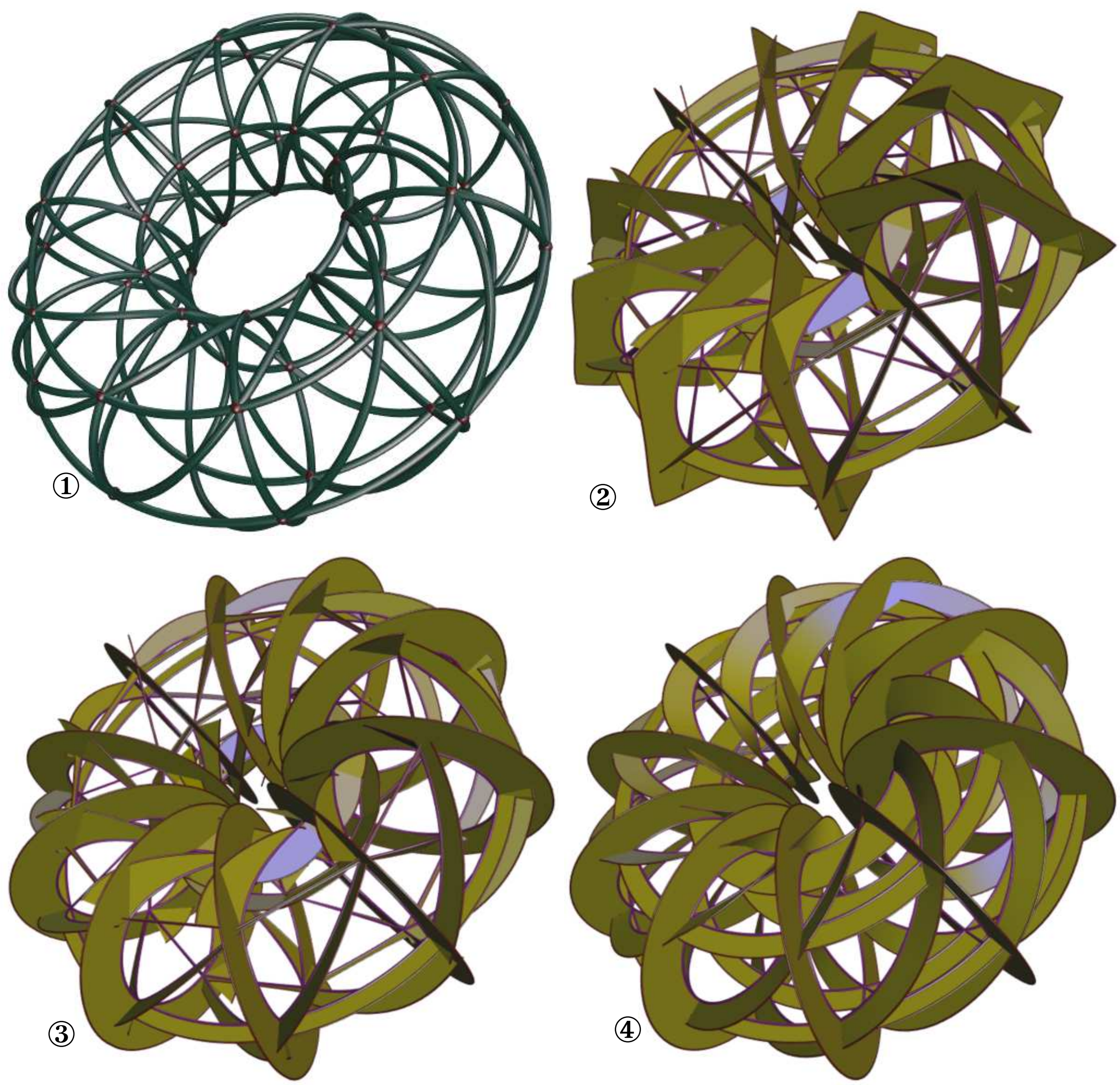

Figure 5.13. ***Torus*.**

① Minimum variation network. ②,③, and ④ are drawn with curvature profiles. ② Shirman-Séquin. ③ DBHS. ④ MVN.

