GENOM3CK - A library for GENus cOMputation of plane Complex algebraiC Curves using Knot theory

Mădălina Hodorog\textsuperscript{1}, Bernard Mourrain\textsuperscript{2}, Josef Schicho\textsuperscript{1}

\textsuperscript{1}Johann Radon Institute for Computational and Applied Mathematics, Doctoral Program "Computational Mathematics"  
Johannes Kepler University Linz, Austria

\textsuperscript{2}INRIA Sophia-Antipolis, France

35\textsuperscript{th} International Symposium on Symbolic and Algebraic Computation, Münich-Germany

July 26, 2010
Table of contents

1 Motivation

2 Describing the library
   • Algorithm specifications
   • Short history
   • Interface

3 Testing the library
   • Setting the input data and parameters
   • Demo (Examples)

4 Conclusion
Motivation

Describing the library
- Algorithm specifications
- Short history
- Interface

Testing the library
- Setting the input data and parameters
- Demo (Examples)

Conclusion
Motivation

Why a library for genus computation of plane complex algebraic curves using knot theory (GENOM3CK)?
Motivation

At present, there exists several...

Symbolic algorithms for Genus computation

Maple
CASA
RISC–Hagenberg

Maple
algcurves
Florida University

CASA
INRIA–Roquencourt

Magma
PAFFnormal.lib
GHS attack
Package by F. Hess
Berlin University

Magma

Singular

Singular
normal.lib
Kaiserslautern University

Singular

Axiom

Axiom

PAFF
INRIA–Roquencourt

PAFF
Kaiserslautern University
Motivation

But...

