

Generating Tool Paths on Surfaces for a Numerically Controlled Calotte Cutting System

Elmar Wings (ProCom GmbH, Aachen, Germany)
Bert Jüttler (Johannes–Kepler–Universität, Linz, Austria)

Abstract. Non–planar driving mirrors have a complex geometry, which is defined by a base surface (the so–called calotte) and a planar contour. We describe and compare methods for generating NC tool–paths for a calotte cutting machine from these data. The methods are based on piecewise circular arcs and on polynomial spline curves. Distance bounds for the resulting tool paths, which are needed in order to check the accuracy, are also discussed. The paper concludes with suggestions for research on the use of Pythagorean hodograph curves in industrial NC tool path generation.

Keywords. NC tool paths, spline curves, distance bounds, Pythagorean hodograph curves.

Introduction

Free–form Geometry in CAD systems is almost always defined by spline curves and surfaces (NURBS). Tool paths for numerically controlled machining, by contrast, are mostly described by polygons or piecewise circles (G–code). Consequently, at the manufacturing stage, the sophisticated descriptions of free–form geometry for CAD have to be broken down in piecewise linear and circular representations. Typically, this conversion leads to a huge inflation of the data volume. In addition, it destroys the analytical continuity of the trajectories, as the piecewise linear/circular representations are merely tangent continuous or continuous.

We report on our experiences with spline curves for the numerically controlled cutting of non–planar driving mirrors. These mirrors, which are frequently used in cars, help the driver to detect other cars or obstacles in the so–called “dead angle”. First instances of these mirrors were manufactured by piecing together two segments. Currently, these mirrors are manufactured as a single segment, consisting of a spherical and a non–spherical zone, which are joined with high–order analytical continuity.

The manufacturing of the mirrors is done by glass

cutting of non–planar calottes, a process which requires highly specialized machines, such as – in our case – the SPHERIC 3D system of the HEGLA group (see Figure 1), with the CNC system ProCom CNC 300. The boundary curve of the mirror is traced with a cutting wheel. Afterwards, the glass surface is broken along this curve, giving the desired piece of the mirror.



Figure 1: The SPHERIC 3D calotte cutting system (picture courtesy of HEGLA Group).

Until recently, special machines for each mirror type have been used, and three–axes were therefore sufficient. Due to the shortening of cycle times in industry, NC machines with 6 axes are increasingly being used. These machines are capable of manufacturing mirrors with different geometries. Also, due to the fact that the cutting wheel is now always kept perpendicular to the base surface, one obtains a better quality of the cut.

In this special application of NC machining, high accuracy (0.1mm) and high order of analytical continuity of the tool path is very important. It leads to a reduction of forces which are present in the machinery. This extends the lifetime and reduces the number of maintenance intervals of the calotte cutting system. Moreover, due to the better quality of the trajectory of

the cutting wheel, it is much easier and more reliable to break the the glass surface along the desired mirror boundary curve. Consequently, less (manual) postprocessing (smoothing) of the rim of the mirror is needed.

Many related references on NC machining can be found in [5]. During the last years, much progress has been made to use high-level geometric primitives, such as spline curves, directly for NC machining. For instance, this can be achieved with the help of Pythagorean-hodograph curves, see [4]. Hermite interpolation with rational cubic splines has been used for robot control in an industrial environment [7].

The paper is organized as follows. First we describe the specification of the geometry, by combining a planar contour with a base surface. We then describe applications of the implicit representation of curves and surfaces, for checking the distance and for generating curves on surfaces. Then we report on several methods for generating the tool path. This includes the definition of an associated frame. We conclude this paper by indicating directions for potential future research.

Specifying the Geometry

We describe the definition of the shape of the mirror, which is obtained by projecting a planar contour onto a non-planar base surface. Distance bounds are also discussed.

