Locally refined approximate implicitisation for design and manufacturing

Andrea Raffo, Oliver Barrowclough, Heidi Dahl, Tor Dokken, Michael Floater and Georg Muntingh
andrea.raffo@sintef.no : SINTEF ICT, Department of Applied Mathematics, P.O. Box 124 Blindern, NO-0314 Oslo, Norway

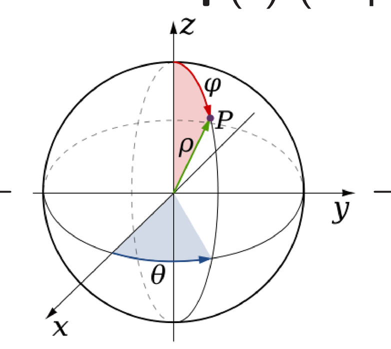
SINTEF

Introduction

Modern Computer Aided Design (CAD) systems are currently based on two complementary representations of curves and surfaces, according to the manipulation they are involved in: **implicit and parametric representations**. Actually:

- in the former, the points $\mathbf{x}$ belonging to a specific curve or surface must satisfy an algebraic equation of the form $q(\mathbf{x}) = 0$;
- in the latter, each choice of one (resp. two) parameters determines a certain point of a curve (resp. surface), that is $\mathbf{x} = \mathbf{p}(t)$ (resp. $\mathbf{x} = \mathbf{p}(s,t)$).

$$x^2 + y^2 + z^2 = r^2 \longleftarrow$$

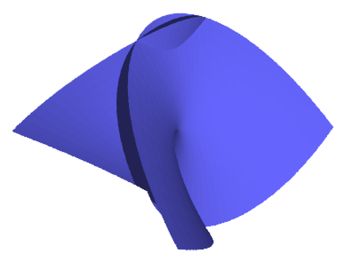


$$\longrightarrow \begin{cases} x = r \sin\theta \cos\varphi, \\ y = r \sin\theta \sin\varphi, \\ z = r \cos\theta \end{cases}$$

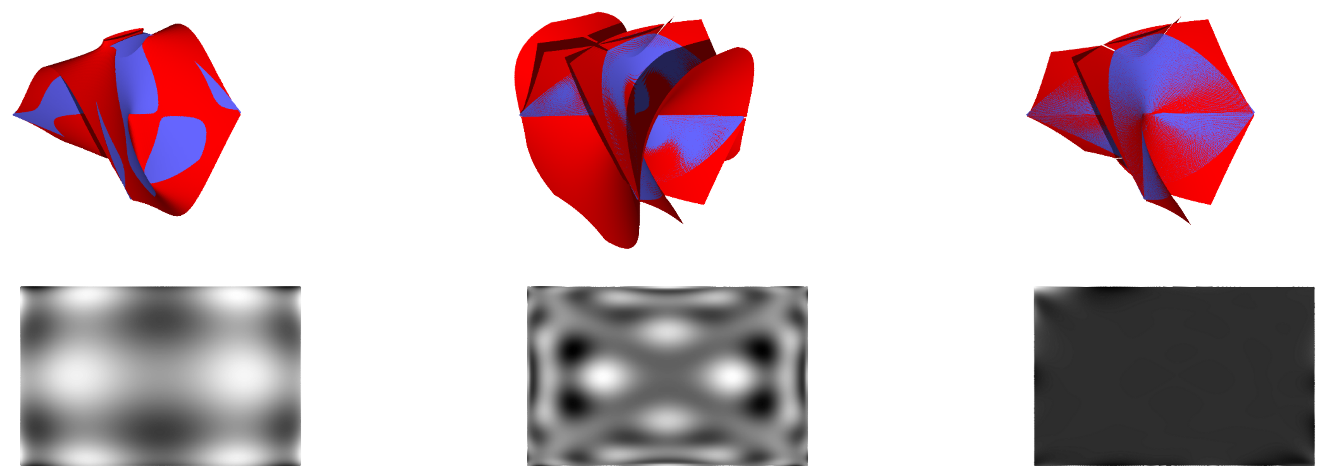
Exact and approximate implicitisation

It is well known that for a parametric rational hypersurface (here: curves in 2D, surfaces in 3D) it is possible to compute the respective implicit form in a process called **implicitisation**. Since the two representations are complementary, several methods for the passage between them have been developed through the years.

- In **elimination theory**, the problem of implicitisation is solved by the elimination of the parametric variables. The result is a curve or surface represented by a single polynomial. There are different computational challenges:
  - **Additional solutions**. The computed implicit polynomial may contain additional factors respect to the implicit equation of the algebraic variety. Numerically, this undesired effect may be unsolvable due to floating point representation.
  - **Numerical stability**. While in CAGD (Computer Aided Geometric Design) rational parametric curves and surfaces are mostly written in terms of Bernstein polynomials, the traditional techniques of implicitisation are based on monomial bases. The passage between the two bases is highly numerically instable.
  - **High polynomial degrees**.The exact implicit representation may be characterized by a high degree, making this form computational expensive and contributing to the numerical instability.
  - **Self-intersections and unwanted branches**. This problem is not directly connected to the use of a specific technique of implicitisation, but to the exact implicit form itself.
- In **approximate implicitisation** (see [1] and [2]), new algorithms for an ''accurate'' single polynomial approximation are introduced on common CAGD tools such as Bézier and Bernstein polynomials.

