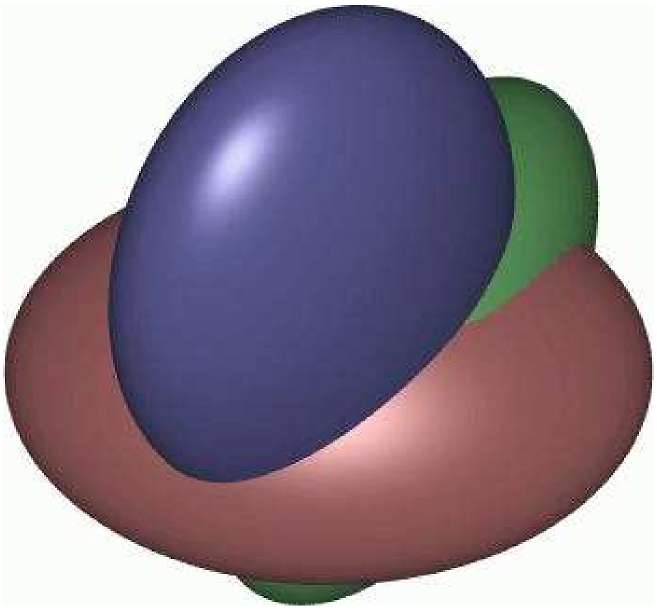
Computing a 3–dimensional Cell in an Arrangement of Quadrics: Exactly and Actually!



Michael Hemmer
hemmer@mpi-sb.mpg.de

Elmar Schömer
schoemer@mpi-sb.mpg.de

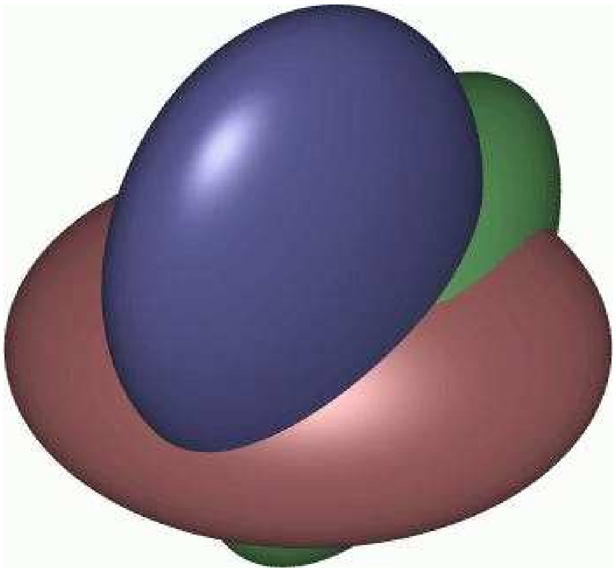
Nicola Wolpert
nicola@cs.uni-sb.de

Motivation

The world is not linear!

We want to compute with curved surfaces directly.

Why are curved surfaces unwieldy?



The intersection points of surfaces defined by *rational* polynomials in general have *irrational* coordinates.

$\Longrightarrow$ we have to deal with algebraic numbers

Computing with algebraic numbers:

- floating point arithmetic $\rightarrow$ fast, inexact
  $\rightarrow$ **incorrect**
- exact algebraic computation $\rightarrow$ exact, very slow

Problem: speed versus exactness

Our aim: Topological description of a cell in an arrangement of quadrics

- exact computation in *all* cases
- implementable
- acceptable running time

Previous work

G. E. Collins; *Cylindrical algebraic decomposition* (1975)

E. Chionh, R. Goldman, J. Miller; *The intersection of three quadric surfaces* (1991)

T. Sakkalis; *The topological configuration of a real algebraic curve* (1991)

J.F. Canny; *Gap theorem* (1987)

P. Pedersen; *Multivariate Sturm sequences* (1991)

and others ...

Geometric and Algebraic Libraries:

