Numerical Algebraic Geometry in the Cloud 1

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
   - jupyterhub, SageMath, and phcpy
2. polynomial homotopy continuation
   - solving polynomial systems numerically
   - polyhedral root counts
   - polyhedral homotopies
3. tutorial
   - sign up and login
   - demonstration

Numerical Algebraic Geometry in the Cloud 1

1. introduction
   - numerical algebraic geometry
   - jupyterhub, SageMath, and phcpy
2. polynomial homotopy continuation
   - solving polynomial systems numerically
   - polyhedral root counts
   - polyhedral homotopies
3. tutorial
   - sign up and login
   - demonstration

numerical algebraic geometry

Introduced in 1995 as a pun on numerical linear algebra.
In numerical algebraic geometry, one applies methods of numerical analysis to solve problems in computational algebraic geometry.

Benefits of numerical analysis:

- classical field in computer science,
- computers are still oriented towards floating-point arithmetic,
- widely accepted practice in scientific computing.

To use numerical analysis, one must know its problems:

- representation errors in the input: $1/10 \neq 0.1$,
- roundoff errors propagate, especially for larger problems,
- ill-conditioned problems need reformulating (computer algebra).

For computational algebraic geometry:

- numerical methods work best for complex analysis,
- not so good if exact answers are required.

cloud computing

Three disruptive trends in computational science:

1. Increased use of framework scripting languages such as Python.
2. Web interfaces, remote execution and data curation.
3. Grapics Processing Units (GPUs) accelerate computations at a low cost, enabling teraflops speeds.

In solving a problem computationally, we then ask:

- Reproducible? If you did it, can you do it again?
- Accessible? If you can do it, can I do it too?
- Scalable? Do I have to wait long?
- ...

Our CASC 2015 paper (with Nathan Bliss, Jeff Sommars, and Xiangcheng Yu) presented a dedicated interface to a numerical solver.

For this tutorial, we run phcpy in SageMath, with jupyterhub.

Numerical Algebraic Geometry in the Cloud 1

1. introduction
   - numerical algebraic geometry
   - jupyterhub, SageMath, and phcpy
2. polynomial homotopy continuation
   - solving polynomial systems numerically
   - polyhedral root counts
   - polyhedral homotopies
3. tutorial
   - sign up and login
   - demonstration

jupyter and jupyterhub

Jupyter (Julia+Python+R) is an open source web application

- to create and share documents that contain live code, equations, visualizations, and explantory text.
- Its uses are in data cleaning and transformation, numerical simulation, statistical modeling, machine learning, etc.

IPython is an earlier notebook system, started in late 2001, by Fernando Perez.

JupyterHub is a multi-user Hub which spawns, manages and proxies multiple instances of the single-user Jupyter notebook server.
Three subsystems make up JupyterHub:

1. a multi-user Hub, tornado process
2. a configurable http proxy, node-http-proxy
3. multiple single-user Jupyter notebook servers, Python/IPython/tornado.

SageMath and python

lead developer: William Stein

SAGE: Software for Algebra and Geometry Experimentation, started in early 2005, now known as SageMath.

SageMath is free and open source mathematical software

- its mission is to create a viable alternative to the big M's,
- builds on top of many free open source packages,
- has a large community of developers and users,
- its scripting language is based on Python,
- in the cloud: CoCalc and SageMathCell.

SageMath has its own notebook interface,
but its provides a kernel for the Jupyter notebook.

PHCpack and phcpy

PHCpack is a free and open source software for Polynomial Homotopy Continuation to solve polynomial systems.
ACM TOMS archived version 1.0 as Algorithm 795 in 1999.

- `phc -b` computes all isolated solutions of a polynomial system.
- `phc -c` computes all positive dimensional solution sets of a system in a numerical irreducible decomposition.