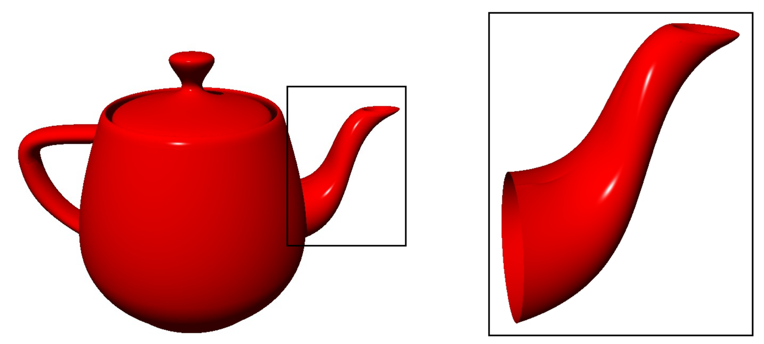


The parametrically defined Enneper 3-degree surface (9-degree implicit form) illustrates strengths and weaknesses of the approximate implicitisation by a single polynomial.



Exact and approximate piecewise implicitisation

Approximate implicitisation can be performed piecewise by dividing the model into smooth components. This approach is of great interest in applications such as computer graphics, where the models usually cannot be described by means of a single polynomial.



In this example, each of the components of the teapot is a bicubic Bezier component, with bidegree (3,3).

	Exact m	Approx. m
rim	9	4
upper body	9	3
lower body	9	3
upper handle	18	4
lower handle	18	4
upper spout	18	5
lower spout	18	6
upper lid	13	3
lower lid	9	4
bottom	15	3

The main application of implicitisation in computer graphics is that it is much faster to ray trace implicit representations than parametric representations. Ray tracing is a high quality rendering method where it is possible to achieve photorealism. Despite the great potential of this technique, one problem is a lack of regularity between the components when computed or approximated non-simultaneously.

Tensor-product B-splines and LR-splines

Traditional B-splines and **NURBS** are formulated, in an n-dimensional space, as tensor products of univariate B-splines. Therefore, a refinement in one of the univariate B-splines will cause the insertion of an entire new row or column of knots in the multivariate B-spline, increasing computational complexity and leading to an unnecessary data explosion.

Locally Refined B-splines (**LR B-splines**) [3, 4] are an innovative approach for a computationally convenient type of refinement of B-splines.

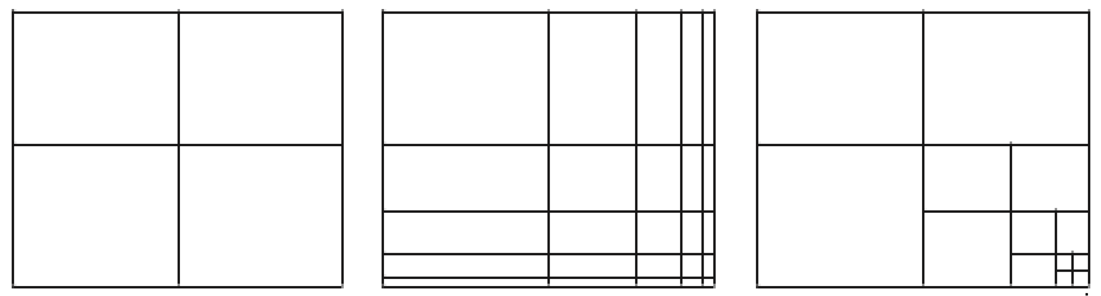
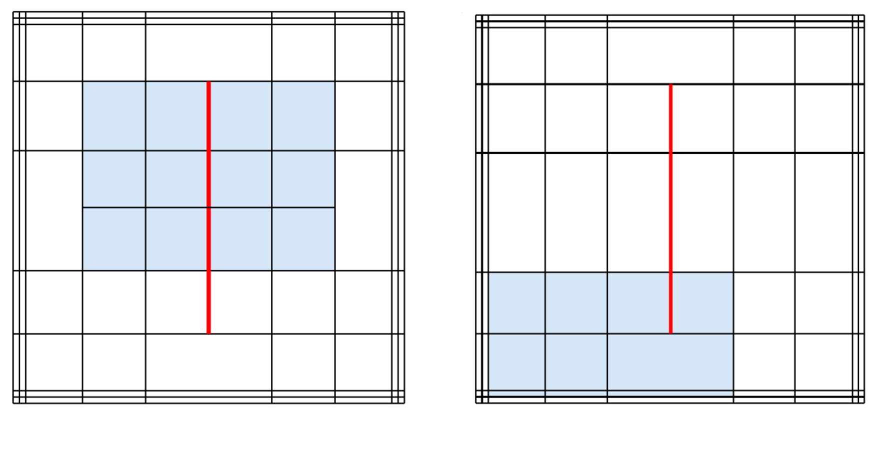


Figure: Initial mesh (left), tensor-product refinement (middle) and truly local refinement (right).

