University of Illinois at Chicago

modernizing PHCpack through phcpy

Jan Verschelde
email: `jan@math.uic.edu`

EuroSciPy 2013
Brussels, August 21-24

0. The Mathematical Problem

Solve $f(\mathbf{x}) = \mathbf{0}$, a polynomial system in several variables $\mathbf{x}$.
A homotopy connects $f(\mathbf{x}) = \mathbf{0}$ to a system $g(\mathbf{x}) = \mathbf{0}$ with known solutions:

$$h(\mathbf{x}, t) = \gamma(1-t)g(\mathbf{x}) + tf(\mathbf{x}) = \mathbf{0}, \quad \gamma \in \mathbb{C}.$$

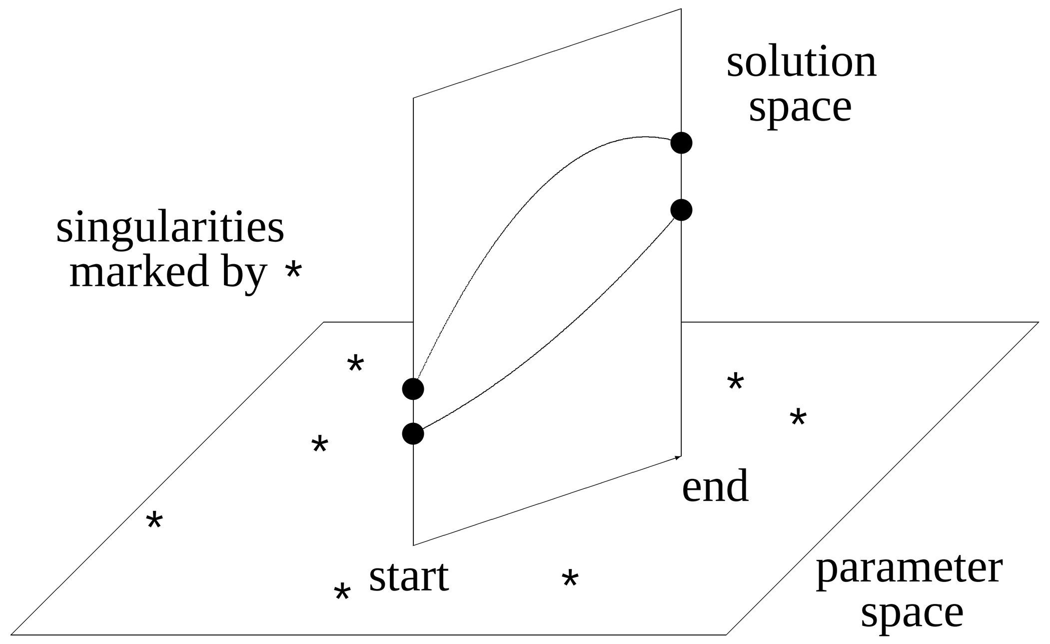
For almost all values for γ, the solutions of $h(\mathbf{x}, t) = \mathbf{0}$ are regular for all $t \in [0, 1)$.
Numerical continuation methods track solution paths defined by $h(\mathbf{x}, t) = \mathbf{0}$.

1. Parameter Continuation

Solve $f(\boldsymbol{\lambda}, \mathbf{x}) = \mathbf{0}$ first for generic values of the parameters $\boldsymbol{\lambda} = \boldsymbol{\lambda}_0$ and then use

$$h(\mathbf{x}, t) = (1-t)f(\boldsymbol{\lambda}_0, \mathbf{x}) + tf(\boldsymbol{\lambda}_1, \mathbf{x}) = \mathbf{0},$$

to solve a specific instance $f(\boldsymbol{\lambda}_1, \mathbf{x}) = \mathbf{0}$.

