COMPUTING COMPLEX SINGULARITIES OF DIFFERENTIAL EQUATIONS WITH CHEBFUN

AUTHOR: MARCUS WEBB[*] AND ADVISOR: LLOYD N. TREFETHEN[†]

Abstract. Given a solution to an ordinary differential equation (ODE) on a time interval, the solution for complex-valued time may be of interest, in particular whether the solution is singular at some complex time value. How can the solution be approximated in the complex plane using only the data on the interval? A polynomial approximation of the solution always fails to capture singularities; to extrapolate solutions with singularities, approximation with rational functions is more appropriate. In this paper, a robust form of rational interpolation and least-squares approximation, due to Pachón, Gonnet et al., is discussed and tested. It is found that the method avoids the issue of spurious poles found by many standard rational approximations, but that it is not suitable when a high degree of accuracy is required.

Key words. Complex singularities, ordinary differential equations, numerical analytic continuation, rational interpolation, Lorenz attractor, Lotka-Volterra, three-body problem.

1. Introduction.

Recent decades have seen increased scientific interest in numerous questions associated with the location of complex singularities of differential equations. For example, the Painlevé equations, whose solutions have many complex singularities, are growing in importance due to the long list of problems described by them: the scattering of neutrons off heavy nuclei, the statistics of the zeros of the Riemann zeta function on the critical line $\mathrm{Re}(z) = 1/2$ and, amongst many others, random matrix theory [3]. As a result, there have been some interesting publications on the numerical methodology and mathematical analysis for such problems [7], [21], [23].

In this paper we explore a central problem in the area, analytic continuation. Suppose one has solved an ordinary differential equation (ODE) on a time interval, and the solution for complex-valued time is of interest, in particular whether the solution is singular at some complex time value. How can the solution be approximated in the complex plane using only the data on the interval?

Let $f : [0, T] \to \mathbb{R}$ be an analytic function solving a given ODE problem, and let $G \subset \mathbb{C}$ be an open, connected domain in the complex plane, containing the interval $[0, T]$. An analytic function $\tilde{f} \to \mathbb{C}$ is an *analytic continuation* of f if $\tilde{f}$ is analytic in G and $\tilde{f}|_{[0,T]} = f$. For any analytic f, there exists an analytic continuation to some open set containing $[0, T]$ by evaluation of the Taylor series inside their discs of convergence, and by the identity theorem in complex analysis, analytic continuations to connected open sets are unique. We extend the definition to include the case where $\tilde{f}$ has a singularity at t_0 (meaning $\tilde{f}$ is not analytic) if we can analytically continue f to any set G which is missing arbitrarily small discs centered at t_0.

If you take a polynomial interpolant p of f on points in $[0, T]$, and evaluate p in the complex plane, as an entire function it could not possibly capture any singularities in $\tilde{f}$. Another issue with polynomial approximations is that they necessarily blow up as you go out towards infinity in the complex plane. However, a polynomial interpolant on Chebyshev points, scaled and shifted to $[0, T]$, is accurate in the *Bernstein ellipse* associated with $\tilde{f}$ and $[0, T]$. For the interval $[-1, 1]$ the boundary of the Bernstein

[*] Cambridge Centre for Analysis, Wilberforce Road, Cambridge, CB3 0WA, UK (m.d.webb@maths.cam.ac.uk, http://damtp.cam.ac.uk/people/mdw42).

[†] Mathematical Institute, Oxford University, Oxford OX1 3LB, UK (trefethen@maths.ox.ac.uk, http://people.maths.ox.ac.uk/trefethen/).

ellipse associated with $\tilde{f}$ is described by

$$E_\rho(\theta) = \frac{1}{2}\left(\rho e^{2\pi i\theta} + \rho^{-1}e^{-2\pi i\theta}\right), \quad \theta \in [0, 2\pi), \tag{1.1}$$

where ρ is taken to be as large as possible so that no singularities of $\tilde{f}$ lie inside the ellipse. For arbitrary intervals, the ellipse is scaled and translated. By accurate, we mean that the interpolants converge geometrically to $\tilde{f}$ inside $E_\rho([0, 2\pi))$ [19, Ch.8].

