JSAG **5** (2013), 20 – 25

The Journal of Software for
Algebra and Geometry

Interfacing with PHCpack

ELIZABETH GROSS, SONJA PETROVIĆ, AND JAN VERSCHELDE

ABSTRACT. This *Macaulay2* package provides an interface to *PHCpack*, a general-purpose polynomial system solver that uses homotopy continuation. The main method is a numerical blackbox solver which is implemented for all Laurent systems. The package also provides a fast mixed volume computation, the ability to filter solutions, homotopy path tracking, and a numerical irreducible decomposition method. As the size of many problems in applied algebraic geometry often surpasses the capabilities of symbolic software, this package will be of interest to those working on problems involving large polynomial systems.

NUMERICAL HOMOTOPY CONTINUATION. Many problems in applied algebraic geometry require solving, or counting the solutions of, a large polynomial or rational system. *PHCpack* is an interface the program `PHCpack` to one of several efficient polynomial system solvers that use numerical homotopy continuation methods [Li].

The basic idea behind homotopy continuation is simple: to solve a polynomial system $f(\mathbf{x}) = 0$, one first constructs a system $g(\mathbf{x}) = 0$ that is easy to solve and then constructs a homotopy,

$$H(\mathbf{x}(t)) = (1-t)g(\mathbf{x}) + tf(\mathbf{x})$$

in order to numerically track paths from known solutions of g (with $t = 0$) to the solutions of the target system f (with $t = 1$).

Available since release 1.4 of *Macaulay2* [M2], this package is motivated by [Ley] and uses the data types defined by Leykin in `NAGtypes.m2`. The main function of the package allows a *Macaulay2* user to solve a system numerically through a blackbox solver, where the creation of the start system and homotopy continuation is done behind the scenes. The package also provides a fast mixed volume computation and allows the user to filter solutions, to track solution paths explicitly, and to perform numerical irreducible decompositions.

This interface to `PHCpack` offers access to most of the functionality of the software, which has been serving as a development platform for many of the algorithms in numerical algebraic geometry [SVW]. Computations in this paper were done with `phc` version 2.3.61 (version 1.0 was archived in [Ve1]). Since version 2.3.13, `PHCpack` contains `MixedVol` [GLW], and more recently added features are described in [Ve2]. `PHCpack` can solve Laurent systems, so the package includes a method to convert a rational system to a Laurent polynomial system. The underlying polyhedral methods perform well on benchmark problems; in many of those, the mixed volume is computed essentially instantaneously.

To use the methods in this package, the user must have the executable program `phc` available, preferably in the execution path. For more information on this, see the documentation for the package. Although `PHCpack` is open source, we follow the idea of OpenXM [MN+] and require only that the executable `phc` is available in the execution path of the computer.

2010 *Mathematics Subject Classification.* 65H10, 14Q99, 62H99, 68W30.

PHCpack version 1.6; `PHCpack` 2.3.80.

NUMERICAL SOLUTIONS OF A POLYNOMIAL SYSTEM. The main function, `solveSystem`, returns solutions of a system of polynomial or rational equations. Solutions are returned using data types from `NAGtypes`: a collection of `Points` which are approximations to all complex isolated solutions, or a `WitnessSet` for positive-dimensional components. The following system consists of 21 polynomial equations in 21 unknowns, related to a Gaussian cycle conjecture [DSS, §7.4, page 159] in algebraic statistics. The corresponding variety is zero-dimensional of degree 67.

```
Macaulay2, version 1.6
with packages: ConwayPolynomials, Elimination, IntegralClosure, LLLBases,
               PrimaryDecomposition, ReesAlgebra, TangentCone

i1 : needsPackage "PHCpack";
--loading configuration for package "PHCpack" from file ../Macaulay2/init-PHCpack.m2

i2 : var0 = {x_11,x_12,x_16,x_22,x_23,x_33,x_34,x_44,x_45,x_55,x_56,x_66,y_13,y_14,
       y_15,y_24,y_25,y_26,y_35,y_36,y_46};

i3 : QQ[var0];

i4 : rationalSystem = {
       (22/3)*x_11+(8/7)*x_12+2*x_16-1,           x_23*y_13+(22/3)*x_12+(8/7)*x_22,
       x_33*y_13+x_34*y_14+(8/7)*x_23,            x_34*y_13+x_44*y_14+x_45*y_15,
       x_45*y_14+x_55*y_15+2*x_56,                x_56*y_15+(22/3)*x_16+2*x_66,
       (8/7)*x_12+(14/11)*x_22+(12/5)*x_23-1,     x_34*y_24+(14/11)*x_23+(12/5)*x_33,
       x_44*y_24+x_45*y_25+(12/5)*x_34,           x_45*y_24+x_55*y_25+x_56*y_26,
       (12/5)*x_23+(28/51)*x_33+(102/144)*x_34-1, x_56*y_25+x_66*y_26+(8/7)*x_16,
       x_45*y_35+(28/51)*x_34+(102/144)*x_44,     x_55*y_35+x_56*y_36+(102/144)*x_45,
       (102/144)*x_34+(205/162)*x_44+(3/2)*x_45-1, x_16*y_13+x_56*y_35+x_66*y_36,
       x_56*y_46+(205/162)*x_45+(3/2)*x_55,       x_16*y_14+x_66*y_46+(3/2)*x_56,
       (3/2)*x_45+(517/784)*x_55+(8/3)*x_56-1,    x_16*y_15+(517/784)*x_56+(8/3)*x_66,
       2*x_16+(8/3)*x_56+(29/196)*x_66-1};
```

