Approximate Implicitization of Space Curves and of Surfaces of Revolution

Mohamed Shalaby and Bert Jüttler

Institute of Applied Geometry,
Johannes Kepler University, Linz, Austria
{mohamed.shalaby,bert.juettler}@jku.at

Abstract. We present techniques for creating an approximate implicit representation of space curves and of surfaces of revolution. In both cases, the proposed techniques reduce the problem to that of implicitization of planar curves. For space curves, which are described as the intersection of two implicitly defined surfaces, we show how to generate an approximately orthogonalized implicit representation. In the case of surfaces of revolution, we address the problem of avoiding unwanted branches and singular points in the region of interest.

1 Introduction

Traditionally, most CAD (Computer Aided Design) systems rely on piecewise rational parametric representations, such as NURBS (Non–Uniform Rational B–Spline) curves and surfaces. The parametric representation offers a number of advantages, such as simple sampling techniques, which can be used for quickly generating an approximating triangulation for visualization. On the other hand, the use of implicitly defined curves/surfaces also offers a number of advantages, e.g., for solving intersection problems, or for visualization via ray–tracing.

In order to exploit the potential benefits of using the implicit representation of curves and surfaces, methods for conversion from parametric to implicit form (implicitization) are needed. As an alternative to exact methods, such as resultants, Groebner bases, moving curves and surfaces, etc. [2, 4, 5, 8, 12], a number of approximate techniques have emerged [3, 7, 9, 10]. As demonstrated in the frame of the European GAIA II project [6, 13, 15], these techniques are well suited to deal with general free–form curve and surface data arising in an industrial environment.

On the other hand, CAD objects typically involve many special curves and surfaces, such as natural quadrics, sweep surfaces, surfaces of revolution, etc. While implicit representations of simple surfaces are readily available, this paper studies approximate approximation of two special objects, namely space curves and surfaces of revolution. Space curves arise frequently in geometric modeling. An implicit representation of a space curve is given by the intersection of two implicitly defined surfaces. A surface of revolution is created by rotating 2D profile curve about an axis in space. Rotation is one of the standard geometric operations defined in any CAD/CAM interface.

This paper presents techniques for approximate implicitization of space curves and of surfaces of revolution, which are based on the (approximate) implicitization of planar curves. The proposed techniques are fully general in the sense that they can be combined with any (exact or approximate) implicitization method for planar curves. For creating the examples shown in this paper, we used a technique for simultaneous approximation of points and associated normal vectors [9, 10, 14].

This paper is organized as follows. First we summarize the approximate implicitization method for planar curves. Section 3 presents techniques for approximate implicitization of space curves, first as the intersection of two general cylinders, and later as the intersection of two general surfaces which intersect approximately orthogonal. Representing the space curve by two 'orthogonal' surfaces provides a more robust definition for the curve. Finally, in Section 4, two methods for approximate implicitization of surfaces of revolution are presented. It is shown that – in many cases – only approximate implicitization is capable of producing an implicit representation that is free of unwanted branches and singularities.

2 Simultaneous approximation of points and normals

For the sake completeness, we give a short description of the approximate implicitization method presented in [9]. This method is characterized by the simultaneous approximation of sampled point data $\mathbf{p}_i = (x_i, y_i)$, $i \in \mathcal{I} = \{1, \ldots, N\}$, and estimated unit normals $\mathbf{n}_i$ at these points. The method consists of three main steps:

- *Step 1 – Preprocessing:* If no other information is available (e.g., from a given parametric or procedural description of the curve), then each unit normal vector $\mathbf{n}_i$ is estimated from the nearest neighbors of the point $\mathbf{p}_i$. A consistent orientation of the normals is achieved by a region–growing–type process. If the data have been sampled from a curve with singularities, then it may be necessary to organize the data into several segments, see [14] for details.
- *Step 2 – Fitting:* We generate an approximate implicit representation of the form

$$f(\mathbf{x}) = \sum_{j \in \mathcal{J}} c_j \; \phi_j(\mathbf{x}) \tag{1}$$

  with certain coefficients $c_j \in \mathbb{R}$, finite index set $\mathcal{J}$ and suitable basis functions ϕ_j. For instance, one may choose tensor–product B-splines with respect to suitable knot sequences, or Bernstein polynomials with respect to a triangle containing the data.
  The coefficients of f are obtained as the minimum of

