Automatic calculation of plane loci using Groebner bases and integration into a Dynamic Geometry System

Michael Gerhäuser, Alfred Wassermann

July 24, 2010

UNIVERSITÄT BAYREUTH

Overview

JSXGraph - A short overview

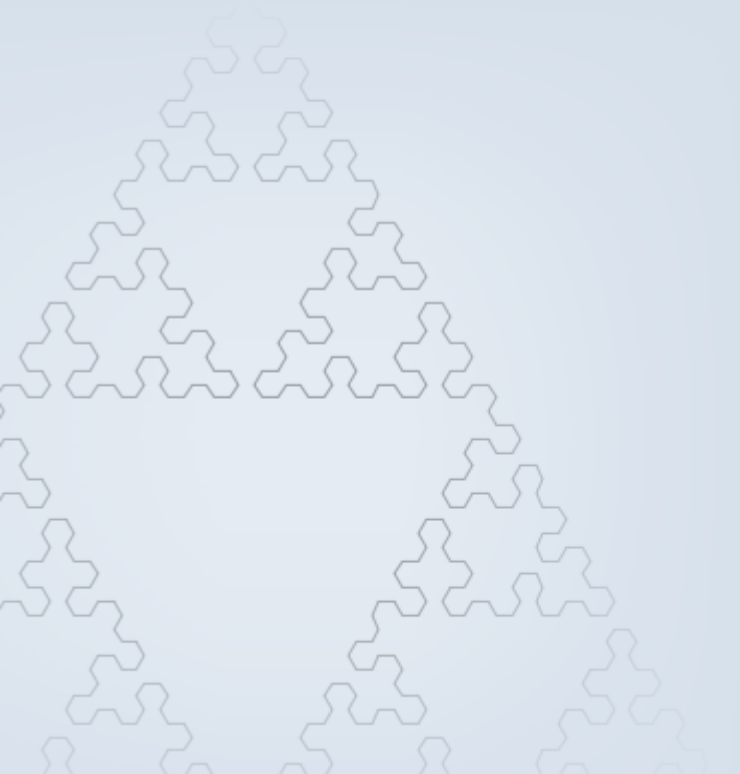
Computing plane loci using Groebner bases

Implementing this algorithm in JSXGraph

Optimizations

Examples

JSXGraph



UNIVERSITÄT BAYREUTH

JSXGraph

What is JSXGraph?

- A library implemented in JavaScript
- Runs in recent versions of all major browsers
- No plugins required
- LGPL-Licensed

Main features

- Dynamic Geometry
- Interactive function plotting
- Turtle Graphics
- Charts

Supported Hardware

- PC (Windows, Linux, Mac)
- Mobile phones
- "Touchpads" like the Apple iPod and iPad
- Basically everything which runs at least one of the supported browsers

