Solving Systems of Multivariate Polynomials

with Application to Distance Computation between Quadratic Complexes

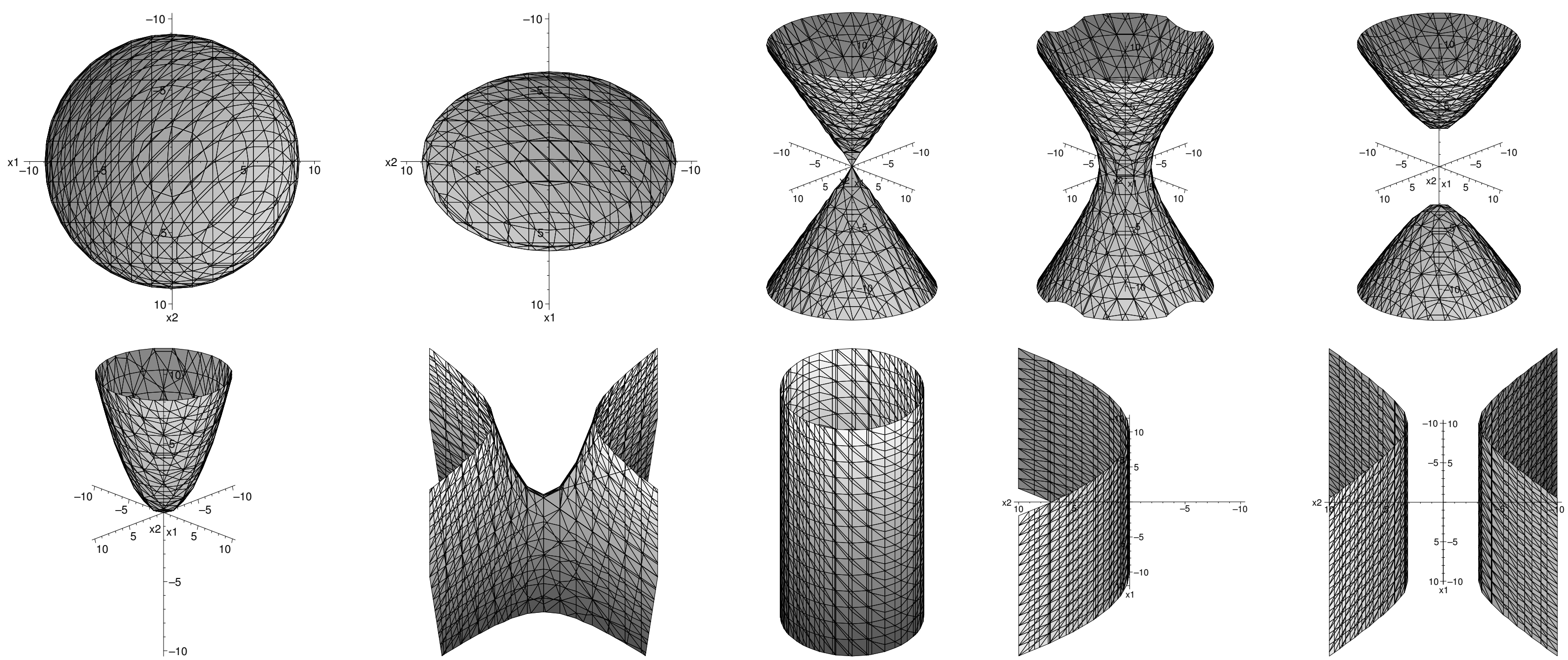
Christian Lennerz

12th November 2002

Quadratic Complexes

Quadratic Complexes can be considered as a generalization of polyhedra with faces embedded on quadrics and conics as edges.