$$\sum_{i \in \mathcal{I}} f(\mathbf{p}_i)^2 + w_1 ||\nabla f(\mathbf{p}_i) - \mathbf{n}_i||^2 + w_2 \, T, \tag{2}$$

  where w_1 and w_2 are positive weights satisfying $1 > w_1 \gg w_2 > 0$. The first weight controls the influence of the estimated normal vectors $\mathbf{n}_i$ to the resulting curve. As observed in our experiments, increasing the weights w_1 or w_2 can be used to 'push away' unwanted branches of the curve from the region of interest.

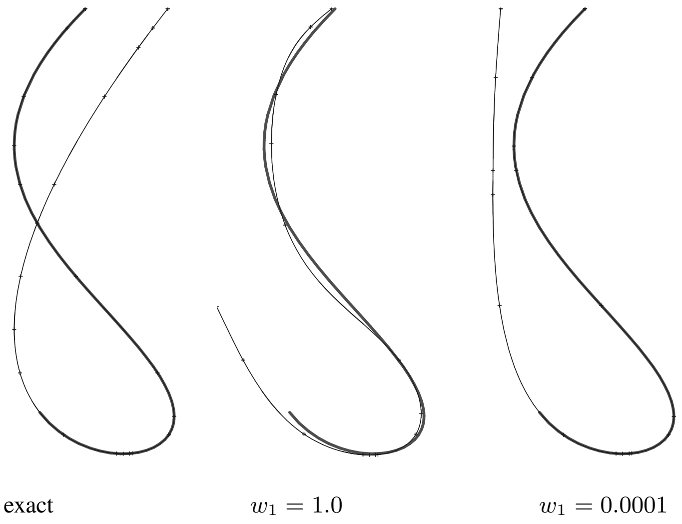

Fig. 1. Exact (left) vs. approximate (center and right) implicitization (thin curves) of a given parametric curve (bold curves), see Example 1.

The tension term T in (2) is added in order to control the shape of the resulting curve. It pulls the approximating curve towards a simpler shape. A possible quadratic tension term is

$$T = \iint_{\Omega} f_{xx}^2 + 2\, f_{xy}^2 + f_{yy}^2 \quad \mathrm{d}x \, \mathrm{d}y \tag{3}$$

This choice of the tension term leads to a positive definite quadratic objective function. Consequently, the coefficients c_j are found by solving a system of linear equations. In the case of tensor–product B-splines, this system is sparse.

- *Step 3 – Iteration:* One may iterate the second step, by replacing the normals $\mathbf{n}_i$ with the gradients $\nabla f(\mathbf{p}_i)$, and re–computing the result. One the one hand, this may help to improve the result of the fitting. On the other hand, it can create problems with unwanted branches. This is described in some detail in [9].

Example 1. We illustrate the behaviour of exact and approximate implicitization by an example. Figure 1 shows the results (algebraic curves of order 4) of both methods (thin curves) for a segment of a rational planar curve of degree 4 (bold curves). The approximate implicitization produces an exact implicitization, but with additional branches and even a singular point in the region of interest. Depending on the choice of w_1, the fitting method produces implicit approximations with different level of accuracy. The weight w_1 can be used to control unwanted branches and singular points. In this example, $w_2 = 0$ has been chosen, and three iterations were applied to improve the result.

3 Approximate implicitization of space curves

After presenting some preliminaries, we discuss the approximate implicitization of two space curves as the intersection of two generalized cylinders and as the intersection of algebraic surfaces which are approximately orthogonal to each other.

3.1 Preliminaries

For any function $f : \mathbb{R}^3 \to \mathbb{R}$, the zero contour (or zero level set) $\mathcal{Z}(f)$ is the set

$$\mathcal{Z}(f) = \{\mathbf{x} \mid f(\mathbf{x}) = 0\} = f^{-1}(\{0\}) \tag{4}$$

A *space curve* C can be defined as the intersection curve of two zero sets of functions f and g,

$$C(f,g) = \mathcal{Z}(f) \cap \mathcal{Z}(g). \tag{5}$$

If both f and g can be chosen as polynomials, then $C(f,g)$ is called an *algebraic* curve. A point $\mathbf{x} \in C(f,g)$ is said to be a *regular* point of the space curve, if the gradient vectors $\nabla f(\mathbf{x})$ and $\nabla g(\mathbf{x})$ are linearly independent. The tangent vector of the space curve is then perpendicular to both gradient vectors.