Before running numerical computations, we embed the system into the polynomial ring over ℂ. The original ring will be used for exact computations (e.g. the degree).

```
i5 : system = (sub(ideal rationalSystem, CC[var0]))_*

o5 = {7.33333x   + 1.14286x   + 2x   - 1, x  y   + 7.33333x   + 1.14286x  , x  y
              11           12     16       23 13           12           22   33 13
     ------------------------------------------------------------------------
     + x  y   + 1.14286x  , x  y   + x  y   + x  y  , x  y   + x  y   + 2x  ,
        34 14           23   34 13    44 14    45 15   45 14    55 15     56
     ------------------------------------------------------------------------
     x  y   + 7.33333x   + 2x  , 1.14286x   + 1.27273x   + 2.4x   - 1, x  y   +
      56 15           16     66          12           22       23       34 24
     ------------------------------------------------------------------------
     1.27273x   + 2.4x  , x  y   + x  y   + 2.4x  , x  y   + x  y   + x  y  ,
             23       33   44 24    45 25       34   45 24    55 25    56 26
     ------------------------------------------------------------------------
     2.4x   + .54902x   + .708333x   - 1, x  y   + x  y   + 1.14286x  , x  y   +
         23          33           34       56 25    66 26           16   45 35
     ------------------------------------------------------------------------
     .54902x   + .708333x  , x  y   + x  y   + .708333x  , .708333x   + 1.26543x
            34           44   55 35    56 36           45          34           44
     ------------------------------------------------------------------------
     + 1.5x   - 1, x  y   + x  y   + x  y  , x  y   + 1.26543x   + 1.5x  , x  y
           45       16 13    56 35    66 36   56 46           45       55   16 14
```

```
     ------------------------------------------------------------------------
     + x  y   + 1.5x  , 1.5x   + .659439x   + 2.66667x   - 1, x  y   + .659439x
        66 46       56      45           55           56       16 15          56
     ------------------------------------------------------------------------
     + 2.66667x  , 2x   + 2.66667x   + .147959x   - 1}
               66    16           56           66
o5 : List
i6 : solutions = solveSystem system;
i7 : # solutions
o7 = 67
```

Solutions are returned as a list, each entry being of type `Point`, which includes diagnostic information such as the condition number and the value of the path-tracking variable t. This allows one to decide if a solution is “good" by using `peek`, suppressed here in the interest of space.

```
i8 : solutions_0
o8 = {-.0507429, -.00357721, .688101, -.108384, .475846, -.251429, -.00563527, ...
o8 : Point
```