Examples of Quadratic Surfaces:



Closest Points Between Faces

Let F_1 and F_2 be disjoint faces of Quadratic Complexes that are embedded on the quadratic surfaces Q_1 and Q_2, where

$$
\begin{aligned}
Q_1 &:= \{\boldsymbol{x} \in \mathbb{R}^3 \,|\, \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} + a_0 = 0\} \\
Q_2 &:= \{\boldsymbol{y} \in \mathbb{R}^3 \,|\, (\boldsymbol{y} - \boldsymbol{c})^T \boldsymbol{B} (\boldsymbol{y} - \boldsymbol{c}) + b_0 = 0\}.
\end{aligned}
$$

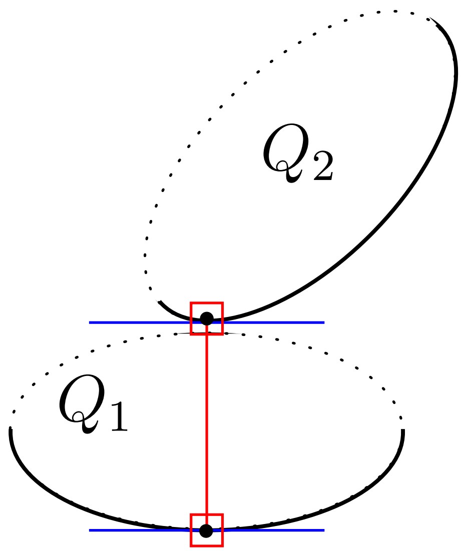
If $(\boldsymbol{p}_1, \boldsymbol{p}_2)$ is a pair of closest points between F_1 and F_2, then either

(i) $(\boldsymbol{p}_1, \boldsymbol{p}_2)$ is an extremum of the quadratic distance function between Q_1 and Q_2 i.e. there are $\alpha, \beta \in \mathbb{R},\ \alpha, \beta \neq 0$ s.t.

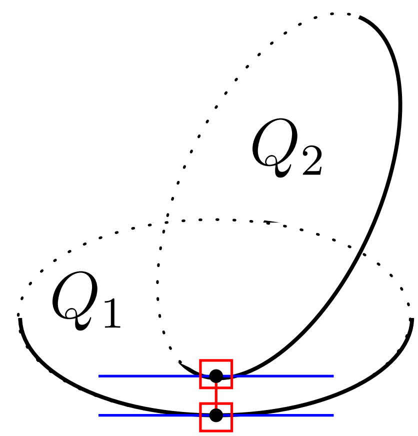
$$\boldsymbol{n}(\boldsymbol{p}_1) = \alpha(\boldsymbol{p}_1 - \boldsymbol{p}_2) \qquad \boldsymbol{n}(\boldsymbol{p}_2) = \beta(\boldsymbol{p}_2 - \boldsymbol{p}_1),$$

where $\boldsymbol{n}(\boldsymbol{p}_i)$ denotes the normal of Q_i in $\boldsymbol{p}_i$, or

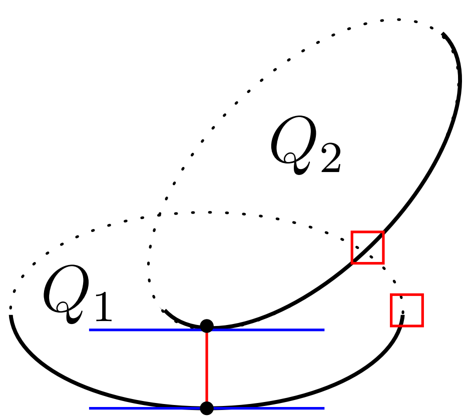
(ii) $\boldsymbol{p}_1$, or $\boldsymbol{p}_2$ lies on the boundary of the face F_1 or F_2, respectively.



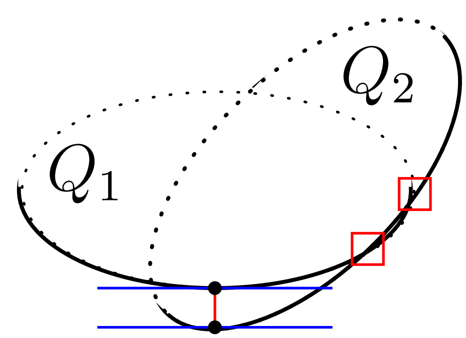
$Q_1 \cap Q_2 = \emptyset$.



$Q_1 \cap Q_2 \neq \emptyset$: case (i).



