Solving Polynomial Systems by Homotopy Continuation Methods

Jan Verschelde

University of Illinois at Chicago
Department of Mathematics, Statistics, and Computer Science

NUMA Seminar, 1 February 2019, K.U. Leuven, Belgium

Outline

1. Introduction
   - polynomial systems
   - PHCpack and phcpy
2. Polynomial Homotopy Continuation
   - solving polynomial systems numerically
   - exploiting sparse structures
3. Numerical Irreducible Decomposition
   - witness sets
   - cascades of homotopies
   - computational experiments
4. Parallel and Cloud Computing
   - speedup and quality up
   - web interfaces

Solving Polynomial Systems

1. Introduction
   - polynomial systems
   - PHCpack and phcpy
2. Polynomial Homotopy Continuation
   - solving polynomial systems numerically
   - exploiting sparse structures
3. Numerical Irreducible Decomposition
   - witness sets
   - cascades of homotopies
   - computational experiments
4. Parallel and Cloud Computing
   - speedup and quality up
   - web interfaces

the cyclic 4-roots system

The equations for the cyclic 4-roots problem are

$$\mathbf{f}(\mathbf{x}) = \left\{ \begin{array}{c} x_1 + x_2 + x_3 + x_4 = 0 \\ x_1x_2 + x_2x_3 + x_3x_4 + x_4x_1 = 0 \\ x_1x_2x_3 + x_2x_3x_4 + x_3x_4x_1 + x_4x_1x_2 = 0 \\ x_1x_2x_3x_4 - 1 = 0. \end{array} \right.$$

This system has two solution curves of degree two.
There are no isolated solutions.

Lemma (Backelin's Lemma)

If n has a quadratic divisor, $n = \ell k^2$, $\ell < k$, then there are $(k-1)$-dimensional cyclic n-roots.

Cases we can do are $n = 8, 9, 12$, $n = 16$ is still too hard.

the cyclic 5-roots system

The equations for the cyclic 5-roots problem are

$$\mathbf{f}(\mathbf{x}) = \left\{ \begin{array}{c} x_1 + x_2 + x_3 + x_4 + x_5 = 0 \\ x_1x_2 + x_2x_3 + x_3x_4 + x_4x_5 + x_5x_1 = 0 \\ x_1x_2x_3 + x_2x_3x_4 + x_3x_4x_5 + x_4x_5x_1 + x_5x_1x_2 = 0 \\ x_1x_2x_3x_4 + x_2x_3x_4x_5 + x_3x_4x_5x_1 + x_4x_5x_1x_2 + x_5x_1x_2x_3 = 0 \\ x_1x_2x_3x_4x_5 - 1 = 0. \end{array} \right.$$

70 isolated solutions, no positive dimensional solutions.

Theorem (Uffe Haagerup)

If $n = p$, a prime, the number of cyclic p-roots equals $\left(\begin{array}{c} 2p-2 \\ p-1 \end{array} \right)$.

For $n = 6, 7, 10, 11, 13$, the number of cyclic n-roots equals respectively 156, 924, 34900, 184756, 2704156.

biunimodular vectors

A biunimodular vector $\mathbf{u} \in \mathbb{C}^n$ of a unitary matrix A:

1. all coordinates of $\mathbf{u}$ have modulus one: $|u_i| = 1$, and
2. for $\mathbf{v} = A\mathbf{u}$, $|v_i| = 1$, $i = 1, 2, \ldots, n$.

This notion dates back to Gauss.

A complex Hadamard matrix H of size n:

1. all entries are complex numbers with modulus one, and
2. $H^* H = nI$, where I is the identity matrix.

There is a one-to-one correspondence between cyclic n-roots and circulant Hadamard matrices.

Conjecture (Björck and Saffari)

If n is not divisible by a square, then the set of cyclic n-roots is finite.

