REVISTA INVESTIGACIÓN OPERACIONAL Vol 27, No. 3, 259-270, 2006

SOME APPLICATIONS OF ALGORITHMS FOR GENERATING POINTS ON REGULAR PARAMETRIC CURVES.

Victoria Hernández Mederos *, Jorge Estrada Sarlabous*
* Instituto de Cibernética, Matemática y Física, ICIMAF, La Habana, Cuba.
e-mails: vicky@icmf.inf.cu, jestrada@icmf.inf.cu

ABSTRACT
Controlling the distribution of points on parametric curves is a very important problem in CAGD and related areas. In this paper we present three different applications, where the solution of the previous problem is crucial. Two algorithms for generating points on regular parametric curves are used in the applications. In the first application, we show how to generate points with control of their distribution on a 3D parametric curve. In the second one, the generation of points on the boundary of a planar grid is considered. Finally, we address the problem of reconstructing surfaces from cross sections and show that the distribution of points on the boundary curve of each cross section, has an strong influence in the quality of the reconstructed surface.

Key words: Parametric curves, Arc length parametrization, Sampling points, Applications.

MSC: 65D17 , 65D07 , 65D05

RESUMEN
Un problema muy importante en el área de diseño geométrico computarizado es controlar la distribución de los puntos sobre una curva paramétrica. En este artículo se presentan tres aplicaciones, donde la solución de este problema es crucial. En las aplicaciones se utilizan dos algoritmos para generar puntos sobre una curva paramétrica regular. En la primera aplicación se m uestra cómo generar puntos sobre una curva 3D controlando su distribución. En la segunda aplicación se generan puntos sobre la frontera de una malla plana. Finalmente, se estudia el problema de la reconstrucción de una superficie a partir de un conjunto de secciones transversales y se muestra que la distribución de los puntos sobre las curvas de la frontera de cada sección tiene una fuerte influencia en la calidad de la superficie reconstruida.

1 INTRODUCTION.

Parametric curves and surfaces are the standard in CAGD problems. One of the main advantages of using these curves and surfaces is the easy way of computing a set of points on it, in order to make a graph. The simplest procedure is to select a uniform partition of the parametric domain and to compute the corresponding sequence of points on the curve or surface. In Figure 1 we show a sequence of points on a Bezier curve of degree 4, which corresponds to equally spaced

parametric values. As we observe, the distribution of points on the curve is very irregular.

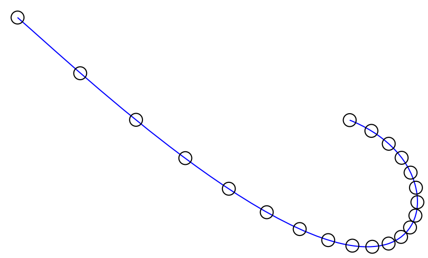

Figure 1: A sequence of points on a 2D Bezier curve corresponding to equally spaced parametric values

In some applications the curve represents the path described by an object in movement and one is interested in obtaining a uniform displacement. In other problems, like the reconstruction of surfaces from cross sections, it is necessary to match points from one parametric curve to another one, a difficult task if the speeds of the motion along the curves are very uncorrelated. There are many other problems where the generation of a sequence of points on a parametric curve with control of their distribution is crucial.

The parametrization of a curve defines the speed at which the points on the curve are traced out. In this sense, arc length parametrization is the "ideal". If a curve is arc length parametrized then any given distribution of values in the parametric space is reproduced on the curve. In particular, the arc length between two consecutive points corresponding to equally spaced parametric values is constant. The main advantages of the arc length parametrization of a curve have been mentioned by several authors, see [Bloomenthal, 1999], [Casciola-Morigi, 1996] and [Gil-Keren, 1997].

It is well known [Farouki –Sakkalis, 1991] that piecewise linear functions are the only real curves that can be parametrized by rational functions of their arc length. In consequence, several methods for computing approximations of the arc length function or the inverse of the arc length function for parametric curves can be found in the literature, see for instance [Fritsch-Nielson, 1992], [Bloomenthal, 1988], [Coquillart 1987]. As we already explained, problems of arc length reparametrization of parametric curves and generation of a sequence of points with control of their distribution are strongly related and there is a significant number of papers dealing with this subject, [Wever, 1991], [Wang-Yang, 1993], [Horsch-Jüttler, 1998], [Elber, 1995], [Blanck-Schlick, 1996], [Casciola-Morigi, 1996], [Farouki, 1997], [Costantini et. al., 2001]. In [Bloomenthal, 1999], a wide summary of previous work can be found.

