PHCpack in Macaulay2

ELIZABETH GROSS, SONJA PETROVIĆ, AND JAN VERSCHELDE

Abstract. The Macaulay2 package PHCpack.m2 provides an interface to some of the functionality of PHCpack, a general-purpose solver for polynomial systems by homotopy continuation. The main function of the package interfaces PHCpack’s numerical solver phc, published as Algorithm 795 in ACM Trans. Math. Softw. (TOMS). The blackbox solver computes mixed volumes using MixedVol, ACM TOMS Algorithm 846, and then applies polyhedral homotopy methods to solve a polynomial system. As numerical algebraic geometry plays an important role in applications, we illustrate the package on an example from algebraic statistics. In particular, we verify the Gaussian cycle conjecture for the undirected 8-cycle by numerically solving a square system in 36 variables and obtaining 321 solutions after filtering.

1. The package and its applications

Numerical homotopy continuation methods [6] have led to efficient solvers of polynomial systems. PHCpack implements algorithms in numerical algebraic geometry [8]. Version 1.0 of PHCpack was archived in [9]. Since version 2.3.13, PHCpack contains MixedVol [2]. The multiprecision arithmetic of the library QD-2.3.9 [4] was integrated in version 2.3.55. Other recently added features are described in [10]. Computations in this paper were done with phc version 2.3.61. Our interface PHCpack.m2 offers Macaulay2 users access to the functionality of PHCpack. Available since release 1.4 of Macaulay2 [3], it is motivated by [5] and uses the data types defined by Leykin in NAGtypes.m2. Although PHCpack is open source, we follow the idea of OpenXM [7] and require only that the executable phc is available to PHCpack.m2.

Many problems in applied algebraic geometry require solving, or counting the solutions of, a large polynomial (rational) system. Algebraic statistics, as a field which applies techniques from algebraic geometry to questions in statistics, provides interesting families of problems that require both symbolic and numerical computations.

We illustrate PHCpack.m2 on a problem in algebraic statistics which is formulated in [1, §7.4, page 159]. The running example in this paper asks to compute the degree of a particular zero-dimensional variety given its defining ideal. For square polynomial systems (ideals generated by as many equations as there are unknowns), the mixed volume of the Newton polytopes of the defining equations gives a generically sharp
upper bound on the degree. A byproduct of the mixed volume computation is the construction of a polyhedral homotopy. The polyhedral homotopy defines solution paths leading to all isolated solutions of the system.

To verify a conjecture from [1], we use the symbolic capabilities of Macaulay2 to formulate various instances of the problem as a polynomial system. PHCpack.m2 then passes the systems to the numerical solver phc. We illustrate the commands to compute mixed volumes, solve the system, and to process the solutions within Macaulay2. In the last section we describe more functionality.

2. Investigating a Polynomial System Arising in Algebraic Statistics

We demonstrate the package on a problem for undirected graphical models where the vertices of the graph represent Gaussian random variables. Namely, we verify the conjecture for the maximum likelihood degree of Gaussian cycles; the reader is referred to [1] for background. In essence, the maximum likelihood (ML) degree of a given algebraic model is the degree of the variety characterized by the system of likelihood equations, or, equivalently, the number of complex solutions of the system for generic data. We consider an undirected Gaussian 5-cycle with edges \{(1, 2), \ldots, (4, 5), (1, 5)\}. Given a sample covariance matrix \(S\), the ML degree of this model is the number of complex solutions to the following system:

\[
\begin{pmatrix}
  s_{11} & s_{12} & y_{13} & y_{14} & s_{15} \\
  s_{12} & s_{22} & s_{23} & y_{24} & y_{25} \\
  y_{13} & s_{23} & s_{33} & s_{34} & y_{35} \\
  y_{14} & y_{24} & s_{34} & s_{44} & s_{45} \\
  s_{15} & y_{25} & y_{35} & s_{45} & s_{55}
\end{pmatrix}
\begin{pmatrix}
  x_{11} \\
  x_{12} \\
  0 \\
  0 \\
  x_{15}
\end{pmatrix}
= \begin{pmatrix}
  1 & 0 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 & 0 \\
  0 & 0 & 1 & 0 & 0 \\
  0 & 0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 0 & 1
\end{pmatrix}.
\]