The two zero contours $\mathcal{Z}(f)$ and and $\mathcal{Z}(g)$ intersect *orthogonally* along the space curve $C(f,g)$, if

$$\nabla f(\mathbf{x}) \cdot \nabla g(\mathbf{x}) = 0 \tag{6}$$

holds for all $\mathbf{x} \in C(f,g)$.

Representing the space curve by two surfaces which intersect orthogonally provides a more robust definition for the curve [1], since small perturbations of the defining two surfaces have less impact on the space curve. It has several additional advantages, e.g., for estimating the *Euclidean distance* of a point to the curve. As a natural generalization of the so–called *Sampson distance* $f(\mathbf{p})/||\nabla f(\mathbf{p})||$, see [11], this distance can be estimated as

$$L = \sqrt{\frac{f^2}{\|\nabla f\|^2} + \frac{g^2}{\|\nabla g\|^2}} \tag{7}$$

In the case of two surfaces which intersect each other orthogonally, L provides a good local (i.e., in the vicinity of the intersection curve) approximation of the distance field.

Example 2. Fig. 2 visualizes this observation. Two surfaces, their intersection curve and a level set of the function L are shown. In the case of two orthogonal surfaces (right), the level set is more similar to a pipe surface than in the general situation (left).

3.2 Intersection of generalized cylinders

A generalized cylinder is obtained by extruding a profile curve $\mathcal{Z}(f)$ along a straight line. If the straight line is parallel to one of the coordinate axes, say the z–axis, then the zero contour of any function of the form $(x,y,z) \to f(x,y)$ defines such a generalized cylinder.

This simple observation leads to algorithm 1 which generates an approximate implicit representation of a space curve.

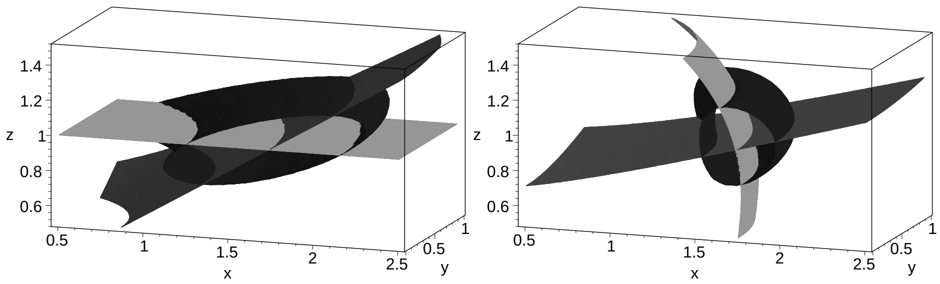

Fig. 2. Two surfaces, their intersection curve and a level set of the function L, see Example 2.

Algorithm 1 Approximate implicitization by generalized cylinders

---

Input A parametric space curve $\mathbf{C}$ or a set of sampled points $\mathbf{p}_i$.
Output An implicit representation of the given space curve as the intersection of two generalized cylinders.

1: Project the parametric space curve $\mathbf{C}$ (the points $\mathbf{p}_i$) orthogonally into two orthogonal planes (e.g. xy-plane and xz-plane).
2: Apply an implicitization method to the data in xy-plane and xz-plane. Let the bivariate functions $f(x,y)$ and $g(x,z)$ define the implicit curves in xy-plane and xz-plane respectively.
3: Define the two generalized cylinders by the polynomials $f(x,y)$ and $g(x,y)$ respectively.
4: Represent the curve $C(x,y,z)$ as the intersection of the two generalized cylinders $f(x,y)$ and $g(x,z)$.

Example 3. The left plot in Figure 4 (see page 7) shows a space curve (white) which is represented as the intersection of two generalized cylinders $\mathcal{Z}(f)$ (black) and $\mathcal{Z}(g)$ (grey), where $f = f(x,y)$ and $g = g(x,z)$.