Calottes

The mirrors are cut out of base surfaces, the so-called calottes. The calottes are mirrors in the shape of surfaces of revolution. Their profile curve $z = z(r)$ is shown in Figure 2 (not to scale). The calotte surface is

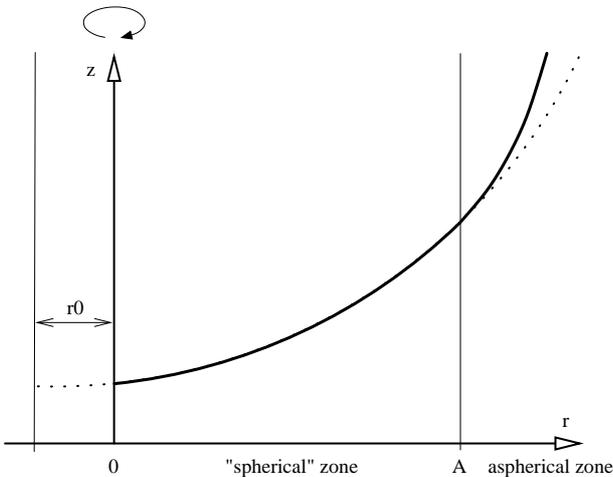


Figure 2: Profile curve of the calotte (not to scale).

generate by a rotation around the z -axis.

The main part of the calotte ($0 < r < A$), called the “spherical” zone, is generated by a circle segment with radius R and center $(-r_0, R)$,

$$z_s(r) = R - \sqrt{R^2 - (r + r_0)^2}. \quad (1)$$

The offset r_0 of this circular segment is introduced due to technical reasons; it facilitates the adjustment of the calotte surface in the machinery.

The outer part ($r > A$), called the *aspherical zone*, is generated by the profile curve

$$z_a(r) = R - \sqrt{R^2 - (r + r_0)^2} + K(r - A)^3. \quad (2)$$

The profile curve is the graph of the piecewise function

$$z(r) = \begin{cases} z_s(r) & \text{if } r \leq A \\ z_a(r) & \text{otherwise} \end{cases} \quad (3)$$

The spherical and aspherical zones are joined with continuous second derivatives at $r = A$. Consequently, the driver will not observe any “cracks” in the images in his mirror¹. After applying the rotation around the z axis, the inner part of the profile curve generates a segment of a spindle torus. If $r_0 = 0$, then the torus degenerates into a sphere. Typical values of the parameters are $r_0 = 7$, $R = 2000$, $A = 190$ (all in mm), and $K = 2.2 \cdot 10^{-5} \text{ mm}^{-2}$.

Planar contour

The boundary curve of the mirror is defined in the xy -plane and then projected onto the calotte surface. A typical boundary contour is shown in Figure 3. The

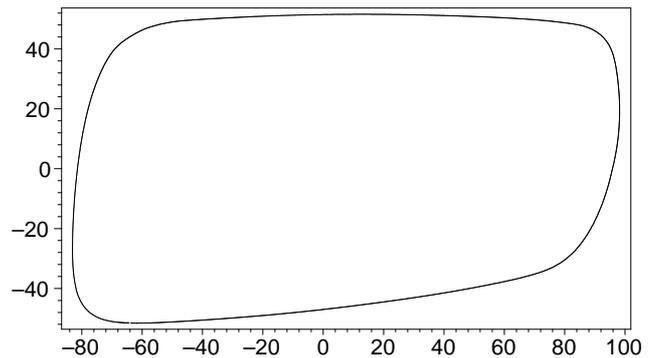


Figure 3: Mirror boundary curve

boundary curve of the mirror is given in digitized form, as a planar polygon (sequence of points).

The horizontal extension of the mirror is equal to approx. 200mm, i.e., about one tenth of the radius of the circular part of the profile curve. The calotte surface is used only in a neighbourhood of the axis of rotation,

¹Note that even jumps of the second derivatives would be visible, causing cracks in the reflected images of lines.

where it is close to being planar. After projecting the boundary curve onto the calotte, the maximum vertical deviation from the xy -plane is approximately 10mm.

Each calotte is used to produce several driving mirrors; they are placed with rotational symmetry around the z -axis. Thus, the mirror boundaries, which are defined in a local coordinate system, have to be moved to the suitable location in the xy -plane.

Implicit representation

In Computer Aided Geometric Design, curves and surfaces can be described either in parametric or in implicit form. In many applications, the availability of *both* representations results in simpler computations. For instance, the problem of computing the intersection(s) of two planar curves can then be reduced to a one-dimensional root finding problem.

We derive the implicit equations of the calotte surface and discuss two potential applications.