Note that the zero entries in the second matrix coincide with the non-edges of the 5-cycle. The polynomial system corresponding to the above matrix equation has 25 equations in 15 variables \(x_{ij}, y_{ij}\). We can use Macaulay2 to generate the equations using a random covariance matrix. The first of the 25 resulting equations is

\[
\frac{1689}{64} x_{11} + \frac{369}{56} x_{12} + \frac{7867}{420} x_{15} - 1.
\]

The degree of the zero-dimensional variety defined by this system is 17, a computation that Macaulay2 does instantaneously. For larger cycles, however, the symbolic computation is too costly: it took about 4 hours to confirm that the degree for a six-cycle is 49. This further motivates the need to use a numerical solver that can not only count the solutions, but approximate them as well.

For optimal numerical performance, a square system is preferred. We exploit some symmetries to eliminate 10 of the equations and produce a square system in 15 variables. The new system, which we will denote by reducedSystem, defines another zero-dimensional variety of strictly higher degree, and we deal with filtering solutions later. The function phcSolve invokes the command phc -b from PHCPACK and returns approximations to all complex isolated roots.
i1: -- The code for generating the reduced system is suppressed.
-- Details are in the Appendix. The system is called:
reducedSystem;
i2 : time solns=phcSolve(reducedSystem);
using temporary files /tmp/M2-3265-54PHCinput and /tmp/M2-3265-54PHCoutput
   -- used 0.023707 seconds
i3 : #solns
  o3 = 19
i5 : peek solns_5 -- let's take a look at one of the solutions:
o5 = Point{ConditionNumber => 67024.1 }
       Coordinates => {-.00593716, .0823659, .0327775, -.037762,
                       .211698, -.042976, .0893084, -.0000846508, .0432623, 6.72812e-17,
                       -9.0925, -19.9947, -4.99234, -5.44097, -8.25825}
       LastT => 1
       SolutionStatus => Regular

Each solution is of type Point, containing also diagnostic information such as the condition
number and the value of the path-tracking variable, which allows one to decide if the
solutions is “good”.

The solutions can be further refined as necessary. For example, [1 Conjecture 7.4] states
that the expected number of solutions to this system is 17. Statistics tells us that we need to
discard those solutions where certain $x$ variables are zero (namely, $x_{55}$ in this case). This is
best done by first refining our 19 solutions to 20 decimal places in working precision. Since
rational coefficients may not always have an exact floating-point representation when the
precision is limited, we also recompute the input coefficients to a higher working precision
for root refinement. We can apply refineSolutions to the whole solution set, or just one
Point. For example, the refinement of the solution above is:
i6 : time P=refineSolutions(reducedSystem,{solns_5},20);
using temporary file /tmp/M2-3265-77PHCoutput for storing refined solutions
   -- used 0.007328 seconds
i7 : P_0#Coordinates
  o7 = {-.00593716, .0823659, .0327775, -.037762,
        .211698, -.042976, .0893084, -.0000846508, .0432623, -2.24619e-24,
        -9.0925, -19.9947, -4.99234, -5.44097, -8.25825}
o7 : List

We see that one of the coordinates is closer to zero after the refinement. When all 19
solutions are refined, two of them have $x_{55}$ coordinate very close to zero. Thus we adjust
the ML degree to 17.

Remark 2.1. In case of the 5-cycle, refining the solutions was good enough. We verified
the conjecture for cycles up to 8 vertices, which required a bit more work in MACAULAY2:
namely, we checked that each solution obtained numerically actually satisfies all of the 25
equations, and discarded the ones which do not.