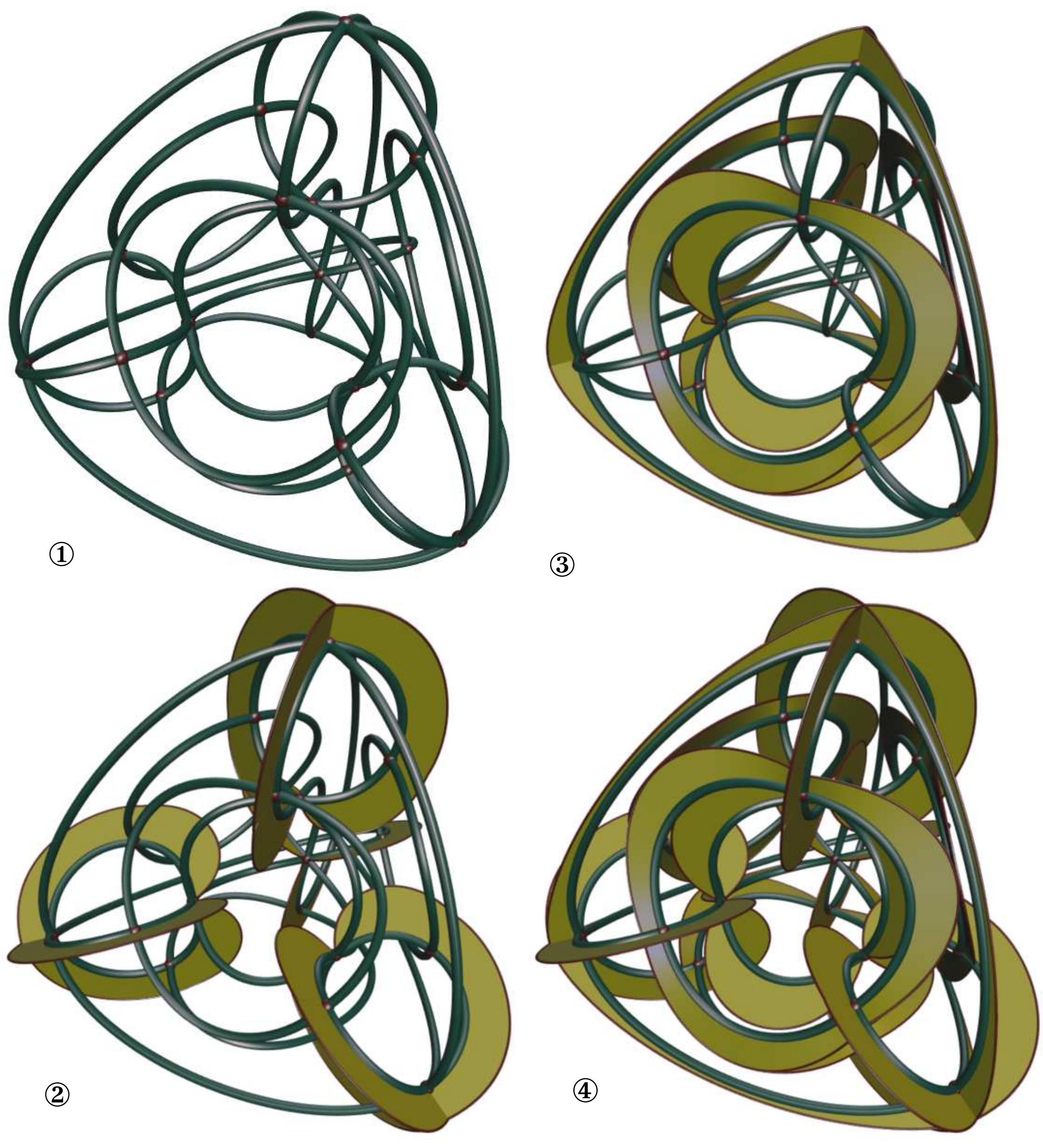

Figure 5.14. ***TetraThing.***

MVN for a prismatic tetrahedral frame. Curves are G^2 continuous through the vertices of the network. The MVN network and interpolation points are shown in ①. The curvature profiles for the separate parametric directions are shown in ② and ③ and combined in ④.

