Root Finding with Chebyshev Polynomials in 2 Dimensions

Lucas C. Bouck

George Mason University
Department of Mathematical Sciences

September 8, 2017

Applied and Computational Math Seminar
George Mason University
Fairfax, VA

Collaborator: Ian H. Bell (NIST Applied Chemicals and Materials Division)

Outline

The Problem

Global 2-D Root Finding Problem

We want to find all solutions $\mathbf{x} \in \Omega \subset \mathbb{R}^2$ to

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}$$

Additional assumptions

- Ω will be a bounded domain and rectangular for now
- Treat $\mathbf{F}$ as $\begin{pmatrix} f(x,y) \\ g(x,y) \end{pmatrix}$, where $x, y \in \mathbb{R}$ and f, g are scalar functions
- Assume f, g are smooth enough (more on this later)
- Assume finitely many roots

Background and Motivation

The Big Picture: `chebtools`

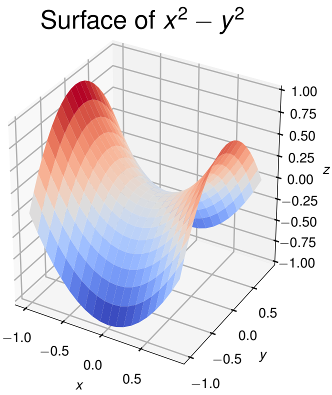
- `chebtools` is a C++ library for working with Chebyshev expansions developed by Ian Bell
- Inspired by the Matlab library `chebfun`
- 2-D root finding will become a central feature of `chebtools`

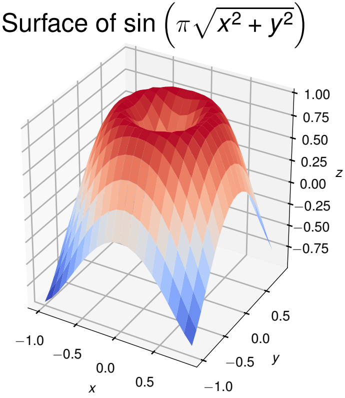
Applications:

- One example is determining thermodynamic properties of steam or water (Kunick, Kretzschmar, and Gampe 2008)
- With a higher level Python interface, `chebtools` could be useful for a wide range of users

Example

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





A Need for a Method to Find All Roots

- Iterative methods may converge to roots outside the domain of interest.
- Globalized Newton methods guarantee convergence to a root but only find them one at a time (Deuflhard, 2011)
- We need a method to find all solutions at once

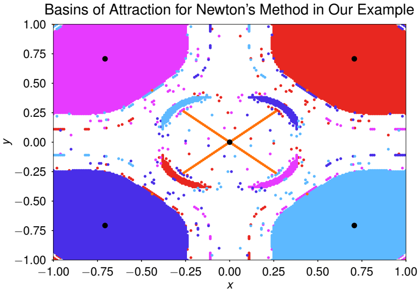


Figure: Initial guesses in white areas did not converge to roots in Ω

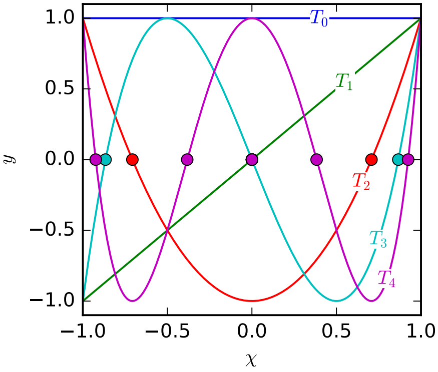
What are Chebyshev Polynomials?