Rational functions are those that can be written as a quotient: $r(z) = p(z)/q(z)$ where p and q are polynomials. Approximations using rational functions are well suited to numerical analytic continuation for singular functions, because a rational function has a singularity at each root of q [19, Ch. 23]. Padé approximants, based on matching the first few terms of the Taylor series of the function we wish to approximate, are the most well known example of rational approximants [1]. Common rational approximation methods have problems, essentially due to the fact that they have as many singularities as the denominator q has roots. As will be explained more in section 3, only exceptional methods do not produce unwanted singularities, even in exact arithmetic. However, there have been developments in algorithms for rational interpolation that are claimed to overcome this issue. Two publications in 2011, Gonnet et al. [9] and Pachón et al. [16], stemming from research in Pachón's D.Phil. thesis (2010) [15], describe an algorithm for rational interpolation that is fast, stable and robust.

In this paper we investigate the use of the algorithm in a straightforward method: First solve the ODE numerically on $[0, T]$, then use numerical analytic continuation to extend the solution into the complex plane with a rational function whose singularities can be found and analysed. This follows an investigation by Weideman [23], although his study had a different emphasis: the equations he considered had solutions in two variables $u(x, t)$, where x is a spatial variable and t is time, with the aim to compute singularities of u for complex x and observe their dynamics as t varies.

The implementation of the method is in Chebfun (www.maths.ox.ac.uk/chebfun), an open-source software project led by Nick Trefethen and Nick Hale at Oxford University and Toby Driscoll of the University of Delaware. Chebfun is an extension of MATLAB which overloads common vector and matrix operations to manipulate functions and operators. The intention is that the commands should feel symbolic, as if we were working with the actual functions, but that the underlying computations are numeric and therefore fast. Experience with MATLAB or Chebfun is not necessary to understand or appreciate the results in this paper.

There is good reasoning behind the use of this particular software package, more than just mere convenience. Polynomial interpolation in Chebyshev points is extremely reliable for smooth functions [18], whereas analytic continuation is in fact an ill-posed problem (small changes in function data can cause large changes in the continuation). As we shall see in the next section, simply the degree of the Chebyshev interpolant produced by Chebfun (called a *chebfun*) gives us an estimate for ρ, the parameter of the Bernstein ellipse associated with the underlying function, and appropriate degrees for p and q in our rational approximation.

In Section 2 we discuss in detail the proposed method mentioned above. Section 3 concerns the results of preliminary experiments using the robust rational interpolation algorithm, leading to heuristics for parameters of the method such as the degrees of p and q. The main section is Section 4, in which we illustrate the use of the method on some ODE problems that are fascinating in their own right. Section 5 is devoted to discussion, conclusions, and possibilities for further work in this area.

2. The Proposed Method. To solve the ODEs on $[0, T]$ we use the chebop methods built into Chebfun because they are naturally suited for returning a chebfun as a numerical solution [6]. However, for some more complicated initial value problems, for example the Lorenz attractor, this approach can fail to converge without a good initial guess for the solution. In this case we will use Chebfun's overload of the MATLAB command `ode113`.

We will denote the underlying analytic solution to the ODE by $f : \mathbb{C} \setminus \{t_j\}_{j\in J} \to \mathbb{C}$, with singularity set $\{t_j\}_{j\in J}$, and our numerically computed chebfun solution by $u : [0, T] \to \mathbb{R}$. Note that u is a polynomial, whose degree (denoted N) is calculated automatically in Chebfun. The degree can be loosely described to be such that any computed solution with higher degree would have to have some coefficients that are zero up to machine precision, but for a better understanding see the chebfun guide on the website, or the original chebop paper [6].

For the analytic continuation step, we ask for a rational function, $r = p/q$ of type (m, n) (i.e. the degrees of p and q are m and n respectively) to approximate f, using u. To do this we use the Chebfun command `ratinterp`, which is a robust implementation of linearised rational interpolation. Here we will explain mathematically how `ratinterp` works, based on the algorithms described in [9] and [16].

Let $\{x_0, \ldots, x_N\}$ be the set of Chebyshev points on $[-1, 1]$, $x_j = \cos(\frac{j\pi}{N})$. The rational interpolation problem for f on $\mathbf{x} = (x_0, \ldots, x_N)^\top$ is as follows: Find a rational function r of type (m, n), such that $m + n = N$ and

$$r(\mathbf{x}) = f(\mathbf{x}), \tag{2.1}$$

where $r(\mathbf{x})$ gives the vector with elements $r(x_j)$ for each j. A solution does not always exist: consider the type $(1, 1)$ interpolant r such that $r(\pm 1) = 0$ and $r(0) = 1$. But when it does exist it is unique, as can be checked by elementary methods.