In [Hernández-Estrada, 2003] two algorithms for generating a sequence of points on a regular parametric curve with control of their distribution are proposed. Given a parametric curve $c(u)$, u [0,1] and a prescribed integer n, algorithm **UniArc-Length** [Hernández-Estrada, 2003] generates a sequence of n+1 points on the curve $c(u)$ whose arc length distribution is approximately uniform. In each step of this algorithm, a reparametrization function which approximates the inverse of the arc length function is computed. The reparametrization function is a C^1 rational linear spline which interpolates the inverse of the arc length function and its derivative. For a good initial approximation, algorithm **UniArcLength** converges quadratically to a vector u such that the corresponding sequence of points on the curve $c(u)$ has uniform arc length distribution.

The second algorithm proposed in [Hernández-Estrada, 2003] is called **CurvaturevsArcL**. This algorithm generates a sequence of points on a regular parametric curve which represents the geometry of the curve, in the sense that most of the points are concentrated in the regions of higher curvature but, at the same time, there are some points in the regions of relative large arc length and low curvature. The algorithm depends of a free parameter q, a real number in [0,1], used to decide which aspect will be emphasized. A small value of q means that we are interested in obtaining a distribution of points close to the uniform arc length distribution, whereas a value of q close to 1 means that we want to generate most of the points in the regions of higher curvature.

In this paper we present several important applications of the above mentioned algorithms discussed in details in [Hernández-Estrada, 2003]. Through different problems, we show why it is crucial to have control of the distribution of points on a regular parametric curve and how the previous algorithms are an efficient tool in the solution of these problems.

2. GENERATING POINTS ON 3D PARAMETRIC CURVES

Algorithms **UniArcLength** and **CurvaturevsArcL**, originally designed for planar curves, can be easily extended to 3D regular curves represented as: $c(u)=(x(u),y(u),z(u))$, u [0,1]. It is enough to take into account that in this case the arc length function $s=l(u)$ is given by,

$$s = l(u) = \int_0^u \sqrt{x'^2(t) + y'^2(t) + z'^2(t)}\,dt = \int_0^u \left\| c'(t) \right\|_2 dt$$

Figure 2 shows a sequence of 21 points on a 3D rational Bezier curve of degree 8. In a) we show the sequence of points corresponding to equally spaced parameter values of the Bezier representation of the curve. As we observe, the arc length between two consecutive points is very irregular. Points in b) have approximately uniform arc length distribution and were computed using Algorithm **UniArcLength**.

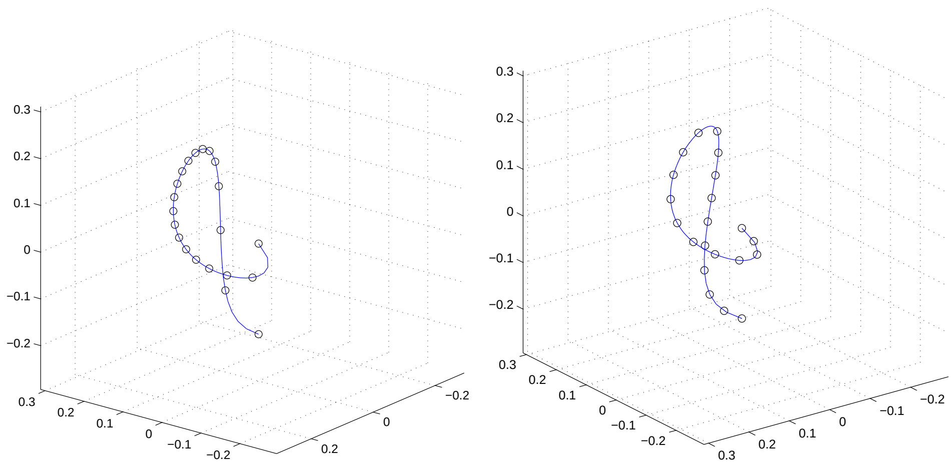

Figure 2: A 3D rational Bezier curve, a) points corresponding to equal space parameter values b) uniform arc length distribution

The generation of points on a 3D parametric curve with control of their positions it is also interesting working with parametric surfaces. Given a parametric surface $F(u,v)$, (u,v) $[a,b]$ $[a,b]$ an important problem is how to construct a grid points on it (for instance triangular or rectangular) with certain "optimal" properties. Grids on a surface are used to render it or to discretize a differential operator, in order to solve a partial differential equation, whose domain is the parametric surface [Garanzha, 2000]. In any case, it is relevant to have a grid of points on the surface providing a good description of its geometry. Boundary curves of the surface are 3D parametric curves and the shape of boundary cells strongly depends on the distribution of points on these curves.

