Fitting a Woven-Cloth Model to a Curved Surface: Mapping Algorithms

Masaki Aono* David E. Breen† Michael J. Wozny*

*Design and Manufacturing Institute
Rensselaer Polytechnic Institute
Troy, NY
USA

†European Computer-Industry Research Centre
Munich, Germany

May 22, 1999

Abstract

The problem of fitting a 2D broadcloth composite ply (modelled as a piece of deformable woven cloth) onto a 3D curved surface (modelled with a NURBS surface) is considered. "Fitting" means that a ply is deformed and applied to a surface so that it is everywhere in contact with the surface. A fitting method is presented that incorporates a flexible method for specifying initial conditions, and allows for the simulation of a wide range of fitting configurations.

1 Introduction

Owing to their outstanding mechanical and structural properties, advanced composite materials, such as fiberglass/epoxy, graphite/epoxy and aramid/epoxy, have become increasingly utilized in aircraft and other vehicles in place of aluminum and duralumin [1, 2, 3]. Bidirectional broadcloth (or woven cloth) composites, one of the major composite production forms [3], have been extensively used for reinforcing structural elements in aircraft. They are either supplied as either rolled sheets of plain woven cloth or in impregnated forms, from which "plies" of composites are cut to the required size and orientation. These plies are then pressed and fitted to molds to form 3D composite parts. By "fitting", we mean that the ply is deformed and applied to the mold so that it is everywhere in contact with the surface without gaps and wrinkles.

Currently, many problems and inefficiencies are associated with the design and manufacturing of 3D composite parts. For the fitting process, or the process of determining the 3D ply outlines, there are few design-automation tools, and the manufacturing process is mostly carried out "by hand". Therefore, the design and manufacture of composite parts are time-consuming and inefficient, and the finished products frequently have inconsistent properties. There is, therefore, a great need to create design technologies and tools that assist in the determination of the 3D ply outlines, and that automatically create the 2D "flattened" pattern from them.

In this paper, we deal with the fitting problem of the broadcloth composite plies as a first step toward design automation for 3D composite parts. Since the deformation of a broadcloth composite ply is usually a complex function of the given material of the ply, the given 3D surface shape, and the initial conditions for the fitting, it is not easy to predict the final 3D ply outlines.

To address this problem, we have developed a general framework for a computer-aided lamination system that helps designers to predict a variety of fittings on the basis of different initial conditions, from which they may select a suitable fitting [4]. We focus on the kernel function in the fitting process, i.e. the mapping algorithms. A "mapping algorithm" refers to the technique that creates a correspondence between an arbitrary point in the given 2D broadcloth composite ply and a point on the given 3D curved surface. We present a mapping algorithm that is more flexible and practical than previous methods [5, 6, 7, 8, 9].

We have modelled a 2D broadcloth composite ply as a set of inextensible vertical and horizontal threads, and a 3D curved surface as a nonuniform rational B-spline (NURBS) surface [10]. With the assumption of thread inextensibility, the ply is mainly deformed by the changing of the angle (called the "shear angle") between vertical and horizontal threads. We also assume that the threads are straight between crossings, and that no slippage occurs at the crossings. This simplifies the mapping calculation. In other words, the problem of fitting a 2D broadcloth composite ply to a 3D surface is reduced

to a set of surface-surface-intersection calculations [11, 12, 13].

In the following sections, we first review related work, and then describe our mapping algorithm, including a flexible method for specifying initial conditions, the mapping calculations as a set of intersection problems, and a scanning algorithm that determines the order of the mapping calculations. Finally, we present two examples, and emphasize how significantly the initial conditions affect the resulting fittings. We also describe several graphical tools via the examples, including a 3D plot of the fitting configuration, a 3D plot of the shear-angle distribution, and a 2D plane development. These tools provide insights into the accumulation of deformation energy over the ply, the location of anomalous events, and the final cutting of the ply.

2 Related Work

There have been several research efforts that have considered how to fit a real woven-cloth material onto a specific surface. The first work in this area was conducted by Mack and Taylor [5], who defined several basic assumptions for a model of woven cloth, and investigated the mathematical conditions necessary for fitting a piece of cloth to a surface of revolution. They carried out experiments by fitting their model to a sphere. Their modelling assumptions also included the inextensibility of threads.

Partly because Mack and Taylor gave an intricate formula in terms of a differential equation that held only for a surface of revolution, and partly because they investigated an analytic solution based on the integration of the equation, there has been little progress since the appearance of their paper with this kind of approach. The first computational method for fitting woven cloth to a 3D surface was proposed by Robertson et al. [6, 7]. They reduced the fitting problem to the solving of an intersection problem for two spheres and a quadric by restricting the 3D surface to quadrics. A similar computational method was developed by Bergsma and Huisman [14]. Heisey and Haller [8] first attempted to solve a differential equation numerically that was similar to Mack and Taylor's. None of these methods, however, provide a technique for fitting to an arbitrary curved surface.

Van West et al. [9, 15] were the first to consider the fitting of woven cloth to a fairly arbitrary surface by modeling it as a bicubic Hermite patch. As in Robertson's method, they reduced the fitting calculation to an intersection problem between two spheres and a bicubic Hermite patch.

Prakash et al. [16] address the inefficiencies involved in the manual construction of composite parts, and describe an interactive system called AUTOLAY. Their model of a ply is a set of polygons, and their composite forms are assumed to be unidirectional tapes. Therefore, their method for approximating flat-pattern development can only

be applied to a limited set of 3D surface shapes such as strips, panel and shell-type components, which have shallow curvatures.

3 Model for Broadcloth Composite Plies

We present here our model for broadcloth (woven-cloth) composite plies, which accounts for most of the properties of advanced composite materials such as glass, aramid, graphite, and boron. A broadcloth composite ply can be regarded as a woven fabric consisting of horizontal and vertical threads interwoven in a specific fashion. The ply generally has strong tensile-strain resistance in the thread directions, and a weaker shear-strain resistance [17]. Consequently, a woven cloth composite deforms its shape chiefly by changing the angles between horizontal and vertical threads without elongation or shrinkage of its threads. In consideration of this, we have made the following assumptions for our broadcloth composite plies:

- A broadcloth composite ply is modelled as a piece of woven cloth, in which both the vertical and horizontal threads are inextensible.
- A thread segment between adjacent crossings is straight.
- No slippage occurs at a crossing when the ply is deformed.

The first assumption has proved to be experimentally true as long as we fit the ply onto a surface biaxially [18]. If, however, we stretch the ply uniaxially, thread elongation may not be ignored. Therefore, we implicitly assume that the process of fitting a ply to a surface is carried out without the application of excessive forces in any particular direction, and that it is carried out with biaxial tensile forces that are equal in both directions. The second assumption is reasonable as long as we deal with a smooth surface whose curvature at any point is sufficiently small in comparison with the distance between adjacent crossings. The third assumption has also proved to be experimentally true up to the limit of the shear resistance, which depends on the material properties.

In addition, we implicitly make the following auxiliary assumptions without loss of generality, We first assume that there is no physical difference between "wefts" (horizontal threads) and "warps" (vertical threads). Second, we assume that the weaving of our cloth is given by a "plain weave", as shown in Figure 1 (a). In a plain weave, wefts and warps are alternately interwoven both vertically and horizontally. We also assume that the intersection between a weft and a warp defines a crossing called a "mesh point", as shown in Figure 1 (b). Topologically, each mesh point is linked with four adjacent mesh points with equal distances at the outset. The entire 2D ply is therefore represented by a linked network of mesh points. Below, we introduce "auxiliary" mesh points that are not necessarily at a crossing of two threads.

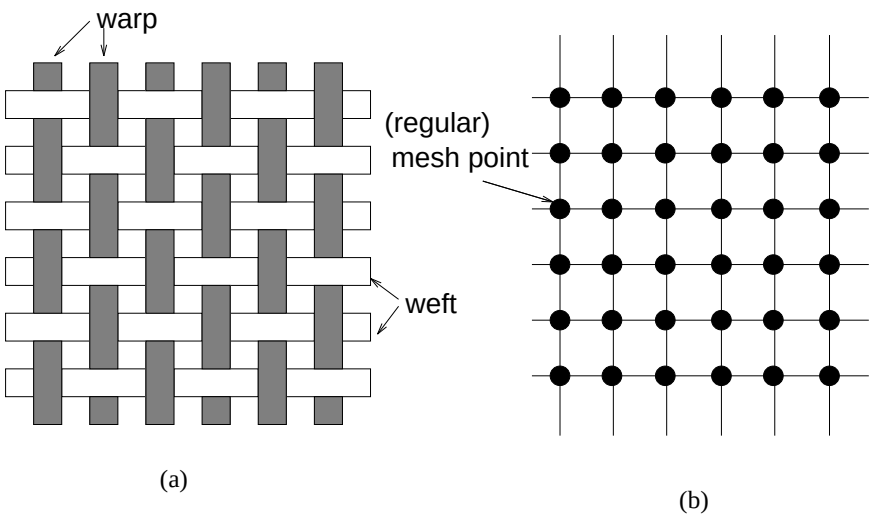

Figure 1: Assumptions; (a) Plain weave, (b) topology of initial mesh points.

4 Methods of Specifying Initial Conditions

Initial conditions must be sufficient, but not excessive, for the mapping between a point on a piece of cloth in 2D space and the corresponding point on a surface in 3D space to be uniquely determined. If, for instance, they are over-specified, it is likely that no solution will satisfy the given initial conditions. If, on the other hand, they are under-specified, a unique solution may not be found. Initial path specifications are very important because the flexibility of the mapping problem is determined by the flexibility of the initial path specification. In this section, we first describe the previously used method for specifying initial conditions for the fitting problem, and then present our new approach.

