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
introduction
- numerical algebraic geometry
- in the cloud

numerical irreducible decomposition
- an illustrative example
- witness sets, cascades, and membership test
- factoring with linear traces and monodromy
- a general solve command

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
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

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
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, ':
    print 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
introduction
- numerical algebraic geometry
- in the cloud

numerical irreducible decomposition
- an illustrative example
- witness sets, cascades, and membership test
- factoring with linear traces and monodromy
- a general solve command

tutorial
- sign up and login
- demonstration
witness sets

To compute the degree of the twisted cubic, consider

\[ \mathcal{E}(f)(x) = \begin{cases} 
    x_2 - x_1^2 = 0 \\
    x_3 - x_1^3 = 0 \\
    c_0 + c_1 x_1 + c_2 x_2 + c_3 x_3 = 0 
\end{cases} \quad c_0, c_1, c_2, c_3 \in \mathbb{C}, \]

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 \( f^{-1}(0) \) 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.
Consider the system

\[ f(x) = \begin{cases} 
(x_1^2 - x_2)(x_1 - 0.5) = 0 \\
(x_1^3 - x_3)(x_2 - 0.5) = 0 \\
(x_1 x_2 - x_3)(x_3 - 0.5) = 0 
\end{cases} \]

The solutions of the system \( f(x) = 0 \) 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

\[
\begin{align*}
\mathbf{h}(\mathbf{x}, z_1, t) &= \begin{bmatrix}
(x_1^2 - x_2)(x_1 - 0.5) \\
(x_1^3 - x_3)(x_2 - 0.5) \\
(x_1 x_2 - x_3)(x_3 - 0.5) \\
t (c_0 + c_1 x_1 + c_2 x_2 + c_3 x_3)
\end{bmatrix} + t \begin{bmatrix}
\gamma_1 \\
\gamma_2 \\
\gamma_3 \\
z_1
\end{bmatrix} z_1 = 0.
\end{align*}
\]

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

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

As \( t \) goes from 1 to 0, the hyperplane is removed from the embedded system, and \( z_1 \) is forced to zero.
Summarizing the progress of the path tracking:

<table>
<thead>
<tr>
<th>13 paths</th>
<th>0 paths to infinity</th>
<th>$\mathcal{W}_1$ witness set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3 solutions with $z_1 = 0$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10 solutions with $z_1 \neq 0$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>10 paths</th>
<th>1 path to infinity</th>
<th>$\hat{\mathcal{W}}_0$ witness superset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>9 converging paths</td>
<td></td>
</tr>
</tbody>
</table>

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 $\mathcal{E}_i(f)(\mathbf{x}, \mathbf{z})$ of $f(\mathbf{x}) = 0$ 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 $f(\mathbf{x}) = 0$;
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 $\mathcal{E}_{i-1}(f)(\mathbf{x}, \mathbf{z}) = 0$. 

Jan Verschelde (UIC) Numerical Algebraic Geometry in the Cloud 2 CASC 2017, 18 September 19 / 38
an algorithm

Input: $f(x) = 0$ a polynomial system;
    $d$ the top dimension of $f^{-1}(0)$.
Output: $\hat{W} = [\hat{W}_d, \hat{W}_{d-1}, \ldots, \hat{W}_0]$
    super witness sets for all dimensions.

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

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

$$h(x, t) = (1 - t) \begin{pmatrix} f(x) = 0 \\ L(x) = 0 \end{pmatrix} + t \begin{pmatrix} f(x) = 0 \\ L(x) = L(y) \end{pmatrix} = 0;$$

$$\mathcal{V} := \{ x \mid h(x, 1) = 0 \};$$
return $y \in \mathcal{V}$. 
A curve $V$ is represented by 3 witness points on $L$:

To decide whether $y \in V$, we create a new witness set for a line $L_y$ through $y$.

As $y \notin V \cap L_y$, we conclude $y \notin V$. 

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
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

\[
q(x, y(x)) = (y - y_1(x))(y - y_2(x))
\]
\[
= y^2 - (y_1(x) + y_2(x))y + y_1(x)y_2(x).
\]

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 \( (x_1, y_1(x^*)), (x_1, y_2(x^*)), \) and \( (x_1, y_3(x^*)) \).

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

Take \( x_2 \neq x_1 \) and consider
\[
\begin{align*}
ax_1 + b &= y_1(x_1) + y_2(x_1) \\
ax_2 + b &= y_1(x_2) + y_2(x_2)
\end{align*}
\]

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

Take a third sample set, at \( x = x_3 \) and test
\[
t(x_3) = ax_3 + b = y_1(x_3) + y_2(x_3).
\]
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}?
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 $f(x + \frac{1}{2}y, y) = 0$: 
Consider \( \begin{cases} f(x, y) = 0 \\ c_0 + c_1 x + c_2 y = 0 \end{cases} \) for random \( c_0, c_1, \) and \( c_2 \).

To sample points, we apply the coordinate transformation:

\[
\phi : \mathbb{C}^2 \to \mathbb{C}^2 : \begin{bmatrix} x \\ y \end{bmatrix} \mapsto \phi(x, y) = \begin{bmatrix} -c_1 & -c_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.
\]

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:

\[ h_{KL}(x, t) = \lambda \begin{pmatrix} f(x) \\ K(x) \end{pmatrix} (1 - t) + \begin{pmatrix} f(x) \\ L(x) \end{pmatrix} t = 0, \quad \lambda \in \mathbb{C}, \]

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

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

\[ h_{LK}(x, t) = \mu \begin{pmatrix} f(x) \\ L(x) \end{pmatrix} (1 - t) + \begin{pmatrix} f(x) \\ K(x) \end{pmatrix} t = 0, \quad \mu \in \mathbb{C}. \]

After \( h_{KL} \) and \( h_{LK} \) we arrive at the same witness set. Permutated 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 $\lambda$;
   4.2 track $d$ paths defined by $h_{KL}$;
   4.3 generate a random $\mu$;
   4.4 track $d$ paths defined by $h_{LK}$;
   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

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
a general solve command

In the code snippets, select solution sets
→ numerical irreducible decomposition
→ an example

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

The sign up procedure requires a functional email address.

Two steps in obtaining an account:


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

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 [3]:

doctest.__doc__

```python
pol0 = '(x1-1)*(x1-2)*(x1-3)*(x1-4)
pol1 = '(x1-1)*(x2-1)*(x2-2)*(x2-3)
pol2 = '(x1-1)*(x2-2)*(x2-3)*(x3-2)
pol3 = '(x1-1)*(x2-1)*(x2-3)*(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)
```

the factorization at dimension 3  #components : 1

```plaintext
[(1), 3.2057689836051395e-14]
```

the factorization at dimension 2  #components : 1

```plaintext
[(1), 7.549516557451064e-15]
```

the factorization at dimension 1  #components : 12

```plaintext
[(1), 3.6637359812630166e-15], [(2), 1.4210854715202004e-14], [(3), 2.4424906541753444e-15],
[(4), 1.1324274851176598e-14], [(5), 2.1094237467877974e-15], [(6), 3.3861802251067274e-15],
[(7), 1.4654943925052066e-14], [(8), 3.4416913763379835e-15], [(9), 2.609024107869118e-15],
[(10), 3.55271367800501e-15], [(11), 3.941291737419306e-15], [(12), 1.1879386363489175e-14]
```

the number of isolated solutions : 4

In [2]:

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

12