

Numerical Algebraic Geometry

ANTON LEYKIN

ABSTRACT. Numerical algebraic geometry uses numerical data to describe algebraic varieties. It is based on numerical polynomial homotopy continuation, which is an alternative to the classical symbolic approaches of computational algebraic geometry. We present a package, whose primary purpose is to interlink the existing symbolic methods of *Macaulay2* and the powerful engine of numerical approximate computations. The core procedures of the package exhibit performance competitive with the other homotopy continuation software.

INTRODUCTION. Numerical algebraic geometry [SVW, SW] is a relatively young subarea of computational algebraic geometry that originated as a blend of the well-understood apparatus of classical algebraic geometry over the field of complex numbers and numerical polynomial homotopy continuation methods. Recently steps have been made to extend the reach of numerical algorithms making it possible not only for complex algebraic varieties, but also for schemes, to be represented numerically. What we present here is a description of “stage one” of a comprehensive project that will make the machinery of numerical algebraic geometry available to the wide community of users of *Macaulay2* [M2], a dominantly symbolic computer algebra system. Our open-source package [L1, M2] was first released in *Macaulay2* distribution version 1.3.1.

Stage one has been limited to implementation of algorithms that solve the most basic problem:

Given polynomials $f_1, \dots, f_n \in \mathbb{C}[x_1, \dots, x_n]$ that generate a 0-dimensional ideal $I = (f_1, \dots, f_n)$, find numerical approximations of all points of the underlying variety $V(I) = \{\mathbf{x} \in \mathbb{C}^n \mid \mathbf{f}(\mathbf{x}) = 0\}$.

This task is accomplished by using homotopy continuation. To solve a target polynomial system $\mathbf{f} = (f_1, \dots, f_n) = 0$, one constructs a start polynomial system $\mathbf{g} = (g_1, \dots, g_n)$ with a similar structure, but readily available solutions. One option for a start system is $\mathbf{g} = (x_1^{\deg(f_1)} - 1, \dots, x_n^{\deg(f_n)} - 1)$. Given such an auxiliary system, we consider the homotopy

$$(\spadesuit) \quad \mathbf{h} = (1 - t)\mathbf{g} + \gamma t \mathbf{f} \in \mathbb{C}[\mathbf{x}, t], \quad \gamma \in \mathbb{C}^* .$$

By specializing the values of t to the real interval $[0, 1]$, we obtain a collection of paths leading from the known solutions of the start system $\mathbf{g} = \mathbf{h}|_{t=0}$ to the unknown solutions of the target system $\mathbf{f} = \mathbf{h}|_{t=1}$. The following statement provides the key to making effective numerical computations.

Theorem. *For all but finitely many values of γ in the homotopy (\spadesuit), the homotopy continuation paths have no singularities with a possible exception of the endpoints corresponding to $t = 1$. Moreover, every solution of the target system, provided there are finitely many, is an endpoint (at $t = 1$) of some continuation path.*

2010 Mathematics Subject Classification. 14Q99; 68N01.
NumericalAlgebraicGeometry version 1.3.1.

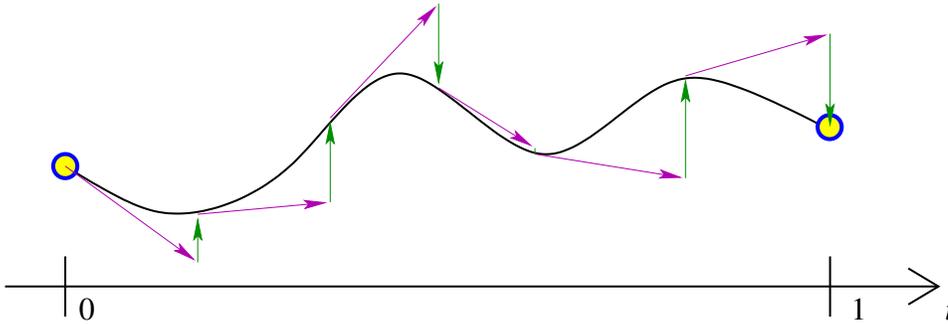


FIGURE 1. Tracking procedure: predictor steps (using the tangent predictor) are followed by corrector steps.