$Q_1 \cap Q_2 \neq \emptyset$: case (ii).



Preconditon violated.

Computing Distance Extrema in the Surface-Surface Case

By setting up the LAGRANGE Formalism for the problem

$$\min \delta(\boldsymbol{x}, \boldsymbol{y}) \;:=\; \|\boldsymbol{x} - \boldsymbol{y}\|^2 \;=\; (\boldsymbol{x} - \boldsymbol{y})^2, \quad \boldsymbol{x} \in Q_1, \boldsymbol{y} \in Q_2$$

we get the LAGRANGE Function $\mathcal{L}$ and -Conditions $(i), \ldots, (iv)$:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \alpha, \beta) \;=\; (\boldsymbol{x} - \boldsymbol{y})^2 + \alpha(\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} + a_0) + \beta((\boldsymbol{y} - \boldsymbol{c})^T B(\boldsymbol{y} - \boldsymbol{c}) + b_0)$$

$$\begin{array}{rlcl}
(i) & \partial\frac{\mathcal{L}(.)}{\partial \boldsymbol{x}} = 0 & \Longleftrightarrow & \alpha \boldsymbol{A}\boldsymbol{x} = (\boldsymbol{y} - \boldsymbol{x}) \\
(ii) & \partial\frac{\mathcal{L}(.)}{\partial \boldsymbol{y}} = 0 & \Longleftrightarrow & \beta \boldsymbol{B}(\boldsymbol{y} - \boldsymbol{c}) = (\boldsymbol{x} - \boldsymbol{y}) \\
(iii) & \partial\frac{\mathcal{L}(.)}{\partial \alpha} = 0 & \Longleftrightarrow & \boldsymbol{x}^T A\boldsymbol{x} + a_0 = 0 \\
(iv) & \partial\frac{\mathcal{L}(.)}{\partial \beta} = 0 & \Longleftrightarrow & (\boldsymbol{y} - \boldsymbol{c})^T B(\boldsymbol{y} - \boldsymbol{c}) + b_0 = 0
\end{array}$$

Solving The Lagrange System

By setting $\lambda := 1/\alpha$ and $\mu := 1/\beta$ we can derive from (i) and (ii):

$$\begin{aligned} \boldsymbol{x} &= \lambda \boldsymbol{A}^{-1} \boldsymbol{C}_{\lambda,\mu}^{-1} \boldsymbol{c} \\ \boldsymbol{y} - \boldsymbol{c} &= -\mu \boldsymbol{B}^{-1} \boldsymbol{C}_{\lambda,\mu}^{-1} \boldsymbol{c}, \end{aligned}$$

with $\boldsymbol{C}_{\lambda,\mu} := \boldsymbol{E} + \lambda \boldsymbol{A}^{-1} + \mu \boldsymbol{B}^{-1}$.

Substituting $\boldsymbol{x}$ and $\boldsymbol{y} - \boldsymbol{c}$ in (iii) and (iv) we get the system:

$$\begin{aligned} f(\lambda, \mu) &= \lambda^2 \boldsymbol{c}^T \text{adj}(\boldsymbol{C}_{\lambda,\mu}) \boldsymbol{A}^{-1} \text{adj}(\boldsymbol{C}_{\lambda,\mu}) \boldsymbol{c} + a_0 \text{det}(\boldsymbol{C}_{\lambda,\mu})^2 \ = \ 0, \\ g(\lambda, \mu) &= \mu^2 \boldsymbol{c}^T \text{adj}(\boldsymbol{C}_{\lambda,\mu}) \boldsymbol{B}^{-1} \text{adj}(\boldsymbol{C}_{\lambda,\mu}) \boldsymbol{c} + b_0 \text{det}(\boldsymbol{C}_{\lambda,\mu})^2 \ = \ 0, \end{aligned}$$