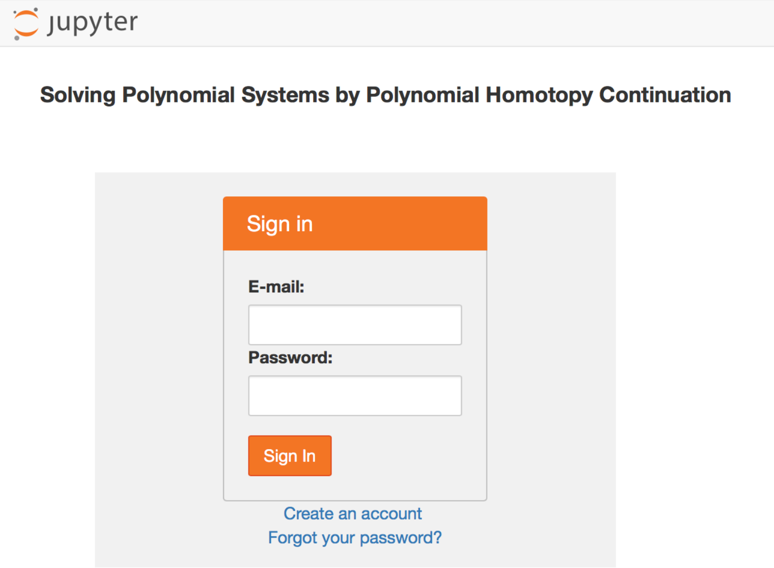
External software packages integrated in PHCpack:

- Fast mixed volume computation by MixedVol of Gao, Li, and Wu, Algorithm 846 of ACM TOMS, vol. 31, pages 555–560, 2005.
- Double double and quad double arithmetic with the QD Library of Hida, Li, and Bailey published in the 15th IEEE Symposium on Computer Arithmetic, pages 155–162. IEEE, 2001.

phcpy is a Python package which gives the Python programmer access to all functionality of PHCpack (EuroSciPy 2013 poster).

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

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



code snippets for phcpy

jupyter bbsolvesnippet

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 [2]:

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

```
total degree : 9
2-homogeneous Bezout number : 7
  with with partition : { x }{ y }
general linear-product Bezout number : 7
  based on the set structure :
     { x }{ y }{ y }
     { x y }{ x }{ x }
mixed volume : 7
stable mixed volume : 7
t :  1.00000000000000E+00   7.13177119756522E+00
m : 1
the solution for t :
 x : -1.14928524947248E+00  -4.33149270057445E-01
 y :  1.28839810793789E-01  -1.63511747105322E+00
== err :  1.650E-16 = rco :  3.038E-01 = res :  2.220E-16
=
t :  1.00000000000000E+00   2.72148344088863E+00
```

Numerical Algebraic Geometry in the Cloud 1

1. introduction
   - numerical algebraic geometry
   - jupyterhub, SageMath, and phcpy
2. polynomial homotopy continuation
   - solving polynomial systems numerically
   - polyhedral root counts
   - polyhedral homotopies
3. tutorial
   - sign up and login
   - demonstration

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.

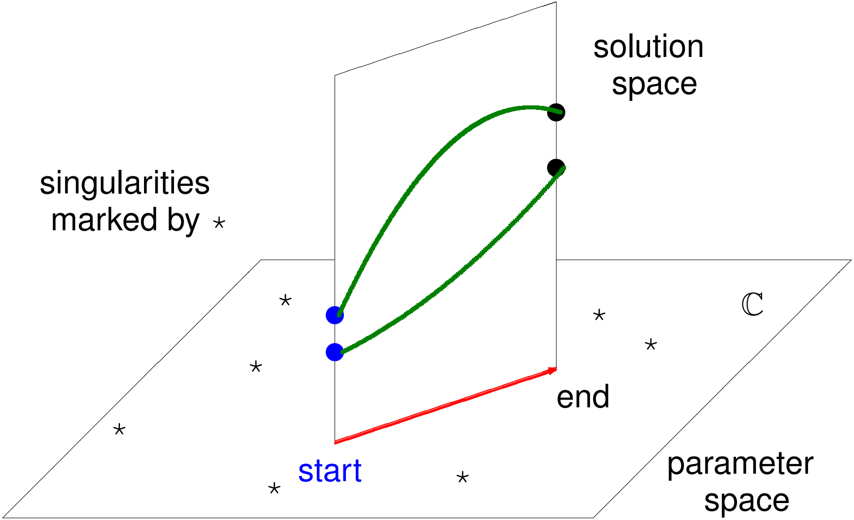
A polynomial in several variables consist of

- 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 a prior 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.