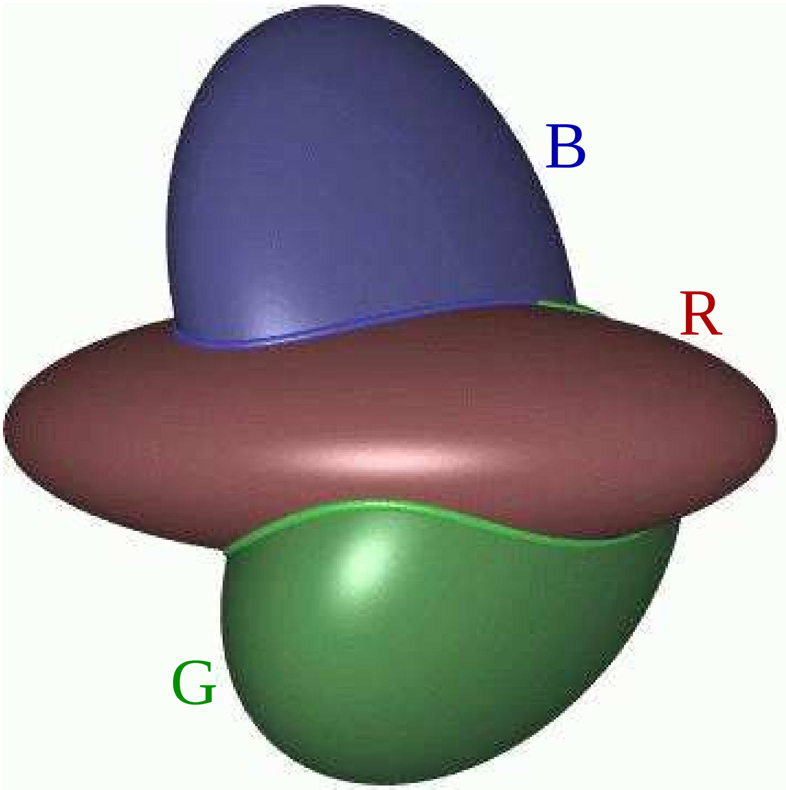
K. Mehlhorn et al; *LEDA*

CGAL

A. Rege; *APU* (1996)

J. Keyser et al; *MAPC* (1999)

Reduction to planar arrangements

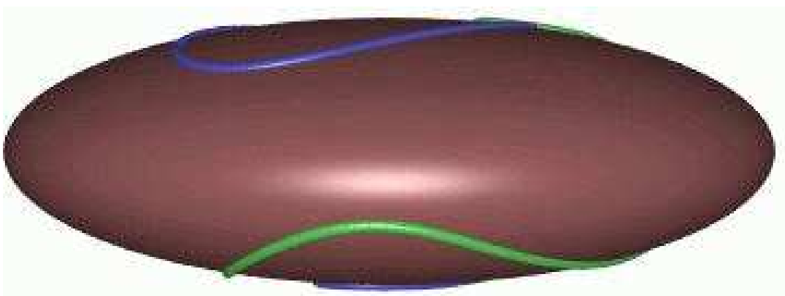


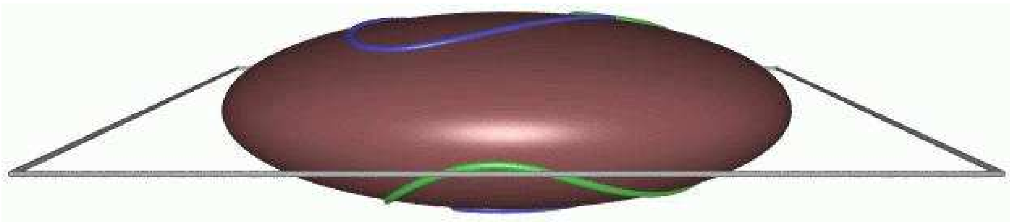
Compute a cell in the arrangement of n quadrics.

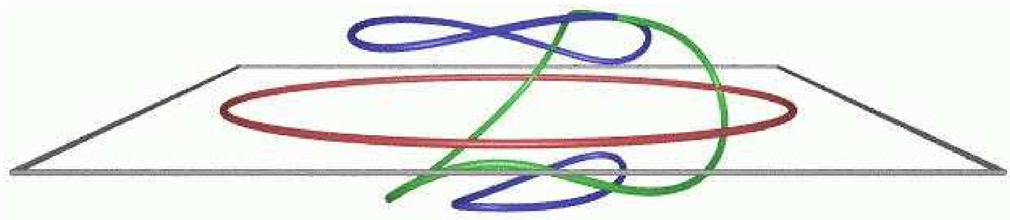
Representation of a quadric:

$$27x^2 + 62y^2 + 249z^2 - 10$$

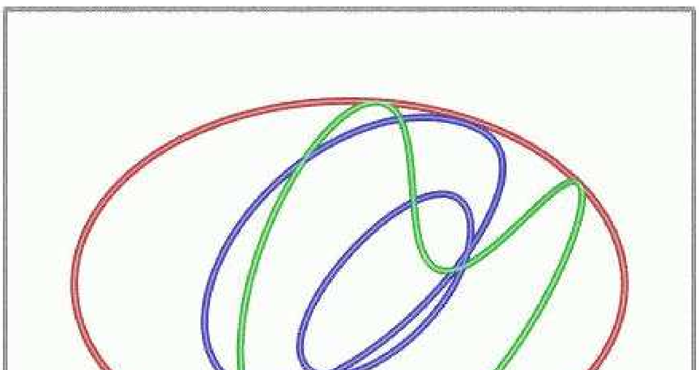
For each quadric compute sub–arrangement on its surface.







Project the intersection curves and the silhouette of R into the (x, y)–plane.

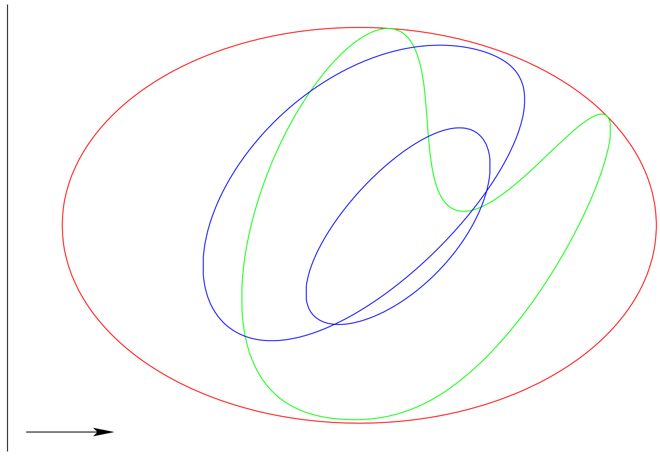


Representation of a planar curve:

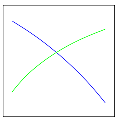
$$408332484x^4 + 51939673y^4 - 664779204x^3y - 24101506y^3x$$
$$+564185724x^2y^2 - 250019406x^3 + 17767644y^3$$
$$+221120964x^2y - 123026916y^2x + 16691919x^2 + 4764152y^2$$
$$+14441004xy + 10482900x + 2305740y - 1763465$$

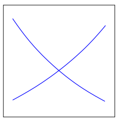
Computing planar arrangements

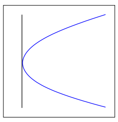
Sweep–line algorithm:



Event points the sweep line has to stop at:

 Intersections of two curves

 Singular points

 Extreme points (points with a vertical tangent)

