Computing Complex Singularities of Differential Equations with Chebfun

Marcus Webb

Cambridge Centre for Analysis,
University of Cambridge

Based on work supervised by Nick Trefethen in 2011, funded by EPSRC

25th Biennial Conference on Numerical Analysis
University of Strathclyde, Glasgow

27th June 2013

Motivation

- Suppose you have solved an ODE on a time interval $[0, T]$ (numerically or analytically),

Motivation

- Suppose you have solved an ODE on a time interval $[0, T]$ (numerically or analytically),
- and now you want to know if it has singularities in the complex plane.

Motivation

- Suppose you have solved an ODE on a time interval $[0, T]$ (numerically or analytically),
- and now you want to know if it has singularities in the complex plane.
- Why?

Motivation

- Suppose you have solved an ODE on a time interval $[0, T]$ (numerically or analytically),
- and now you want to know if it has singularities in the complex plane.
- Why?
- The singularities may have physical significance. E.g. complex singularities of Painlevé equations determine the oscillations and asymptotics along the real line.

Motivation

- Suppose you have solved an ODE on a time interval $[0, T]$ (numerically or analytically),
- and now you want to know if it has singularities in the complex plane.
- Why?
- The singularities may have physical significance. E.g. complex singularities of Painlevé equations determine the oscillations and asymptotics along the real line.
- It can inform the mathematical analysis of the ODE. E.g. if all singularities lie outside the strip $|\mathrm{Im}(t)| \leq \tau$, then the transformation

$$\zeta = \frac{\exp(\pi t/2\tau) - 1}{\exp(\pi t/2\tau) + 1}$$

maps the strip to the unit disc. The solution must have a convergent expansion in powers of ζ.

Motivation

- We are thinking about *Numerical Analytic Continuation*.

Motivation

- We are thinking about *Numerical Analytic Continuation*.
- First idea: If we solve the ODE in Chebfun, we get a chebfun u. What does this polynomial look like in the complex plane?

- We are thinking about *Numerical Analytic Continuation*.
- First idea: If we solve the ODE in Chebfun, we get a chebfun u. What does this polynomial look like in the complex plane?

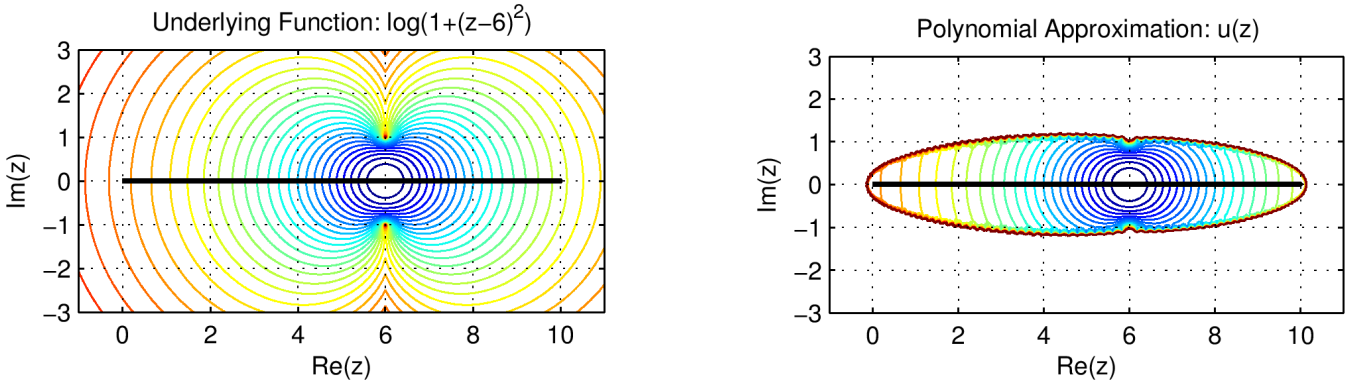


Figure : A polynomial interpolant (in Chebyshev points scaled and shifted to $[0, 10]$ here) cannot possibly approximate complex singularities because it is an entire function.

- We are thinking about *Numerical Analytic Continuation*.
- First idea: If we solve the ODE in Chebfun, we get a chebfun u. What does this polynomial look like in the complex plane?

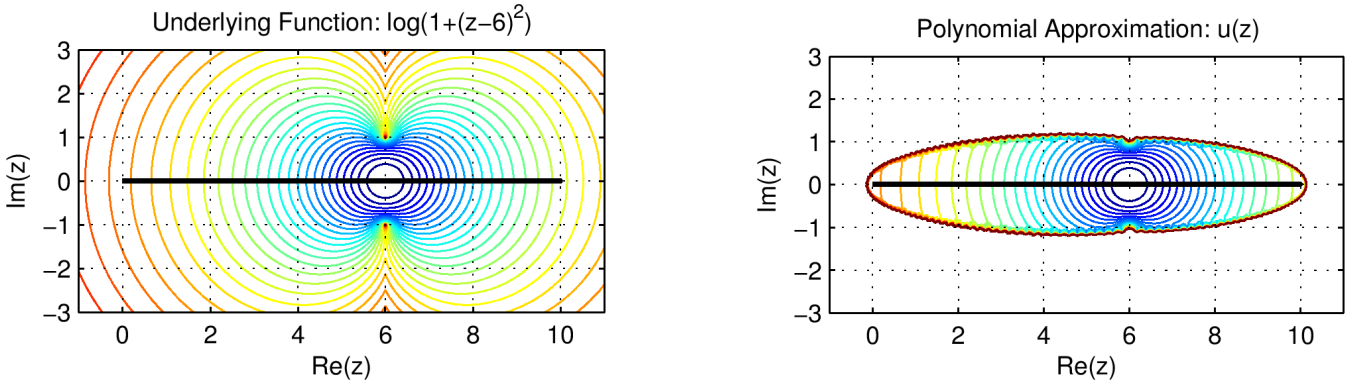


Figure : A polynomial interpolant (in Chebyshev points scaled and shifted to $[0, 10]$ here) cannot possibly approximate complex singularities because it is an entire function.

- A better idea is to use rational functions, because they can have singularities in the complex plane.

Motivation: Issues with rational approximation

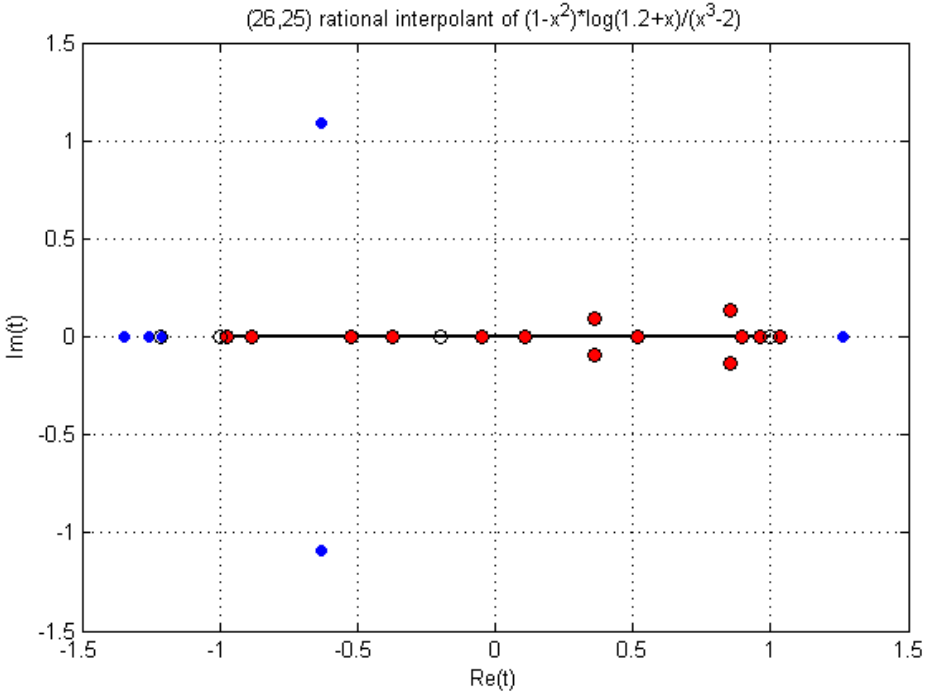
- Rational approximation is not as popular or as well known as polynomial approximation.

