Numerical Algebraic Geometry in the Cloud 2

Jan Verschelde
joint work with Nathan Bliss, Jasmine Otto, and Jeff Sommars

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

`www.phcpack.org`
or `https://pascal.math.uic.edu`

Computer Algebra in Scientific Computing (CASC) 2017
18 September 2017

supported in part by NSF ACI 1440534

Outline

1. introduction
   - numerical algebraic geometry
   - in the cloud
2. numerical irreducible decomposition
   - an illustrative example
   - witness sets, cascades, and membership test
   - factoring with linear traces and monodromy
   - a general solve command
3. tutorial
   - sign up and login
   - demonstration

Numerical Algebraic Geometry in the Cloud 2

1. introduction
   - numerical algebraic geometry
   - in the cloud
2. numerical irreducible decomposition
   - an illustrative example
   - witness sets, cascades, and membership test
   - factoring with linear traces and monodromy
   - a general solve command
3. tutorial
   - sign up and login
   - demonstration

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.

Numerical Algebraic Geometry in the Cloud 2

1. introduction
   - numerical algebraic geometry
   - in the cloud
2. numerical irreducible decomposition
   - an illustrative example
   - witness sets, cascades, and membership test
   - factoring with linear traces and monodromy
   - a general solve command
3. tutorial
   - sign up and login
   - demonstration

in the cloud

`www.phcpack.org` provides access to a Jupyter notebook (alternative site: `https://pascal.math.uic.edu`) with a SageMath 8.0 kernel, where phcpy is installed.

Code snippets are defined via Jupyter's notebook extensions:

- each snippet illustrates a particular feature of phcpy; and
- each snippet runs independently.

Users have actual accounts on the server:

- a terminal window to a Linux computer.
- Facilitates collaborations, sharing notebooks and data.

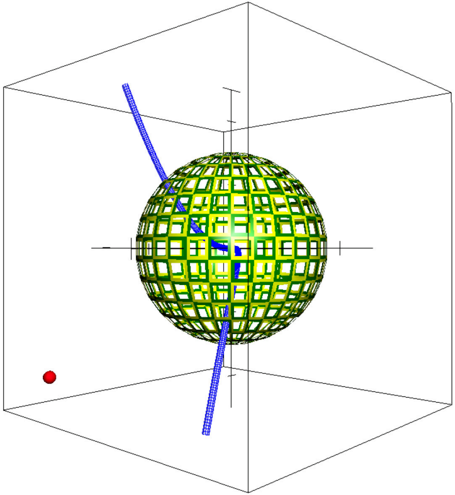
Numerical Algebraic Geometry in the Cloud 2

an illustrative example

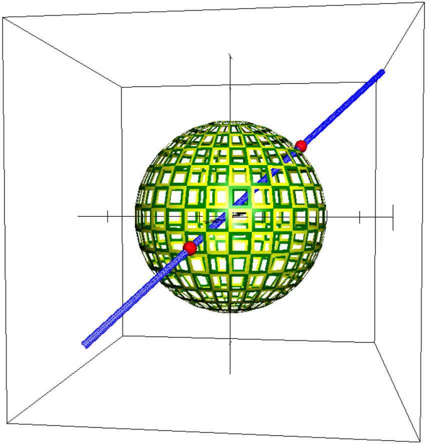
In the code snippets, select `solution sets`
→ `cascade of homotopies`
→ `an illustrative example`

```
pol1 = '(x^2 + y^2 + z^2 - 1)*(y - x^2)*(x - 0.5);'
pol2 = '(x^2 + y^2 + z^2 - 1)*(z - x^3)*(y - 0.5);'
pol3 = '(x^2 + y^2 + z^2 - 1)*(z - x*y)*(z - 0.5);'
pols = [pol1, pol2, pol3]
from phcpy.cascades import run_cascade
otp = run_cascade(3, 2, pols)
dims = otp.keys()
dims.sort(reverse=True)
for dim in dims:
    print 'number of solutions at dimension', \
        dim, ':', len(otp[dim][1])
```