Supported Browsers

- Firefox
- Chrome/Chromium
- Safari
- Internet Explorer
- Opera

JSXGraph

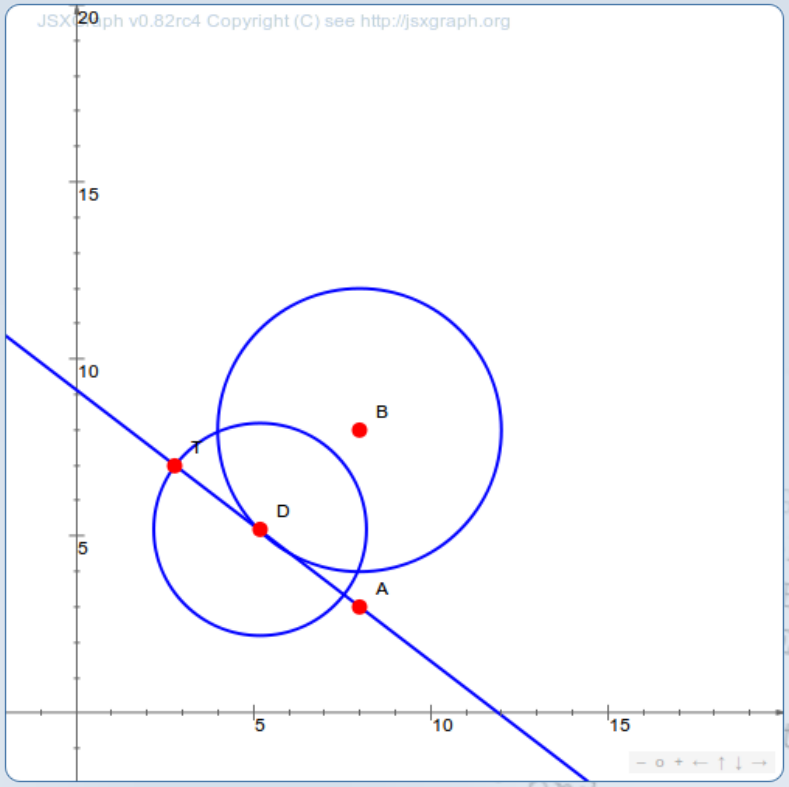
Example/Input

```
<link rel="stylesheet" type="text/css" href="css/jsxgraph.css" />
<script type="text/javascript" src="js/jsxgraphcore.js"></script>
[...]
<div id="jxgbox" class="jxgbox" style="width:500px; height:500px;"></div>
<script type="text/javascript">
  /* <![CDATA[ */
  board = JXG.JSXGraph.initBoard('jxgbox', {boundingbox: [-2, 20, 20, -2], axis:
        true, grid: false, keepaspectratio: true, showcopyright: false});
  p3 = board.create('point', [8, 3]);
  p4 = board.create('point', [8, 8]);
  c1 = board.create('circle', [p4, 4]);
  p6 = board.create('glider', [0, 0, c1], {name: 'D'});
  g = board.create('line', [p3, p6]);
  c2 = board.create('circle', [p6, 3]);
  p14_1 = board.create('intersection', [c2,g,0], {name: 'T'});
  /* ]]> */
</script>
```