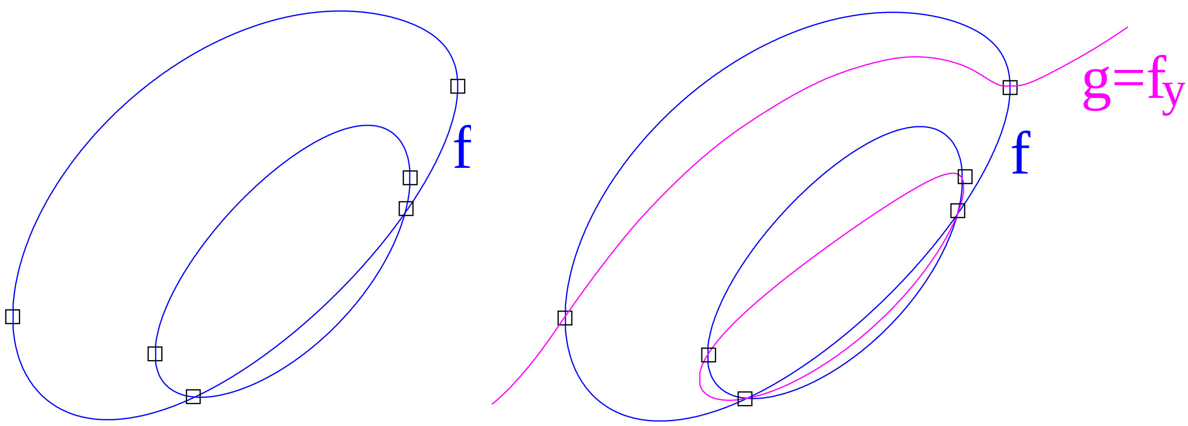
We reduce everything to: Event point = intersection point of two curves f and g

We identify f with its zero set:

$$f := \{(x, y) \mid f(x, y) = 0\}$$

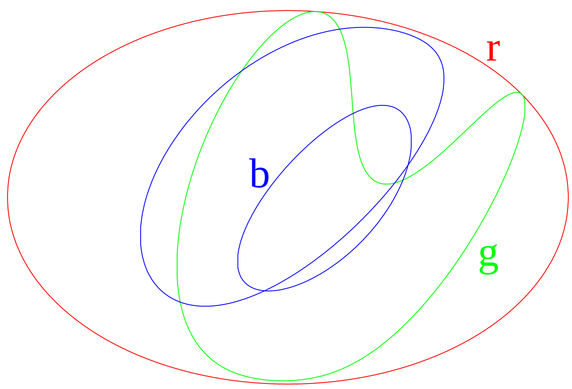
We reduce everything to: Event point $=$ intersection point of two curves f and g

- Intersection of two curves f and g $\surd$
- Singular and extreme points of a curve f $=$ intersection points of f and $g = f_y = \partial f / \partial y$



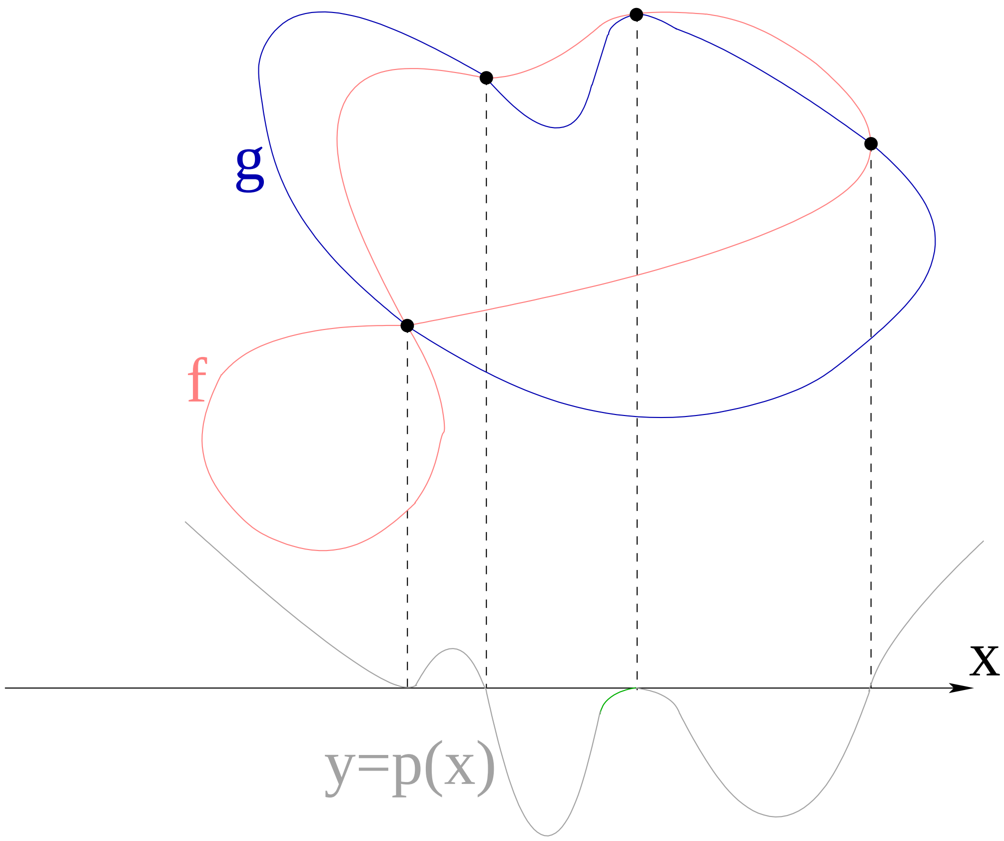
Event points: all intersections between

- r and b
- r and g
- b and g
- r and r_y
- b and b_y
- g and g_y



Determine the intersection points between 2 curves !

Determine the intersection points between 2 curves !