To avoid the problem of nonexistence, we must take a more general approach. Consider the following bilinear form on continuous functions:

$$\langle f, g\rangle_N = \frac{2}{N} \sum_{i=0}^{N}{}'' f(x_i)\overline{g(x_i)}. \tag{2.2}$$

Here the $''$ indicates having the first and last entries halved. It is an inner product on the space Π_N of polynomials of degree less than or equal to N, and it satisfies the following orthogonality property:

$$\langle T_k, T_l\rangle_N = \begin{cases} 2 & \text{if } k = l = 0, N, \\ 1 & \text{if } k = l \neq 0, N, \\ 0 & \text{if } k \neq l. \end{cases} \tag{2.3}$$

Here T_k is the Chebyshev polynomial of degree k, defined by $T_k(x) = \cos(k \cos^{-1}(x))$ for $x \in [-1, 1]$. Consequently, (2.3) can be proved using trigonometric identities. Now, define $\|\cdot\|_N = (\langle\cdot,\cdot\rangle_N)^{\frac{1}{2}}$, and consider finding $p \in \Pi_m$ and $q \in \Pi_n$ such that

$$\text{minimise } \|p - fq\|_N \text{ such that } \|q\|_N = 1. \tag{2.4}$$

The normalisation of q is necessary to avoid the trivial solution $p = q = 0$. Any solution $r = p/q$ to (2.1) can be made to satisfy (2.4). Conversely, we will see by the following linear algebra that if $m + n = N$, then (2.4) has a solution such that $\|p - fq\|_N = 0$, called a *linearised* solution. Furthermore, (2.4) can be used to define more general solutions in the case that $n + m < N$, called *linearised least-squares* solutions for the rational interpolation problem [8]. The notion of a least-squares problem is a general one, in which the objective is to minimise a sum of squares.

Now we will see why the specific bilinear form was chosen. Let $\hat{\mathbf{a}} = (\hat{a}_0, \ldots, \hat{a}_N)^\top$ and $\hat{\mathbf{b}} = (\hat{b}_0, \ldots, \hat{b}_N)^\top$ define polynomials $\hat{p}$ and $\hat{q}$ as follows:

$$\hat{p}(x) = \sum_{j=0}^{N}{}'' \hat{a}_k T_k(x), \quad \hat{q}(x) = \sum_{j=0}^{N}{}'' \hat{b}_k T_k(x). \tag{2.5}$$

Then $\|\hat{p}\|_N = \|\hat{\mathbf{a}}\|_2$, $\|\hat{q}\|_N = \|\hat{\mathbf{b}}\|_2$ and so any linear problem in the values of $\hat{p}$ and $\hat{q}$ can be restated as a linear problem in $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$. The transformation from coefficient space to value space can be stated as $\hat{p}(\mathbf{x}) = CI''\hat{\mathbf{a}}$, and $\hat{q}(\mathbf{x}) = CI''\hat{\mathbf{b}}$ where $C = (T_j(x_i))_{i,j=0}^N$, and I'' is the identity matrix with the top-left and bottom-right entries halved. By the orthogonality relation (2.3), $\frac{2}{N}C^\top I''CI'' = I$ and so if $\hat{p}(\mathbf{x}) = (f \cdot \hat{q})(\mathbf{x})$ then we have

$$\hat{\mathbf{a}} = \frac{2}{N}C^\top I''CI''\hat{\mathbf{a}} = \frac{2}{N}C^\top I''FCI''\hat{\mathbf{b}}, \tag{2.6}$$

where F is diagonal matrix with diagonal entries $f(\mathbf{x})$.

Now let $\mathbf{a}$ and $\mathbf{b}$ be vectors containing the first $m+1$ and $n+1$ rows of $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ respectively, and let $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{b}}$ be vectors containing the last $N-m$ and $N-n$ rows of $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ respectively. Let $q \in \Pi_n$ be the truncation of $q \in \Pi_N$ (i.e. set $\tilde{\mathbf{b}} = 0$) and suppose that p and q solve the linearised least squares problem (2.4). Then since by definition of $\| \cdot \|_N$ we have

$$\|p - fq\|_N = \|p - \hat{p}\|_N, \tag{2.7}$$

where $\hat{p} \in \Pi_N$ interpolates $f \cdot q$, minimality implies that p must be the truncation of $\hat{p}$ (i.e. $\tilde{\mathbf{a}} = 0$ for p). Therefore, we have $\|p - fq\|_N = \|\tilde{\mathbf{a}}\|_2$.