The Inverse of $\boldsymbol{C}_{\lambda,\mu}$

The inverse of $\boldsymbol{C}_{\lambda,\mu} = \boldsymbol{E} + \lambda \boldsymbol{A}^{-1} + \mu \boldsymbol{B}^{-1}$ is given by

$$
\begin{aligned}
\mathrm{adj}(\boldsymbol{C}_{\lambda,\mu}) &= \mathrm{adj}(\boldsymbol{A}^{-1})\lambda^2 + \mathrm{adj}(\boldsymbol{B}^{-1})\mu^2 + \boldsymbol{T}_A\lambda + \boldsymbol{T}_B\mu + \\
&\quad (\boldsymbol{T}_A\boldsymbol{T}_B - \boldsymbol{T}_{AB})\lambda\mu, \\
\det(\boldsymbol{C}_{\lambda,\mu}) &= \det(\boldsymbol{A}^{-1})\lambda^3 + \det(\boldsymbol{B}^{-1})\mu^3 + \\
&\quad \mathrm{tr}(\mathrm{adj}(\boldsymbol{A}^{-1}))\lambda^2 + \mathrm{tr}(\mathrm{adj}(\boldsymbol{B}^{-1}))\mu^2 + \\
&\quad \mathrm{tr}(\boldsymbol{A}^{-1})\lambda + \mathrm{tr}(\boldsymbol{B}^{-1})\mu + \\
&\quad \mathrm{tr}(\mathrm{adj}(\boldsymbol{A}^{-1})\boldsymbol{B}^{-1})\lambda^2\mu + \mathrm{tr}(\mathrm{adj}(\boldsymbol{B}^{-1})\boldsymbol{A}^{-1})\lambda\mu^2 + \\
&\quad (\mathrm{tr}(\boldsymbol{A}^{-1})\mathrm{tr}(\boldsymbol{B}^{-1}) - \mathrm{tr}(\boldsymbol{A}^{-1}\boldsymbol{B}^{-1}))\lambda\mu,
\end{aligned}
$$

where $\boldsymbol{T}_M := \mathrm{tr}(\boldsymbol{M}^{-1})\boldsymbol{E} - \boldsymbol{M}^{-1}$ for the non-singular matrix $\boldsymbol{M}$.

Degree and Matrix Size Complexity of the Systems

Problem Class	*# Var.*	$deg(v_1)$	$deg(v_2)$	$deg(v_1, v_2)$	$s(M)$
Point - Curve	1	4		4	4
Point - Face	1	6		6	6
Curve-Curve	2 2	4 2	2 4	6 6	24
Curve-Face	2 2	6 2	4 4	10 6	32
Face-Face	2 2	6 6	6 6	6 6	72

Solving a System of Bivariate Polynomial Equations

Given: A system of bivariate polynomial equations

$$f(x, y) = 0$$

$$g(x, y) = 0$$

with positive degrees in x and y.

Goal: Compute all common roots of f and g.

Classification and Previous Work

1. **Newton- and Interval Newton Based Methods**
   [Hansen 88, Leclerc 90, Kearfott 90, Van Hentenryck et al. 95]
2. **Elimination Methods**
   (a) *Methods Based on Gröbner Bases Theory*
   [Ratz 95]
   (b) *Methods Based on Resultant Theory*
   i. Reduction to Univariate Polynomial Solving
   [Canny 93]
   ii. Reduction To Computing Eigenvalues of a Matrix
   [Cox et al. 91, Manocha 94, Stetter 95, Emiris 97].

Newton Methods in Dimension 2

Advantages:

- The multivariate system is solved directly.
- Interval variant guaranties global convergence.
- Reliable results guarantied by inclusion properties.

Our Experience:

- The method works very well for the Curve-Curve system
  Running Time: ≈ 10 msec.
- ... but breaks down on our Surface-Surface formulation.
  Running Time: > 30 min.

Alternatives

1. **Reformulation of the Multivariate Problem:**
   If Q_1 and Q_2 are given explicitly then the conditions of case (i) lead to a system of equations that is no longer polynomial.
   Running Time: ≈ 45 msec.

2. **Special Case: Disjoint Ellipsoids:**
   When starting with the points where the line through the centers intersects the ellipsoids, a convex optimization method will converge to the global minimum of distance.
   Running Time: < 0.01 msec.