For moderate symbolic input data: they are too expensive

Disadvantages of Symbolic algorithms for Genus computation

For numeric input data: they are unusable
Motivation

For instance, in Maple using `algcurves` package...

```maple
> 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

> genus(f, x, y)

> g := 1.02 * x^2 * y + 1.12 * y^4

> genus(g, x, y)

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

Thus we need...

(Symbolic) Numeric algorithms for Genus computation
Motivation

Hopefully...

- Symbolic-Numeric techniques for genus computation (initiated by J. Schicho).

Other numeric method was reported (in the group of R. Sendra).
1 Motivation

2 Describing the library
   - Algorithm specifications
   - Short history
   - Interface

3 Testing the library
   - Setting the input data and parameters
   - Demo (Examples)

4 Conclusion
Algorithm specifications

- **Input:**
  - $F(x, y) \in \mathbb{C}[x, y]$ squarefree with exact and inexact coefficients;
  - $C = \{(x, y) \in \mathbb{C}^2 \mid F(x, y) = 0\} \subseteq \mathbb{C}^2 \cong \mathbb{R}^4$ of degree $d$;
  - $\epsilon \in \mathbb{R}^*$ input parameter.

- **Output:**
  - $\text{Sing}(C)$ set of singularities of $C$;
  - A set of invariants of $C$:
    - algebraic link of each singularity (topological type);
    - Milnor fibration of each singularity;
    - Alexander polynomial of each algebraic link;
    - $\delta(s) \in \mathbb{N}$, $\delta$-invariant of each singularity $s \in \text{Sing}(C)$;
    - genus $(C) \in \mathbb{Z}$, genus of $C$.

- A set of operations from knot theory on each algebraic link:

- **Method:** shortly presented on the next slides.
Algorithm specifications

- **Input:**
  - $F(x, y) \in \mathbb{C}[x, y]$ squarefree with exact and inexact coefficients;
  - $C = \{ (x, y) \in \mathbb{C}^2 | F(x, y) = 0 \} \subseteq \mathbb{C}^2 \simeq \mathbb{R}^4$ of degree $d$;
  - $\epsilon \in \mathbb{R}^*$ input parameter.

- **Output:**
  - $\text{Sing}(C)$ set of singularities of $C$;
  - A set of invariants of $C$:
    - algebraic link of each singularity (topological type);
    - Milnor fibration of each singularity;
    - Alexander polynomial of each algebraic link;
    - $\delta(s) \in \mathbb{N}$, $\delta$-invariant of each singularity $s \in \text{Sing}(C)$;
    - $\text{genus}(C) \in \mathbb{Z}$, genus of $C$.
  - A set of operations from knot theory on each algebraic link:
    - diagram (crossings, arcs), type of crossings.

- **Method:** shortly presented on the next slides.
\( \mathcal{C} \subseteq \mathbb{R}^4 \) with \( \text{Sing}(\mathcal{C}) \)

Move each \( s \in \text{Sing}(\mathcal{C}) \) in 0

Let 0 singularity of \( \mathcal{C} \subseteq \mathbb{R}^4 \)

\( S_{(0,\varepsilon)} \subseteq \mathbb{R}^4 \) small sphere

\( X = \mathcal{C} \cup S_{(0,\varepsilon)} \subseteq \mathbb{R}^4 \)

\( f : S_{(0,\varepsilon)} \setminus \{(0,0,0,\varepsilon)\} \to \mathbb{R}^3 \)

\( (a, b, c, d) \mapsto (u = \frac{a}{\varepsilon - d}, v = \frac{b}{\varepsilon - d}, w = \frac{c}{\varepsilon - d}) \)

\( f \) is stereographic projection

Subdivision methods
\( C \subseteq \mathbb{R}^4 \) with \( Sing(C) \)

Move each \( s \in Sing(C) \) in 0

Let 0 singularity of \( C \subseteq \mathbb{R}^4 \)

\( C \) defined by \( F(x, y) \in \mathbb{C}[x, y] \)

\( S_{(0,\epsilon)} \subseteq \mathbb{R}^4 \) small sphere

\( X = C \cup S_{(0,\epsilon)} \subseteq \mathbb{R}^4 \)

For sufficiently small \( \epsilon \)

\( f(X) \subseteq \mathbb{R}^3 \) differentiable algebraic link

\( f(X) = \{(u, v, w)|\text{Re}F(\ldots) = \text{Im}F(\ldots) = 0\} \)

Subdivision methods

Genom3ck (Axel)

Adapted Milnor’s research (Our)
$C \subseteq \mathbb{R}^4$ with $Sing(C)$
Move each $s \in Sing(C)$ in 0

Let 0 singularity of $C \subseteq \mathbb{R}^4$

$S_{(0,\epsilon)} \subseteq \mathbb{R}^4$ small sphere

$X = C \cup S_{(0,\epsilon)} \subseteq \mathbb{R}^4$

\[ f(X) \subseteq \mathbb{R}^3 \] differentiable algebraic link

\[ \tilde{f}(X) \subseteq \mathbb{R}^3 \] piecewise linear algebraic link

Let $\pi_f(X) \subseteq \mathbb{R}^2$ projection of $\tilde{f}(X)$

**Subdivision methods**

**Adapted Milnor's research (Our)**

**Algo for $\tilde{f}(X)$**

i.e. topology of $f(X)$

**GENOM3CK (Axel)**
Let \( C \subseteq \mathbb{R}^4 \) with \( \text{Sing}(C) \),
Move each \( s \in \text{Sing}(C) \) in 0

Let 0 singularity of \( C \subseteq \mathbb{R}^4 \)