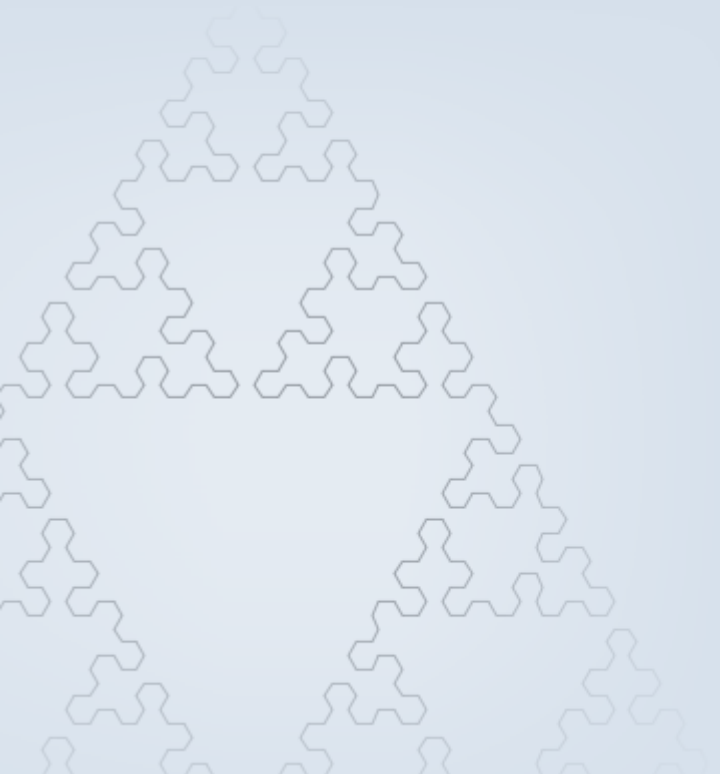
UNIVERSITÄT BAYREUTH

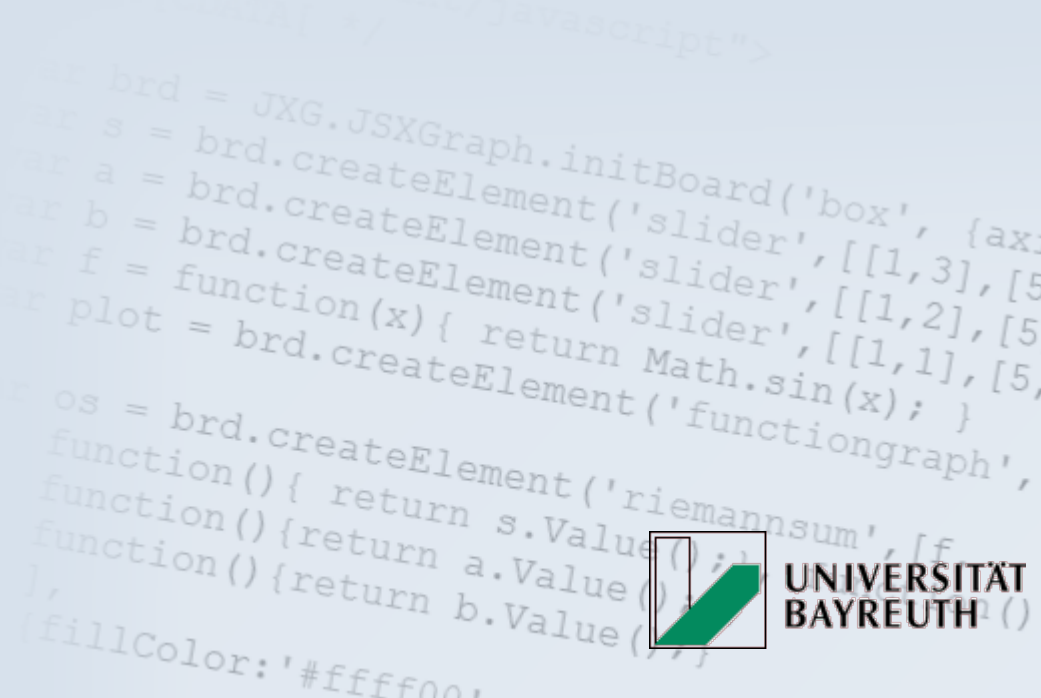
JSXGraph

Example/Output



Computing plane loci using Groebner bases (in a nutshell)