3.3 Approximately orthogonal representation

Our method for generating an approximate implicitization by two approximately orthogonal surfaces is based on the following simple observation.

Lemma 4. *At all regular points $\mathbf{x} \in C(f,g)$, the gradients of the two functions*

$$F(\mathbf{x}) = \|\nabla f(\mathbf{x})\|\; g(\mathbf{x}) + \|\nabla g(\mathbf{x})\|\; f(\mathbf{x}) \tag{8}$$
$$G(\mathbf{x}) = \|\nabla f(\mathbf{x})\|\; g(\mathbf{x}) - \|\nabla g(\mathbf{x})\|\; f(\mathbf{x}) \tag{9}$$

are orthogonal.

This observation can be verified by a direct computation.

Remark 5. This result cannot be used at points where the two original surfaces intersect each other tangentially. In the case of two generalized cylinders produced by Algorithm 1, this happens only if the curve $\mathbf{C}$ has a tangent which lies in a plane that is perpendicular to both projection planes. One may easily choose the two projection planes such that this is not the case.

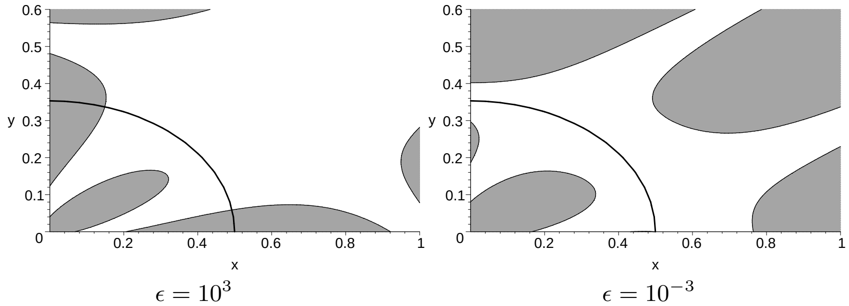

Fig. 3. Approximation of the scalar field $||\nabla f||$, see Example 6.

Clearly, even if the function f and g are piecewise polynomials, neither F nor G are piecewise polynomials in general. We propose to approximate them by piecewise polynomials, as follows.

The functions $\|\nabla f\|$ and $\|\nabla g\|$ depend on x, y and x, z respectively. We would like to approximate them by two piecewise polynomials $\bar{f}(x, y)$ and $\bar{g}(x, z)$ in the area of interest, which is the region near the zero contours of the functions f and g. The two approximating functions are to minimize

$$\iint\limits_{\Omega'} w(f)\ (\bar{f} - \|\nabla f\|)^2\ \mathrm{d}x\,\mathrm{d}y \quad \text{and} \quad \iint\limits_{\Omega''} w(g)\ (\bar{g} - \|\nabla g\|)^2\ \mathrm{d}x\,\mathrm{d}z \qquad (10)$$

where w is a suitable weight function. For instance, one may use

$$w(h) = \frac{1}{h^2 + \epsilon}, \qquad (11)$$

where $\epsilon > 0$ is used in order to avoid division by zero.

Note that the objective functions depend quadratically on $\bar{f}$ and $\bar{g}$. Consequently, if these approximants are represent as a linear combination of certain basis functions (such as tensor–product B-splines), similar to (1), then the minimizers of (10) can be computed by solving symmetric positive definite systems of linear equations. In the B-spline case, these systems are sparse. The coefficients of the equations have to be evaluated by numerical integration, e.g., by Gaussian quadratures.

Example 6. We consider the gradient field of $f = 4x^2 + 8y^2 - 1$ on $[0, 1] \times [0, 0.6]$ and approximate the scalar field $||\nabla f|| = 8\sqrt{x^2 + 4y^2}$ by a quadratic polynomial. For different values of ϵ we obtain different approximations. The white region in Fig. 3 shows where the error $|\bar{f} - ||\nabla f||\,|$ is below 0.15. For smaller values of ϵ, this region follows the elliptic arc $\mathcal{Z}(f)$, which is shown as a black line.

Algorithm 2 combines the previous algorithm with the approximation of the norms of the gradients.