4.1 Previous Method for Specifying Initial Conditions

Although there are minor differences between the previously used approaches [6, 7, 9, 14], an assumption common to all of them is the specification of initial conditions by the fixing of two yarn paths in both 2D and 3D spaces. Such yarn paths are sometimes called "constrained yarn paths." In 2D space, one weft yarn and one warp yarn are selected as a pair of constrained yarns. They are naturally perpendicular to each other. In 3D space, each constrained yarn path is mapped into a curve that is defined by a sequence of equidistant points on the given surface, as shown in Figure 2. The distance between adjacent points on the 3D surface is assumed to be exactly equal to the distance between adjacent points in 2D space.

By the specification of a pair of constrained yarn paths, both the 2D cloth region and the 3D surface region are generally divided into four regions. The mapping calculation of points between 2D and 3D spaces is carried out independently for each quadrant.

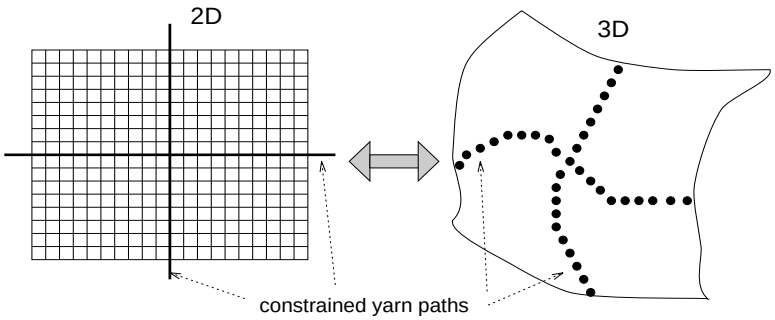

Figure 2: Constrained yarn paths (previously used method).

This previous method has the following advantages: (a) it is easy to calculate the mapping between 2D and 3D points on the initial paths, because the cloth threads are aligned with the paths, and (b) it is straightforward to scan all the mesh points during the mapping calculation, because scanning along warp yarns and weft yarns will produce the same results, owing to symmetry.

The advantages of the method stem from its simplicity in specifying 2D initial conditions. Several problems are, however, associated with the technique for specifying initial conditions.

The first problem is that the method does not allow sufficient flexibility in specifying the constrained yarn paths in 2D space. It only allows the paths to lie on perpendicular warp and weft threads as shown in Figure 2. It does not allow the specification of arbitrary constrained yarn paths where the angle between a path and any thread intersecting the path has a value other than 90 degrees.

The second problem is that the method imposes on users the cumbersome task of calculating a sequence of equidistant points on the 3D surface to be fitted. As long as the shape of the surface is simple (such as a sphere), this may not be a difficult task. If, however, we deal with an arbitrary curved surface (such as a sculptured surface), this is as difficult as determining the whole mapping between 2D and 3D points.

The third problem is that there is a possibility of leaving "uncalculated regions" (See Figure 3) within the original surface area. There are two cases in which this problem arises. The first case may occur when the constrained yarn paths on a 3D surface are badly selected so that the mapping calculation cannot reach into the tinted region as shown in Figure 3 (a). The second case may occur when the surface area has a concave shape, as shown in Figure 3 (b), preventing the mapping calculation from reaching into the tinted region, regardless of the yarn paths.

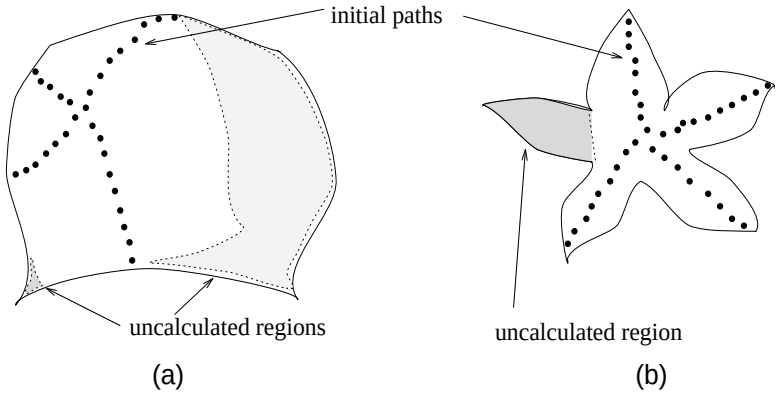

Figure 3: Uncalculated regions; (a) badly selected constrained yarn paths leave the uncalculated regions on the surface, (b) a concave surface area may not be totally covered.

4.2 New Method for Specifying Initial Conditions

We describe here a new method for specifying initial conditions. With this method, the first two problems with the previously used method can be overcome. The solution to the third problem (of an "uncalculated surface region") is addressed further below.

4.2.1 Initial conditions in 2D space

Our method for specifying initial conditions in 2D space is based on the actual manufacturing practices used by laminators during the lamination process. Laminators usually fix *one* arbitrary path that is not necessarily aligned with a particular thread. Then, they sweep the ply along arbitrary directions. This process is analogous to the process of ironing out wrinkled clothing after washing and drying. A crucial concept here is that the resulting deformation on a 3D surface generally depends on the history of how the ply has been swept so far.

To simulate this practice, we define *one* initial path instead of defining two perpendicular initial paths. We refer to this as the "base path" or the "guideline". As shown in Figure 4, the base path is not necessarily aligned with any weft or warp thread, but it must be a straight line in 2D space. A starting point must be chosen somewhere on the base path, but it is not expected to coincide with a thread crossing. When a base path is specified as above, "auxiliary" mesh points are automatically defined as the intersections between the base path and threads, as shown in open circles in Figure 4. Once auxiliary mesh points have been defined, the topology of the initial (regular) mesh points (shown in Figure 1 (b)) has to be modified. (See the section that describes Figure 7 for more details.)

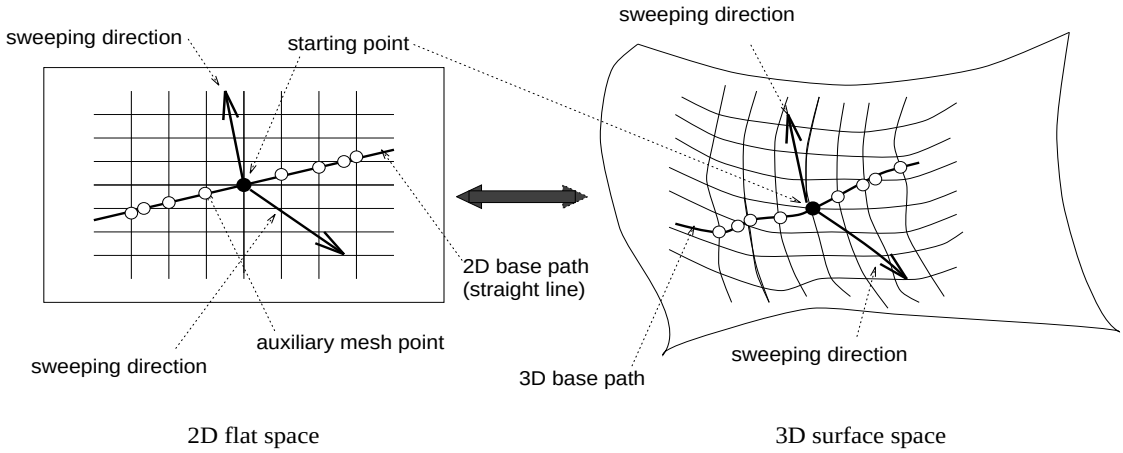

Figure 4: New method for specifying initial conditions; (a) 2D flat space, (b) 3D surface space

The base path generally subdivides the 2D space into two half-spaces, and, for each half-space, we define a "sweeping direction" by a 2D vector. It is the sweeping direction that determines the order of mapping calculations for the regular mesh points belonging to the half space.

As a special case, if we align the base path with a specific thread, and if we allow the two sweeping directions to be parallel to each other and perpendicular to the base path, the result is equivalent to that of the previously used method. By choosing the sweeping directions judiciously, and taking advantage of the local surface geometry, it is possible to decrease the "uncalculated surface regions."

4.2.2 Initial conditions in 3D space

The result of fitting a ply depends not only on the initial conditions in 2D space, but also on those in 3D space. We present here initial conditions in 3D space, the main aim of which is to provide a solution to the second problem associated with the previously used method, i.e. the problem that every mesh point on the initial paths in 3D surface space must be calculated by hand.

To automatically calculate the mapping of every mesh point on the base path in 3D space, we first need to define the curve on the surface corresponding to the base path. There are several ways to define such a curve. When defining the 3D curve as the base path, it is important that it be easy to specify, and that it should lead to flexible fitting results.

Probably, the simplest way of defining the base path in 3D is an isoparametric curve

(i.e. $u = const$), lying on the given parametric surface. This is, however, not a flexible method, because it depends on the surface parameterization. Similarly, parametric curves ($u = u(t)$ and $v = v(t)$) could be used. They are, however, difficult to specify, because a large number of control points on the surface may be required, which is equivalent to specifying a sequence of equidistant points.

Our method specifies the 3D base path as the intersection between two surfaces; one is given by the curved surface to which a ply is fitted, and the other is a plane called a "base plane". All that is needed to specify a base plane in 3D space is a reference point and a normal vector to the base plane. It is therefore very simple to specify, and yet this is significantly more flexible than the approach based on isoparametric curves.

For the mapping calculations of mesh points on the base path, we also assume that the distance between adjacent (auxiliary) mesh points on the base path is kept fixed in both 2D and 3D spaces. With this assumption, the mapping calculation of mesh points on the base path is reduced to the intersection problem between the base plane, a sphere whose radius is the distance between two adjacent mesh points, and a given NURBS surface as shown in Figure 5. Thus, we have reduced the problem of automatically calculating the mappings of sampled points on the base path in 3D space to repeated applications of an intersection problem.

The assumption that the base path must be a planar curve in 3D might be a strong constraint. Research is currently under way in which we allow an arbitrary geodesic for the base path in 3D space. It is also possible to extend the assumption that the base path in 2D space must be a straight line. It is, however, generally true that the shear-strain energy defined in Equation 14 tends to increase in the final fitting configuration if we start with a curved base path in 2D space.