Proof. Let P be the linear space of systems of n complex polynomials in n variables, so that $P \cong \mathbb{C}^m$ for some $m \in \mathbb{N}$. Systems with at least one singular solution form a Zariski closed set $A \subseteq P$. The homotopy (\spadesuit) satisfies the conditions of Lemma 7.1.3 of [SW]. Hence, it follows that for all but finitely many choices of γ the homotopy (\spadesuit) for $t \in [0, 1]$ misses the set A . \square

Differentiating the equation $\mathbf{h} = \mathbf{0}$ gives the following system of ordinary differential equations

$$(\clubsuit) \quad \frac{d\mathbf{x}}{dt} = \mathbf{h}_x^{-1} \mathbf{h}_t,$$

where \mathbf{h}_x is the Jacobian of the homotopy (with respect to \mathbf{x}) and \mathbf{h}_t is the derivative with respect to the parameter t . The solutions of (\clubsuit) for $t \in [0, 1]$ with initial conditions given by the solutions of the start system are the continuation paths we need. Finding continuation paths approximately, therefore, reduces to numerical solving systems of ordinary differential equations. There is an additional advantage: at any point $t = t_0$ we can refine an approximate solution to the polynomial system $\mathbf{h}|_{t=t_0} = \mathbf{0}$ with Newton's method. Provided the numerical tracking procedure has not deviated from the given path, this brings the approximation as close to the path as desired.

The basic *tracker* operates by alternating *predictor* (a numerical integration step) and *corrector* (several applications of Newton's operator) as shown in Figure 1. The ultimate goal of the tracker is to approximate the end of a continuation path given an approximation of its beginning. Here is an example illustrating the tracking a homotopy.

```
Macaulay2, version 1.4
with packages: ConwayPolynomials, Elimination, IntegralClosure, LLLBases,
               PrimaryDecomposition, ReesAlgebra, TangentCone

i1 : needsPackage "NumericalAlgebraicGeometry"
i2 : R = CC[x,y];
i3 : S = {x^2-1,y^2-1}; T = {x^2+(y-5)^2-16, x*y};
i5 : solsS = {(1,-1),(1,1),(-1,1),(-1,-1)};
i6 : track(S,T,solsS)
o6 = {[M,t=.0476212], {2.95499e-17, 1}, {9.9121e-14, 9}, [M,t=.0476212]}
i7 : track(S,T,solsS,gamma=>0.6+0.8*ii)
```

```
o7 = {{1.37478e-13-4.76293e-14*ii, 9}, {3*ii, -5.90007e-17+1.11575e-16*ii},
-----
      {-2.74447e-17+4.14299e-17*ii, 1}, {-3*ii, 3.05714e-18-1.24616e-17*ii}}
```

The output line o6 indicates that the tracking of two paths fails as $\mathbf{h}|_{t \approx 0.04762} = 0$ has singular solutions. When both start and target systems have real coefficients, the parameter value $\gamma = 1$ often results in a homotopy that goes through a singular point. Picking a random complex value for γ (in practice, a number is chosen on the unit circle with the uniform probability distribution) results in a regular homotopy with probability 1. The black-box solver `solveSystem` does exactly that when it calls `track`.

SOFTWARE IMPLEMENTATION STRATEGY. Despite the seeming simplicity of this approach, there are many technical challenges which must be overcome in any effective implementation, such as choosing a predictor mechanism and handling dynamic step adjustment. The development of *NumericalAlgebraicGeometry* started with the basic function called `track` that given a target system, a start system, and a list of start solutions numerically tracks the corresponding homotopy paths as the continuation parameter t is varied from 0 to 1. This was carried out in the top level language of *Macaulay2* and included more than a dozen optional parameters allowing users to experiment with the numerical thresholds that control the precision of the computation.

The language of *Macaulay2* is, however, an interpreted language, so the performance of the code was far from optimal. As a result, it was crucial to implement the computationally intensive parts of the code in C++ as part of the *Macaulay2* engine. At present, parts of the source code are written in three languages:

- The *Macaulay2* language processes the input, sets-up the evaluation, predictor, and corrector routines, launching the routines in the engine, and managing the output;
- The C++ language is used for fast execution of the predictor-corrector steps using standard double floating point arithmetic;
- The D language provides an interface for converting objects of *Macaulay2* top level classes into objects of types and classes used by the C++ engine.

The C++ part relies on LAPACK [ABB⁺] for numerical linear algebra. This is the only external library that has been used so far.

PERFORMANCE. In addition to native implementation, the user is given an option to outsource the computation to one of the three polynomial homotopy continuation softwares: *Bertini* [BHSW1], *Hom4PS2* [LLT], or *PHCpack* [V]. Naturally, this depends on the availability of the relevant software on the user's platform.

One goal of stage one was to achieve performance competitive with the existing software. The following timings (obtained on a single core of a 64-bit Linux system) demonstrate the results for several test problems with a moderate number of solutions (less than 10,000):

- Random _{n} ^{d} : a system of dense polynomials of degree d in n variables with random coefficients;
- Katsura _{n} : a classical benchmark with one linear and $n - 1$ quadratic equations in n variables;
- GEVP _{n} : the system corresponding to a generalized eigenvalue problem, $Av = \lambda Bv$ for $n \times n$ (randomly generated) matrices A and B .