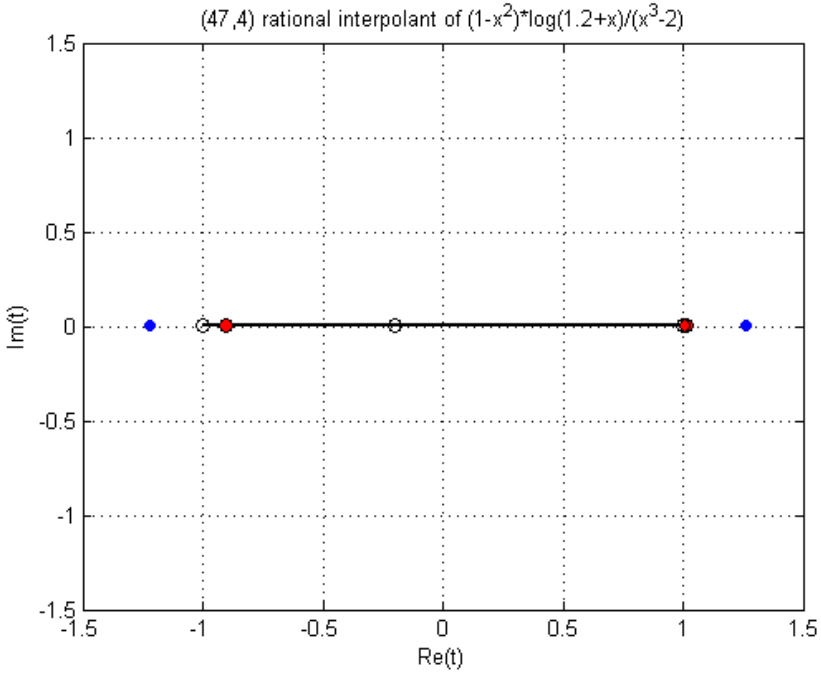
Motivation: Issues with rational approximation

- Rational approximation is not as popular or as well known as polynomial approximation.
- One reason is the phenomenon of *spurious poles*.

Motivation: Issues with rational approximation

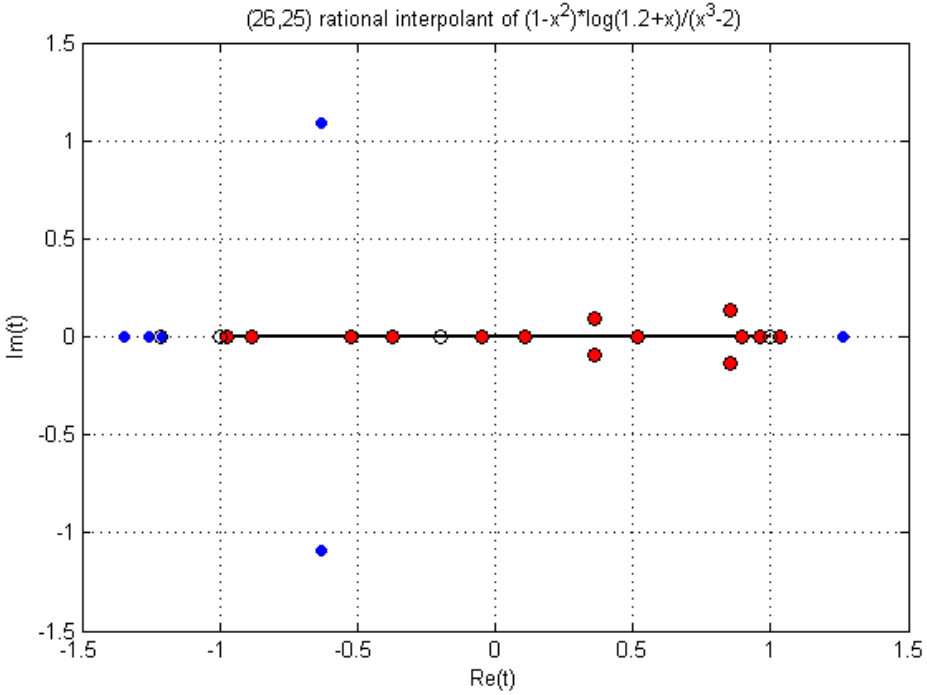
- Rational approximation is not as popular or as well known as polynomial approximation.
- One reason is the phenomenon of *spurious poles*.

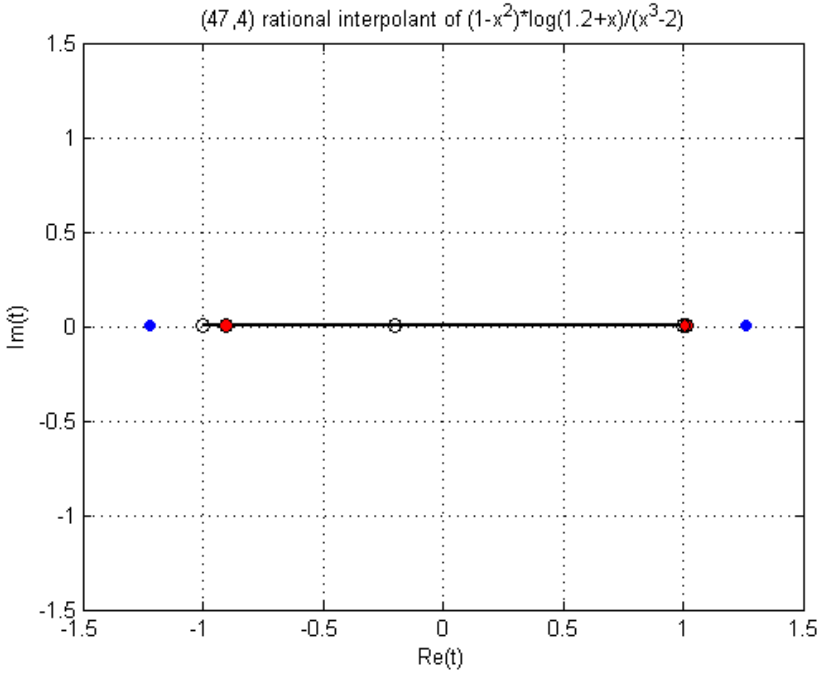




Motivation: Issues with rational approximation

- Rational approximation is not as popular or as well known as polynomial approximation.
- One reason is the phenomenon of *spurious poles*.





- We want a *robust* rational approximation.

Rational interpolation and least squares

- Let $\mathbf{x} = (x_0, x_1, \ldots, x_N) \in \mathbb{C}^{N+1}$, and let $f : G \subset \mathbb{C} \to \overline{\mathbb{C}}$

Rational interpolation and least squares

- Let $\mathbf{x} = (x_0, x_1, \ldots, x_N) \in \mathbb{C}^{N+1}$, and let $f : G \subset \mathbb{C} \to \overline{\mathbb{C}}$
- For $m + n = N$, the *Cauchy interpolation problem* is to find $r \in \mathcal{R}_{m,n}$ such that

$$r(\mathbf{x}) = f(\mathbf{x})$$

Rational interpolation and least squares

- Let $\mathbf{x} = (x_0, x_1, \ldots, x_N) \in \mathbb{C}^{N+1}$, and let $f : G \subset \mathbb{C} \to \overline{\mathbb{C}}$
- For $m + n = N$, the *Cauchy interpolation problem* is to find $r \in \mathcal{R}_{m,n}$ such that

$$r(\mathbf{x}) = f(\mathbf{x})$$

- **May not exist**: $r \in \mathcal{R}(1,1)$ such that $r(\pm 1) = 0$, $r(0) = 1$.

Rational interpolation and least squares

- Let $\mathbf{x} = (x_0, x_1, \ldots, x_N) \in \mathbb{C}^{N+1}$, and let $f : G \subset \mathbb{C} \to \overline{\mathbb{C}}$
- For $m + n = N$, the *Cauchy interpolation problem* is to find $r \in \mathcal{R}_{m,n}$ such that

$$r(\mathbf{x}) = f(\mathbf{x})$$

- **May not exist**: $r \in \mathcal{R}(1,1)$ such that $r(\pm 1) = 0$, $r(0) = 1$.
- To deal with this, consider the more general approach: Define

$$\langle f, g \rangle_N = \sum_{i=0}^{N} \lambda_i f(x_i) \overline{g(x_i)},$$

where $\lambda_i > 0$, and find $p \in \mathcal{P}_m$, $q \in \mathcal{P}_n$ (and take $r = p/q$) to

$$\text{minimise } \|p - fq\|_N \text{ such that } \|q\|_N = 1.$$

Rational interpolation and least squares

- Reminder: minimise $\|p - fq\|_N$ such that $\|q\|_N = 1$

Rational interpolation and least squares

- Reminder: minimise $\|p - fq\|_N$ such that $\|q\|_N = 1$
- If $m + n = N$, then there **always** exists a solution with $\|p - fq\|_N = 0$, called a linearised solution.