For other initial conditions in 3D space, we define a starting point of the fitting in 3D space and sweeping directions as 3D vectors; this is similar to the corresponding concepts in 2D space. More details of the mapping calculation are discussed in the next section.

5 Mapping Calculations

In this section, we describe the algorithm that maps 2D points onto a 3D surface. We first focus on the problem of mapping a sequence of auxiliary mesh points on the base path, and then on the problem of mapping regular mesh points that are *not* on the base path. There is great similarity between the two cases. In fact, both problems are reduced to a surface-surface-intersection problem [11, 12, 13].

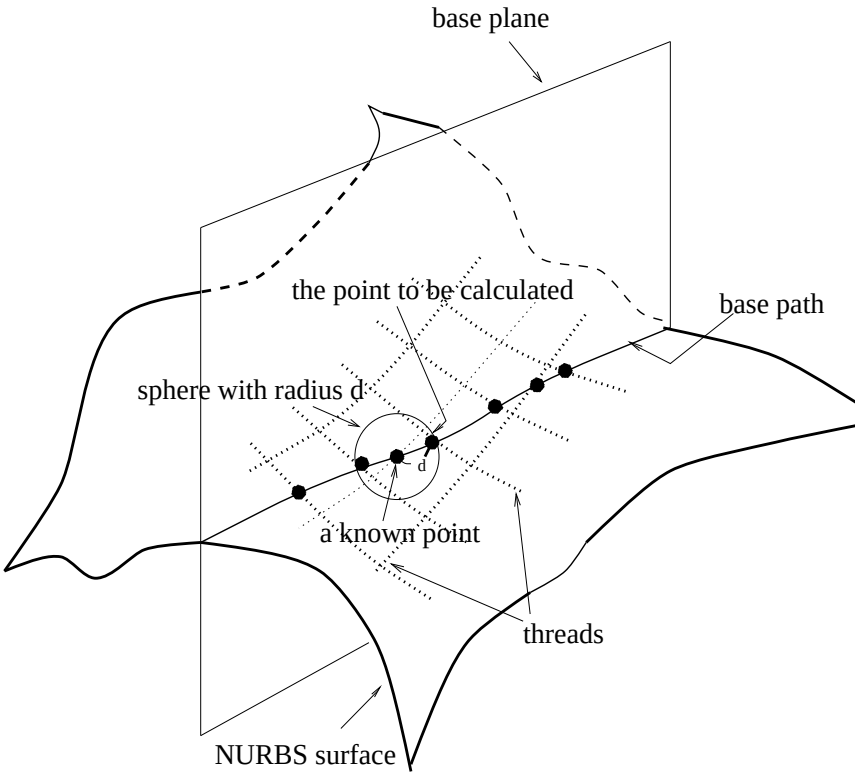

Figure 5: Mapping calculation of an auxiliary mesh point on the base path

5.1 Mapping Calculations for Auxiliary Mesh Points on Base Path

As shown in Figures 4 and 5, the solution to the mapping calculation for an auxiliary mesh point on the base path is defined as the intersection between the base plane, a sphere, and a NURBS surface.

Suppose that (x_0, y_0, z_0) represents a mesh point that is already known on the intersection curve in 3D space and d_{xyz} represent the step distance in 3D space between the mesh point and the next auxiliary mesh point to be calculated. The problem of obtaining the next point lying on the base plane whose normal is given by (x_e, y_e, z_e), is expressed by the following system of non-linear simultaneous equations.

For the plane,

$$x_e(x - x_0) + y_e(y - y_0) + z_e(z - z_0) = 0, \tag{1}$$

For the sphere,

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = d_{xyz}^2, \tag{2}$$

For the NURBS surface [10] (degree m $\times$ n),

$$\mathbf{r}(u,v) = \frac{\sum_{i=0}^{Cu-1}\sum_{j=0}^{Cv-1} w_{i,j}\mathbf{c}_{i,j}N_i^m(u)N_j^n(v)}{\sum_{i=0}^{Cu-1}\sum_{j=0}^{Cv-1} w_{i,j}N_i^m(u)N_j^n(v)}. \tag{3}$$

where the $\mathbf{c}_{i,j}$ are control points, the $w_{i,j}$ are weights, N is a B-spline basis function, and C_u and C_v are the number of control points along the u and v axes, respectively.

Substituting Equation (3) into Equations (1) and (2), we obtain

$$\left.\begin{array}{rcl} F_1(u,v) &=& x_e(x(u,v)-x_0)+y_e(y(u,v)-y_0) \\ &+& z_e(z(u,v)-z_0)=0 \\ F_2(u,v) &=& (x(u,v)-x_0)^2+(y(u,v)-y_0)^2 \\ &+& (z(u,v)-z_0)^2-d_{xyz}^2=0 \end{array}\right\}, \tag{4}$$

where $\mathbf{r}(u,v) = (x(u,v), y(u,v), z(u,v))$.

At this point, we have two simultaneous non-linear equations with two variables. This system of non-linear equations can be solved by the Newton-Raphson method [19], assuming the existence of first partial derivatives of the NURBS surface with respect to variables u and v. Let $(u^{(k)}, v^{(k)})$ denote the pair of variables at the k-th iteration. The relationship between (k+1)-th pair and the k-th pair is expressed as follows:

$$\begin{bmatrix} u^{(k+1)} \\ v^{(k+1)} \end{bmatrix} = \begin{bmatrix} u^{(k)} \\ v^{(k)} \end{bmatrix} - \begin{bmatrix} \Delta u^{(k)} \\ \Delta v^{(k)} \end{bmatrix}, \tag{5}$$

where

$$\begin{bmatrix} \frac{\partial F_1}{\partial u} & \frac{\partial F_1}{\partial v} \\ \frac{\partial F_2}{\partial u} & \frac{\partial F_2}{\partial v} \end{bmatrix} \begin{bmatrix} \Delta u^{(k)} \\ \Delta v^{(k)} \end{bmatrix} = \begin{bmatrix} F_1(u^{(k)}, v^{(k)}) \\ F_2(u^{(k)}, v^{(k)}) \end{bmatrix}. \tag{6}$$

The initial condition defines the first approximated point at a position distance d_{uv} in uv space from the starting point in the direction $\mathbf{r}_a = (u_a, v_a)$ as follows:

$$\begin{bmatrix} u^{(0)} \\ v^{(0)} \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + d_{uv}\begin{bmatrix} u_a \\ v_a \end{bmatrix}, \tag{7}$$

where

$$\mathbf{r}_a = u_a\frac{\partial \mathbf{r}}{\partial u} + v_a\frac{\partial \mathbf{r}}{\partial v}.$$

The distance d_{uv} is approximated by the following equation:

$$d_{uv} = d_{xyz} R, \tag{8}$$

where R is the ratio of the length of a unit vector $\tilde{\mathbf{V}}_{xyz}$ in xyz space to the length of the corresponding vector $\mathbf{V}_{uv}$ in uv space. At the starting point on the base path, $\tilde{\mathbf{V}}_{xyz}$ is given as a parameter. Once a next mesh point is calculated, it is updated every time by the following formula:

$$\tilde{\mathbf{V}}_{xyz} = (\mathbf{P}_{-1} - \mathbf{P}_0)/|\mathbf{P}_{-1} - \mathbf{P}_0|,$$

where $\mathbf{P}_{-1} = (x_{-1}, y_{-1}, z_{-1})$ (previous point) and $\mathbf{P}_0 = (x_0, y_0, z_0)$ (current point). $\mathbf{V}_{uv}$ is obtained by the orthonormal projection of $\tilde{\mathbf{V}}_{xyz}$ onto the tangent plane at $\mathbf{P}_0$.

The terminating condition is given as follows:

$$|F_1(u^{(k+1)}, v^{(k+1)})| + |F_2(u^{(k+1)}, v^{(k+1)})| < \varepsilon, \tag{9}$$

where ε is a small positive constant.

5.2 Mapping Calculations for Regular Mesh Points

Not surprisingly, the mapping calculation for (regular) mesh points that are *not* on the base path is similar to the intersection calculation given in the previous section. However, we have to deal with two cases: (a) *only one* neighbor of the current mesh point is known, and (b) *two* neighbors of the current mesh point are known. Note that a "neighbor" here refers to a mesh point adjacent to the mesh point of interest and has a direct link to it. Case (a) is considered in the next section. Here, we focus on case (b).

Figure 6 shows how to perform the mapping calculation for case (b). Given that the positions of the mesh points $p(i-1, j)$ and $p(i, j-1)$ on the NURBS surface are known, it is straightforward to calculate the position of $p(i, j)$. Since it is assumed that the warp and weft threads are inextensible, the point $p(i, j)$ can be defined by the intersection of a sphere of radius d^{weft} centered at $p(i-1, j)$, a sphere of radius d^{warp} centered at $p(i, j-1)$, and the NURBS surface.

Since the subsequent calculations are basically similar to the mapping calculations for points on the base path, only the details pertaining to the initial and terminating conditions are described here.