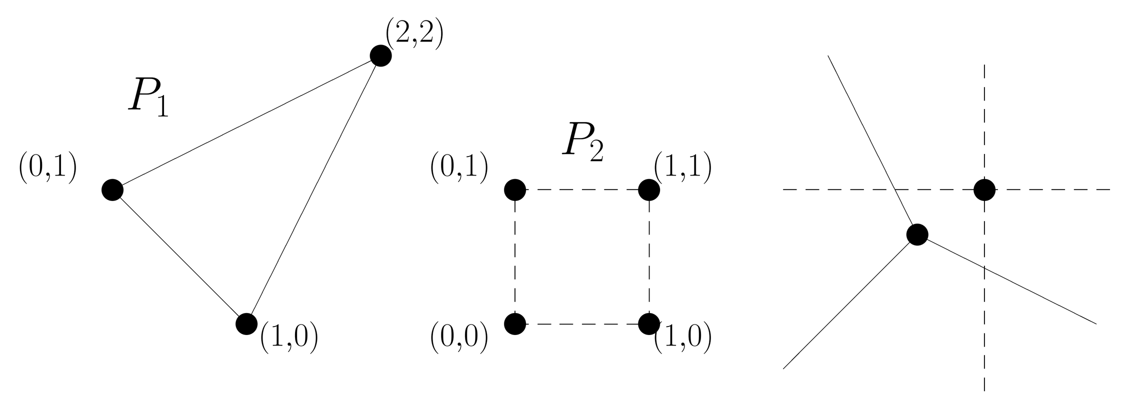


A generic choice for start parameters avoids singularities along the paths.

2. Polyhedral Homotopies

Newton polytopes of a system $f(\mathbf{x}) = \mathbf{0}$ model the sparse structure:

$$f_1(\mathbf{x}) = a_{(2,2)}x_1^2x_2^2 + a_{(1,0)}x_1 + a_{(0,1)}x_2$$
$$f_2(\mathbf{x}) = b_{(1,1)}x_1x_2 + b_{(1,0)}x_1 + b_{(0,1)}x_2 + b_{(0,0)}$$



Bernshteĭn: the mixed volume $V(P_1, P_2) \geq$ #isolated roots in $(\mathbb{C}^*)^n$, $\mathbb{C}^* = \mathbb{C} \setminus \{0\}$.
Polyhedral homotopies are optimal for systems with generic coefficients.

References

[1] T. Gao, T. Y. Li, M. Wu. Algorithm 846: MixedVol: a software package for mixed-volume computation. *ACM Trans. Math. Softw.*, 31(4):555-560, 2005.

[2] J. Verschelde. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Trans. Math. Softw.*, 25(2):251–276, 1999. `http://www.math.uic.edu/~jan/download.html`.

3. Using `phcpy`

```
>>> from phcpy.solver import solve
>>> f = ['x**2*y**2 + x + y;','x*y + x + y + 1;']
>>> s = solve(f,silent=True)
>>> len(s)
4
>>> print s[0]
t :  1.00000000000000E+00 0.00000000000000E+00
m :  1
the solution for t :
x :  -1.00000000000000E+00 0.00000000000000E+00
y :  -1.61803398874989E+00 0.00000000000000E+00
== err :  9.930E-17 = rco :  4.775E-02 = res :  2.220E-16 =
```

Indicators for the quality of the solution:

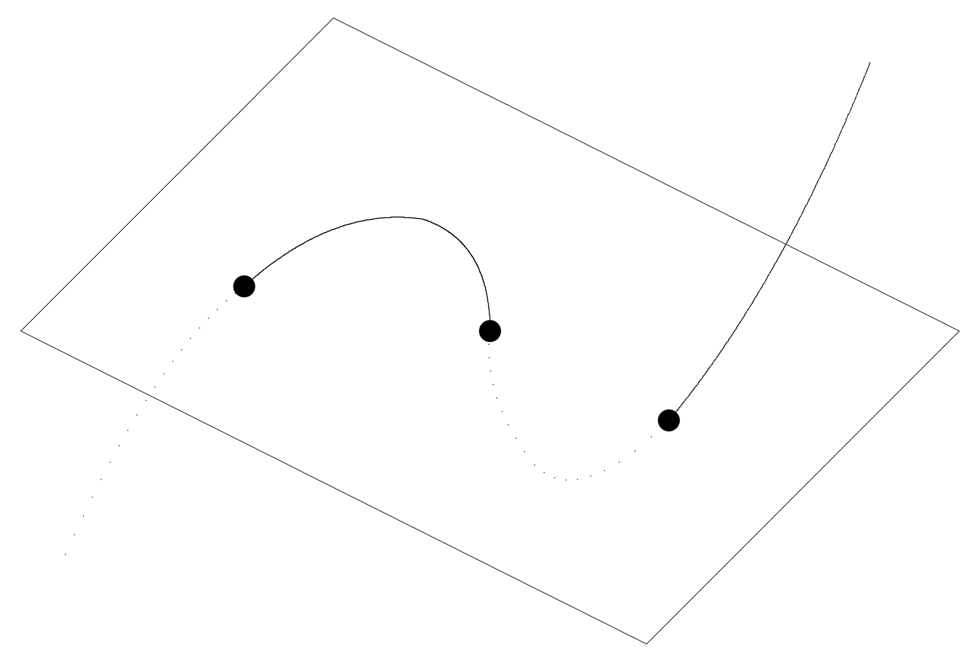
- `err`: the norm of the last update made by Newton's method (forward error),
- `rco`: estimate for the inverse condition number of the Jacobian matrix,
- `res`: norm of the evaluated solution (backward error).