Solving Polynomial Systems

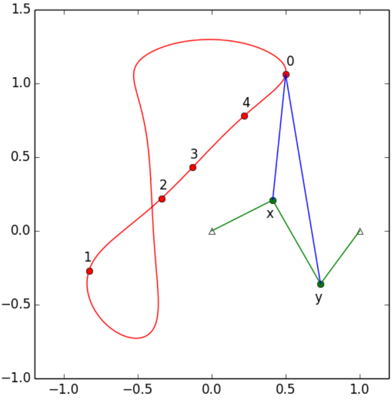
1. Introduction
   - polynomial systems
   - PHCpack and phcpy
2. Polynomial Homotopy Continuation
   - solving polynomial systems numerically
   - exploiting sparse structures
3. Numerical Irreducible Decomposition
   - witness sets
   - cascades of homotopies
   - computational experiments
4. Parallel and Cloud Computing
   - speedup and quality up
   - web interfaces

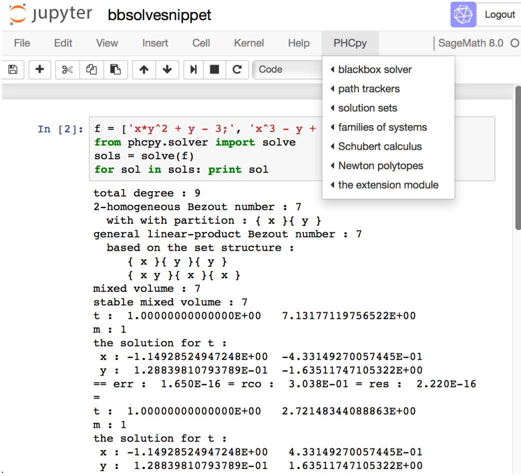
Solving Polynomial Systems with PHCpack and phcpy

PHCpack is software for Polynomial Homotopy Continuation

phcpy is a new Python package, available at `www.phcpack.org`

use case from the phcpy tutorial:





reproduces J. Mech. Design paper

Solving Polynomial Systems

1. Introduction
   - polynomial systems
   - PHCpack and phcpy
2. Polynomial Homotopy Continuation
   - solving polynomial systems numerically
   - exploiting sparse structures
3. Numerical Irreducible Decomposition
   - witness sets
   - cascades of homotopies
   - computational experiments
4. Parallel and Cloud Computing
   - speedup and quality up
   - web interfaces

solving polynomial systems numerically

What does *numerically* solving a polynomial system mean?

- The input data may be given with limited accuracy.
- The output is approximate.

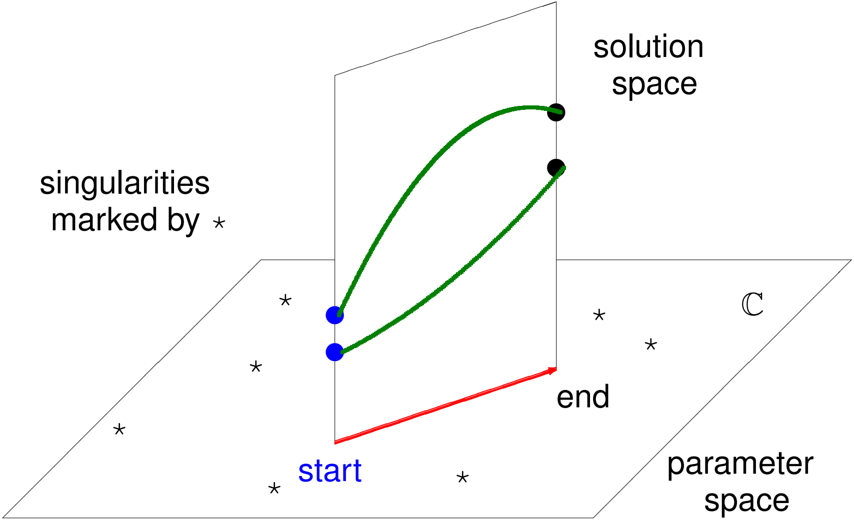
A polynomial in several variables consist of

- exact data: exponents span its Newton polytope; and
- approximate data: coefficients, parameter values.

Based on the exact data (the exponents) we compute an upper bound on the number of solutions.

At the end of the numerical computations, we verify whether the number of solutions matches the apriori computed upper bound.

parameter continuation schematic in $\mathbb{C}$



polynomial homotopy continuation

$\mathbf{f}(\mathbf{x}) = \mathbf{0}$ is a polynomial system we want to solve,
$\mathbf{g}(\mathbf{x}) = \mathbf{0}$ is a start system ($\mathbf{g}$ is similar to $\mathbf{f}$) with known solutions.