To apply Equation (5), variables for the initial condition must be represented in uv space. Let $p(i, j)_{uv}$ denote a mesh point represented in uv space at a crossing of i-th weft thread and j-th warp thread. We assume that $p(i-1, j)_{uv}$ is on the same weft

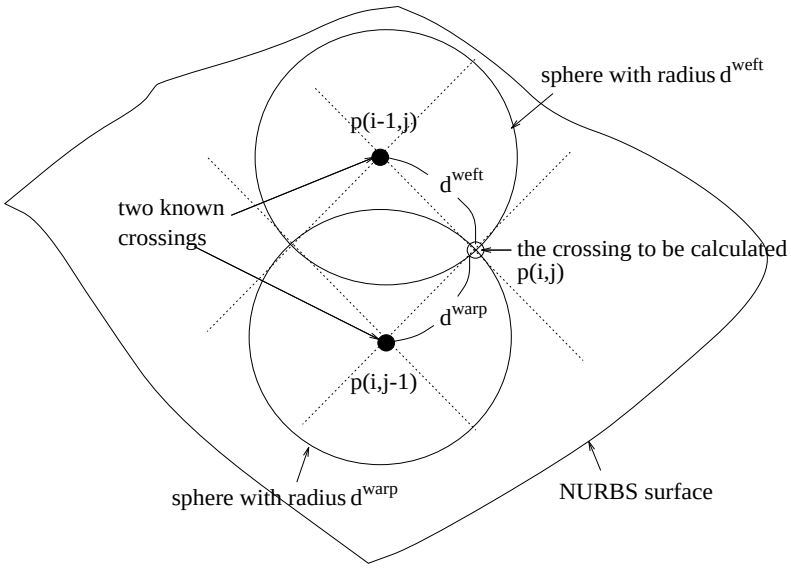

Figure 6: Mapping calculation of a regular mesh point, with two known neighbors.

thread with $p(i,j)_{uv}$, and $p(i,j-1)_{uv}$ is on the same warp thread with $p(i,j)_{uv}$. Let us define the *weft gradient vector* $\mathbf{v}_{uv}^{weft}$ at $p(i-1,j)_{uv}$ as

$$\mathbf{v}_{uv}^{weft} = p(i-1,j)_{uv} - p(i-2,j)_{uv}.$$

Similarly, we define the *warp gradient vector* $\mathbf{v}_{uv}^{warp}$ at $p(i,j-1)_{uv}$ as

$$\mathbf{v}_{uv}^{warp} = p(i,j-1)_{uv} - p(i,j-2)_{uv}.$$

Let us then define two points $p(i,j)_{uv}|^{weft}$ and $p(i,j)_{uv}|^{warp}$ as follows:

$$\left.\begin{aligned} p(i,j)_{uv}|^{weft} &\equiv \begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix}^{weft} = \begin{bmatrix} u_{i-1,j} \\ v_{i-1,j} \end{bmatrix} + (d_{uv}^{weft})\tilde{\mathbf{v}}_{uv}^{weft} \\ p(i,j)_{uv}|^{warp} &\equiv \begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix}^{warp} = \begin{bmatrix} u_{i,j-1} \\ v_{i,j-1} \end{bmatrix} + (d_{uv}^{warp})\tilde{\mathbf{v}}_{uv}^{warp} \end{aligned}\right\}, \tag{10}$$

where $\tilde{\mathbf{v}}_{uv}^{weft}$ and $\tilde{\mathbf{v}}_{uv}^{warp}$ are normalized weft and warp gradient vectors in uv space and distances d_{uv}^{warp} and d_{uv}^{weft} are obtained in a similar fashion as d_{uv} in Equation (8). Then, we approximate the initial point for Newton's method with the following formula:

$$\begin{bmatrix} u^{(0)} \\ v^{(0)} \end{bmatrix} \equiv \begin{bmatrix} u_{i,j}^{(0)} \\ v_{i,j}^{(0)} \end{bmatrix} = \frac{1}{2}\left(\begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix}^{weft} + \begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix}^{warp} \right). \tag{11}$$

This assumption works well in most cases where the uv parameterization has a nearly linear relationship with the xyz parameterization in 3D Cartesian space.

For the termination conditions, we add the following two cases to the normal convergence condition given in Equation (9):

$$\left.\begin{array}{rcl} \gamma^{(k+1)} & < & \gamma_{min} \\ \gamma^{(k+1)} & > & \gamma_{max} \end{array}\right\}, \tag{12}$$

where

$$\begin{array}{rcl} \gamma^{(k+1)} & = & \arccos \frac{(\mathbf{x}_1 - \mathbf{x}^{(k+1)}) \cdot (\mathbf{x}_2 - \mathbf{x}^{(k+1)})}{|\mathbf{x}_1 - \mathbf{x}^{(k+1)}||\mathbf{x}_2 - \mathbf{x}^{(k+1)}|}, \\ \mathbf{x}_1 & = & \mathbf{r}\big(u_{i-1,j}, v_{i-1,j}\big), \\ \mathbf{x}_2 & = & \mathbf{r}\big(u_{i,j-1}, v_{i,j-1}\big), \\ \mathbf{x}^{(k+1)} & = & \mathbf{r}\big(u^{(k+1)}, v^{(k+1)}\big). \end{array}$$

These conditions (Equation (12)) correspond to the situation in which a given 2D ply cannot be fitted to a 3D surface because the thread angle between weft and warp exceeds γ_{min} or γ_{max}. Note that the values of γ_{min} and γ_{max} depend only on the material property. Values of γ_{min} and γ_{max} are available from experimental data obtained by Lindberg et al. [20]. They developed a special apparatus for measuring shear-angle limits. For example, shear-angle limits for cotton range from 0.18 degrees (for sheeting) to 9.0 degrees (for shirting), and, for wool, the range is 16-27 degrees. These values basically correspond to γ_{min}, and γ_{max} can be roughly estimated as $\pi - \gamma_{min}$.

6 Scanning Algorithms

The mapping algorithms mentioned in the previous section are applied to each mesh point in a given 2D ply. Scanning algorithms, on the other hand, are applied to a set of mesh points. The purpose of a scanning algorithm is to determine the order of the mapping calculations through the mesh points. This order has to conform to the given initial conditions including the base path and the sweeping directions. In general, the resulting mapping calculations depend on the scanning order of the mesh points.

Consider the example shown in Figure 7, in which regular mesh points are given by a 4×4 network of nodes labeled from '1' to '16', auxiliary mesh points on the base path are given by a sequences of nodes labeled from 'a' to 'f', and the sweeping directions are assumed to be perpendicular to the base path, as shown by arrows in the figure.

In terms of topology, once auxiliary mesh points 'a' to 'f' are defined, neighboring relations are altered. For example, nodes '7' and '8' are originally neighbors to each other. Once the base path is defined, however, they are no longer neighbors. Instead nodes '7' and 'b' are new neighbors, and so are nodes '8' and 'b'.

Regular mesh points are divided into two groups by the base path. Each group is

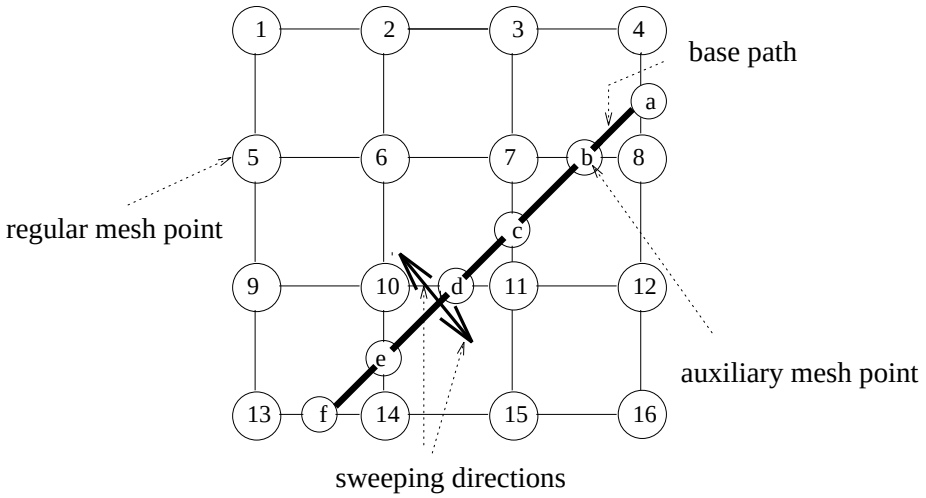

Figure 7: Example of scanning with the initial paths defined by base path and sweeping directions.

scanned in the corresponding sweeping direction. Since the scanning of mesh points in each group can be done independently, we will describe the group that includes node '1'.

Figure 8 shows the scanning order in a directed acyclic graph (DAG), which we call the *dependency graph* of the mesh points in a given 2D ply. Specifically, nodes in the dependency graph represent mesh points, and arcs represent the dependency relations between the mesh points. Two nodes linked by an arc are neighbors to each other. Every node has at most two incoming arcs and at most two outgoing arcs. In this example, assuming that the nodes labeled from 'a' to 'f' are already known, we first calculate the mapping of a group of nodes '4','7','10', and '13.' The exact order among them is determined by the distance between the nodes and the base path measured along the sweeping direction. For instance, mapping calculation of node '3' can be carried out once nodes '4' and '7' have become available. In general, the mapping calculation of a particular node in the dependency graph can be carried out once all the nodes which have outgoing arcs to the node of interest have become available.

It should be noted that there are nodes with only one incoming arc, such as nodes '4' and '13' in Figure 8. We hereafter call these nodes *indegree one nodes*, and other nodes *indegree two nodes*. It is this indegree one node that corresponds to case (a) in the previous section on mapping calculations. The indegree one nodes require special consideration because the ordinary mapping algorithm that requires two known neighbors (case (b)) cannot be used to calculate them. This is the source of the "uncalculated surface regions" problem as shown in Figure 3. In the following section, we first describe a mapping calculation for indegree one nodes, and then describe how to scan

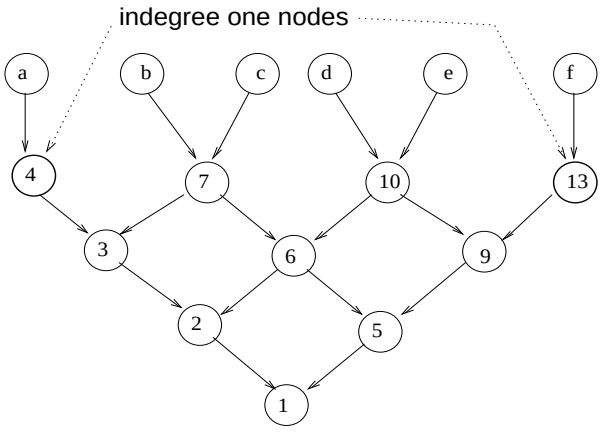

Figure 8: Dependency graph of the example in Figure 7.

mesh points.