Elimination Methods

To eliminate the variable x we write

$$f = \sum_{i=0}^{m} a_i(y)x^i \qquad g = \sum_{j=0}^{n} b_j(y)x^j \qquad a_m, b_n \neq 0,$$

where a_i, $0 \leq i \leq m$, and b_j, $0 \leq j \leq n$, are polynomials in y.

Proposition

Given $f, g \in \mathbb{C}[x, y]$, let $a_m, b_n \in \mathbb{C}[y]$ and $\bar{y} \in \mathbb{C}$. If $Res(f, g, x) \in \mathbb{C}[y]$ vanishes at $\bar{y}$, then either

(i) a_m or b_n vanishes at $\bar{y}$ or

(ii) there is $\bar{x} \in \mathbb{C}$ such that f and g vanish at $(\bar{x}, \bar{y}) \in \mathbb{C}^2$.

Sample Point Method

We have to find the roots of $Res(f, g, x)$ that is a polynomial of degree N in y.

First Step: *Determining the Power Representation of* $Res(f, g, x)$*:*
This can be done either by

- Symbolic Evaluation of $\det(S(f, g, x))$ using polynomial arithmetic or
- 1. Numerical Evaluation of $Res(f, g, x)$ at $N + 1$ sample points.
  2. Computing its coefficients by determining the interpolating polynomial.

Evaluating $Res(f, g, x)$ at a Sample Point

Given: $N + 1$ sample points $y_0 \dots y_N$.

Goal: Compute $Res(f, g, x)(y_k) \in \mathbb{C}$.

Substituting y_k, $0 \leq k \leq N$, for y in f and g gives:

$$f_k := f(x, y_k) = \sum_{i=0}^{m} a_{k,i} x^i \qquad g_k := g(x, y_k) = \sum_{j=0}^{n} b_{k,j} x^j$$

The determinant d_k of the Sylvester Matrix $\boldsymbol{S}_k := \boldsymbol{S}(f_k, g_k, x)$ is the desired evaluation of $Res(f, g, x)$ at the sample point y_k, i.e.

$$d_k := \mathrm{det}(\boldsymbol{S}_k) = Res(f_k, g_k, x)(y_k)\,, \quad 0 \leq k \leq N$$

Determining the Interpolating Polynomial

Having evaluated $Res(f,g,x)$ at $N+1$ sample points, we get the coefficients c_i, $0 \le i \le N$, of the interpolating polynomial

$$Res(f,g,x) \quad = \quad \sum_{i=0}^{N} c_i y^i$$

by solving the following linear Vandermode System

$$\begin{bmatrix} 1 & y_0 & y_0^2 & \dots & y_0^N \\ 1 & y_1 & y_1^2 & \dots & y_1^N \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & y_N & y_N^2 & \dots & y_N^N \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_N \end{bmatrix} \quad = \quad \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_N \end{bmatrix}.$$

Second Step: *Solving The Resultant Polynomial* $Res(f, g, x)$*:*

The characteristic polynomial of the Companion Matrix

$$M := \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -\frac{c_0}{c_N} & -\frac{c_1}{c_N} & -\frac{c_2}{c_N} & \cdots & -\frac{c_{N-1}}{c_N} \end{bmatrix}$$

has exactly the same roots as the polynomial $q(y) := \sum_{i=0}^{N} c_i y^i$.

Evaluation of the Sample Point Method

- Really fast, if we could compute in double precision.
- Numerical stability depends on the distribution of sample points.
- To obtain coefficients that are accurate enough one has to choose the sample points close to the actual roots.

Main Reasons for the Lack of Stability and Accuracy

1. The evaluation of the determinant of the $(m+n) \times (m+n)$ matrix is numerically difficult and leads to inaccurate input values for the interpolation routine.
2. It is well known that the Vandermonde System can be quite ill-conditioned.