A homotopy $\mathbf{h}(\mathbf{x}, t) = \gamma(1 - t)\mathbf{g}(\mathbf{x}) + t\,\mathbf{f}(\mathbf{x}) = \mathbf{0}$, $t \in [0, 1]$, $\gamma \in \mathbb{C}$,
to solve $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ defines solution paths $\mathbf{x}(t)$: $\mathbf{h}(\mathbf{x}(t), t) \equiv \mathbf{0}$.

Numerical continuation methods track the paths $\mathbf{x}(t)$, from $t = 0$ to 1.

Newton's method is the most computationally intensive stage:

1. Evaluation and differentiation of all polynomials in the system.
2. Solve a linear system for the update to the approximate solution.

Bootstrapping to solve a start system $\mathbf{g}(\mathbf{x}) = \mathbf{0}$:

- Random coefficients of $\mathbf{g}$ imply that all solutions are regular.
- Polyhedral homotopies deform $\mathbf{g}$ to 2-nomial systems.

the gamma trick

A homotopy $\mathbf{h}(\mathbf{x}, t) = \gamma(1 - t)\mathbf{g}(\mathbf{x}) + t\,\mathbf{f}(\mathbf{x}) = \mathbf{0}$ deforms the polynomials in the start system $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ to the polynomials in the system $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ that has to be solved, as t goes from zero to one.

The constant $\gamma \in \mathbb{C}$ is generated at random.

1. For $t = 0$, $\mathbf{h}(\mathbf{x}, 0) = \mathbf{g}(\mathbf{x}) = \mathbf{0}$ has only regular solutions.
2. The number of solutions of $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ equals the upper bound, is maximal for all systems in the homotopy $\mathbf{h}(\mathbf{x}, t) = \mathbf{0}$.

The main theorem in elimination theory states that, in projective space, the projection of an algebraic set is again an algebraic set.

Consider the discriminant variety of $\mathbf{h}(\mathbf{x}, t) = \mathbf{0}$, eliminate $\mathbf{x}$. After elimination, the polynomial $p(t) = 0$ has its roots where the solutions of $\mathbf{h}(\mathbf{x}, t) = \mathbf{0}$ are singular. Because $p(0) \neq 0$, $p \not\equiv 0$ and there are only finitely many singularities ***in the complex plane***.

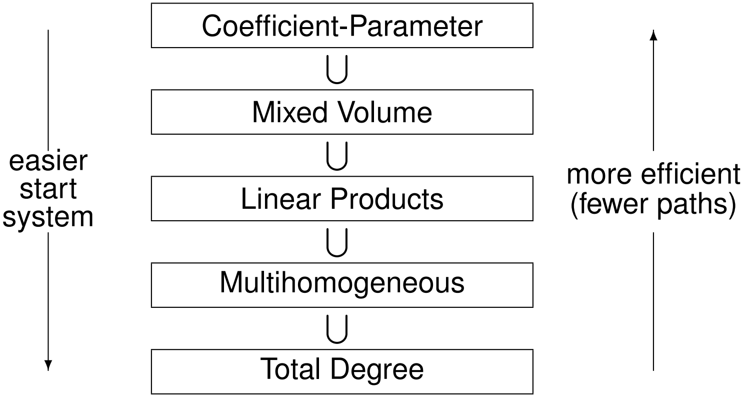
optimal homotopies

Exploiting the structure correctly is critical for the performance of a homotopy. We say that a homotopy is ***optimal*** if every solution path converges to a solution of a generic instance of the problem.

We have optimal homotopies for three classes of systems:

1. Linear-product start systems in linear homotopies.
   Given a polynomial in several variables, deform the polynomial to a product of linear factors.
2. Polyhedral homotopies for sparse polynomial systems.
   The sparsest kind of systems have two monomials with nonzero coefficient in every equation.
3. Pieri homotopies and Littlewood-Richardson homotopies for Schubert problems in enumerative geometry.
   Given four lines in three space, compute all lines which meet the four given lines in a point.

