Root Finding with Chebyshev Polynomials in Two Dimensions

Lucas Bouck and Ian Bell

George Mason University, Fairfax, VA, U.S.A. and National Institute of Standards and Technology, Boulder, CO, U.S.A

Summary

We want to find all isolated, common roots of $f, g : \mathbb{R}^2 \to \mathbb{R}$ in a bounded domain. We replicate the root finding algorithm found in [5], which approximates f, g with Chebyshev polynomials and finds the roots of the polynomials. We are working to extend the algorithm to non-rectangular domains.

Background, Applications, and Motivation

Background and Applications

- `ChebTools` [1] is a C++ library developed by Ian Bell
- Inspired by the Matlab library `chebfun` [3]
- Determine thermodynamic properties of a fluid [4]

Motivating Example

$f(x,y) = x^2 - y^2, g(x,y) = \sin\left(\pi\sqrt{x^2+y^2}\right)$ with the domain $[-1,1]^2$

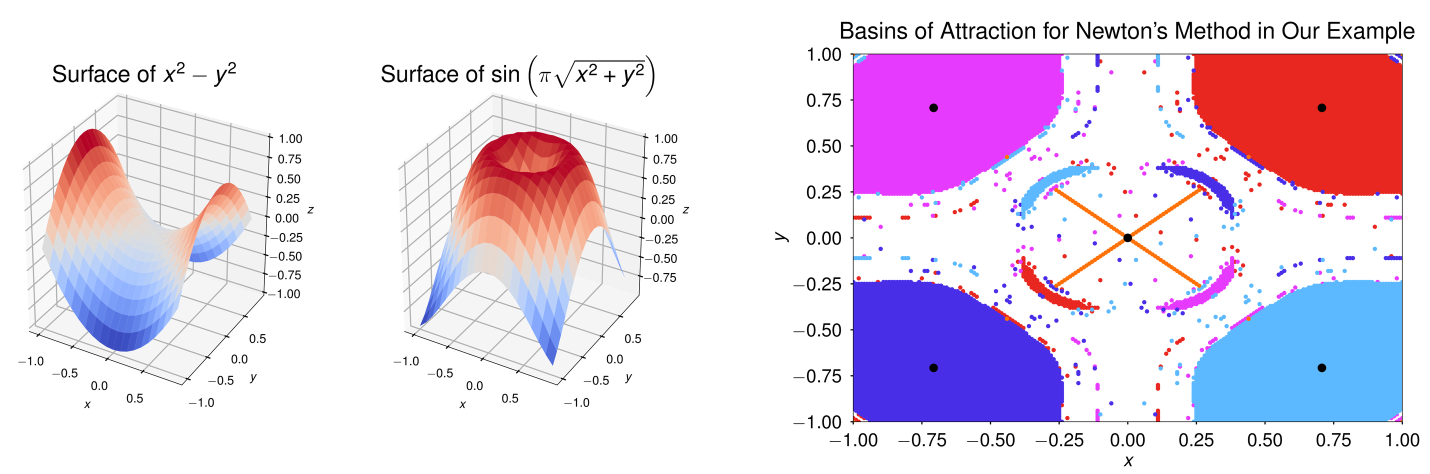


Figure: Newton's method would require multiple runs to find all roots, which would be inadequate for our application.

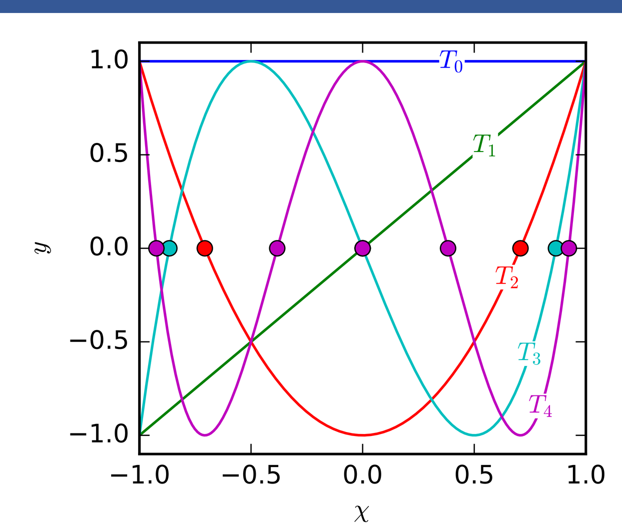
Chebyshev Polynomials

Definition

$$T_n(x) = \cos(n \arccos(x)),\ x \in [-1,1]$$

Interpolation Convergence [8, Ch. 7, 8]