6.1 Approximation to the Mapping Calculation for Indegree One Node

Indegree one nodes require a special mapping algorithm. It is impossible to uniquely determine the mapping of an indegree one node uniquely with only *one* known adjacent mesh point. Additional information must be supplied for the mapping to be performed from 2D to 3D.

We have developed a heuristic which assumes that the local geometry is preserved around the indegree one node to be calculated. Specifically, we assume that the angular relationship between a thread and the base path at a mesh point in 2D space is preserved on the tangent plane at the mapped point in 3D space, as shown in Figure 9.

For example, suppose we are looking at the mapping of node '4', where the angle in the 2D space between the base path and the thread passing through nodes '4' and 'a' is θ as shown in Figure 9 (a). Then, as shown in Figure 9 (b), this angular relationship is preserved on the tangent plane at the mapped point of 'a' in 3D.

Thus, the problem of mapping node '4' is reduced to the intersection problem between a plane (shown with dotted lines passing through nodes '4' and 'a'), a sphere whose radius is d, and a NURBS surface. All the mapping calculations beyond node '4' from the base path are dependent upon this approximation, as shown in the dependency graph of Figure 8. This is why the entire mapping calculation is generally affected by the scanning order.

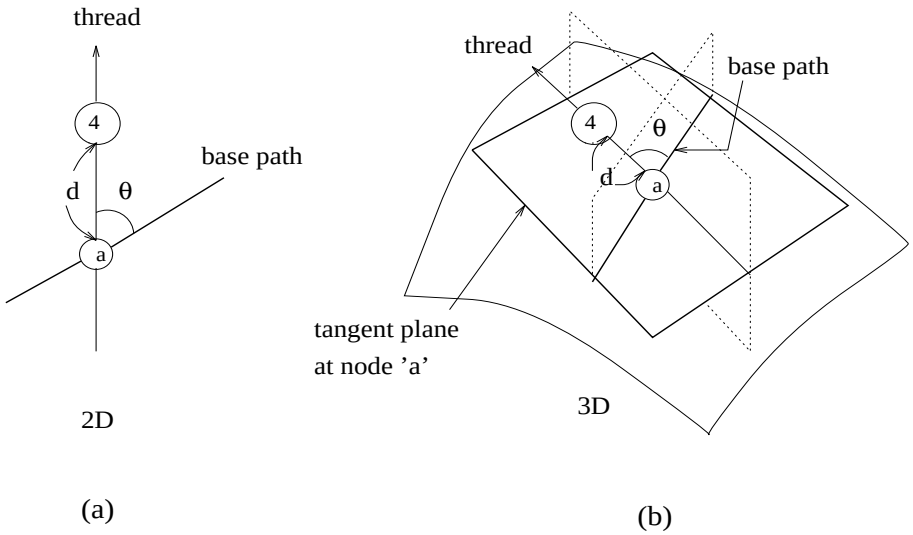

Figure 9: Approximation of the mapping calculation for node ('4') with only one known neighbor ('a').

To obtain the plane equation which passes through nodes '4' and 'a', we first calculate the direction vector from 'a' to '4' at the mapped point of 'a' in 3D space by rotating the direction vector of the base plane by θ around the surface normal at 'a'. The plane equation is then determined by this direction vector and the surface normal as expressed in the following formula:

$$\begin{vmatrix} x & y & z & 1 \\ x_0 & y_0 & z_0 & 1 \\ n_x & n_y & n_z & 0 \\ d_x & d_y & d_z & 0 \end{vmatrix} = 0, \tag{13}$$

where (n_x, n_y, n_z) denotes the surface normal at 'a', (d_x, d_y, d_z) denotes the direction vector from 'a' to '4', and (x_0, y_0, z_0) denotes the mapped point of 'a'. Equation (13) is a general plane equation that is used to approximate the mapping of a mesh point with only *one* known neighbor.

6.2 Scanning Mesh Points

Once auxiliary mesh points on the base path have been mapped to 3D, the scanning order of the remaining regular mesh points must be determined. During the scanning process, indegree one nodes in the dependency graph and the surface boundary may

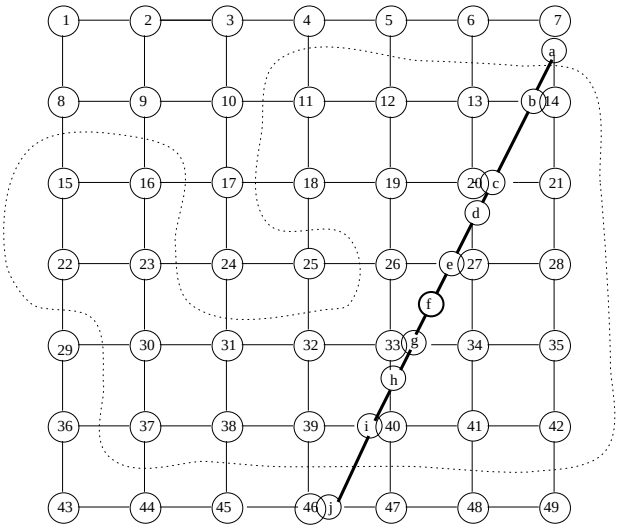

Figure 10: Complex scanning example that includes runtime recognition of the surface boundary.

be encountered.

Consider the detailed example shown in Figure 10. Suppose that regular mesh points are labeled from '1' to '49', and auxiliary mesh points on the base path are labeled from 'a' to 'j'. Node 'f' represents the starting point, which does not lie on any thread. Suppose also that the dotted lines indicate the surface boundary that we will encounter during the mapping calculation. Note that the location of the surface boundary is determined when the mapping calculations reach the boundary at runtime. The scanning algorithm must identify the surface boundary at runtime, the scanning order thus being kept consistent with the given initial conditions.

Once nodes 'a' to 'j' (except for node 'f') have been calculated, the scanning algorithm proceeds as follows. Let us first define the *frontier* of the scanning as a group of mesh points that include candidates for the next mapping calculation. Initially, it consists of mesh points adjacent to those on the base path. For example, in Figure 10, we have two independent frontiers: {'7','13', '20', '26', '33', '39', '46'} and {'14', '21', '27', '34', '40', '47'}. For the sake of simplicity, we focus only on the first frontier. The frontier is sorted at creation time in the order specified by the sweeping direction. For example, if we assume the sweeping direction is perpendicular to the base path, the first element in the frontier is the nearest mesh point to the base path. If there are two or more mesh points that have the same distance to the base path, the mesh point nearest to a given starting (reference) point is chosen. The frontier is therefore regarded as an ordered list whose elements are mesh points. In addition, we maintain the number of known neighbors for each list element. In this particular example, the result of the sorting

is {<'33',2>, <'20',2>, <'46',1>, <'7',1>, <'39',1>, <'13',1>, <'26',1>}, where the second number appearing in each list element is the number of known neighbors at sorting time. For instance, node '33' has two known neighboring nodes 'g' and 'h'.

We then proceed to the mapping calculation for the first element in the frontier list. The first element is extracted from the list, and its number of known neighbors is examined. If it is equal to *two*, the ordinary mapping calculation based on the intersection between a NURBS surface and two spheres is performed. If it is equal to *one*, the mapping based on the intersection between a NURBS surface, a plane and a sphere is performed. In both case, if the mapping calculation is successfully completed, the extracted element is replaced by its *unknown* neighbors. Specifically, if the unknown neighbor is already in the frontier list, its number of known neighbors is incremented; otherwise, the new elements are sorted and inserted into the frontier list, and their number of known neighbors is set to one. If the mapping calculation terminates because the mesh point is outside the NURBS surface, the element is simply removed from the list. This process is repeated until the frontier list is empty. To find the surface boundary at runtime, the in-out determination of the mapped point is performed during the mapping calculation by checking whether $(u^{(k+1)}, v^{(k+1)})$ defined in Equation (5) is inside or outside the NURBS patch of interest.

7 Simulation Examples

In this section, we present two examples. Throughout the discussion of these examples, we explain how the initial conditions for the fitting are specified, the resulting 2D plane development, and how two or more fittings with different initial conditions compare.

7.1 Octant of a Sphere

First, we present an example of an octant of a sphere. A sphere is a typical quadric that can be represented exactly by a NURBS surface. In this example, we present the basic idea of how initial conditions are specified for the fitting, and how they affect the resulting configurations. We assume that R denotes the radius of the sphere, and the geometry is defined in the first octant ($X \geq 0$, $Y \geq 0$, $Z \geq 0$).

We have carried out simulation experiments with three different initial conditions, which are summarized in Table 1. Specifically, they have a common base path along the circle on the $Y = 0$ plane in 3D space. In 2D space, cases (a) and (b) have a base path that is parallel to the weft threads, and case (c) has a base path at a 45 degree angle to both the wefts and the warps. Since the base path coincides with the boundary of the octant of a sphere, only one sweeping direction is specified. Each

Table 1: Initial conditions for the three fitting in Figure 11.

	(a)	(b)	(c)
3D			
starting point	(0,0,R)	($\frac{R}{2}$,0,$\frac{R}{2}$)	(0,0,R)
sweeping direction	(0,1,0)	(0,1,0)	(0,1,0)
base path	Y = 0	Y = 0	Y = 0
2D			
starting point	(0,0)	(20,0)	(15,0)
sweeping direction	(0,1)	(0,1)	(-1,1)
base path	Y = 0	X = 20	Y = X - 15

Table 2: Comparison of results for various initial conditions.

	(a)	(b)	(c)
area	716	657	725
$\min(\arccos\gamma_{i,j})$	0.707	1.167	0.353
$\max(\arccos\gamma_{i,j})$	1.570	1.570	1.570
$\sum_{i,j}(\cos\gamma_{i,j})^2$	82.5	17.4	128.2

sweeping direction is defined by a pair of vectors both in 2D and 3D spaces. The starting points for cases (a) and (c) are located at the end points of the base paths, and the starting point for case (b) is located at the midpoint of the base path.

