HAL
open science

PTOPO: A Maple package for the topology of parametric curves

Christina Katsamaki, Fabrice Rouillier, Elias Tsigaridas, Zafeirakis Zafeirakopoulos

To cite this version:

Christina Katsamaki, Fabrice Rouillier, Elias Tsigaridas, Zafeirakis Zafeirakopoulos. PTOPO: A Maple package for the topology of parametric curves. ACM Communications in Computer Algebra, 2020, 54 (2), pp.49-52. 10.1145/3427218.3427223 . hal-02953909

HAL Id: hal-02953909

https://inria.hal.science/hal-02953909v1

Submitted on 1 Oct 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PTOPO: A Maple package for the topology of parametric curves

Christina Katsamaki *
Inria Paris, IMJ-PRG,
Sorbonne Université and Paris Université
F-75005, Paris, France
christina.katsamaki@inria.fr

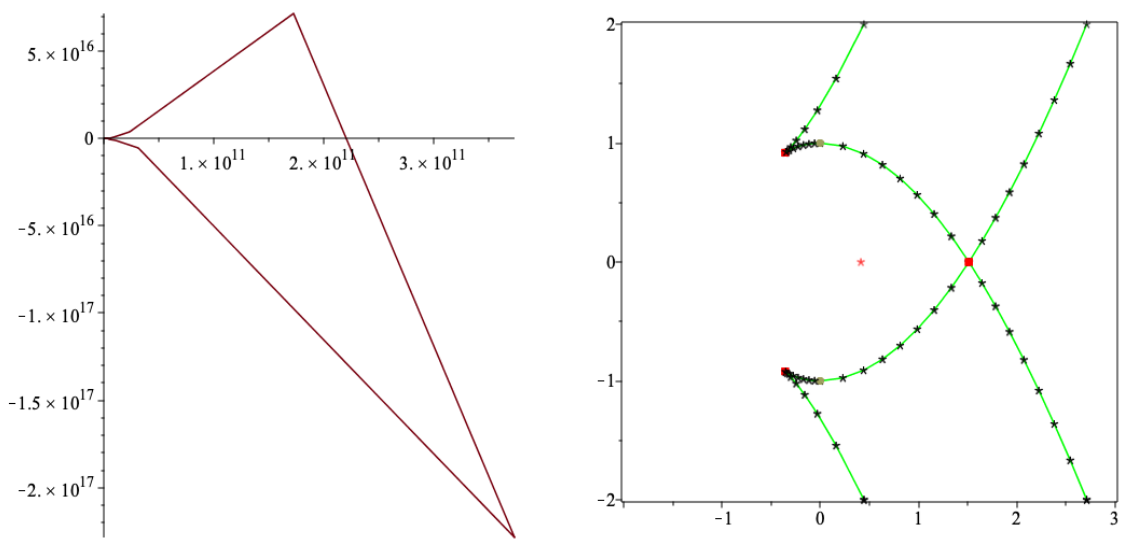
Fabrice Rouillier
Inria Paris, IMJ-PRG,
Sorbonne Université and Paris Université
F-75005, Paris, France
Fabrice.Rouillier@inria.fr

Zafeirakis Zafeirakopoulos
Institute of Information Technologies
Gebze Technical University
Kocaeli, Turkey
zafeirakopoulos@gtu.edu.tr

Elias Tsigaridas
Inria Paris, IMJ-PRG,
Sorbonne Université and Paris Université
F-75005, Paris, France
elias.tsigaridas@inria.fr

Abstract

PTOPO is a MAPLE package computing the topology and describing the geometry of a parametric plane curve. The algorithm behind PTOPO constructs an abstract graph that is isotopic to the curve. PTOPO exploits the benefits of the parametric representation and performs all computations in the parameter space using exact computing. PTOPO computes the topology and visualizes the curve in less than a second.



Comparison of MAPLE parametric plot vs PTOPO

*Supported by the Fondation Sciences Mathématiques de Paris (FSMP)

1 Topology of Parametric Curves

The study of parametric curves is a classical topic in computational algebra and geometry Sendra and Winkler (1999); Boissonnat and Teillaud (2006); Busé et al. (2019). The interest for computing with parametric curves has been motivated, among others, by the omnipresence of parametric representations in computer modeling and computer aided geometric design, e.g., Farouki et al. (2010).

PTOPO allows the computation of the topology of a real parametric plane curve in MAPLE. In particular, it computes an abstract graph that is isotopic (Boissonnat and Teillaud, 2006, p. 184) to the curve in the embedding space.