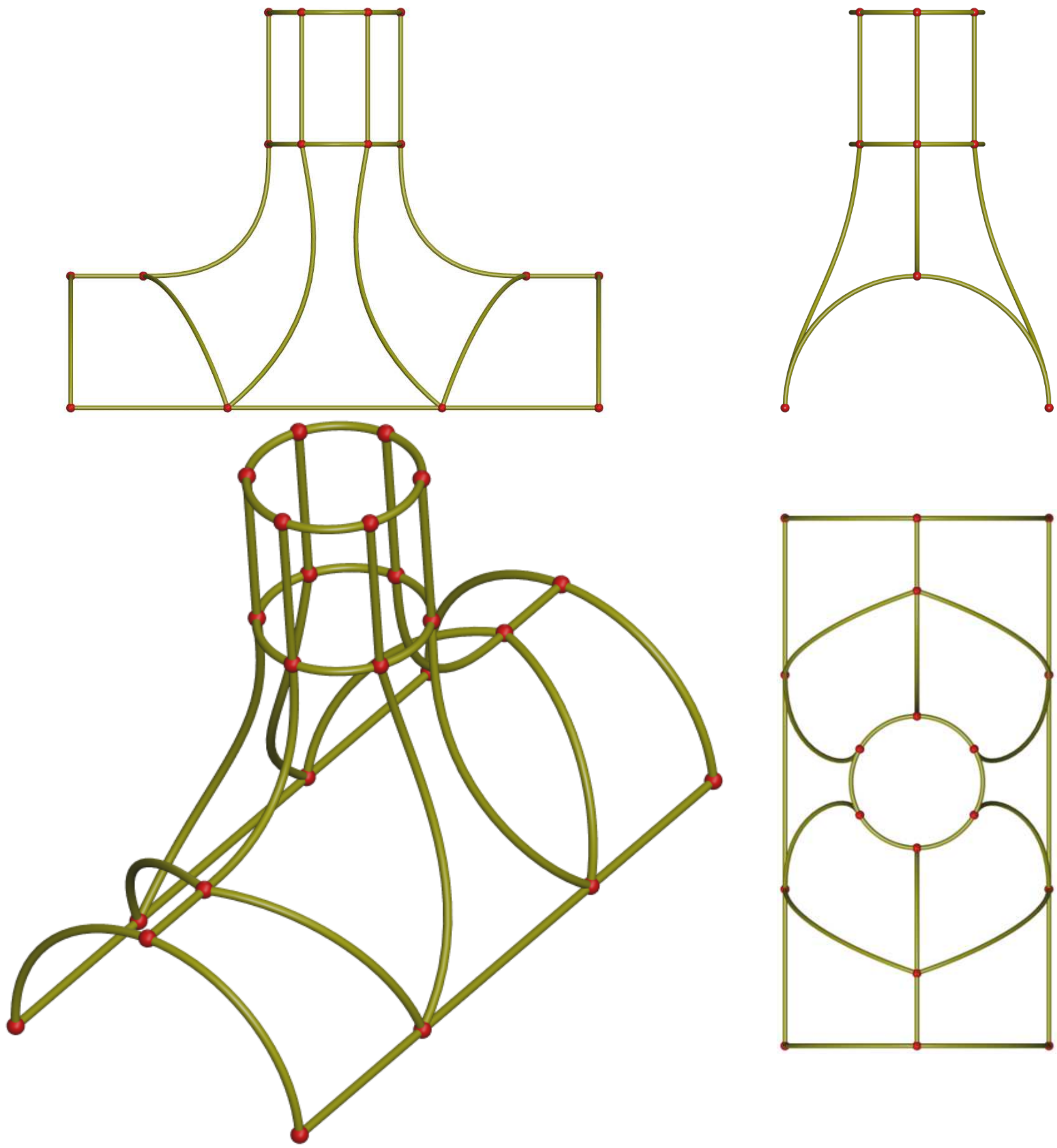

Figure 5.15. Cylinder Blending.

This network is defined by position, tangent, and curvature information at each vertex.