

Polynomial Systems Arising From Discretizing Systems of Nonlinear Differential Equations

Andrew J. Sommese
University of Notre Dame
Notre Dame, Indiana
sommese@nd.edu

ABSTRACT

This article is an extended abstract of the ISSAC 2018 talk “Polynomial systems arising from discretizing systems of nonlinear differential equations” by Andrew Sommese.

CCS CONCEPTS

• **Mathematics of computing** → **Discretization; Nonlinear equations;**

KEYWORDS

discretizations; nonlinear differential equations; numerical solutions; polynomial systems

ACM Reference Format:

Andrew J. Sommese. 2018. Polynomial Systems Arising From Discretizing Systems of Nonlinear Differential Equations. In *ISSAC '18: 2018 ACM International Symposium on Symbolic and Algebraic Computation, July 16–19, 2018, New York, NY, USA*. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3208976.3209029>

1 INTRODUCTION

Systems of differential equations arise from applications throughout engineering and science. Discretizations of nonlinear differential equations often lead to systems of polynomials. These discretizations, which often have thousands of equations in thousands of variables, are sparse with a fair amount of structure. For example, finite difference discretizations with related grids resemble each other, and in particular many of the solutions of a given discretization may be associated with solutions of discretizations that are “finer.” The polynomial systems arising through discretization are a rich source of significant problems. In this talk, I give an overview of what I have learned working on these problems.

In §2, which might also be called “Truth in Advertising,” I discuss issues, which (though for the most part standard for applied mathematicians) might not be expected by those used to exact computation and the certainty afforded by proofs.

In §3, I discuss some of the approaches we have used to investigate the polynomial systems that arise in discretizing systems of differential equations.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ISSAC '18, July 16–19, 2018, New York, NY, USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5550-6/18/07...\$15.00

<https://doi.org/10.1145/3208976.3209029>

It is worth noting (and satisfying to this algebraic geometer) that many systems that at first appear “nonalgebraic” can be converted to (and often first arose from) systems of differential equations with polynomial discretizations. For example if e^x occurs in an equation, we can replace e^x by a new quantity v and add an equation

$$\frac{dv}{dx} - v = 0.$$

This same adjustment applies to the occurrence of many classical functions (such as Bessel functions), which satisfy “algebraic differential equations.” Of course, in practice this is not necessarily a good fix since it significantly increases the number of variables and equations in discretizations.

I would like to here thank my many collaborators on this work and in particular Bei Hu, Wenrui Hao, and Jonathan Hauenstein.

2 RULES OF THE GAME

Relatively few systems of nonlinear differential may be solved exactly, and must instead be solved numerically. As a simple example (see [17] for more realistic examples), consider computing a function $y(x)$ on $[0, 1]$ that satisfies

$$y'' + p(y) = 0 \quad \text{with } y(0) = a \quad \text{and } y(1) = b, \quad (1)$$

where $p(y)$ is a polynomial and a and b are real numbers. For each positive integer N , choose $N + 2$ points $x_j = jh$ where $h = 1/(N + 1)$ and $j = 0, \dots, N + 1$. We would like to find numbers y_1, \dots, y_N such that there is a solution $y(x)$ of Eq.1 with y_j appropriately close to $y(x_j)$. One way to approach this is to set $y_0 = a$ and $y_{N+1} = b$ and try to solve the polynomial system

$$\frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} + p(y_i) = 0 \quad j = 1, \dots, N. \quad (2)$$

If $p(y) = y$, then Eq.2 is a system on linear equations with a unique solution (y_1, \dots, y_N) and for large enough N , there is a constant $B > 0$ such that $|y(x_j) - y_j| \leq Bh^2$. Going from a solution of Eq.1 in this case to the conclusion that the solution of Eq.2 satisfies $|y(x_j) - y_j| \leq Bh^2$ is straightforward. It follows from the fact that given a function $y(x)$ with enough differentiability on $[0, 1]$, there is a positive constant C such that for all sufficiently large N and any $x \in (0, 1)$

$$\left| y''(x) - \frac{y(x-h) - 2y(x) + y(x+h)}{h^2} \right| \leq Ch^2.$$

Going the other way takes work (not much in the case of Eq.1 with $p(y)$ linear). For a wide variety of linear ordinary and partial differential equations, i.e., differential equations with y and its derivatives entering linearly, this sort of procedure works very well. The elementary text [21] is a good place for general information.