Rational interpolation and least squares

- Reminder: minimise $\|p - fq\|_N$ such that $\|q\|_N = 1$
- If $m + n = N$, then there **always** exists a solution with $\|p - fq\|_N = 0$, called a linearised solution.
- If $m + n < N$, then this gives a linearised least squares solution. These are **not** interpolants if $\|p - fq\|_N > 0$.

Rational interpolation and least squares

- Reminder: minimise $\|p - fq\|_N$ such that $\|q\|_N = 1$
- If $m + n = N$, then there **always** exists a solution with $\|p - fq\|_N = 0$, called a linearised solution.
- If $m + n < N$, then this gives a linearised least squares solution. These are **not** interpolants if $\|p - fq\|_N > 0$.
- Idea: **convert the problem** into a linear algebra problem for **coefficients a** of p and **b** of q in a certain polynomial expansion.

Rational interpolation and least squares

- Reminder: minimise $\|p - fq\|_N$ such that $\|q\|_N = 1$
- If $m + n = N$, then there **always** exists a solution with $\|p - fq\|_N = 0$, called a linearised solution.
- If $m + n < N$, then this gives a linearised least squares solution. These are **not** interpolants if $\|p - fq\|_N > 0$.
- Idea: **convert the problem** into a linear algebra problem for **coefficients a** of p and **b** of q in a certain polynomial expansion.
- To this aim, we find orthogonal polynomials $(P_j)_{j=0}^N$ with respect to the discrete inner product $\langle \cdot, \cdot \rangle_N$.

Rational interpolation and least squares

- Reminder: minimise $\|p - fq\|_N$ such that $\|q\|_N = 1$
- If $m + n = N$, then there **always** exists a solution with $\|p - fq\|_N = 0$, called a linearised solution.
- If $m + n < N$, then this gives a linearised least squares solution. These are **not** interpolants if $\|p - fq\|_N > 0$.
- Idea: **convert the problem** into a linear algebra problem for **coefficients a** of p and **b** of q in a certain polynomial expansion.
- To this aim, we find orthogonal polynomials $(P_j)_{j=0}^N$ with respect to the discrete inner product $\langle \cdot, \cdot \rangle_N$.
- Simplest example: if **x** are roots of unity, take $\lambda_i = 1$ and $P_j(x) = x^j$. Merely orthogonality of the discrete Fourier basis.

Rational interpolation and least squares

- If $\mathbf{x}$ are Chebyshev points $x_i = \cos(i\pi/N)$, take $\lambda_0 = \lambda_N = \frac{1}{2N}$, $\lambda_i = \frac{2}{N}$, so that

$$\langle f, g \rangle_N = \frac{2}{N} \sum_{i=0}^{N}{}'' f(x_i)\overline{g(x_i)}$$

Rational interpolation and least squares

- If $\mathbf{x}$ are Chebyshev points $x_i = \cos(i\pi/N)$, take $\lambda_0 = \lambda_N = \frac{1}{2N}$, $\lambda_i = \frac{2}{N}$, so that

$$\langle f, g \rangle_N = \frac{2}{N} \sum_{i=0}^{N}{}'' f(x_i)\overline{g(x_i)}$$

- The $''$ indicates halving the first and last enties. Then we have

$$\langle T_j, T_k \rangle_N = \begin{cases} 2 & \text{if } j = k = 0, N, \\ 1 & \text{if } j = k \neq 0, N, \\ 0 & \text{if } j \neq k. \end{cases}$$

for the Chebyshev polynomials $T_j(x) = \cos(j \cos^{-1}(x))$.

Rational interpolation and least squares

- If $\mathbf{x}$ are Chebyshev points $x_i = \cos(i\pi/N)$, take $\lambda_0 = \lambda_N = \frac{1}{2N}$, $\lambda_i = \frac{2}{N}$, so that

$$\langle f, g \rangle_N = \frac{2}{N} \sum_{i=0}^{N}{}'' f(x_i)\overline{g(x_i)}$$

- The $''$ indicates halving the first and last enties. Then we have

$$\langle T_j, T_k \rangle_N = \begin{cases} 2 & \text{if } j = k = 0, N, \\ 1 & \text{if } j = k \neq 0, N, \\ 0 & \text{if } j \neq k. \end{cases}$$

  for the Chebyshev polynomials $T_j(x) = \cos(j \cos^{-1}(x))$.

- Assume we have **normalised** T_0 and T_N.

Rational interpolation and least squares

- Let $p \in \mathcal{P}_m$ and $q \in \mathcal{P}_n$ be a candidate solution, and let $\hat{p} \in \mathcal{P}_N$ interpolate $f \cdot q$ on $\mathbf{x}$. We write them as

$$p = \sum_{j=0}^{N} a_j T_j, \quad q = \sum_{j=0}^{N} b_j T_j, \quad \hat{p} = \sum_{j=0}^{N} \hat{a}_j T_j.$$

Rational interpolation and least squares

- Let $p \in \mathcal{P}_m$ and $q \in \mathcal{P}_n$ be a candidate solution, and let $\hat{p} \in \mathcal{P}_N$ interpolate $f \cdot q$ on $\mathbf{x}$. We write them as

$$p = \sum_{j=0}^{N} a_j T_j, \quad q = \sum_{j=0}^{N} b_j T_j, \quad \hat{p} = \sum_{j=0}^{N} \hat{a}_j T_j.$$

- Let $C = (T_j(x_i))_{i,j=0}^{N}$ and $I'' = \mathrm{diag}(\frac{1}{2}, 1, \ldots, 1, \frac{1}{2})$. Then we have

$$p(\mathbf{x}) = C\mathbf{a}, \quad \|p\|_N = \|\mathbf{a}\|_2 \text{ etc.}, \quad \frac{2}{N} C^\top I'' C = I.$$

Rational interpolation and least squares

- Let $p \in \mathcal{P}_m$ and $q \in \mathcal{P}_n$ be a candidate solution, and let $\hat{p} \in \mathcal{P}_N$ interpolate $f \cdot q$ on $\mathbf{x}$. We write them as

$$p = \sum_{j=0}^{N} a_j T_j, \quad q = \sum_{j=0}^{N} b_j T_j, \quad \hat{p} = \sum_{j=0}^{N} \hat{a}_j T_j.$$

- Let $C = (T_j(x_i))_{i,j=0}^{N}$ and $I'' = \mathrm{diag}(\frac{1}{2}, 1, \ldots, 1, \frac{1}{2})$. Then we have

$$p(\mathbf{x}) = C\mathbf{a}, \quad \|p\|_N = \|\mathbf{a}\|_2 \text{ etc.}, \quad \frac{2}{N} C^\top I'' C = I.$$

- Interpolation property of $\hat{p}$ implies

$$\hat{\mathbf{a}} = \frac{2}{N} C^\top I'' F C \mathbf{b}, \text{ where } F = \mathrm{diag}(f(x_0), \ldots, f(x_N))$$

Rational interpolation and least squares

- Let $p \in \mathcal{P}_m$ and $q \in \mathcal{P}_n$ be a candidate solution, and let $\hat{p} \in \mathcal{P}_N$ interpolate $f \cdot q$ on $\mathbf{x}$. We write them as

$$p = \sum_{j=0}^{N} a_j T_j, \quad q = \sum_{j=0}^{N} b_j T_j, \quad \hat{p} = \sum_{j=0}^{N} \hat{a}_j T_j.$$

- Let $C = (T_j(x_i))_{i,j=0}^{N}$ and $I'' = \mathrm{diag}(\frac{1}{2}, 1, \ldots, 1, \frac{1}{2})$. Then we have

$$p(\mathbf{x}) = C\mathbf{a}, \quad \|p\|_N = \|\mathbf{a}\|_2 \text{ etc.}, \quad \frac{2}{N} C^\top I'' C = I.$$

- Interpolation property of $\hat{p}$ implies

$$\hat{\mathbf{a}} = \frac{2}{N} C^\top I'' F C \mathbf{b}, \text{ where } F = \mathrm{diag}(f(x_0), \ldots, f(x_N))$$

- Interpretation: coeff. space $\mathbf{a}$, $\mathbf{b} \leftarrow$ DCT $\rightarrow p(\mathbf{x})$, $q(\mathbf{x})$ value space

Rational interpolation and least squares

- Now note, $\|p - fq\|_N = \|p - \hat{p}\|_N = \|\mathbf{a} - \hat{\mathbf{a}}\|_2$

Rational interpolation and least squares

- Now note, $\|p - fq\|_N = \|p - \hat{p}\|_N = \|\mathbf{a} - \hat{\mathbf{a}}\|_2$
- Solution to rational interpolation and least squares problem is to take $\hat{\mathbf{a}} = (\mathbf{a}, \tilde{\mathbf{a}})^\top$, and choose $\mathbf{b}$ to minimise $\|\tilde{\mathbf{a}}\|_2$ such that $\|\mathbf{b}\|_2 = 1$.