Improvements

Reducing the Size of the Sylvester Matrix

$$
\left[\begin{array}{ccc|ccccc}
a_5 & & & b_3 & & & & \\
a_4 & a_5 & & b_2 & b_3 & & & \\
a_3 & a_4 & a_5 & b_1 & b_2 & b_3 & & \\
\hline
a_2 & a_3 & a_4 & b_0 & b_1 & b_2 & b_3 & \\
a_1 & a_2 & a_3 & & b_0 & b_1 & b_2 & b_3 \\
a_0 & a_1 & a_2 & & & b_0 & b_1 & b_2 \\
 & a_0 & a_1 & & & & b_0 & b_1 \\
 & & a_0 & & & & & b_0
\end{array}\right]
=
\left[\begin{array}{c|c}
A_1 \in \mathbb{R}^{n\times n} & B_1 \in \mathbb{R}^{n\times m} \\
\hline
A_2 \in \mathbb{R}^{m\times n} & B_2 \in \mathbb{R}^{m\times m}
\end{array}\right]
$$

Decomposition of $\boldsymbol{S}$ by the Schur Complement Theorem

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{A}_2\boldsymbol{A}_1^{-1} & \boldsymbol{I} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{B}_1 \\ \boldsymbol{0} & \boldsymbol{B}_2 - \boldsymbol{A}_2\boldsymbol{A}_1^{-1}\boldsymbol{B}_1 \end{bmatrix},$$

where $\boldsymbol{B}_2 - \boldsymbol{A}_2\boldsymbol{A}_1^{-1}\boldsymbol{B}_1$ is the Schur Complement of $\boldsymbol{A}_1$ in $\boldsymbol{S}$.

If f_k, $0 \leq k \leq N$, is normalized, the determinant of $\boldsymbol{S}_k$ simplifies to

$$\begin{aligned} \det(\boldsymbol{S}_k) &= \det(\boldsymbol{A}_1)\det(\boldsymbol{B}_2 - \boldsymbol{A}_2\boldsymbol{A}_1^{-1}\boldsymbol{B}_1) \\ &= \det(\boldsymbol{B}_2 - \boldsymbol{A}_2\boldsymbol{A}_1^{-1}\boldsymbol{B}_1) \end{aligned}$$

Remark: $\boldsymbol{B}_2 - \boldsymbol{A}_2\boldsymbol{A}_1^{-1}\boldsymbol{B}_1$ is of size $m \times m$.

Reducing the Degree of the Resultant Polynomial

Conjecture:

Let f and g be our system of polynomial equations, i.e.

$$\begin{aligned} f(\lambda, \mu) &= \lambda^2 \boldsymbol{c}^T \text{adj}(\boldsymbol{C}_{\lambda,\mu}) \boldsymbol{A}^{-1} \text{adj}(\boldsymbol{C}_{\lambda,\mu}) \boldsymbol{c} + a_0 \text{det}(\boldsymbol{C}_{\lambda,\mu})^2 = 0, \\ g(\lambda, \mu) &= \mu^2 \boldsymbol{c}^T \text{adj}(\boldsymbol{C}_{\lambda,\mu}) \boldsymbol{B}^{-1} \text{adj}(\boldsymbol{C}_{\lambda,\mu}) \boldsymbol{c} + b_0 \text{det}(\boldsymbol{C}_{\lambda,\mu})^2 = 0, \end{aligned}$$

and the system $\boldsymbol{h}$ be defined as follows:

$$\boldsymbol{h}(\lambda, \mu) := (h_1, h_2, h_3)^T = \boldsymbol{C}_{\lambda,\mu}^{-1} \boldsymbol{c} = \boldsymbol{0}.$$

Then the roots of the polynomial $Res(h_i, h_j)$, $1 \leq i < j \leq 3$, define a polynomial p_{ij} of degree 12 that divide $Res(f, g)$.

Consequence:
- Sufficient to solve $Res(f, g)/p_{ij}$ of degree 24.
- Division can be done sample pointwise.

Some Ideas to Consider

- **Smoothing instead of Interpolating**
  Compute the best fit polynomial of degree N through a larger set of sample points.
- **Implicit instead of Explicit Representation**
  - There are root finding algorithms that only evaluate the polynomial (and its derivative) at a given value: Secant-, Regula Falsi-, or NEWTON's Method.
  - There are numerically more accurate techniques to accomplish this task when the polynomial is given implicitly by its set of sample points: LAGRANGE- or BERNSTEIN -Representation and NEVILLE's evaluation scheme [Teukolsky et al. 94].