Figure 3 shows a planar triangulation on the parameter space and the corresponding 3D triangulation on a tensor product Bezier surface of degree 4x4, obtained calculating the image of each vertex of the planar triangulation by the parametrization of the surface. Computing the parameter values (vertices of the planar triangulation) such that the corresponding 3D triangulation has "optimal" properties is a difficult task, since usually the parametrization $F(u,v)$ of the surface is not an isometry and therefore it introduces deformations in the lengths and angles of the triangles. If one is interested in a 3D triangulation where all triangles have approximately the same area, as in Figure 3, algorithms for generating points on 3D boundary curves with control of their distribution could be very useful, since the area of boundary triangles depends on the position of its vertices on 3D boundary curves.

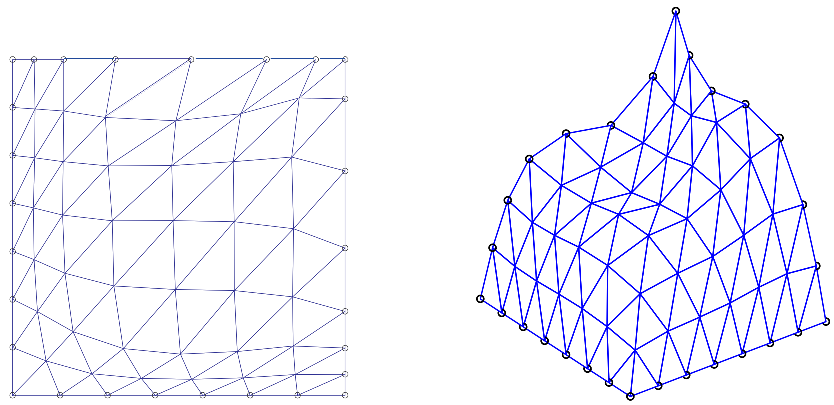

Figure 3: A planar triangulation in the parameter space and the corresponding 3D triangulation on a tensor product Bezier surface.

3. THE GRID GENERATION PROBLEM.

In the numerical solution of partial differential equations (PDE) an important problem is the generation of a grid in the domain in order to approximate the differential operator. When the domain is a bounded subset of R^2 its boundary is usually described by a parametric closed spline curve.

There are in the literature a great amount of papers dealing with the grid generation problem, a couple of references are [Ho-Le, 1988], [George, 1991]. The finite element method is widely used in the numerical solution of PDE and has extended its range of applicability and its computational efficiency. However, it relies on the discretization of the domain region, provided by mesh generation methods. Most of these methods use triangular or quadrilateral elements and the shape of the boundary cells strongly depends on the position of the grid points lying on the boundary curves. Therefore, having a control of the distribution of these points is crucial for getting a grid with optimal properties.

Our main goal in this application is to generate a sequence of points on the boundary curve of a planar region , where a PDE has to be solved. Points generated by the previously mentioned algorithms are the boundary points of a planar grid constructed in .

The input data in this application is a sequence of points describing the boundary of the region . First, we approximate it with a C^1 conic spline curve $c(u)$, composed by lines and conic sections. Our general strategy is to compress the data and approximate them by a polygonal [Ray-Ray, 1994]. Then, we smooth the polygonal cutting corners by means of conic sections, which are written as

quadratic rational Bezier curves. Briefly, the method used consists of the following steps:

1. Compression of data, in order to eliminate the noise.
2. Construction of the control polygon, i.e. selection of the vertices b_0^i, b_1^i, b_2^i of the control polygon of the i-th section of the conic spline curve, *i*=0,...,*m*-1.
3. Computation of the weight w_1^i associated to the i-th conic section, *i*=0,...,*m*-1.
4. Determination of the knots $v_0, v_1, ..., v_m$ of the spline curve. In order to obtain a C^1 curve we use the following formulas[Farin, 1992],

$$v_0 = 0$$

$$v_1 = \left\| b_2^0 - b_0^0 \right\|_2$$

$$v_{i+1} = v_i + (v_i - v_{i-1}) \frac{w_1^i}{w_1^{i-1}} \frac{\left\| \Delta b_0^i \right\|_2}{\left\| \Delta b_1^{i-1} \right\|_2} \tag{1}$$

where $\Delta b_0^i = b_1^i - b_0^i$ and $\Delta b_1^{i-1} = b_2^{i-1} - b_1^{i-1}$.