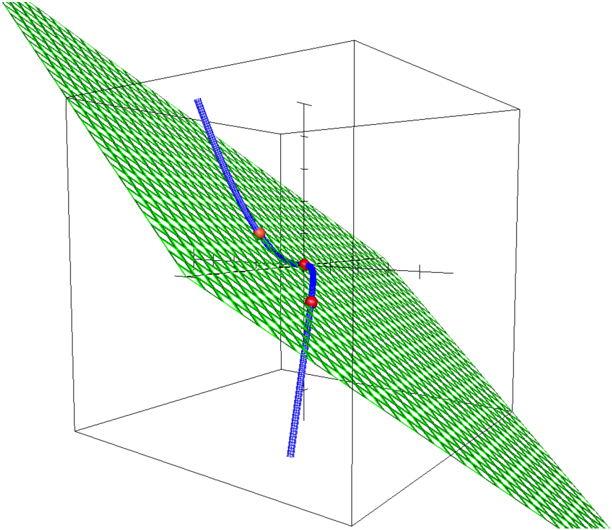
a sphere, the twisted cubic, an isolated point



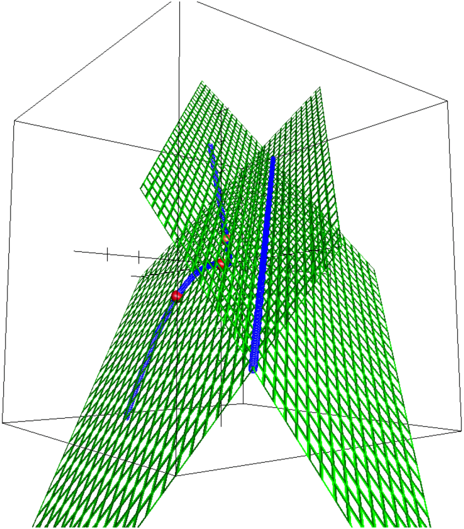
a witness set for the sphere



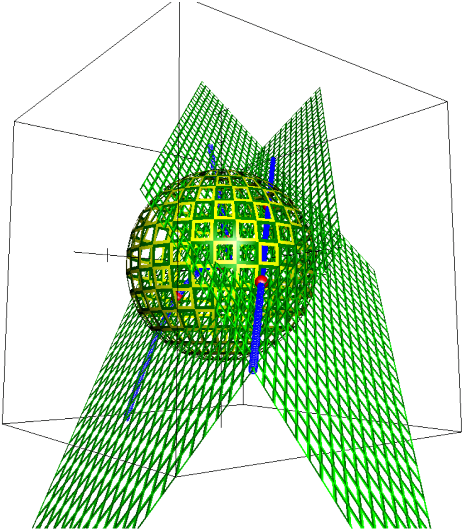
a witness set for the twisted cubic



a random line will miss the twisted cubic



a random line will intersect the sphere



Numerical Algebraic Geometry in the Cloud 2

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
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;
$\widehat{W}_0 := \{ (\mathbf{x}, \mathbf{z}) \in V \mid \mathbf{z} = \mathbf{0} \}$.

deciding membership

Given a witness set representation for a solution set, we can decide whether a point belongs to the solution set, via:

Algorithm HomotopyMembershipTest(W_L,$\mathbf{y}$)

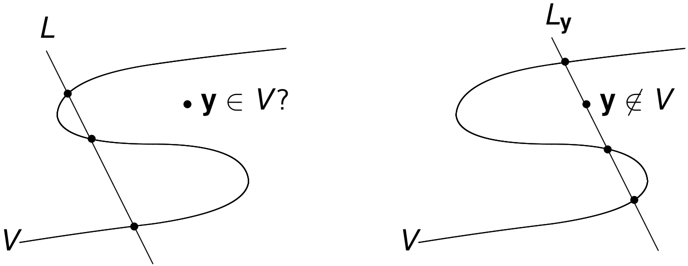
Input: W_L is witness set for a solution set;
$\mathbf{y}$ is any point in space.
Output: yes or no, depending whether $\mathbf{y}$ belongs to the set.

$$\mathbf{h}(\mathbf{x}, t) = (1 - t) \left(\begin{array}{c} \mathbf{f}(\mathbf{x}) = \mathbf{0} \\ L(\mathbf{x}) = \mathbf{0} \end{array} \right) + t \left(\begin{array}{c} \mathbf{f}(\mathbf{x}) = \mathbf{0} \\ L(\mathbf{x}) = L(\mathbf{y}) \end{array} \right) = \mathbf{0};$$

$V := \{ \mathbf{x} \mid \mathbf{h}(\mathbf{x}, 1) = \mathbf{0} \}$;
return $\mathbf{y} \in V$.

schematic membership

A curve V is represented by 3 witness points on L:



To decide whether $\mathbf{y} \in V$,
we create a new witness set for a line $L_\mathbf{y}$ through $\mathbf{y}$.