Implicit form of the calotte equation

The “spherical” segment of the calotte is part of a spindle torus with the equation

$$(r + r_0)^2 + (z - R)^2 = R^2 \quad (4)$$

where $r^2 = x^2 + y^2$. The implicit equation is obtained as the resultant of both polynomials with respect to r , leading to a polynomial of degree 4 in x, y, z ,

$$\begin{aligned} F^{(s)}(x, y, z) = & (x^2 + y^2)^2 + z^4 + 2(x^2 + y^2)z^2 \\ & - 4R(x^2 + y^2)z - 2r_0^2(x^2 + y^2) - 4Rz^3 \\ & + 2r_0^2z^2 + 4R^2z^2 - 4r_0^2Rz + r_0^4 = 0 \end{aligned} \quad (5)$$

The aspherical segment of the calotte is generated by the contour curve with the equation

$$G^{(a)}(r, z) = R^2 - (r + r_0)^2 - (R - z + K(r - A)^3)^2 = 0. \quad (6)$$

see (2). The implicit equation of the aspherical zone is again obtained as the resultant of $G^{(a)}$ and of $H(x, y, r) = x^2 + y^2 - r^2$ with respect to r . This leads to a polynomial $F^{(a)}(x, y, z)$ of degree 12 in x, y, z , consisting of 66 terms, if expressed in products of powers of $(x^2 + y^2)$ and z . (For the sake of brevity, the formula for $F^{(a)}$ is omitted; it can be generated with standard computer algebra tools.)

Distance estimates

The implicit form can be used to bound the distance of a point from the surface, as follows.

Lemma. *Let a surface in implicit form be given, $F(x, y, z) = 0$, with the domain $(x, y, z) \in \Omega \subset \mathbb{R}^3$. Consider a point $\mathbf{p}_0 = (x_0, y_0, z_0) \in \Omega$. We assume that the curve of steepest descent (that is, the integral curve of the vector field ∇F , see Figure 4) which starts at \mathbf{p}_0 reaches the surface at a point \mathbf{q} , and the curve segment from \mathbf{p}_0 to \mathbf{q} is contained in the domain Ω . Under these assumptions, the distance d from the surface satisfies*

$$d \leq F(x_0, y_0, z_0)/C, \quad (7)$$

where $C = \min_{(x,y,z) \in \Omega} |\nabla F|$.

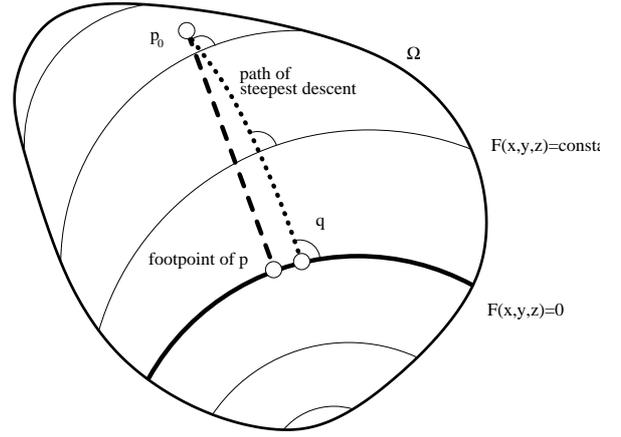


Figure 4: Distance of a point from a surface.

This result (without noting the technical assumption about the curve of steepest descent) is due to Sederberg [10]. The proof results by applying the mean value theorem to the restriction of F onto the curve of steepest descent (shown as dotted curve in the Figure). Clearly, the length of this curve is an upper bound of the orthogonal footpoint distance (dashed line).

If the given surface is algebraic (that is, $F(x, y, z)$ is simply a polynomial), then this Lemma can be used to bound the distance of a spline curve $\mathbf{p}(t) = [x(t), y(t), z(t)]$ from the surface, as follows. The composition $F \circ \mathbf{p}$ is again a piecewise polynomial function,

$$F(x(t), y(t), z(t)) = \sum_i d_i B_i(t) \quad (8)$$

with B-Splines, or even simply Bernstein polynomials, $B_i(t)$ and coefficients d_i . Hence, due to the convex hull property, the value of $F \circ \mathbf{p}$ is bounded by $\max_i |d_i|$. Consequently, the distance of the curve from the surface is bounded by

$$\max_i |d_i|/C. \quad (9)$$

The constant C , which is a lower bound of the length of the gradient, can be found with the help of the convex hull property of polynomials, applied to the gradients.