From the C^1 continuity condition (1) it is easy to realize that if the ratio between the weights w_1^i and w_1^{i-1} is close to 0 then v_{i+1} is very close to v_i . It means that the i-th section of the conic spline is traced out at a very high speed, independently of the arc length of this section of the curve. That situation would be inconvenient, since the convergency of the algorithm **UniArcLength** is guaranteed under the hypothesis that $\left\| c^{'}(u) \right\|_2 \le C < \infty$. On the contrary, if the ratio w_1^i / w_1^{i-1} is very high, then $v_{i+1} >> v_i$ and consequently, the i-th section of the curve is traced out very slow; it means that $\left\| c^{'}(u) \right\|_2$ is close to 0. Therefore, in order to be sure that the algorithm **UniArcLength** converges, we have to select the weights of each conic section in such away that,

$$0 << \frac{w_1^i}{w_1^{i-1}} << \infty$$

In our experiments we smooth out the polygonal computing the weighs by the following formula,

$$w_1^i = 2 + cos\, \theta_i$$

where θ_i is the angle between the segments $b_0^i b_1^i$ and $b_1^i b_2^i$. Observe that (see Figure 4), $w_1^i \to 3$ when $\theta_i \to 0^\circ$, $w_1^i \to 2$ when $\theta_i \to 90^\circ$ and $w_1^i \to 1$ when $\theta_i \to 180^\circ$. Therefore, with this selection the ratio between the weights is bounded by, $\frac{1}{3} < \frac{w_1^i}{w_1^{i-1}} < 3$ which guarantees the convergency of the algorithm **UniArcLength**.

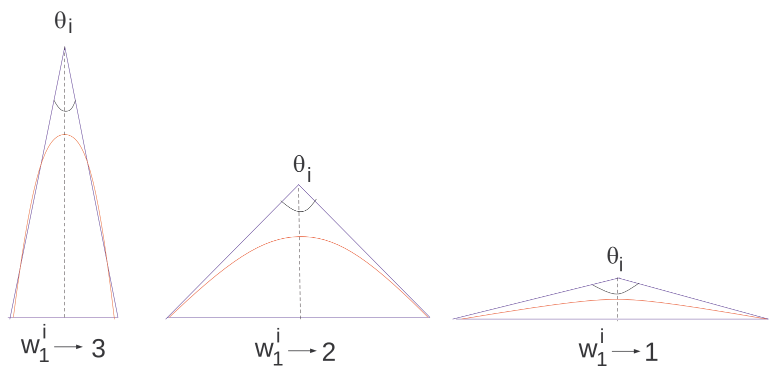

Figure 4: Computing the weight w_1^i of the i-th conic section in terms of the angle θ_i

The first step in the construction of a quadrilateral grid on a planar region is the subdivision of the boundary in 4 sections: two “horizontal” curves and two “vertical” curves. Grid is composed by 2 kind of lines: “horizontal lines” joining points of opposite “horizontal” boundary curves and “vertical lines” which join points of opposite “vertical” boundary curves. Algorithm **UniArcLength** was incorporated to UNAMALLA (see *http://www.mathmoo.unam.mx/unamalla*), a system for grid generation over irregular planar regions. **UniArcLength** is used in UNAMALLA for generating a sequence of points with uniform arc length distribution on each of the 4 boundary curves. Moreover, the reparametrization function obtained as byproduct of algorithm **UniArcLength** have been also used in UNAMALLA for generating points on the 4 boundary curves with other special arc length distributions. The grid is obtained by an optimization process using functionals that measure geometric properties [Barrera et. al., 2003].

Figure 5 show the results of algorithm **UniArcLength** applied to the conic spline approximating the boundary of two planar regions, which represent Havana harbor and England map, respectively. System UNAMALLA was used to generate a grid of 30 30 points in Havana harbor and 50 50 points in England map. In means

that on the “horizontal” boundary curves of Havana harbor we have generated 30 points with uniform arc length distribution, while in the “vertical” boundary curves 30 points were also computed. Similarly, on the “horizontal” and “vertical” boundary curves of England map we generate 50 points with uniform arc length distribution.