3. More functionality of PHCpack.m2

PHCpack.m2 can also compute the mixed volume of the polynomial system, which provides
an upper bound on the number of complex isolated roots without zero components. With
stable mixed volumes, we count and compute solutions with zero components as well, but for the application at hand, it is convenient to exclude solutions with zero components in advance. For the output of this function to be most meaningful, the input needs to be a square system. We make use of the reduced system once again:

```
--for using the option startSystem, we need complex numbers:
i8 : reducedSystem = gens sub(ideal reducedSystem, CC[var]);
i9 : (mv,sv,q,qssols)=mixedVolume(reducedSystem, stableMV => true, startSystem => true);
using temporary files /tmp/M2-3265-84PHCinput and /tmp/M2-3265-84PHCoutput
i10 : (mv,sv) --mixed volume and stable mixed volume
     o10 = (19, 19)
```

While mixed volume counts solutions on the torus, the stable mixed volume counts solutions with zero components as well. Note that the mixed volume is, in general, sharp, but not for our particular system.

The original system from our example is overdetermined, resulting in insertion of slack variables when we call the black box solver. This means that \texttt{phcSolve} will likely return more solutions than the original system has. To remedy this, we can filter the solutions to return only those where each slack variable is zero. Often in applications we are only interested in real solutions; \texttt{realFilter} selects only those solutions with complex components smaller than a given tolerance. In the interest of space, we refer the reader to the documentation for examples.

In general, the likelihood equations for Gaussian graphical models provide many examples of rational systems. \texttt{PHCpack} can solve Laurent systems, so the package includes a \texttt{convertToPoly} function, which converts a rational system to a Laurent polynomial system.

### Table 1. Summary of numerical computations for Gaussian cycles.

| cycle | variables | Reduced system | Likelihood equations |
|-------|-----------|----------------|----------------------|
|       |           | mixed vol      | degree               |
| 5     | 15        | 19             | 19                   |
| 6     | 21        | 75             | 67                   |
| 7     | 28        | 291            | 230                  |
| 8     | 36        | 1111           | 791                  |
|       |           | 25             | 17                   |
|       |           | 36             | 49                   |
|       |           | 49             | 129                  |
|       |           | 64             | 321                  |

### References

[1] M. Drton, B. Sturmfels, and S. Sullivant. *Lectures on Algebraic Statistics*, volume 39 of *Oberwolfach Seminars*. Birkhäuser, 2009.

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

[3] D.R. Grayson and M.E. Stillman. Macaulay 2, a software system for research in algebraic geometry. Available at http://www.math.uiuc.edu/Macaulay2/.

[4] 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, software at http://crd.lbl.gov/~dhbailey/mpdist/qd-2.3.9.tar.gz.
PHCpack in Macaulay2

[5] A. Leykin. Numerical algebraic geometry. *JSAG*, 3:5–10, 2011.

[6] T.Y. Li. Numerical solution of polynomial systems by homotopy continuation methods. In F. Cucker, editor, *Handbook of Numerical Analysis. Volume XI. Special Volume: Foundations of Computational Mathematics*, pages 209–304. North-Holland, 2003.

[7] M. Maekawa, M. Noro, K. Ohara, N. Okutani, Y. Takayama, and Y. Tamura. OpenXM – an open system to integrate mathematical softwares. Available at [http://www.OpenXM.org/](http://www.OpenXM.org/).

[8] A.J. Sommese, J. Verschelde, and C.W. Wampler. Introduction to numerical algebraic geometry. In *Solving Polynomial Equations. Foundations, Algorithms and Applications*, volume 14 of *Algorithms and Computation in Mathematics*, pages 301–337. Springer-Verlag, 2005.

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

[10] J. Verschelde. Polynomial homotopy continuation with PHCpack. *ACM Communications in Computer Algebra*, 44(4):217–220, 2010.
Appendix: code for the examples

```plaintext
i1 : loadPackage "PHCpack";

i2 : n=5;

i3 : f=(y)->x_y;

i4 : f2=(x)->y_x;

i5 : g=(x,y)->10*(x+1)+(y+1);

i6 : k=(x,y)->abs(x-y);

i7 : t=table(n,n, (i,j)->if k(i,j)>1 and k(i,j)!=(n-1) then 0 else if i<=j then f(g(i,j))
   else f(g(j,i)));

i8 : flatten(oo);

i9 : delete(0,oo);

i10 : xvars=unique oo

o10 = {x , x , x , x , x , x , x , x , x , x }
       11 12 15 22 23 33 34 44 45 55

i11 : ty=table(n,n, (i,j)->if k(i,j)<=1 or k(i,j)==(n-1) then 0 else if i<=j then f2(g(i,j))
   else f2(g(j,i)))

i12 : flatten(oo);

i13 : delete(0,oo);

i14 : yvars=unique oo

o14 = {y , y , y , y , y }
       13 14 24 25 35

i15 : var=join(xvars,yvars)

o15 = {x , x , x , x , x , x , x , x , x , x , y , y , y , y , y }
       11 12 15 22 23 33 34 44 45 55 13 14 24 25 35

i16 : R=QQ[ var]

o16 = R

o16 : PolynomialRing

i17 : -- system and reducedSystem
   K=matrix table(n,n, (i,j)->if k(i,j)>1 and k(i,j)!=(n-1) then 0 else if i<=j then f(g(i,j))
   else f(g(j,i)));

o17 = | x_{11} x_{12} 0 0 x_{15} |
     | x_{12} x_{22} x_{23} 0 0 |
```