Let $\widetilde{\mathcal{C}}$ be an algebraic curve over $\mathbb{C}^2$, parametrized by the map

$$\begin{aligned} \phi: \quad & \mathbb{C} \dashrightarrow \widetilde{\mathcal{C}} \\ & t \mapsto \big(\phi_1(t), \phi_2(t)\big) = \Big(\frac{p_1(t)}{q_1(t)}, \frac{p_2(t)}{q_2(t)}\Big), \end{aligned} \tag{1}$$

where $p_i, q_i \in \mathbb{Z}[t]$ are of size (d, τ) for $i = 1, 2$, and $\widetilde{\mathcal{C}}$ is the Zariski closure of $\mathrm{Im}(\phi)$. We call $\phi(t)$ a *parametrization* of $\widetilde{\mathcal{C}}$. We study the real trace of $\tilde{\mathcal{C}}$, that is $\mathcal{C} := \tilde{\mathcal{C}} \cap \mathbb{R}^2$.

The theoretical background of the PTOPO package, together with a detailed complexity analysis, is given in Katsamaki et al. (2020).

PTOPO computes all points necessary for representing the geometry of the curve, as well as for producing a certified visualization of plane curves. We call them special points and these are the cusps, the multiple, the extreme and the isolated points. On top of that, PTOPO constructs a planar graph whose vertices correspond to points on the curve connected accordingly. The drawing of the graph offers a visualization of the curve in an $\mathbb{R}^2$-area that contains all its special points.

The computation of the parameters of special points is done by solving over $\mathbb{R}$ a bivariate system of polynomials and univariate real solving. Using SLV **?** and RootFinding[Isolate], we have a certified implementation of general purpose exact computations with one and two real algebraic numbers, like comparison and sign evaluations. For a parametrization ϕ, we consider the following system of bivariate polynomials:

$$h_i(s,t) = \frac{p_i(s)q_i(t) - q_i(s)p_i(t)}{s-t}, \qquad \text{for } i = 1, 2. \tag{2}$$

We partition the roots of the bivariate system according to the kind of special point they correspond to, denoting the sets by $\mathtt{T}_C$ (cusps), $\mathtt{T}_M$ (multiple points), $\mathtt{T}_E$ (extreme points), $\mathtt{T}_I$ (isolated points) and $\mathtt{T}_P^{\mathbb{R}}$ (real poles). Then, we compute the set of all intersections of the curve with a box containing all special points, denoted by $\mathtt{T}_B$. Within this box, called characteristic box, the topology of the curve is fully described.

Let T be the union of the above sets of parameters. Add a vertex v_i for every distinct point corresponding to a parameter in T and let $\lambda(v_i)$ be its label set (containing the parameter values giving the point). If for two consecutive elements $t_1 < t_2$ in $\mathtt{T}$ there exists a pole $s \in \mathtt{T}_P^{\mathbb{R}}$ such that $t_1 < s < t_2$, then we split $\mathtt{T}$ into two lists: $\mathtt{T}_1$ containing the elements $\leq t_1$ and $\mathtt{T}_2$ containing the elements $\geq t_2$. Continue recursively for $\mathtt{T}_1$ and $\mathtt{T}_2$, until there are no poles between any two elements of the resulting list. This procedure partitions $\mathtt{T}$ into $\mathtt{T}_1, \ldots, \mathtt{T}_\ell$.

Consider each $\mathtt{T}_i$ with more than one element separately. For any two consecutive elements $t_1 < t_2$ in $\mathtt{T}_i$, with $t_1 \in \lambda(v_{i,1})$ and $t_2 \in \lambda(v_{i,2})$, we add the edge $\{v_{i,1}, v_{i,2}\}$. If $\mathbf{p}_\infty$ exists, we add an edge to the graph connecting the vertices corresponding to the last element of $\mathtt{T}_\ell$ and the first element of the $\mathtt{T}_1$.

For plane curves, if $N = \max\{d, \tau\}$, the complexity of computing the topology becomes $\widetilde{\mathcal{O}}_B(N^6)$. However, visualizing the curve on top of the abstract graph construction, increases the bound to $\widetilde{\mathcal{O}}_B(N^7)$. We tested PTOPO for many parametric curves from the literature. In practice, for all examples, PTOPO computes the topology in less than a second.

