Symbolic-Numeric Algorithms for Invariants of Plane Curve Singularities

Mădălina Hodorog[1], Bernard Mourrain[2], Josef Schicho[1]

[1]Johann Radon Institute for Computational and Applied Mathematics,
Doctoral Program "Computational Mathematics"
Johannes Kepler University Linz, Austria
Research Institute for Symbolic Computation

[2]INRIA Sophia-Antipolis, France

Doctoral Program "Computational Mathematics" Status Seminar,
Strobl-Austria
October 6-8, 2010

Table of contents

Motivation

We investigate the topology (i.e. roughly speaking the shape) of plane complex algebraic curves. These curves can be identified with objects in $\mathbb{R}^4$ we cannot visualize! We sketch the equivalent objects in $\mathbb{R}^2$ for a rough "idea"!

Motivation

For instance,
We visualize the topology of the algebraic curve $\mathcal{C} = \{(x, y) | -x^3 - xy + y^2 = 0\}$ in $\mathbb{R}^2$!
We notice an "involved" topology around the point $(0, 0)$, which is called singularity!

Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Motivation

Roughly, if we consider $\mathcal{C} = \{(x,y) | -x^3 - xy + y^2 = 0\}$ then

For topology of $(0,0)$ of $\mathcal{C}$

In $\mathbb{R}^2$

we visualize the solutions of $-x^3 - xy + y^2 = 0$

In $\mathbb{R}^4$

we study invariants of $(0,0)$(i.e. algebraic link) (Alexander polynomial, etc)

Motivation

Our goal is to compute and

Roughly, if we consider $\mathcal{C} = \{(x,y) | -x^3 - xy + y^2 = 0\}$ then

For topology of $(0,0)$ of $\mathcal{C}$

In $\mathbb{R}^2$

we visualize the solutions of $-x^3 - xy + y^2 = 0$

In $\mathbb{R}^4$

to study invariants of $(0,0)$ of $\mathcal{C}$ (algebraic link) (Alexander polynomial, etc) and invariants of $\mathcal{C}$ (genus, etc)

Motivation

Our goal is also to design for:

Rationals and floating point numbers → Symbolic-numeric algorithms (in GENOM3CK) (in Axel) → to compute Alexander polynomial, genus, etc

Because at present (from our knowledge) there exists only for:

Rationals → Symbolic algorithm (in Singular) → to compute the Alexander polynomial

Rationals → Symbolic algorithm (in Maple with algcurves) → to compute the genus

Other numerical algorithms for genus: C. Wampler's group (Bertini system), R. Sendra's group.

Motivation

For instance:

```
> with(algcurves);
[AbelMap, Siegel, Weierstrassform, algfun_series_sol, differentials, genus,
   homogeneous, homology, implicitize, integral_basis, is_hyperelliptic,
   j_invariant, monodromy, parametrization, periodmatrix, plot_knot,
   plot_real_curve, puiseux, singularities]
```

> $f := x^2\,y + y^4$

$$f := x^2\,y + y^4$$

> $genus(f, x, y)$

$$-1$$

> $g := 1.02 \cdot x^2\,y + 1.12 \cdot y^4$

$$g := 1.02\,x^2\,y + 1.12\,y^4$$

> $genus(g, x, y)$

```
Error, (in content/polynom) general case of floats not handled
>
```

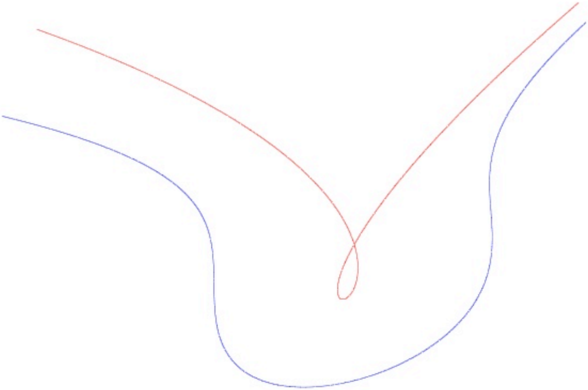
Problem specifications

- Input:
  - $f(x,y) \in \mathbb{C}[x,y]$ squarefree with exact and inexact coefficients;
  - $\mathcal{C} = \{(x,y) \in \mathbb{C}^2 | f(x,y) = 0\} \subseteq \mathbb{C}^2 \simeq \mathbb{R}^4$ of degree m;
  - $\epsilon \in \mathbb{R}_+^*$ input parameter.
- Output:
  - The set of numerical singularities of $\mathcal{C}$;
    IF $\mathcal{C} \cap S_\epsilon$ (with S_ϵ the sphere of radius epsilon centered in the singularity $(0,0)$ of $\mathcal{C}$) has no singularities, THEN :
  - A set of invariants for each numerical singularity:
    - ϵ-algebraic link;
    - ϵ-Alexander polynomial;
    - ϵ-Milnor number, ϵ-delta-invariant;
  - A set of invariants from knot theory for each ϵ-algebraic link:
    - ϵ-diagram, ϵ-crossings, ϵ-arcs, ϵ-genus, ϵ-determinant;
    - ϵ-unknotting number, ϵ-linking number, ϵ-colorability.
  - A set of invariants for $\mathcal{C}$:
    - ϵ-genus, ϵ-Euler characteristic.