Most systems of differential equations are not linear. Discretization still makes sense and solutions of the original differential equations are “close” to solutions of the discretized equations, but there are some serious issues with going from the solutions of the discretized equations to the approximate solutions of the original system of differential equations. Let’s enumerate them.

- (1) Even for ordinary differential equations, it is often unknown in the nonlinear case how many solutions exists or even whether any solution exists.
- (2) The discretization for any reasonable number of gridpoints often has more solutions than estimates of the number of atoms in the universe. (In fact, more than the square of the estimates.)
- (3) It is not clear which solutions of the discretization are approximate solutions of the original system of differential equations.
- (4) the discretization is numerically very difficult to work with when using double precision.

The first point is an ever present issue. As noted in [17], the system of differential equations in a simple two-dimensional tumor growth model [11] is more complicated than the classical Hele-Shaw problem with surface tension (with many works devoted to it in the literature, i.e., searching Hele-Shaw on the title alone on MathSciNet returns 517 entries).

For traditional polynomial systems arising in applications areas such as Theoretical Kinematics, sorting through all solutions for those that are physically realistic is standard. For nonlinear differential equations, this is rarely possible. For example, a discretization of a simple two-dimensional Lotka-Volterra population model with diffusion, that consists of two partial differential equations ([17, §1.1] and [18, §9.4]), has is a polynomial system with $2g$ equations of total degree 2^{2g} with 2^g complex solutions. A relatively coarse 20×20 grid gives $g = 324$ for a system with $\approx 3.4 \cdot 10^{97}$ solutions. This rules out finding all solutions as a viable option.

I know no absolutely certain solution for the third point about whether a solution of the discretization corresponds to a solution of the system of differential equations. What can be done is to interpolate to extend the solution on one grid to a finer grid and then check if the extension is near a solution of the analogous discretized system for the finer grid.

The last point about needing double precision is key. At typical nonsingular solutions, condition numbers of Jacobian matrices of the polynomial system in the range of 10^9 and larger as the grid size grows are common, e.g., [12]. If we restrict ourselves to double precision, the linear algebra is not trustworthy and continuation sometimes fails. Moreover deciding whether a numerical scheme is converging (and what the order of convergence is) seems hopeless with just double precision, e.g., see [19, Table 1]. In our work we used Bertini [4, 5] which uses adaptive multiprecision algorithms based on [6?, 7].

3 COMPUTING SOLUTIONS

The goal is to find solutions of the systems of differential equations. In light of the points raised in the last section, what can be done?

There are several approaches we have followed.

The simplest, which might be labeled the *brute force method* is to compute all solutions for a coarse grid and sort through them for possible solutions to the system of differential equation. We tried with some success for a system of four ordinary differential equations arising in a pattern formation problem [9]. In 2011, a discretization of this system with ten gridpoints took about eleven hours on a 200 core cluster. Each additional gridpoint add four new equations and four new variables, with the count of paths which need to be followed increased by a multiple of five. Software and hardware have improved since then, but the exponential growth of the number of solutions as the number of gridpoints increase means that at most a few more nodes could be added.

