RESEARCH STATEMENT

MICHAEL BURR

When I describe my research in a sound bite, I call my research program "Geometry+." What I mean by this is that, in my research, I consider problems in mathematics and the sciences, and I study the underlying geometry of these problems. Therefore, even though the applications differ (and hence the work is published in seemingly unrelated journals), the underlying tools, techniques, and ways of looking at the problem flow through and connect all of my work. What excites me the most in my work is when disparate areas of mathematics (surprisingly) appear side-by-side to complement each other.

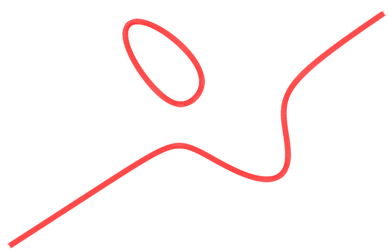
For example, consider the two papers [7] and [8]. The first paper is in probability and statistics, and it improves the theoretical convergence rate of the half-space depth of a sample to that of the underlying distribution. The second paper, on the other hand, is a symbolic computation paper bounding the computational complexity of a family of algorithms for approximating curves and surfaces. Although the applications vary widely, in the heart of the proofs in both papers, a very similar technique appears: a high-dimensional problem is reduced to a family of one-dimensional problems by looking at all possible projections or restrictions to lines through a point. In other words, similar geometric approaches are successful in two seemingly unrelated papers.

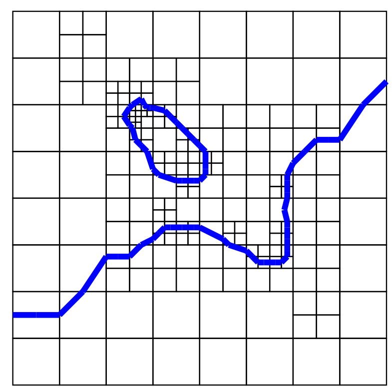
We may also consider the two papers [26] and [12]. The first paper is in symbolic computation and numerical algebraic geometry and it provides a certified homotopy continuation tracker to compute the roots of a univariate polynomial. On the other hand, the second paper deals with computability in dynamical systems, i.e., what quantities can possibly computed for a dynamical system, where mixing amplifies all numerical errors. Although the applications are very different, the type of question being considered is very similar between these two papers: in particular, the goal in both papers is to provide algorithms which produce results with error estimates even when the input is approximate. Since these two papers are asking similar questions, the details behind the scenes between these papers are highly related.

In this statement, I describe my main research program, consisting of finding algorithms for solving systems of polynomials and computing their complexity. This research program continues to be an active program for me. Following, I briefly describe some of the other projects that I have worked on and indicate some of the connections to my main research program.

Solving Polynomial Systems. Computing the solutions to systems of polynomial equations is often called a fundamental problem in both computer algebra [23] and numerical algebraic geometry [2]. Work coming from these two communities has distinct advantages and disadvantages. For example, algorithms in computer algebra are guaranteed to be correct, but they are notorious for being slow, e.g., computation of Gröbner bases is well-known to be exponential in the size of the input [21]. Algorithms from the numerical algebraic geometry community are often fast and can handle large systems, but their output is not guaranteed to be correct, e.g., most correctness statements can be proved only in the case of infinite precision, which is not possible on a real computer. Recent work has focused on hybrid methods which combine the positive aspects of both symbolic and numeric methods. In other words, the idea is to use fast evaluation-based methods to construct efficient algorithms, while using a few tools from symbolic algebra to generate a correctness statement.

Isotopic Approximation. My work in [6] develops such a hybrid algorithm. This paper provided the first algorithm whose computations were completely numerical, i.e., based on the evaluation of a function and its derivatives at a point, that produces a topologically correct approximation to a (potentially unbounded) real planar curve, which may have singularities. More precisely, given a polynomial $f(x, y)$ in two variables, the goal is to correctly approximate the set $V = \{(x, y) \in \mathbb{R}^2 : f(x, y) = 0\}$ for any polynomial f. The correctness guarantee means that there is a continuous deformation of $\mathbb{R}^2$, which transforms our approximation into V, e.g., the singularities match up and no components are missed or mistakenly connected. For example, the algorithm may produce the following blue curve, which approximates the red curve:





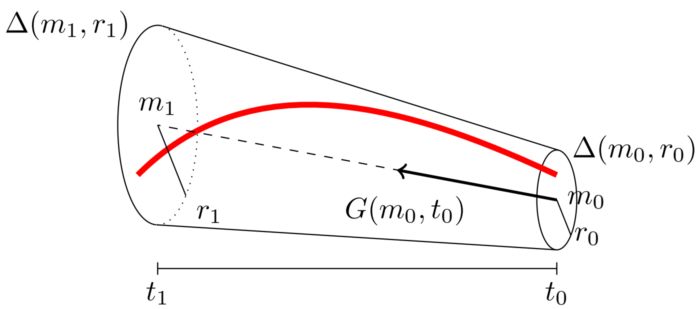
The algorithm in our work for computing these curves is based on the Plantinga and Vegter algorithm [22], which considered only the smooth and bounded case. Our contributions were to extend the algorithm to the singular and unbounded case, which is significantly more complex (as our work was the first to succeed in this task).

