

Rootfinding with Chebyshev Polynomials in 2 Dimensions

Lucas C. Bouck

George Mason University
Department of Mathematical Sciences

August 2, 2017

SURF Colloquium
National Institute of Standards and Technology
Boulder, CO

Collaborator: Ian H. Bell (NIST Division 647)

Outline

- 1 The Problem
- 2 What are Chebyshev Polynomials and why use them?
- 3 The Algorithm
- 4 Current Ideas for Moving Beyond Rectangular Domains

1 The Problem

2 What are Chebyshev Polynomials and why use them?

3 The Algorithm

4 Current Ideas for Moving Beyond Rectangular Domains

The Problem

We want to find all solutions to

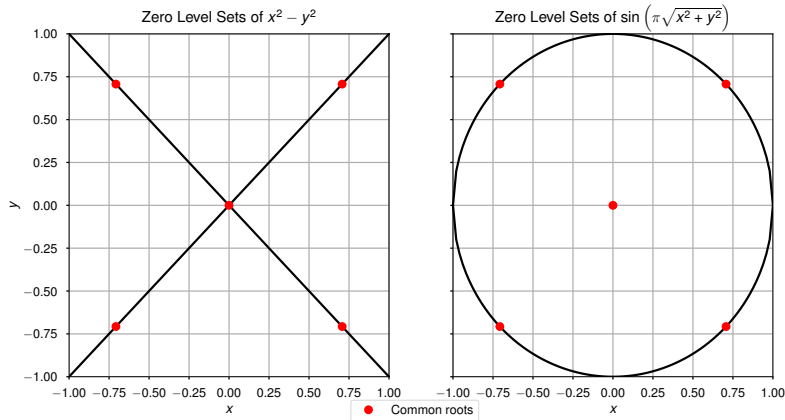
$$\begin{pmatrix} f(x, y) \\ g(x, y) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

in a given bounded domain Ω

- This is a global root finding problem: we want all solutions inside Ω
- Root finding is has applications in many fields
- An application for NIST would be in making flash calculations to determine properties of a fluid from its equation of state

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 $\Omega = [-1, 1]^2$



Local Methods May Not Be Good for Global Root Finding

There are actually infinitely many roots outside Ω in our example. Local methods may converge to roots outside the domain of interest.

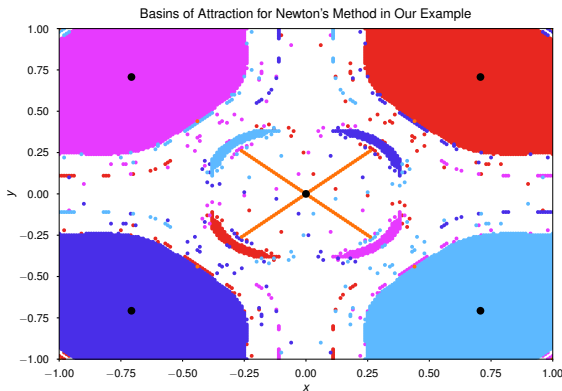


Figure: Note that initial guesses in white areas converged to roots outside our domain and the basin of attraction for $(0,0)$ is relatively small

A Need for a Global Method

Our example illustrates a few lessons

- Many iterative and local methods like Newton's method can only guarantee convergence to a root if the initial guess is sufficiently close
- Sufficiently close depends on the problem
- This may be extremely close for some roots like $(0,0)$ in our example

Goal:

To develop a truly global root finding method

1 The Problem

2 What are Chebyshev Polynomials and why use them?

3 The Algorithm

4 Current Ideas for Moving Beyond Rectangular Domains

What are Chebyshev polynomials?

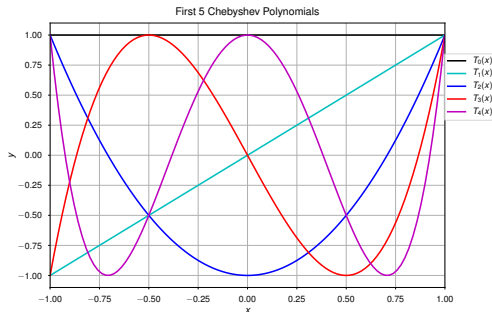
Chebyshev polynomials

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

They are orthogonal under a weighted inner product:

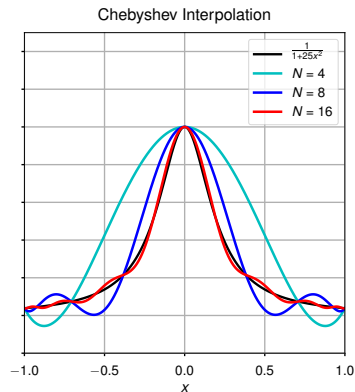
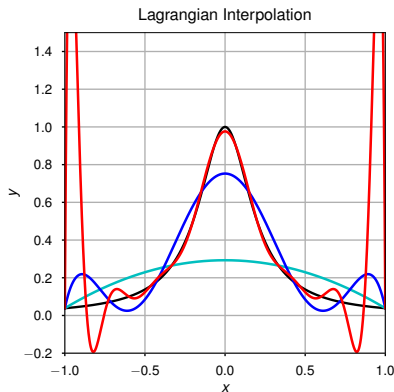
$$\int_{-1}^1 T_n(x) T_k(x) w(x) dx = 0$$

when $n \neq k$
and $w(x) = \frac{1}{\sqrt{1-x^2}}$



Chebyshev polynomials are accurate and numerically stable

- Typical interpolation is susceptible to the **Runge phenomenon**
- Chebyshev interpolation minimizes this effect.