Numerical Algebraic Geometry in the Cloud 1

1. introduction
   - numerical algebraic geometry
   - jupyterhub, SageMath, and phcpy
2. polynomial homotopy continuation
   - solving polynomial systems numerically
   - polyhedral root counts
   - polyhedral homotopies
3. tutorial
   - sign up and login
   - demonstration

Newton polygons

Consider the system

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

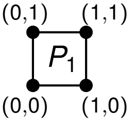
The coefficients $c_{(i,j)}$ are nonzero.

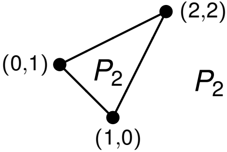
The polynomials in **f** have support sets A_1 and A_2:

$$A_1 = \{(1,1),(1,0),(0,1),(0,0)\}, \quad A_2 = \{(2,2),(1,0),(0,1)\}.$$

The convex hull of the supports span the Newton polygons of **f**:

$P_1 = \mathrm{conv}(A_1)$





$P_2 = \mathrm{conv}(A_2)$

Polyhedral Root Counting

$$f_i(\mathbf{x}) = \sum_{\mathbf{a} \in A_i} c_{i\mathbf{a}} \mathbf{x}^{\mathbf{a}}$$

$$c_{i\mathbf{a}} \in \mathbb{C}^* = \mathbb{C} \setminus \{0\}$$

$$\mathbf{f} = (f_1, f_2, \ldots, f_n)$$

$$P_i = \mathrm{conv}(A_i)$$

Newton polytope

$$\mathcal{P} = (P_1, P_2, \ldots, P_n)$$

$L(\mathbf{f})$ root count in $(\mathbb{C}^*)^n$	$V(\mathcal{P})$ mixed volume
$L(\mathbf{f}) = L(f_2, f_1, \ldots, f_n)$	$V(P_2, P_1, \ldots, P_n) = V(\mathcal{P})$
$L(\mathbf{f}) = L(f_1 \mathbf{x}^{\mathbf{a}}, \ldots, f_n)$	$V(P_1 + \mathbf{a}, \ldots, P_n) = V(\mathcal{P})$
$L(\mathbf{f}) \leq L(f_1 + \mathbf{x}^{\mathbf{a}}, \ldots, f_n)$	$V(\mathrm{conv}(P_1 + \mathbf{a}), \ldots, P_n) \geq V(\mathcal{P})$
$L(\mathbf{f}) = L(f_1(\mathbf{x}^{U\mathbf{a}}), \ldots, f_n(\mathbf{x}^{U\mathbf{a}}))$	$V(UP_1, \ldots, UP_n) = V(\mathcal{P})$
$L(f_{11} f_{12}, \ldots, f_n)$ $= L(f_{11}, \ldots, f_n) + L(f_{12}, \ldots, f_n)$	$V(P_{11} + P_{12}, \ldots, P_n)$ $= V(P_{11}, \ldots, P_n) + V(P_{12}, \ldots, P_n)$

exploit sparsity $\boxed{L(\mathbf{f}) = V(\mathcal{P})}$ 1st theorem of Bernshteĭn

mixed volumes

Theorem (Minkowski)

Let $P_1, P_2, \ldots, P_n$ be polytopes in $\mathbb{R}^n$. For a corresponding sequence of real numbers $\lambda_1, \lambda_2, \ldots, \lambda_n$, the volume of $\lambda_1 P_1 + \lambda_2 P_2 + \cdots + \lambda_n P_n$ is a homogeneous polynomial in $\lambda_1, \lambda_2, \ldots, \lambda_n$ of degree n. In particular:

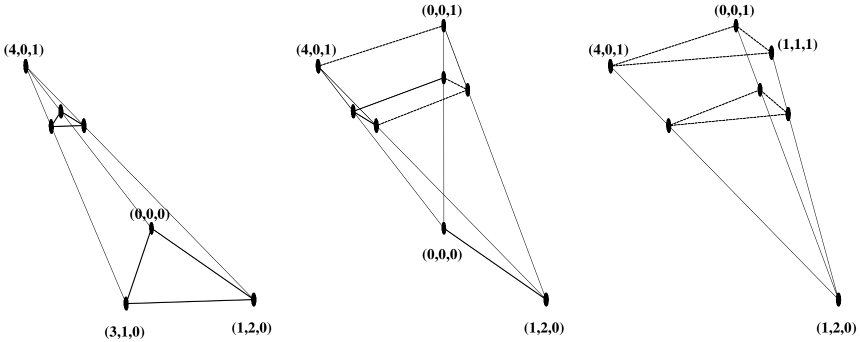
$$\mathrm{vol}\left(\sum_{i=1} \lambda_i P_i \right) = \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \cdots \sum_{i_n=1}^{n} \lambda_{i_1} \lambda_{i_2} \cdots \lambda_{i_n} V(P_{i_1}, P_{i_2}, \ldots, P_{i_n}).$$

The coefficient $V(P_{i_1}, P_{i_2}, \ldots, P_{i_n})$ is the mixed volume of the tuple $(P_{i_1}, P_{i_2}, \ldots, P_{i_n})$.

an example of mixing areas in a polytope

Consider $A_1 = \{(3,1),(1,2),(0,0)\}$ and $A_2 = \{(4,0),(1,1),(0,0)\}$.
The Cayley polytope of A_1 and A_2 is the convex hull of

$$\{\underbrace{(3,1,0),(1,2,0),(0,0,0)}_{A_1\times\{0\}},\underbrace{(4,0,1),(1,1,1),(0,0,1)}_{A_2\times\{1\}}\}.$$



The middle simplex is mixed, other simplices are unmixed.

Initial Forms

An *initial form* of a polynomial consists of those terms of the polynomial supported on a face of its Newton polytope.
We denote the inner product of two vectors by $\langle \cdot, \cdot \rangle$.

Definition (Initial Form)

Let $\mathbf{v}$ be a direction vector. Consider $f = \sum_{\mathbf{a} \in A} c_{\mathbf{a}} \mathbf{x}^{\mathbf{a}}$.

The ***initial form of f in the direction $\mathbf{v}$*** is

$$\mathrm{in}_{\mathbf{v}}(f) = \sum_{\substack{\mathbf{a} \in A \\ \langle \mathbf{a}, \mathbf{v} \rangle = m}} c_{\mathbf{a}} \mathbf{x}^{\mathbf{a}},$$

where $m = \min\{ \langle \mathbf{a}, \mathbf{v} \rangle \mid \mathbf{a} \in A \}$.

Bernshteĭn's Second Theorem 1975

Theorem (Bernshteĭn Theorem B 1975)

Consider $f(\mathbf{x}) = \mathbf{0}$, $\mathbf{f} = (f_1, f_2, \ldots, f_n)$, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$.
Denote by $\mathcal{P}$ *the tuple of Newton polytopes of* $\mathbf{f}$ *and* $\mathbb{C}^* = \mathbb{C} \setminus \{0\}$.
If for all nonzero $\mathbf{v}$: $\mathrm{in}_\mathbf{v}(\mathbf{f})(\mathbf{x}) = \mathbf{0}$ *has no solutions in* $(\mathbb{C}^*)^n$,
then $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ *has exactly as many isolated solutions in* $(\mathbb{C}^*)^n$
as the mixed volume of $\mathcal{P}$.

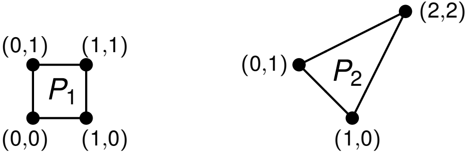
- The second theorem helps to prove the first theorem, using a homotopy deformation argument.
- Directions of diverging paths coincide with the normals $\mathbf{v}$ which define the $\mathrm{in}_\mathbf{v}(\mathbf{f})$ and are computed by polyhedral end games.
- The apriori computation of all normals which lead to an initial form system which may have solutions gives the *tropical prevariety*.

polytopes 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 $(\mathbb{C}^*)^2$.

Numerical Algebraic Geometry in the Cloud 1

1 introduction
- numerical algebraic geometry
- jupyterhub, SageMath, and phcpy

2 polynomial homotopy continuation
- solving polynomial systems numerically
- polyhedral root counts
- polyhedral homotopies

3 tutorial
- sign up and login
- demonstration

polyhedral homotopies

We first solve $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ via polyhedral homotopies.