  ELSE "false", i.e. $\mathcal{C} \cap S_\epsilon$ has singularities.

Ill-posedness of the problem

The problem is ill-posed! Small changes in input produce huge changes in the output!
Example. Let $s_1 = (0,0)$ of $\mathcal{C} = \{(x,y) \in \mathbb{R}^2 | -x^3 - xy + y^2 = 0\}$ and
$s_2 = (0,0)$ of $\mathcal{D} = \{(x,y) \in \mathbb{R}^2 | -x^3 - xy + y^2 - 0.01 = 0\}$!
The topology of $(0,0)$ is not stable under small changes in input!



The same situation happens in $\mathbb{R}^4$, but we cannot visualize it!

Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Techniques for dealing with the ill-posedness

How to deal with the **ill-posedness** of **a problem**?

- We construct numerical methods that approximate solutions to ill-posed problems, that are stable under small changes of the input! (i.e. regularization method)
- Similar methods are subjects of *approximate algebraic computation* in order to compute: greatest common divisor of polynomials, root of polynomials, etc.

Techniques for dealing with the ill-posedness

How to deal with the **ill-posedness** of **our problem**?

- *Example.* For $s_1 = (0,0)$ of $\mathcal{C} = \{(x,y) \in \mathbb{R}^4 | -x^3 - xy + y^2 = 0\}$ and $s_2 = (0,0)$ of $\mathcal{D} = \{(x,y) \in \mathbb{R}^4 | -x^3 - xy + y^2 - 0.01 = 0\}$, we compute their ϵ-*algebraic links* denoted $L_\epsilon(s_1), L_\epsilon(s_2)$.

Note 1: For sufficiently small ϵ, L_ϵ are stable under small changes in the input and they characterize the topology of s_1, s_2.

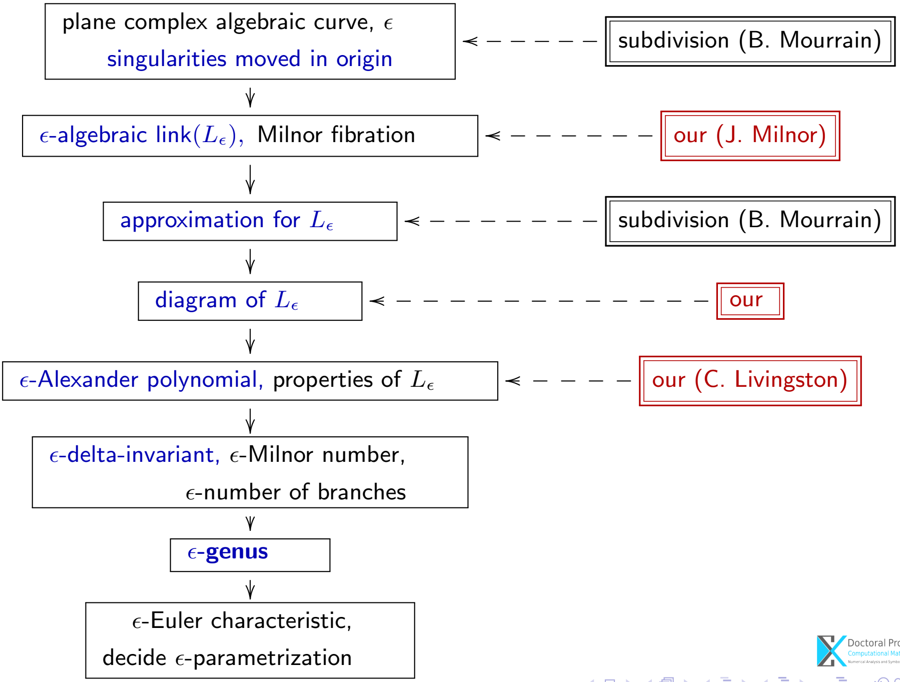
Note 2: From L_ϵ we compute the ϵ-Alexander polynomial. This polynomial is a complete invariant for L_ϵ! (Yamamoto's result)

- If the ϵ-Alexander polynomials of $L_\epsilon(s_1), L_\epsilon(s_2)$ are equal, then s_1, s_2 have the same topology, else they have different topology!

Note 3: From L_ϵ, ϵ-Alexander polynomial we compute other invariants.