The solutions can be further refined as necessary. To best illustrate refinement, consider the system of equations with same support as above, but where the rational coefficients have been changed to larger rational numbers:

```
i9 : newSystem = {
       (22531/300)*x_11+(821/70)*x_12+(4507/210)*x_16-1,
       x_23*y_13+(22531/300)*x_12+(821/70)*x_22,
       x_33*y_13+x_34*y_14+(821/70)*x_23, x_34*y_13+x_44*y_14+x_45*y_15,
       x_45*y_14+x_55*y_15+(4507/210)*x_56,
       x_56*y_15+(22531/300)*x_16+(4507/210)*x_66,
       (821/70)*x_12+(140953/11025)*x_22+(12325/504)*x_23-1,
       x_34*y_24+(140953/11025)*x_23+(12325/504)*x_33,
       x_44*y_24+x_45*y_25+(12325/504)*x_34,
       x_45*y_24+x_55*y_25+x_56*y_26,
       x_56*y_25+x_66*y_26+(821/70)*x_16,
       (12325/504)*x_23+(282013/5184)*x_33+(10231/1440)*x_34-1,
       x_45*y_35+(282013/5184)*x_34+(10231/1440)*x_44,
       x_55*y_35+x_56*y_36+(10231/1440)*x_45,
       x_16*y_13+x_56*y_35+x_66*y_36,
       (10231/1440)*x_34+(205697/16200)*x_44+(30529/2520)*x_45-1,
       x_56*y_46+(205697/16200)*x_45+(30529/2520)*x_55,
       x_16*y_14+x_66*y_46+(30529/2520)*x_56,
       (30529/2520)*x_45+(5175321/78400)*x_55+(897/35)*x_56-1,
       x_16*y_15+(5175321/78400)*x_56+(897/35)*x_66,
       (4507/210)*x_16+(897/35)*x_56+(293581/19600)*x_66-1};
i10 : newSolutions = solveSystem newSystem;
i11 : # newSolutions
o11 = 67
```

Solutions with coordinates below or above a given tolerance can be extracted by `zeroFilter` and `nonZeroFilter`, respectively. In the following example, we ask for solutions whose 12th coordinate is effectively zero (i.e. smaller than 10^{-18}). Then, we confirm this by refining the answer to precision 64; notice that the 12th coordinate is now on the order of 10^{-67}.

```
i12 : smallSolution = zeroFilter(newSolutions, 11, 1.0e-18)
o12 = {{.0677823, -.386278, .0204925, -1.44743, .982877, -.366596, -.435274,
      ------------------------------------------------------------------------
      .725281, -.422346, .0841728, .0218581, -2.54132e-19, 46.7882, -12.922,
      ------------------------------------------------------------------------
      -70.411, 8.2731, -10.9958, 202.197, -43.8649, 306.199, 198.688}}
o12 : List
i13 : smallerSolution = refineSolutions(newSystem, smallSolution, 64)
o13 = {{.0677823, -.386278, .0204925, -1.44743, .982877, -.366596, -.435274,
      ------------------------------------------------------------------------
      .725281, -.422346, .0841728, .0218581, 4.71756e-67, 46.7882, -12.922,
      ------------------------------------------------------------------------
      -70.411, 8.2731, -10.9958, 202.197, -43.8649, 306.199, 198.688}}
o13 : List
```

When refining solutions, `phc` also recomputes input coefficients to a higher precision, since rational coefficients may not always have an exact floating-point representation when the precision is limited.

MIXED VOLUME. If the system has as many equations as unknowns, the mixed volume gives an upper bound on the number of isolated solutions with nonzero coordinates. For sufficiently generic coefficients, this bound is sharp. The function `mixedVolume` is illustrated below.

```
i14 : mixedVolume system
o14 = 75
```

This polyhedral computation is faster than solving the system and provides an upper bound on the number of complex isolated roots in the torus. Computing the degree is much slower (and we note that it takes just as long to verify that the variety is zero-dimensional):

```
i15 : time degree ideal rationalSystem
     -- used 231.67 seconds
o15 = 67
```

While mixed volume counts solutions on the torus, one can also compute the stable mixed volume, which counts solutions with zero components as well, by using optional inputs to the method `mixedVolume`. `phc` offers additional functionality and flexibility, not all of which we can illustrate in this short note. Most interestingly, `mixedVolume` offers an option to use a start system, and creates a polyhedral homotopy from a random start system to the given system. The interested reader is referred to the documentation of the package for more information.