Example 7. We consider a given a space curve and apply the two algorithms to it. Figure 4 shows the approximate implicitization by two generalized cylinders (left) and by two approximately orthogonal algebraic surfaces (right).

Algorithm 2 Approximate implicitization by approximately orthogonal surfaces

Input A parametric space curve $\mathbf{C}$ or a set of sampled points $\mathbf{p}_i$.
Output An approximate implicit representation as the intersection of two approximately orthogonal surfaces.

1: Run Steps 1, 2, 3 of Algorithm 1.
2: Approximate $\|\nabla f\|$ and $\|\nabla g\|$ by polynomials or piecewise polynomials $\bar{f}$ and $\bar{g}$ by minimizing (10).
3: Introduce the two auxiliary function F and G as in (8) and (9), where the norms of the gradients are replaced by their piecewise polynomial approximants.
4: Represent the given curve as the intersection of the two approximately orthogonal algebraic surfaces F, G.

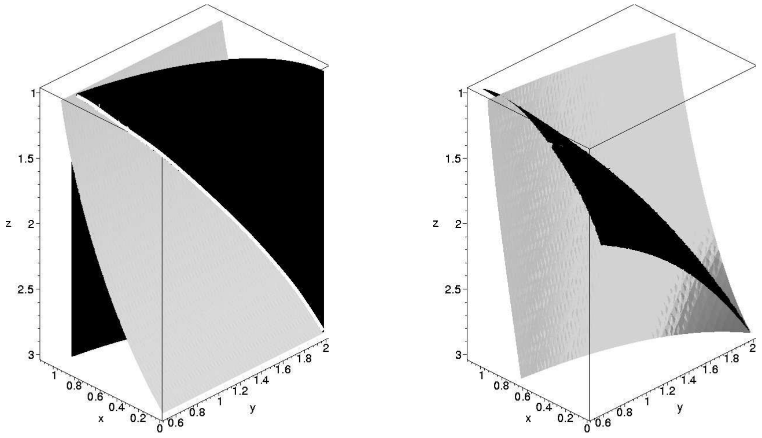

Fig. 4. Approximate implicitization of a space curve using Algorithm 1 (left, intersection of two generalized cylinders) and 2 (right, intersection of two approximately orthogonal surfaces).

4 Approximate Implicitization of Surfaces of Revolution

A surface of revolution is obtained by rotating a profile curve $\mathbf{q}(v)$ about (e.g.) the z–axis. We propose two techniques for generating an approximate implicit representation by a piecewise polynomial. Both techniques reduce the problem to the implicitization problem of a planar curve.

4.1 Implicitization via elimination

First we apply a method for approximate (or exact) implicitization to the profile curve in the rz–plane, where the radius r denotes the distance to the z–axis. For example, one may use the method which was described in Section 2. We obtain an implicit representation of the form $f(r, z) = 0$, where f is a (piecewise) polynomial.

In order to obtain an implicit representation of the form $g(x, y, z) = 0$, one could substitute $r = \sqrt{x^2 + y^2}$. However, the resulting scalar field

$$(x, y, z) \mapsto f(\sqrt{x^2 + y^2}, z) \tag{12}$$

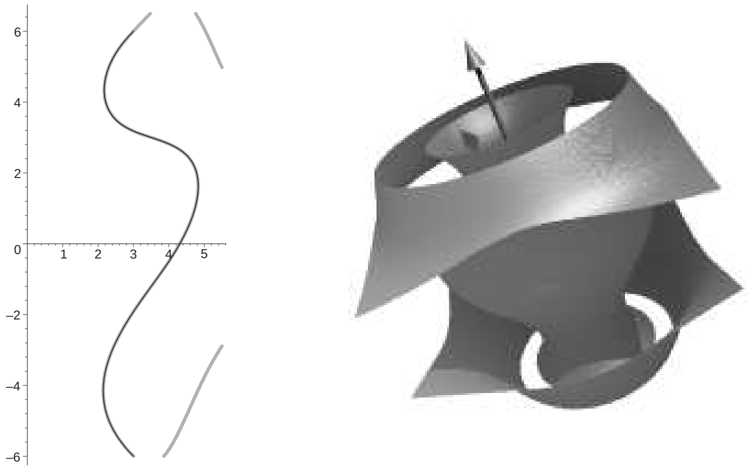

Fig. 5. Approximate implicitization of a surface of revolution using elimination, see Example 8. Left: profile curve, right: the surface.

is no longer given by a piecewise polynomial representation, due to the square root. Instead, we eliminate r using a resultant,

$$g(x, y, z) = \mathrm{Res}_r(f(r, z), r^2 - x^2 - y^2). \tag{13}$$

Clearly, the resultant can be evaluated only if f is a polynomial. In the case of a piecewise polynomial (spline function), this approach has to be applied to the polynomial segments.