Strategy for solving the problem

We split our problem into smaller interdependent subproblems!


plane complex algebraic curve, ϵ
singularities moved in origin
subdivision (B. Mourrain)
ϵ-algebraic link(L_ϵ), Milnor fibration
our (J. Milnor)
approximation for L_ϵ
subdivision (B. Mourrain)
diagram of L_ϵ
our
ϵ-Alexander polynomial, properties of L_ϵ
our (C. Livingston)
ϵ-delta-invariant, ϵ-Milnor number,
ϵ-number of branches
ϵ-genus
ϵ-Euler characteristic,
decide ϵ-parametrization

plane complex algebraic curve, ϵ
singularities moved in origin ← algebraic geometry

↓

ϵ-algebraic link(L_ϵ), Milnor fibration ← topology

↓

approximation for L_ϵ ← algorithms, data structures

↓

diagram of L_ϵ ← computational geometry

↓

ϵ-Alexander polynomial, properties of L_ϵ ← combinatorial methods

↓

ϵ-delta-invariant, ϵ-Milnor number,
ϵ-number of branches

↓

ϵ-genus

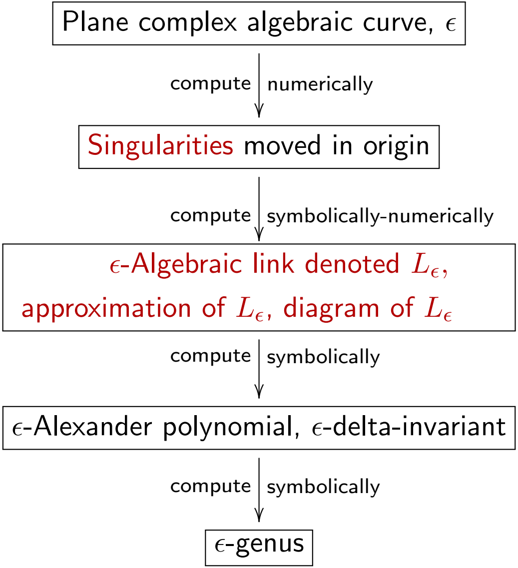
↓

ϵ-Euler characteristic,
decide ϵ-parametrization

Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Algorithms for invariants of plane curve singularities



Doctoral Program Computational Mathematics
Numerical Analysis and Symbolic Computation

First

Plane complex algebraic curve, ϵ

↓ compute numerically

Singularities

↓ compute symbolically-numerically

ϵ-Algebraic link denoted L_ϵ,
approximation of L_ϵ, diagram of L_ϵ

↓ compute symbolically

ϵ-Alexander polynomial, ϵ-delta-invariant

↓ compute symbolically

ϵ-genus

Algorithm for the singularities of the curve

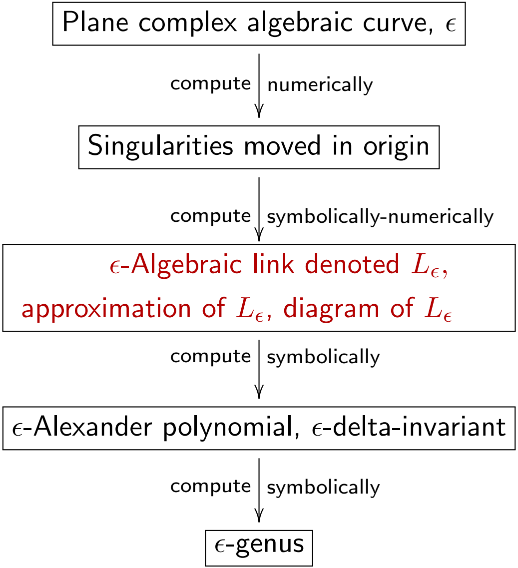
- Input:
  - $f(x, y) \in \mathbb{C}[x, y]$ squarefree with exact and inexact coefficients
  - $\mathcal{C} = \{(x, y) \in \mathbb{C}^2 | f(x, y) = 0\}$ complex algebraic curve of degree m.
- Output:
  - $Sing(\mathcal{C}) = \{(x_0, y_0) \in \mathbb{C}^2 | f(x_0, y_0) = 0, \frac{\partial f}{\partial x}(x_0, y_0) = 0, \frac{\partial f}{\partial y}(x_0, y_0) = 0\}$
- Method: We solve the system in $\mathbb{C}^2 : f(x_0, y_0) = \frac{\partial f}{\partial x}(x_0, y_0) = \frac{\partial f}{\partial y}(x_0, y_0) = 0.$

  We use subdivision methods from Axel.

  We get the numerical singularities, i.e. a list of points P in the plane s.t.:
  - the value of $f(x, y)$ and its derivatives in the points from P are small;
  - every singularity from $Sing(\mathcal{C})$ is in the neighborhood of one point from P.

  QUESTION: Other method (with implementation) available!?

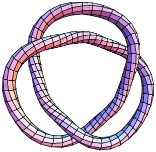
Next


Plane complex algebraic curve, ϵ
compute
numerically
Singularities moved in origin
compute
symbolically-numerically
ϵ-Algebraic link denoted L_ϵ,
approximation of L_ϵ, diagram of L_ϵ
compute
symbolically
ϵ-Alexander polynomial, ϵ-delta-invariant
compute
symbolically
ϵ-genus

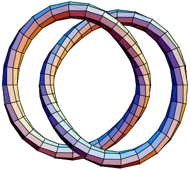
Algorithm for the ϵ-algebraic link

- A **knot** is a piecewise linear or a differentiable simple closed curve in $\mathbb{R}^3$.
- A **link** is a finite union of disjoint knots.
- Links resulted from the intersection of a given curve with the sphere are called **algebraic links**.