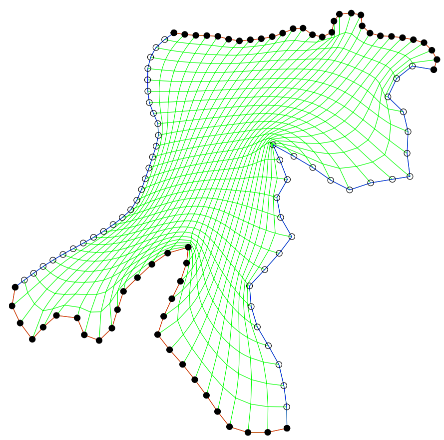

Figure 5a)

Figure 5b)

Figure 5: Points with uniform distribution on the boundary curves of two bounded regions. a) A grid of 30 30 points in Havana harbor, b) a grid of 50 50 points in England map.

4 SURFACE RECONSTRUCTION FROM CROSS SECTIONS

The reconstruction of 3D objects from 2D data on slices has received great attention in the last years specially in connection with medical images. The problem has been historically solved using parametric spline curves and surfaces. There is a lot of papers dealing with the subject, see [Shumaker, 1990] and the long list of references therein.

Given a set of heights and contour data describing the corresponding cross sections of a 3D object, the problem consists in finding a "convenient mathematical" representation of the object, i.e. a representation that can be stored in a digital computer and can be used to display and manipulate an image of the object in a graphical device [Shumaker, 1990].

Algorithms to solve the reconstruction problem can be classified in two categories: volume rendering methods and surface reconstruction methods. Certain classes of methods try to reconstruct the surface approximating each contour data by a curve, then selecting a set of representative points on each cross section and finally connecting two neighboring contours by means of triangles or quadrilaterals. The resulting surface is composed by triangular or quadrilateral facets. A smoother approximation of the surface may be obtained constructing a spline surface, defined by triangular or quadrilateral patches, interpolating the vertices of the polygons and having continuous varying tangent plane.

Usually, each set of contour data is approximated by a parametric spline curve and then a set of representative points on the curve has to be selected. When we say a set of representative points we mean a set of points that describe well the geometry of the curve. In this sense, it is natural, for instance, to require that there are more points on the segments of the curve with higher curvature, but there are also some points on the segment of the curve with relative large arc length but low curvature. Otherwise, some details of the global surface could be lost in the reconstruction process. In other words, the distribution of the selected points on two neighboring cross sections, as well as the method chosen for constructing the triangles or quadrilaterals joining two consecutive sections, have a strong influence on the shape of the final facet surface. Algorithm **CurvaturevsArcL** (with q close to 1) described in details in [Hernández-Estrada, 2003] is very useful to achieve that goal.

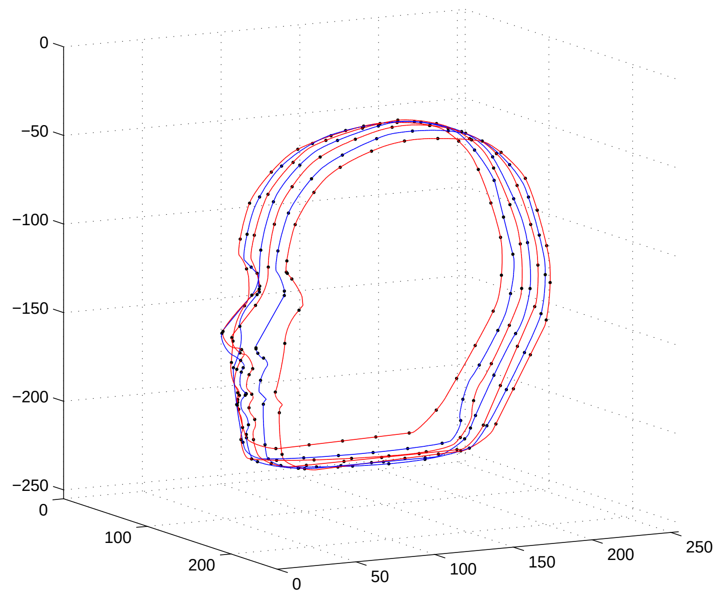

Figure 6: Cross sections of a human head approximated by conic splines

Figure 6 shows a small set of cross sections of a human head which have been approximated by conic splines. As we appreciate in the figure, points generated by algorithm **CurvaturevsArcL** are more concentrated in the regions of higher curvature (near the nose) avoiding losing important details in the reconstruction process.