\( S_{(0,\epsilon)} \subseteq \mathbb{R}^4 \) small sphere
\[ X = C \cup S_{(0,\epsilon)} \subseteq \mathbb{R}^4 \]

\[ f(X) \subseteq \mathbb{R}^3 \] differentiable algebraic link

\[ \tilde{f}(X) \subseteq \mathbb{R}^3 \] piecewise linear algebraic link
Let \( \pi_{f(X)} \subseteq \mathbb{R}^2 \) projection of \( \tilde{f}(X) \)

Alexander polynomial of \( f(X) \)
is a complete invariant of \( f(X) \)

Subdivision methods
Adapted Milnor’s research (Our)
Algo for \( \tilde{f}(X) \)
i.e. topology of \( f(X) \)
Computational geometry algos (Our)
Yamamoto’s result
\( C \subseteq \mathbb{R}^4 \text{ with } \text{Sing}(C) \)
Move each \( s \in \text{Sing}(C) \) in 0
Let 0 singularity of \( C \subseteq \mathbb{R}^4 \)
\( S_{(0,\epsilon)} \subseteq \mathbb{R}^4 \) small sphere
\( X = C \cup S_{(0,\epsilon)} \subseteq \mathbb{R}^4 \)

\[ f(X) \subseteq \mathbb{R}^3 \text{ differentiable algebraic link} \]
\[ \tilde{f}(X) \subseteq \mathbb{R}^3 \text{ piecewise linear algebraic link} \]
Let \( \pi_{f(X)} \subseteq \mathbb{R}^2 \) projection of \( \tilde{f}(X) \)

Alexander polynomial of \( f(X) \)

\( \delta \)-invariant of \( s \)

Subdivision methods
Adapted Milnor’s research (Our)
Algo for \( \tilde{f}(X) \)
Computational geometry algos (Our)
Yamamoto’s result
Milnor’s research
Let $C \subseteq \mathbb{R}^4$ with $\text{Sing}(C)$.

Move each $s \in \text{Sing}(C)$ in 0.

Let 0 singularity of $C \subseteq \mathbb{R}^4$.

$S_{(0,\epsilon)} \subseteq \mathbb{R}^4$ small sphere

$X = C \cup S_{(0,\epsilon)} \subseteq \mathbb{R}^4$

$f(X) \subseteq \mathbb{R}^3$ differentiable algebraic link

$\tilde{f}(X) \subseteq \mathbb{R}^3$ piecewise linear algebraic link

Let $\pi_{f(X)} \subseteq \mathbb{R}^2$ projection of $\tilde{f}(X)$

Alexander polynomial of $f(X)$

$\delta$-invariant of $s$

$\text{genus}(C) = \frac{(d - 1)(d - 2)}{2} - \sum_{s \in \text{Sing}(C)} \delta(s)$
$C \subseteq \mathbb{R}^4$ with $Sing(C)$

Move each $s \in Sing(C)$ in 0

Let 0 singularity of $C \subseteq \mathbb{R}^4$

$S_{(0,\epsilon)} \subseteq \mathbb{R}^4$ small sphere

$X = C \cup S_{(0,\epsilon)} \subseteq \mathbb{R}^4$

$f(X) \subseteq \mathbb{R}^3$ differentiable algebraic link

$\tilde{f}(X) \subseteq \mathbb{R}^3$ piecewise linear algebraic link

$\tilde{f}(X) = S_1 \cap S_2 \cap (S_1 + S_2) \cap (S_1 - S_2)$

Alexander polynomial of $f(X)$

$\delta$-invariant of $s$

$\text{genus}(C) = \frac{(d-1)(d-2)}{2} - \sum_{s \in Sing(C)} \delta(s)$
Short history: GENOM3CK

is written in Axel
C++, Qt Script for Applications (QSA)

is written in Mathemagix
C++

what is Axel?
- algebraic geometric modeler
- INRIA, Galaad team (2006)
- http://axel.inria.fr/

what is Mathemagix?
- computer algebra system
- http://www.mathemagix.org/
Short history: GENOM3CK

uses from Axel
- unique algebraic tools (for visualization of implicit algebraic curves in 3D)
- easy-to-use interface
- plugins that allow extension of the system

uses from Mathemagix
- subdivision techniques (for computing singularities)
- operations on polynomials, matrices, determinants, etc.
part of Axel\textsuperscript{a};

main menu is Complex Invariant;

contains 3 types of properties:
  \begin{itemize}
  \item geometric;
  \item invariant;
  \item algebraic.
  \end{itemize}

contains computing time (at most polynomial).

Examples in the Demo!

\textsuperscript{a}Acknowledgements: Julien Wintz
1 Motivation

2 Describing the library
   ● Algorithm specifications
   ● Short history
   ● Interface

3 Testing the library
   ● Setting the input data and parameters
   ● Demo (Examples)

4 Conclusion
Setting the input data and parameters

Input data and parameters:

- \( F(x, y) \in \mathbb{C}[x, y] \) defining an input algebraic curve \( C \);

Restrictions!

- Introduce multiplication, power as \( x \ast y \) and \( x^\wedge n \).
- Introduce \( F(x, y) \) in its expanded form.

Examples in the Demo!
Setting the input data and parameters

Input data and parameters:

- \( F(x, y) \in \mathbb{C}[x, y] \) defining an input algebraic curve \( C \);
- \( \epsilon = \frac{n}{d} \in \mathbb{R}^* \) with \( n, d \in \mathbb{N}^* \);

Restrictions!

- Introduce multiplication, power as \( x \ast y \) and \( x^\wedge n \).
- Introduce \( F(x, y) \) in its expanded form.
- Introduce \( \epsilon \) by introducing \( n, d \).
- Choose \( \epsilon \) small s.t. the algorithm is correct (heuristic methods).

