Computation of Adjoints for Surfaces
Annual Report for 1998

Josef Schicho
RISC-Linz
Johannes Kepler University, A-4040 Linz
e-mail: Josef.Schicho@risc.uni-linz.ac.at
http: http://www.risc.uni-linz.ac.at/people/jschicho

March 10, 1999

1 Goal of the project

The goal is to devise an efficient algorithm for the following problem.

Input: A surface in 3-space, given implicitly.
Two integers n, m.
Output: The 'm-adjoints of degree n' of the surface.

An 'm-adjoint of degree n' of a surface is a polynomial of degree n that vanishes with a certain order (depending on m) on the singularities of the surface. For example, look at the surface in figure 1. It has a singular line L. The polynomial given by G in figure 2 is a 1-adjoint of degree 1.

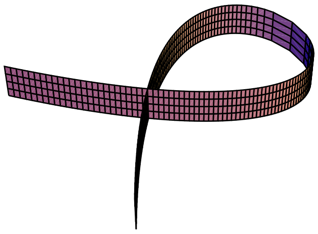

Figure 1: A surface with a singular line.

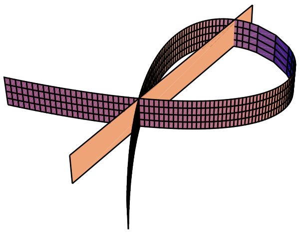

Figure 2: An 1-adjoint.

The set of all m-adjoints of degree n forms a vector-space. The output of the algorithm should be a basis.

2 Applications

Computation of adjoints is important because it is currently the only way to solve the parameterization problem: Given an equation in three variables, defining a surface in 3-space, find a parametrization of the surface in terms of rational functions in two parameters. There are surfaces for which no such parametrization exists, so an algorithmic solution of the parametrization problem has to decide existence first.

Currently, we have only one parameterization algorithm (due to Schicho), except for very limited classes of surfaces, like quadric and cubic surfaces (due to Abhyankar/Bajaj and Sederberg/Snively) or canal surfaces (Pottmann). Computing the adjoints is the most difficult step in Schicho's general parameterization algorithm.

The parametrization problem has become especially important, because it is the most difficult step in computing NURBS- (non-uniform rational B-spline) -represention, which are used for many applications in Computer Aided Design and Manufacture, such as

- reliable surface plotting and display,
- motion display (computing transformations),
- computing cutter offset surfaces,
- computing curvatures for shading and colouring,
- and many others.

NURBSs are already a de facto standard in CAD/CAM environments and their popularity is still growing.

Moreover, the computation of adjoints is important because it can be applied to solve classification problems for surfaces.

3 Main Methods

A natural approach for adjoints computation is to use a resolution of the singularities. Computing resolutions of singularities is generally considered as an extremely hard problem (see Teitelbaum,Lipman). Thus, we proposed to try another appoach, namely using fractional power series, in the original proposal.

Surprisingly enough, it turned out that effective resolution of singularities *is* possible with nowadays' means. See also section 4

4 Main results achieved in 1998

As a fact which turned out as a surprise even to experts, Bodnár and Schicho succeeded in implementing an algorithm for effective desingularization which is basically due to Villamayor. Before this implementation, the existing resolution programs could only deal with curves. Surface desingularization was considered as an "open problem in constructive algebraic geometry" in (Eisenbud 1993). The new package desingularizes hypersurfaces of arbitrary dimension!

The foundations have been laid by Villamayor (1991) in giving a constructive proof for the celebrated result of Hironaka claiming the existence of a resolution. His proof implicitly entails the algorithm, however there were some open computational questions.

One of the main difficulties in automated resolution is the following. The resolution is obtained by *blowing up* subvarieties in the singular locus. Now, it is quite easy to blowup a point. But for more complicated blowing up centers, the construction is not so trivial. There are implementations for performing blowing ups in various computer algebra systems like SINGULAR or Macaulay. But these computations are time consuming and the output is a very cumbersome thing. In the resolution algorithm, we need to apply blowing up repeatedly, which would be rather difficult when using these procedures.

Bodn'ar and Schicho solved this problem by working simultanuously with affine coordinates on the one side and regular parameters on the other side. Doing this, they compute blowing ups without any computations more difficult than simple polynomial arithmetic and differentiation.

We have published a full implementation in Maple on the web (`http:www.-risc.uni-linz.ac.at/projects/basic/adjoints/blowup`). The output of the whole resolution algorithm is a tree of charts, where

- the root node is $\mathbf{A}^n$, the ambient space of the given hypersurface,
- the children of each node are obtained from the parent node by blowing up or by some other operation described in section **??** (cover and exchange),
- the proper transform of the given hypersurface in the leaf nodes is non-singular, and the total transform is a normal crossing divisor.

The output tree may be viewed by "clicking through the resolution" in an html hierarchy.

To indicate the relevance of this contribution, I want to tell that the author's have been asked twice to contribute to books on singularities! The first contribution is already accepted and will appear in 1999.

Other published results are related to the main application, namely the parametrization problem: Schicho could give a theoretical upper bounds for the degrees of the polynomials in the smallest possible parametrization (technical report, invited talk as a guest, and journal paper which has already been accepted for publication). This bound happens to be polynomial. In the same paper, also a series of examples is provided showing that the given bounds are sharp up to a factor 2.

B. Matasaru has completed a technical report on the detailed complexity analysis of a well-known algorithm for adjoints computation in the easier, but similar case of curves.

Currently, W. Heiss is working on an implementation of Schicho's parametrization algorithm in Maple. He is expected to finish the implementation in May 1999.

The second approach for computing adjoints (fractional power series) is still pursued. We plan to finish reports, programs, etc., during 1999.

5 International Cooperations

During the year 1998, we had cooperations with the following people.

R. Sendra (Madrid): application of the parametrization algorithm to the special case of offset surfaces;

J. Gutierrez (Santander): started joint work on reparametrization;

O. Villamayor (Madrid): provided help in the clarification of details of his desingularization algorithm;

S. Encinas (Valladolid): provided help in the clarification of details in Villamayor's algorithm;

H. Hauser (Innsbruck): suggested extensions to our desingularization implementation.

6 Scientific Output

Papers published or accepted for publication in journals:

1. J. Schicho. Rational parametrization of surfaces. *J. Symb. Comp.*, 26(1):1–30, July 1998.
2. J. Schicho. A degree bound for the parameterization of a rational surface. *J. Pure Appl. Alg.*, 1999. to appear.

Submitted papers:

1. G. Bodnar and J. Schicho. A computer program for the resolution of singularities. Technical Report 99-03, RISC-Linz, Univ. Linz, A-4040 Linz, 1999.
2. G. Bodnar and J. Schicho. Automated resolution of singularities for hypersurfaces. Technical Report 99-06, RISC-Linz, Univ. Linz, A-4040 Linz, 1999.

Technical reports:

1. B. Matasaru and J. Schicho. The complexity analysis for an algorithm to compute adjoints for curves due to Jeremy Teitelbaum. Technical Report 98-21, RISC-Linz, Univ. Linz, A-4040 Linz, 1998.

7 Talks (ad 5)

1. G. Bodnar and J. Schicho. On algorithmic desingularization of hypersurfaces, 1998. talk at IMACS-ACA'98, Prague.
2. J. Schicho. Rational parameterization of real algebraic surfaces, 1998. invited talk at Univ. of Cantabria, Spain.
3. J. Schicho. A degree bound for the parameterization of a rational surface, 1996. invited talk at Univ. Complutense, Spain.
4. J. Schicho. Rational parameterization of real algebraic surfaces, 1998. talk at ISSAC-98, Rostock.