Numerical algorithms for certified topological and geometrical description of singular curves

Advisors	Marc Pouget	Guillaume Moroz
Emails	`marc.pouget@inria.fr`	`guillaume.moroz@inria.fr`
Location	INRIA Nancy, France	

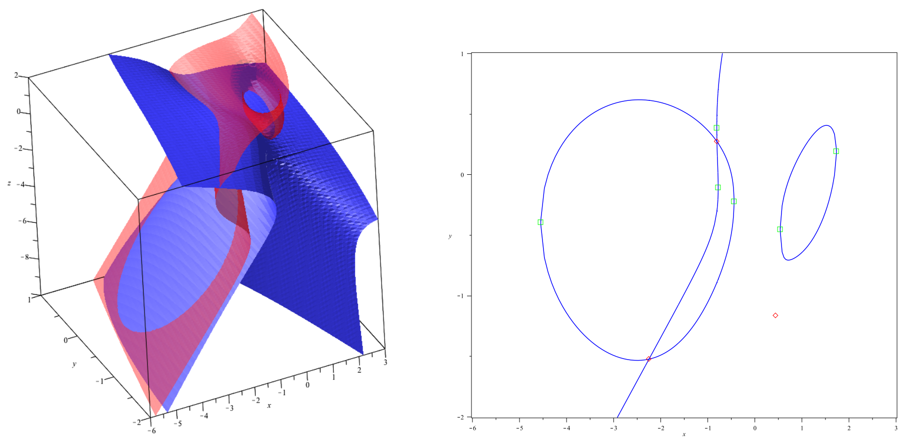

Figure 1: Left: two surfaces intersecting along a smooth 3D curve. Right: the projection on a plane of this 3D curve is singular, but generically only with node singularities.

Introduction

Motivation. The description of the topology and geometry of curves arises in a wide range of applications, from scientific visualization to the design of robotic mechanisms. Currently, most software provide either (a) numerical approximations without guarantee for singular curves; or (b) rely on heavier tools of computer algebra to handle singular curves but suffer from efficiency in practice. This internship addresses the design and the implementation of *certified numerical* algorithms for restricted classes of singular curves, namely the projection of the intersection of two surfaces (Fig. 1) or the projection of the silhouette of a surface.

Topological and geometrical descriptions. In general, a curve $\mathcal{C}$ solution of a system of polynomial equations cannot be represented exactly. A common approach is to compute a set of points at a distance smaller than a given treshold from $\mathcal{C}$. The main drawback of this approach is that it misses topological information, such that the number of connected components, self-intersections or isolated points.

Another approach is to compute a piecewise-linear graph that has the same topology as $\mathcal{C}$, as in Figure 2. The main challenge is to perform this computation efficiently with numerical and certified algorithms.

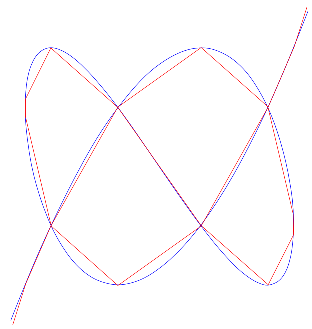

Figure 2: The red piecewise-linear graph is a topology preserving vectorial representation of the blue curve (computed with the software Isotop[1]).

Numerical methods. When the curve is smooth, numerical methods ([Rum10]) are efficient to represent its topology and geometry with a piecewise linear graph. However, as we can see in Figure 1, the projection of a curve of $\mathbb{R}^3$ is not smooth in general. No numerical methods can yet certify its topology. A challenge is to design a numerical method to represent efficiently the topology and geometry of the projection in this case.

Objective of the internship

Let $\mathcal{S}$ be a curve of $\mathbb{R}^3$ given by the two polynomial equations system:

$$S \left\{ \begin{array}{rcl} f(x,y,z) & = & 0 \\ g(x,y,z) & = & 0 \end{array} \right. .$$

The projection of $\mathcal{S}$, denoted by $\mathcal{C}$, is the set of points in the xy-plane such that S has a solution. As shown in Figure 1, this set is a curve that can contain self-intersection and isolated points. For any point p in the xy-plane, denote by N_p the number of points of $\mathcal{C}$ that project on p. If $\mathcal{C}$ is generic, then N_p can be either zero, one or two. Computing the topology of $\mathcal{C}$ can be reduced to isolating with certification the points p such that $N_p = 2$.

Recall that fast numerical algorithms based on the Newton method can only certify systems with as many equations as unknowns. The first challenge is to find a system with as many equations as unknowns such that its solutions are exactly the points p such that $N_p = 2$. From there, according to the motivations of the candidate, two lines of research can be explored.

On the one hand, design of a complete certified numerical algorithm to isolate these points. This requires the implementation of several numerical algorithms based on the Newton method, and their experimental comparison.

On the other hand, generalize the system to higher dimensions. For curves defined by three polynomial equations in four variables, what system describes the points in the xy-plane such that $N_p = 2$? Is it possible to generalize this approach for curves in $\mathbb{R}^n$?

Profile of the internship candidate

The candidate should have a taste for both mathematics (geometry or numerical analysis) and computer science. Programming skills are appreciated (C/C++, Python or Mathlab).

References

[Rum10] Siegfried M. Rump. Verification methods: Rigorous results using floating-point arithmetic. *Acta Numerica*, 19:287–449, 5 2010.

[1] `http://www.loria.fr/equipes/vegas/cgi-bin/isotop_form.pl`