Rational Motion–based Surface Generation

Bert Jüttler
University of Technology, Darmstadt, Dept. of Mathematics
Schloßgartenstr. 7, 64289 Darmstadt, Germany
Phone: +49 6151 16 2789, Fax: +49 6151 16 2131
Email: juettler@mathematik.tu-darmstadt.de

Michael G. Wagner
Arizona State University, Dept. of Computer Science and Engineering
Box 875406, Tempe, AZ 85287-5406, USA
Phone: +1 602 965 1735, Fax: +1 602 965 2751
Email: wagner@asu.edu

November 28, 2003

This article is devoted to motion–based techniques for generating NURBS surfaces. We present a highly accurate approximation of the rotation–minimizing frame (RMF) of a space curve, that leads to a RMF–based scheme for rational sweep surface modeling. Furthermore we study envelopes of moving developable surfaces emphasizing the special cases of a moving cylinder or cone of revolution.

Keywords: surface modeling, NURBS, sweep surface, enveloped surface, rational motion.

INTRODUCTION

Motion–based surface generation is a fundamental principle of shape construction. Surfaces can be generated by sweeping a profile curve (also called cross–section curve) along a given spine curve, see e.g. References 8, 15. As a special case of this construction, one obtains the so–called pipe surfaces, which are generated by a moving circle.

On the other hand, surfaces can be generated as envelopes of moving objects. As an important special case one gets developable surfaces; they are the envelopes of moving planes, see e.g. Reference 1. Enveloped surfaces are particularly interesting in the context of milling and layered manufacturing, but also for the construction of gear tooth surfaces.

The introduction of NURBS as the standard representation for geometric data in CAD systems has made many new shape features available, see Reference 9. Using NURBS one may exactly describe surfaces of revolution, quadric surfaces such as ellipsoids and hyperboloids, developable surfaces, and also sweeping surfaces. Piecewise rational (or NURBS) motions (see Reference 7) are the most appropriate tool for developing NURBS techniques for motion–based surface generation.

The first part of the present paper is devoted to sweeping surfaces. We present a rational approximation scheme for the rotation minimizing frame (RMF) of a spine curve. In geometric modeling, this frame has been introduced by Klok[8]. Based on biarc techniques, Wang and Joe[15] have recently developed an elegant approximation scheme. The newly developed rational scheme as described below improves the accuracy of the approximation to the RMF. It produces NURBS representations of sweeping surfaces that are pieced together of segments of degree $(6, k)$, where k is the degree of the profile (or cross–section) curve. Unlike the biarc scheme, it produces a true C^1 motion, which gives smooth sweeping surfaces also for non–planar profile curves. The scheme

can readily be modified to generate sweeping surfaces matching more general input data, such as a sequence of positions of the profile curve.

The second part of the paper deals with the construction of envelopes of moving developable surfaces. Developable surfaces are envelopes of one parameter sets of planes. This includes cylinders and cones as degenerate cases. If both the moving surface and the motion are rational, the resulting envelope will be a rational TP NURBS surface. This extends the ideas introduced in Reference 7 for moving polyhedra to a much more general surface type.

BASICS FROM KINEMATICS

The points of Euclidean 3–space are described by their coordinates $\underline{\mathbf{p}} = (\underline{p}_1\ \underline{p}_2\ \underline{p}_3)^\top$ with respect to a Cartesian coordinate system. Sometimes, however, it will be advantageous to use the *homogeneous coordinates* $\mathbf{p} = (p_0\ p_1\ p_2\ p_3)^\top$ instead, $\mathbf{p} \neq (0\ 0\ 0\ 0)^\top$. If the homogeneous coordinates of a point are given, then the corresponding Cartesian coordinates are $\underline{p}_i = p_i/p_0$, $i = 1, 2, 3$. Conversely, the possible homogeneous coordinates of the point $\underline{\mathbf{p}}$ are $\mathbf{p} = (\lambda\ \lambda\underline{p}_1\ \lambda\underline{p}_2\ \lambda\underline{p}_3)^\top$ with $\lambda \in \mathbb{R}$, $\lambda \neq 0$. The coefficient λ is sometimes referred to as the *weight* of the point $\mathbf{p}$. The homogeneous coordinate vectors with $p_0 = 0$ correspond to *points at infinity*; they can be identified with the equivalence classes of parallel lines. See References 9, 3, 5 for more informations on weights, homogeneous coordinates, and NURBS techniques.

With the help of homogeneous coordinates, the equation of a plane in 3–space,

$$L_0 + L_1\underline{p}_1 + L_2\underline{p}_2 + L_3\underline{p}_3 = 0 \qquad (1)$$

(with certain constant coefficients $L_i \in \mathbb{R}$), can be rewritten as

$$L_0\,p_0 + L_1\,p_1 + L_2\,p_2 + L_3\,p_3 = \mathbf{L}^\top\mathbf{p} = 0. \qquad (2)$$

$\mathbf{L} = (L_0\ L_1\ L_2\ L_3)^\top \in \mathbb{R}^4$, $\mathbf{L} \neq \mathbf{0}$, is called the *homogeneous coordinate vector of the plane.* Any two linearly dependent coordinate vectors correspond to the same plane. The vector $(1\ 0\ 0\ 0)^\top$ describes the *plane at infinity* which is formed by all points with $p_0 = 0$.

In order to study motions and motion–based surface generation, we take two copies of the Euclidean 3–space. The first one is called the *fixed space*, its points and planes will be denoted with $\mathbf{p}$ (resp. $\underline{\mathbf{p}}$) and $\mathbf{P}$. These coordinates are often called the *world coordinates.* The second copy is called the *moving space*, (with the points $\widehat{\ }$ $\underline{\ }$) and planes $\widehat{\mathbf{P}}$. For instance, the moving space can be identified with a moving Cartesian coordinate systems which is moved along a certain curve of the fixed space. At each instant t, the moving space has a certain position in the fixed space. This is described by the coordinate transformation between both spaces,

$$\widehat{\mathbf{p}} \mapsto \mathbf{p}(t) = \underbrace{\left(\begin{array}{c|ccc} 1 & 0 & 0 & 0 \\ \hline \underline{\mathbf{u}}(t) & & R(t) & \end{array}\right)}_{M(t)} \widehat{\mathbf{p}}. \qquad (3)$$

Here, the real 3×3–matrix $R = R(t)$ is a special orthogonal matrix for all t. It represents the difference of the orientations of the fixed and the moving space. The vector $\underline{\mathbf{u}}(t)$ specifies the origin of the moving space in world coordinates.

At the instant $t = t_0$, the point $\widehat{\mathbf{p}}$ of the moving space has the world coordinates $\mathbf{p}(t_0)$, it coincides with that point of the fixed space. The curve $\mathbf{p}(t)$, $t \in \mathbb{R}$, is called the *path* or *trajectory* of $\widehat{\mathbf{p}}$. In particular, the origin of the moving system travels along the curve $\underline{\mathbf{u}}(t)$, $t \in \mathbb{R}$.

In addition to (3), there is another transformation which applies to the *planes* of the moving system. At the instant t, the plane $\widehat{\mathbf{P}}$ of the moving system coincides with the plane

$$\widehat{\mathbf{P}} \mapsto \mathbf{P}(t) = \underbrace{\left(\begin{array}{c|c} 1 & -\underline{\mathbf{u}}(t)^\top R(t) \\ \hline 0 & \\ 0 & R(t) \\ 0 & \end{array}\right)}_{M^*(t)} \widehat{\mathbf{P}} \qquad (4)$$

in world coordinates. In fact, a short calculation confirms that $\mathbf{P}^\top\mathbf{p} = \widehat{\mathbf{P}}^\top M^{*\top} M\widehat{\mathbf{p}} = \widehat{\mathbf{P}}^\top\widehat{\mathbf{p}}$.