| 0  x_{23}  x_{33}  x_{34}  0  |
| 0  0  x_{34}  x_{44}  x_{45}  |
| x_{15}  0  0  x_{45}  x_{55}  |

5  5

o17 : Matrix R <--- R

i18 : MY=matrix table(n,n, (i,j)->if k(i,j)<=1 or k(i,j)==(n-1) then 0 else if i<=j then f2(g(i,j))
               else f2(g(j,i)))

o18 = | 0  0  y_{13}  y_{14}  0  |
| 0  0  0  y_{24}  y_{25}  |
| y_{13}  0  0  0  y_{35}  |
| y_{14}  y_{24}  0  0  0  |
| 0  y_{25}  y_{35}  0  0  |

5  5

o18 : Matrix R <--- R

i19 : X=matrix table(n,n, (i,j)->random QQ)

o19 = | 1  1  9/2  3/8  2  |
| 3/10  1  5/7  1/5  1  |
| 1  2/3  1  1  3/4  |
| 9/4  2/3  4/9  2  7/3  |
| 1/10  5/7  3  10/9  2  |

5  5

o19 : Matrix QQ <--- QQ

i20 : S=X*transpose(X)

o20 = | 1689/64  369/56  193/24  31/3  7867/420  |
| 369/56  1293/4900  221/84  11069/2520  32189/6300  |
| 193/24  221/84  577/144  62/9  1949/315  |
| 31/3  11069/2520  62/9  19633/1296  22487/2520  |
| 7867/420  32189/6300  1949/315  22487/2520  5856169/396900  |

5  5

o20 : Matrix QQ <--- QQ

i21 : SigmaHat=matrix table(n,n, (i,j)->if k(i,j)<=1 or k(i,j)==(n-1) then S_(i,j) else MY_(i,j))

o21 = | 1689/64  369/56  y_{13}  y_{14}  7867/420  |
| 369/56  1293/4900  221/84  y_{24}  y_{25}  |
| y_{13}  221/84  577/144  62/9  y_{35}  |
| y_{14}  y_{24}  62/9  19633/1296  22487/2520  |
| 7867/420  y_{25}  y_{35}  22487/2520  5856169/396900  |

5  5

o21 : Matrix R <--- R

i22 : SigmaHat=sub(SigmaHat,R)

o22 = | 1689/64  369/56  y_{13}  y_{14}  7867/420  |
| 369/56  1293/4900  221/84  y_{24}  y_{25}  |
| y_{13}  221/84  577/144  62/9  y_{35}  |
| y_{14}  y_{24}  62/9  19633/1296  22487/2520  |
| 7867/420  y_{25}  y_{35}  22487/2520  5856169/396900  |