Resultant computation: $p(x) \in \mathbb{Q}[x]$

x–coordinate of an intersection point between two curves f and g

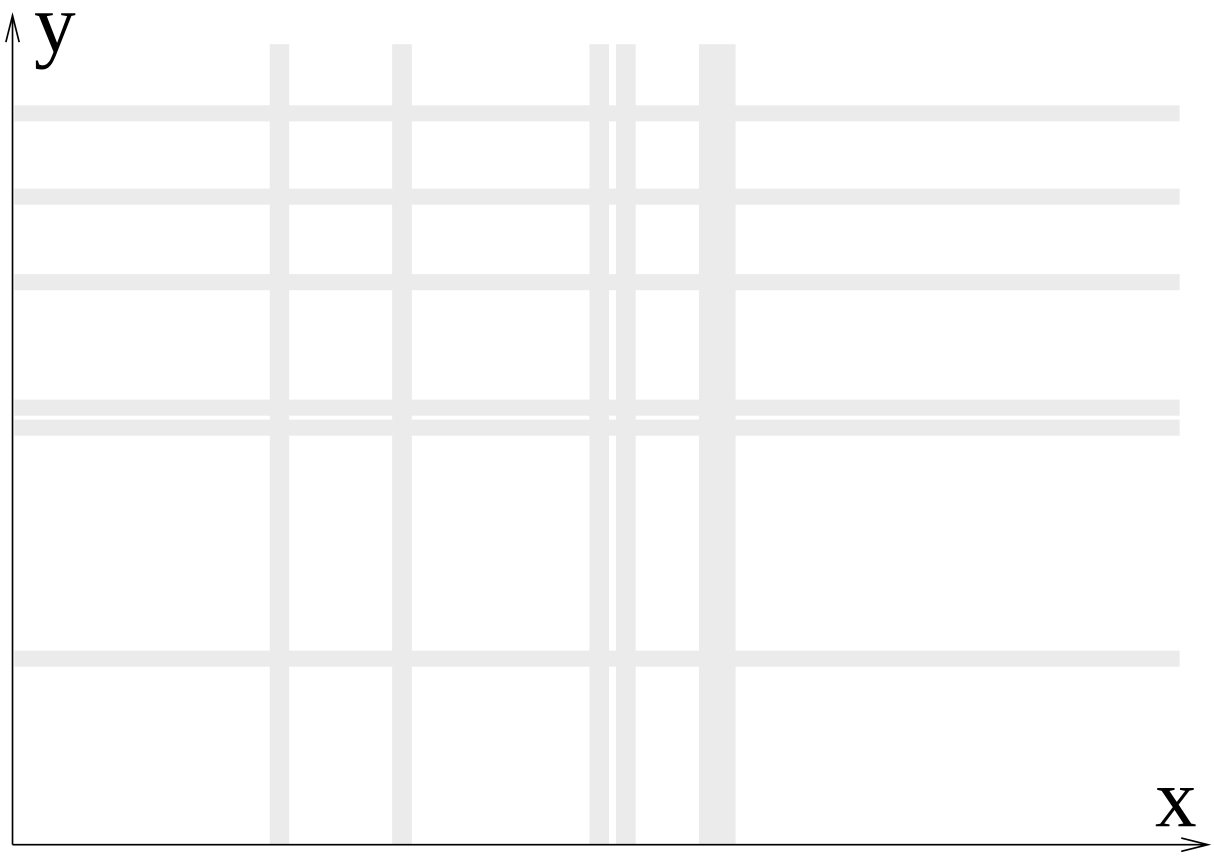
$\in$ roots of $p(x)$ (p can have extraneous roots)

multiplicity of a root of p indicates the multiplicity of the according intersection point between f and g

$\deg(p) \leq \deg(f) \cdot \deg(g)$

Intersection points of f and g: irrational coordinates

- Compute narrow *rational* stripes parallel to x–axis containing all real intersection points (resultant + root separation); stripes can be empty
- Do the same parallel to the y–axis.
- Intersection of the stripes yields boxes with *rational* corners.
  Real intersection points are contained in the boxes.
  Test each box for an intersection point.

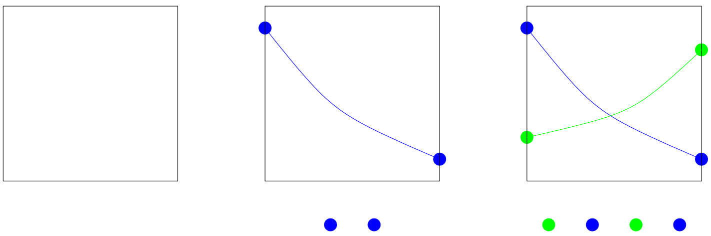


Question: Which box contains an intersection point?

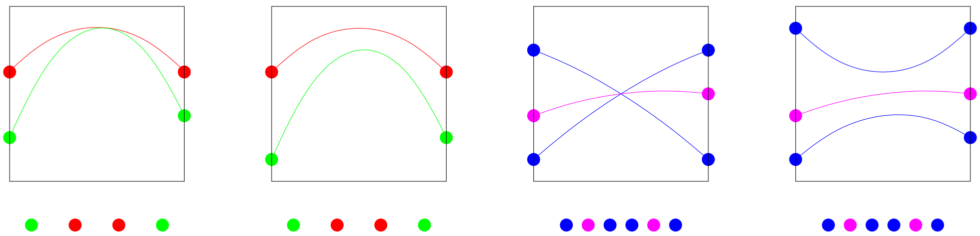
Question: Which box contains an intersection point?