The rotation matrix $R(t)$ can be described with the help of its homogeneous *Euler parameters* $(d_0, d_1, d_2, d_3) \in \mathbb{R}^4$, $\neq \{(0,0,0,0)\}$,

$$R(t) = \frac{1}{\Delta}\begin{pmatrix} d_0^2+d_1^2-d_2^2-d_3^2 & 2(d_1d_2-d_0d_3) & 2(d_1d_3+d_0d_2) \\ 2(d_1d_2+d_0d_3) & d_0^2-d_1^2+d_2^2-d_3^2 & 2(d_2d_3-d_0d_1) \\ 2(d_1d_3-d_0d_2) & 2(d_2d_3+d_0d_1) & d_0^2-d_1^2-d_2^2+d_3^2 \end{pmatrix},$$
$$\Delta = d_0^2+d_1^2+d_2^2+d_3^2, \tag{5}$$

see Bottema and Roth[2]. For any choice of these parameters, we obtain a special orthogonal matrix R. The Euler parameters will be said to be normalized if $\Delta = 1$ holds. The Euler parameters

$$(d_0, d_1, d_2, d_3) = (\cos\tfrac{\phi}{2}, \sin\tfrac{\phi}{2}v_1, \sin\tfrac{\phi}{2}v_2, \sin\tfrac{\phi}{2}v_3)$$

correspond to the rotation with axis $\vec{\mathbf{v}} = (v_1\ v_2\ v_3)^\top$, $\|\vec{\mathbf{v}}\| = 1$ and angle ϕ. Any two linearly dependent quadruples of Euler parameters correspond to the same rotation matrix.

The normalized Euler parameters of a rotation matrix can be identified with the components of the corresponding unit quaternion. Many quaternion–based techniques for orientation interpolation in computer graphics and robotics are available, see Reference 13 for references.

A general motion is obtained by choosing in (3) and (5) arbitrary functions $\underline{u}_1(t), \ldots, \underline{u}_3(t)$ (which specify the trajectory of the origin) and $d_0(t), \ldots, d_3(t)$ (representing the rotation), with suitable order of differentiability. A remarkable class of motions is formed by the so–called *rational motions* which are distinguished by the property that the trajectories of all points are rational curves. That is, the trajectories can be represented as NURBS curves. Rational motions can be obtained from (3) and (5) by choosing not arbitrary functions but polynomials $\underline{u}_1(t), \ldots, \underline{u}_3(t)$ and $d_0(t), \ldots, d_3(t)$. One may even choose *rational* functions $\underline{u}_1(t), \ldots, \underline{u}_3(t)$. By choosing the above parameters as (rational) spline functions, we obtain a rational spline motion. Various methods for computer aided design with rational spline motions have been developed in Reference 7. In particular, an interpolation scheme is described there.

SWEEP SURFACE MODELING

Sweeping of profile curves along a given spine curve is a powerful method to generate surfaces in computer graphics and geometric modeling, see Wang and Joe[15] and the references cited therein. We recall some results concerning the rotation minimizing frame (RMF) of space curve and derive a highly accurate rational approximation scheme. This leads to an algorithm for RMF–based sweep surface modeling with NURBS.

Sweeping surfaces and frames

Consider a spatial motion $M = M(t)$, see (3). We take a rigid curve $\widehat{\mathbf{p}}(s)$, $s \in [0,1]$, in the moving space and consider the surface

$$\mathbf{x}(t,s) = M(t)\,\widehat{\mathbf{p}}(s). \tag{6}$$

This surface is called a (general) sweeping surface. The generating curve $\widehat{\mathbf{p}}(s)$ is referred to as the profile curve (or cross–section curve). The parameter lines $s =$ constant are point trajectories, and the lines $t =$ constant are simply copies of the profile curve. See Figure 1.12 of Bézier's preface to Reference 3 for a nice illustration of sweep surface generation.

Now consider another curve $\underline{\mathbf{q}}(t)$ in the fixed space. Any motion M(t) that satisfies the conditions

$$R(t)\begin{pmatrix}0\\0\\1\end{pmatrix} = \frac{\dot{\underline{\mathbf{q}}}(t)}{\|\dot{\underline{\mathbf{q}}}(t)\|} \quad \text{and} \quad \underline{\mathbf{u}}(t) = \underline{\mathbf{q}}(t) \tag{7}$$

for all t is called a *frame* with the *spine curve* $\underline{\mathbf{q}}(t)$. Frames are frequently used in sweep surface modeling, as they provide a simple possibility to associate a motion with a given curve. If a motion is a frame, then the origin of the moving system travels along the spine curve, and the xy–plane of the moving system is always mapped onto the normal plane of the spine curve.

There are several ways to associate a frame with a given spine curve. The well–known Frenet frame of a curve (see any textbook on differential geometry) is not suitable for applications, as the generated surface shapes often behave badly. The Frenet frame of a C^2 curve is generally only continuous, and points with vanishing curvature (inflections) may even cause undesired flippings of the Frenet frame. Thus, using the Frenet frame imposes severe restrictions on the spine curve. Even for C^2 space curves with non–zero curvature, the Frenet frame exhibits a certain amount of twist about the tangent which

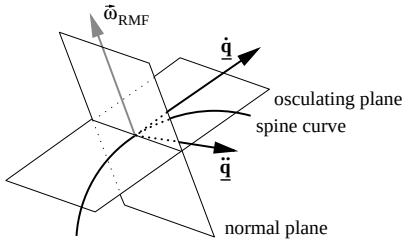


Figure 1: Angular velocity of the RMF.

may produce unnatural–looking sweeping surfaces, see Reference 15 for examples.

Klok[8] has proposed to use the so–called rotation minimizing frames instead. Recall that the angular velocity $\vec{\omega}(t) = (\omega_1\, \omega_2\, \omega_3)^\top$ of the motion (3) can be computed from

$$\dot{R}\, R^\top = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}, \qquad (8)$$

see Reference 2. The matrix product $\dot{R}\, R^\top$ yields a antisymmetric matrix as $R\, R^\top = I$ holds for all t. A frame of a curve is called a *rotation minimizing frame* of the spine curve $\underline{\mathbf{q}}(t)$, if and only if its angular velocity equals

$$\vec{\omega}_{\mathrm{RMF}}(t) = \frac{\dot{\underline{\mathbf{q}}}(t) \times \ddot{\underline{\mathbf{q}}}(t)}{(\dot{\underline{\mathbf{q}}}(t), \dot{\underline{\mathbf{q}}}(t))} \qquad (9)$$

with the inner vector product $(.,.)$, cf. Figure 1. In fact, if $\vec{\omega}$ is the angular velocity of a frame associated with the given spine curve $\underline{\mathbf{q}}$, then it has to satisfy the necessary condition*

$$\vec{\omega} \times \frac{\dot{\underline{\mathbf{q}}}}{\|\dot{\underline{\mathbf{q}}}\|} = \frac{d}{dt}\left(\frac{\dot{\underline{\mathbf{q}}}}{\|\dot{\underline{\mathbf{q}}}\|}\right).$$

The general solution of this equation can be shown to be

$$\vec{\omega}(t) = \vec{\omega}_{\mathrm{RMF}}(t) + \lambda(t)\, \dot{\underline{\mathbf{q}}}(t)$$

with the free parameter $\lambda = \lambda(t)$, where $\vec{\omega}_{\mathrm{RMF}}$ is defined as in (9). Thus, $\vec{\omega}_{\mathrm{RMF}}$ is the *smallest possible angular velocity* satisfying the necessary condition, as it is perpendicular to $\dot{\underline{\mathbf{q}}}$. See Reference 8 for a more detailed proof of this fact.

*Differentiating (7) gives $\dot{R}\, \vec{\mathbf{e}}_3 = (d/dt)(\dot{\underline{\mathbf{q}}}/\|\dot{\underline{\mathbf{q}}}\|)$. On the other hand, it leads to $\dot{R}\, \vec{\mathbf{e}}_3 = \dot{R}\, R^\top(\dot{\underline{\mathbf{q}}}/\|\dot{\underline{\mathbf{q}}}\|) = \vec{\omega} \times (\dot{\underline{\mathbf{q}}}/\|\dot{\underline{\mathbf{q}}}\|)$.

If a frame of a spine curve is given, then any other frame can be obtained by composing the frame with rotations around the tangent. That is, the difference vector between the angular velocities of any two frames is always linearly dependent on the tangent vector $\dot{\underline{\mathbf{q}}}$.