As $\mathbf{y} \not\in V \cap L_\mathbf{y}$, we conclude $\mathbf{y} \not\in V$.

Numerical Algebraic Geometry in the Cloud 2

the linear trace

Consider $f \in \mathbb{C}[x, y]$, $\deg(f) = 3$. Does f factor?

Assume f has a quadratic factor q.

We view $f \in \mathbb{C}[x][y]$ and write q as

$$\begin{aligned} q(x, y(x)) &= (y - y_1(x))(y - y_2(x)) \\ &= y^2 - (y_1(x) + y_2(x))y + y_1(x)y_2(x). \end{aligned}$$

Observe: if q is a quadratic factor of f,
then $y_1(x) + y_2(x)$ must be a linear function of x,
otherwise the degree of q would be higher than two.

Denote $t_1(x) = y_1(x) + y_2(x)$ and call t_1 the linear trace.

interpolating the linear trace

Fix $x = x_1$ and solve $f(x_1, y) = 0$ for y.

As $\deg(f) = 3$, we find three roots and write them as as $(x_1, y_1(x^*))$, $(x_1, y_2(x^*))$, and $(x_1, y_3(x^*))$.

If f has a quadratic factor q, its linear trace t_1 is $t_1(x) = y_1(x) + y_2(x) = ax + b$, for some $a, b \in \mathbb{C}$.

Take $x_2 \neq x_1$ and consider

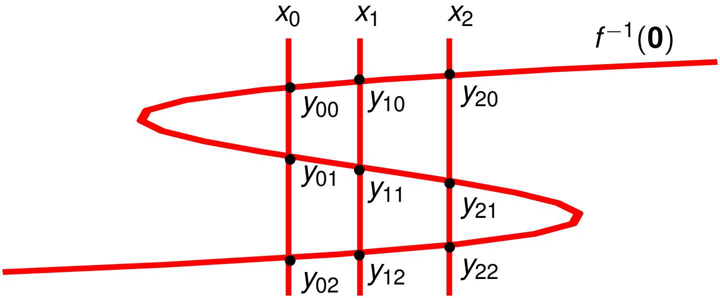
$$\left\{ \begin{array}{l} ax_1 + b = y_1(x_1) + y_2(x_1) \\ ax_2 + b = y_1(x_2) + y_2(x_2) \end{array} \right.$$

Solving the linear system for a and b determines $t_1(x)$.

Take a third sample set, at $x = x_3$ and test

$$t(x_3) = ax_3 + b \stackrel{?}{=} y_1(x_3) + y_2(x_3).$$

an example



Use $\{(x_0, y_{00}), (x_0, y_{01}), (x_0, y_{02})\}$ and $\{(x_1, y_{10}), (x_1, y_{11}), (x_1, y_{12})\}$ to find $t_1(x) = c_0 + c_1 x$.
At $\{(x_2, y_{20}), (x_2, y_{21}), (x_2, y_{22})\}$: $c_0 + c_1 x_2 = y_{20} + y_{21} + y_{22}$?

combinatorial enumeration

A linear trace test answers each question:

```
Is 1 a factor?
 |- Yes: Is 2 a factor?
 |    |- Yes: 1,2,3 is the factorization
 |    |- No: 1,23 is the factorization
 |- No: Is 12 a factor?
      |- Yes: 12,3 is the factorization
      |- No: is 13 a factor?
           |- Yes: 13,2 is the factorization
           |- No: 123 is the factorization
```