Example 8. We apply the technique of Section 2 to the profile curve (black line) shown in Figure 5 (left) and obtain an approximate implicitization by a bi–quartic tensor–product polynomial (grey curve). After computing the resultant, this leads to an approximate implicit representation of the the corresponding surface of revolution (right). The function g is a tensor–product polynomial in x, y, z of degree (8,8,8). Only even powers of x and y are present. Note that the approximate implicitization produces two additional branches, which do not intersect the surface.

This method for approximate implicitization of surfaces of revolution has two major drawbacks.

- First, in the case of a piecewise polynomial representation $f(r, z) = 0$ of the profile curve, the resulting piecewise polynomial g will not necessarily inherit the smoothness properties of f. E.g., if f is a C^1 spline function, then g will not necessarily be C^1.
- Second, even if the approximate implicitization of the profile curve has no unwanted branches and singular points in the region of interest, these problems may be introduced by the eliminating r, see Example 9. Indeed, this elimination is equivalent to computing the polynomial g from

$$g(x, y, z) = f(-\sqrt{x^2 + y^2}, z) \cdot f(\sqrt{x^2 + y^2}, z). \tag{14}$$

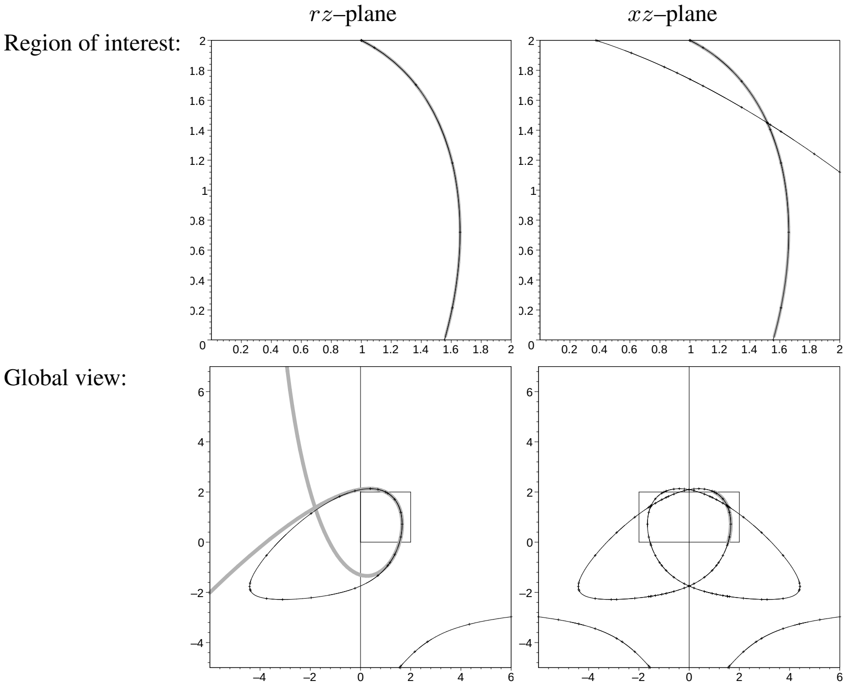

Fig. 6. The elimination of r may produce additional branches and singular points. Top row: Region of interest $[0,2]^2$, Bottom row: global view. Left: Approximate implicitization $\mathcal{Z}(f)$ of the profile curve in the rz–plane. Right: Intersection of the approximate implicitization $\mathcal{Z}(g)$ with the xz–plane. The original profile curve is shown in grey.