Open Questions

- Are the polynomials of the system $\boldsymbol{h} = \mathbf{0}$ also common factors of the system $f = g = 0$.
  If yes, is it possible to find the respective factorization?
  Is there a symbolic multivariate GCD algorithm and implementation?
- How can we find good choices for the set of sample points?
  Can we make use of the geometric interpretation?

Eigenvalues of the Generalized Companion Matrix

Given: A system of bivariate polynomial equations

$$f(x,y) \;=\; \sum_{i=0}^{m} a_i(y)x^i \;=\; 0, \quad a_m \neq 0$$

$$g(x,y) \;=\; \sum_{j=0}^{n} b_j(y)x^j \;=\; 0, \quad b_n \neq 0$$

with positive degrees in x and y.

Proposition:

Let $f, g \in \mathbb{C}[x,y]$ be polynomials of positve degrees m and n in x Let $\bar{y} \in \mathbb{C}$. Then $f(x,\bar{y})$ and $g(x,\bar{y})$ have a common factor if and only if there are polynomials $\alpha, \beta \in \mathbb{C}[x]$ s.t.

(i) α, β are not both the zero polynomial.

(ii) α has degree at most $n-1$ and β at most $m-1$.

(iii) $\alpha f(x,\bar{y}) + \beta g(x,\bar{y}) \equiv 0$.

Observation: There exist $\boldsymbol{\xi}, \boldsymbol{v} \in \mathbb{C}^{m+n}$ $s.t.$ $\alpha f + \beta g = \boldsymbol{\xi}^T \boldsymbol{S}(f, g, x)\boldsymbol{v}$

$$
\begin{bmatrix} x^{m+n-1} \\ x^{m+n-2} \\ \vdots \\ x^0 \end{bmatrix}^T
\underbrace{\begin{bmatrix}
a_m & & & & b_n & & & \\
a_{m-1} & a_m & & & b_{n-1} & b_n & & \\
a_{m-2} & a_{m-1} & \ddots & & b_{n-2} & b_{n-1} & \ddots & \\
\vdots & & \ddots & a_m & \vdots & & \ddots & b_n \\
 & \vdots & & a_{m-1} & & \vdots & & b_{n-1} \\
a_0 & & & & b_0 & & & \\
 & a_0 & & \vdots & & b_0 & & \vdots \\
 & & \ddots & & & & \ddots & \\
 & & & a_0 & & & & b_0
\end{bmatrix}}_{n \text{ columns} \quad\quad m \text{ columns}}
\begin{bmatrix} \alpha_{n-1} \\ \alpha_{n-2} \\ \vdots \\ \alpha_0 \\ \beta_{m-1} \\ \beta_{m-2} \\ \vdots \\ \beta_0 \end{bmatrix}
$$

Conclusion:

The y-coordinate of all common roots of f and g are exactly the values for which

$$\boldsymbol{\xi}^T \boldsymbol{S}(f,g,x)\boldsymbol{v} \equiv 0, \quad \boldsymbol{v} \neq \mathbf{0} \iff \boldsymbol{S}(f,g,x)\boldsymbol{v} = \mathbf{0}, \quad \boldsymbol{v} \neq \mathbf{0}$$

Writing $\boldsymbol{S}(f,g,x)$ as the matrix polynomial

$$\boldsymbol{S}(f,g,x) = \sum_{i=0}^{d} \boldsymbol{S}_i y^i,$$

where $\boldsymbol{S}_i$, $0 \leq i \leq d$, is the matrix consisting of all coefficients of degree i in y and $d := \max\{\text{deg}_y(f), \text{deg}_y(g)\}$, we get

$$\sum_{i=0}^{d} \boldsymbol{S}_i y^i \boldsymbol{v} = \mathbf{0}, \quad \boldsymbol{v} \neq \mathbf{0}.$$