Chebyshev polynomials

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

- Orthogonal under an inner product with weight $w(x) = \frac{1}{\sqrt{1-x^2}}$
- Lipschitz continuity $\implies$ uniform convergence of Chebyshev interpolations
- Analytic function $\implies$ geometric convergence

Figure: First 5 Chebyshev Polynomials



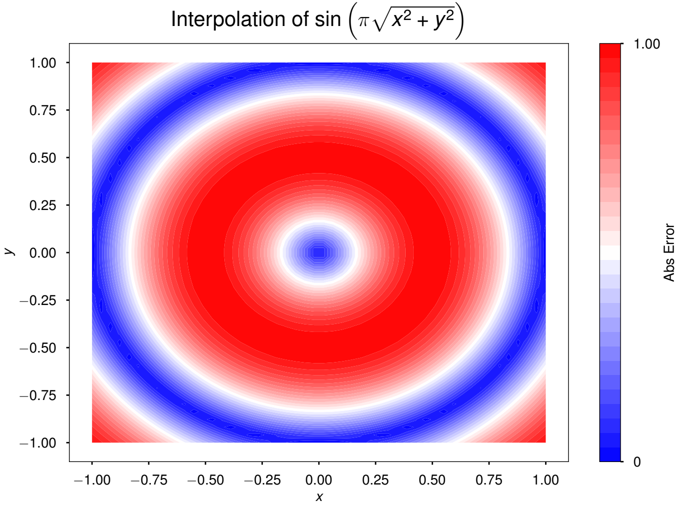
2-D Interpolation

- Define our first error function as $e_0(x, y) = f(x, y)$
- Define later error functions as $e_k(x, y) \coloneqq e_{k-1}(x, y) - P_{k-1}(x, y)$ where P_{k-1} is our approximation

Algorithm Outline

- Find (x_k, y_k) s.t. $|e_k(x_k, y_k)| = \max |e_k(x, y)|$
- Do 1-D interpolations of $e_k(x, y_k)$ and $e_k(x_k, y)/e_k(x_k, y_k)$ denoted $p_x(x)$ and $p_y(y)$
- Compute new approximation $P_k(x, y) = P_{k-1}(x, y) + p_x(x)p_y(y)$

Interpolation From Our Example



Bézout Resultant Method

- From our Chebyshev approximations $p_f(x, y), p_g(x, y)$, we can construct a Bézout matrix polynomial $B(x)$
- $\det\left(B(x_0)\right) = 0 \iff p_f(x_0, \cdot)$ and $p_g(x_0, \cdot)$ have common root

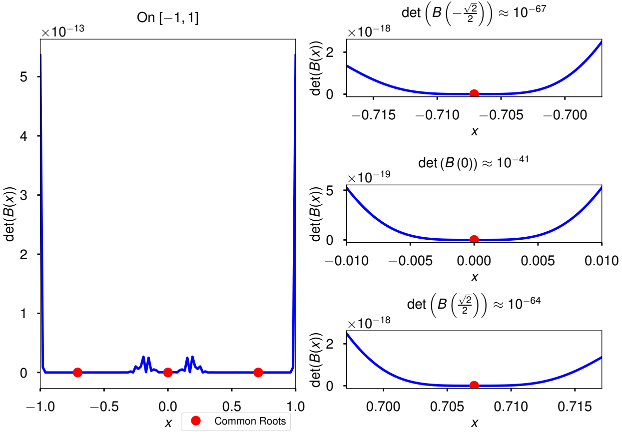
If p_f, p_g are of degree $(m_f, n_f), (m_g, n_g)$ respectively, then the resulting form of the matrix polynomial of degree M is

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

- B_i are square matrices of size $n = \max(n_f, n_g)$ and $M \leq m_f + m_g$.
- Solving $\det\left(B(x_0)\right) = 0$ involves linearizing $B(x)$ (Nakatsukasa et. al. 2016)
- Finally solve a generalized eigenvalue problem with computational complexity of $\mathcal{O}\left(M^3 n^3\right)$

Conditioning of the Bézout Matrix Polynomials

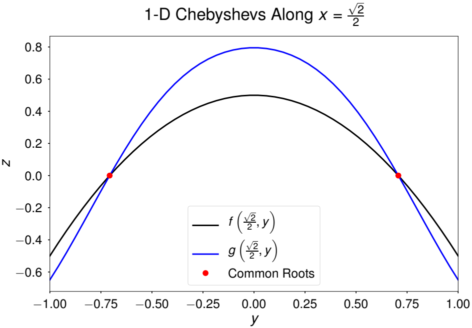
- The Bézout matrix polynomial technique can square the condition number of the common root.
- The `chebfun` algorithm refines the roots by recomputing the matrix polynomial problem on a zoomed in region of the root