Our algorithm is a hybrid symbolic-numeric algorithm and it uses techniques and ideas from both fields to achieve its results. On the numeric side, the main computations which are made are to evaluate f and its derivatives at dyadic points; in other words, the computations are to substitute values into f and perform ordinary arithmetic. In order to achieve the correctness statement, we use interval arithmetic and separation bounds. Interval arithmetic performs set-wise operations on intervals in order to develop conservative estimates on the values of functions. In other words, the sum of two intervals is the interval $[a, b] + [c, d] = [a + c, b + d]$, and, more generally, for any operation $\odot$, we define $[a, b] \odot [c, d] = \{x \odot y : x \in [a, b], y \in [c, d]\}$. Using interval arithmetic and function evaluation, we can easily construct over-approximations of functions, i.e., for any rectangle $J \subseteq \mathbb{R}^2$, the interval approximation to f is $\Box f$, and $\Box f(J) \supseteq f(J) = \{f(x, y) : (x, y) \in J\}$. With such conservative estimates, we can be sure that our algorithm does not miss any small interesting features of V. The symbolic-based tool that we use are separation bounds. Separation bounds are positive lower bounds on the distance between interesting features of the zero set of a function. They are developed by bounding the possible sizes of values that may arise in a symbolic computation.

In this case, apply these bounds to get a lower bound on the numerical precision necessary to find all interesting features.

Homotopy Continuation. Homotopy continuation has become a widely popular technique for solving systems of polynomials in the numeric algebraic geometry community. The idea behind homotopy continuation is to choose a polynomial start system whose solutions are known and to slowly deform the system into the desired target system. During this deformation, the roots of the polynomial system are tracked (using predictor-corrector techniques) as the system of polynomials changes. At the end of this process, the roots of the start system have been transformed into the roots of the desired system. This technique has been popular because it is easy to use and has been successful on very large systems, including systems for which symbolic methods have no hope of solving. The challenge with this method is that it is not certified, i.e., it may potentially produce errors due to very high condition numbers, path jumping, or other numerical issues. There has been some work to develop a certified homotopy continuation, see, e.g., [4]. However, such approaches are typically very conservative and require extremely small predictor steps.

My work [26] presents a new predictor-corrector method that is comparable to the best (noncertified) homotopy continuation methods, but certifies the solution path. In other words, our algorithm takes nearly the same number of steps as the most popular homotopy continuation solvers currently available. In order to certify our results, our method constructs a tubular neighborhood that is guaranteed to contain the solution path as it varies from the start system to the target system. The following figure depicts a step in such a tubular neighborhood and the red curve represents the solution path.



The key to this work was to replace the standard predictor-corrector and certification methods with alternatives. In particular, the tests are based on interval arithmetic along with the combination of the Pellet test and Graeffe iteration for confirming the existence of roots, see [3]. The main advantage of these tests over previous approaches is that the tests are less restrictive. For example, certification in [4] is based on alpha theory [25]. Unfortunately, alpha theory is known to be extremely conservative, which forces an unnecessarily large number of steps. Our work is able to overcome this drawback. Moreover, we have released this software in the univariate case, which can be found at `https://cs.nyu.edu/exactunderprogs/homotopyPath`.

Continuous Amortization. A subdivision-based algorithm is one that iteratively and adaptively subdivides an initial domain until some test succeeds on every subdomain. For example, the squares appearing in the curve approximation algorithm above depict a subdivision of the plane. These types of algorithms are common in symbolic algebra and are frequently used in other sciences, but their complexity is less well understood. In particular, these algorithms are adaptive, performing more subdivisions near difficult features while quickly terminating near easier features. Therefore, a tight complexity analysis must respond to this adaptivity.

My series of papers [16], [15], and [8] develop the technique of continuous amortization and apply this tool to calculate the complexity of various subdivision-based algorithms. The simplest case of continuous amortization can be described in one dimension: we start an interval I and iteratively bisect subintervals J of I whenever they fail a test. We define a local size bound as a function $F : I \to \mathbb{R}_+$ with the property that for each $x \in I$, $F(x)$ is smaller than the width of the smallest subinterval J containing x which must be subdivided, i.e.,

$$F(x) \leq \min_{\substack{J \ni x \\ J \text{ not terminal}}} \text{width}(J).$$

A local size bound represents a local condition number, and it quantifies the worst possible behavior possible at the point x. In the case of a bisection algorithm, we prove that the number of regions produced by the bisection algorithm is bounded above by the maximum of 1 and the integral

$$\int_I \frac{2dx}{F(x)}.$$

In this way, we turn a complicated problem into an integral, which may be easier to evaluate. The idea behind this integral is that the function $F(x)$ is chosen so that on each terminal interval, the integral over that interval is at least 1. Therefore, integrating over a partition counts the number of regions in that partition.