The rotation–minimizing frame associated with a given spine curve can be obtained by integrating the differential equation (8) with suitable initial conditions, after substituting the components of $\vec{\omega}_{\mathrm{RMF}}$ (see (9)) into the right–hand side. There would exist a solution, though generally not in a closed form. In the sequel we develop a rational approximation scheme for rotation minimizing frames.

Highly accurate rational approximation of the rotation minimizing frame

To begin with we consider a segment of the spine curve $\underline{\mathbf{q}}(t)$, $t \in [t_0, t_1]$, defined over a parameter interval of length $\Delta t = t_1 - t_0$. We choose a unit vector $\vec{\mathbf{v}}_0$ which is contained in the normal plane of the curve at $t = t_0$, i.e. $(\vec{\mathbf{v}}_0, \dot{\underline{\mathbf{q}}}(t_0)) = 0$. By choosing $\vec{\mathbf{v}}_0$ as the direction of the x–axis of the moving system, we specify the position of the frame at $t = t_0$. This leads to the orthogonal matrix

$$R_0 = R(t_0) = \left(\vec{\mathbf{v}}_0, \frac{\dot{\underline{\mathbf{q}}}(t_0) \times \vec{\mathbf{v}}_0}{\|\dot{\underline{\mathbf{q}}}(t_0)\|}, \frac{\dot{\underline{\mathbf{q}}}(t_0)}{\|\dot{\underline{\mathbf{q}}}(t_0)\|}\right) \qquad (10)$$

whose columns are obtained by collecting the three individual vectors. In order to find the corresponding motion of the RMF at the segment end point, we transport the vector $\vec{\mathbf{v}}_0$ along the curve segment with the help of Klok's method[8], see Algorithm 1. This can be seen as numerical integration of the differential equation

$$\dot{\vec{\mathbf{w}}}(t) = \vec{\omega}_{\mathrm{RMF}}(t) \times \vec{\mathbf{w}}(t)$$

of the rotation minimizing frame[8] with the step length $\Delta t/N$. Clearly, Klok's method is a rather coarse approximation of the differential equation. One may use more sophisticated integration techniques, e.g., the biarc scheme developed by Wang and Joe[15] or appropriate tools from numerical analysis.

As the result, we obtain a vector $\vec{\mathbf{v}}_1$ which is obtained by rotation minimizing transportation of

Algorithm 1 (Klok[8]):
Rotation–minimizing transportation of the vector $\vec{\mathbf{v}}_0$ along the spine curve $\underline{\mathbf{q}}(t)$.
INPUT: vector $\vec{\mathbf{v}}_0$, number of steps N;
$\vec{\mathbf{w}} = \vec{\mathbf{v}}_0$;
`for` i `from` 1 `to` N `do`
$\vec{\mathbf{t}} = \dot{\underline{\mathbf{q}}}(t_0 + \frac{i}{N}\Delta t)$; $\vec{\mathbf{t}} = \vec{\mathbf{t}}/\|\vec{\mathbf{t}}\|$;
$\vec{\mathbf{w}} = \vec{\mathbf{w}} - (\vec{\mathbf{w}}, \vec{\mathbf{t}})\vec{\mathbf{t}}$;
`od`;
$\vec{\mathbf{v}}_1 = \vec{\mathbf{w}}/\|\vec{\mathbf{w}}\|$;
OUTPUT: vector $\vec{\mathbf{v}}_1$.

$\vec{\mathbf{v}}_0$ from $\underline{\mathbf{q}}(t_0)$ to $\underline{\mathbf{q}}(t_1)$. This vector specifies the corresponding position of the RMF at $t = t_1$; it defines the orthogonal matrix

$$R_1 = R(t_1) = \left(\vec{\mathbf{v}}_1, \frac{\dot{\underline{\mathbf{q}}}(t_1) \times \vec{\mathbf{v}}_1}{\|\dot{\underline{\mathbf{q}}}(t_1)\|}, \frac{\dot{\underline{\mathbf{q}}}(t_1)}{\|\dot{\underline{\mathbf{q}}}(t_1)\|}\right).$$

With the help of Algorithm 2 we compute the normalized Euler parameters $\mathbf{d}_0 = (d_{0,0}, d_{0,1}, d_{0,2}, d_{0,3})$, $\mathbf{d}_1 = (\ldots)$ of the orthogonal matrices R_0, R_1. The sign of the Euler parameters is adjusted so that $(\mathbf{d}_0, \mathbf{d}_1) \geq 0$ holds.

The angular velocities $\vec{\omega}_0$ and $\vec{\omega}_1$ of the RMF at $t = t_0, t_1$ can be computed from (9). Now we apply Algorithm 3 to the C^1 Hermite boundary data $(\mathbf{R}_0, \vec{\omega}_0)$ and $(\mathbf{R}_1, \vec{\omega}_1)$. This gives the rational Bézier representation of the orthogonal matrices

$$R(t_0 + \tau\Delta t) = \sum_{i=0}^{6} A_i \, H_i^6(\tau) \qquad (11)$$

with the local parameter $\tau \in [0, 1]$ and the rational basis functions

$$H_i^6(\tau) = w_i \, B_i^6(\tau) / \sum_{j=0}^{6} w_j \, B_j^6(\tau), \quad i = 0, \ldots, 6,$$

where $B_l^k(\tau)$ are the standard Bernstein polynomials. By choosing cubic Euler parameters (or quaternions) $\mathbf{d}(\tau)$, we obtain from Euler's formula (5) interpolating orthogonal matrices of degree 6. They are described by the coefficient matrices $A_0, \ldots, A_6$ with the associated weights $w_0, \ldots, w_6$. With the help of formulas for the angular velocity, expressed in terms of quaternions (e.g. Reference 2, p. 512), it can easily be verified that they interpolate to the given C^1 data at $t = t_0$ and $t = t_1$. In addition, the construction of Algorithm 3 entails

$$w_0 = w_1 = w_5 = w_6 = 1.$$

Also the remaining weights are close to 1; this leads to a uniform distribution of the parametric speed. Note that the denominator of the rational basis functions is guaranteed to be positive, as it is the sum of the squared Euler parameters, see (5).

Now we convert the spine curve segment approximately into a rational Bézier curve of degree 6 with

Algorithm 2 (Weiss[16]):
Euler parameters of an orthogonal matrix.
INPUT: orthogonal matrix R;
$\mathbf{v}_1 = (1 + r_{1,1} + r_{2,2} + r_{3,3}, r_{3,2} - r_{2,3}, r_{1,3} - r_{3,1}, r_{2,1} - r_{1,2})$;
$\mathbf{v}_2 = (r_{3,2} - r_{2,3}, 1 + r_{1,1} - r_{2,2} - r_{3,3}, r_{1,2} + r_{2,1}, r_{1,3} + r_{3,1})$;
$\mathbf{v}_3 = (r_{1,3} - r_{3,1}, r_{1,2} + r_{2,1}, 1 - r_{1,1} + r_{2,2} - r_{3,3}, r_{2,3} + r_{3,2})$;
$\mathbf{v}_4 = (r_{2,1} - r_{1,2}, r_{1,3} + r_{3,1}, r_{2,3} + r_{3,2}, 1 - r_{1,1} - r_{2,2} + r_{3,3})$;
Choose the vector $\mathbf{d}$ from $\mathbf{v}_1, \ldots, \mathbf{v}_4$ with maximum length!
$\mathbf{d} := \mathbf{d}/\|\mathbf{d}\|$;
OUTPUT: normalized Euler parameters $\pm\mathbf{d}$, the sign is arbitrary.