Information: Sequence of hits of the curves with the boundary of the box (root isolation)

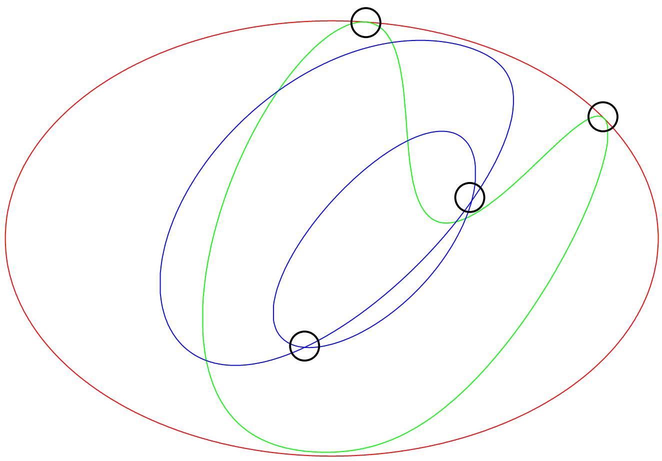
A first answer: Simple box-hit counting



But



Tangential intersections and singular points appear quite often in our arrangements



$\Longrightarrow$ using gap theorems (Canny, 87) or multivariate Sturm calculation (Pedersen, 91) too expensive

A second answer: Explicit solutions

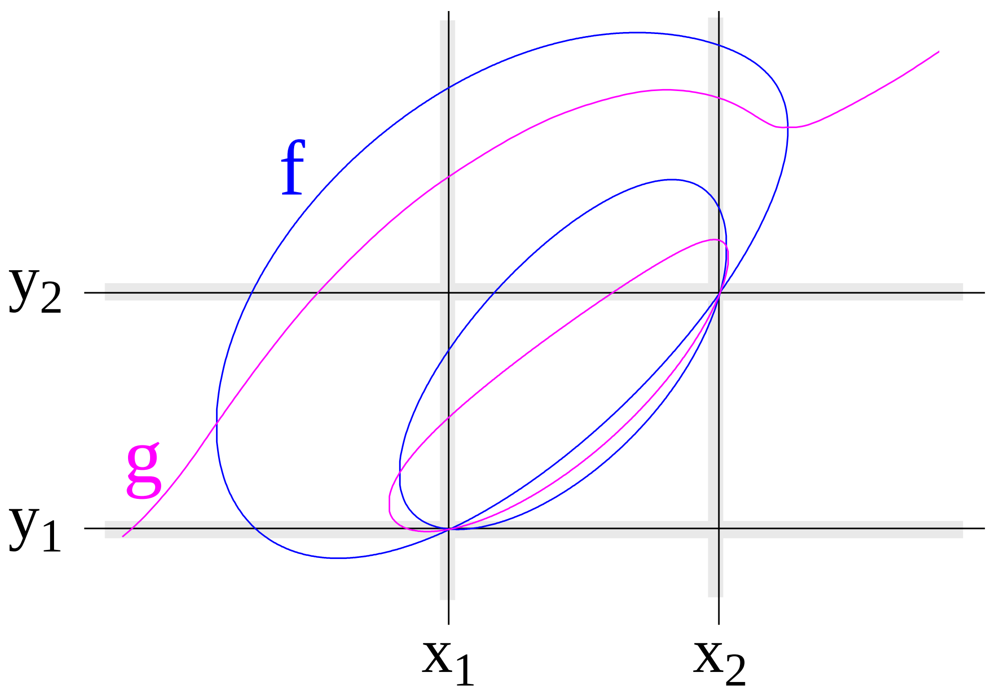
Intersection points are defined by quadratic polynomials:

$$\begin{aligned} u(x) &= ax^2 + bx + c \\ v(y) &= \alpha y^2 + \beta y + \gamma \end{aligned}$$

Roots of $u(x)$ and $v(y)$:

$x_{1,2} = 1/(2a) \cdot (b \ -/+ \ \sqrt{b(b-2a)})$
$y_{1,2} = 1/(2\alpha) \cdot (\beta \ -/+ \ \sqrt{\beta(\beta-2\alpha)})$

$\Rightarrow$ Explicit terms involving only one square–root



Test: $f(x_i, y_j) = 0 = g(x_i, y_j)$ for $i, j = 1, 2$

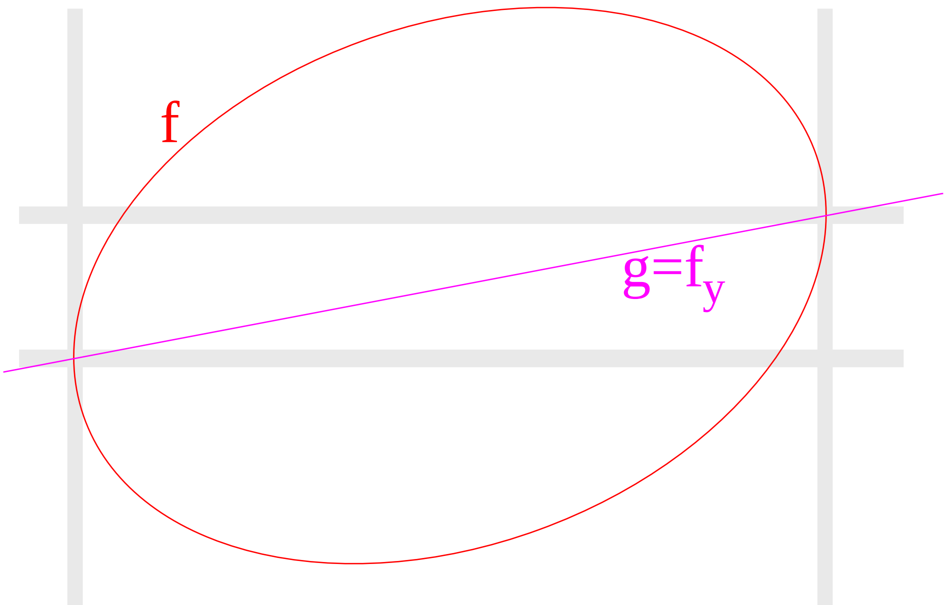
Main Theorem

Each event point in a planar arrangement that results from the projection of 3d quadrics can be determined by box hit counting or explicitly.