In [1] a *filtering* approach was proposed (see also [2, 3]). To understand the basic idea in the case of a single ordinary differential that has a discretization into a system of polynomials $F_n(y) = 0$ for n gridpoints (x_1, \dots, x_n) at which we want to compute solutions (y_1, \dots, y_n) giving approximations of solutions of the differential equation at gridpoints:

- (1) start with a set S_{n_0} of isolated solutions of $F_{n_0}(y_1, \dots, y_{n_0}) = 0$ for some n_0 ;
- (2) construct a homotopy $H_{n_0}(y_1, \dots, y_{n_0+1}, t) = 0$ with $H_{n_0}(y, 0) = F_{n_0+1}(y)$ and

$$H_{n_0}(y, 1) = \begin{pmatrix} F_{n_0}(y_1, \dots, y_{n_0}) \\ g_{n_0}(y_1, \dots, y_{n_0+1}) \end{pmatrix}$$

- for a polynomial $g_{n_0}(y_1, \dots, y_{n_0+1})$;
- (3) for each solution s of S_{n_0} , compute the solutions (s, y_{n_0+1}) of $g_{n_0}(s, y_{n_0+1}) = 0$ and denote the union of these solutions when running over all $s \in S_{n_0}$ by T_{n_0+1} ;
- (4) use $H_{n_0}(y, t) = 0$ to compute the continuations to $t = 0$ of the solutions T_{n_0+1} of $H_{n_0}(y, 1) = 0$ and denote the set of these solutions of $H_{n_0}(y, t) = 0$ by \widehat{S}_{n_0+1} ; and
- (5) remove from \widehat{S}_{n_0+1} those solutions which are “far” from being discretizations of solutions of the differential equation and call the remaining solutions S_{n_0+1} .

Repeating this procedure, we can hopefully compute for large n the solutions of $F_n(y) = 0$ corresponding the solutions of the differential equation.

There are two main problems with this procedure. First, it is not easy to apply it partial differential equations. Second, the filtering procedure going from \widehat{S}_{n_0+1} to S_{n_0+1} is not obvious, e.g., complex solutions in S_{n_0} can turn into real solutions of S_{n_0+1} . This said, there is a lot of flexibility in this approach, and it can be quite effective when applied to ordinary differential equations.

A third (and very powerful) approach [14] called the *bootstrap method* (related in spirit to the filtering approach and to domain decomposition methods in numerical partial differential equations) is to use the geometry of the chosen grid to find a homotopy to the given discretization from a nearby easier-to-solve system. For example, consider Eq.2

$$\begin{aligned} a - 2y_1 + y_2 + h^2 p(y_1) &= 0 \\ y_1 - 2y_2 + y_3 + h^2 p(y_2) &= 0 \\ y_2 - 2y_3 + b + h^2 p(y_3) &= 0 \end{aligned} \quad (3)$$

with $N = 3$, which consists of 3 polynomials in the variables y_1, y_2, y_3 with $y_0 = a$ and $y_4 = b$ for the constants a, b . A choice of

a nearby easier-to-solve system could consist of

$$\begin{aligned} a - 2y_2 + b + (2h)^2 p(y_2) &= 0 \\ a - 2y_1 + y_2 + h^2 p(y_1) &= 0 \\ y_2 - 2y_3 + b + h^2 p(y_3) &= 0. \end{aligned} \quad (4)$$

Note the first equation of the simpler system is just Eq.2 with the nodes x_0, x_2, x_4 . After solving this for y_2 , the second equation of the simpler system is Eq.2 with the nodes x_0, x_1, x_2 on $[x_0, x_2]$ and the third equation is just Eq.2 with the nodes x_2, x_3, x_4 on $[x_2, x_4]$. This will work in the same way when $N + 1 = km$ is a product of two integers $k \geq 2$ and $m \geq 2$. The simpler system would consist of

- (1) Eq.2 for the nodes $0 = x_0, x_m, x_{2m}, \dots, x_{km} = b$;
- (2) Eq.2 on $[x_k, x_{(k+1)j}]$ for the nodes $x_{kj}, x_{k(j+1)}, \dots, x_{k(j+k)}$ for $j = 0, \dots, m - 1$.