To the supports of $\mathbf{g}$ we apply a lifting function $\omega = (\omega_1, \omega_2, \ldots, \omega_n)$, $\omega_i : A_i \rightarrow \mathbb{Z} : \mathbf{a} \mapsto \omega_i(\mathbf{a})$.

This leads to the system $\widehat{\mathbf{g}}(\mathbf{x}, t)$ with equations

$$\widehat{g}_i(\mathbf{x}, t) = \sum_{\mathbf{a} \in A_i} \bar{c}_{i\mathbf{a}} \mathbf{x}^{\mathbf{a}} t^{\omega_i(\mathbf{a})}, \quad \bar{c}_{i\mathbf{a}} \in \mathbb{C}^*,$$

where the coefficients $\bar{c}_{i\mathbf{a}}$ are random complex numbers.

To solve $\widehat{\mathbf{g}}(\mathbf{x}, t) = \mathbf{0}$, we look for inner normals $\mathbf{v} = (\mathbf{u}, 1)$ for which the corresponding initial form system $\mathrm{in}_{\mathbf{v}}\widehat{\mathbf{g}}(\mathbf{x}, t) = \mathbf{0}$ has a solution in $(\mathbb{C}^*)^n$.

changing coordinates

Changing coordinates $x_j = y_j t^{v_j}$, $j = 1, 2, \ldots, n$, then

$$\begin{aligned} \widehat{g}_i(\mathbf{y}, t) &= \sum_{\mathbf{a} \in A_i} \bar{c}_{i\mathbf{a}} \left(y_1 t^{v_1} y_2 t^{v_2} \cdots y_n t^{v_n} \right)^{\mathbf{a}} t^{\omega_i(\mathbf{a})} \\ &= \sum_{\mathbf{a} \in A_i} \bar{c}_{i\mathbf{a}} \mathbf{y}^{\mathbf{a}} t^{v_1 a_1 + v_2 a_2 + \cdots + v_n a_n + \omega_i(\mathbf{a})} \\ &= \sum_{\mathbf{a} \in A_i} \bar{c}_{i\mathbf{a}} \mathbf{y}^{\mathbf{a}} t^{\langle \mathbf{a}, \mathbf{u} \rangle + \omega_i(\mathbf{a})}. \end{aligned}$$

As $\mathbf{v}$ determines the coordinates change, denote $\widehat{g}_{\mathbf{v}}(\mathbf{y}, t) = \widehat{g}_i(x_j = y_j t^{v_j})$, and $m_i = \min_{\widehat{\mathbf{a}} \in A_i} \langle \mathbf{a}, \mathbf{v} \rangle$, so: monomials of $\widehat{g}_{\mathbf{v}}$ with lowest exponent m_i belong to $\mathrm{in}_{\mathbf{v}} \widehat{g}_{\mathbf{v}}$.

Thus $(t^{-m_i} \widehat{g}_{\mathbf{v},i})(\mathbf{y}, 0) = \mathrm{in}_{\mathbf{v}}(t^{-m_i} \widehat{g}_{\mathbf{v},i})(\mathbf{y})$.
Initial forms of $\widehat{g}(\mathbf{x}, t)$ are start systems.
The $\mathbf{v}$ are the leading powers of Puiseux series.

a totem pole of homotopies

Coefficient-Parameter

$\cup$

Mixed Volume

$\cup$

Linear Products

$\cup$

Multihomogeneous

$\cup$

Total Degree

easier start system ↓

more efficient (fewer paths) ↑

Numerical Algebraic Geometry in the Cloud 1

1. introduction
   - numerical algebraic geometry
   - jupyterhub, SageMath, and phcpy
2. polynomial homotopy continuation
   - solving polynomial systems numerically
   - polyhedral root counts
   - polyhedral homotopies
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:

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 1

1. introduction
   - numerical algebraic geometry
   - jupyterhub, SageMath, and phcpy
2. polynomial homotopy continuation
   - solving polynomial systems numerically
   - polyhedral root counts
   - polyhedral homotopies
3. tutorial
   - sign up and login
   - demonstration

demonstration