LR B-splines are computed from an initial set of tensor-product B-splines by applying local refinement algorithms, where each spline is split only if its support is completely traversed by the inserted meshline. Therefore, they can be seen as a generalization of the notion of B-splines on tensor product meshes.



LR B-splines use on the approximate implicitisation problem

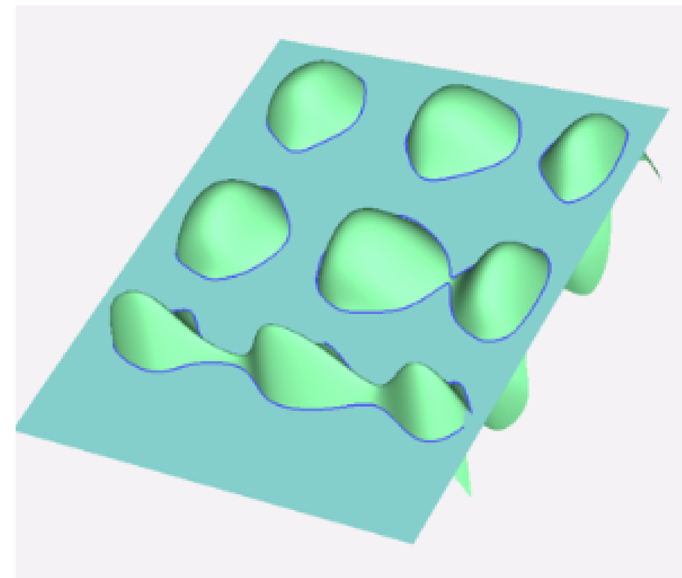
The use of LR B-splines in approximate implicitisation makes it possible to:

- **Keep the degree low**. A low degree is useful both from a computational (numerical stability, computational complexity,...) and from a geometric viewpoint (problem of extra branches when the degree is "high").
- **Add degrees of freedom where they are needed**. The ability to refine locally ensures that the degrees of freedom are present in the area of interest (again: numerical stability, computational complexity,...).
- **Guarantee watertight models**. Three-dimensional models may be affected by small gaps. In applications such as 3D printing, a better model is required.
- **Avoid self-intersection**. This feature is accomplished using the first two properties.

Question: can we establish criteria guaranteeing correct behavior for a given degree?

An application: intersection problems

An important feature of CAD systems is their ability to perform Boolean operations, such as the **intersection of curves/surfaces**. Given two bounded surfaces, then the intersection can be either empty or made up of points, curves, surfaces regions, possibly combined. The difficulty of finding the solution depends, in general, on the relative behavior of the surfaces along the intersection itself.



When the curves/surfaces are available in both the parametric and implicit form (at least one representation for each algebraic variety in the considered intersection), the problem is simplified by the combination of the two expressions: the intersection of 2D algebraic curves (resp. surfaces) can be reduced, from two polynomial equations, to a single univariate (resp. bivariate) polynomial equation.

In an approximate implicitisation approach, a rational parametric surface $\mathbf{p}(s,t)$, $(s,t) \in \Omega \subset \mathbb{R}^2$, is approximated by an algebraic equation $q(x,y,z) = 0$ of degree m. The composition of the implicit and the parametric representations can be written as:

$$q(\mathbf{p}(s,t)) = (\mathbf{D}\mathbf{b}^T)\alpha(s,t),$$

where $\mathbf{b}$ is a vector consisting of the unknown coefficients of the algebraic surface to be found, $\alpha(s,t)$ is a vector consisting of rational basis functions and $\mathbf{D}$ is a matrix having as elements products of m components of $\mathbf{p}(s,t)$ (see [5]).

References

[1] Tor Dokken, *Aspects of intersection algorithms and approximation*, PhD thesis, University of Oslo, 1997.
[2] Oliver Barrowclough, *Approximate methods for change of representation and their applications in CAGD*, PhD thesis, University of Oslo, 2012.
[3] Kjetil André Johannessen, Trond Kvamsdal and Tor Dokken, *Isogeometric analysis using LR B-splines*, Comput. Methods Appl. Mech. Engrg. **269** (2014), 471-514.
[4] Tor Dokken, Tom Lyche and Kjell Fredrik Pettersen, *Polynomial splines over locally refined box-partitions*, Computer Aided Geometric Design, **30** (2013), 331-356.
[5] Geir Hasle, Knut-Andreas Lie and Ewald Quak, *Geometric Modelling, Numerical Simulation and Optimization: Applied Mathematics at SINTEF*, Springer, (2007).