- $f^{(m)}$ has bounded variation $\Longrightarrow$ polynomial convergence of order m
- f is analytic $\Longrightarrow$ geometric convergence



Summary of Algorithm from [5]

- Construct interpolations, p_f, p_g, of f, g [6, Sec. 2]
- Apply the Bézout Resultant Method for x values
- Apply 1-D root finding techniques for y values

Bézout Resultant Method

With p_f, p_g of degree $(m_f, n_f), (m_g, n_g)$, we can construct a square **Bézout Matrix Polynomial** of degree $M \le m_f + m_g$ and size $n = \max(n_f, n_g)$:

$$B(x) = \sum_{i=0}^{M} B_i T_i(x),$$

- $\det(B(x_0)) = 0 \iff p_f(x_0,\cdot)$ and $p_g(x_0,\cdot)$ have a common root
- Solving $\det(B(x_0)) = 0$ involves linearizing $B(x)$

Challenges with the Algorithm

Poor Conditioning

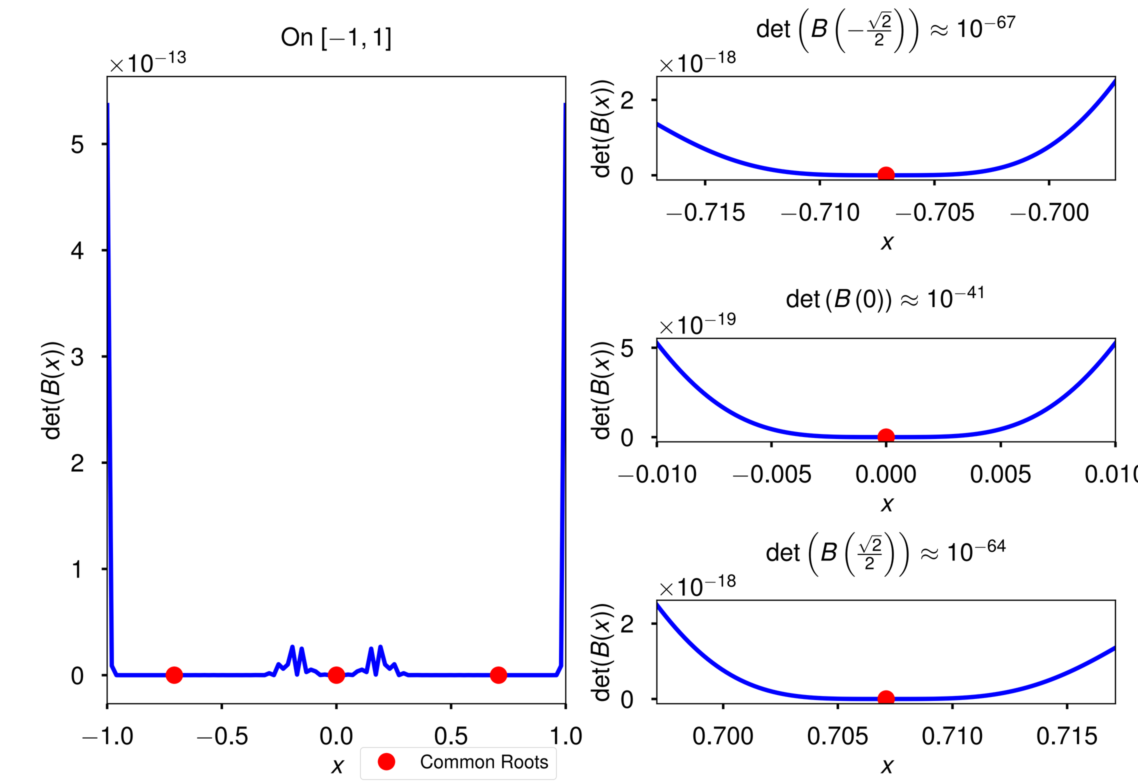


Figure: $\det(B(x))$ from our example. The Bézout technique can square the condition number. [5, Sec. 5]