Now define $\hat{Z} = \frac{2}{N}C^\top I''FCI''$, $\tilde{Z}$ to be the matrix consisting of the first $n+1$ columns and last $N-m$ rows of $\hat{Z}$, and Z to be the matrix consisting of the first $n+1$ columns and first $m+1$ rows of $\hat{Z}$. Then we have that

$$\hat{\mathbf{a}} = \hat{Z}\hat{\mathbf{b}}, \quad \mathbf{a} = Z\mathbf{b}, \quad \tilde{\mathbf{a}} = \tilde{Z}\mathbf{b}. \tag{2.8}$$

Minimality of $\|\tilde{\mathbf{a}}\|_2$ over all $q \in \Pi_n$ implies that $\mathbf{b}$ must be made to minimise $\|\tilde{Z}\mathbf{b}\|_2$. Therefore, in order for $\mathbf{a}$ and $\mathbf{b}$ to be a solution for (2.4) it suffices to take $\mathbf{b}$ to minimise $\|\tilde{Z}\mathbf{b}\|_2$, then set $\mathbf{a} = Z\mathbf{b}$. The residual $\|p - fq\|_N$ is exactly $\|\tilde{Z}\mathbf{b}\|_2$.

If $n + m = N$, then $\tilde{Z}$ is an $n \times (n+1)$ matrix, so has a non-zero kernel. Hence there exists a vector $\mathbf{b}$ with $\|\tilde{Z}\mathbf{b}\|_2 = 0$, and the associated polynomials p and q exactly solve the linearised rational interpolation problem.

In the general case where $n+m \leq N$, we can use the Singular Value Decomposition (SVD [20]) of $\tilde{Z}$ to find the minimal singular value $\sigma_{\min}$ of $\tilde{Z}$ and associated singular vector $\mathbf{b}$ such that $\sigma_{\min} = \|\tilde{Z}\mathbf{b}\|_2$, and indeed this is how `ratinterp` solves the linearised least squares problem (2.4). For a matrix $A \in \mathbb{R}^{m \times n}$, its singular value decomposition is

$$A = U\Sigma V^\top, \tag{2.9}$$

where U is an $m \times m$ orthogonal matrix whose columns are eigenvectors of $AA^\top$, V is an $n \times n$ orthogonal matrix whose columns are eigenvectors of $A^\top A$, and Σ is an

$m \times n$ diagonal matrix, whose diagonal entries are the square roots of the eigenvalues of $A^\top A$ or $AA^\top$, whichever is the smaller set. These diagonal entries are called the *singular values* of A, and we take the decomposition so that they are ordered from left to right in Σ, greatest ($\sigma_{\max}$) to smallest ($\sigma_{\min}$). By expanding x in the ordered orthonormal basis $\{v_1, \ldots, v_n\}$ consisting of the columns of V, we see that $\|Ax\|_2$ achieves its maximum $\sigma_{\max}$ at $x = v_1$ and its minimum $\sigma_{\min}$ at $x = v_n$. MATLAB has the command `svd` for its computation.

Noting that $T_j(x_i) = \cos(\frac{ij\pi}{N})$, multiplication by CI'' and $\frac{2}{N}C^\top I''$ can be interpreted as taking the discrete cosine transform (DCT) and inverse discrete cosine transform (iDCT) respectively. The Fast Fourier Transform (FFT) is used by `ratinterp` to efficiently compute the DCT and iDCT when computing $\tilde{Z}$.

This is not the end of the story because we have not discussed uniqueness. If the minimal singular value $\sigma_{\min}$ is multiple, of multiplicity d, say, then there exist d linearly independent vectors $\mathbf{b}$ such that $\|\tilde{Z}\mathbf{b}\|_2 = \sigma_{\min}$, corresponding to d different polynomials q. This case corresponds to p and q potentially sharing d roots. In exact arithmetic these will cancel, but on a computer they almost certainly will not, producing a zero-pole pair known as a *Froissart doublet*.