Rational interpolation and least squares

- Now note, $\|p - fq\|_N = \|p - \hat{p}\|_N = \|\mathbf{a} - \hat{\mathbf{a}}\|_2 = \|\tilde{\mathbf{a}}\|_2$
- Solution to rational interpolation and least squares problem is to take $\hat{\mathbf{a}} = (\mathbf{a}, \tilde{\mathbf{a}})^\top$, and choose $\mathbf{b}$ to minimise $\|\tilde{\mathbf{a}}\|_2$ such that $\|\mathbf{b}\|_2 = 1$.

Rational interpolation and least squares

- Now note, $\|p - fq\|_N = \|p - \hat{p}\|_N = \|\mathbf{a} - \hat{\mathbf{a}}\|_2 = \|\tilde{\mathbf{a}}\|_2$
- Solution to rational interpolation and least squares problem is to take $\hat{\mathbf{a}} = (\mathbf{a}, \tilde{\mathbf{a}})^\top$, and choose $\mathbf{b}$ to minimise $\|\tilde{\mathbf{a}}\|_2$ such that $\|\mathbf{b}\|_2 = 1$.
- We can view $\hat{\mathbf{a}} = \frac{2}{N} C^\top I'' F C \mathbf{b}$ using $Z \in \mathbb{C}^{m+1 \times n+1}$, $\tilde{Z} \in \mathbb{C}^{N-m \times n+1}$:

$$\begin{pmatrix} \mathbf{a} \\ \tilde{\mathbf{a}} \end{pmatrix} = \begin{pmatrix} Z \\ \tilde{Z} \end{pmatrix} \mathbf{b}$$

Rational interpolation and least squares

- Now note, $\|p - fq\|_N = \|p - \hat{p}\|_N = \|\mathbf{a} - \hat{\mathbf{a}}\|_2 = \|\tilde{\mathbf{a}}\|_2$
- Solution to rational interpolation and least squares problem is to take $\hat{\mathbf{a}} = (\mathbf{a}, \tilde{\mathbf{a}})^\top$, and choose $\mathbf{b}$ to minimise $\|\tilde{\mathbf{a}}\|_2$ such that $\|\mathbf{b}\|_2 = 1$.
- We can view $\hat{\mathbf{a}} = \frac{2}{N} C^\top I'' F C \mathbf{b}$ using $Z \in \mathbb{C}^{m+1 \times n+1}$, $\tilde{Z} \in \mathbb{C}^{N-m \times n+1}$:

$$\begin{pmatrix} \mathbf{a} \\ \tilde{\mathbf{a}} \end{pmatrix} = \begin{pmatrix} Z \\ \tilde{Z} \end{pmatrix} \mathbf{b}$$

- Then $\tilde{\mathbf{a}} = \tilde{Z}\mathbf{b}$, so the solution is:

$$\text{minimise } \|\tilde{Z}\mathbf{b}\|_2 \text{ such that } \|\mathbf{b}\|_2 = 1,$$

then compute $\mathbf{a} = Z\mathbf{b}$.

Rational interpolation and least squares

- Now note, $\|p - fq\|_N = \|p - \hat{p}\|_N = \|\mathbf{a} - \hat{\mathbf{a}}\|_2 = \|\tilde{\mathbf{a}}\|_2 = \|\tilde{Z}\mathbf{b}\|_2$
- Solution to rational interpolation and least squares problem is to take $\hat{\mathbf{a}} = (\mathbf{a}, \tilde{\mathbf{a}})^\top$, and choose $\mathbf{b}$ to minimise $\|\tilde{\mathbf{a}}\|_2$ such that $\|\mathbf{b}\|_2 = 1$.
- We can view $\hat{\mathbf{a}} = \frac{2}{N} C^\top I'' F C \mathbf{b}$ using $Z \in \mathbb{C}^{m+1 \times n+1}$, $\tilde{Z} \in \mathbb{C}^{N-m \times n+1}$:

$$\begin{pmatrix} \mathbf{a} \\ \tilde{\mathbf{a}} \end{pmatrix} = \begin{pmatrix} Z \\ \tilde{Z} \end{pmatrix} \mathbf{b}$$

- Then $\tilde{\mathbf{a}} = \tilde{Z}\mathbf{b}$, so the solution is:

$$\text{minimise } \|\tilde{Z}\mathbf{b}\|_2 \text{ such that } \|\mathbf{b}\|_2 = 1,$$

then compute $\mathbf{a} = Z\mathbf{b}$.

Uniqueness and robustness

- If $m + n = N$, then $\tilde{Z}$ is a $N - m \times n + 1 = n \times n + 1$ matrix, so has a nontrivial kernel. I.e. we can find $\mathbf{b}$ such that $\|\tilde{Z}\mathbf{b}\|_2 = 0$.

Uniqueness and robustness

- If $m + n = N$, then $\tilde{Z}$ is a $N - m \times n + 1 = n \times n + 1$ matrix, so has a nontrivial kernel. I.e. we can find $\mathbf{b}$ such that $\|\tilde{Z}\mathbf{b}\|_2 = 0$.
- If $m + n < N$, then $\tilde{Z}$ may **not** have a nontrivial kernel.

Uniqueness and robustness

- If $m + n = N$, then $\tilde{Z}$ is a $N - m \times n + 1 = n \times n + 1$ matrix, so has a nontrivial kernel. I.e. we can find $\mathbf{b}$ such that $\|\tilde{Z}\mathbf{b}\|_2 = 0$.
- If $m + n < N$, then $\tilde{Z}$ may **not** have a nontrivial kernel.
- Either way, we are finding the **minimal singular vector** of $\tilde{Z}$. Easily done with SVD.

Uniqueness and robustness

- If $m + n = N$, then $\tilde{Z}$ is a $N - m \times n + 1 = n \times n + 1$ matrix, so has a nontrivial kernel. I.e. we can find $\mathbf{b}$ such that $\|\tilde{Z}\mathbf{b}\|_2 = 0$.
- If $m + n < N$, then $\tilde{Z}$ may **not** have a nontrivial kernel.
- Either way, we are finding the **minimal singular vector(s)** of $\tilde{Z}$. Easily done with SVD.

Uniqueness and robustness

- If $m + n = N$, then $\tilde{Z}$ is a $N - m \times n + 1 = n \times n + 1$ matrix, so has a nontrivial kernel. I.e. we can find $\mathbf{b}$ such that $\|\tilde{Z}\mathbf{b}\|_2 = 0$.
- If $m + n < N$, then $\tilde{Z}$ may **not** have a nontrivial kernel.
- Either way, we are finding the **minimal singular vector(s)** of $\tilde{Z}$. Easily done with SVD.
- If there are d minimal singular vectors, this corresponds to **non-uniqueness** and a risk of **spurious poles**. Most straightforward thing to do for robustness is to reduce n by $d - 1$ and start again.

Uniqueness and robustness

- If $m + n = N$, then $\tilde{Z}$ is a $N - m \times n + 1 = n \times n + 1$ matrix, so has a nontrivial kernel. I.e. we can find $\mathbf{b}$ such that $\|\tilde{Z}\mathbf{b}\|_2 = 0$.
- If $m + n < N$, then $\tilde{Z}$ may **not** have a nontrivial kernel.
- Either way, we are finding the **minimal singular vector(s)** of $\tilde{Z}$. Easily done with SVD.
- If there are d minimal singular vectors, this corresponds to **non-uniqueness** and a risk of **spurious poles**. Most straightforward thing to do for robustness is to reduce n by $d - 1$ and start again.
- Repeat this until we have a unique $\mathbf{b}$. The resulting r should have no spurious poles!

Robustness

- In Chebfun, there is the `ratinterp` command that does all of this for abritrary points in $\mathbb{C}$.

Robustness

- In Chebfun, there is the `ratinterp` command that does all of this for abritrary points in $\mathbb{C}$.
- In machine precision arithmetic, all the robustness procedure of reducing the degree of q is done to a tolerance parameter *tol*.

Robustness

- In Chebfun, there is the `ratinterp` command that does all of this for abritrary points in $\mathbb{C}$.
- In machine precision arithmetic, all the robustness procedure of reducing the degree of q is done to a tolerance parameter *tol*.
- We reduce n by $d-1$ if there are $d-1$ singular vectors with singular values within *tol* of the minimal singular value.

