Opening the Development of PHCpack

Jan Verschelde

University of Illinois at Chicago
Department of Mathematics, Statistics, and Computer Science
`http://www.math.uic.edu/~jan`
`jan@math.uic.edu`

Ada devroom, FOSDEM 2015, 31 January, Brussels, Belgium

Outline

1. Development of Mathematical Software
   - solving systems of polynomial equations
   - goals of the development

2. Opening the Development
   - github repository
   - suggestions for projects

Opening the Development of PHCpack

1. Development of Mathematical Software
    - solving systems of polynomial equations
    - goals of the development

2. Opening the Development
    - github repository
    - suggestions for projects

mathematical software to solve polynomial systems

Some examples of algorithms and computed objects:

- evaluation and differentiation of polynomials (algebra),
- trajectories of solutions defined by parameters (geometry).

Example of an input:

```
2
 x**2 + 4*y**2 - 4;
        2*y**2 - x;
```

The polynomials define an ellipse and a parabola.
The solutions of the system are points of intersection.
How? Deform systems, starting at the solutions of

```
2
 x**2 - 1;
 y**2 - 1;
```

What areas of mathematics?

Solving polynomial systems requires many methods.

- computational algebraic geometry
  What does solving a system of polynomials mean?
- discrete and computational geometry
  The exponents of the monomials define a Newton polytope.
- symbolic-numeric computation
  Polynomials are algebraic objects (symbolic algebra), that we evaluate and deform (numerical analysis).
- parallel and distributed computing
  Parallelism includes message passing, shared memory multithreading, and acceleration with graphics processor units.

Quoting [Sturmfels, 2002]: "*Systems of polynomial equations are for everyone: from graduate students in computer science, engineering, or economics, to experts in algebraic geometry.*"

solving polynomial systems with PHCpack

PHCpack is a package for Polynomial Homotopy Continuation.
ACM Transactions on Mathematical Software achived version 1.0 (Ada 83) as Algorithm 795, vol. 25, no. 2, pages 251–276, 1999.

blackbox solver:

`phc -b` computes all isolated solutions of a polynomial system.

Version 2.0 was rewritten using concepts of Ada 95 and extended with arbitrary multiprecision arithmetic.

Versions 2.1, 2.2, and 2.3 added a *numerical irreducible decomposition* for positive dimensional solution sets. Since summer of 2005, smaller and more frequent releases enable evolutionary improvements.

Distributed under the GNU General Public License.

Public repository under version control at `https://github.com/janverschelde/PHCpack`.

users and applications

Google scholar counts the citations to over four hundred.
More than eighty documented users in the scientific literature.

What counts as a user?

- Reported results obtained with the software,
- obtained without assistance that would make the software developer a co-author.

Some sampling of the application areas:

- real algebraic geometry: counterexample to Kouchnirenko's conjecture [Haas 2002],
- mechanism design: constrained robots [Perez-McCarthy 2004],
- geomagnetism: stable states in chains of crystals [Newell 2009].

The list is at `http://www.math.uic.edu/∼jan/users.html`.

Opening the Development of PHCpack

1. Development of Mathematical Software
   - solving systems of polynomial equations
   - goals of the development
2. Opening the Development
   - github repository
   - suggestions for projects

new methods and algorithms

PHCpack prototyped polyhedral homotopies for isolated solutions and a numerical irreducible decomposition for sets of solutions.

Two current projects:

- Littlewood-Richardson homotopies

  In collaboration with Abraham Martin del Campo, Anton Leykin, Frank Sottile, Ravi Vakil, we compute linear planes that meet a given set of planes in specific ways.

- methods from tropical algebraic geometry

  As a continuation of joint work with Danko Adrovic,
  we develop Puiseux power series to represent solution sets.

  Power series are a hybrid form of symbolic-numeric computation.

accuracy, reliability, and robustness

The hardware double precision is often insufficient for accurate results.

- double double, quad double, and multiprecision

  A quad double is an unevaluated sum of 4 doubles,
  improves working precision from 2.2×10^{-16} to 2.4×10^{-63}.

  Implementation by [Y. Hida, X.S. Li, and D.H. Bailey, 2001] has predictable overhead and simple memory management.
  Cost of double double is similar to complex arithmetic.