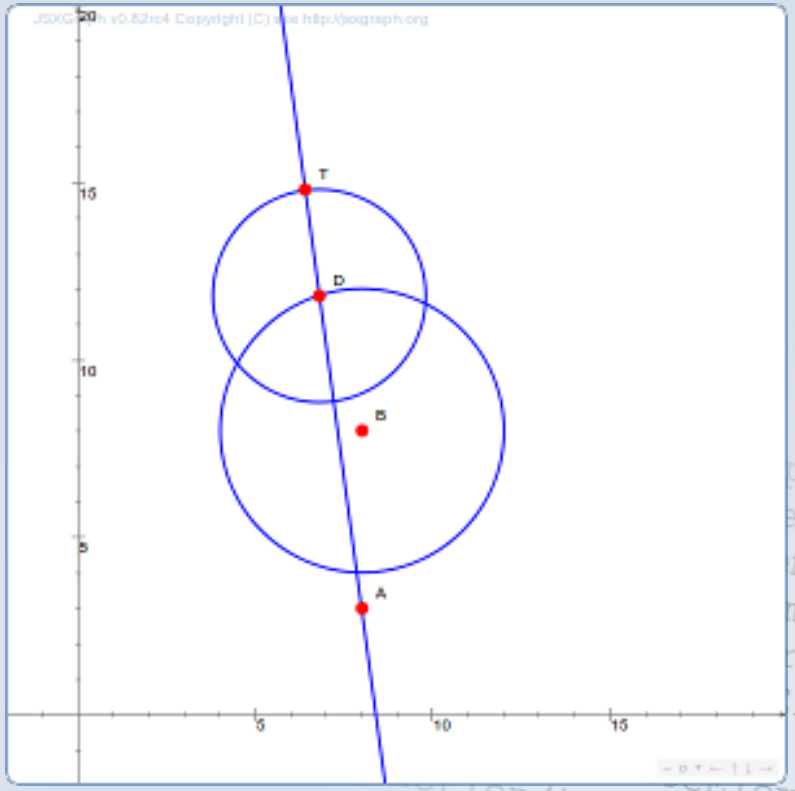




UNIVERSITÄT BAYREUTH

Computing plane loci using Groebner bases

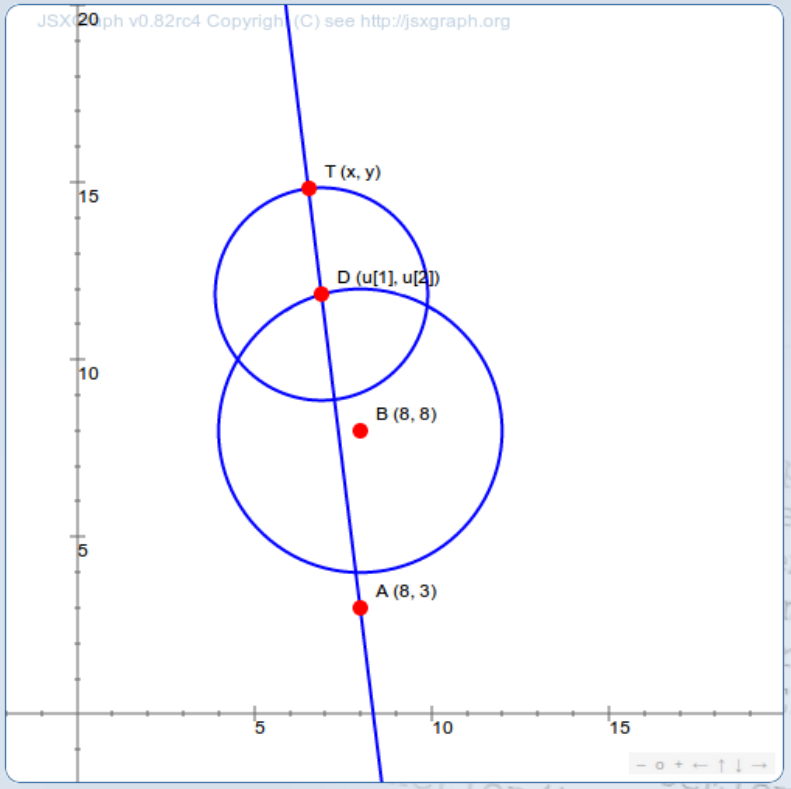
- Given a set of free and dependent points,