Note that this produces indead a polynomial, since only even power of the square root are present! The product (14) leads to a *symmetrized version* of the approximate implicitization of the profile curve. Consequently, additional branches from the half–plane $r < 0$ may cause problems.

Example 9. Approximate implicitization of the profile curve (a cubic Bézier curve) by a cubic polynomial using the method described in Section 2 produces an implicit curve without additional branches and singular points, see Fig. 6, left. However, these problems are present after the elimination step (13), see Fig. 6, right. The reason for this phenomenon can be seen from the global view (bottom row in the picture): the elimination produces a symmetrized version of the approximate implicitization. Note that methods for exact implicitization of the profile curve have similar problems.

Remark 10. The first problem can be resolved by using Eq. (14) instead of (13).

4.2 Implicitization via substitution

In order to avoid the problems of the first approach, we propose to implicitize the profile curve $\mathbf{q}(v)$ in the rz-plane by the zero contour of a bivariate function $f(r^2, z)$. The

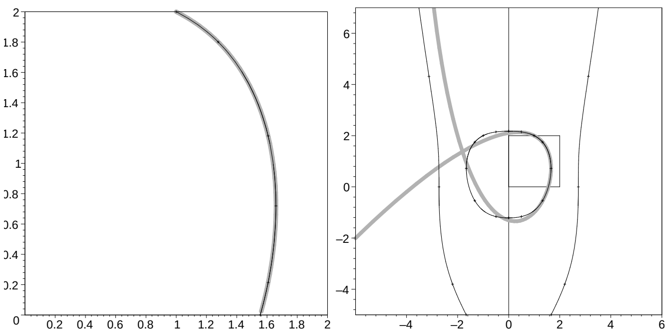

Fig. 7. Approximate implicitization of a surface of revolution via substitution avoids potential problems with additional branches and unwanted singular points. Left: Region of interest, right: global view. The original profile curve is shown in grey.

bivariate function $f(r^2, z)$ can be chosen from the space of all bivariate functions with even power in r. We may use any basis (e.g., tensor–product B–splines) and express the bivariate function $f(r^2, z)$ as

$$F(r^2, z) = \sum_{i \in \mathcal{I}} c_i \; \phi_i(r^2, z) \tag{15}$$

with real coefficients c_i, where $\mathcal{J}$ is a certain index set. The method for approximate implicitization described in Section 2 is applied to this representation. The approximate implicit representation of the surface of revolution is then obtained by a substitution,

$$g(x, y, z) = F(x^2 + y^2, z). \tag{16}$$

Example 11. We apply this approach to the profile curve of Example 9, using a polynomial F of total degree 3. The implicit equation of the profile curve has degree (6,3), and the approximate implicit equation of the surface of revolution has degree (6,3,3). As shown in the figures, we may achieve a similar accuracy in the region of interest by using an approximate implicitization of the profile curve that is symmetric with respect to the axis of revolution. Due to this symmetry, no problems with unwanted branches and singular points are present.

Example 12. We consider the discretized profile curve shown in Fig. 8, left, and apply the method of Section 2 to it. The function F is a bi–quadratic tensor–product spline function whose domain is the union of the cells shown in the figure. This leads to an approximate implicit representation of the profile curve (Fig. 8, center) and of the surface (right) of degree $4(\times 4) \times 2$. In the surface case, the spline function is defined with respect to ring–shaped cells, obtained by rotating the cells shown in the left figure.

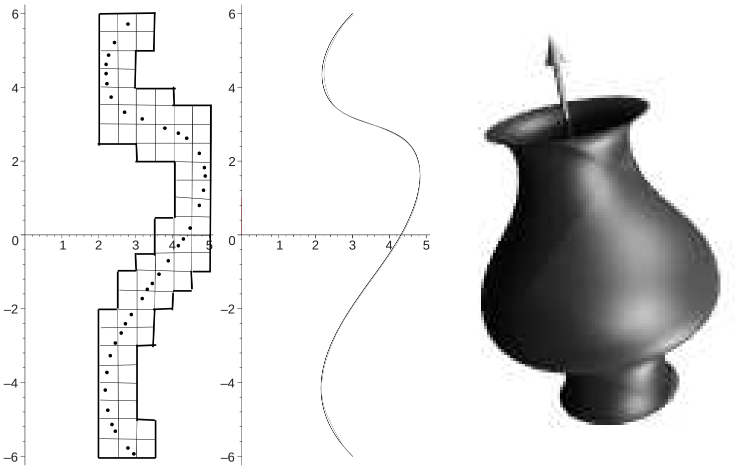

Fig. 8. Approximate implicitization of a surface of revolution of degree $4 \times 4 \times 2$, using a biquadratic spline function F, see Example 12.