Algorithm 3:
C^1 Hermite interpolation with orthogonal matrices.
INPUT: normalized Euler parameter $\mathbf{d}_0$, $\mathbf{d}_1$, angular velocities $\vec{\omega}_0$, $\vec{\omega}_1$;

$$\mathbf{v}_i = \begin{pmatrix} -\omega_{i,1} d_{i,1} - \omega_{i,2} d_{i,2} - \omega_{i,3} d_{i,3} \\ \omega_{i,1} d_{i,0} + \omega_{i,2} d_{i,3} - \omega_{i,3} d_{i,2} \\ \omega_{i,2} d_{i,0} - \omega_{i,1} d_{i,3} + \omega_{i,3} d_{i,1} \\ \omega_{i,3} d_{i,0} + \omega_{i,1} d_{i,2} - \omega_{i,2} d_{i,1} \end{pmatrix}, \; i = 0, 1;$$

$$\mathbf{d}(\tau) = B_0^3(\tau)\mathbf{d}_0 + B_1^3(\tau)(\mathbf{d}_0 + \tfrac{\Delta t}{6}\mathbf{v}_0) + B_2^3(\tau)(\mathbf{d}_1 - \tfrac{\Delta t}{6}\mathbf{v}_1) + B_3^3(\tau)\mathbf{d}_1;$$

Compute the rational Bézier representation (11) from Euler's formula (5)! Explicit formulas for the weights and coefficient matrices are given in Reference 7.
OUTPUT: coefficient matrices $A_0, \ldots, A_6$ with associated weights $w_0, \ldots, w_6$.

Algorithm 4:
Conversion of the spine curve into a sextic Bézier curve with the prescribed weights w_i.

$\underline{\mathbf{b}}_0 = \underline{\mathbf{q}}(t_0)$; $\underline{\mathbf{b}}_1 = \underline{\mathbf{b}}_0 + (\Delta t/6w_0w_1)\dot{\underline{\mathbf{q}}}(t_0)$;

$\underline{\mathbf{b}}_6 = \underline{\mathbf{q}}(t_1)$; $\underline{\mathbf{b}}_5 = \underline{\mathbf{b}}_6 - (\Delta t/6w_5w_6)\dot{\underline{\mathbf{q}}}(t_1)$;

$A = (\ \int_0^1 H_i^6 H_j^6\ d\tau\ )_{i,j=2,\ldots,4}$;

$C = (\ \int_0^1 (\underline{q}_j - \sum_{i=0,1,5,6} H_i^6\ \underline{b}_{i,j}) H_k^6\ d\tau\ )_{j=1,\ldots,3;k=2,\ldots,4}$;

The integrals are evaluated numerically.

Find components of the inner control points from $(\ \underline{b}_{i,j}\ )_{j=1,\ldots,3;k=2,\ldots,4} = A^{-1}C$;

OUTPUT: control points $\underline{\mathbf{b}}_0, \ldots, \underline{\mathbf{b}}_6$.

the prescribed weights $w_0, \ldots, w_6$,

$$\underline{\mathbf{u}}(t_0 + \tau\Delta t) = \sum_{i=0}^{6} \underline{\mathbf{b}}_i\ H_i^6(\tau).$$

The control points $\underline{\mathbf{b}}_0$, $\underline{\mathbf{b}}_1$, $\underline{\mathbf{b}}_5$, $\underline{\mathbf{b}}_6$ are computed from the C^1 boundary conditions

$$\begin{aligned} \underline{\mathbf{u}}(t_0) &= \underline{\mathbf{q}}(t_0),\ \dot{\underline{\mathbf{u}}}(t_0) = \dot{\underline{\mathbf{q}}}(t_0), \\ \underline{\mathbf{u}}(t_1) &= \underline{\mathbf{q}}(t_1),\ \dot{\underline{\mathbf{u}}}(t_1) = \dot{\underline{\mathbf{q}}}(t_1). \end{aligned}$$

The remaining inner control points are found by minimizing the L^2 norm of the difference between the original spine curve and its approximation,

$$F(\underline{\mathbf{b}}_2, \underline{\mathbf{b}}_3, \underline{\mathbf{b}}_4) = \int_{t_0}^{t_1} \|\underline{\mathbf{u}}(t) - \underline{\mathbf{q}}(t)\|^2\ dt.$$

They are computed by solving a 3×3 system of linear equations, see Algorithm 4.

Summing up, we have constructed a rational motion of degree 6,

$$M(t_0+\tau\Delta t) = \sum_{i=0}^{6} \left(\begin{array}{c|ccc} 1 & 0 & 0 & 0 \\ \hline \underline{\mathbf{b}}_i & \multicolumn{3}{c}{A_i} \end{array}\right) H_i^6(\tau) \quad (12)$$

which approximates the rotation minimizing frame of the segment of the spine curve. It satisfies C^1 boundary conditions at $t = t_0$ and $t = t_1$.

In order to generate a highly accurate approximation to the rotation minimizing frame, one may decompose the spine curve into smaller segments, $t \in [t_i, t_{i+1}]$, with certain knots $(t_i)_{i=0,..,m}$. The above construction is applied to the individual segments. This leads to a rational C^1 spline motion of degree 6 which approximates the RMF.

Now consider a profile curve $\widehat{\mathbf{p}}(s)$. For instance, the profile curve can be chosen as a NURBS curve. The generated sweeping surface (6) is a NURBS surface patch; its control points can easily be generated by combining (12) with the NURBS representation of the profile curve.

As an example we consider the sweeping surface which is shown in Figure 2; it is traced out by a quadratic rational Bézier profile curve. This surface has been generated by the rational approximation with 4 segments to the rotation minimizing frame of the spine curve from Figure 3. The sweeping surface is a tensor–product NURBS patch with 4×1 segments of degree (6,2); both the surface and its Bézier control net are shown in Figure 2.

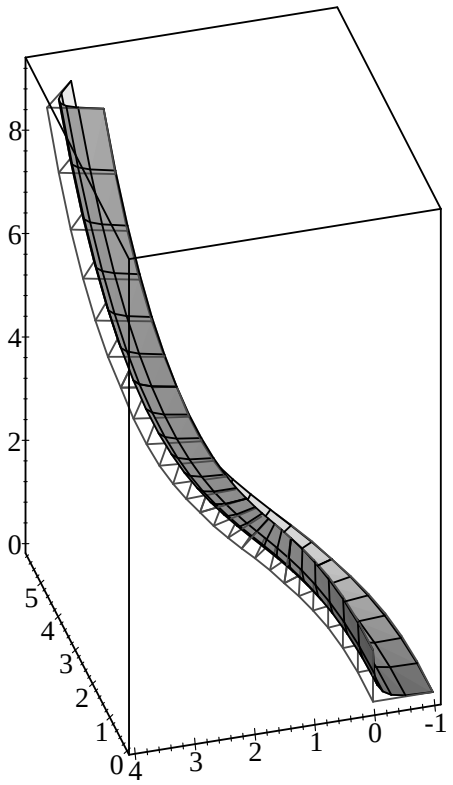


Figure 2: Sweeping surface generated by a rational approximation to the RMF of a given spine curve.

Figure 3 visualizes the accuracy of the approximation to the RMF. The angular velocity vectors of the rational approximation (black lines) and of the exact RMF (grey lines, see (9)) have been plotted along the given spine curve. There is only little difference between both vector fields. Also, the rational approximation can clearly be seen to be a C^1 motion, as the angular velocity is continuous.

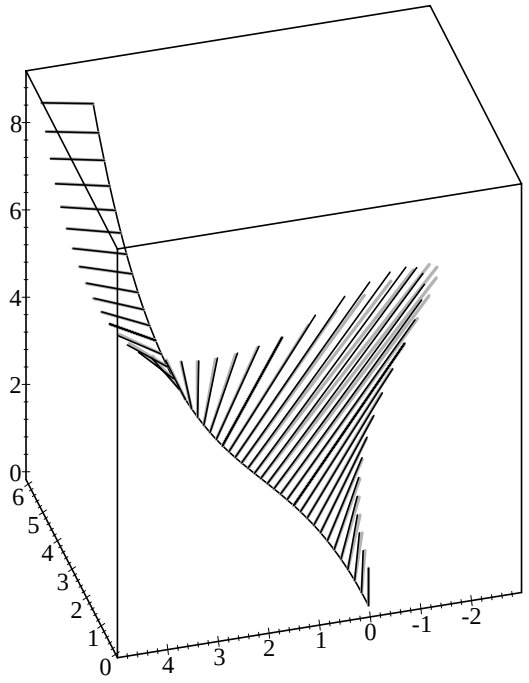


Figure 3: Angular velocity of the RMF (grey) and of its rational approximation along the spine curve.