Robustness

- In Chebfun, there is the `ratinterp` command that does all of this for abritrary points in $\mathbb{C}$.
- In machine precision arithmetic, all the robustness procedure of reducing the degree of q is done to a tolerance parameter *tol*.
- We reduce n by $d-1$ if there are $d-1$ singular vectors with singular values within *tol* of the minimal singular value.
- We remove trailing coefficients of $\mathbf{a}$ and $\mathbf{b}$ that are smaller than *tol*, further reducing the degrees of p and q.

Robustness

- In Chebfun, there is the `ratinterp` command that does all of this for abritrary points in $\mathbb{C}$.
- In machine precision arithmetic, all the robustness procedure of reducing the degree of q is done to a tolerance parameter *tol*.
- We reduce n by $d-1$ if there are $d-1$ singular vectors with singular values within *tol* of the minimal singular value.
- We remove trailing coefficients of $\mathbf{a}$ and $\mathbf{b}$ that are smaller than *tol*, further reducing the degrees of p and q.
- Key point: if we ask for $r \in \mathcal{R}_{m,n}$, we will in fact get $r \in \mathcal{R}_{\mu,\nu}$ with $\mu \leq m$, $\nu \leq n$. This is the *exact type* of the interpolant.

Rational interpolation and least squares: Literature

- We call this the PGVT approach after Pachón, Gonnet, Van Deun, and Trefethen
- PGV 2011 introduces the novel approach for interpolation in arbitrary points
- GPT 2011 extends to least squares approximation, enabling robustness, but only for roots of unity
- Covered nicely in Trefethen's book Approximation Theory and Approximation Practice
- W 2013 discusses least squares for Chebyshev points, gives some heuristics for parameters and its usage, and demonstrates with some interesting ODE examples.

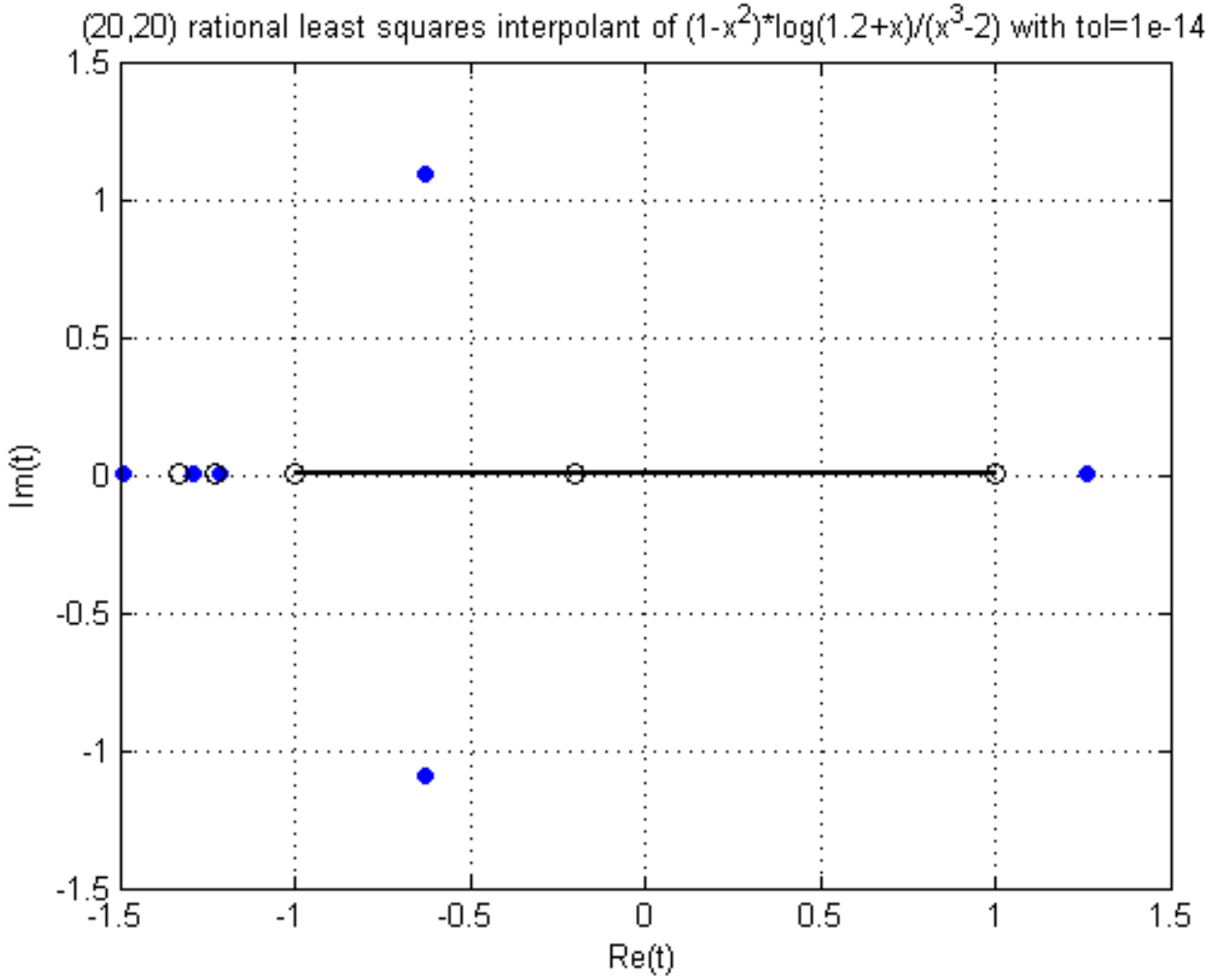


Figure : Ratinterp returns an $(20, 6)$ exact type rational least squares approximant with appropriate singularity structure.

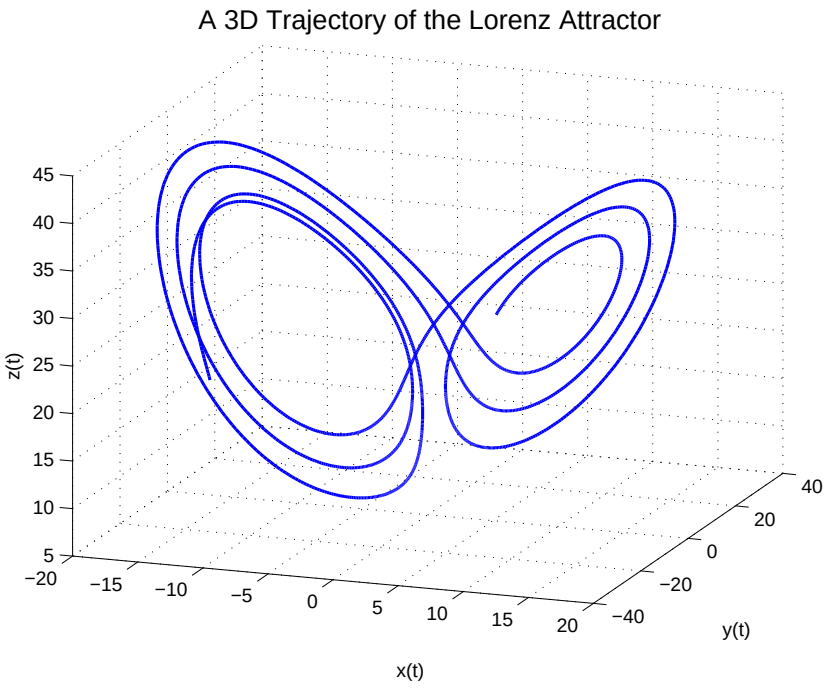
Lorenz Attractor

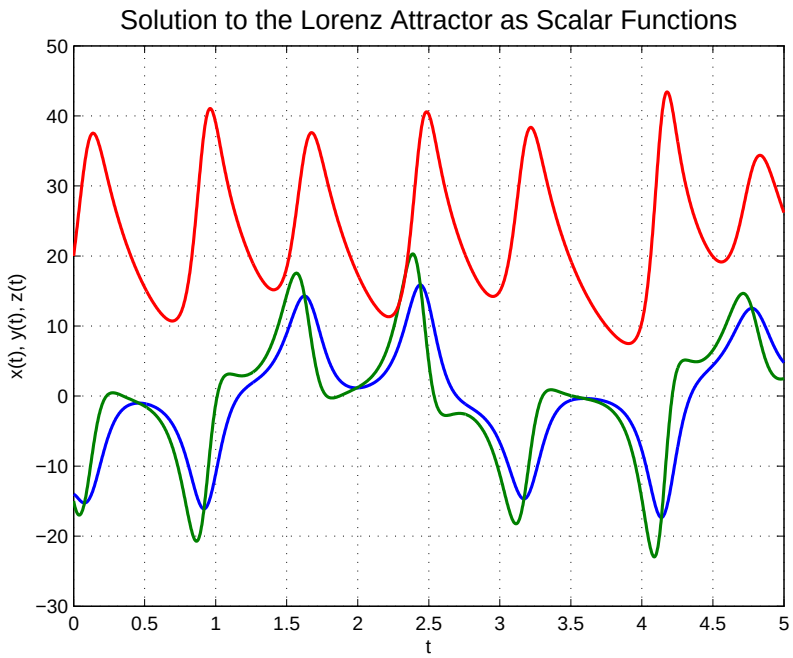
- The Lorenz system is a system of ODEs first studied by Edward Lorenz in the 1960s as a simplified model of convection rolls in the upper atmosphere.