a totem pole of homotopies



Solving Polynomial Systems

1. Introduction
   - polynomial systems
   - PHCpack and phcpy
2. Polynomial Homotopy Continuation
   - solving polynomial systems numerically
   - exploiting sparse structures
3. Numerical Irreducible Decomposition
   - witness sets
   - cascades of homotopies
   - computational experiments
4. Parallel and Cloud Computing
   - speedup and quality up
   - web interfaces

Newton polytopes and mixed volumes

recognizing the sparse structure of a polynomial system

Most polynomials have few nonzero coefficients:

$$f(\mathbf{x}) = \sum_{\mathbf{a} \in A} c_{\mathbf{a}} \mathbf{x}^{\mathbf{a}}, \quad c_{\mathbf{a}} \neq 0, \quad \mathbf{x}^{\mathbf{a}} = x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}.$$

The *support* A of f spans the Newton polytope $P = \mathrm{ConvHull}(A)$.
$\mathcal{P} = (P_1, P_2, \ldots, P_n)$ collects the Newton polytopes of a system f.

Remember the principle of conservation of number (classical)
or coefficient-parameter polynomial continuation (numerical):
$N_{\mathbf{c}}$ = the number of solutions for generic coefficients $\mathbf{c}$.

Bernshteǐn's theorem (1975): $N_{\mathbf{c}}$ depends only on $\mathcal{P}$.

In particular: $N_{\mathbf{c}} = V(\mathcal{P})$, the mixed volume of $\mathcal{P}$.
Special case: $P = P_1 = P_2 = \cdots = P_n$: $N_{\mathbf{c}} = n!\mathrm{volume}(P)$.

the theorems of Bernshteĭn

Mixed volumes relate volume to surface area:

$$V_n(P_1, P_2, \ldots, P_n) = \sum_{\mathbf{v}} p_1(\mathbf{v}) \ V_{n-1}(\mathrm{in}_{\mathbf{v}} P_2, \ldots, \mathrm{in}_{\mathbf{v}} P_n),$$

$\mathbf{v} \in \mathbb{Z}^n$, $\gcd(\mathbf{v}) = 1$, $p_1(\mathbf{v}) = \min\limits_{\mathbf{x} \in P_1} \langle \mathbf{x}, \mathbf{v} \rangle$ is a support function
$\mathrm{in}_{\mathbf{v}} P_k = \{ \ \mathbf{x} \in P_k \mid \langle \mathbf{x}, \mathbf{v} \rangle = p_k(\mathbf{v}) \ \}$ is a face of P_k.

Theorem A: The number of roots of a generic system equals the mixed volume of its Newton polytopes.

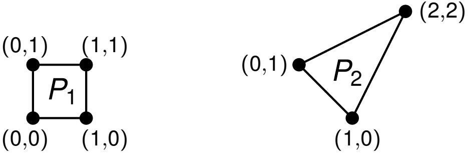
Theorem B: Solutions at infinity are solutions of systems supported on faces of the Newton polytopes.

polygons in general position

The system

$$\mathbf{f}(\mathbf{x}) = \begin{cases} c_{1,(1,1)}x_1x_2 + c_{1,(1,0)}x_1 + c_{1,(0,1)}x_2 + c_{1,(0,0)} = 0 \\ c_{2,(2,2)}x_1^2x_2^2 + c_{2,(1,0)}x_1 + c_{2,(0,1)}x_2 = 0. \end{cases}$$

has Newton polygons:



$$\forall \mathbf{v} \neq \mathbf{0} : \#\mathrm{in}_\mathbf{v} A_1 + \#\mathrm{in}_\mathbf{v} A_2 \leq 3 \Rightarrow V(P_1, P_2) = 4 \text{ is always exact,}$$

for all nonzero coefficients of $\mathbf{f}$, because ≥ 4 monomials are needed for $\mathrm{in}_\mathbf{v}\mathbf{f}(\mathbf{x}) = \mathbf{0}$ to have all its roots in $(\mathbb{C}^*)^2$.

polyhedral homotopies

constructive proofs of Bernshteĭn's theorems

Polyhedral homotopies implement Bernshteĭn's theorems.

An effective complement to the *cheater's* homotopy.

The methods are *optimal* in the sense that every solution path converges to an isolated solution . . .

. . . *provided* the system is sufficiently generic.