5  5
\( Y = \Sigma \Hat{H} \times K \)
The following shows how we filtered the solutions for the Gaussian 8-cycle. This was done by hand before the filtering methods were created. Using PHCPACK we were thus able to confirm the Gaussian cycle conjecture for n=8.
\begin{verbatim}
170 : SigmaHat=matrix{
{2827/36, 851/24, y_13,y_14,y_15,y_16, y_17,4703/60},
{851/24,160297/1296, 32747/1890,y_24,y_25,y_26,y_27,y_28},
{y_13,32747/1890, 28838041/6350400, 171247/18144, y_35,y_36,y_37,y_38},
{y_14, y_24, 171247/18144, 148675/1728, 117437/10080, y_46, y_47,y_48},
{y_15,y_25, y_35, 117437/10080, 145709/19600, 3127/504, y_57,y_58},
{y_16,y_26, y_36,y_46, 3127/504, 290047/198450, 48929/5040,y_68},
{y_17, y_27,y_37,y_47,y_57,48929/5040, 3390829/254016, 76271/5040},
{4703/60, y_28,y_38,y_48,y_58,y_68, 76271/5040, 2062169/14400} ;
);

171 : solutions=parseSolutions("/tmp/gcn8sols",R);
172 : Kinv=(i)->matrix{{(solutions_i#Coordinates)_0,(solutions_i#Coordinates)_1,0,0,0,0,
(solutions_i#Coordinates)_2},
{(solutions_i#Coordinates)_1,(solutions_i#Coordinates)_3,(solutions_i#Coordinates)_4,
0,0,0,0,0},
{0,(solutions_i#Coordinates)_4,(solutions_i#Coordinates)_5,(solutions_i#Coordinates)_6,
6,0,0,0,0,0},
{0,0, (solutions_i#Coordinates)_6,(solutions_i#Coordinates)_7,(solutions_i#Coordinates)_8,
8,0,0,0,0,0},
{0,0,0,(solutions_i#Coordinates)_8,(solutions_i#Coordinates)_9,(solutions_i#Coordinates)_10,0,0,0,0},
{0,0,0, (solutions_i#Coordinates)_10,(solutions_i#Coordinates)_11,(solutions_i#Coordinates)_12,0,0,0,0},
{0,0,0,0,(solutions_i#Coordinates)_12,(solutions_i#Coordinates)_13,(solutions_i#Coordinates)_14,0,0,0,0},
{(solutions_i#Coordinates)_2,0,0,0,0,0,(solutions_i#Coordinates)_14,0,0,0,0,0,0}};
072 = Kinv
072 : FunctionClosure

173 : s=(i)->sub(SigmaHat,{y_13=>(solutions_i#Coordinates)_16,y_14=>(solutions_i#Coordinates)_17,
y_15=>(solutions_i#Coordinates)_18,y_16=>(solutions_i#Coordinates)_19,
y_17=>(solutions_i#Coordinates)_20, y_24=>(solutions_i#Coordinates)_21,
y_25=>(solutions_i#Coordinates)_22,y_26=>(solutions_i#Coordinates)_23,
y_27=>(solutions_i#Coordinates)_24, y_28=>(solutions_i#Coordinates)_25,
y_35=>(solutions_i#Coordinates)_26,y_36=>(solutions_i#Coordinates)_27,
y_37=>(solutions_i#Coordinates)_28, y_38=>(solutions_i#Coordinates)_29,
y_46=>(solutions_i#Coordinates)_30,y_47=>(solutions_i#Coordinates)_31,
y_48=>(solutions_i#Coordinates)_32, y_57=>(solutions_i#Coordinates)_33,y_58=>(solutions_i#Coordinates)_34,
y_68=>(solutions_i#Coordinates)_35});

174 : solns={};
175 : for i from 0 to #solutions-1 do
   (if abs(sub(sum apply((ideal flatten(Kinv(i)*s(i)-id_(R^8)))_*,i->i^2),CC))<.000000000000001
then solns=append(solns, solutions_i));
176 : #solns 076 = 321
\end{verbatim}
PHCpack in Macaulay2

Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, 851 S. Morgan Street (m/c 249), Chicago, IL 60607-7045, USA
E-mail address: egross7@uic.edu
URL: www.math.uic.edu/~lizgross

Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, 851 S. Morgan Street (m/c 249), Chicago, IL 60607-7045, USA
E-mail address: petrovic@math.uic.edu
URL: www.math.uic.edu/~petrovic
Current address: Department of Statistics, Penn State University, University Park, PA 16802, USA

Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, 851 S. Morgan Street (m/c 249), Chicago, IL 60607-7045, USA
E-mail address: jan@math.uic.edu
URL: www.math.uic.edu/~jan