The resulting fitting configurations and the associated 2D plane developments produced from the mappings are shown in Figure 11. By convention, we denote the starting point by P_s, the sweeping directions by directed arrows incident from the starting point, and the base path by a sequence of solid circles in the figures. Note that the bold-line segments in the 2D plane developments represent the area mapped to the surface.

Of the three cases, case (a) produces the same results as the previous methods [6, 7, 9, 14] when a thread line is aligned with the base path and the sweeping direction is perpendicular to the base path.

To compare the results, we calculated the values for the following, which are given in Table 2:

1. the area of the 2D ply that is required for the mapping,
2. the maximum and minimum shear angles between vertical and horizontal threads,
3. the shear deformation energy.

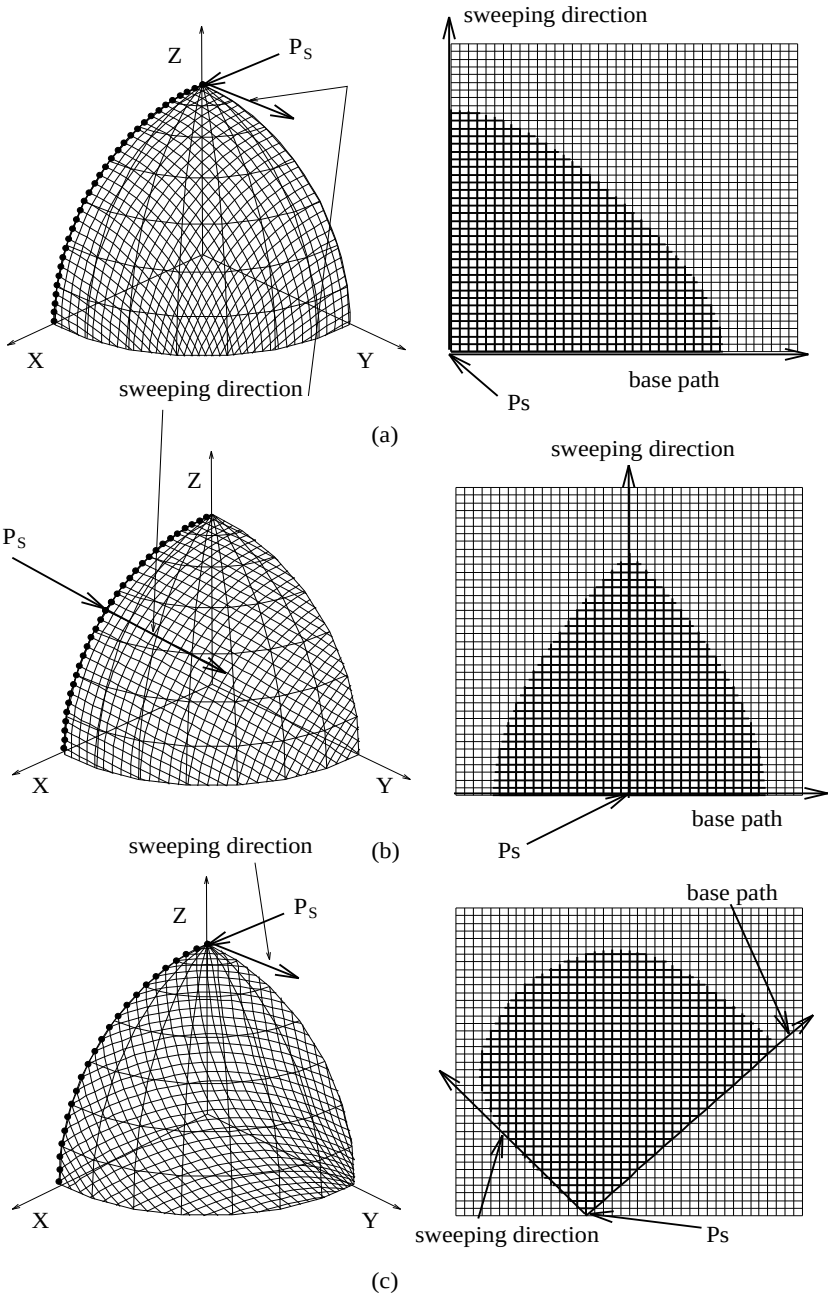

Figure 11: Fittings to an octant of a sphere and their plane developments.

First, the area of the 2D ply that is required for the mapping is approximated by summing up the number of mesh points in the 2D plane development pattern.

Second, the maximum (γ_{max}) and minimum (γ_{min}) shear angles are simply calculated as $\gamma_{max} = max(\gamma_{i,j})$ and $\gamma_{min} = min(\gamma_{i,j})$, where $\gamma_{i,j} = \arccos(\mathbf{r_i} \cdot \mathbf{r_j})$, and $\mathbf{r_i}$ is the unit weft vector and $\mathbf{r_j}$ is the unit warp vector at mesh point (i, j).

Finally, the shear-deformation (strain) energy is approximated by the value $\sum_{i,j} \cos^2 \gamma_{i,j}$. This comes from the plane strain assumption [21] for a sheet of given composite material. Specifically, as we have assumed no extension in the thread directions, the strain energy V is given by $V = (1/2)\sigma_{ij}\varepsilon_{ij}$, where σ_{ij} is the shear stress, and ε_{ij} is the shear strain. Since $\varepsilon_{ij} = (1/G)\sigma_{ij} = \mathbf{r_i} \cdot \mathbf{r_j} = |\mathbf{r_i}||\mathbf{r_j}|cos(\gamma_{i,j}) = cos(\gamma_{i,j})$, the strain energy is reduced to the following:

$$V = \frac{1}{2}G\varepsilon_{ij}^2 = \frac{1}{2}Gcos^2(\gamma_{i,j}) \tag{14}$$

where G is a modulus of elasticity in the shear. Generally speaking, a greater deviation from the starting value ($\pi/2$) implies a shear accumulation of deformation energy, and a greater danger of anomalies and undesirable results.

Given two or more fittings and the associated values as defined above, we may evaluate the quality of the fittings. The best fitting is the one that consumes the smallest ply area in 2D space, has the least variation of the thread angles (i.e. minimum γ_{max} and maximum γ_{min}), and has the minimum deformation energy. It should be noted, however, that there may not always be a unique "best" fitting.

In our simulation experiments, the fitting that consumes the least ply area is case (b). In case (b), there is also the least variation of the thread angles, and has the minimum deformation energy. We can therefore conclude that case (b) is the most desirable fitting of the three cases.

7.2 One-Third of a Surface of Revolution

We present here several fittings onto one-third of a surface of revolution. Figure 12 shows three views of the shape before the fitting: (a) an oblique view, (b) a side view, and (c) a top view. The purpose of this example is to compare the results of various fittings with the aid of the plots of the thread-angle values.

Figure 13 shows the results of six different fittings. As in the previous example, the first three cases ((a), (b), and (c)) have a common base path along the boundary curve on the $Y = 0$ plane, and the other three cases ((d), (e), and (f)) have a common base path along the curve on the $X = Y$ plane. In 2D space, in cases (a), (c), (d), and (f), the base path is parallel to the warps, and in cases (b) and (e), the base path is at 45 degrees to the warps. The definitions of the sweeping directions and the starting points

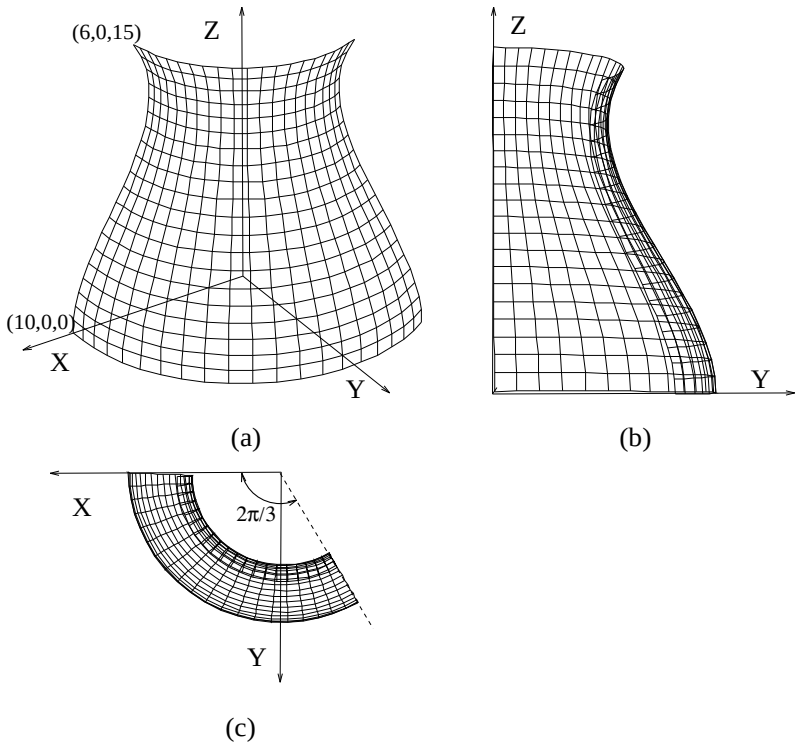

Figure 12: Views of a quadrant of a surface of revolution, (a) oblique view, (b) side view, (c) top view.

are similar to those in the previous example, and the initial conditions are summarized in Table 3. We assume that the mesh size of the 2D ply in this example is 70×70. Figure 14 includes the 2D plane developments that correspond to the fittings of cases (a) to (f) in Figure 13.

Figures 13 and 14 clearly demonstrate the critical effect of initial conditions on the final mappings. Table 4 summarizes the results of the six different cases. Case (f) requires the smallest ply area. Case (d) has the minimum γ_{max}, and case (f) has the maximum γ_{min}. Case (d) has the smallest deformation energy. We cannot therefore conclude which is the best fitting, but we can clearly state that cases (d), (e), and (f) produce better results than cases (a), (b), and (c). Of the six results, case (c) is exceptional, since it fails to cover the entire surface region with the given 2D ply.