1-D Root Finding

1-D Root Finding

- We now have the possible x values for where there are common roots
- Employ a companion matrix method finding the roots of 1-D Chebyshev polynomials



First Results

	chebtools		chebfun	
Functions	2 Norm Error	Time (s)	2 Norm 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

$$F_1(x,y) = T_3(x) - 13T_1(x) \qquad F_2(x,y) = T_3(y) - 13T_1(y)$$

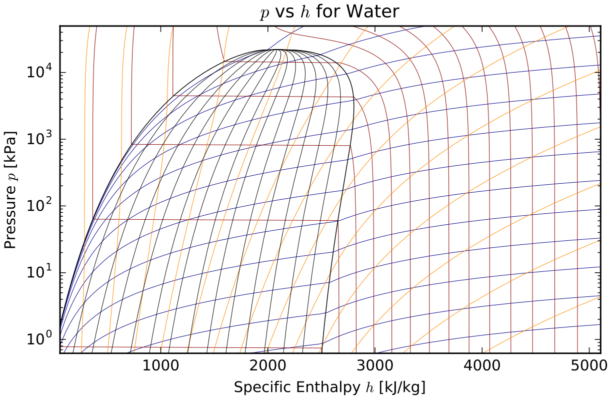
$$G_1(x,y) = \cos(\pi x)(y-2) \qquad G_2(x,y) = (y-.9)(x-2)$$

$$H_1(x,y) = \cos\left(\pi x - \frac{\pi}{10}\right)(y-2) \qquad H_2(x,y) = (y-.1)(y-.9)(x-2)$$

- `chebfun` is much faster than `chebtools`
- `chebfun` has a domain subdivision strategy for reducing the size of the generalized eigenvalue problem

Application Motivates Future Work

- Empirical equations of state in thermodynamics have ranges of validity which are not rectangular domains
- Solutions outside the domain will not make physical sense
- Some properties are not defined at certain points (ex: Critical Point)



Current Work for Moving Beyond Rectangular Domains

Suppose we have

- Subdivided such that part of the boundary of the subdomain of interest, Ω_s, can be expressed as a function
- mapped the rectangle containing Ω_s has been mapped to a reference square $[-1, 1]^2$

Our Idea:

- Same 2-D interpolation procedure as with the rectangular domain
- Instead of 1-D interpolations, solve a least squares fitting problem with nodes inside Ω_s
- Constrain the least squares solution s.t. $p(\pm 1) \leq B$ for some bound B

Future Work

For Non-Rectangular Domains

- Further develop ideas for Chebyshev approximation methods
- Provide analysis of the new method of approximations

Other Future Work

- Introduce GPU/parallel computing to the root finding process

Conclusions and Acknowledgements

Contributions:

- Made significant progress in replicating the 2-D root finding algorithm in `chebfun` with some modifications in C++11
- Began to develop ideas for extending the algorithm to non-rectangular domains

Acknowledgments

- Ian Bell and Bradley Alpert (NIST Applied Math Division)
- The Summer Undergraduate Research Fellowship program at NIST

Questions?

References

[1] R. Bartels and G. Stewart. *Solution of the Matrix Equation $AX + XB = C$: Algorithm 432*, Comm. ACM, 1972.
[2] I. Bell et. al. *Pure and Pseudo-pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp*, Industrial & Engineering Chemical Research, 2014.
[3] P. Deuflhard. *Newton Methods for Nonlinear Problems*, Springer, Berlin, 2011.
[4] G. Golub and C. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 2013.
[5] M. Kunick et. al. *Fast Calculation of Thermodynamic Properties of Water and Steam in Process Modelling using Spline Interpolation*. 2008.
[6] Y. Nakatsukasa et. al. *Vector Spaces of Linearizations for Matrix Polynomials: A Bivariate Polynomial Approach*, arXiv:1610.01859v1, 2016.
[7] A. Townsend. *Computing with Functions in Two Dimensions* (Doctoral Dissertation). Oxford University, 2014.
[8] L. N. Trefethen, *Approximation Theory and Approximation Practice*, SIAM, Philadelphia, 2013.