Computing Monodromy via Parallel Homotopy Continuation

Anton Leykin and Frank Sottile

ABSTRACT

Numerical homotopy continuation gives a powerful tool for the applied scientist who seeks solutions to a system of polynomial equations. Techniques from numerical homotopy continuation can also be useful in pure mathematical research. We discuss applications of a particular homotopy continuation idea that leads to probabilistic numerical algorithms for construction of monodromy groups.

One such application is used to analyze positive-dimensional solutions of polynomial systems. It is called the *monodromy breakup* method and partitions a *witness set* representing a positive-dimensional solution into irreducible components. Leykin and Jan Verschelde have implemented two parallel versions of this algorithm which show good speedups.

We use numerical homotopy continuation to compute Galois groups of certain enumerative geometric problems coming from Schubert calculus. The basic idea is similar: given a parametric family of 0-dimensional polynomial systems, we construct loops in the parameter space, follow the solution paths along these loops to obtain a permutation of the set of the solutions. These permutations are used to compute the subgroup of the full symmetric group that they generate.

Keywords

Polynomial homotopy continuation, monodromy, Schubert problems, Galois group, parallel processing, distributed computing.

1. IRREDUCIBLE DECOMPOSITION VIA MONODROMY BREAKUP

Polynomial equations are ubiquitous not only in pure and applied mathematics, but also in the other sciences and engineering. Indeed, the increasing computational power of modern computer systems gives resources to devote to theoretical models that are not necessarily linear.

Some basic information one would like about a system of polynomial equations is the structure of its solution set. When there are finitely many solutions (the solution set has dimension zero), this amounts to computing the solutions. Numerical homotopy methods are very effective here: they can approximate any solution to any degree of precision.

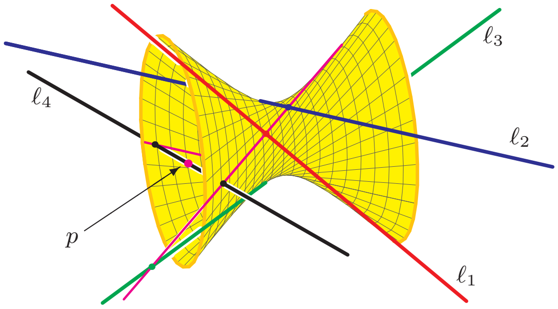
Numerical algebraic geometry [8, 9] represents positive-dimensional solution sets through *witness sets*, which consist of general points in the solution set. We simplify this representation by finding an *irreducible decomposition* of the solution set as an algebraic variety.

Monodromy breakup is an algorithm to do this. With it, witness points on the same irreducible component are connected by path tracking techniques applying the idea of monodromy, i.e, creating loops around singularities. Computing a *linear trace* for each component certifies the decomposition. Leykin and Verschelde [6] implemented a parallel version which performs well well on solution sets of relatively high degrees defined by systems of low degree polynomials. Another monodromy breakup algorithm was later developed and implemented [5]. This new algorithm works faster even on a single processor. On multiple processors, it avoids synchronization issues of the original algorithm, and therefore makes making fuller use of the resources.

Both implementations – the original version and the more efficient new one – are based on parallel PHCpack routines [4], which in turn use MPI for communication.

2. GALOIS GROUP COMPUTATION OF SCHUBERT PROBLEMS

The Schubert calculus [3] is a well-studied class of geometric problems involving linear subspaces. Here is the prototypical problem: how many lines in space meet four given lines? To answer this, note that three lines ℓ_1, ℓ_2, ℓ_3 lie on a unique doubly-ruled hyperboloid.



These lines lie in one ruling and the second ruling consisting of the lines meeting the given three lines. The fourth line ℓ_4 meets the hyperboloid in two points. Through each of these points there is a line in the second ruling, and these are the two lines meeting our four given lines. The *Galois group* is the group of permutations of the solutions which is generated by following loops in the space of lines $\ell_1, \ldots, \ell_4$. One such loop is given by rotating ℓ_4 about the point p. This interchanges the two solution lines, showing that the Galois group of this problem is the full symmetric group on two letters.

A general description of the basic problem of Schubert calculus and a symbolic-numerical approach for finding its solution is presented in [2]. Galois/monodromy groups for Schubert problems are discussed in [10] and for more general enumerative geometric problems in [1].

The application of numerical homotopy to Galois group computation is straightforward. We consider a parametric family of such problems $P(\mathbf{x}, \mathbf{a})$, numerically obtain the finite solution set $S_0 = S(\mathbf{a}_0)$ for one instance obtained by specialization to a generic set of parameters $\mathbf{a}_0$. Then we create random loops in the space of parameters (in this case flags) and numerically follow the solutions to get a permutation of S_0. Repeating this gives a list of permutations which generate a subgroup of the Galois group of our Schubert problem. We repeat this until either the full symmetric group is found, or we reach some other stopping criteria.

The initial implementation of this idea in Maple using the PHCmaple package [7] works for small (14 or fewer solutions) Schubert problems in Grassmannians of subspaces of $\mathbb{C}^6$. Using more resources – in particular, distributed resources – and lower-level programming languages will speed up the implementation making the computation for more complicated Schubert problems possible.

Similarly to the monodromy breakup procedures, our algorithm for Galois groups if Schubert problems is highly parallelizable. We are developing code involving parallel PHCpack to use in a project to perform the computation for all small (fewer than 100 solutions) Schubert problems on all small Grassmannian (involving subspaces in $\mathbb{C}^{10}$ or smaller vector spaces). In all, we expect to determine Galois groups of perhaps 10,000 Schubert problems.

3. REFERENCES

[1] J. Harris. Galois groups of enumerative problems. *Duke Math. J.*, 46:685–724, 1979.

[2] B. Huber, F. Sottile, and B. Sturmfels. Numerical Schubert calculus. *J. Symb. Comp.*, 26(6):767–788, 1998.

[3] S.L. Kleiman and Dan Laksov. Schubert calculus. *Amer. Math. Monthly*, 79:1061–1082, 1972.

[4] Anton Leykin, Verschelde Jan, and Yang Zhuang. Parallel homotopy algorithms to solve polynomial systems. In Nobuki Takayama and Andres Iglesias, editors, *Proceedings of ICMS 2006*, pages 225–234, 2006.

[5] Anton Leykin and Jan Verschelde. Decomposing solution sets of polynomial systems: A new parallel monodromy breakup algorithm. To appear in Int. Journal of Comp.Science and Eng.

[6] Anton Leykin and Jan Verschelde. Factoring pure dimensional solution sets of polynomial systems in parallel. In *Proceedings of the 2005 International Conference on Parallel Processing Workshops*, pages 173–180, 2005.

[7] Anton Leykin and Jan Verschelde. Interfacing with the numerical homotopy algorithms in PHCpack. In Nobuki Takayama and Andres Iglesias, editors, *Proceedings of ICMS 2006*, pages 354–360, 2006.

[8] A.J. Sommese, J. Verschelde, and C.W. Wampler. Introduction to numerical algebraic geometry. In A. Dickenstein and I. Emiris, editors, *Graduate School on Systems of Polynomial Equations: From Algebraic Geometry to Industrial Applications. 14-25 July 2003, Buenos Aires, Argentina*, pages 229–247. INRIA, 2003. A revised collection of the course notes is scheduled to be published by Springer–Verlag.

[9] Andrew J. Sommese and Charles W. Wampler, II. *The numerical solution of systems of polynomials.* World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2005.

[10] Ravi Vakil. Schubert induction. *Ann. of Math. (2)*, 164(2):489–512, 2006.

PASCO'07, July 27–28, 2007, London, Ontario, Canada.