Figure 15 shows the distributions of thread angles between wefts and warps at mesh points projected above the associated plane developments, in which the vertical axis (or height field) represents the value of the thread angle, ranging from 0 to π.

The plot of thread-angle values provides an insight into the differences between two or more fittings onto a common surface. Generally speaking, a good fitting has little bend in the surface of the plot. It is clearly observed in Figure 15 that cases (d), (e), and (f) have less bend in the surface of the thread angle plot than do cases (a), (b), and (c).

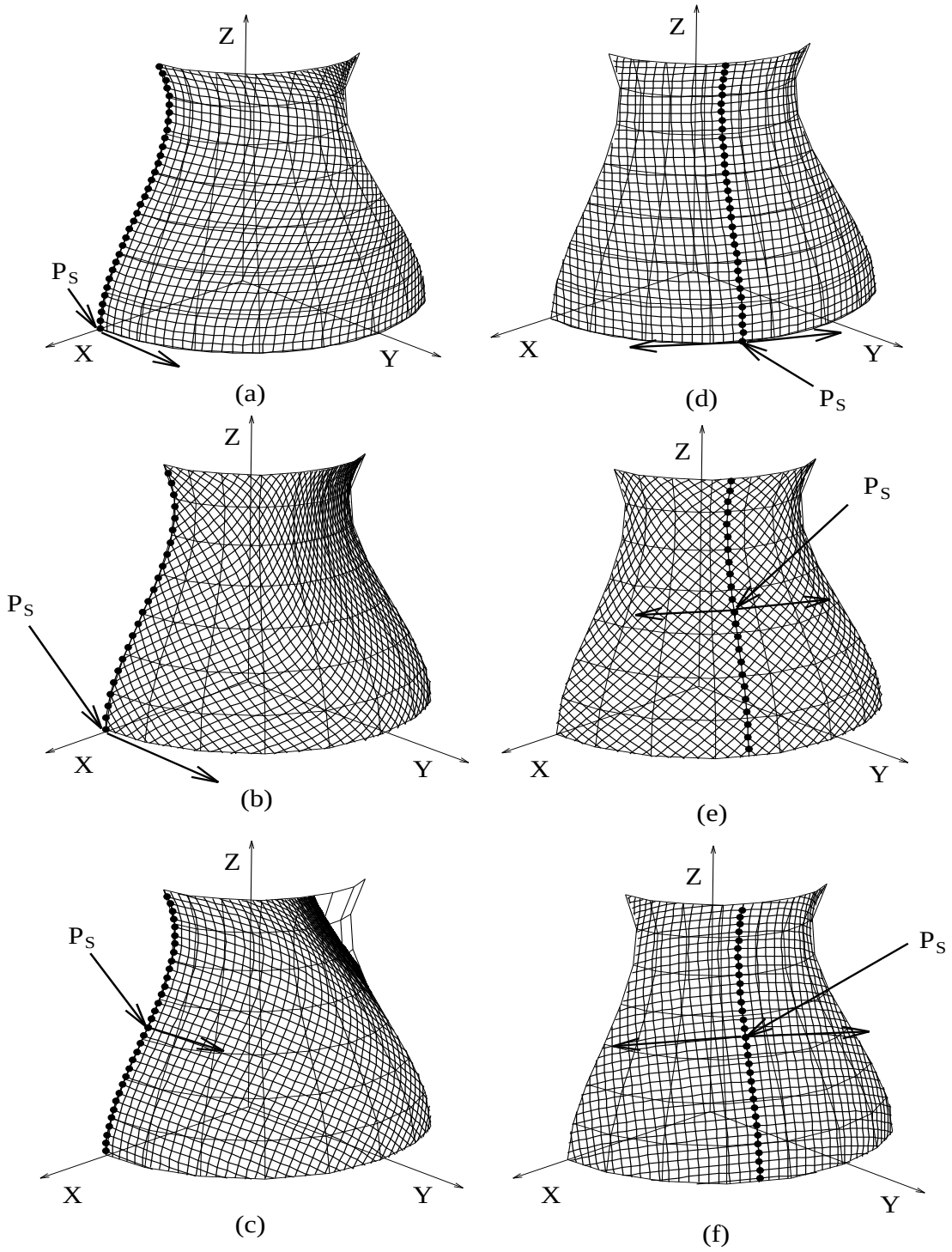

Figure 13: Fittings to a third part of a surface of revolution with six different initial conditions.

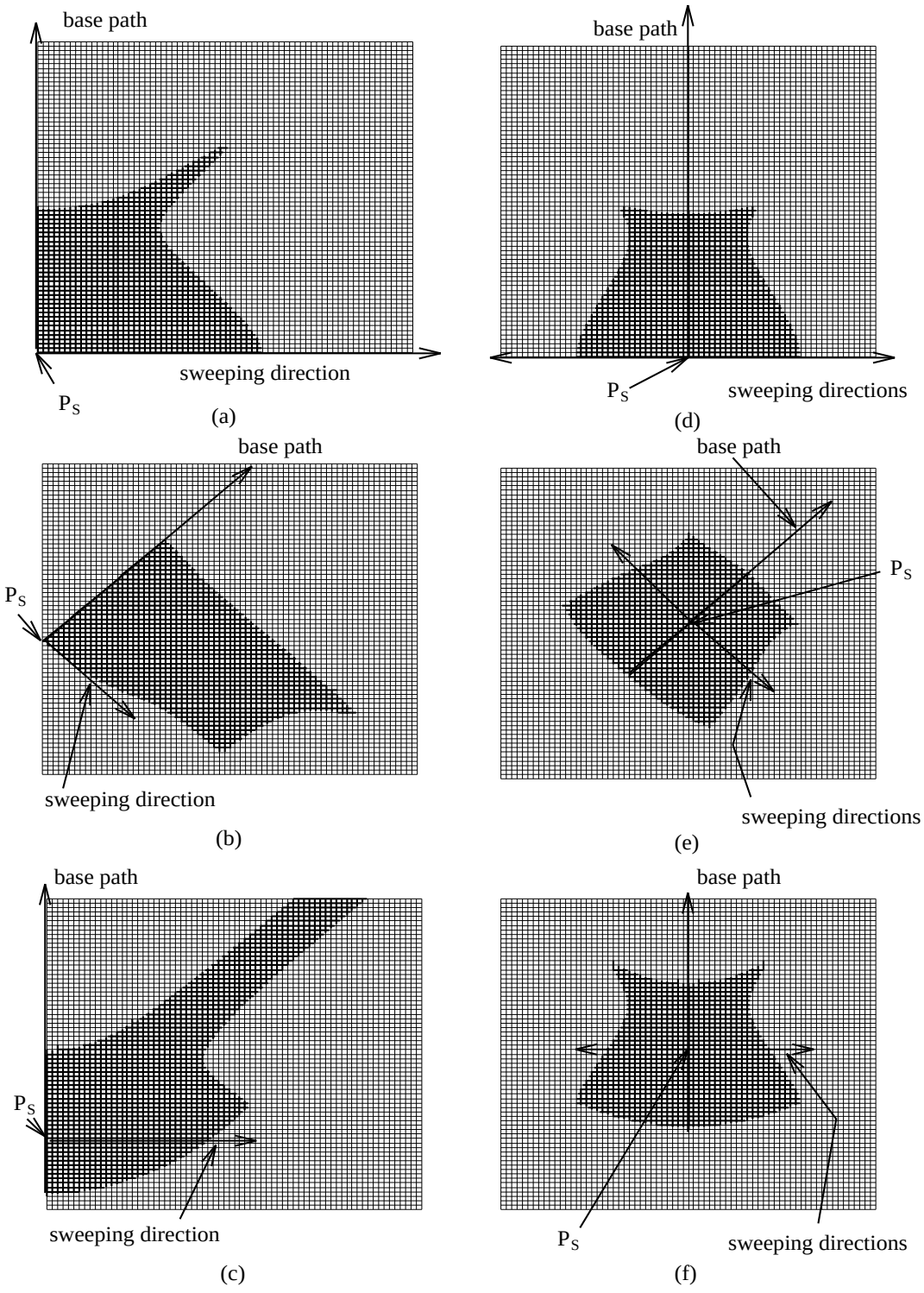

Figure 14: Plane development patterns used for the fitting to a third of a surface of revolution.