Finally, the accuracy of the approximation to the spine curve is illustrated by Figure 4, which shows the parametric distance function $\|\underline{\mathbf{u}}(t) - \underline{\mathbf{q}}(t)\|$. Clearly, this distance function is somewhat misleading, as it depends on the parameterization of the curves. However, it gives an upper bound of the maximum distance error. Also, in applications it is desirable to approximate not only the curve, but also its parametric speed distribution. Thus, it is well justified to use the parametric distance function as an error measure. Note that the distance error shown in the Figure is already in the order of the numerical noise.

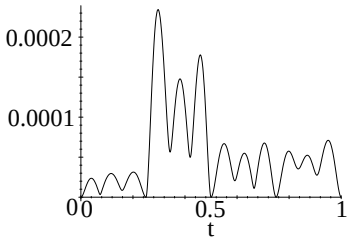


Figure 4: Distance error of the rational approximation to the spine curve.

Comparison with the biarc scheme

Recently, Wang and Joe[15] have described another approximation to the rotation minimizing frame of a spine curve. The given spine curve is converted into a C^1 biarc spline curve, consisting of so–called equal chord biarcs. The converted spline curve consists of circular segments. Hence, the rotation minimizing frame of each segment can be computed exactly; it is simply the rotation which generates the circular arc. Thus, one gets an approximation to the RMF of the original spine curve which consists of segments of rotations; the axes are those of the circular segments. Consequently, the sweeping surface which is generated by a moving profile curve is approximated by segments of surfaces of revolution.

With the help of the quadratic NURBS representation of the circular arcs one obtains a quadratic rational motion which approximates the RMF, see Reference 15. Generally, this approximation is only a C^0 motion, as the axes of adjacent circular arcs may be different. Nevertheless, a profile curve which travels in the normal plane of the spine curve generates a G^1 sweeping surface. This, however, is not true for general profile curves.

In order to compare our rational approximation with the biarc scheme, we have computed the biarc approximation to the RMF for the spine curve from Figure 3. We used a biarc spline with 6 segments, i.e., which consists of 12 circular arcs. This leads to sweeping surfaces which have a similar data volume as the rational approximation with only 4 segments. Using the biarc approximation, the trajectory of a point is described by $12 \cdot 2 + 1 = 25$ control points. The rational approximation needs $4 \cdot 6 + 1 = 25$ control points too. Here, the boundary control points of each segment are counted only once.

Similar to Figure 3, Figure 5 visualizes the accuracy of the approximation to the RMF. The angular velocity vectors of the biarc approximation (black lines) and of the exact RMF (grey lines, see (9)) have been plotted along the given spine curve. The direction of the biarc angular velocity is constant for each circular segment, but its length may vary, due to the quadratic NURBS representation. Even if the number of circular segments (12) is relatively large, there difference between the exact angular velocity and that of the biarc approximation is still fairly big. Also, it can clearly be seen that the biarc approximation is far from being a C^1 motion.

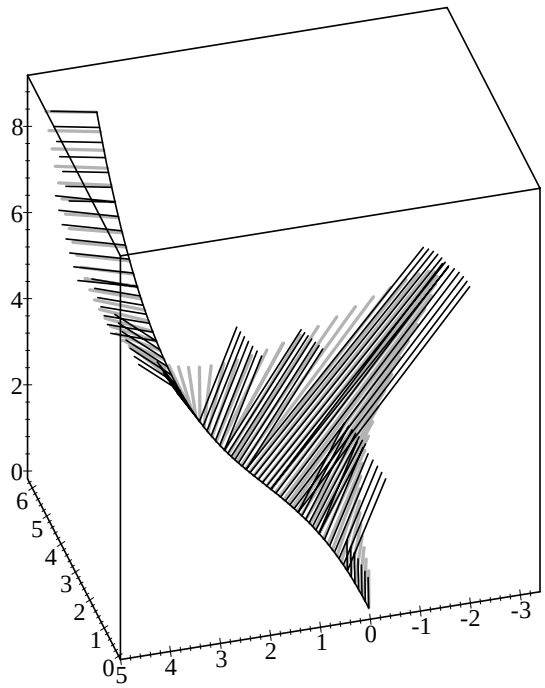


Figure 5: Angular velocity of the RMF (grey) and of its biarc approximation along the spine curve, compare with Fig. 3.

The accuracy of the approximation to the spine curve is illustrated by Figure 6, which shows the distance function $\|\underline{\mathbf{b}}(t) - \underline{\mathbf{q}}(t)\|$, where $\underline{\mathbf{b}}(t)$ is the biarc approximation. Again, the *parametric* distance function has been plotted. Nevertheless it can clearly be seen, that the maximum distance error is now in the order of ≈ 0.15, compared with 0.0002 for the rational approximation scheme.

As a second example, the RMF of a helix segment has been approximated, both with the rational and the biarc scheme. The results are shown in Figure 7. It shows a sweeping surface with 2×1 segments which is generated by the rational approximation (a), the comparison between the exact angular velocity of the RMF and its approximations (b), and the parametric distance error functions (c). The rational motion consists of two segments of degree 6. This is to be compared with a biarc approximation with 3 biarc segments, i.e., with 6 circular arc segments. Again, both the rational and the biarc approximation have a similar data volume.

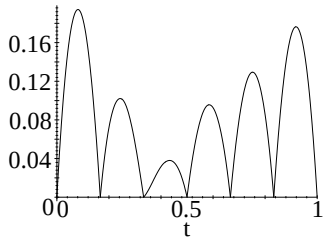


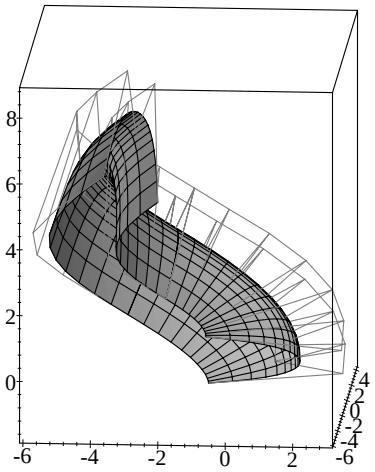
Figure 6: Distance error of the biarc approximation to the spine curve, compare with Fig. 4.

Clearly, the rational approximation is much more precise than the biarc one. For the rational scheme, the distance error along the spine curve is already in the order of the numerical noise.

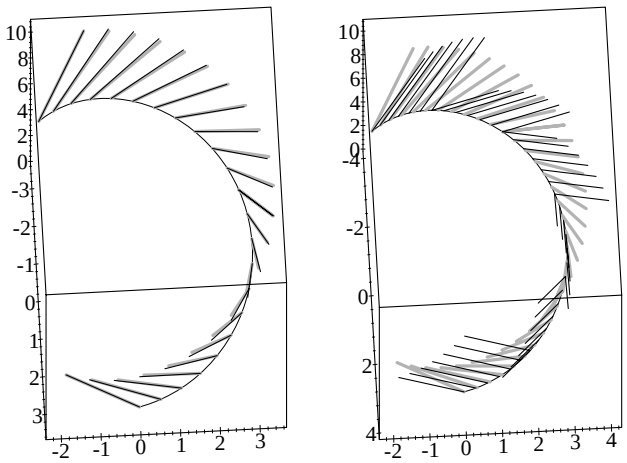
Summing up, the rational approximation method yields a high quality approximation of the rotation minimizing frame. Unlike the biarc scheme, it produces a true C^1 motion, which generates smooth surfaces also for profile curves which are not contained in the normal plane of the spine curve. In addition, the rational scheme can easily be modified in order to generate sweeps matching more general data. This is explained in the next section.

Modified RMF sweeping surfaces

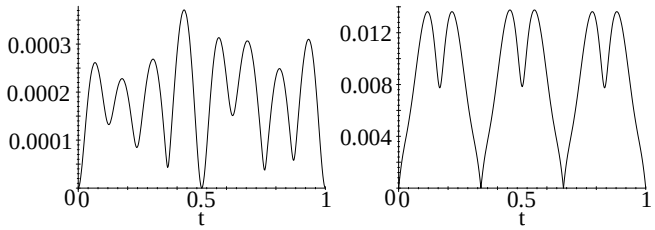
In most applications, using only the RMF of a spine curve will not be sufficient, because the sweeping surface is then fully determined by a single given position of the profile curve. Consequently, it is generally impossible to model closed surfaces with RMF sweeps. In order to overcome these problems, we introduce a modification of the RMF.