Note that in the case of calottes – where all surfaces are surfaces of revolution – it suffices to analyze the gradients in the plane $y = 0$, due to the rotational symmetry of the gradient field ∇F ! This is demonstrated by Figure 5, which shows the profile curve of the aspherical part and the gradient field $\nabla F^{(a)}(x, 0, z)$ (left).

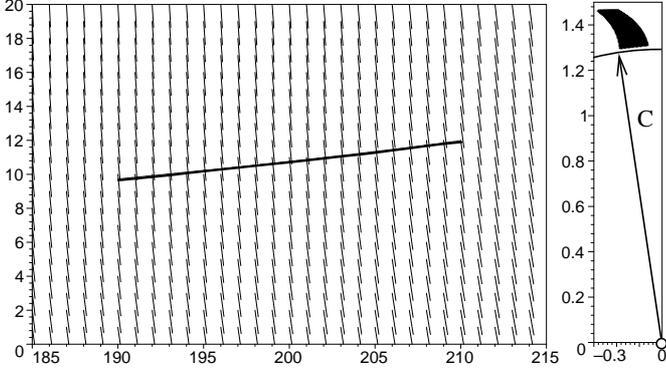


Figure 5: Gradient field of the aspherical part.

Here, the gradient field is a two-dimensional tensor-product Bézier patch of degree (11,3). A suitable constant C is found as the minimum distance of the convex hull of the gradients from the origin (right Figure).

Similar techniques can be applied to the spherical part, but one has to restrict oneself to a narrow strip around the calotte surface. This is due to the fact that the inner branch of the spindle torus surface is relatively close to the calotte surface.

Construction of curves on surfaces

The implicit form of a surface is also useful for generating curves $\mathbf{p}(t) = [x(t), y(t), z(t)]$ which lie (exact or approximately) on a given surface $F(x, y, z) = 0$. Exact solutions to this problem are available for special classes of surfaces, such as quadrics, tori, and cyclides. For more general implicit surfaces, approximate solutions can be found by minimizing – in addition to satisfying conditions such as interpolation or approximation of given data – an objective function of the form

$$\mathcal{G} = \int_a^b [F(x(t), y(t), z(t))]^2 dt. \quad (10)$$

Another objective function, which also takes the gradient information into account, is

$$\mathcal{H} = \int_a^b \frac{[F(x(t), y(t), z(t))]^2}{|\nabla F(x(t), y(t), z(t))|^2} dt. \quad (11)$$

We demonstrate the different behaviour of both objective functions by an example. We construct a

Bézier curve of degree 4 on an elliptic cylinder (diameters 1 and 3, with vertical axis) which interpolates 3 given points. The remaining degrees of freedom are used to minimize the objective functions \mathcal{G} and \mathcal{H} . Both objective functions are regularized by adding $\omega \int_0^1 |p'(t)|^2 dt$, with a small positive weight ω . The solution to the resulting optimization problem is found via Newton iterations, where the interpolating quadratic Bézier curve serves as the initial value.

The results are shown in Figure 6. The distance from the cylinder surface is visualized by the vertical vector fields along the curves (amplified by factor 10). The second objective leads to more uniform distribution of the distance error, and to a smaller global deviation from the cylinder surface.

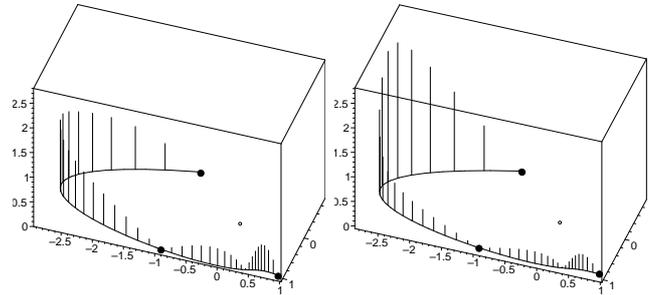


Figure 6: Curves through three points on an elliptic cylinder, computed with (left) and without (right) gradient information.

In many examples, where the norms of the gradients do not vary too much (such as in the case of the aspherical part of the calotte surface), the first objective function gives (almost) as good results as the second one. Also, this objective function is computationally easier to deal with, as its derivatives can be evaluated in closed form. In contrast to this, the evaluation of the derivatives of \mathcal{H} needs (far more expensive) numerical quadratures.