UNIVERSITÄT BAYREUTH

Computing plane loci using Groebner bases

- we first choose a coordinate system,



UNIVERSITÄT BAYREUTH

Computing plane loci using Groebner bases

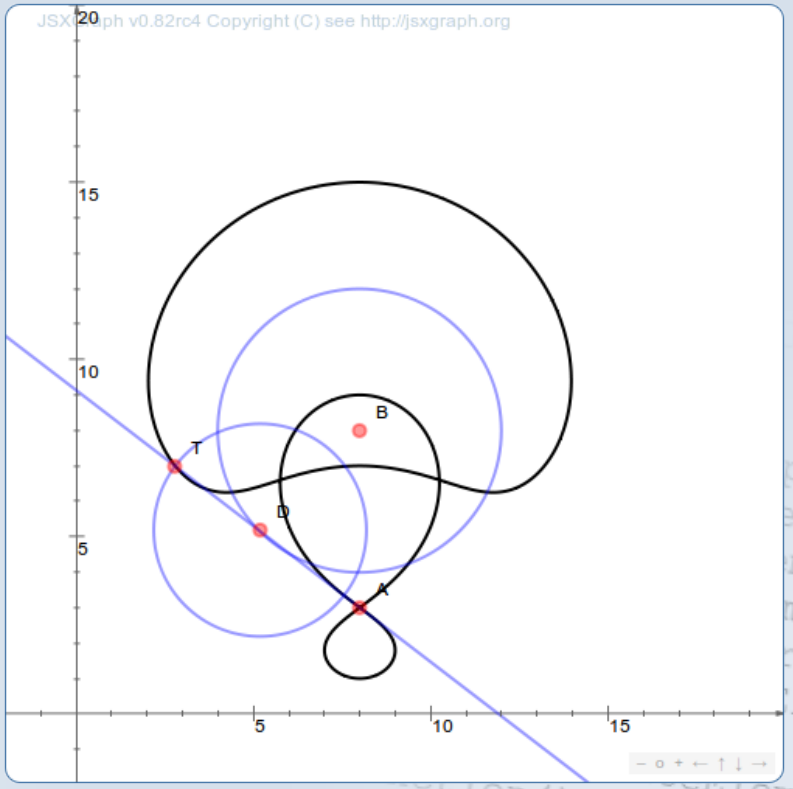
- translate geometric constraints into an algebraic form,
  - $(u[1]-8)^2+(u[2]-8)^2-16=0$
  - $(x-u[1])^2+(y-u[2])^2-9=0$
  - $3x-3u[1]+yu[1]-8y+8u[2]-xu[2]=0$

Computing plane loci using Groebner bases

- calculate the Gröbner basis of the given ideal,
  - $x^6 + 3x^4y^2 + 3x^2y^4 + y^6 - 48x^5 - 38x^4y - 96x^3y^2 - 76x^2y^3 - 48xy^4 - 38y^5 + 1047x^4 + 1216x^3y + 1774x^2y^2 + 1216xy^3 + 727y^4 - 13024x^3 - 16596x^2y - 16096xy^2 - 8404y^3 + 97395x^2 + 109888xy + 63535y^2 - 415536x - 300806y + 790009 = 0$

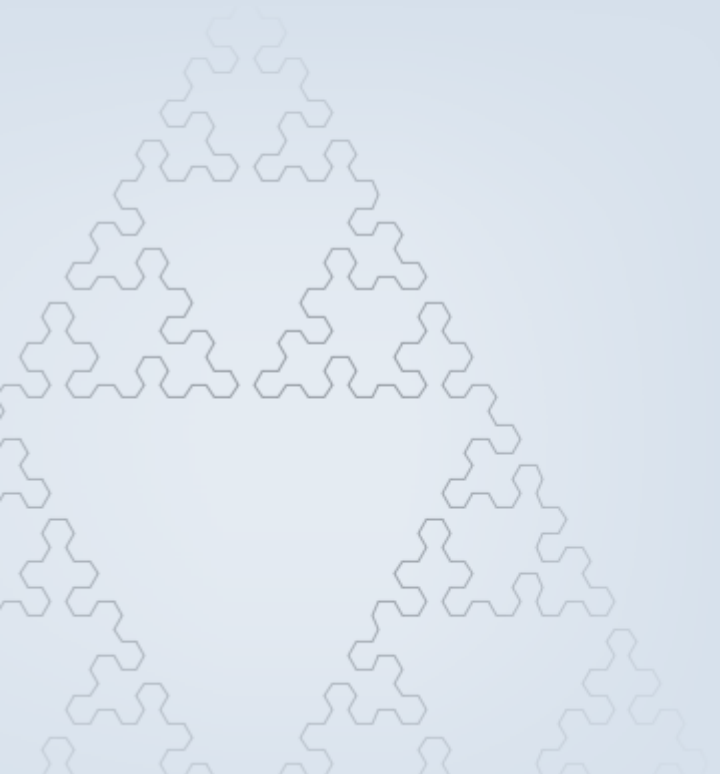
Computing plane loci using Groebner bases

- and finally plot the calculated implicit equation.