Poor Scaling

- Have to solve a generalized eigenvalue problem [5, Sec. 3]
- Computational complexity of $\mathcal{O}(M^3 n^3)$ [5, Sec. 4]

Our Implementation

Refinement

- `chebfun2` refines roots by repeating the Bézout method in a smaller region surrounding the root [5, Sec. 7]
- We refine the roots with Newton's method

1-D Root Finding

- Our 1-D algorithm is a mix of a bisection and secant method
- Faster than companion matrix methods [2]

Domain Subdivision

- `chebfun2` divides the domain to reduce the matrix polynomial problem size [5, Sec. 4]
- We have yet to implement a domain subdivision method

Current Results

	ChebTools		chebfun	
Functions	Max $\|\cdot\|_2$ Error	Time (s)	Max $\|\cdot\|_2$ Error	Time (s)
$F_1(x,y), F_2(x,y)$	6.97×10^{-32}	0.004268	7.7×10^{-16}	0.522
$G_1(x,y), G_2(x,y)$	6.21×10^{-16}	214.059	2.92×10^{-10}	0.294
$H_1(x,y), H_2(x,y)$	3.24×10^{-16}	195.903	2.77×10^{-11}	0.296

$$F_1(x,y) = T_3(x) - 13T_1(x) \qquad F_2(x,y) = T_3(y) - 13T_1(y)$$
$$G_1(x,y) = \cos(\pi x)(y-2) \qquad G_2(x,y) = (y - .9)(x-2)$$
$$H_1(x,y) = \cos\left(\pi x - \frac{\pi}{10}\right)(y-2) \qquad H_2(x,y) = (y-.1)(y-.9)(x-2)$$

Discussion

- `chebfun` is much faster than `ChebTools`
- This speed difference can be explained by the lack of a subdivision strategy in our implementation
- A new subdivision strategy is part of current and future work