1 The Problem

2 What are Chebyshev Polynomials and why use them?

3 The Algorithm

4 Current Ideas for Moving Beyond Rectangular Domains

1st: 2-D Interpolation

The algorithm works by choosing a pivot position which is the maximum residual and doing 1-D interpolations in the x and y directions

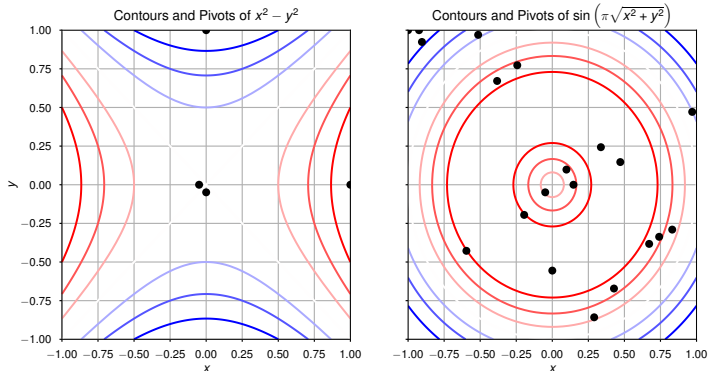
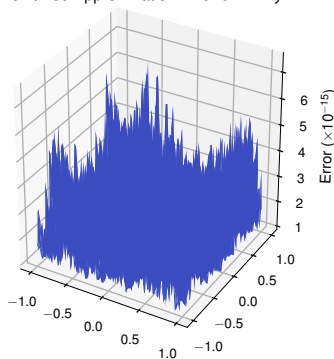


Figure: Plot of the contours of our example functions with pivots from the algorithm

Our Algorithm Can Provide Accurate Approximations

- Almost machine precision accuracy with $x^2 - y^2$
- Struggles with $\sin\left(\pi\sqrt{x^2 + y^2}\right)$ due to the lack of differentiability at $(0, 0)$

Pointwise Approximation Error of $x^2 - y^2$



Pointwise Approximation Error of $\sin\left(\pi\sqrt{x^2 + y^2}\right)$

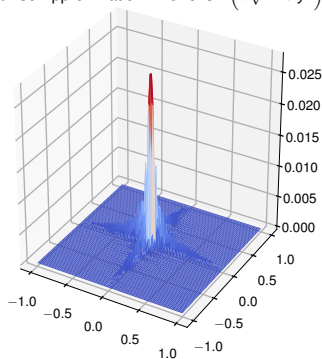


Figure: Plot of the pointwise errors of our algorithm

2nd: Bézout Matrix Polynomials

- We can create Bézout matrices at each Chebyshev node along the x or y axis
- Interpolating the Bézout matrices gives us a matrix polynomial $B(x)$

The matrix polynomial can be expressed as

$$B(x) = \sum_{k=0}^M B_k T_k(x)$$

where B_k are $N \times N$ matrices.

- $\det(B(x)) = 0 \iff$ common roots along the specific x value
- Solving $\det(B(x)) = 0$ is a **matrix polynomial eigenvalue problem**
- Can be solved by linearizing and solving generalized eigenvalue problem of size $MN \times MN$
- The most computational intensive part of the algorithm

3rd: 1-D Root Finding

- We now have the possible x (or y) values for where there are common roots
- Employ a 1-D root finding method again using Chebyshev Polynomials

We do the following:

- Find the 1-D Chebyshev polynomials $p_f(y)$, $p_g(y)$ at the specific x values
- Utilize 1-D Chebyshev rootfinding techniques
- Polish the roots with Newton's method

- 1 The Problem
- 2 What are Chebyshev Polynomials and why use them?
- 3 The Algorithm
- 4 Current Ideas for Moving Beyond Rectangular Domains

Conclusions and Future Work

Key Contributions:

- Replicated Alex Townsend's algorithm in chebfun2 with some modifications in C++
- Developed ideas and have a strategy for extending the algorithm to non-rectangular domains

Future Work:

- Introduce GPU/parallel computing to the rootfinding process
- Further develop ideas on non-rectangular domains and subdivision strategies

Other ideas while at NIST:

- Idea for using Gram-Schmidt process to create orthogonal terms for developing equations of state

Acknowledgements

- Dr. Ian Bell for the mentorship and guidance
- Dr. Bradley Alpert for the lively discussions and brainstorming
- The SURF/SHIP program and it's directors

References

References here