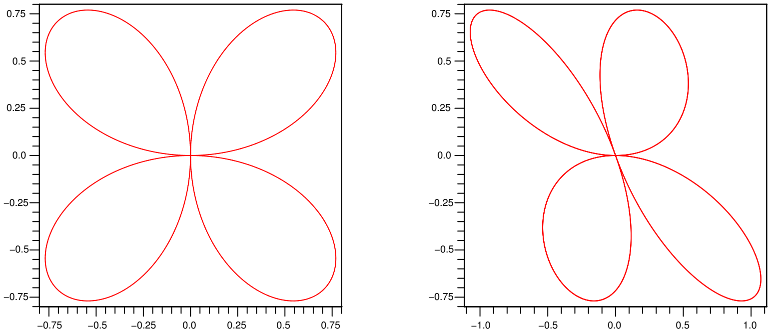
This combinatorial enumeration works for low degrees,
and is improved via LLL to solve the knapsack problem.

avoiding wrong factorizations

Consider $f(x, y) = (x^2 + y^2)^3 - 4x^2y^2 = 0$.

By symmetry: if $f(a, b) = 0$, then also $f(\pm a, \pm b) = 0$.

Pictures of $f(x, y) = 0$ and $f(x + \frac{1}{2}y, y) = 0$:



factoring witness sets

Consider $\left\{ \begin{array}{c} f(x,y) = 0 \\ c_0 + c_1 x + c_2 y = 0 \end{array} \right.$ for random c_0, c_1, and c_2.

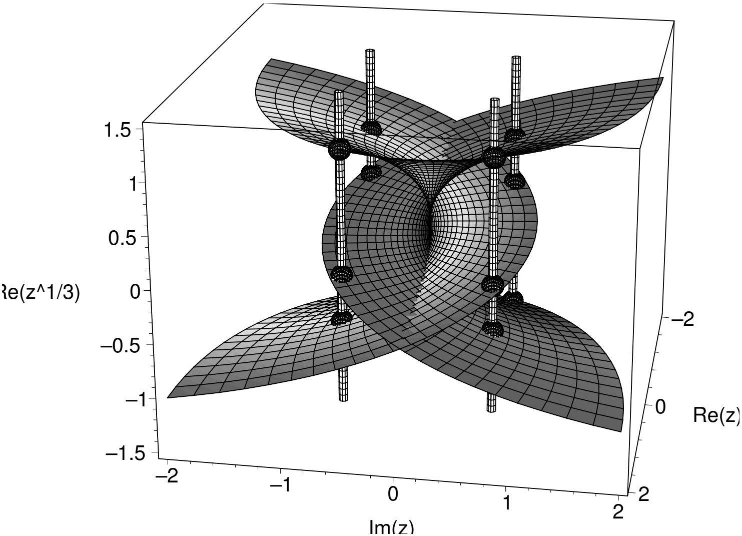
To sample points, we apply the coordinate transformation:

$$\phi : \mathbb{C}^2 \rightarrow \mathbb{C}^2 : \left[\begin{array}{c} x \\ y \end{array} \right] \mapsto \phi(x,y) = \left[\begin{array}{cc} -c_1 & -c_2 \\ 0 & 1 \end{array} \right] \left[\begin{array}{c} x \\ y \end{array} \right].$$

As the samples satisfy the equation $c_0 + c_1 x + c_2 y = 0$,
we have $\phi(x,y) = (c_0, y)$.

The coordinate transformation applies in any dimension.

$z^3 - w = 0$ as a Riemann surface



monodromy loops

Moving between witness sets:

$$\mathbf{h}_{KL}(\mathbf{x}, t) = \lambda \left(\begin{array}{c} \mathbf{f}(\mathbf{x}) \\ K(\mathbf{x}) \end{array} \right) (1 - t) + \left(\begin{array}{c} \mathbf{f}(\mathbf{x}) \\ L(\mathbf{x}) \end{array} \right) t = \mathbf{0}, \quad \lambda \in \mathbb{C},$$

we find new witness points on the hyperplanes $K(\mathbf{x}) = \mathbf{0}$, starting at those witness points satisfying $L(\mathbf{x}) = \mathbf{0}$,
letting t move from one to zero.