Trefoil Knot



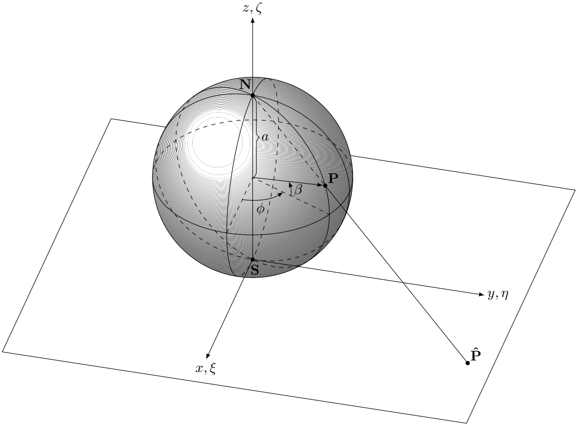
Hopf Link



Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Algorithm for the ϵ-algebraic link

- How do we compute the link of a plane curve singularity?
  - use the stereographic projection;



Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Algorithm for the ϵ-algebraic link

1. Let $\mathcal{C} = \{(a,b,c,d) \in \mathbb{R}^4 | f(a,b,c,d) = 0\}$ with $(0,0,0,0) \in Sing(\mathcal{C})$.

2. For $f(a,b,c,d) = R(a,b,c,d) + iI(a,b,c,d)$ with $R(a,b,c,d), I(a,b,c,d) \in \mathbb{R}[a,b,c,d]$, rewrite $\mathcal{C} = \{(a,b,c,d) \in \mathbb{R}^4 | R(a,b,c,d) = I(a,b,c,d) = 0\}$.

3. Intersect $\mathcal{C}$ with a sphere $S_\epsilon = \{(a,b,c,d) \in \mathbb{R}^4 | a^2 + b^2 + c^2 + d^2 = \epsilon^2\}$ and obtain $X = \mathcal{C} \bigcap S_\epsilon \subset \mathbb{R}^4$.

4. For $N \in S_\epsilon \setminus \mathcal{C}$, consider the stereographic projection $\pi : S_\epsilon \setminus \{N\} \rightarrow \mathbb{R}^3, (a,b,c,d) \mapsto (u = \frac{a}{\epsilon - d}, v = \frac{b}{\epsilon - d}, w = \frac{c}{\epsilon - d})$, and compute $\pi^{-1} : \mathbb{R}^3 \rightarrow S_\epsilon \setminus \{N\}, \pi^{-1}(u,v,w) \mapsto (a = ..., b = ..., c = ..., d = ...)$.

Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Algorithm for the ϵ-algebraic link

4. For $\mathcal{C} = \{(a,b,c,d) \in \mathbb{R}^4 | R(a,b,c,d) = I(a,b,c,d) = 0\}$, project $X = \mathcal{C} \cap S_\epsilon$ from $\mathbb{R}^4$ to $\mathbb{R}^3$ with the stereographic projection π and compute
$\pi(X) = \{(u,v,w) \in \mathbb{R}^3 | \exists (a,b,c,d) = \pi^{-1}(u,v,w,) \in X = \mathcal{C} \cap S_\epsilon\}$,
$\pi(X) = \{(u,v,w) \in \mathbb{R}^3 | R(a,b,c,d) = I(a,b,c,d) = 0\}$.

5. Obtain $\pi(X) = \{(u,v,w) \in \mathbb{R}^3 | g(u,v,w) = h(u,v,w) = 0\}$ with $g, h \in \mathbb{R}[u,v,w]$.

Remark!
$\pi(X)$ is an implicit algebraic curve in $\mathbb{R}^3$ given as the intersection of two surfaces in $\mathbb{R}^3$ with the defining equations g, h.
For small $\epsilon, \pi(X) := L_\epsilon$ is an algebraic link, (based on Milnor's result), i.e. $\mathcal{C} \cap S_\epsilon$ has no singularities.

Approximation of the ϵ-algebraic link

We use Axel for implementing the proposed algorithm.

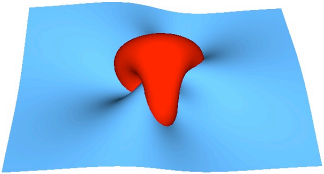
- For $\mathcal{C} = \{(x,y) \in \mathbb{C}^2 | x^3 - y^2 = 0\} \subset \mathbb{R}^4, \epsilon = 1$ we compute with the algorithm in Axel:

Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Approximation of the ϵ-algebraic link

We use Axel for implementing the proposed algorithm.

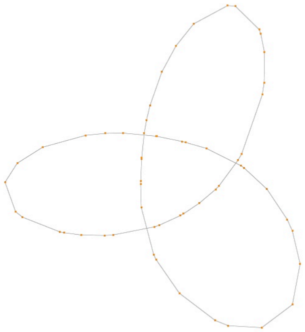
- For $\mathcal{C} = \{(x, y) \in \mathbb{C}^2 | x^3 - y^2 = 0\} \subset \mathbb{R}^4, \epsilon = 1$ we compute with the algorithm in Axel:
- $\pi(\mathcal{C} \cap S) = \pi(X) := L_\epsilon =$
  $= \{(u, v, w) \in \mathbb{R}^3 | g(u, v, w) = 0, h(u, v, w) = 0\}$



Approximation of the ϵ-algebraic link

We use Axel for implementing the proposed algorithm.