Tool path generation

We discuss three different methods for tool path generation and interpolation. After generating an analytical description of the tool path, we define an associated frame which is needed to specify the orientation of the tool. Finally, the tool path is traced according to the desired speed distribution.

Arc Splines

This method is based on an arc spline representation of the boundary. We interpolate the given points with a biarc spline, where the tangents are estimated from the data. See [12] for further information on biarcs.

The circular spline curve is projected onto the calotte surface, leading to a C^1 continuous space curve.

Planar Splines

The second method uses planar spline curves. First, we parameterize the points using one of the standard methods for parameter estimation, such as centripetal parameterization, cf. [11, Chapter 9.2]. Next, we use the algorithm A9.1 of [11] to generate a polynomial (at least) C^2 B-spline curve. Finally, this curve is projected onto the calotte surface. This results in a curve of the form

$$\mathbf{p}(t) = [x(t), y(t), z(\sqrt{x(t)^2 + y(t)^2})] \quad (12)$$

where $x(t)$ and $y(t)$ are spline functions.

Spatial Splines

Finally, the third method used spatial spline curves for describing the boundary curve. The algorithm is almost the same as for method 2, but the steps of projection and interpolation are now swapped. The given points are projected onto the calotte surface. Again, we use one of the standard methods for parameterization to find an associated sequence of parameter values. The spatial spline curve is again generated with the help of algorithm A9.1 from [11].

Note that, however, the spatial curve is not exactly contained in the calotte surface. In order to guarantee the desired accuracy, we need to check the distance from the base surface. Exact error bounds are available from (9). If needed, that accuracy of the curve can be improved with the help of an optimization procedure based on the objective functions \mathcal{G} or \mathcal{H} . The additional degrees of freedom can be generated by inserting additional knots, or by degree raising.

Associated Frame

At each point $\mathbf{p}(t) = (x, y, z)$ of the mirror boundary we define an associated frame, which specifies the orientation of the cutting wheel of the calotte cutting system. The frame consists of the tangent vector of the tool path,

$$\vec{\mathbf{t}}(t) = \dot{\mathbf{p}}(t)/|\dot{\mathbf{p}}(t)|, \quad (13)$$

the normal vector of the calotte surface, which can be obtained from the implicit equation,

$$\vec{\mathbf{n}}(t) = \nabla F(x, y, z)/|\nabla F(x, y, z)| \quad (14)$$

and of their cross-product $\vec{\mathbf{b}}(t) = \vec{\mathbf{t}}(t) \times \vec{\mathbf{n}}(t)$. These three vectors are then used to compute the three angles of the rotational drives of the calotte cutting system. Due to the choice of the second vector, the cutting tool is always kept orthogonal to the calotte surface.

Tracing the tool path

During the cutting process, the control unit of the calotte cutting system generates – at intervals of 1 millisecond – positions (point and frame data) along the tool path. These positions are used to control the six axes of the calotte cutting machine. The distribution of the positions is governed by the physical limitations of the machine. We use a generalization of the bang–bang principle, as described by Farouki et al. [6]. In our implementation, we take the maximum velocity, acceleration and jerk along the tool path and at the six axes of the machine into account, and use \sin^2 acceleration characteristics. Note that the arc length of the tool path, which is needed for controlling the tool speed, cannot be computed exactly for spline curves; in this case, our implementation uses numerical quadratures to approximate it.

Experimental results

We compare the tool path descriptions by arc splines and by planar polynomial spline curves, using a part of the mirror boundary (shown in grey in Figure 7). We

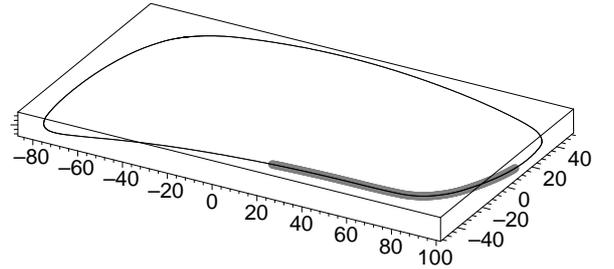


Figure 7: The segment of the mirror boundary.

tested the path descriptions by arc splines and polynomial spline curves, and recorded the machine data at intervals of 4 milliseconds². The experimental results are visualized in Figure 8, showing the distribution of the accelerations along the path (grey curves). These accelerations were computed as second order difference quotients from the actual position data. In order to reduce the inherent noise, we have also plotted the average acceleration of 11 neighbouring positions (black curve).