2 Using PTOPO

The PTOPO package can be obtained from `https://webusers.imj-prg.fr/~christina.katsamaki/ptopo/`. PTOPO requires the SLV package.

```
> restart;
LIBPATH := "...the path to ptopo...":
read( cat(LIBPATH, "SLV.mpl"));
read( cat(LIBPATH, "ptopo.mpl"));
> with(PTOPO);
```

The easiest way to see the topology of a curve using PTOPO is to call the draw function with arguments the parametrization of the curve $\phi = \left(\frac{p_1}{q_1}, \frac{p_2}{q_2}\right)$.

The draw function can be called in two ways. It can take as arguments a list of numerators and a list of denominators.

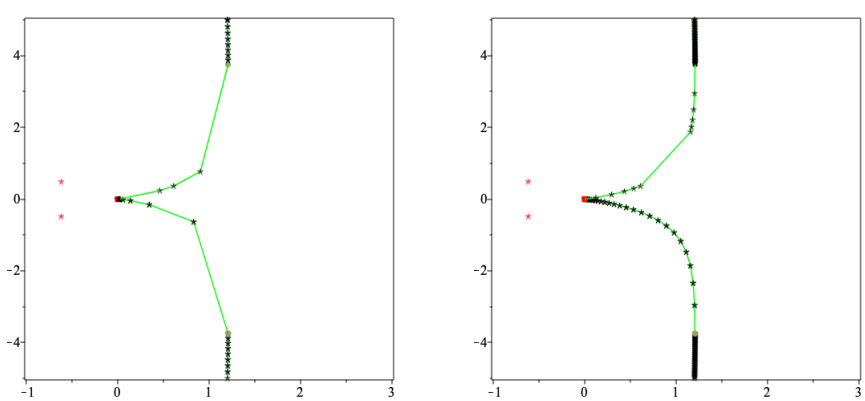
```
> p := [ t^2 + 1, 1];
q := [t^4 + 1, t^3];
PTOPO:-draw(p, q);
```

Alternatively, the argument can be a list of rational functions.

```
> phi := normal([ seq(p[i]/q[i], i=1..2)]);
PTOPO:-draw(phi);
```

It is possible to use different variable names, the defaults are 't' and 's'. The number of points to be drawn within the smooth arcs is controlled by the last argument. If the argument is n ,then $2n+1$ points are computed. Increasing n increases the computational cost, but improves the drawing. Note that it does not affect the topological correctness of the drawing.

```
> PTOPO:-draw(p, q,'t','s',5);
```



Although the draw command is enough, if a picture of the topology is needed, PTOPO provides commands for obtaining the topologically interesting points separately. In that case, we initialize by giving a parametrization, and just as like for the draw command, the parametrization can be given as a pair of lists, the first containing the numerators and the second the denominators, or as a list of rational functions.

```
> p := [ t^2 + 1, 1];
q := [t^4 + 1, t^3];
PTOPO:-parametrization(p, q);
```

```
> phi := [(3*t^4+ 4*t^3 + 32*t^2 + 28*t + 99)/(t^2+t+7)*(t^2 + 1),
  ((t^2 + t + 7)^3)/((t+6)*(t^2 +1)^2)];
PTOPO:-parametrization(phi);
```

```
    There is a point at infinity. We reparametrize....

 ----****---- The parametrization is  proper  ----****----
...
```

To compute the topology of the given parametric curve, we call

```
> PTOPO:-topology();
```

After the topology is computed we can see a summary of the points of interest.

```
> PTOPO:-point_summary();
```

```
* Isolated points:
 : [  -0.618034,  -0.485868]    i
 : [  -0.618034,  +0.485868]    i
* Point at infinity:
 : [  +0.020816,  +0.002915]
* Poles:
 :  t:  -0.142857   p
* Extreme points:
 : [  +1.207107,  +3.751142],   t:  -0.866081   e
 : [  +1.207107,  -3.751144],   t:  +0.458575   e
* Boundary points:
 : [  +1.201470,  +5.000000],   t:  -0.793994   b
 : [  +1.201470,  -5.000000],   t:  +0.407871   b
* Cusps:
 : [  +0.000000,  +0.000000],   t:  +7.000000   c
* Double points:
```