ACKNOWLEDGEMENTS:

The working group around UNAMALLA system at UNAM University, México, successfully included our algorithm **UniArcLength** in their system for automatic grid generation. We would like to express our gratitude to them, specially to Dr. Pablo Barrera. We also appreciate their generosity providing us with graphics for Figure 5.

REFERENCES

BARRERA SÁNCHEZ, P., G.F. GONZÁLEZ FLORES and F.J. DOMÍNGUEZ MOTA (2003): Robust discrete grid generation on plane irregular regions, **Computational Mathematics and Mathematical Physics** 43, 845–853.

BLANC C. and C. SCHLICK (1996): Accurate parametrization of conics by NURBS**, IEEE Computer Graphics and Applications** 16**,** 64–71.

BLOOMENTHAL, M. (1988): **Approximation of sweep surfaces by tensor product B-splines**, Tech. Rep. UUCS-88-008, Dept. of Computer Science, Univ. of Utah.

BLOOMENTHAL, M. (1999): **Error bounded approximate reparametrization of NURBS curves**, M.Sc. Thesis, University of Utah, 1999.

CASCIOLA G. and S. MORIGI (1996): **Reparametrization of NURBS curves, Int. Journal of Shape Modeling** 2**,** 103–116.

COQUILLART. S. (1987) : A control-point-based sweeping technique, **IEEE Computer Graphics and Applications 11**, 36–45.

COSTANTINI, P., R. FAROUKI, C. MANNI and A. SESTINI [2001) : Computation of optimal composite re-parametrizations, **Computer Aided Geometric Design** 18, 875–897.

ELBER, G. [1995) : Symbolic and numeric Computation in curve interrogation, **Computer Graphics Forum** 14, 25–34.

FARIN, G. [1992) : **Curves and Surfaces for Computer Aided Geometric Design,** A Practical Guide, 3rd edition, Academy Press. N. York.

FAROUKI, R. T. and T. SAKKALIS [1991): Real rational curves are not unit speed, **Computer Aided Geometric Design 8**, 151–157.

FAROUKI, R. T. [1997), Optimal parametrizations, **Computer Aided Geometric Design 8,** 153–168.

FRITSCH F.N. and G.M. NIELSON (1992), **On the problem of determining the distance between parametric curves**, in: Curves and Surface Design, H. Hagen, Ed. SIAM, 123–141.

GARANZHA, V.A. (2000) : Barrier Method for Quasi-Isometric Grid generation, **Computational Mathematics and Mathematical Physics 11**, 1617–1637.

GEORGE, P.L. (1991), **Automatic Mesh Generation. Application to Finite Elements Methods,** J. Wiley & Sons, New York.

GIL J. and D. KEREN (1997) : New approach to the arc length parametrization problem, in: **13th Spring Conference on Computer**

Graphics, Ed. W. Strasser, 27–42.

HERNÁNDEZ MEDEROS V. and J. ESTRADA SARLABOUS(2003): Sampling points on regular parametric curves with control of their distribution, **Computer Aided Geometric Design, 20**, 363–382.

HO-LE, K. (1988) : Finite element mesh generation methods: a review and classification, **Computer Aided Design** 20, 27–38.

HORSCH, T. and B. JÜTTLER (1998) : Cartesian spline interpolation for industrial robots, **Computer Aided Geometric Design 30**, 217–224.

RAY, B. KR. and K.S. RAY (1994) : A non-parametric sequential method for polygonal approximation of digital curves, **Pattern Recognition Letters, 15**, 161–167.

SHUMAKER, L.L. (1990) : **Reconstructing 3D objects from cross sections,** in: Computation of Curves and Surfaces, Kluwer Academic Publishers, Dordrecht, . 275–309.

WANG, F. C and D.C. YANG (1993) : Nearly arc-length parameterized quintic-spline interpolation for precision machining, **Computer Aided Design 25,** 281–288.

\WEVER, U. (1991) : Optimal parametrization for cubic splines’**, Computer Aided Geometric Design 23**, 641–644.