(a) Sweeping surface generated by the rational approximation to the RMF.



(b) Angular velocity of the RMF (grey), of its rational approximation (left) and of the biarc approximation (right).



(c) Distance error of the rational approximation (left) and of the biarc approximation (right).

Figure 7: The rational and the biarc approximation to the RMF of a helix.

Along with the spine curve, we assume that the position of the profile curve has been specified by the designer at some points $\underline{\mathbf{q}}(t_0)$, $\underline{\mathbf{q}}(t_1), \ldots, \underline{\mathbf{q}}(t_m)$. Each position, however, is still assumed to be a frame of the spine curve, i.e., the z–axis of the moving coordinate system is to have the same direction as the tangent of the curve. (See References 6, 12 for constructions of even more general sweeping surfaces.) Under this additional assumption, each position can be described by the angle β_j, $j = 0, \ldots, m$, of the rotation around the tangent of the spine curve, which generates the position from that of the RMF. That is, the position of the frame at $t = t_i$ is specified by the orthogonal matrix

$$\left(\vec{\mathbf{v}}_j, \frac{\dot{\underline{\mathbf{q}}}(t_j) \times \vec{\mathbf{v}}_j}{\|\dot{\underline{\mathbf{q}}}(t_j)\|}, \frac{\dot{\underline{\mathbf{q}}}(t_j)}{\|\dot{\underline{\mathbf{q}}}(t_j)\|} \right) \underbrace{\begin{pmatrix} \cos\beta_j & -\sin\beta_j & 0 \\ \sin\beta_j & \cos\beta_j & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{= Z(\beta_j)}, \tag{13}$$

cf. (10). The given data is illustrated by Figure 8, where the given positions of the profile curve have been plotted along with the x–axes of the rotation minimizing frame (dotted) and of the modified RMF (solid).

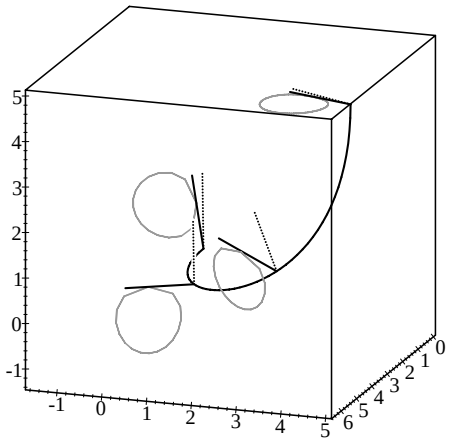


Figure 8: The given profile curves for the modified RMF sweeping surface.

The modified frame of the spine curve results from the RMF by a suitable rotation around the tangent of the curve. Its rotational part is described by the orthogonal matrix

$$R_{\mathrm{RMF}}(t) \;\; Z(\beta(t)) \tag{14}$$

with the matrix $R_{\mathrm{RMF}}(t)$ of the rotation minimizing frame and $Z(\beta)$ as in (13). In order to find a suitable function $\beta = \beta(t)$, we interpolate the given angles β_j with a natural cubic spline (see Reference 3) with the knots $(t_j)_{j=0,\ldots,m}$. For modeling closed surfaces one may use periodical splines instead. Also, it is possible to add multiples of 2π to the given angles, without altering the given positions. We choose the angles β_j so that the absolute values of the differences between adjacent angles is smaller than 2π.

In order to increase the accuracy of the rational approximation, one may wish to insert additional knots t_i into the original knot sequence, where no associated angles β_i are given. These angles can then be computed by sampling values from the natural cubic spline that interpolates the given angles.

Now we are ready to generate a rational approximation to the modified rotation minimizing frame. In addition to the positions (13), we compute the angular velocities of the modified RMF (14),

$$\vec{\omega}_j = \underbrace{\frac{\dot{\underline{\mathbf{q}}}(t_j) \times \ddot{\underline{\mathbf{q}}}(t_j)}{(\dot{\underline{\mathbf{q}}}(t_j), \dot{\underline{\mathbf{q}}}(t_j))}}_{= \vec{\omega}_{\mathrm{RMF}}} + \dot{\beta}(t_j) \, \frac{\dot{\underline{\mathbf{q}}}(t_j)}{\|\dot{\underline{\mathbf{q}}}(t_j)\|}.$$

Using the above Hermite interpolation scheme (Algorithms 2,3, and 4) we may now find a sextic rational approximation to the modified RMF. As an example, Figure 9 shows a sweeping surface which is generated by the rational approximation to the modified RMF for the data from Figure 8. The surface is a NURBS patch with 3×1 segments of degree $(6, 4)$.

Clearly, the above–described sweeping scheme can easily be generalized by allowing the shape of the profile curve to evolve during the motion. See References 12, 14 for examples.

ENVELOPED SURFACES

An important task in the simulation of manufacturing processes is the determination of enveloping surfaces of moving objects, for example of the cutter of a milling machine as illustrated in Figure 10. This section shows how to compute the envelope of a moving developable NURBS surface covering the important cases of a moving cylinder or cone.

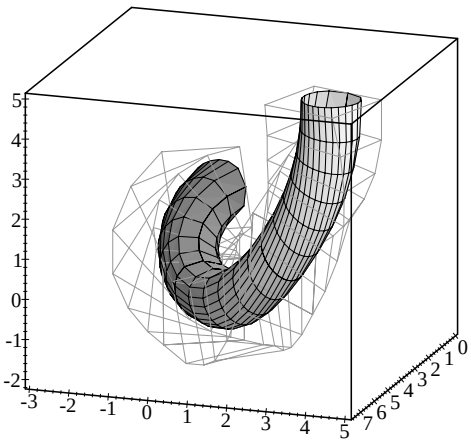


Figure 9: Modified RMF sweeping surface.

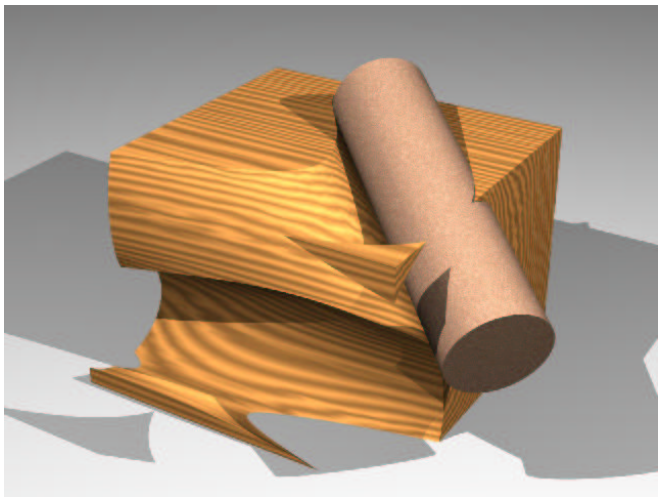


Figure 10: Milling with a cylindrical cutting tool.

Moving planes

According to equation (2) it is possible to represent a plane in three dimensional space by a homogeneous coordinate vector $\mathbf{P} \in \mathbb{R}^4$. Furthermore we have seen that a point $\mathbf{p}$ lies in the plane $\mathbf{P}$, if the scalar product $\mathbf{P}^\top \mathbf{p}$ vanishes.

Now consider a spatial rational motion $M = M(t)$. As noted earlier such a motion is described by a set of four polynomials $d_0(t), \ldots, d_3(t)$ and three rational functions $\underline{u}_1(t), \ldots, \underline{u}_3(t)$. In order to investigate the trajectory of a plane $\widehat{\mathbf{P}}$ in the moving system, one first has to convert into its *dual* form M^*. This can be done by applying equation (4) either to the above functions or to M itself.

We note that the resulting *dual rational motion* $M^* = M^*(t)$ is associated with a *dual control structure* analogous to the control structure of $M(t)$ as introduced in Reference 7. The corresponding control positions of the motion are described by the constant coefficient matrices A_i^* of M^* if M^* is converted into NURBS form. Figure 11 shows the control positions of a moving dice in red along with the dual control positions in green. Initial and end position of the cube belong to both sets of control positions.