- For $\mathcal{C} = \{(x, y) \in \mathbb{C}^2 | x^3 - y^2 = 0\} \subset \mathbb{R}^4, \epsilon = 1$ we compute with the algorithm in Axel:
- $\pi(\mathcal{C} \cap S) = \pi(X) := L_\epsilon =$ $= \{(u, v, w) \in \mathbb{R}^3 | g(u, v, w) = 0, h(u, v, w) = 0\}$
- $Graph(L_\epsilon) = \langle \mathcal{V}, \mathcal{E} \rangle$ with $\mathcal{V} = \{p = (m, n, q) \in \mathbb{R}^3\}$ $\mathcal{E} = \{(i, j) | i, j \in \mathcal{V}\}$

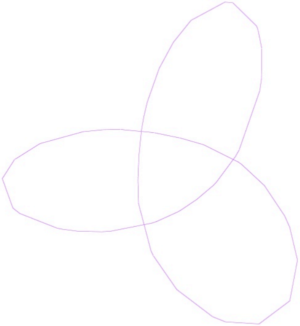


Doctoral Program Computational Mathematics Numerical Analysis and Symbolic Computation

Approximation of the ϵ-algebraic link

We use Axel for implementing the proposed algorithm.

- For $\mathcal{C} = \{(x, y) \in \mathbb{C}^2 | x^3 - y^2 = 0\} \subset \mathbb{R}^4, \epsilon = 1$ we compute with the algorithm in Axel:
- $\pi(\mathcal{C} \cap S) = \pi(X) := L_\epsilon =$
  $= \{(u, v, w) \in \mathbb{R}^3 | g(u, v, w) = 0, h(u, v, w) = 0\}$
- $Graph(L_\epsilon) = \langle \mathcal{V}, \mathcal{E} \rangle$ with
  $\mathcal{V} = \{p = (m, n, q) \in \mathbb{R}^3\}$
  $\mathcal{E} = \{(i, j) | i, j \in \mathcal{V}\}$
- s.t. $Graph(L_\epsilon) \cong_{isotopic} L_\epsilon$

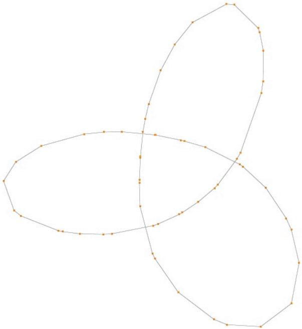


Doctoral Program Computational Mathematics Numerical Analysis and Symbolic Computation

Approximation of the ϵ-algebraic link

We use Axel for implementing the proposed algorithm.

- For $\mathcal{C} = \{(x,y) \in \mathbb{C}^2 | x^3 - y^2 = 0\} \subset \mathbb{R}^4, \epsilon = 1$ we compute with the algorithm in Axel:
- $\pi(\mathcal{C} \cap S) = \pi(X) := L_\epsilon =$
  $= \{(u,v,w) \in \mathbb{R}^3 | g(u,v,w) = 0, h(u,v,w) = 0\}$
- $Graph(L_\epsilon) = \langle \mathcal{V}, \mathcal{E} \rangle$ with
  $\mathcal{V} = \{p = (m,n,q) \in \mathbb{R}^3\}$
  $\mathcal{E} = \{(i,j) | i,j \in \mathcal{V}\}$
- s.t. $Graph(L_\epsilon) \cong_{isotopic} L_\epsilon$
- $Graph(L_\epsilon)$ is a piecewise linear approximation of L_ϵ
- Why Axel? It is the only system to implement a method which returns such an approximation!
  QUESTION: Other method (with implementation) available!?

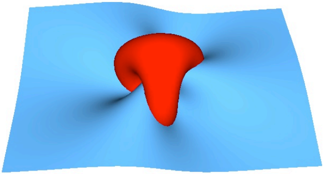


Doctoral Program Computational Mathematics Numerical Analysis and Symbolic Computation

Approximation of the ϵ-algebraic link

We use Axel for implementing the proposed algorithm.

- For $\mathcal{C} = \{(x, y) \in \mathbb{C}^2 | x^3 - y^2 = 0\} \subset \mathbb{R}^4, \epsilon = 1$
- and $L_\epsilon =$
  $= \{(u, v, w) \in \mathbb{R}^3 | g(u, v, w) = 0, h(u, v, w) = 0\}$

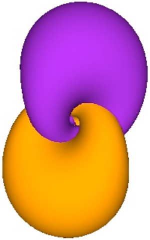


Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Approximation of the ϵ-algebraic link

We use Axel for implementing the proposed algorithm.