Continuous amortization is a unifying technique that can be applied to compute the complexity of many subdivision-based algorithms. Before the development of this technique, each subdivision-based algorithm was analyzed separately using somewhat ad-hoc methods to calculate the complexity. With the development of continuous amortization, a single technique can be used across most instances of subdivision-based methods.

In the first paper [16] in this sequence, we introduce and apply continuous amortization to a simple evaluation-based root isolation algorithm. This resulted in a state-of-the-art complexity bound for this algorithm, which was fairly straight-forward to compute. In the second paper [15], the general theory of continuous amortization is further developed and extended in many directions, e.g., to apply to higher dimensions or bit complexity computations. In this paper, I was able to reprove or improve the results of several other papers using one technique and in much simpler ways. Therefore, this one paper replaces many of the results in the papers [18], [19], and [24].

Finally, in the third paper [8], we are able to compute the complexity of the curve and surface approximation algorithm developed by Plantinga and Vegter [22], described above. In [22], the authors claimed that the technique was efficient in practice, but provided no complexity analysis. Moreover, this subdivision-based algorithm had never been analyzed (it is too complicated for the previously developed ad-hoc techniques). This situation made it ripe for analysis with continuous amortization.

Using continuous amortization, we were able to derive an expression for the complexity of this algorithm in terms of the geometry of the input. In particular, we were able to explicitly quantify the number of regions produced by the bisection algorithm in terms of the distance between solutions to $f(x, y) = 0$ and $\nabla f(x, y) = 0$. While this is an unusual expression to appear in a complexity statement, it brings to light how the hidden geometry in the shapes of the solutions to $f(x, y) = 0$ affect the algorithm. Moreover, we were able to show that, in the worst case, the algorithm did not have polynomial complexity and we found an instance when this occurs. On the other hand, our continuous amortization integral allowed us to show that many applications of this algorithm were polynomial complexity. This paper is particularly interesting because the proof required contributions from many different fields, including symbolic algebra, Fourier analysis, real algebraic geometry, complexity of algorithms, and computational geometry.

Other Projects. In this section, I describe some of the other projects on which I have worked. Due to space limitations, I do not describe them in great detail, but I do try to highlight some of the connections to my main research program.

In my papers [12] and [13], I study the computability properties of dynamical systems. A number or function is computable if it can be approximated arbitrarily well via an algorithm. In these papers, we provide specific conditions that indicate the computability of features of interest to dynamicists. In addition, we provide examples of noncomputability, i.e., quantities for which it is impossible to write a program to compute with any sort of confidence in the results. This work is the first to prove computability for an entire family of dynamical systems.

The ideas behind these questions are concerned with whether we can trust the result of a computation. This is precisely the same motivation as for the problem above for certifying the results of hybrid symbolic-numeric computations. Additionally, the main object of interest in [12] is a rotation set, which is a convex object in $\mathbb{R}^n$; we compute this set, when possible, using techniques from computational geometry, and methods from this field are critical in algorithms for approximating varieties.

In my papers [17], [1], [11], and [7], I study the the geometric properties of data depth functions. Data depth functions are nonparametric statistics that quantify the centrality of a query point with respect to a data set or distribution. In these papers, I study the geometric and computational properties of these depth functions. As discussed above, several of the proof techniques used in these papers surprisingly also appear in my symbolic computation papers. In fact, the application of the projection technique [7] motivated me to consider the restriction approach in [9].

In my other papers, [5], [14], [20], and [10], although I do not describe them here, the same patterns appear. In other words, while these may appear to be somewhat stand-alone projects by topic, every single one of these projects involves geometry, either algebraic or computational geometry, and the techniques that appear, repeat throughout the rest of my work.

References

[1] T. Abbott, M. Burr, M. Chan, E. Demaine, M. Demaine, J. Hugg, D. Kane, S. Langerman, J. Nelson, E. Rafalin, K. Seyboth, and V. Yeung. Dynamic ham-sandwich cuts in the plane. *Computational Geometry: Theory and Applications*, 42(5):419–428, 2009.

[2] Daniel J. Bates, Jonathan D. Hauenstein, Andrew J. Sommese, and Charles W. Wampler. *Numerically solving polynomial systems with Bertini*, volume 25 of *Software, Environments, and Tools*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2013.