$$\begin{aligned}\frac{\mathrm{d}x}{\mathrm{d}t} &= 10(y-x)\\ \frac{\mathrm{d}y}{\mathrm{d}t} &= 28x - y - xz\\ \frac{\mathrm{d}z}{\mathrm{d}t} &= -8z/3 + xy.\end{aligned}$$

- It is an example of a chaotic system.

Lorenz Attractor: Numerical Solution





- The two straightforward viewpoints for the solution are as a trajectory in 3 dimensions, or as three scalar functions.

Lorenz Attractor: Analytical Solution

- However, a natural way to see the analytical solution is as a function of a complex variable (see "Complex Singularities of the Lorenz Attractor", Viswanath and Sahutoglu 2010)

Lorenz Attractor: Analytical Solution

- However, a natural way to see the analytical solution is as a function of a complex variable (see "Complex Singularities of the Lorenz Attractor", Viswanath and Sahutoglu 2010)
- The analytical solution can be expressed **locally** as a Psi-series:

$$
\begin{aligned}
x(t) &= \frac{P_{-1}(\eta)}{t-t_0} + P_0(\eta) + P_1(\eta)(t-t_0) + P_2(\eta)(t-t_0)^2 + \dots, \\
y(t) &= \frac{Q_{-2}(\eta)}{(t-t_0)^2} + \frac{Q_{-1}(\eta)}{t-t_0} + Q_0(\eta) + Q_1(\eta)(t-t_0) + Q_2(\eta)(t-t_0)^2 + \dots, \\
z(t) &= \frac{R_{-2}(\eta)}{(t-t_0)^2} + \frac{R_{-1}(\eta)}{t-t_0} + R_0(\eta) + R_1(\eta)(t-t_0) + R_2(\eta)(t-t_0)^2 + \dots.
\end{aligned}
$$

- Here $\eta = \log(b(t-t_0))$ where $b = \pm i$, and the P_js, Q_js and R_js are polynomials. x has order 1 pseudo-pole at t_0; y and z have order 2 pseudo-poles.

Lorenz attractor in the complex plane

- The solution on the previous slide is 3 chebfuns of degrees $N = 462$, 509 and 498.

Lorenz attractor in the complex plane

- The solution on the previous slide is 3 chebfuns of degrees $N = 462$, 509 and 498.
- A good strategy for `ratinterp` to find $r \in \mathcal{R}_{m,n}$ is to set $m \approx N/2$ and n big enough to find some singularities, on N Chebyshev points in the interval.

Lorenz attractor in the complex plane

- The solution on the previous slide is 3 chebfuns of degrees $N = 462$, 509 and 498.
- A good strategy for `ratinterp` to find $r \in \mathcal{R}_{m,n}$ is to set $m \approx N/2$ and n big enough to find some singularities, on N Chebyshev points in the interval.
- We take the *tol* parameter to be 10^{-12}, because there will be noise with magnitude around 10^{-14} in the numerical solution.

Lorenz attractor in the complex plane

- The solution on the previous slide is 3 chebfuns of degrees $N = 462$, 509 and 498.
- A good strategy for `ratinterp` to find $r \in \mathcal{R}_{m,n}$ is to set $m \approx N/2$ and n big enough to find some singularities, on N Chebyshev points in the interval.
- We take the *tol* parameter to be 10^{-12}, because there will be noise with magnitude around 10^{-14} in the numerical solution.
- The command `ratinterp(u(:,1), 231, 20, 463, [], 1e-12)` is computed in a fraction of a second.

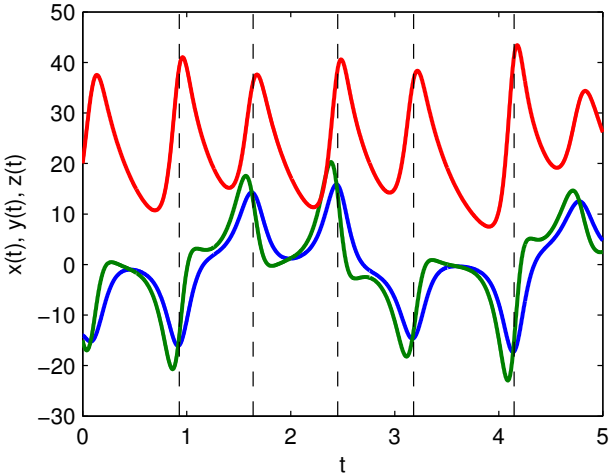
Lorenz attractor in the complex plane

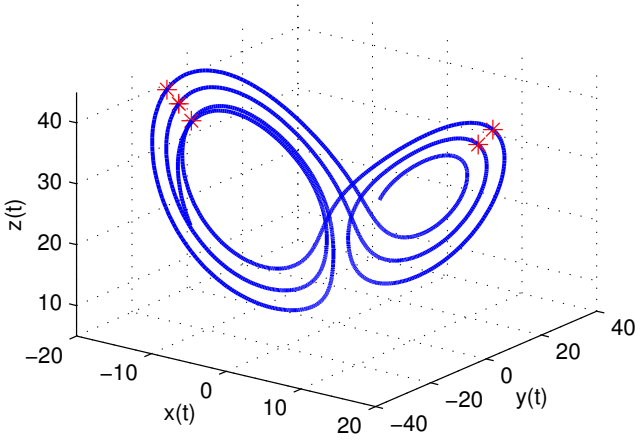
- The solution on the previous slide is 3 chebfuns of degrees $N = 462$, 509 and 498.
- A good strategy for `ratinterp` to find $r \in \mathcal{R}_{m,n}$ is to set $m \approx N/2$ and n big enough to find some singularities, on N Chebyshev points in the interval.
- We take the *tol* parameter to be 10^{-12}, because there will be noise with magnitude around 10^{-14} in the numerical solution.
- The command `ratinterp(u(:,1), 231, 20, 463, [], 1e-12)` is computed in a fraction of a second.
- A type $(173, 10)$ rational function is returned, with no spurious poles.

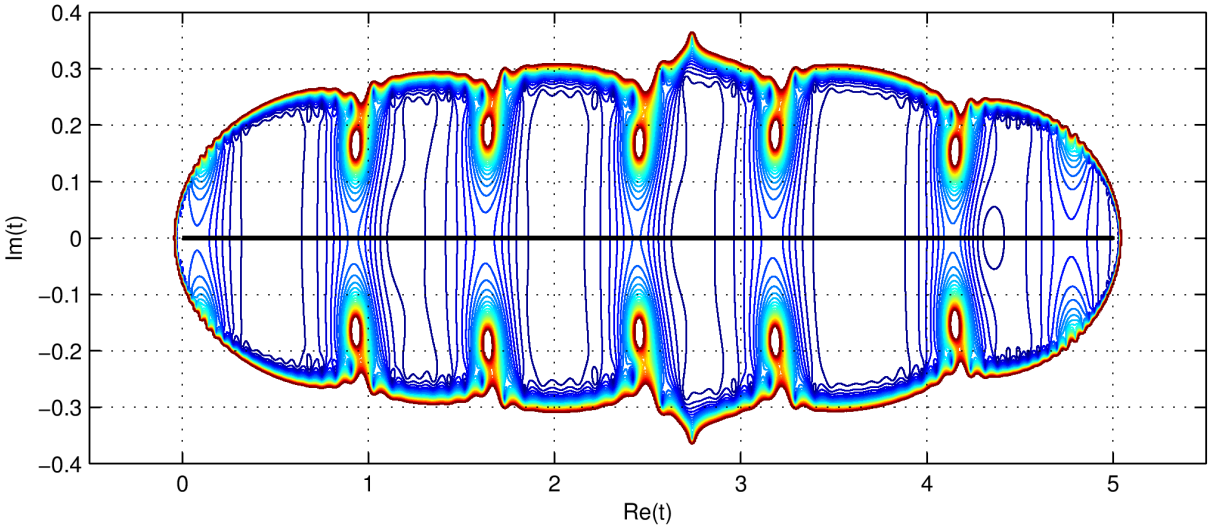
Lorenz attractor in the complex plane

- The solution on the previous slide is 3 chebfuns of degrees $N = 462$, 509 and 498.
- A good strategy for `ratinterp` to find $r \in \mathcal{R}_{m,n}$ is to set $m \approx N/2$ and n big enough to find some singularities, on N Chebyshev points in the interval.
- We take the *tol* parameter to be 10^{-12}, because there will be noise with magnitude around 10^{-14} in the numerical solution.
- The command `ratinterp(u(:,1), 231, 20, 463, [], 1e-12)` is computed in a fraction of a second.
- A type $(173, 10)$ rational function is returned, with no spurious poles.
- Similarly, we get type $(227, 10)$ and $(221, 10)$ rational approximants for the other two components.