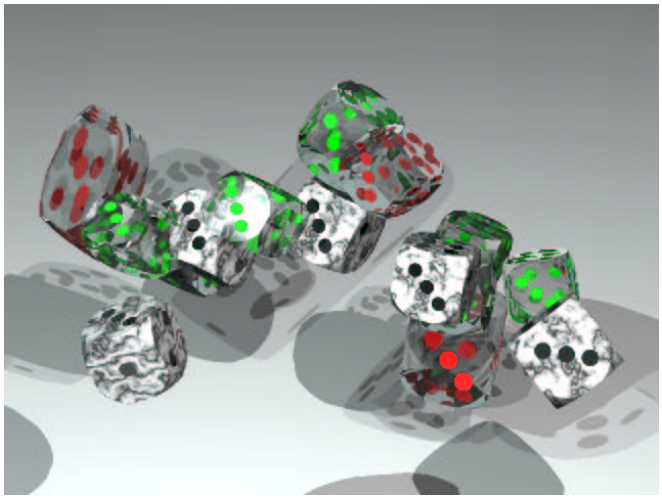


Figure 11: Dual control structure of a moving dice.

At any time instant t, the position of a plane $\widehat{\mathbf{P}}$ is given by

$$\widehat{\mathbf{P}} \mapsto \mathbf{P}(t) = M^*(t)\widehat{\mathbf{P}}. \qquad (15)$$

The set of all position of $\widehat{\mathbf{P}}$ forms a one parameter set/family of planes $\mathbf{P}(t)$. Such a one parameter set of planes envelopes a developable surface, which in general is a tangent surfaces of a twisted curve $\mathbf{c}(t)$. In non-degenerate cases, the set $\mathbf{P}(t)$ is called the *dual representation* of $\mathbf{c}(t)$, and $\mathbf{c}(t)$ the *line of regression* of $\mathbf{P}(t)$, respectively.

In degenerate cases the curve $\mathbf{c}(t)$ can also contain pieces of conical, cylindrical or planar surfaces. As an example, consider the plane

$$\mathbf{P}_{cy}(t) = (\ -r \quad \cos(t) \quad \sin(t) \quad 0\ )^\top \qquad (16)$$

for a fixed r. Obviously, $\mathbf{P}_{cy}(t)$ is parallel to the z-axis for every t. Furthermore the distance of $\mathbf{P}_{cy}$ to the point $\mathbf{o} = (0\,0\,0\,1)^\top$ is equal to r which is a constant. Hence, $\mathbf{P}_{cy}(t)$ is the dual representation of a cylinder of revolution. The axis of this cylinder is the z-axis of the coordinate system. A similar argument shows that

$$\mathbf{P}_{co}(t) = (\ 0 \quad \cos(t) \quad \sin(t) \quad -\tan(\alpha)\ )^\top \qquad (17)$$

describes the dual representation of a cone with apex $\mathbf{o} = (0\,0\,0\,1)^\top$, direction of axis $(0\ 0\ 1)^T$ and opening angle 2α.

Dual representations of twisted curves have been studied extensively by various authors (see Hoschek[4], Pottmann and Hoschek[11], Bodduluri and Ravani[1]) in the context of developable surface design. The most straightforward way to obtain $\mathbf{c}(t)$ from $\mathbf{P}(t)$ in the NURBS case, is to split the NURBS representation of $\mathbf{P}(t)$ into its rational parts and apply the algorithm of de Casteljau to convert the individual segments (see Pottmann[10]). This procedure is outlined in Algorithm 5. A similar technique was used in Reference 7 for computing the control positions of a rational motion. Alternatively, one could also use product formulae for B-spline basis functions. Due to the complexity of these products, however, the latter approach is only useful for theoretical purposes.

It has to be noted that Algorithm 5 assumes that the all segments of $\mathbf{P}(t)$ are tangent surfaces of twisted curves. In order to make the implementation stable one in addition has to check for degenerate cases for each individual segment. Further notice that line of regression $\mathbf{c}(t)$ and corresponding dual representation $\mathbf{P}(t)$ are dual to each other. Consequently, Algorithm 5 can also be used to calculate $\mathbf{P}(t)$ from $\mathbf{c}(t)$. One simply has to call the algorithm with $\mathbf{c}(t)$ as input. The corresponding output will be $\mathbf{P}(t)$ (see Pottmann[10]).

Moving developable surfaces

In general one is not interested in the motion of a single plane. One rather needs algorithms which allow the computation of enveloping surfaces of moving objects. In Reference 7 it was shown how to compute the enveloping surface of a moving polyhedra. In the sequel we will extend these ideas and describe

Algorithm 5:
Computation of the line of regression $\mathbf{c}(t)$ of a piecewise rational one parameter set of planes $\mathbf{P}(t)$.
INPUT: NURBS representation of $\mathbf{P}(t)$;
Convert $\mathbf{P}(t)$ into Bézier spline form by repeated knot insertion (see Reference 5), this results in N Bézier segments
$\mathbf{P}^i(\tau) = \sum_{j=0}^{n} B_j^n(\tau)\mathbf{Q}_j^i, j = 0, \ldots, N-1$
of degree n.
`for` i `from` 0 `to` $N-1$ `do`
  `for` j `from` 0 `to` n `do`
    $\mathbf{d}_j^i = \binom{6(n-1)}{j}^{-1} \sum_{a+b+c=j} \binom{2(n-1)}{a}\binom{2(n-1)}{b}\binom{2(n-1)}{c}\mathbf{Q}_a^i \times \mathbf{Q}_b^i \times \mathbf{Q}_c^i$;
  `od;`
  $\mathbf{c}^i(\tau) = \sum_{j=0}^{6(n-1)} B_j^{6(n-1)}(\tau)\mathbf{d}_j^i$;
`od;`
The exterior product of the vectors $\mathbf{Q}_a^i, \mathbf{Q}_b^i, \mathbf{Q}_c^i$ is defined by

$$\mathbf{A}\times\mathbf{B}\times\mathbf{C} = \left(\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}, - \begin{vmatrix} a_0 & b_0 & c_0 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}, \begin{vmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ a_3 & b_3 & c_3 \end{vmatrix}, - \begin{vmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{vmatrix} \right).$$

Collect all segments in a single NURBS curve $\mathbf{c}(t)$ and remove all redundant knots;
OUTPUT: NURBS representation of $\mathbf{c}(t)$.

the computation of the envelope of a moving rational developable surface. As mentioned above, these surface type includes cylinders and cones of revolution. It are these degenerate cases which makes the following study important for practical applications.

Let us consider a rational developable surface

$$\widehat{\mathbf{P}}(s) = \sum_{j=0}^{m} N_j^n(s)\,\widehat{\mathbf{Q}}_j \tag{18}$$

in dual coordinates. This surface $\widehat{\mathbf{P}}(s)$ is assumed to lie entirely in the moving space. Its NURBS representation (18) is given in terms of its knot vector and a set of $m+1$ control planes $\widehat{\mathbf{Q}}_j$. If $\widehat{\mathbf{P}}(s)$ is moved according to a rational motion $M(t)$ with dual representation $M^*(t)$, all positions of all planes of $\widehat{\mathbf{P}}(s)$ form a two parameter set of planes

$$\mathbf{P}(s,t) = M^*(t)\,\widehat{\mathbf{P}}(s) \tag{19}$$

which is rational in s and t. Therefore, $\mathbf{P}(s,t)$ describes the dual representation of a tensor product (TP) NURBS surface $\mathbf{c}(s,t)$ (cf. 3). There are various ways how $\mathbf{P}(s,t)$ can be converted into its point representation $\mathbf{c}(s,t)$. The most straightforward method results from intersecting the plane $\mathbf{P}(s,t)|_{s=s_0,t=t_0}$ with the first derivative planes in s and t direction, i.e.

$$\frac{\partial}{\partial s}\mathbf{P}(s,t)|_{s=s_0,t=t_0}, \qquad \frac{\partial}{\partial t}\mathbf{P}(s,t)|_{s=s_0,t=t_0}.$$

In analogy to algorithm 5 one first has to convert $\mathbf{P}(s,t)$ into an array of TP Bézier patches. Each individual patch then has to be converted into point form. After collecting all resulting Bézier patch one finally has to collect them in a single TP NURBS patch and delete all redundant knots. Alternatively, $\mathbf{P}(s,t)$ can also be converted into point form $\mathbf{c}(s,t)$ by applying algorithm 5 twice, once in s and a second time in t direction. This, however, requires the application of symbolic computational methods.