Proof

case 1:

f projection of the quadric and $g = f_y$

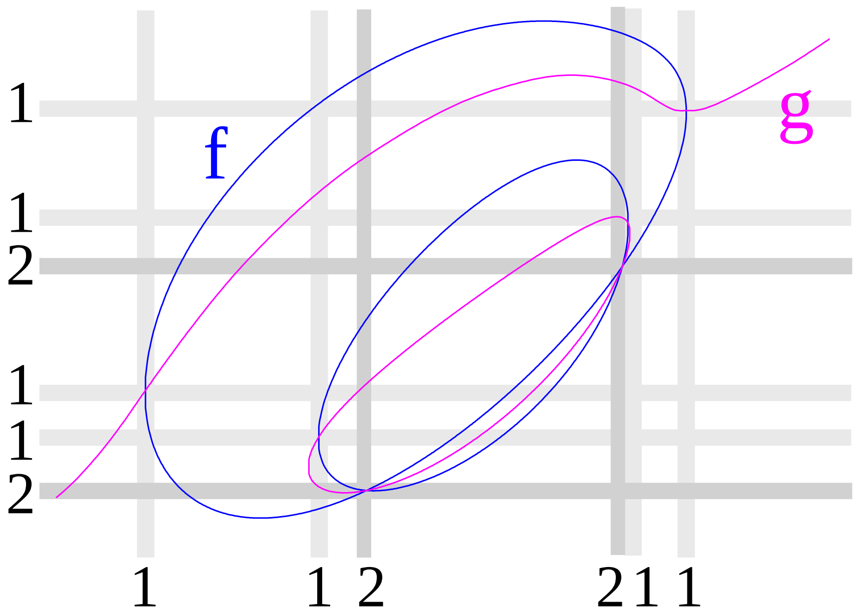


only quadratic polynomials

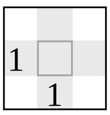
$\Longrightarrow$ explicit solutions

case 2:

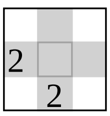
f projection of an intersection curve and $g = f_y$



- label each stripe by 1 or 2 according to multiplicities

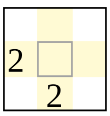
- 

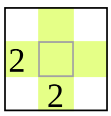
transversal intersection points:
simple box hit counting

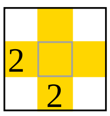
- 

singular points: at most 6
$\Longrightarrow$ we cannot apply explicit solutions

but we can classify the singular points:
(work in progress: intersection curve = four lines)



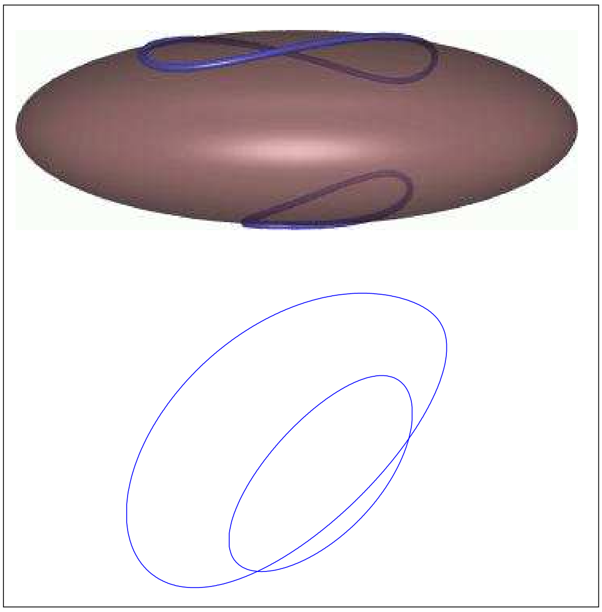




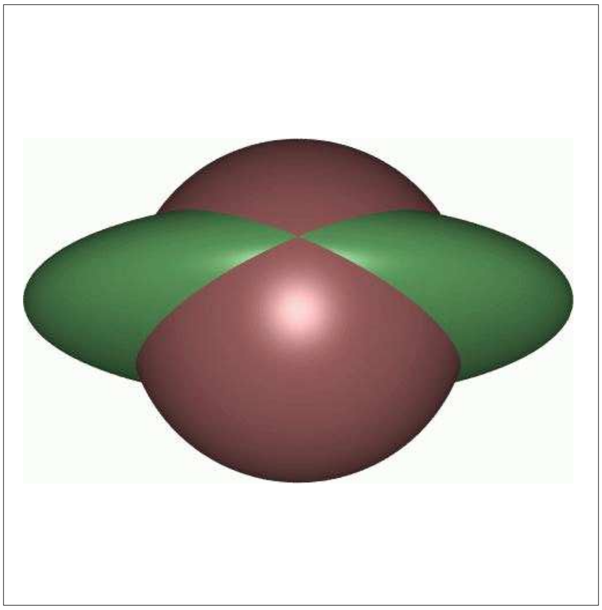
quadratic polynomials:

explicit solutions

There are 2 kinds of singular points



Top–bottom: results from the projection



Genuine: already exists in space
(quadrics have the same tangential plane)

Theorem 3

Either: number of top–bottom points ≤ 2 and
we know a line through these points

Or: "all" points are top–bottom.