Solving Polynomial Systems

1. Introduction
   - polynomial systems
   - PHCpack and phcpy
2. Polynomial Homotopy Continuation
   - solving polynomial systems numerically
   - exploiting sparse structures
3. **Numerical Irreducible Decomposition**
   - **witness sets**
   - cascades of homotopies
   - computational experiments
4. Parallel and Cloud Computing
   - speedup and quality up
   - web interfaces

numerical algebraic geometry

Introduced in 1995 as a pun on numerical linear algebra.

In numerical algebraic geometry, we apply homotopy continuation to compute positive dimensional solutions of polynomial systems.

Four homotopies compute a numerical irreducible decomposition:

1. Cascade homotopies compute generic points on all solution components, over all dimensions.
2. A homotopy membership test decides whether a given point belongs to a component of the solution set.
3. Monodromy loops factor pure dimensional solution sets into irreducible components.
4. A diagonal homotopy intersects solution sets.

The data structure to represent a solution set is a witness set:

1. a polynomial system augmented with random linear equations;
2. solutions of the augmented system are generic points.

an illustrative example

Consider the following polynomial system:

$$\left\{ \begin{array}{rcl} (x^2 + y^2 + z^2 - 1)(y - x^2)(x - 0.5) & = & 0 \\ (x^2 + y^2 + z^2 - 1)(z - x^3)(y - 0.5) & = & 0 \\ (x^2 + y^2 + z^2 - 1)(z - xy)(z - 0.5) & = & 0 \end{array} \right.$$

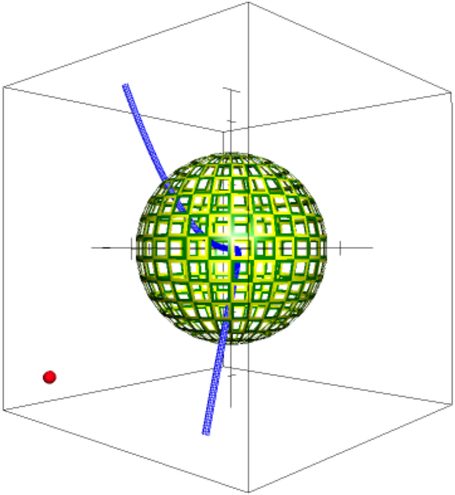
In factored form, we read off the solutions:

dimension = 2: the sphere,

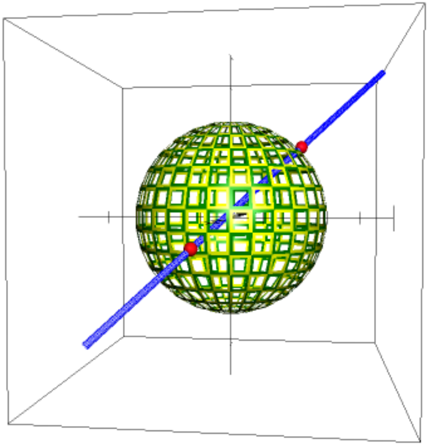
dimension = 1: the twisted cubic and three lines,

dimension = 0: one isolated point.

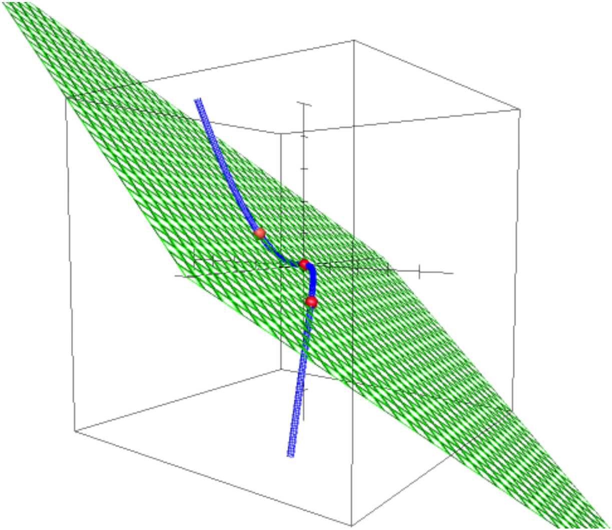
a sphere, the twisted cubic, an isolated point