5 Conclusion

Several techniques for approximate implicitization of space curves and surfaces of revolution have presented. These techniques are based on algorithms for (exact or approximate) implicitization of planar curves. In the case of space curves, a representation of two approximately orthogonal surface can be obtained, which provides several advantages, such as a geometrically robust definition of the curve and the possibility to obtain a good approximation of the distance field to a space curve. As shown in the case of surfaces of revolution, only approximate implicitization is able to produce a representation which is free of unwanted branches and singular points in the region of interest.

Acknowledgments

This research has been supported by the European Commission through project IST-2001-35512 'Intersection algorithms for geometry based IT-applications using approximate algebraic methods' (GAIA II), and by the Austrian Science Fund (FWF) in the frame of the special research area SFB F013.

References

1. Aigner, M., Jüttler, B., Kim, M.–S.. Analyzing and enhancing the robustness of implicit representations, in: Geometric Modelling and Processing 2004, IEEE Press, 131–140.
2. Chuang, J., Hoffmann, C.: On local implicit approximation and it's applications. ACM Trans. Graphics **8**, 4:298–324, (1989)
3. Corless, R., Giesbrecht, M., Kotsireas, I., Watt, S.: Numerical implicitization of parametric hypersurfaces with linear algebra. In: AISC'2000 Proceedings, Springer, LNAI 1930.

4. Cox, D., Little, J., O'Shea, D,: Using algebraic geometry, Springer Verlag, New York 1998.
5. Cox, D., Goldman, R., Zhang, M.: On the validity of implicitization by moving quadrics for rational surfaces with no base points, J. Symbolic Computation, **11**, (1999)
6. Dokken, T., et al.: Intersection algorithms for geometry based IT-applications using approximate algebraic methods, EU project IST–2001–35512 GAIA II, 2002-2005.
7. Dokken, T. and Thomassen, J., Overview of Approximate Implicitization, in: Topics in Algebraic Geometry and Geometric Modeling, AMS Cont. Math. **334** (2003), 169–184.
8. Gonzalez-Vega, L.: Implicitization of parametric curves and surfaces by using multidimensional Newton formulae. J. Symb. Comput. **23** (2-3), 137-151 (1997)
9. Jüttler, B.: Least-squares fitting of algebraic spline curves via normal vector estimation, in: Cipolla, R., Martin, R.R. (eds.), The Mathematics of Surfaces IX, Springer, London, 263–280, 2000.
10. Jüttler, B., Felis, A.: Least–squares fitting of algebraic spline surfaces, Adv. Comp. Math. **17** (2002), 135–152.
11. Sampson, P. D. Fitting conic sections to very scattered data: an iterative refinement of the Bookstein algorithm, Computer Graphics and Image Processing **18** (1982), 97–108.
12. Sederberg, T., Chen F.: Implicitization using moving curves and surfaces. Siggraph 1995, **29**, 301–308, (1995)
13. Shalaby, M. F., Thomassen, J. B., Wurm, E. M., Dokken, T., Jüttler, B., Piecewise approximate implicitization: Experiments using industrial data, in: Algebraic Geometry and Geometric Modelling (Mourrain, B., Elkadi, M., Piene, R., eds.), Springer, in press.
14. Wurm, E., Jüttler, B., Approximate implicitization via curve fitting, in Kobbelt, L., Schröder, P., Hoppe, H. (eds.), Symposium on Geometry Processing, Eurographics / ACM Siggraph, New York 2003, 240-247.
15. Wurm, E., Thomassen, J., Jüttler, B., Dokken, T.: Comparative Benchmarking of Methods for Approximate Implicitization, in: Neamtu, M., and Lucian, M. (eds.), Geometric Design and Computing, Nashboro Press, Brentwood 2004, 537–548.