All these problems have regular solutions and do not encounter near-singularities (i.e. the continuation paths are sufficiently far from each other), so that double-precision arithmetic is enough to carry out the computations. All runs for all software are made in standard double precision with default settings.¹

| | Random ₅ ⁴ | Random ₅ ⁵ | Katsura ₁₁ | Katsura ₁₂ | GEVP ₃₅ |
|-----------------------------------|----------------------------------|----------------------------------|-----------------------|-----------------------|--------------------|
| # solutions | 1024 | 3125 | 1024 | 2048 | 35 |
| <i>NumericalAlgebraicGeometry</i> | 4 s | 30 s | 4 s | 11 s | 3 s |
| <i>Hom4PS2</i> | 11 s | 78 s | 7 s | 19 s | ? ² |
| <i>Bertini</i> | 51 s | 402 s | 15 s | 37 s | 40 s |
| <i>PHCpack</i> | 63 s | 550 s | 37 s | 102 s | 323 s |

For all systems except the last one, the number of solutions equals the total degree; for GEVP_n, we supply an optimal homotopy with a start system having exactly n solutions. One may rerun the examples used to obtain the timings using the files `showcase.m2` and `benchmarks.m2` in the *Macaulay2* repository³. The (current) default options of the function `track` were used to complete the tests. In particular, `Predictor => RungeKutta4` specified a predictor of the fourth order, which performed slightly better than lower order numerical integration algorithms on all mentioned examples.

TECHNICAL DETAILS. There are many factors that may affect performance of the homotopy tracking procedure including the choice of predictor, parameters that control dynamical step adjustment, and efficiency of linear algebra subroutines. Nevertheless, one common bottleneck in the computation is the evaluation of polynomials. All test examples mentioned above, except Katsura_n, are evaluation-intense. The evaluation of polynomials presented in the dense form, which is the form *Macaulay2* uses, is quite expensive. To speed up the computation we employ an *extended Horner scheme* (see [CG]), a generalization of the Horner scheme for evaluating univariate polynomials. While not necessarily an optimal scheme (in the sense the univariate Horner scheme is), it makes an attractive design choice for two reasons. First, we can encode the evaluation procedure in a *straight-line program* (SLP), a program evaluating which can be achieved without branching or looping. An SLP corresponding to an extended Horner form is often much shorter than that for the dense form. Second, the automatic differentiation of polynomials represented in Horner form is convenient as the size of an SLP representing the evaluation of a polynomial function and its derivatives simultaneously is not much larger than the size of an SLP for the polynomial function itself (see the corresponding theoretical complexity result in [BS]).

We experimented with speeding up the evaluation of SLPs even further by compiling SLPs at runtime. This results, on some evaluation-intense examples, in up to 10 times faster execution.⁴ At the moment, the option of compiled SLPs needs to be studied further — the current compilation time may exceed the gain in computation time in some cases.

¹The timing ranks may depend on our selection of benchmarks. In fact, there are many problems which can not be solved with one program, but are solved by another quickly. In addition, timings depend on a variety of factors including the hardware, the operating system, and the parameters of continuation.

²*Hom4PS2* has no option to handle a user-defined homotopy.

³`svn://macaulay2.math.uiuc.edu/Macaulay2/trunk/M2/Macaulay2/packages/NumericalAlgebraicGeometry`

⁴This trick seems to be used in *Hom4PS2* as well. However, the timings reported for the test examples above are obtained by running all programs without runtime compilation.

CERTIFICATION. One important feature that distinguishes *NumericalAlgebraicGeometry* from other software is the *certified homotopy tracking* procedure developed in joint work with Beltrán [BL1]. In general, the heuristic algorithms discussed above need careful tuning before path-jumping (landing on a wrong path) is eliminated. The most commonly used step control procedures are described in [BHSW2] along with *Bertini*'s adaptive multiprecision approach for overcoming poor conditioning in numerical linear algebra while tracking the near-singular continuation paths.

In contrast, a certified homotopy tracking algorithm makes a safe choice for a size of step in the continuation, guaranteeing that an approximate zero created in the next step is associated to the same homotopy path. If all computations are carried out in exact arithmetic the certified algorithm gives a rigorous proof of the results obtained. Our implementation of the certified tracking [BL1] is invoked by passing the option `Predictor=>Certified` to the function `track`. At the moment the linear algebra computations are carried out in the standard double precision, thus stopping one step short of rigorous certification. The work accomplishing this last step using exact linear algebra and the robust α -theory is underway [BL2].

FUTURE. The package in its current state provides the base for the higher-level numerical algebraic geometry routines, including robust computing of approximations to singular solutions using endgames [SW, §10.3] and deflation [LVZ1, LVZ2], irreducible decomposition of positive-dimensional varieties [SW, §15], and numerical primary decomposition [L2].