With double double and quad double arithmetic we get more accurate solutions.
To predict the number of isolated solutions with the mixed volume:

```
>>> from phcpy.solver import mixed_volume
>>> mixed_volume(f)
4
```

4. Numerical Irreducible Decomposition

Numerical representations of positive dimensional solution sets.



We cut the space curve with a random plane to find its degree.

References

[3] Y. Hida, X.S. Li, and D.H. Bailey. Algorithms for quad-double precision floating point arithmetic. In *15th IEEE Symposium on Computer Arithmetic (Arith-15 2001)*, 11-17 June 2001, Vail, CO, USA, pages 155-162. IEEE Computer Society, 2001. Shortened version of Technical Report LBNL-46996.

[4] A.J. Sommese, J. Verschelde, and C.W. Wampler. Introduction to Numerical Algebraic Geometry. Chapter 8 of Volume 14 of *Algorithms and Computation in Mathematics*, edited by A. Dickenstein and I.Z. Emiris, Springer 2005.

5. The Software Problem

At least three motivations to develop `phcpy`:

1. PHCpack is a large Ada package, its executable `phc` operates via menus, with input and output to files.
   → provide `phc` with an interpreter interface.
2. The code in PHCpack lacks adequate *user* documentation so that many of its features are not obviously accessible to users.
   → refactor PHCpack into Python modules, documented by Sphinx.
3. Reproducibility of published computational results.
   → perform regression tests with Python scripts.

6. The Package `phcpy`

Version 0.1.0 of `phcpy` contains the following modules:

- `solver`: a blackbox solver, mixed-volume calculator, linear-product root count and start system, path trackers, deflation for isolated singular solutions;
- `examples`: a selection of interesting benchmark systems.
- `families`: some problems can be formulated for any number of variables.
- `phcsols`: conversion of PHCpack solution strings into Python dictionaries.
- `phcsets`: basic tools to manipulate positive dimensional solution sets.
- `phcwulf`: basic client/server setup to solve many systems.
- `schubert`: the Pieri homotopies solve particular polynomial systems arising in enumerative geometry.

Currently 893 functions are exported, documented by Sphinx.

7. Implementation and Applications

The Python implementation builds on the C interface to PHCpack.
We developed the C interface to PHCpack for use with MPI (Message Passing Interface) to track many solution paths on distributed memory multiprocessor computers.

Data structures for polynomials and representations of solutions are not replicated in the interface. Data is passed to and from PHCpack via strings.
Using PHCpack as a state machine, we can hold persistent sessions: data passed to functions is not lost after the execution of the function.

Some applications of `phcpy`:

- Further development of our web interface (joint with Xiangcheng Yu) at `https://kepler.math.uic.edu` (beta version).
- Replacement of `phc.py` (v3 by Marshall Hampton and Alex Jokela) in Sage.
- Interoperability with Gfan (Anders Jensen) for tropical algebraic geometry.

References

[5] K. Piret. Computing Critical Points of Polynomial Systems using PHCpack and Python. PhD Thesis, University of Illinois at Chicago, 2008.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1115777.

University of Illinois at Chicago, Department of Mathematics, Statistics and Computer Science, 851 S. Morgan (Mail Code 249), IL 60607-7045, USA