A fourth approach is to take a solution we know, which depends on a parameter, and change the parameter until a bifurcation occurs, and then follow the solution along a different branch occurring at the given bifurcation. This has been a surprisingly successful approach applied to a sequence of successively more complicated systems of differential equations arising in tumor growth (see [10–13, 16]). All of these examples have very nice bifurcations, i.e., the tangent cones consist of two reduced lines. A numerical algorithm to compute the tangent cone in general would be useful. For more complicated bifurcations, something along the lines of a local irreducible decomposition computed off local information is needed: the result [8] depends on global information that is not available on the large polynomial systems arising from differential equations.

Finally, I would like to mention a surprising success [15] that came from applying algebraic geometric ideas to the numerical solution of polynomial systems. Hyperbolic conservation laws are a class of differential equations that arise in many places often with discontinuous solutions, e.g., shockwaves from explosions. For these problems, there are often positive dimensional sets of solutions with the physically realistic time-invariant solution picked out by minimizing auxiliary quantities. For many of these a theoretical way to pick out the realistic solution is to add a term ϵD , where D is something like a Laplacian to the system. This perturbed system has a unique solution, which “theoretically” goes to the true solution when ϵ goes to zero. I say theoretically, because the discretization is very singular when $\epsilon = 0$, and the standard numerical method to find the physically realistic solution is time-marching. Of course, for numerical algebraic geometers, seeing a parameter, i.e., ϵ in this case, going to zero, the first thought is *use an endgame*. So as a “proof of concept,” we used the Cauchy Endgame [20] in Bertini [5] to compute the limit when $\epsilon \rightarrow 0$. This led to [15]. The surprise is that the endgame applied to the theoretical approach (not thought of as practical numerically) is remarkably better than the standard numerical approach.

ACKNOWLEDGMENTS

The work is supported in part by the National Science Foundation under Grant No. NSF ACI-1440607.