[3] Ruben Becker, Michael Sagraloff, Vikram Sharma, and Chee Yap. A near-optimal subdivision algorithm for complex root isolation based on the pellet test and newton iteration. *Journal of Symbolic Computation*, 86:51 – 96, 2018.

[4] Carlos Beltrán and Anton Leykin. Certified numerical homotopy tracking. *Experimental Mathematics*, 21(1):69–83, 2012.

[5] M. Burr, A. Cheng, R. Coleman, and D. Souvaine. An intuitive approach to measuring protein surface curvature. *PROTEINS: Struture, Function, and Bioinformatics*, 61:1068–1074, 2005.

[6] M. Burr, S. Choi, B. Galehouse, and C. Yap. Complete subdivision algorithms II: isotopic meshing of general algebraic curves. *Journal of Symbolic Computation*, 47(2):153–166, 2012.

[7] M. Burr and R. Fabrizio. Error probabilities for halfspace depth. *Statistics & Probability Letters*, 124:33–40, 2017.

[8] M. Burr, S. Gao, and E. Tsigaridas. The complexity of the plantinga-vegter curve and surface approximation algorithms. In *Proceedings of the 42nd International Symposium on Symbolic and Algebraic Computation*, pages 61–68. Association for Computing Machinery, 2017.

[9] M. Burr, S. Gao, and E. Tsigaridas. The complexity of subdivision for diameter-distance tests. Technical Report arXiv:1801.05864 [cs.SC], arXiv, 2018.

[10] M. Burr, F. Knoll, and S. Gao. Optimal bounds of johnson-lindenstrauss transformations. *Journal of Machine Learning Research*, 2018.

[11] M. Burr, E. Rafalin, and D. Souvaine. Dynamic maintenance of half-space depth for points and contours. Technical Report arXiv:1109.1517 [cs.CG], arXiv, 2011.

[12] M. Burr, M. Schmoll, and C. Wolf. On the computability of rotation sets and their entropies. *Ergodic Theory and Dynamical Systems*, 2018.

[13] M. Burr and C. Wolf. Computability at zero temperature. Technical Report arXiv:1809.00147 [math.DS], arXiv, 2018.

[14] Michael A. Burr. Asymptotic purity for very general hypersurfaces of $\mathbb{P}^n \times \mathbb{P}^n$ of bidegree (k, k). *Central European Journal of Mathematics*, 10(2):530–542, 2012.

[15] Michael A. Burr. Continuous amortization and extensions: with applications to bisection-based root isolation. *Journal of Symbolic Computation*, 77:78–126, 2016.

[16] Michael A. Burr and Felix Krahmer. `SqFreeEVAL`: an (almost) optimal real-root isolation algorithm. *Journal of Symbolic Computation*, 47(2):153–166, 2012.

[17] Michael A. Burr, Eynat Rafalin, and Diane L. Souvaine. Simplicial depth: an improved definition, analysis, and efficiency for the finite sample case. In *Data depth: robust multivariate analysis, computational geometry and applications*, volume 72 of *DIMACS Series on Discrete Mathematics Theoretical Computer Science*, pages 195–209. American Mathematical Society, Providence, RI, 2006.

[18] Zilin Du, Vikram Sharma, and Chee K. Yap. Amortized bound for root isolation via Sturm sequences. In *Symbolic-numeric computation*, Trends Math., pages 113–129. Birkhäuser, Basel, 2007.

[19] Arno Eigenwillig, Vikram Sharma, and Chee K. Yap. Almost tight recursion tree bounds for the Descartes method. In *ISSAC 2006*, pages 71–78. ACM, New York, 2006.

[20] D. Lipman and M. Burr. Quadratic-monomial generated domains from mixed signed, directed graphs. *International Journal of Algebra and Computation*, 2018.

[21] Ernst W. Mayr and Albert R. Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. *Advances in Mathematics*, 46(3):305–329, 1982.

[22] S. Plantinga and G. Vegter. Isotopic meshing of implicit surfaces. *The Visual Computer*, 23:45–58, 2007.

[23] Michael Sagraloff and Kurt Mehlhorn. Computing real roots of real polynomials. *Journal of Symbolic Computation*, 73:46 – 86, 2016.

[24] Michael Sagraloff and Chee K. Yap. A simple but exact and efficient algorithm for complex root isolation. In *ISSAC 2011—Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation*, pages 353–360. ACM, New York, 2011.

[25] Steve Smale. Newton's method estimates from data at one point. In *The merging of disciplines: new directions in pure, applied, and computational mathematics (Laramie, Wyo., 1985)*, pages 185–196. Springer, New York, 1986.

[26] J. Xu, M. Burr, and C. Yap. An approach for certifying homotopy continuation paths: Univariate case. In *Proceedings of the 43nd International Symposium on Symbolic and Algebraic Computation*, pages 399–406. Association for Computing Machinery, 2018.