By taking an appropriate linear combination, there exists a $\mathbf{b}$ with the last $d-1$ entries zero, corresponding to a $q \in \Pi_{n-(d-1)}$. From a robustness point of view, it is better to take this $\mathbf{b}$ because it will reduce the number of poles in r, reducing the chance of a Froissart doublet. It would be possible to use linear algebra to find this $\mathbf{b}$, but it is simpler to reduce n by $d-1$ (keeping N and m the same) and start the procedure again. This process can be repeated, giving the algorithm a unique solution.

This idea can be taken further to improve robustness when implemented on a computer. The user sets a tolerance parameter *tol*, and if there are d singular values within *tol* of $\sigma_{\min}$, then n is decreased by $d-1$. The resulting p and q have their degrees reduced even further by discarding the higher coefficients smaller than *tol* in absolute value.

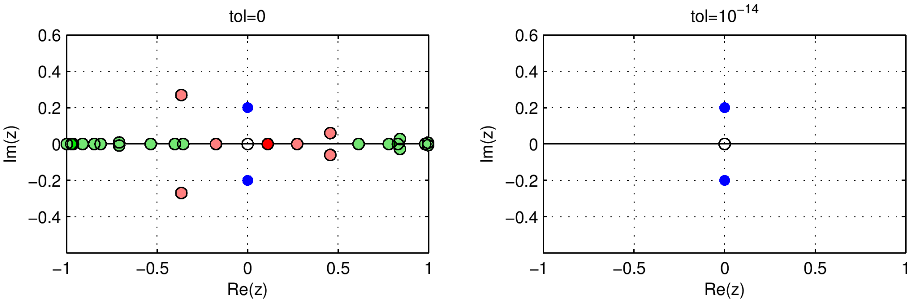

FIG. 2.1. *`ratinterp` is used to compute two type* $(30,30)$ *rational approximants of* $f(x) = x/(1+25x^2)$ *on* $[-1,1]$ *with tol* $= 0$ *(left) and tol* $= 10^{-14}$ *(right), returning* $(30,30)$ *and type* $(1,2)$ *approximants respectively. The zeros of* r *are plotted as empty black circles and the poles are plotted as coloured dots by the colour scale in the table below. We emphasise that this example is typical and not exceptional.*

|Residue|	$(0, 10^{-14})$	$[10^{-14}, 10^{-12})$	$[10^{-12}, 10^{-9})$	$[10^{-9}, 10^{-6})$	$[10^{-6}, 10^{-3})$	$[10^{-3}, \infty)$
Colour	red	pink	light green	green	light blue	blue

The tolerance parameter is by default 10^{-14}, which in practice works well when approximating functions accurate to machine precision (around 10^{-16}), but if you know that f contains noise of magnitude ϵ, then $tol = 100\epsilon$ is a good suggestion for the tolerance. If the tolerance is set at a value smaller than this, then `ratinterp` will futilely use this noise for its approximation. We will use $tol = 10^{-12}$ since u is computed numerically as the solution to an ODE and will therefore not be accurate to machine precision: there will be noise in u with magnitude between 10^{-13} and 10^{-12}.

In Figure 2.1, the plotted results follow the style of [9] where the poles are coloured dots with their colour decided by their residue according to the table below. Notice the appearance of Froissart doublets when $tol = 0$, where coloured dots (poles) have an empty black circle (zero) in (almost) the same place.

The `ratinterp` algorithm delivers a rational function of type (μ, ν) where $0 \le \mu \le m$, $0 \le \nu \le n$. We take the roots of q to approximate the singularities of f. `ratinterp` can also calculate the residues of the poles, which is implemented using MATLAB's `residue` command [5].

Chebfun's ratinterp is not restricted to using the Chebyshev points on $[-1, 1]$ for the nodes. The method explained above generalises to arbitrary points in the complex plane, and the user can specify whatever nodes they want `ratinterp` to use. The general approach for the linearised interpolation problem is discussed in [16]. The case where the nodes are the Nth roots of unity is particularly elegant because the polynomials satisfying a discrete orthogonality relation are simply the monomials (powers of roots of unity make up the Fourier basis). The linearised least squares problem and robustness steps for the case of roots of unity can be found in [9].