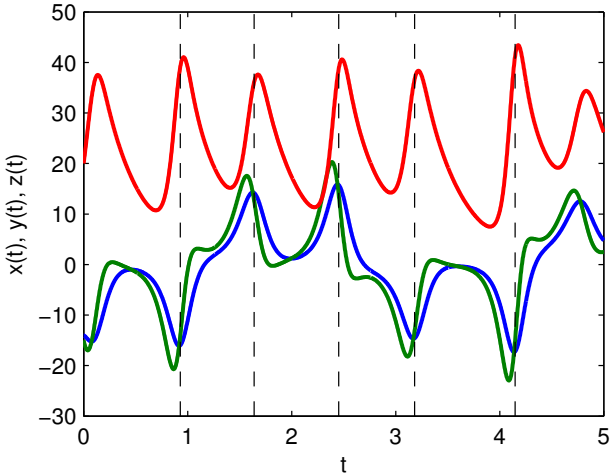
Lorenz attractor in the complex plane

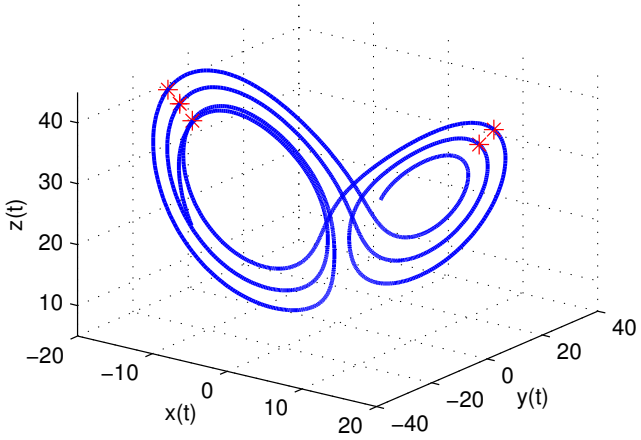


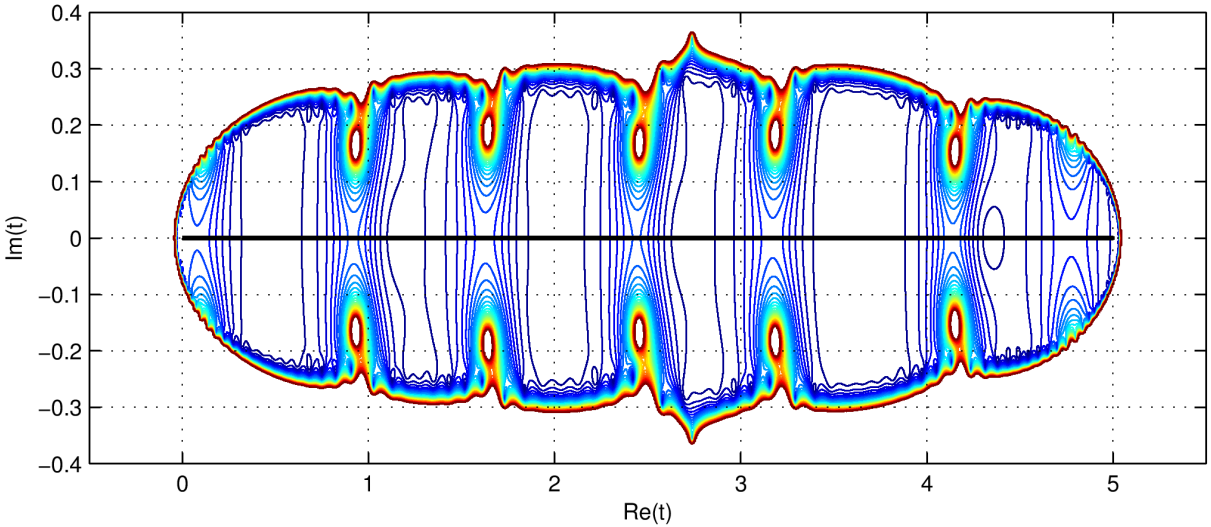




Lorenz attractor in the complex plane







There are still open questions related to the analysis of the Lorenz system!

Lotka–Volterra

- The Lotka–Volterra system is a simple model for the population of predators (y) and their prey (x).

$$\frac{dx}{dt} = \alpha x - \beta xy, \quad \frac{dy}{dt} = -\gamma y + \delta xy.$$

$$x, y, \alpha, \beta, \gamma, \delta > 0.$$

Lotka–Volterra

- The Lotka–Volterra system is a simple model for the population of predators (y) and their prey (x).

$$\frac{dx}{dt} = \alpha x - \beta xy, \quad \frac{dy}{dt} = -\gamma y + \delta xy.$$

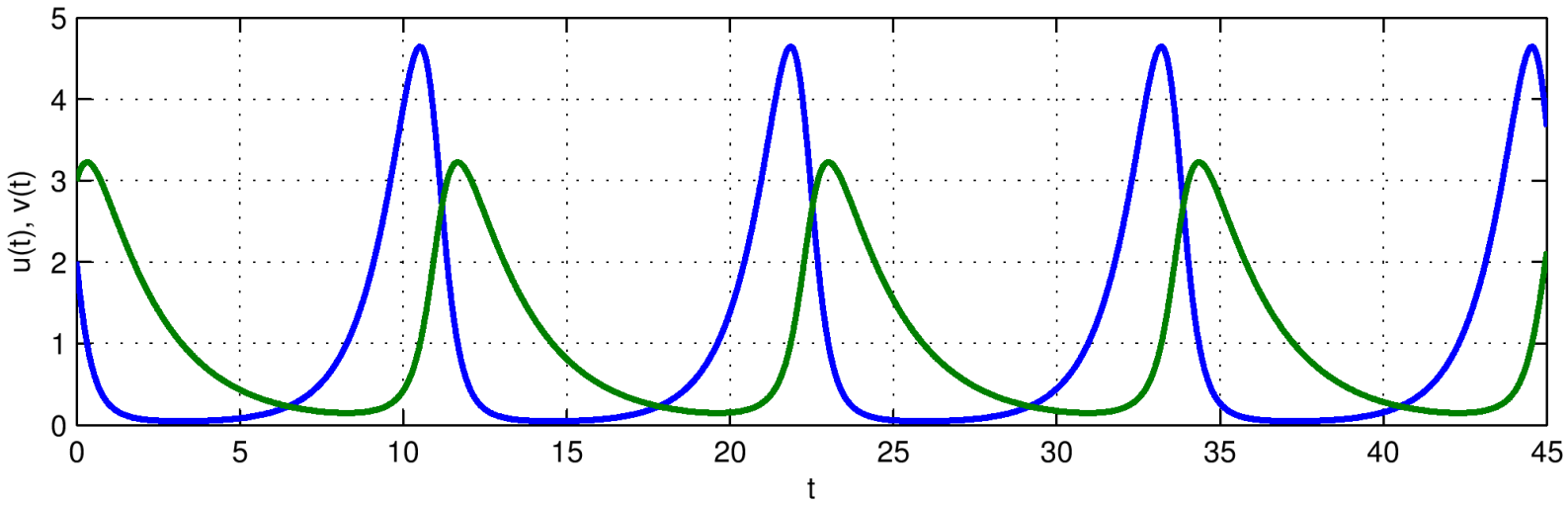
$$x, y, \alpha, \beta, \gamma, \delta > 0.$$

- The analysis is quite well understood compared to 3D systems. E.g. we know that there are **always** Psi-series singularities in the complex plane.