Homotopy tracking with arbitrary precision, which both *Bertini* and *PHCpack* implement to various extents, should be available in *Macaulay2* in the future. Although arbitrary precision floating point arithmetic is already in place in *Macaulay2* (via the MPFR library [FHL⁺]), a fast linear algebra implementation on the level of the engine is necessary for solution refinement and tracking near-singular paths.

Due to independence of tracking procedures for any collection of homotopy paths most of the algorithms in numerical algebraic geometry scale well if parallelized. Since the amount of data that needs to be communicated between CPUs is small in relation to the computational costs, the tasks can be distributed over heterogeneous clusters with slow interconnect. For the large problems these properties give homotopy continuation an edge over Gröbner bases techniques, which are currently the main engine of *Macaulay2*.

We will also explore ways to convert numerical results into symbolic results and vice versa. For instance, the algorithms in the package will be able to sample points on a component of a variety with an arbitrarily high precision, which could be used to construct the defining ideal of the component bypassing the need for (symbolic) primary decomposition.

ACKNOWLEDGEMENTS. This work would not be possible without support of the authors of *Macaulay2*. It was partially accomplished at the *Macaulay2* workshops at the American Institute of Mathematics and Colorado College as well as during the author's stays at MSRI in 2009 and Institut Mittag-Leffler 2011. The author would also like to thank the referees and editors for their careful remarks and suggestions. This work is partially supported by NSF grant DMS-0914802.

REFERENCES.

- [ABB⁺] E. Anderson, Z. Bai, C. Bischor, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK User's Guide*, 3rd ed., SIAM, Philadelphia, PA, 1999.

- [BHSW1] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler, *Bertini: software for numerical algebraic geometry*, available at www.nd.edu/~sommese/bertini.
- [BHSW2] ———, *Adaptive multiprecision path tracking*, SIAM J. Numer. Anal. **46** (2008), no. 2, 722 – 746.
- [BS] W. Baur and V. Strassen, *The complexity of partial derivatives*, Theoret. Comput. Sci. **22** (1983), no. 3, 317 – 330.
- [BL1] C. Betrán and A. Leykin, *Certified numerical homotopy tracking*, available at [arXiv:0912.0920v2](https://arxiv.org/abs/0912.0920v2) [math.NA].
- [BL2] ———, *Robust certified numerical homotopy tracking*, in preparation.
- [CG] J. Carnicer and M. Gasca, *Evaluation of multivariate polynomials and their derivatives*, Math. Comp. **54** (1990), no. 189, 231 – 243.
- [FHL⁺] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélicissier, and P. Zimmermann, *MPFR: a multiple-precision binary floating-point library with correct rounding*, ACM Trans. Math. Software **33** (2007), no. 2, Art. 13, 15.
- [M2] D.R. Grayson and M.E. Stillman, *Macaulay2, a software system for research in algebraic geometry*, available at www.math.uiuc.edu/Macaulay2/.
- [LLT] T.L. Lee, T.Y. Li, and C.H. Tsai, *Hom4ps-2.0: A software package for solving polynomial systems by the polyhedral homotopy continuation method*, available at hom4ps.math.msu.edu/HOM4PS_soft.htm.
- [L1] A. Leykin, *NAG4M2: Numerical Algebraic Geometry for Macaulay2*, available at people.math.gatech.edu/~aleykin3/NAG4M2.
- [L2] ———, *Numerical primary decomposition*, ISSAC 2008, ACM, New York, 2008, pp. 165 – 172.
- [LVZ1] A. Leykin, J. Verschelde, and A. Zhao, *Newton's method with deflation for isolated singularities of polynomial systems*, Theoret. Comput. Sci. **359** (2006), no. 1-3, 111 – 122.
- [LVZ2] ———, *Higher-order deflation for polynomial systems with isolated singular solutions*, Algorithms in algebraic geometry, IMA Vol. Math. Appl., vol. 146, Springer, New York, 2008, pp. 79 – 97.
- [SVW] A.J. Sommese, J. Verschelde, and C.W. Wampler, *Introduction to numerical algebraic geometry*, Solving polynomial equations, Algorithms Comput. Math., vol. 14, Springer, Berlin, 2005, pp. 301 – 335.
- [SW] A.J. Sommese and C.W. Wampler, *The numerical solution of systems of polynomials*, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2005.
- [V] J. Verschelde, *Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation*, ACM Trans. Math. Softw. **25** (1999), no. 2, 251 – 276.

RECEIVED : 2009-11-09

REVISED : 2011-03-03

ACCEPTED : 2011-05-20

leykin@math.gatech.edu : School of Mathematics, Georgia Institute of Technology, 686 Cherry Street Atlanta, GA 30332-0160 USA.