- For $\mathcal{C} = \{(x, y) \in \mathbb{C}^2 | x^3 - y^2 = 0\} \subset \mathbb{R}^4, \epsilon = 1$
- and $L_\epsilon =$
  $= \{(u, v, w) \in \mathbb{R}^3 | g(u, v, w) = 0, h(u, v, w) = 0\}$
- we also compute (for visualization reasons)
  $\mathcal{S}' = \{(u, v, w) \in \mathbb{R}^3 | g(u, v, w) + h(u, v, w) = 0\}$
  $\mathcal{S}'' = \{(u, v, w) \in \mathbb{R}^3 | g(u, v, w) - h(u, v, w) = 0\}$

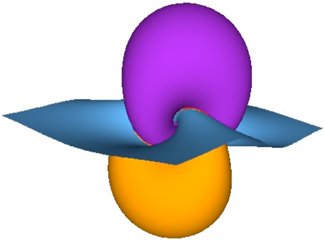


Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Approximation of the ϵ-algebraic link

We use Axel for implementing the proposed algorithm.

- For $\mathcal{C} = \{(x, y) \in \mathbb{C}^2 | x^3 - y^2 = 0\} \subset \mathbb{R}^4, \epsilon = 1$
- and $L_\epsilon =$
  $= \{(u, v, w) \in \mathbb{R}^3 | g(u, v, w) = 0, h(u, v, w) = 0\}$
- we also compute (for visualization reasons)
  $\mathcal{S}^{'} = \{(u, v, w) \in \mathbb{R}^3 | g(u, v, w) + h(u, v, w) = 0\}$
  $\mathcal{S}^{''} = \{(u, v, w) \in \mathbb{R}^3 | g(u, v, w) - h(u, v, w) = 0\}$
- L_ϵ is the intersection of any 2 of the surfaces:
  $g(u, v, w), h(u, v, w)$
  $g(u, v, w) + h(u, v, w), g(u, v, w) - h(u, v, w)$

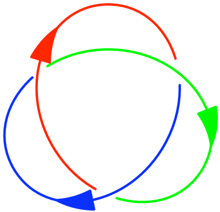


Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Algorithm for the diagram of the ϵ-algebraic link

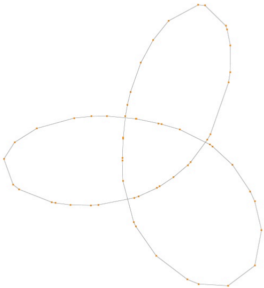
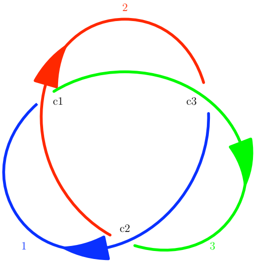
When we work with (algebraic) links, we work with a special projection of them (**diagram**), containing the information on each double point (**crossing**) telling which branch goes over and which under.

- An **arc** is the part of a diagram between two undercrossings.



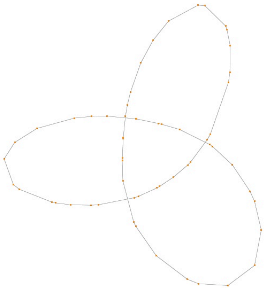
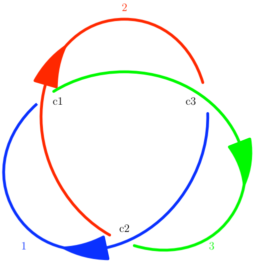
- Example. Diagram with 3 crossings and 3 arcs

Algorithm for the diagram of the ϵ-algebraic link

-  $\overset{?}{\Rightarrow}$ 

- $G(L_\epsilon) = \langle P, E \rangle$ $\qquad\qquad$ $D(L_\epsilon)$
- We need to transform the graph data structure $G(L_\epsilon)$ returned by Axel into the diagram of the algebraic link $D(L_\epsilon)$.

Algorithm for the diagram of the ϵ-algebraic link

-  $\overset{?}{\Rightarrow}$ 

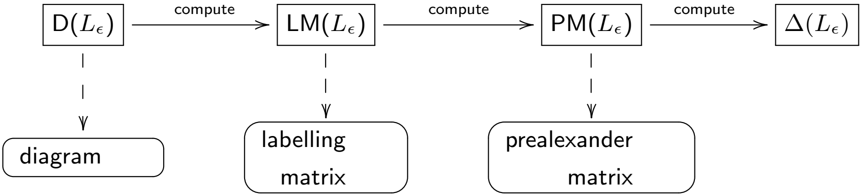
- We developed several computational geometry and combinatorial algorithms!
  ☛ M. Hodorog, J.Schicho. *Computational geometry and combinatorial algorithms for the genus computation problem.* DK 10-07 Report.

  ☛ M. Hodorog, B. Mourrain, J.Schicho. *Topology analysis of complex curves singularities using knot theory.* International Conference on Curves and Surfaces, Avignon, 2010.

Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Why do we need the diagram of the ϵ-algebraic link?

We compute the ϵ-Alexander polynomial of L_ϵ denoted $\Delta(L_\epsilon)$ in 3 combinatorial steps:



In order to compute it, we need D(L_ϵ), the diagram of L_ϵ!