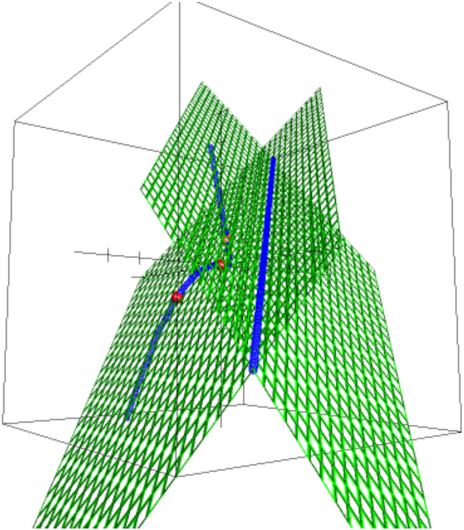
a witness set for the sphere



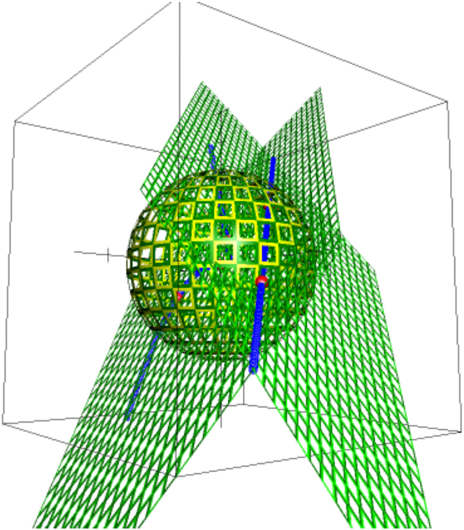
a witness set for the twisted cubic



a random line will miss the twisted cubic



a random line will intersect the sphere



Solving Polynomial Systems

witness sets

To compute the degree of the twisted cubic, consider

$$\mathcal{E}(\mathbf{f})(\mathbf{x}) = \left\{ \begin{array}{r} x_2 - x_1^2 = 0 \\ x_3 - x_1^3 = 0 \\ c_0 + c_1 x_1 + c_2 x_2 + c_3 x_3 = 0 \end{array} \right. \qquad c_0, c_1, c_2, c_3 \in \mathbb{C},$$

where c_0, c_1, c_2, and c_3 are random numbers.
The substitution $x_2 = x_1^2$ and $x_3 = x_1^3$ in the last equation shows that the degree of $\mathbf{f}^{-1}(\mathbf{0})$ equals three.

A *witness set* for a k-dimensional solution set consists of

- k hyperplanes with random coefficients; and
- the set of d isolated solutions on those hyperplanes.

Because the hyperplanes are random, all d isolated solutions are generic points and d is the degree of the set.

an example

Consider the system

$$\mathbf{f}(\mathbf{x}) = \left\{ \begin{array}{r} (x_1^2 - x_2)(x_1 - 0.5) = 0 \\ (x_1^3 - x_3)(x_2 - 0.5) = 0 \\ (x_1 x_2 - x_3)(x_3 - 0.5) = 0 \end{array} \right.$$

The solutions of the system $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ are

- the twisted cubic, a one dimensional solution set; and
- four isolated points.

Can we compute all solutions with one homotopy?

a cascade homotopy

To compute numerical representations of the twisted cubic and the four isolated points, use

$$\mathbf{h}(\mathbf{x}, z_1, t) = \left[\begin{array}{ccc} \left[\begin{array}{c} (x_1^2 - x_2)(x_1 - 0.5) \\ (x_1^3 - x_3)(x_2 - 0.5) \\ (x_1 x_2 - x_3)(x_3 - 0.5) \end{array} \right] & + & t \left[\begin{array}{c} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{array} \right] z_1 \\ t\,(c_0 + c_1 x_1 + c_2 x_2 + c_3 x_3) & + & z_1 \end{array} \right] = \mathbf{0}.$$

At $t = 1$: $\mathbf{h}(\mathbf{x}, z_1, t) = \mathcal{E}_1(\mathbf{f})(\mathbf{x}, z_1) = \mathbf{0}$.

At $t = 0$: $\mathbf{h}(\mathbf{x}, z_1, t) = \mathbf{f}(\mathbf{x}) = \mathbf{0}$.