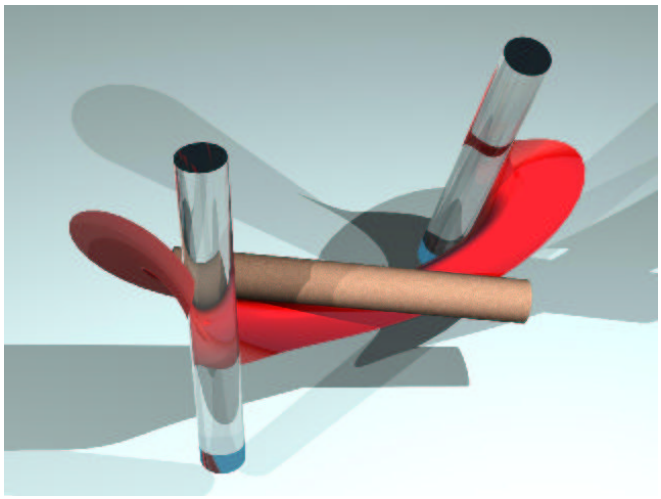


Figure 12: Envelope of a moving cylinder of revolution.

Notice that the developable surface $\widehat{\mathbf{P}}(s)$ is considered to extend infinitely. If one wants to prescribe bounds, such as top and bottom circle of a cylinder, one has to use methods similar to the one developed in Reference 7 for calculating the enveloping surface

of a moving polygon. Figure 12 shows an example of a surface patch enveloped by a moving cylinder. For a fixed $t = t_0$, the curve $c(s, t_0)$ contains all points at which the moving surface touches its envelope. This curve is called the characteristic curve of the position

$$\mathbf{P}_{t_0}(s) = M(t_0)\,\widehat{\mathbf{P}}(s)$$

of the moving surface at time instant t_0. A set of such parameter lines is illustrated in Figure 13.

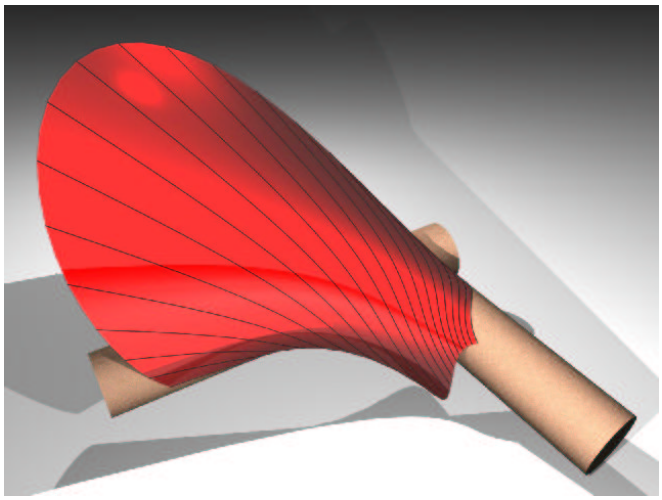


Figure 13: Set of characteristic curves on the envelope.

Although the computation of such envelopes is rather straightforward, exact methods lead to a relatively high degree of the TP NURBS representation. A cylinder which is subject to a rational motion of degree 6 as constructed earlier in this paper will generate a dual TP product surface of degree $(6, 2)$. It is easy to see that converting this surface into an ordinary TP NURBS surface will increase the degree to $(16, 4)$. One way to overcome this problem is to approximate the resulting surface by a surface constructed from appropriate patches such as a piecewise Coons surface. The information about the tangent planes can be obtained directly from $\mathbf{P}(s, t)$. Points on the surface on the other hand can be computed using de Boor's algorithm as mentioned in Reference 10.

REFERENCES

1. R.M.C. Bodduluri, B. Ravani, Design of developable surfaces using duality between plane and point geometries, *Comput.-Aided Des.* **25**, 621-632, 1993.

2. O. Bottema, B. Roth, *Theoretical Kinematics*, North–Holland, Amsterdam, 1979.

3. G. Farin, *Curves and surfaces for computer aided geometric design: a practical guide* (4th ed.), Academic Press, San Diego CA, 1996.

4. J. Hoschek, Dual Bézier curves and surfaces, in: R.E. Barnhill, W. Boehm (eds.), *Surfaces in Computer Aided Geometric Design*, North–Holland, 1983, 147–156.

5. J. Hoschek, D. Lasser, *Fundamentals of Computer Aided Geometric Design,* AK Peters, Wellesley MA, 1993.

6. B. Jüttler, Spatial rational motions and their application in computer aided geometric design, in M. Dæhlen, T. Lyche, L.L. Schumaker (eds.), *Mathematical methods for curves and surfaces*, Vanderbilt University Press, Nashville TN, 1995, 271–280.

7. B. Jüttler, M.G. Wagner, Computer Aided Design with Spatial Rational B–Spline Motions, *ASME J. of Mech. Design*, **118**, 193–201, 1996.

8. F. Klok, Two moving coordinate frames for sweeping along a 3D trajectory, *Comput. Aided Geom. Design*, **3**, 217-229, 1986.

9. L. Piegl, W. Tiller, *The NURBS book,* Berlin, Springer 1995.

10. H. Pottmann, Applications of the dual Bézier representation of rational curves and surfaces, in: P.J. Laurent, A. LeMéhauté, L.L. Schumaker (eds.), *Curves and Surfaces in Geometric Design*, AK Peters, Boston, 1994, 377–384.

11. H. Pottmann, J. Hoschek, Interpolation and approximation with developable B-spline surfaces, in: M. Dæhlen, T. Lyche, L.L. Schumaker (eds.), *Mathematical Methods for Curves*

and Surfaces, Vanderbilt University Press, Nashville TN, 1995, 255–264.

12. H. Pottmann, M.G. Wagner, Contributions to Motion Based Surface Design, *International Journal of Shape Modeling*, 1998, to appear.

13. O. Röschel, Rational motion design – a survey, *Comput.–Aided Design*, **30**, 169–178, 1998.

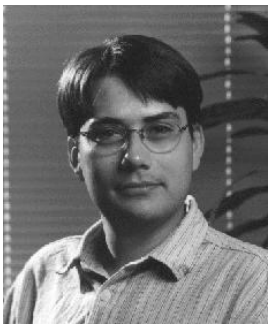
14. M.G. Wagner, B. Ravani, Curves with Rational Frenet–Serret motion, *Comput. Aided Geom. Design*, **15**, 79–101, 1997.

15. W. Wang, B. Joe, Robust computation of the rotation minimizing frame for sweep surface modeling, *Comput.–Aided Design*, **29**, 379–391, 1997.

16. E.A. Weiss, *Einführung in die Liniengeometrie und Kinematik*, Teubner, Leipzig, 1935.

Bert Jüttler studied Mathematics at the Universities of Technology in Dresden and Darmstadt. He graduated as a Diplom–Mathematiker in 1992 and he gained a PhD in 1994, both from Darmstadt. From 1994 to 1996 he was a postdoctoral research assistant in Darmstadt and at the University of Dundee, Scotland. Currently he is a Lecturer at the Department of Mathematics, Darmstadt University of Technology. His research interests include Computer Aided Geometric Design, Robotics and Kinematics.

(Photograph is attached.)



Michael G. Wagner received his M.S. in 1990 and a Ph.D. in mathematics in 1994 from Vienna University of Technology. From 1990 to 1997 he was Universitätsassistent at Vienna University of Technology, spending one year from as a Research Associate at University of California at Davis. Since 1997 he is Assistant Professor at the Computer Science and Engineering Department at Arizona State University. His research interests include Computer Aided Geometric Design, Classical Geometry, Computer Animation, Kinematics and Robotics.

(Original GIF file wagner.gif is available.)