☛ M. Hodorog, B. Mourrain, J. Schicho. *A symbolic-numeric algorithm for computing the Alexander polynomial of a plane curve singularity*. International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. Timișoara, Romania, 2010.

Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Implementation

- *Axel* free algebraic geometric modeler (INRIA Sophia-Antipolis) [a]

http://axel.inria.fr/

[a] Acknowledgements: Julien Wintz

Implementation

- *Axel* free algebraic geometric modeler (INRIA Sophia-Antipolis) [a]
  - written in *C*++;
  - Qt Script for Applications (QSA);
  - Open Graphics Library (OpenGL).

[a] Acknowledgements: Julien Wintz

Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Implementation

- *Axel* free algebraic geometric modeler (INRIA Sophia-Antipolis) [a]
  - written in *C++*;
  - Qt Script for Applications (QSA);
  - Open Graphics Library (OpenGL).
- GENOM3CK-our library in Axel. Support: http://people.ricam.oeaw.ac.at/m.hodorog/software.html and madalina.hodorog@oeaw.ac.at
- ☛ M. Hodorog, B. Mourrain, J. Schicho. *GENOM3CK - A library for GENus cOMputation of plane Complex algebraiC Curves using Knot theory.* International Symposium on Symbolic and Algebraic Computation. Münich, Germany, 2010.

[a] Acknowledgements: Julien Wintz

Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Implementation

- *Axel* free algebraic geometric modeler (INRIA Sophia-Antipolis) [a]
  - written in *C++*;
  - Qt Script for Applications (QSA);
  - Open Graphics Library (OpenGL).
- GENOM3CK-our library in Axel. Support: http://people.ricam.oeaw.ac.at/m.hodorog/software.html and madalina.hodorog@oeaw.ac.at
- ☛ M. Hodorog, B. Mourrain, J. Schicho. *GENOM3CK - A library for GENus cOMputation of plane Complex algebraiC Curves using Knot theory.* International Symposium on Symbolic and Algebraic Computation. Münich, Germany, 2010.
- Version 0.2 of GENOM3CK is released!

[a]Acknowledgements: Julien Wintz

The algorithms and "approximate algebraic computation"

We interpret the algorithms in the frame of approximate algebraic computation!

Plane complex algebraic curve, ϵ

↓ compute numerically

Singularities moved in origin

↓ compute symbolically-numerically

ϵ-Algebraic link denoted L_ϵ,
approximation of L_ϵ, diagram of L_ϵ

↓ compute symbolically

ϵ-Alexander polynomial, ϵ-delta-invariant

↓ compute symbolically

ϵ-genus

The algorithms and "approximate algebraic computation"

With the notations:

- $E : I \to O$ the symbolic algorithm s.t.
  Given $f \in I$ polynomial, compute $E(f)$ the Alexander polynomial (ill-posed)
- $A : I \times \mathbb{R}_+ \to O$ the symbolic-numeric algorithm s.t.
  Given $(f, \epsilon) \in I \times \mathbb{R}_+$, compute the ϵ-Alexander polynomial
- $\forall f \in I \ \forall \delta \in \mathbb{R}_+, f_- : \mathbb{R}_+ \to I, \delta \mapsto f_\delta : |f - f_\delta| \leq \delta$

and based on:

- Milnor's theorem:

$$\lim_{\epsilon \to 0} A(f, \epsilon) = E(f)(\textbf{convergence for exact data}).$$

- $A(f_\delta, \epsilon)$ depends continuously on the perturbed input polynomial f_δ (**continuity**);
- $\exists \alpha : \mathbb{R}_+ \to \mathbb{R}_+$ continuous, monotonically and decreasing with $\lim_{\delta \to 0} \alpha(\delta) = 0$
  s.t. $\forall f \in I \ \forall \delta \in \mathbb{R}_+ : |f - f_\delta| \leq \delta$

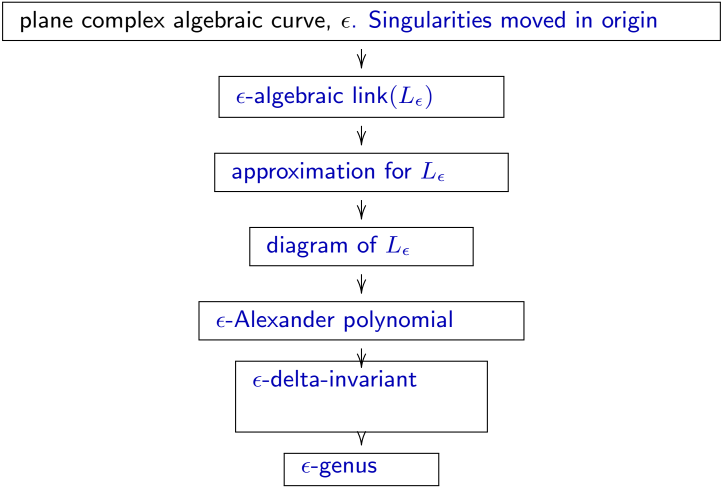
$$\lim_{\delta \to 0} A(f_\delta, \alpha(\delta)) = E(f)(\textbf{convergence for perturbed data}).$$

The algorithm A_ϵ is a regularization.

Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Extension of the algorithms

Originally, we developed the algorithms for the following invariants of algebraic curves:



Extension of the algorithms

We extended the algorithms to compute more invariants of algebraic curves:

plane complex algebraic curve, ϵ. Singularities moved in origin

↓

-algebraic link(L_ϵ), Milnor fibration

↓

approximation for L_ϵ

↓

diagram of L_ϵ

↓

ϵ-Alexander polynomial, properties of L_ϵ

↓

ϵ-delta-invariant, ϵ-Milnor number,ϵ-number of branches

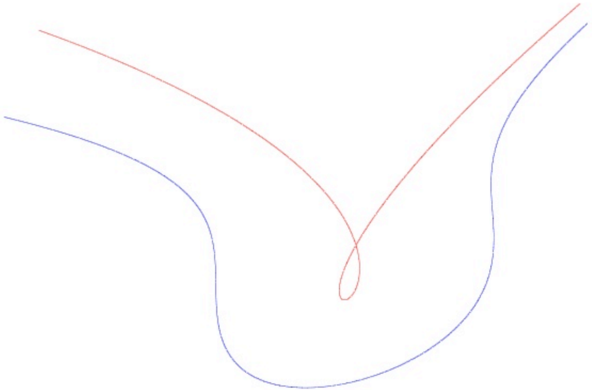
↓

ϵ-genus, ϵ-Euler characteristic, decide ϵ-parametrization

Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Test experiments

Let us review the first example: Let $s_1 = (0,0)$ of $\mathcal{C} = \{(x,y) \in \mathbb{R}^2 | -x^3 - xy + y^2 = 0\}$ and $s_2 = (0,0)$ of $\mathcal{D} = \{(x,y) \in \mathbb{R}^2 | -x^3 - xy + y^2 - 0.01 = 0\}$!
The topology of $(0,0)$ is not stable under small changes of the input!

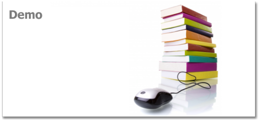


The same situation happens in $\mathbb{R}^4$, but we cannot visualize it!

Test experiments

But the ϵ-algebraic link is stable under small changes of the input for sufficiently small ϵ!

Equation in $\mathbb{C}^2$	**Results**
$-x^3 - xy + y^2, \epsilon = 1.00$	Trefoil, $\Delta(t_1) = t_1^2 - t_1 + 1, \delta = 1, g = 0$
$-x^3 - xy + y^2, \epsilon = 0.25$	Hopf link, $\Delta(t_1, t_2) = 1, \delta = 1, g = 0$
$-x^3 - xy + y^2 - 0.01, \epsilon = 1.00$	Trefoil, $\Delta(t_1) = t_1^2 - t_1 + 1, \delta = 1, g = 0$
$-x^3 - xy + y^2 - 0.01, \epsilon = 0.25$	Hopf link, $\Delta(t_1, t_2) = 1, \delta = 1, g = 0$



Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Conclusion and future work

✓ DONE:

- automatization of symbolic-numeric algorithms for invariants of plane curves singularities in GENOM3CK;
- describe partially algorithms with principles from regularization theory;

✗ TO DO's:

Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Conclusion and future work

✓ DONE:

- automatization of symbolic-numeric algorithms for invariants of plane curves singularities in GENOM3CK;
- describe partially algorithms with principles from regularization theory;
- test experiments show that the algorithms have the continuity and the convergence for exact and perturbed data properties;
- proofs of the continuity and the convergence for exact data property are constructed.

✗ TO DO's:

Doctoral Program Computational Mathematics Numerical Analysis and Symbolic Computation

Conclusion and future work

✓ DONE:

- automatization of symbolic-numeric algorithms for invariants of plane curves singularities in GENOM3CK;
- describe partially algorithms with principles from regularization theory;
- test experiments show that the algorithms have the continuity and the convergence for exact and perturbed data properties;
- proofs of the continuity and the convergence for exact data property are constructed.

✗ TO DO's:

- proof of the convergence for perturbed data property is needed;

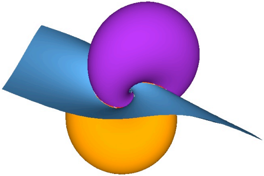
Conclusion and future work

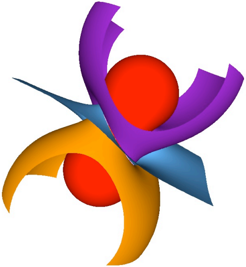
✓ DONE:

- automatization of symbolic-numeric algorithms for invariants of plane curves singularities in GENOM3CK;
- describe partially algorithms with principles from regularization theory;
- test experiments show that the algorithms have the continuity and the convergence for exact and perturbed data properties;
- proofs of the continuity and the convergence for exact data property are constructed.

✗ TO DO's:

- proof of the convergence for perturbed data property is needed;
- include other operations, i.e. from knot theory, algebraic geometry.





Thank you for your attention.