Choosing a random $\mu \neq \lambda$, we move back from K to L:

$$\mathbf{h}_{LK}(\mathbf{x}, t) = \mu \left(\begin{array}{c} \mathbf{f}(\mathbf{x}) \\ L(\mathbf{x}) \end{array} \right) (1 - t) + \left(\begin{array}{c} \mathbf{f}(\mathbf{x}) \\ K(\mathbf{x}) \end{array} \right) t = \mathbf{0}, \quad \mu \in \mathbb{C}.$$

After $\mathbf{h}_{KL}$ and $\mathbf{h}_{LK}$ we arrive at the same witness set.
Permuted points belong to the same irreducible component.

monodromy breakup algorithm

Input: W_L, d, N
Output: $\mathcal{P}$
0. initialize $\mathcal{P}$ with d singletons;
1. generate two slices L' and L'' parallel to the given L;
2. track d paths for witness set with L';
3. track d paths for witness set with L'';
4. **for** k **from** 1 **to** N **do**
    4.1 generate new slices K and a random λ;
    4.2 track d paths defined by $\mathbf{h}_{KL}$;
    4.3 generate a random μ;
    4.4 track d paths defined by $\mathbf{h}_{LK}$;
    4.5 compute the permutation and update $\mathcal{P}$;
    4.6 **if** linear trace test certifies $\mathcal{P}$
        then leave the loop;
        end if;
    end for.

Numerical Algebraic Geometry in the Cloud 2

a general solve command

In the code snippets, select `solution sets`

→ `numerical irreducible decomposition`

→ `an example`

```
pol0 = '(x1-1)*(x1-2)*(x1-3)*(x1-4);'
pol1 = '(x1-1)*(x2-1)*(x2-2)*(x2-3);'
pol2 = '(x1-1)*(x1-2)*(x3-1)*(x3-2);'
pol3 = '(x1-1)*(x2-1)*(x3-1)*(x4-1);'
pols = [pol0, pol1, pol2, pol3]
from phcpy.factor import solve, write_decomposition
deco = solve(4, 3, pols, verbose=False)
write_decomposition(deco)
```

To get the witness set at dimension one:

```
(witpols, witsols, dim) = deco[1]
print len(witsols)
```

Numerical Algebraic Geometry in the Cloud 2

sign up and login, at `www.phcpack.org`

The sign up procedure requires a functional email address.

Two steps in obtaining an account:

1. Visit `www.phcpack.org` and fill out a form.
   `www.phcpack.org` redirects to
   `https://pascal.math.uic.edu`.
2. Click on the link sent in the email to your email address.

Two kernels offer `phcpy`, do `import phcpy` in both:

1. python 2 (the code snippets work for version 2 of python).
2. SageMath uses python 2 as the scripting language.

Select the kernel from the `new` menu in the upper right.

Numerical Algebraic Geometry in the Cloud 2

demonstration

jupyter gensolvesnippet Last Checkpoint: a few seconds ago (autosaved) Logout

File Edit View Insert Cell Kernel Help PHCpy SageMath 8.0

Code

- blackbox solver
- path trackers
- solution sets
- families of systems
- Schubert calculus
- Newton polytopes
- the extension module

In [3]:
```
pol0 = '(x1-1)*(x1-2)*(x1-3)*(x1-4)
pol1 = '(x1-1)*(x2-1)*(x2-2)*(x2-3)
pol2 = '(x1-1)*(x1-2)*(x3-1)*(x3-2)
pol3 = '(x1-1)*(x2-1)*(x3-1)*(x4-1)
pols = [pol0, pol1, pol2, pol3]
from phcpy.factor import solve, wri
deco = solve(4, 3, pols, verbose=False)
write_decomposition(deco)
```

```
the factorization at dimension 3  #components : 1
[([1], 3.2057689836051395e-14)]
the factorization at dimension 2  #components : 1
[([1], 7.549516567451064e-15)]
the factorization at dimension 1  #components : 12
[([1], 3.6637359812630166e-15), ([2], 1.4210854715202004e-14), ([3], 2.4424906541753444e-15),
([4], 1.1324274851176597e-14), ([5], 2.1094237467877974e-15), ([6], 3.3861802251067274e-15),
([7], 1.4654943925052066e-14), ([8], 3.4416913763379853e-15), ([9], 2.609024107869118e-15), (
[10], 3.552713678800501e-15), ([11], 3.941291737419306e-15), ([12], 1.1879386363489175e-14)]
the number of isolated solutions : 4
```

In [2]:
```
(witpols, witsols, dim) = deco[1]
print len(witsols)
```

```
12
```