UNIVERSITÄT BAYREUTH

Implementing this algorithm in JSXGraph



Implementing this algorithm in JSXGraph

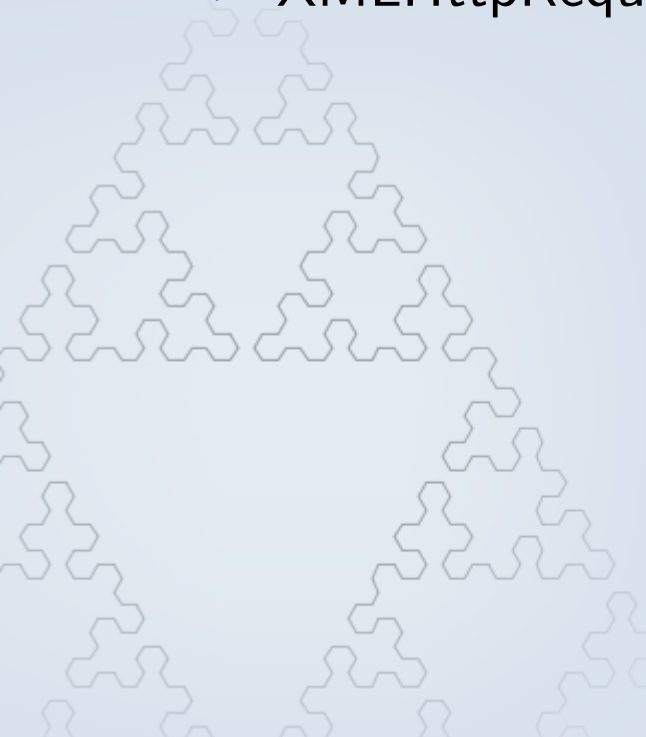
Problems

- No JavaScript implementation of any Gröbner basis algorithm
- Can't use C-libraries directly in JavaScript
- No implicit plotting in JSXGraph by now

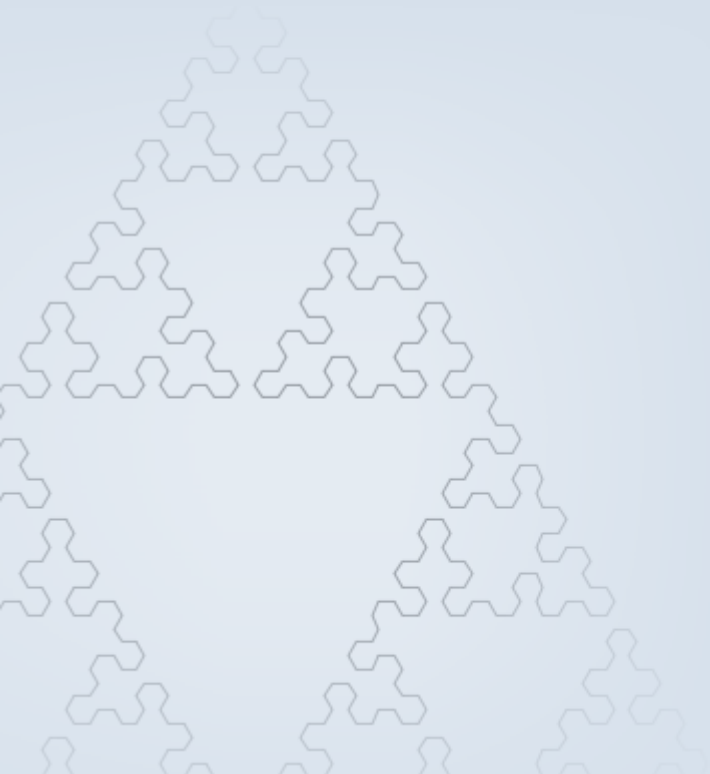
Implementing this algorithm in JSXGraph

Our solution

- XMLHttpRequest/AJAX to



Optimizations



Examples