As t goes from 1 to 0, the hyperplane is removed from the embedded system, and z_1 is forced to zero.

a superwitness set cascade

Summarizing the progress of the path tracking:

13 paths	$\longrightarrow$	0 paths to infinity		
		3 solutions with $z_1 = 0$	$\longrightarrow$	W_1 witness set
$\downarrow$		10 solutions with $z_1 \neq 0$		
10 paths	$\longrightarrow$	1 path to infinity		
		9 converging paths	$\longrightarrow$	$\widehat{W_0}$ witness superset

Starting with 13 paths of the embedded system,
the cascade produces three witness points for the cubic
and 9 points which may be isolated or lie on the cubic.

regularity results

Theorem (superwitness set generation)

For an embedding $\mathcal{E}_i(\mathbf{f})(\mathbf{x}, \mathbf{z})$ of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ with i random hyperplanes and i slack variables $\mathbf{z} = (z_1, z_2, \ldots, z_i)$, we have

1. *solutions with $\mathbf{z} = \mathbf{0}$ contain $\deg W$ generic points on every i-dimensional component W of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$;*
2. *solutions with $\mathbf{z} \neq \mathbf{0}$ are regular; and*
3. *the solution paths defined by the cascading homotopy starting at $t = 0$ with all solutions with $z_i \neq 0$ reach at $t = 1$ all isolated solutions of $\mathcal{E}_{i-1}(\mathbf{f})(\mathbf{x}, \mathbf{z}) = \mathbf{0}$.*

an algorithm

Input: $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ a polynomial system;
d the top dimension of $\mathbf{f}^{-1}(\mathbf{0})$.

Output: $\widehat{W} = [\widehat{W}_d, \widehat{W}_{d-1}, \ldots, \widehat{W}_0]$
super witness sets for all dimensions.

$V := \mathrm{Solve}(\mathcal{E}_d(\mathbf{f})(\mathbf{x}, \mathbf{z}) = \mathbf{0})$;
for k from d down to 1 do
    $\widehat{W}_k := \{ \ (\mathbf{x}, \mathbf{z}) \in V \mid \mathbf{z} = \mathbf{0} \ \}$;
    $V := \{ \ (\mathbf{x}, \mathbf{z}) \in V \mid z_k \neq 0 \ \}$;
    if $V = \emptyset$ then return $\widehat{W}$;
    else $\mathbf{h}(\mathbf{x}, \mathbf{z}, t) := (1-t)\mathcal{E}_k(\mathbf{f})(\mathbf{x}, \mathbf{z}) + t \left(\begin{array}{c} \mathcal{E}_{k-1}(\mathbf{f})(\mathbf{x}, \mathbf{z}) \\ z_k \end{array} \right)$;
        $V := \{ \ (\mathbf{x}, \mathbf{z}) \mid \mathbf{h}(\mathbf{x}, \mathbf{z}, 1) = \mathbf{0} \ \}$;
    end if;
end for;
$\widehat{W}_0 := \{ \ (\mathbf{x}, \mathbf{z}) \in V \mid \mathbf{z} = \mathbf{0} \ \}$.

Solving Polynomial Systems

solving cyclic 8- and cyclic 9-roots systems

Both cyclic 8 and cyclic 9-roots are relatively small problems.

Wall clock times in seconds with `phc -B -tp` for `p` threads:

	cyclic 8-roots		cyclic 9-roots	
p	seconds	speedup	seconds	speedup
1	181.765	1.00	2598.435	1.00
2	167.871	1.08	1779.939	1.46
4	89.713	2.03	901.424	2.88
8	47.644	3.82	427.800	6.07
16	32.215	5.65	267.838	9.70
32	22.182	8.19	153.353	16.94
64	20.103	9.04	150.734	17.24

With 64 threads, we can solve the cyclic 9-roots problem faster than solving the cyclic 8-roots problem on one thread.

running in double double and quad double precision

Double double and quad double arithmetic are implemented in QDlib, a software library by Y. Hida, X. S. Li, and D. H. Bailey, 2001.

In double precision, with 64 threads, the time

- for cyclic 8-roots reduces from 3 minutes to 20 seconds and
- for cyclic 9-roots from 43 minutes to 2 minutes and 30 seconds.