  Ported to the GPU by [M. Lu, B. He, and Q. Luo, 2010].

- variable precision path tracking

  We estimate how sensitive the output is to errors on the input with condition number estimators. The precision level is based on

  - the expected loss of accuracy, and
  - the desired accuracy of the results.

parallelism and performance

Most computers are multiprocessor and multicore.
Graphic cards have surpassed ordinary CPUs in computing power.

- message passing (with Yusong Wang, Yun Guan, Anton Leykin)

  By design, the main program is written in C, responsible for the job scheduling, with the aid of MPI. The jobs execute Ada procedures.
- multicore shared memory programming (with Genady Yoffe)

  The goal of this project is to offset the cost of double double and quad double arithmetic with multithreaded code.
  `phc -b -t4` runs path trackers in blackbox solver with 4 threads, using the tasking mechanism provided by Ada.
- acceleration with graphics processors (with Xiangcheng Yu).

The code is a mix of Ada, C, and C++ CUDA.

interfaces

Interfaces translate inputs (polynomials) and outputs (solutions).

- with computer algebra and mathematical software systems:

  Maple (with Anton Leykin), MATLAB & Octave (with Yun Guan), Macaulay2 (with Elizabeth Gross and Sonja Petrovic).

  PHCpack is an optional component of Sage, the current `phc.py` interface was developed by Marshall Hampton and Alex Jokela.
- Python scripting with `phcpy` [EuroSciPy 2013 proceedings].
- A web interface (with Xiangcheng Yu) is in beta stage running at `https://kepler.math.uic.edu`.

Opening the Development of PHCpack

1. Development of Mathematical Software
   - solving systems of polynomial equations
   - goals of the development

2. Opening the Development
   - github repository
   - suggestions for projects

open development at github

PHCpack is free and open source.

- Distributed under the GNU General Public License.
- Public repository under version control at `https://github.com/janverschelde/PHCpack`.
  - most recent changes for development code
  - releases are "tags", most recent is 2.3.96
  - facility to report issues with the code
- The long term goal is to develop a sustainable software element for the scientific computing community.

PHCpack is almost twenty years old. Current efforts are dedicated to build the foundations for the next twenty years.

building executable versions and testing

The GNAT GPL compiler is the main development tool.

- The main executable is `phc`, build with `make phc`.

  Most users just download the statically linked binary version and run the blackbox solver as `phc -b`.
- More than 350 test programs, just do `make testall`.

  Test programs verify specific features, e.g.: compute the convex hull of a polytope spanned by points with integer coordinates.

  Two typical modes of input:
  - test for user given data, or
  - on randomly generated inputs.

The development of `phcpy` as a scripting environment for PHCpack gives the user many testing and verification tools.

the structure of the code

A line count with `wc` of semicolons in the .adb files gives 314,174.
The collection of test procedures counts for 65,237 semicolons.
$\rightarrow$ Quick location and isolation of bugs in the code.

Packages are distributed in several directories.

We may distinguish between general and specific code:

- the mathematical library:
  number representations, QD (Quad Double) library, vectors, matrices, integer linear algebra, numerical linear algebra, polynomials, functions to evaluate and differentiate, support sets
- code specific for polynomial homotopy continuation
  - Newton's method is the computational engine
  - data structures for solutions of systems with parameters
  - polyhedral methods to extract structure of a system
  - ...

Opening the Development of PHCpack

1. Development of Mathematical Software
   - solving systems of polynomial equations
   - goals of the development

2. Opening the Development
   - github repository
   - suggestions for projects

suggestions for projects

All help and suggestions for improvement are appreciated.

Below are some suggestions for project:

- much of the code is still Ada 95
- extend mathematical number library with interval arithmetic
- interface with the numerical linear algebra of BLAS, LAPACK
- expand the mathematical library, do *math with Ada*
- add capabilities for large mathematical objects
- help to port code to new architectures and operating systems
- write a graphical user interface
- develop infrastructure for cloud computing
- . . .

Conclusion: PHCpack counts as an Ada success story,
and is open for improvement.

acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1440534.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.