Theorem 4

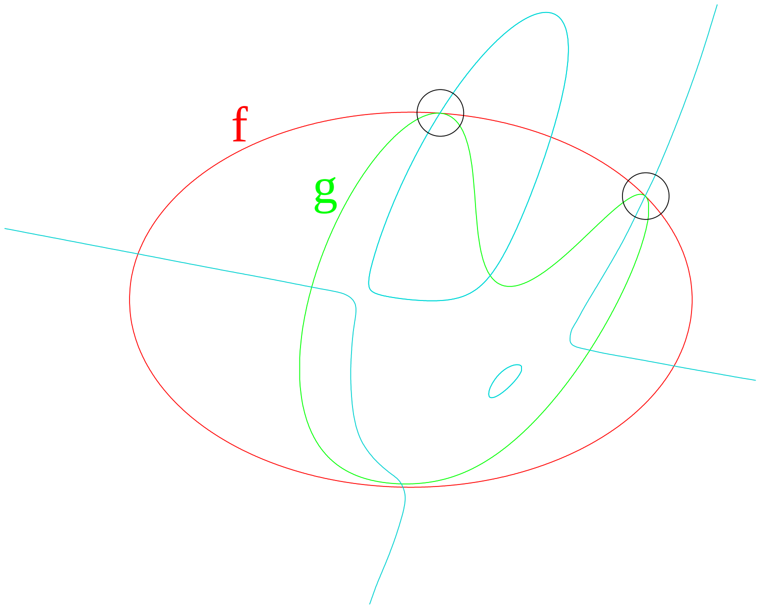
Either: number of genuine points ≤ 2

Or: number of genuine points ≤ 4 and
intersection curve = two lines + conic

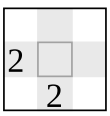
Or: "all" points are genuine.

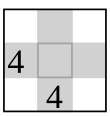
case 3:

f projection of the quadric and g projection of an intersection curve



All intersections = tangential intersections !

- label each stripe by 2 or 4 according to multiplicities
- 

auxiliary **Jacobi curve** $h := f_x g_y - f_y g_x$ transversally cuts through f and g
- 

defined by quadratic polynomial:

  explicit solutions

The Jacobi curve can be generalized:

$$\begin{aligned} h_1 &:= f_x g_y - f_y g_x \\ h_{i+1} &:= f_x (h_i)_y - f_y (h_i)_x \end{aligned}$$

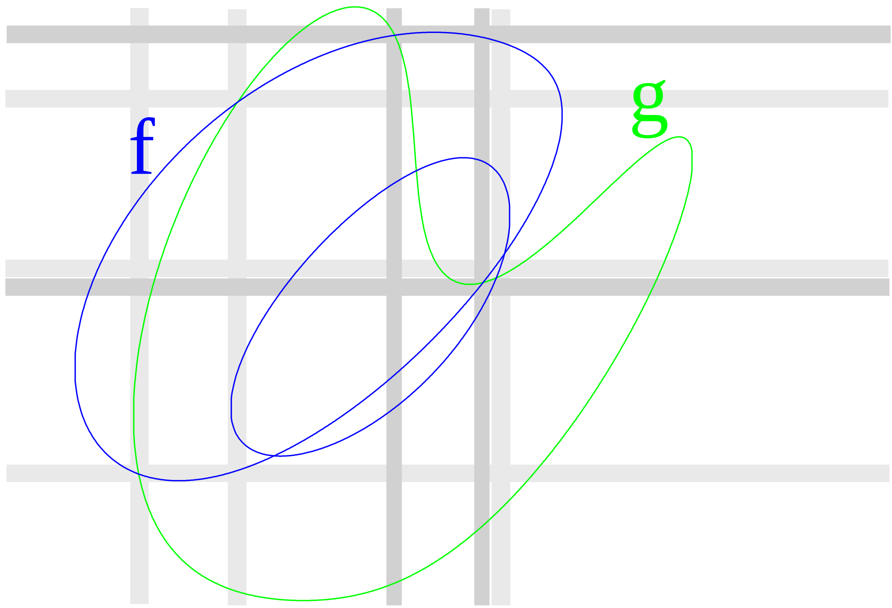
Theorem

The curves f and g have an intersection of multiplicity $k \geq 2$ in the point p and neither f nor g has a singular point in p.

$\Longrightarrow$ h_{k-1} cuts transversally through f and g in p

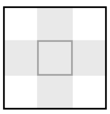
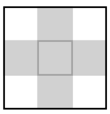
case 4:

f, g projections of intersection curves

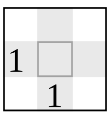


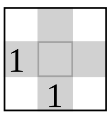
- at most 16 intersection points $\Longrightarrow$ too many

  but we can classify the intersections:

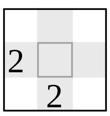
    defined by polynomials of degree ≤ 8

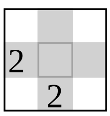
- label each stripe by 1, 2, or 3 (multiplicities)