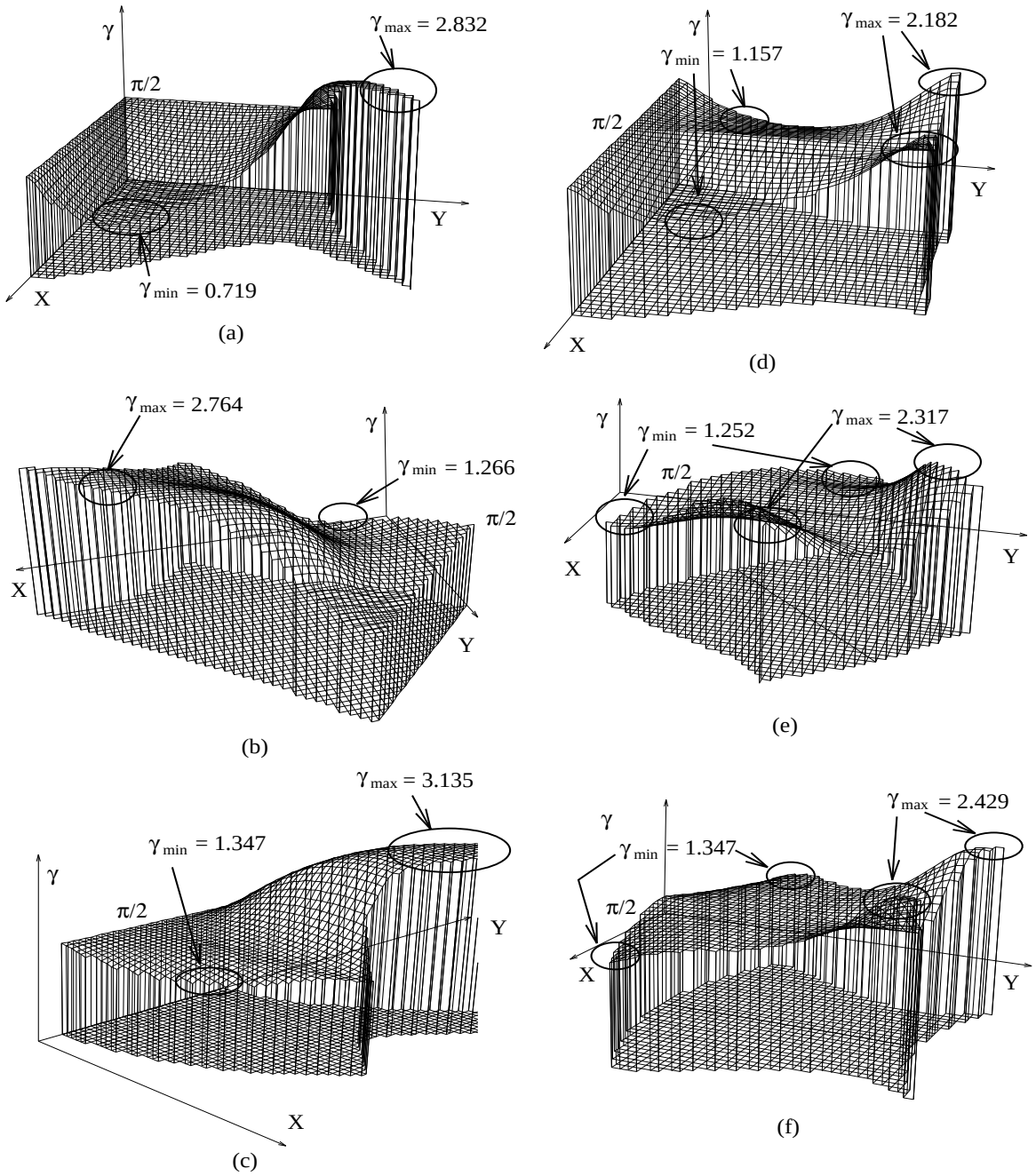

Figure 15: Distribution of thread angles between wefts and warps at mesh points projected above the associated plane developments.

Table 3: Initial conditions for the fitting in Figure 13.

	(a)	(b)	(c)
3D			
starting point	(10,0,0)	(10,0,0)	$(\frac{55}{8},0,\frac{15}{2})$
sweeping directions	(0,1,0)	(0,1,0)	(0,1,0)
base path	Y = 0	Y = 0	Y = 0
2D			
starting point	(0,0)	(0,30)	(0,20)
sweeping directions	(1,0)	(1,-1)	(1,0)
base path	X = 0	Y = X + 30	X = 0

	(d)	(e)	(f)
3D			
starting point	$(10,10\sqrt{3},0)$	$(\frac{55}{8},\frac{55}{8}\sqrt{3},\frac{15}{2})$	$(\frac{55}{8},\frac{55}{8}\sqrt{3},\frac{15}{2})$
sweeping directions	(1,-1,0)	(1,-1,0)	(1,-1,0)
	(-1,1,0)	(-1,1,0)	(-1,1,0)
base path	X = Y	X = Y	X = Y
2D			
starting point	(35,0)	(35,35)	(35,35)
sweeping directions	(-1,0)	(-1,1)	(-1,0)
	(1,0)	(1,-1)	(1,0)
base path	X = 35	Y = X	X = 35

Table 4: Comparison of results with various initial conditions.

	(a)	(b)	(c)	(d)	(e)	(f)
area	1159	1246	1474	1023	1017	1010
$\min(\arccos\gamma_{i,j})$	0.719	1.266	1.347	1.157	1.252	1.347
$\max(\arccos\gamma_{i,j})$	2.832	2.764	3.135	2.182	2.317	2.429
$\sum_{i,j}(\cos\gamma_{i,j})^2$	211.5	385.9	588.6	36.7	50.5	40.9

8 Conclusions

We have presented a general technique for fitting a given 2D broadcloth composite ply to a given 3D curved surface represented by a NURBS surface. We have described a model for a 2D ply that is based on a limited number of assumptions including the inextensibility of threads.

As a key component of the fitting algorithm, we have developed a mapping method that makes it possible to start with an arbitrarily oriented base path. This method provides greater flexibility than the previously used methods [6, 7, 9, 14], which require the alignment of two perpendicular threads with the initial paths. We have also described a method of automatically calculating the mapping of mesh points on the base path by defining it as the intersection between a plane and the given surface. The "uncalculated surface region" problem has been avoided by introducing a heuristic that assumes that the local geometry around a mesh point is preserved.

With two simulation experiments, we have demonstrated the usefulness of the graphical tools that we have developed, which include the display of the 3D fitting, the display of a 2D "flattened" pattern before the ply is deformed, and the display of thread-angle distribution.

9 Acknowledgements

We are indebted to the referees of this paper, who suggested numerous improvements to our earlier draft. Thanks are due to Jospeh Piteo and Joseph Santiso of Sikorsky Aircraft for providing us with information on broadcloth-composite manufacturing. This work has been sponsored by the Overseas Scholarship Program of IBM Japan Ltd. and the Industrial Associates Program of the Rensselaer Design Research Center, USA. Any opinions, findings and conclusions expressed in this publication are those of the authors, and they do not necessarily reflect the views of any of the sponsors.

References

[1] B. C. Hoskin and A. A. Baker, *Composite Materials for Aircraft Structures*, AIAA, Inc., New York, NY (1986)

[2] D. P. Raymer, *Aircraft Design: A Conceptual Approach*, AIAA Inc., New York, NY (1989)

[3] K. K. Chawla, *Composite Materials - Science and Engineering*, Springer-Verlag, New York Inc. NY (1987)

[4] M. Aono, D. E. Breen, and M. J. Wozny, "A Computer-Aided Broadcloth Composite Layout Design System," in P. R. Wilson et al. (Ed.), *Geometric and Product Modeling*, Elsevier (1993)

[5] C. Mack and H. M. Taylor, "The Fitting of Woven Cloth to Surfaces," *Journal of the Textile Institute*, Vol. 47, T477-488 (1956)

[6] R. E. Robertson, H. S. Hsiue, E. N. Sickafus, and G. S. Y. Yeh, "Fiber Rearrangements During the Modeling of Continuous Fiber Composites – I. Flat Cloth to a Hemisphere," *Polymer Composites*, Vol. 2, No. 3, pp. 126-131, July (1981)

[7] R. E. Robertson, E. S. Hsiue, and G. S. Y. Yeh, "Continuous Fiber Rearrangements During the Molding of Fiber Composites – II. Flat Cloth to a Rounded Cone," *Polymer Composites*, Vol. 5, No. 3, pp. 191-197, July (1984)

[8] F. L. Heisey and K. D. Haller, "Fitting Woven Fabric to Surfaces in Three Dimensions," *Journal of the Textile Institute*, Vol. 79, No. 2, pp. 250-263 (1988)

[9] B. P. Van West, R. B. Pipes, and M. Keefe, "A Simulation of the Draping of Bidirectional Fabrics over Arbitrary Surfaces," *Journal of the Textile Institute*, Vol. 81, No. 4, pp. 448-460 (1990)

[10] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design*, Third Edition, Academic Press Inc., San Diego, CA (1992)

[11] M. J. Pratt and A. D. Geisow, "Surface/Surface Intersection Problems" in *The Mathematics of Surfaces*, ed. by J. A. Gregory, Clarendon Press, Oxford, pp. 117-142 (1986)

[12] C. M. Hoffmann, *Geometric & Solid Modeling: An Introduction*, Morgan Kaufmann Publishers, Inc. San Mateo, CA (1989)

[13] R. E. Barnhill ed., *Geometry Processing for Design and Manufacturing*, SIAM, Philadelphia, PA (1992)

[14] O. K. Bergsma, and J. Huisman, "Deep Drawing of Fabric Reinforced Thermoplastics," in *Proc. CADCOMP*, eds. by C. A. Brebbia, W. P. deWilde, and W. R. Blain, Springer-Verlag, pp. 323-334 (1988)

[15] B. P. Van West, "A Simulation of the Draping and a Model of the Consolidation of Commingled Fabrics," Ph.D. Thesis, University of Delaware, DE (1990)

[16] B.G. Prakash, T. G. Simha, D. D. Sundararaju, D. D. Ravindranath, and K. G. Shatry, "AUTOLAY - An Interactive Graphics System for the Design of Aircraft Composite Components" in *Graphics, Design and Visualization* ed. by S. P. Mudur and S. S. Pattanaik, Jaico Publishing House, Bombay, India pp 189-197 (1993)

[17] N. E. Dowling, *Mechanical Behavior of Materials*, Prentice-Hall, Inc. NY (1993)

[18] R. J. Bassett and R. Postle, "Fabric Mechanical and Physical Properties – Part 4: The Fitting of Woven Fabrics To a Three-Dimensional Surface," *International Journal of Clothing Science and Technology*, Vol. 2, No. 1, pp. 26-31 (1990)

[19] R. L. Burden and J. D. Faires, *Numerical Analysis*, Fourth Edition, PWS-KENT Publishing Co. Boston, MA (1989)

[20] J. Lindberg, B. Behre, and B. Dahlberg, "Part III: Shearing and Buckling of Various Commercial Fabrics," *Textile Research Journal*, Vol. 31, February, pp. 99-122 (1961)

[21] S. P. Timoshenko and J. N. Goodier, *Theory of Elasticity*, Third Edition, McGraw-Hill, Inc. (1982)