Linearization

To linearize the system we use the substitution

$$\boldsymbol{u}_i \;:=\; y^i \boldsymbol{v}, \quad 0 \le i \le d-1$$

and transform the condition to isolate $\boldsymbol{S}_d$:

$$\begin{aligned}
\sum_{i=0}^{d} \boldsymbol{S}_i y^i \boldsymbol{v} &= \mathbf{0} \quad \boldsymbol{v} \neq \mathbf{0} \\
\Leftrightarrow \quad \sum_{i=0}^{d} (-\boldsymbol{S}_i) y^i \boldsymbol{v} &= y \boldsymbol{S}_d y^{d-1} \boldsymbol{v}, \quad \boldsymbol{v} \neq \mathbf{0} \\
\Leftrightarrow \quad \sum_{i=0}^{d-1} (-\boldsymbol{S}_i) \boldsymbol{u}_i &= y \boldsymbol{S}_d \boldsymbol{u}_{d-1}
\end{aligned}$$

Case 1: $\boldsymbol{S}_d$ is non-singular

Multiplying by $\boldsymbol{S}_d^{-1}$ gives a Simple Eigenvalue Problem:

$$\sum_{i=0}^{d-1}(-\tilde{\boldsymbol{S}}_i)\boldsymbol{u}_i \;=\; y\boldsymbol{u}_{d-1} \quad \text{with } \tilde{\boldsymbol{S}}_i = \boldsymbol{S}_d^{-1}\boldsymbol{S}_i,\ 0 \le i \le d-1.$$

$$\Leftrightarrow \quad \begin{bmatrix} \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{I} \\ -\tilde{\boldsymbol{S}}_0 & -\tilde{\boldsymbol{S}}_1 & -\tilde{\boldsymbol{S}}_2 & \cdots & -\tilde{\boldsymbol{S}}_{d-1} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \\ \vdots \\ \boldsymbol{u}_{d-1} \end{bmatrix} = y \cdot \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \\ \vdots \\ \boldsymbol{u}_{d-1} \end{bmatrix}$$

$$\Leftrightarrow \qquad \boldsymbol{M}\boldsymbol{u} = y\boldsymbol{u}$$

$$\Leftrightarrow \qquad (\boldsymbol{M} - y\boldsymbol{I})\boldsymbol{u} \;=\; \mathbf{0}$$

Case 2: $\boldsymbol{S}_d$ is singular

Since $\boldsymbol{S}_d$ is singular we get a Generalized Eigenvalue Problem:

$$\sum_{i=0}^{d-1}(-\boldsymbol{S}_i)\boldsymbol{u}_i = y\boldsymbol{S}_d\boldsymbol{u}_{d-1} \qquad \Leftrightarrow$$

$$\begin{bmatrix} \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{I} \\ -\boldsymbol{S}_0 & -\boldsymbol{S}_1 & -\boldsymbol{S}_2 & \cdots & -\boldsymbol{S}_{d-1} \end{bmatrix} \cdot \boldsymbol{u} = y \cdot \begin{bmatrix} \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{S}_d \end{bmatrix} \boldsymbol{u}$$

$$\Leftrightarrow \qquad \boldsymbol{M}_1\boldsymbol{u} = y\boldsymbol{M}_2\boldsymbol{u}$$

$$\Leftrightarrow \qquad (\boldsymbol{M}_1 - y\boldsymbol{M}_2)\boldsymbol{u} = \mathbf{0}$$

Open Questions

- In our application $\boldsymbol{S}_d$ is singular. If we can show that $\boldsymbol{S}_0$ is non-singular we only have to solve the Simple Eigenvalue Problem. Then we could consider $\bar{\boldsymbol{S}}(f, g, x) = \sum_{i=0}^{d} \boldsymbol{S}_{d-i} y^i$ that leads to the reciprocal roots of $\boldsymbol{S}(f, g, x)$.
- How can we make use of the sparsity of the matrix $\boldsymbol{M}$?
  Potential Solutions:
  1. *Power Iteration Methods* which only perform matrix-vector multiplications [Wilkinson 65].
     Problem: Elimination of computed eigenvalues.
     - Domain Decomposition and Pruning [Manocha 94].
     - Deflation [Saad 91].
  2. *Review of QR-Methods* [Teukolsky et al 94].