  – 

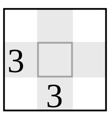


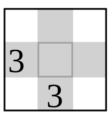
simple box hit counting

  – 



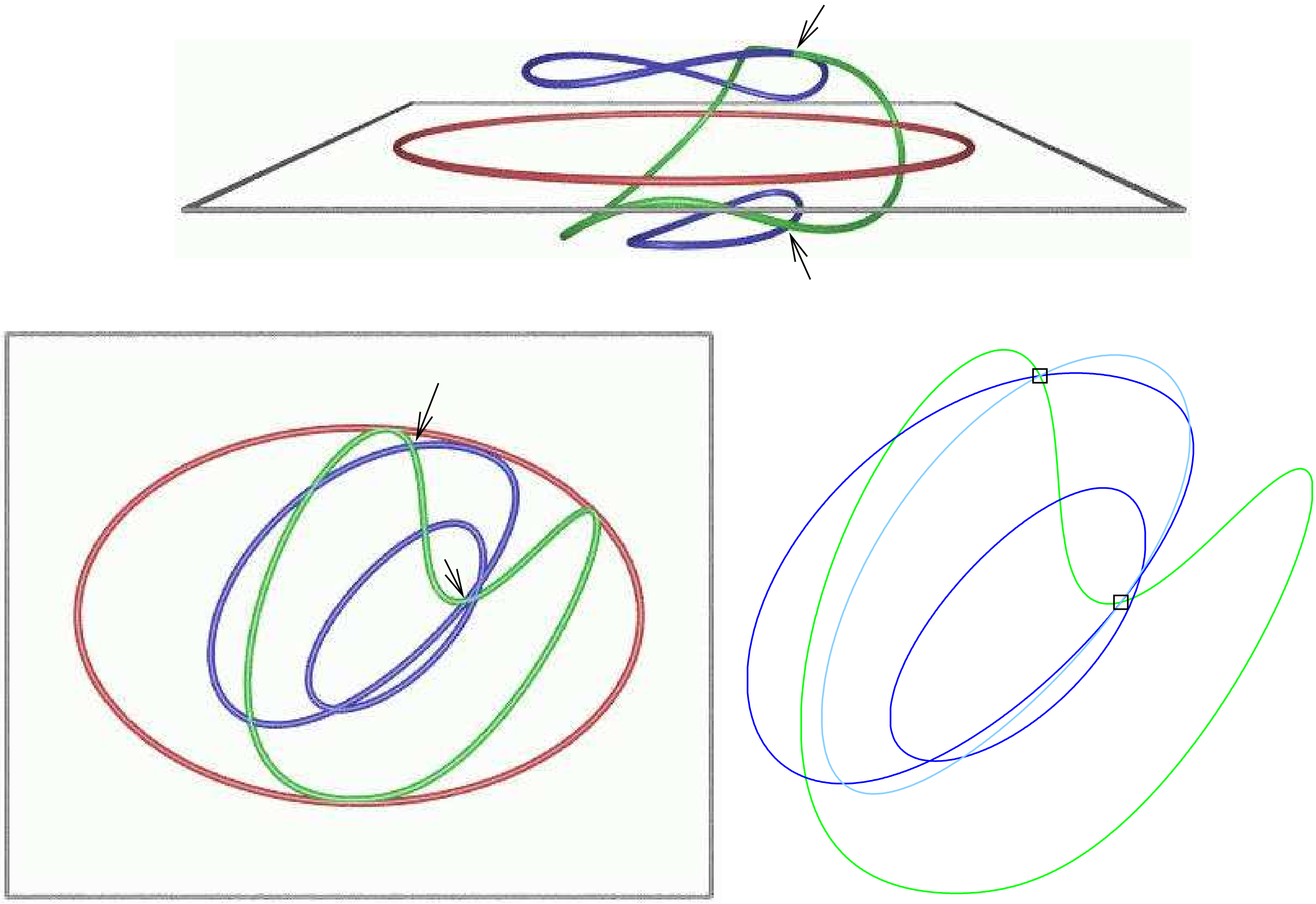
Jacobi curve

  – 



explicit solutions

There are two kinds of intersections of the blue and the green curve



Artificial: results from the projection

Spatial: already exists in space (the red, green, and blue quadric have a common intersection point)

Theorem

There are at most 8 artificial and at most 8 spatial intersection points. $\mathrm{res}(B, G, z)$ is a curve through the spatial points.

Conclusions and further research

Our main results

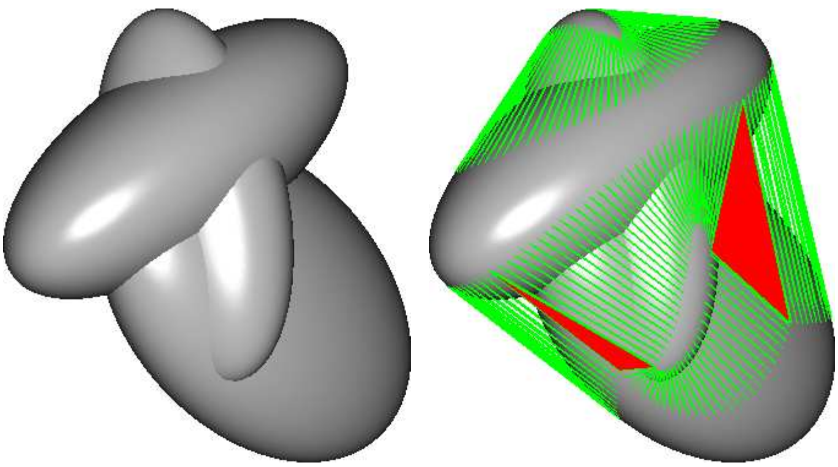
- tangential intersections (no singular points) in arbitrary planar arrangements:
  - multiplicity 2: Jacobi curve + box hit counting
  - arbitrary multiplicity (generalization): auxiliary curve + box hit counting

- arrangements that result from projecting 3d quadrics into the plane:
  - singular points
    - 3 different types,
    - each type ≤ 2 times,
    - quadratic polynomials (one square root?) for each type
  - intersections of two projected intersection curves
    - 2 different types (canonical),
    - each type ≤ 8 times,
    - polynomials of degree ≤ 8 for each type (auxiliary quadric + algorithm of Chionh et al)

We have a prototypical implementation for determining the event points in the plane.

Running time for computing the planar arrangements of three quadrics:

number of digits	5	10	15	20	25
running time in seconds	18	33	56	92	126



Via duality the method can be used to compute the convex hull of ellipoids. (video)

Further research: How can we handle

- arbitrary algebraic surfaces,
- arbitrary algebraic curves?