The piecewise circular description of the tool trajectory leads to a discontinuous distribution of the acceleration, which implies relatively large forces at the machinery. The polynomial spline curve produces a relatively smooth distribution with only small jumps.

On the other hand, in the regions where the boundary curve has a relatively low curvature (which corre-

²Due to technical reasons, only every fourth timestep was recorded.

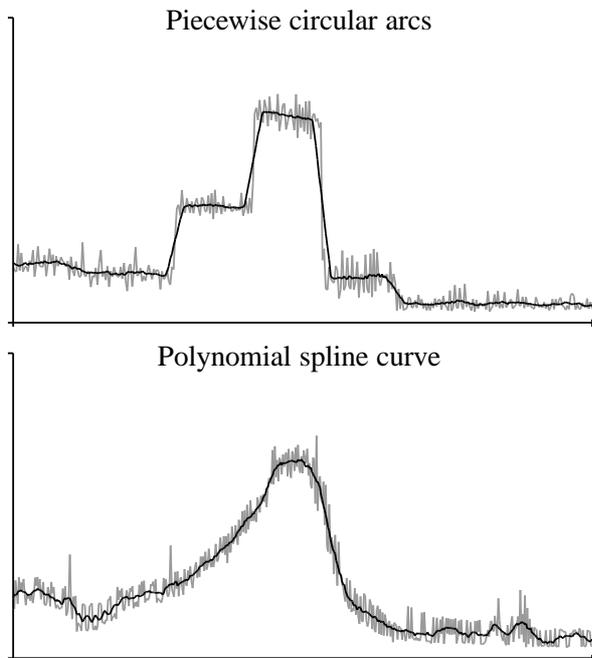


Figure 8: Accelerations (grey curve) and average of 11 adjacent accelerations (black).

sponds to the small accelerations in the right part of the figure), the acceleration produced by the spline curve oscillate much more than the accelerations produced by the arc spline curve. This is due to the approximate arc length computation via numerical quadratures, which has to be used in the spline case.

Note that – for both methods – the tracing of the tool path is based on the arc length of the *planar* curves. This systematically underestimates the exact arc length of the spatial trajectory, which lies on the calotte surface. This can be avoided by using the third interpolation scheme, generating a 3D spline curve. This method, however, needs an additional distance check, which currently appears to be too expensive. In this specific application, where the tool path is rather close to being a planar curve, it is well justified to neglect the deviation from the plane for the arc length computation. For more curved objects, however, it will become necessary to use spatial trajectories of the tool for controlling the speed.

Clearly, the reduction of forces which are present in the machinery extends the lifetime and reduces the number of maintenance intervals of the calotte cutting system. Also, it increases the accuracy of the trajectory of the cutting wheel, and it is therefore simpler and more reliable to break the glass surface along the desired mirror boundary. Consequently, less postprocessing (smoothing) of the rim of the mirror is needed, which reduces the amount of manual work.

Concluding remarks

As demonstrated in our case study, polynomial spline curves can directly be used for NC machining. Compared to the traditional piecewise linear/circular representations, the higher analytical continuity entails severe advantages, such as extended lifetime of the machinery, longer maintenance intervals, and higher quality of the products.

The arc length of polynomial spline curves is not directly available, which causes additional computational problems. In our implementation, this issue has been addressed by numerical quadratures. The results, however, are not fully satisfying yet. We think that this problem needs to be addressed by further research.

As one possibility, one should further explore the potential of Pythagorean hodograph (PH) curves, which have been developed by Farouki in a series of papers (see [2] and the references cited therein). For this remarkable class of curves, the arc length is a simple polynomial function of the parameter. The construction of PH spline curves with C^2 or G^2 continuity, however, causes again some computational problems. In the planar case, it either involves the solution of global systems of polynomial equations [1], or it requires root finding procedures for quartic polynomials, and the existence of solutions is not guaranteed a priori [9]. The situation is even more difficult for spatial PH curves, where no related techniques seem to be currently available.