REFERENCES

- [1] E. L. Allgower, D. J. Bates, A. J. Sommese, and C. W. Wampler. 2010. Solution of polynomial systems derived from differential equations. *Computing* 76 (2010), 1–10.
- [2] E. L. Allgower, S. G. Cruceanu, and S. Tavener. 2009. Application of numerical continuation to compute all solutions of semilinear elliptic equations. *Adv. Geom.* 9, 3 (2009), 371–400. <https://doi.org/10.1515/ADVGEOM.2009.020>
- [3] E. L. Allgower, S. G. Cruceanu, and S. Tavener. 2009. Turning points and bifurcations for homotopies of analytic maps. In *Interactions of classical and numerical algebraic geometry*. Contemp. Math., Vol. 496. Amer. Math. Soc., Providence, RI, 1–10. <https://doi.org/10.1090/conm/496/09715>
- [4] D. J. Bates, J. D. Hauenstein, A. J. Sommese, and Charles W. Wampler. [n. d.]. Bertini: Software for Numerical Algebraic Geometry. Available at bertini.nd.edu with permanent doi: [dx.doi.org/10.7274/R0H41PB5](https://doi.org/10.7274/R0H41PB5).
- [5] D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler. 2013. *Numerically solving polynomial systems with Bertini*. Software, Environments, and Tools, Vol. 25. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA. xx+352 pages.
- [6] D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler, II. 2008. Adaptive multiprecision path tracking. *SIAM J. Numer. Anal.* 46, 2 (2008), 722–746. <https://doi.org/10.1137/060658862>
- [7] D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler, II. 2009. Stepsize control for path tracking. In *Interactions of classical and numerical algebraic geometry*. Contemp. Math., Vol. 496. Amer. Math. Soc., Providence, RI, 21–31. <https://doi.org/10.1090/conm/496/09717>
- [8] D. A. Brake, J. D. Hauenstein, and A. J. Sommese. 2016. Numerical Local Irreducible Decomposition. In *Mathematical Aspects of Computer and Information Sciences*, I. S. Kotsireas, S. M. Rump, and C. K. Yap (Eds.). Springer International Publishing, Cham, 124–129.
- [9] W. Hao, J. D. Hauenstein, B. Hu, Y. Liu, A. J. Sommese, and Y.-T. Zhang. 2011. Multiple stable steady states of a reaction-diffusion model on zebrafish dorsal-ventral patterning. *Discrete Contin. Dyn. Syst. Ser. S* 4, 6 (2011), 1413–1428. <https://doi.org/10.3934/dcdss.2011.4.1413>
- [10] W. Hao, J. D. Hauenstein, B. Hu, Y. Liu, A. J. Sommese, and Y.-T. Zhang. 2012. Bifurcation for a free boundary problem modeling the growth of a tumor with a necrotic core. *Nonlinear Anal. Real World Appl.* 13, 2 (2012), 694–709. <https://doi.org/10.1016/j.nonrwa.2011.08.010>
- [11] W. Hao, J. D. Hauenstein, B. Hu, Y. Liu, A. J. Sommese, and Y.-T. Zhang. 2012. Continuation along bifurcation branches for a tumor model with a necrotic core. *J. Sci. Comput.* 53, 2 (2012), 395–413. <https://doi.org/10.1007/s10915-012-9575-x>
- [12] W. Hao, J. D. Hauenstein, B. Hu, T. McCoy, and A. J. Sommese. 2013. Computing steady-state solutions for a free boundary problem modeling tumor growth by Stokes equation. *J. Comput. Appl. Math.* 237, 1 (2013), 326–334. <https://doi.org/10.1016/j.cam.2012.06.001>
- [13] W. Hao, J. D. Hauenstein, B. Hu, and A. J. Sommese. 2011. A three-dimensional steady-state tumor system. *Appl. Math. Comput.* 218, 6 (2011), 2661–2669. <https://doi.org/10.1016/j.amc.2011.08.006>
- [14] W. Hao, J. D. Hauenstein, B. Hu, and A. J. Sommese. 2014. A bootstrapping approach for computing multiple solutions of differential equations. *J. Comput. Appl. Math.* 258 (2014), 181–190. <https://doi.org/10.1016/j.cam.2013.09.007>
- [15] W. Hao, J. D. Hauenstein, C.-W. Shu, A. J. Sommese, Z. Xu, and Y.-T. Zhang. 2013. A homotopy method based on WENO schemes for solving steady state problems of hyperbolic conservation laws. *J. Comput. Phys.* 250 (2013), 332–346.
- [16] W. Hao, B. Hu, and A. J. Sommese. 2013. Cell Cycle Control and Bifurcation for a Free Boundary Problem Modeling Tissue Growth. *Journal of Scientific Computing* 56, 2 (01 Aug 2013), 350–365. <https://doi.org/10.1007/s10915-012-9678-4>
- [17] W. Hao, B. Hu, and A. J. Sommese. 2014. Numerical algebraic geometry and differential equations. In *Future vision and trends on shapes, geometry and algebra*. Springer Proc. Math. Stat., Vol. 84. Springer, London, 39–53. https://doi.org/10.1007/978-1-4471-6461-6_3
- [18] J. D. Hauenstein, A. J. Sommese, and C. W. Wampler. 2011. Regeneration homotopies for solving systems of polynomials. *Math. Comp.* 80 (2011), 345–377.
- [19] A. E. Lindsay, W. Hao, and A. J. Sommese. 2015. Vibrations of thin plates with small clamped patches. *Proc. A* 471, 2184 (2015), 20150474, 19. <https://doi.org/10.1098/rspa.2015.0474>
- [20] A. P. Morgan, A. J. Sommese, and C. W. Wampler. 1991. Computing singular solutions to nonlinear analytic systems. *Numer. Math.* 58, 7 (1991), 669–684.
- [21] J. Stoer and R. Bulirsch. 2002. *Introduction to numerical analysis* (third ed.). Texts in Applied Mathematics, Vol. 12. Springer-Verlag, New York. xvi+744 pages. Translated from the German by R. Bartels, W. Gautschi and C. Witzgall.