As well as the rational approximant of u, we can compute the *Chebfun ellipse* associated with the chebfun u. The Chebfun ellipse is the Bernstein ellipse with ρ parameter such that $\rho^N = \epsilon$ where ϵ is machine precision (approximately 10^{-16}). This ellipse is a good estimate for the Bernstein ellipse associated with f, discussed in Section 1 [19, Ch.8]. This estimate is useful because at least one singularity of f lies on the edge of the Bernstein ellipse and poles deep inside the ellipse must be spurious.

Recall that we denote the degree of the chebfun solution to the ODE problem u by N, and this integer is computed automatically by Chebfun. Using few than $N+1$ nodes for our rational approximation would not be utilising all the information we have about u, and using more than $N+1$ nodes would be approximating data that has been interpolated from $N+1$ nodes anyway. Therefore we perform our rational approximation on $N+1$ nodes if we are approximating a chebfun of degree N.

In summary, we perform the following list of operations as our proposed method: Solve the ODE problem using chebops or `ode113` on $[0, T]$, producing a chebfun u of degree N; compute a rational interpolant $r = p/q$ from u on $N+1$ Chebyshev points in $[0, T]$; compute the locations of the poles by finding the roots of q and compute their residues using `residue`; and compute the Chebfun ellipse associated for u to estimate the Bernstein ellipse for f.

3. Preliminary Experiments.

A key question to ask when computing rational approximants is, what are good choices for m and n? This turns out to be rather troublesome. The following quote is from *Padé Approximants* by G.A. Baker and P.R. Graves-Morris [1].

> "In practice, whether one expects them or not, *defects* occur for all but the simplest functions."

Here, defect is synonymous with the spurious poles mentioned in the previous section. Baker and Graves-Morris are referring to Padé approximants rather than the rational interpolation and least-squares approximants considered in this paper, but the principle still applies (see Figure 2.1). For the case of robust rational interpolation on the unit circle, this issue is discussed in [9, sections 4 and 6]. For this project, the author performed some preliminary experiments with robust rational least-squares approximation on an interval.

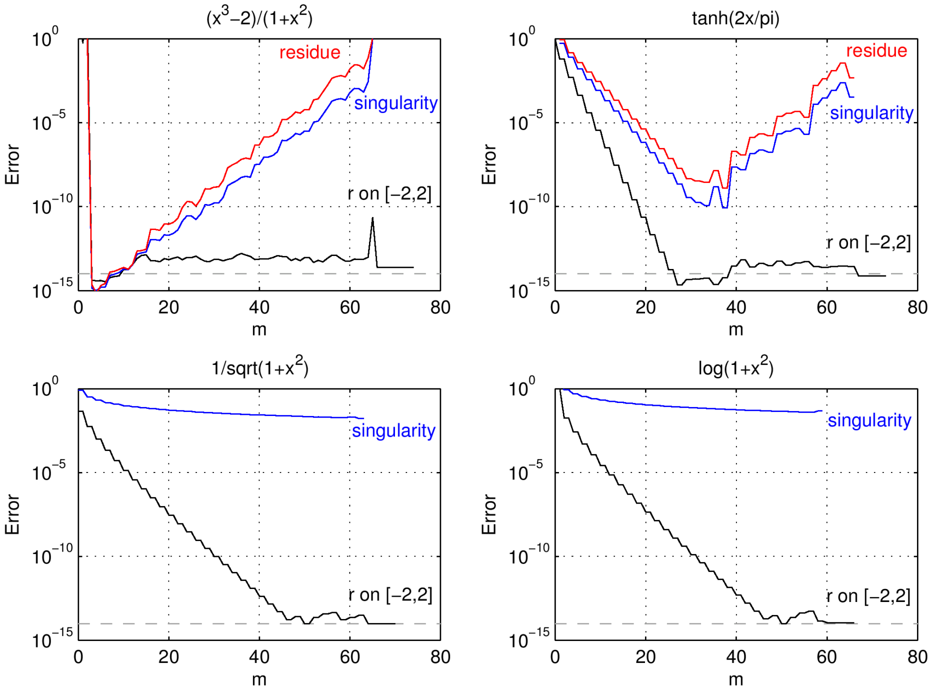

FIG. 3.1. *Type* $(m, 2)$ *rational approximants computed with* `ratinterp` *for various chebfuns on* $[-2, 2]$*, for* m *in the range* $[0, N-2]$ *(where* N *is the degree of the chebfun). Each function has a singularity at* $\pm i$*, the top two having simple poles and the bottom two having branch points. We plot the error in:* r *on* $[-2, 2]$*, computed location of singularities, and (in the case of poles) the computed residue at the poles.*