Instead of resorting to this somewhat exotic – although remarkable – class of curves, it might be a useful alternative to augment the existing tools for spline curves with techniques for improving the distribution of the parametric speed; this could help to reduce the error which is introduced by the numerical quadratures. So far, the existing techniques use piecewise rational curves [3].

Acknowledgments

The authors wish to thank the anonymous referee for his helpful comments. The second author was supported by the Austrian Science foundation (FWF) through project 15 of the SFB F013 “Numerical and Symbolic Scientific Computing” at the University of Linz, and by the European commission through the 5th framework programme, project GAIA II “Intersection algorithms for geometry based IT-applications using approximate algebraic methods”, contract no. IST-2002-35512.

References

- [1] G. Albrecht and R.T. Farouki, Construction of C^2 Pythagorean-hodograph interpolating splines by the homotopy method, *Advances in Computational Mathematics* 5 (1996), 417-442.
- [2] R.T. Farouki, Pythagorean-hodograph curves, in: *Handbook of Computer Aided Geometric Design* (G. Farin, J. Hoschek, and M-S. Kim, eds.), Elsevier, to appear (2002).
- [3] P. Costantini, R.T. Farouki, C. Manni, and A. Sestini, Computation of optimal composite parameterizations, *Computer Aided Geometric Design* 18 (2001), 875-897.
- [4] R.T. Farouki, Y.-F. Tsai, and B. Feldman, Performance analysis of CNC interpolators for time-dependent feedrates along PH curves, *Computer Aided Geometric Design* 18 (2001), 245-265.
- [5] R.T. Farouki and Y.-F. Tsai, Exact Taylor series coefficients for variable-feedrate CNC curve interpolators, *Computer Aided Design* 33 (2001), 155-165.
- [6] R.T. Farouki, T.S. Smith and S.D. Timar, Specification of time-optimal feedrates for curved tool paths in 3-axis machining, preprint (2002) (available from <http://mae.ucdavis.edu/~farouki>)
- [7] T. Horsch and B. Jüttler, Cartesian Spline Interpolation for Industrial Robots, *Computer-Aided Design* 30 (1998), 217-224.
- [8] J. Hoschek and D. Lasser, *Fundamentals of Computer Aided Geometric Design*, AK Peters, Wellesley (Mass.), 1993.
- [9] B. Jüttler, Hermite interpolation by Pythagorean hodograph curves of degree seven, *Mathematics of Computation* 70 (2001), 1089-1111.
- [10] T.W. Sederberg, Planar piecewise algebraic curves, *Computer Aided Geom. Des.* 1 (1984), 241-255.
- [11] L. Piegl and W. Tiller, *The NURBS Book*, Springer, Berlin 1995.
- [12] A.W. Nutbourne and R.R. Martin, R.R., *Differential Geometry Applied to Curve and Surface Design*. Vol. 1: Foundations, Ellis Horwood, Chichester, 1988.



Elmar Wings studied mathematics at the Aachen University of Technology (RWTH Aachen, Germany). In 1995, he gained a PhD with a thesis in the field of Group Theory. He then joined the Procom GmbH at Aachen, a software company which develops – among other products – NC software for highly specialized machinery, such as glass cutting systems. Currently he is the head of the CNC development unit at Procom. His research interests include computer aided geometric design, computational geometry, and kinematics.



Bert Jüttler studied mathematics at the Universities of Technology at Dresden and Darmstadt (both in Germany). After gaining a PhD in 1994, he was a postdoctoral research assistant at Darmstadt and at the University of Dundee (Scotland), and later became lecturer at Darmstadt (since 1998). Since October 2000 he is professor of mathematics at the Johannes Kepler University of Linz, Austria. His research interests include Computer Aided Geometric Design, Kinematics and Robotics, Differential Geometry and Computer Graphics.

Authors' addresses

Elmar Wings
ProCom Systemhaus und Ingenieurunternehmen GmbH
Luisenstr. 41, D-52070 Aachen, Germany
Phone: +49 241 51804-155
Fax: +49 241 51804-30
Email: ws@procom.de
WWW: <http://www.procom.de>

Bert Jüttler
Johannes-Kepler-Universität Linz
Inst. für Analysis, Abt. für Angewandte Geometrie
Altenberger Str. 69, A-4040 Linz, Austria
Phone: +43 732 2468 9178
Fax: +43 732 2468 9142
Email: bert.juettler@jku.at
WWW: <http://www.ag.jku.at>