NUMERICAL IRREDUCIBLE DECOMPOSITION. Given a list of generators of an ideal I, the package can also compute a `NumericalVariety` with a `WitnessSet` for each irreducible component of $V(I)$. The example below appears in [DSS].

```
i16 : var1 = {x11, x22, x21, x12, x23, x13, x14, x24};
i17 : QQ[var1];
i18 : rationalSystem = { x11*x22-x21*x12, x12*x23-x22*x13, x13*x24-x23*x14};
i19 : system = (sub(ideal rationalSystem, CC[var1]))_*
o19 = {x11*x22 - x21*x12, x12*x23 - x22*x13, - x23*x14 + x13*x24}
o19 : List
```

```
i20 : V = numericalIrreducibleDecomposition system
found 3 irreducible factors

o20 = A variety of dimension 5 with components in
      dim 5:  (dim=5,deg=2) (dim=5,deg=2) (dim=5,deg=4)

o20 : NumericalVariety

i21 : WitSets = V#5;

i22 : w = first WitSets

o22 = (dim=5,deg=2)

o2 : WitnessSet

i23 : w#IsIrreducible

o23 = true
```

In the above example we found three components of dimension five. Let us verify the solutions.

```
i24 : PD = primaryDecomposition ideal rationalSystem

o24 = {ideal (x23*x14 - x13*x24, x21*x14 - x11*x24, x22*x14 - x12*x24, x12*x23 -
      ---------------------------------------------------------------------------
      x22*x13, x11*x23 - x21*x13, x11*x22 - x21*x12), ideal (x13, x23, x11*x22 -
      ---------------------------------------------------------------------------
      x21*x12), ideal (x12, x22, x23*x14 - x13*x24)}

o24 : List
```

As we see, the dimension and degree of each component agree with the numerical calculation:

```
i25 : for I in PD do << "(dim=" << dim I << ", deg=" << degree I << ") "
(dim=5, deg=4) (dim=5, deg=2) (dim=5, deg=2)
```

ACKNOWLEDGMENTS. This material is based upon work partly supported by the National Science Foundation under Grants No. 0713018 and 1115777. The authors are grateful to Anton Leykin for his generous help with `NAGtypes.m2`. Work on this package was carried out while SP was in residence at the University of Illinois at Chicago, and latest contribution was made during the 2012 *Macaulay2* workshop, supported by the US National Science Foundation.

REFERENCES.

[DSS] M. Drton, B. Sturmfels, and S. Sullivant, *Lectures on algebraic statistics*, Oberwolfach Seminars, vol. 39, Birkhäuser Verlag, Basel, 2009.

[GLW] Tangan Gao, T. Y. Li, and Mengnien Wu, *Algorithm 846: MixedVol: a software package for mixed-volume computation*, ACM Trans. Math. Software **31** (2005), no. 4, 555–560.

[Ley] Anton Leykin, *Numerical algebraic geometry*, JSAG **3** (2011), 5–10.

[Li] T.Y. Li, *Numerical solution of polynomial systems by homotopy continuation methods*, Handbook of numerical analysis, Vol. XI, Handb. Numer. Anal., XI, North-Holland, Amsterdam, 2003, pp. 209–304.

[M2] D.R. Grayson and M.E. Stillman, *Macaulay2, a software system for research in algebraic geometry*, available at `www.math.uiuc.edu/Macaulay2/`.

[MN+] Masahide Maekawa, Masayuki Noro, Katsuyoshi Ohara, Nobuki Takayama, and Yasushi Tamura, *OpenXM, an open system to integrate mathematical softwares*, available at `www.OpenXM.org`.

[SVW] Andrew J. Sommese, Jan Verschelde, and Charles W. Wampler, *Introduction to numerical algebraic geometry*, Solving polynomial equations, Algorithms Comput. Math., vol. 14, Springer, Berlin, 2005, pp. 301–335.

[Ve1] Jan Verschelde, *Algorithm 795: PHCPACK: A general-purpose solver for polynomial systems by homotopy continuation*, ACM Trans. Math. Software **25** (1999), no. 2, 251–276, software available at `www.math.uic.edu/~jan/download.html`.

[Ve2] ______, *Polynomial homotopy continuation with PHCpack*, ACM Commun. Comput. Algebra **44** (2010), no. 4, 251–276.

RECEIVED : 2011-05-24 REVISED : 2013-05-15 ACCEPTED : 2013-07-11

egross7@uic.edu : Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, Chicago, IL 60607, USA.

Sonja.Petrovic@iit.edu : Department of Applied Mathematics, Illinois Institute of Technology, Chicago, IL 60616, USA.

jan@math.uic.edu : Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, Chicago, IL 60607, USA.