Current Work

Our Application Motivates Current Work

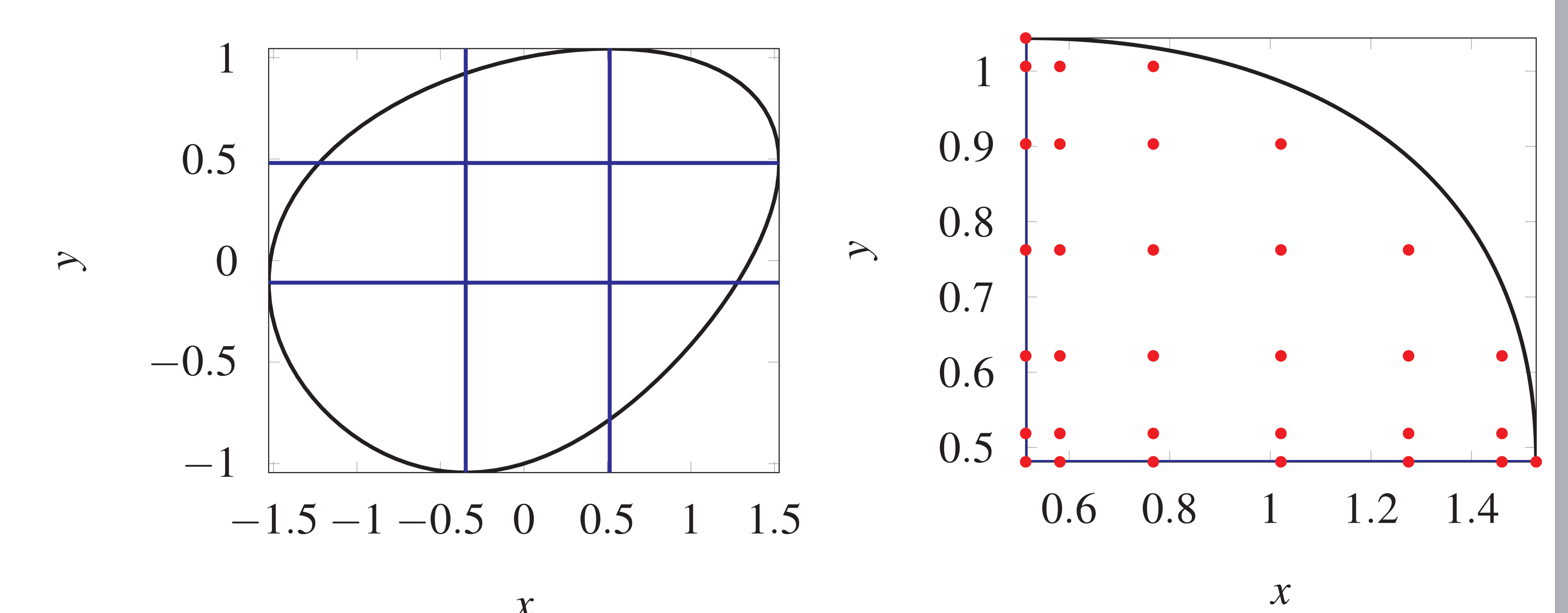
- Empirical equations of state in thermodynamics have ranges of validity which are not rectangular domains
- Have to use coordinate transformations [7] or to work on non-rectangular domains in `chebfun2`

Our Subdivision Idea

- Subdivide where domain boundary is horizontal, vertical, or has a derivative discontinuity
- Interior subdomains can proceed with rectangular subdivision like in the `chebfun` algorithm
- Approximate on exterior subdomains using interior Chebyshev points

Example Domain

Interior of the closed curve $(1.5\cos(t) + .15\sin(2t), \sin(t) + .3\cos(t))$ with $t \in [0, 2\pi]$



(a) Boundary in black and initial subdivisions in blue.

(b) Interior 7x7 Chebyshev nodes in red.

Figure: Example domain on the left and the upper right subdomain on the right

Future Work

- Parallelize the `ChebTools` library
- Adaptive capabilities for approximations

Add Features To Construct a Chebyshev expansion from:

- linear least squares
- linear boundary value problems

References

[1] I.H. Bell, L. Bouck, and B.K. Alpert. Chebtools: A c++11 based library for working with chebyshev expansions. *Journal of Open Source Software*, 2018. To be submitted.

[2] J.P. Boyd. Finding the zeros of a univariate equation: Proxy rootfinders, chebyshev interpolation, and the companion matrix. *SIAM Review*, 55(2):375–396, 2013.

[3] T. A Driscoll, N. Hale, and L. N. Trefethen. *Chebfun Guide*. Pafnuty Publications, 2014.

[4] M. Kunick, H.J. Kretzschmar, and U. Gampe. Fast calculation of thermodynamic properties of water and steam in process modelling using spline interpolation. 01 2008.

[5] Y. Nakatsukasa, V. Noferini, and A. Townsend. Computing the common zeros of two bivariate functions via bézout resultants. *Numerische Mathematik*, pages 1–29, 2013.

[6] A. Townsend and L. N. Trefethen. An extension of chebfun to two dimensions. *SIAM Journal on Scientific Computing*, 35(6):C495–C518, 2013.

[7] Alex Townsend. Chebfun2 objects on non-rectangular domains. http://www.chebfun.org/examples/approx2/Other2DDomains.html, 2013.

[8] L.N. Trefethen. *Approximation Theory and Approximation Practice (Other Titles in Applied Mathematics)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2012.

This work was supported by the NIST SURF program.

Department of Mathematical Sciences, George Mason University, Fairfax, VA 22030

lbouck@gmu.edu