Examples in the Demo!
Setting the input data and parameters

Input data and parameters:

- \( F(x, y) \in \mathbb{C}[x, y] \) defining an input algebraic curve \( C \);
- \( \epsilon = \frac{n}{d} \in \mathbb{R}^{+} \) with \( n, d \in \mathbb{N}^{*} \);
- \( B = [-a, a] \times [-b, b] \times [-c, c] \in \mathbb{R}^{3}, a, b, c \in \mathbb{N}^{*} \);

Restrictions!

- Introduce multiplication, power as \( x \ast y \) and \( x^{\wedge}n \).
- Introduce \( F(x, y) \) in its expanded form.
- Introduce \( \epsilon \) by introducing \( n, d \).
- Choose \( \epsilon \) small s.t. the algorithm is correct (heuristic methods).
- Introduce \( B \) by introducing \( a, b, c \).
- Choose \( B \) big s.t. it contains all the singularities of \( C \) (heuristic methods).

Examples in the Demo!
Summary

- We have a symbolic-numeric algorithm (i.e. approximate algorithm) for plane complex algebraic curves, in the library GENOM3CK. About GENOM3CK (download, installation, documentation): http://people.ricam.oeaw.ac.at/m.hodorog/software.html

- Support: madalina.hodorog@oeaw.ac.at.

Run GENOM3CK in two ways:

- click on the icon of Axel (see output).
- run command at terminal (see intermediate computations):

  ~/pathToAxelLinux/build$ ./bin/axel
  ~/pathToAxelMacOS/src$ ./Axel.app/Contents/MacOSs/Axel
## Demo (Numeric and Symbolic Examples)

<table>
<thead>
<tr>
<th>Equation</th>
<th>Box</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x^2 - y^2, \epsilon = 1.0$</td>
<td>$[-4, 4, -6, 6, -6, 6]$</td>
</tr>
<tr>
<td>$x^2 - y^3, \epsilon = 1.0$</td>
<td>$[-4, 4, -6, 6, -6, 6]$</td>
</tr>
<tr>
<td>$x^3 - y^3, \epsilon = 1.0$</td>
<td>$[-4, 4, -6, 6, -6, 6]$</td>
</tr>
<tr>
<td>$-x^3 - 1.875xy + y^2, \epsilon = 0.25$</td>
<td>$[-4, 4, -6, 6, -6, 6]$</td>
</tr>
<tr>
<td>$1.02x^2y + 1.12y^4, \epsilon = 0.25$</td>
<td>$[-4, 4, -6, 6, -6, 6]$</td>
</tr>
</tbody>
</table>
1 Motivation

2 Describing the library
   • Algorithm specifications
   • Short history
   • Interface

3 Testing the library
   • Setting the input data and parameters
   • Demo (Examples)

4 Conclusion
Conclusion and future work

DONE:
- automatization of an approximate algorithm for complex curves in GENOM3CK;

TO DO’s:
- prove the properties of the approximate algorithm (i.e. convergency, continuity);
- deform the input curve and keep the invariants constant.
- include other operations, i.e. from knot theory, algebraic geometry.
Conclusion and future work

DONE:
- automatization of an approximate algorithm for complex curves in GENOM3CK;

TO DO’s:
- prove the properties of the approximate algorithm (i.e. convergency, continuity);
- deform the input curve and keep the invariants constant.
Conclusion and future work

**DONE:**
- automatization of an approximate algorithm for complex curves in GENOM3CK;
- describe algorithm with principles from regularization theory;

**TO DO’s:**
- prove the properties of the approximate algorithm (i.e. convergency, continuity);
- deform the input curve and keep the invariants constant.
Conclusion and future work

☑️ **DONE:**

- automatization of an approximate algorithm for complex curves in GENOM3CK;
- describe algorithm with principles from regularization theory;
- integrate symbolic, numeric, graphical capabilities into a single library GENOM3CK (use of Axel);
- provide a natural graphical user interface (use of QSA);
- users can visualize the ongoing computations and the results;

✗ **TO DO’s:**

- prove the properties of the approximate algorithm (i.e. convergency, continuity);
- deform the input curve and keep the invariants constant.
Conclusion and future work

✓ DONE:

- automatization of an approximate algorithm for complex curves in GENOM3CK;
- describe algorithm with principles from regularization theory;
- integrate symbolic, numeric, graphical capabilities into a single library GENOM3CK (use of Axel);
- provide a natural graphical user interface (use of QSA);
- users can visualize the ongoing computations and the results;

✗ TO DO’s:

- prove the properties of the approximate algorithm (i.e. convergency, continuity);
- deform the input curve and keep the invariants constant.
- include other operations, i.e. from knot theory, algebraic geometry.
“...in programming mathematical elegance is not a dispensable luxury but a matter of life and death” (E.W. Dijkstra, 1978).

Thank you for your attention.