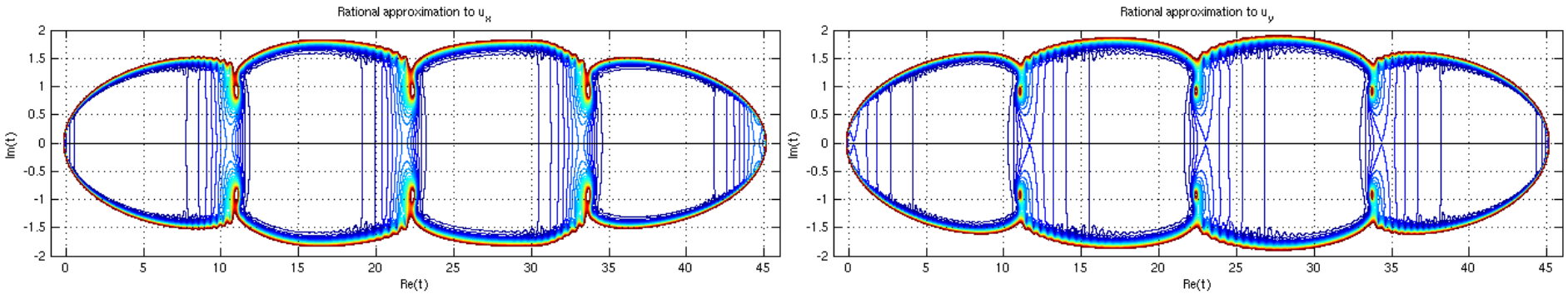
Lotka–Volterra

- We solve using $\alpha = \beta = 1/2$, $\gamma = \delta = 1$, $x(0) = 2$, $y(0) = 3$.

- We solve using $\alpha = \beta = 1/2$, $\gamma = \delta = 1$, $x(0) = 2$, $y(0) = 3$.
- The populations fluctuate periodically.



- The Chebfuns are of degrees 743 and 737. We compute $(371, 20)$ and $(366, 20)$ `ratinterp` least squares approximants on 743 and 737 Chebyshev points.
- `ratinterp` returns type $(297, 6)$ and $(287, 6)$ exact type approximants.



Error estimates

- Errors creep in at every stage of the approximation.

Error estimates

- Errors creep in at every stage of the approximation.
- Error in solving ODE is (typically) $\sim 10^{-13}$

Error estimates

- Errors creep in at every stage of the approximation.
- Error in solving ODE is (typically) $\sim 10^{-13}$
- Error in ratinterp on real line (with tolerance 10^{-12}) is $\sim 10^{-10}$

Error estimates

- Errors creep in at every stage of the approximation.
- Error in solving ODE is (typically) $\sim 10^{-13}$
- Error in ratinterp on real line (with tolerance 10^{-12}) is $\sim 10^{-10}$
- Error in ratinterp in region up to singularity is $\sim 10^{-8}$

Error estimates

- Errors creep in at every stage of the approximation.
- Error in solving ODE is (typically) $\sim 10^{-13}$
- Error in ratinterp on real line (with tolerance 10^{-12}) is $\sim 10^{-10}$
- Error in ratinterp in region up to singularity is $\sim 10^{-8}$
- Error in location of singularities is $\sim 10^{-7}$ for poles, $\sim 10^{-2}$ for branch points.

Error estimates

- Errors creep in at every stage of the approximation.
- Error in solving ODE is (typically) $\sim 10^{-13}$
- Error in ratinterp on real line (with tolerance 10^{-12}) is $\sim 10^{-10}$
- Error in ratinterp in region up to singularity is $\sim 10^{-8}$
- Error in location of singularities is $\sim 10^{-7}$ for poles, $\sim 10^{-2}$ for branch points.
- However, the existence of each singularity is **reliable**. Afterwards other methods such as steepest ascent can be used to gain accuracy.

Error estimates

- Errors creep in at every stage of the approximation.
- Error in solving ODE is (typically) $\sim 10^{-13}$
- Error in ratinterp on real line (with tolerance 10^{-12}) is $\sim 10^{-10}$
- Error in ratinterp in region up to singularity is $\sim 10^{-8}$
- Error in location of singularities is $\sim 10^{-7}$ for poles, $\sim 10^{-2}$ for branch points.
- However, the existence of each singularity is **reliable**. Afterwards other methods such as steepest ascent can be used to gain accuracy.
- This makes it good for automated singularity location in parabolic PDEs, parametrised ODEs etc. (see Weideman 2003).

Vector-valued rational approximation

- Recently I realised the PGVT approach can easily be modified to **simultaneously** compute rational approximants that each have the **same** singularities, AKA *vector-valued rational approximation*. (I am grateful to D. Viswanath for a fruitful correspondence).

Vector-valued rational approximation

- Recently I realised the PGVT approach can easily be modified to **simultaneously** compute rational approximants that each have the **same** singularities, AKA *vector-valued rational approximation*. (I am grateful to D. Viswanath for a fruitful correspondence).
- For 3D systems, the goal was to find 3 minimal singular vectors $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$, of 3 matrices $\tilde{Z}_1$, $\tilde{Z}_2$, $\tilde{Z}_3$, hoping that $\mathbf{b}_1 \approx \mathbf{b}_2 \approx \mathbf{b}_3$.

Vector-valued rational approximation

- Recently I realised the PGVT approach can easily be modified to **simultaneously** compute rational approximants that each have the **same** singularities, AKA *vector-valued rational approximation*. (I am grateful to D. Viswanath for a fruitful correspondence).
- For 3D systems, the goal was to find 3 minimal singular vectors $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$, of 3 matrices $\tilde{Z}_1$, $\tilde{Z}_2$, $\tilde{Z}_3$, hoping that $\mathbf{b}_1 \approx \mathbf{b}_2 \approx \mathbf{b}_3$.
- Instead, find 1 minimal singular vector $\mathbf{b}$ of the block matrix:

$$\tilde{Z} = \begin{pmatrix} \tilde{Z}_1 \\ \tilde{Z}_2 \\ \tilde{Z}_3 \end{pmatrix}$$

Vector-valued rational approximation

- Recently I realised the PGVT approach can easily be modified to **simultaneously** compute rational approximants that each have the **same** singularities, AKA *vector-valued rational approximation*. (I am grateful to D. Viswanath for a fruitful correspondence).
- For 3D systems, the goal was to find 3 minimal singular vectors $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$, of 3 matrices $\tilde{Z}_1$, $\tilde{Z}_2$, $\tilde{Z}_3$, hoping that $\mathbf{b}_1 \approx \mathbf{b}_2 \approx \mathbf{b}_3$.
- Instead, find 1 minimal singular vector $\mathbf{b}$ of the block matrix:

$$\tilde{Z} = \begin{pmatrix} \tilde{Z}_1 \\ \tilde{Z}_2 \\ \tilde{Z}_3 \end{pmatrix}$$

- Then $\mathbf{a}_1 = Z_1\mathbf{b}$, $\mathbf{a}_2 = Z_2\mathbf{b}$, $\mathbf{a}_3 = Z_3\mathbf{b}$.

Summary

- PGVT approach is based on SVD of the linearised rational interpolation problem

Summary

- PGVT approach is based on SVD of the linearised rational interpolation problem
- Uses the singular values to reduce the number of poles

Summary

- PGVT approach is based on SVD of the linearised rational interpolation problem
- Uses the singular values to reduce the number of poles
- Usage: use the degree of a chebfun interpolant (N) to choose degree of numerator, $m \approx N/2$

Summary

- PGVT approach is based on SVD of the linearised rational interpolation problem
- Uses the singular values to reduce the number of poles
- Usage: use the degree of a chebfun interpolant (N) to choose degree of numerator, $m \approx N/2$
- An interesting application: find complex singularities of ODEs.

Summary

- PGVT approach is based on SVD of the linearised rational interpolation problem
- Uses the singular values to reduce the number of poles
- Usage: use the degree of a chebfun interpolant (N) to choose degree of numerator, $m \approx N/2$
- An interesting application: find complex singularities of ODEs.
- The PGVT approach can be modified to compute vector-valued rational approximants too.

Summary

- PGVT approach is based on SVD of the linearised rational interpolation problem
- Uses the singular values to reduce the number of poles
- Usage: use the degree of a chebfun interpolant (N) to choose degree of numerator, $m \approx N/2$
- An interesting application: find complex singularities of ODEs.
- The PGVT approach can be modified to compute vector-valued rational approximants too.
- Thank you for your attention!