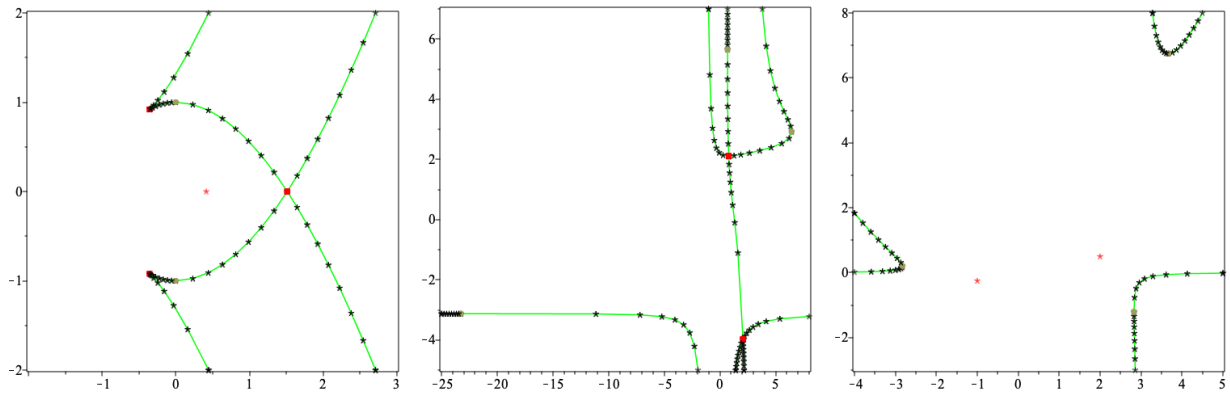
The output contains points in 2D (except for poles) and the corresponding parameter values (when applicable). The labels are **i** for isolated points, **p** for poles, **e** for extreme points, **d** for double points, **c** for cusps, **b** for boundary points.

Alternatively, can get separately isolated points, point at infinity, poles, extreme points, double points and cusps. Moreover, we can see points intersecting the boundary of the box containing everything topologically interesting.

3 Conclusion

The PTOPO package is practical and easy to use. It computes the topology of a curve and can also visualize it. It is fast enough to make its use practical for most applications.

Since the algorithmic foundations for plane and space curves are not different, we plan to extend the implementation to handle space curves in the future.



References

Juan Gerardo Alcázar and Gema María Díaz-Toca. 2010. Topology of 2D and 3D rational curves. *CAGD* 27, 7 (2010), 483 – 502. https://doi.org/10.1016/j.cagd.2010.07.001

Jean-Daniel Boissonnat and Monique Teillaud (Eds.). 2006. *Effective Computational Geometry for Curves and Surfaces*. Springer-Verlag, Mathematics and Visualization.

Laurent Busé, Clément Laroche, and Fatmanur Yıldırım. 2019. Implicitizing rational curves by the method of moving quadrics. *Computer-Aided Design* 114 (2019), 101–111.

Dimitris I. Diochnos, Ioannis Z. Emiris, and Elias P. Tsigaridas. 2009. On the asymptotic and practical complexity of solving bivariate systems over the reals. *J. Symb. Comput.* 44, 7 (2009), 818–835. (Special issue on ISSAC 2007).

Rida T. Farouki, Carlotta Giannelli, and Alessandra Sestini. 2010. Geometric Design Using Space Curves with Rational Rotation-Minimizing Frames. In *Mathematical Methods for Curves and Surfaces*, Morten Dæhlen, Michael Floater, Tom Lyche, Jean-Louis Merrien, Knut Mørken, and Larry L. Schumaker (Eds.). Springer, 194–208.

Christina Katsamaki, Fabrice Rouillier, Elias Tsigaridas, and Zafeirakis Zafeirakopoulos. 2020. On the geometry and the topology of parametric curves. (May 2020). https://hal.archives-ouvertes.fr/hal-02573423

Dinesh Manocha and John F. Canny. 1992. Detecting cusps and inflection points in curves. *CAGD* 9, 1 (1992), 1 – 24. https://doi.org/10.1016/0167-8396(92)90050-Y

Sonia Pérez-Díaz. 2006. On the problem of proper reparametrization for rational curves and surfaces. *CAGD* 23, 4 (2006), 307–323.

R. Rubio, J.M. Serradilla, and M.P. Vélez. 2009. Detecting real singularities of a space curve from a real rational parametrization. *J. Symb. Comput.* 44, 5 (2009), 490 – 498. https://doi.org/10.1016/j.jsc.2007.09.002

J. Rafael Sendra and Franz Winkler. 1999. Algorithms for Rational Real Algebraic Curves. *Fundam. Inf.* 39, 1,2 (April 1999), 211–228. http://dl.acm.org/citation.cfm?id=2378083.2378093

J Rafael Sendra, Franz Winkler, and Sonia Pérez-Díaz. 2008. Rational algebraic curves. *Algorithms and Computation in Mathematics* 22 (2008).