The wall clock times below are with 64 threads in higher precision.

	cyclic 8-roots			cyclic 9-roots		
	seconds	=	hms format	seconds	=	hms format
dd	53.042	=	53s	498.805	=	8m19s
qd	916.020	=	15m16s	4761.258	=	1h19m21s

With 64 threads, we can compensate for the computational overhead caused by double double precision and achieve *quality up*.

solving the cyclic 12-roots system in parallel

The wall clock time on the blackbox solver on one thread is about 95 hours (almost 4 days), which includes the linear-product bound.
The time reduces from 4 days to less than 3 hours with 64 threads:

	solving top system			cascade and filter			grand	
p	start	contin	total	cascade	filter	total	total	speedup
2	62813	47667	110803	44383	2331	46714	157518	1.00
4	21181	25105	46617	24913	1558	26471	73089	2.16
8	8933	14632	23896	13542	946	14488	38384	4.10
16	4656	7178	12129	6853	676	7529	19657	8.01
32	4200	3663	8094	3415	645	4060	12154	12.96
64	4422	2240	7003	2228	557	2805	9808	16.06

The solving of the top dimensional system breaks up in two stages:

- the solving of a start system (start) and the
- continuation to the solutions of the top dimensional system (contin).

Speedups are good in the cascade, but the filtering contains the factorization in irreducible components, which does not run in parallel.

running in double double precision

A run in double double precision with 64 threads ends after 7 hours and 37 minutes.

This time lies between the times in double precision

- with 8 threads, 10 hours and 39 minutes, and
- with 16 threads, 5 hours and 27 minutes.

Confusing quality with precision, from 8 to 64 threads, the working precision can be doubled, with a reduction in time by 3 hours, from 10.5 hours to 7.5 hours.

Solving Polynomial Systems

parallel granularity

Quality up: how much better are the results on p processors
if we can afford the same amount of wall clock time?

1. Distributed Memory Parallel Programming with Message Passing. Solutions paths are distributed in a manager/worker paradigm, using dynamic load balancing.
2. Shared Memory Parellel Programming with Multithreading. A number of threads collaborate in a work crew model:
   - $+$ granularity can be much finer, less overhead;
   - $-$ thread safety is a concern, careful with memory management.
3. Acceleration on Graphics Processing Units (GPU):
   - memory bound up to real double double arithmetic,
   - compute bound starts at complex double double arithmetic.

Solving Polynomial Systems

1. Introduction
   - polynomial systems
   - PHCpack and phcpy
2. Polynomial Homotopy Continuation
   - solving polynomial systems numerically
   - exploiting sparse structures
3. Numerical Irreducible Decomposition
   - witness sets
   - cascades of homotopies
   - computational experiments
4. Parallel and Cloud Computing
   - speedup and quality up
   - web interfaces

phcpy in a SageMath kernel of a Jupyter notebook

jupyter **code_snippet** Last Checkpoint: 4 minutes ago (autosaved)

File Edit View Insert Cell Kernel Help PHCpy

- solving trinomials ▸
- representations of isolated solutions ▸
- reproducible runs with fixed seeds ▸
- shared memory parallelism ▸
- root counting methods ▸
- Newton's method and deflation ▸
- equation and variable scaling ▸

- solving a random case
- solving a specific case

- ◂ solution se
- ◂ families of
- ◂ Schubert c
- ◂ Newton polytopes
- ◂ the extension module

```
f = ['x^2*y^2 + 2*x - 1;', 'x^2*y^2 - 3*y + 1;']
from phcpy.solver import solve
sols = solve(f)
for sol in sols: print sol
```

In [1]:
```
f = [
from
sols
for sol in sols: print sol
```

```
PHCv2.4.64 released 2019-01-21 works!
total degree : 16
2-homogeneous Bezout number : 8
  with with partition : { x }{ y }
general linear-product Bezout number : 8
  based on the set structure :
     { x }{ x }{ y }{ y }
     { x }{ x }{ y }{ y }
mixed volume : 4
stable mixed volume : 4
t :  1.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x :  4.86132470489966E-01   0.00000000000000E+00
 y :  3.42578353006690E-01  -2.28597478256455E-100
== err :  3.499E-17 = rco :  8.882E-01 = res :  4.857E-17 =
```

- Code snippets suggest typical applications, and guide the novice user.
- Solve polynomials by pointing and clicking.