In the preliminary experiments, the author found that the behaviour of `ratinterp` depended heavily on N, the degree of the chebfun being approximated. N is a good indication of how complicated f is on $[0, T]$ and therefore indicative of how complicated good rational approximations of f will be. The experiments agreed with this line of argument and the author recommends the following for m and n: take $m = \frac{1}{2}N$ and take n to be greater than the number of singularities you expect r to have. This will not always give the best results; it depends on the number of singularities and their types. Branch points usually require a higher value of m to be approximated well and simple poles require a lower value. This is demonstrated in Figure 3.1: the bottom two plots have better accuracy in both r and the location of the singularities to the right of $m = 40$, whereas the top two plots have better accuracy to the left of $m = 40$.

In experiments, when m is taken in the range of $\frac{1}{3}N$ and $\frac{2}{3}N$, Froissart doublets and spurious poles are avoided. In some cases meaningful poles are removed, but this appears to be the only real issue encountered. A situation where this occurs is when m is set too high, robust `ratinterp` removes *all* of the poles from the approximant and returns a polynomial. An entire function is not helpful if we want to find singularities!

We see this in all four of the plots in Figure 3.1, because the red and blue lines, which rely on the existence of singularities, do not extend as far as the black lines, which only rely on the existence of a rational approximant.

Ideally, we would state and prove theorems that explain results such as those illustrated in Figure 3.1, but at present, little is understood about the robust least-squares algorithm. Can the relationship between the SVD of the linearised problem and the singularities of the resulting rational approximant be made precise? A more specific question is how high m must be for the robust algorithm to remove all of the poles from the approximant. The rest of this article is an investigation into what `ratinterp` is capable of for the computation of singularities of differential equations, as a first step towards the required understanding.

4. ODEs under Investigation. This is the main section of the paper. Here we discuss some specific ODEs with results showing computations of complex singularities for their solutions. The first two examples are ones with straightforward explicit solutions, and the last three are some examples from applied mathematics that are interesting in their own right.

4.1. Simple Poles. The following nonlinear ODE (with appropriate boundary conditions) has solution $\tanh(\alpha + \beta x)$ for arbitrary constants α, β:

$$\frac{d^2 f}{dx^2} + 2\beta f \frac{df}{dx} = 0. \tag{4.1}$$

We used Chebfun to solve (4.1) with $\alpha = 0, \beta = 1$ on the domain $[-5, 5]$ and boundary conditions $f(\pm 5) = \tanh(\pm 5)$. The numerically computed solution u is a chebfun of degree $N = 105$, so following the strategy described in the previous two sections we used `ratinterp` to generate a type $(\lfloor \frac{1}{2} N \rfloor, 10) = (52, 10)$ rational approximant r on the 106 Chebyshev points in $[-5, 5]$. `ratinterp` returned a type $(31, 2)$ approximant which appears in Figures 4.1 and 4.2.

Table 4.1
Error Data for Rational Approximation of $\tanh(z)$

Quantity	Max Error	Mean Error
u on $[-5,5]$	1.0×10^{-12}	2.0×10^{-13}
r on $[-5,5]$	7.1×10^{-12}	1.4×10^{-12}
poles(r)	4.0×10^{-8}	4.0×10^{-8}
residues(r)	2.4×10^{-7}	2.4×10^{-7}

When $f(x) = \tanh(x)$ is extended into the complex plane it has simple poles at $(2j+1)\frac{\pi}{2} i$ for each $j \in \mathbb{Z}$, each with residue 1. In Figure 4.1 and Table 4.1 we can see that just using our heuristic from Section 3 for the values of m and n, we get fairly accurate results for the locations of the poles and residues. If one decreases m, r can approximate the zeros at $\pm\pi i$ and even the poles further out at $\pm\frac{3\pi}{2} i$, but accuracy of the approximation on $[-5, 5]$ is sacrificed.

In Figure 4.2, bottom right, we can see that the rational approximation is only accurate in a small oval around the interval, a barrier beyond which r blows up polynomially (of very high degree). Nonetheless, if we are only interested in the nearest singularities to the real line this small window of meaningful continuation is satisfactory. The same cannot be said for the chebfun u, which blows up quickly outside of the Bernstein ellipse (bottom left).

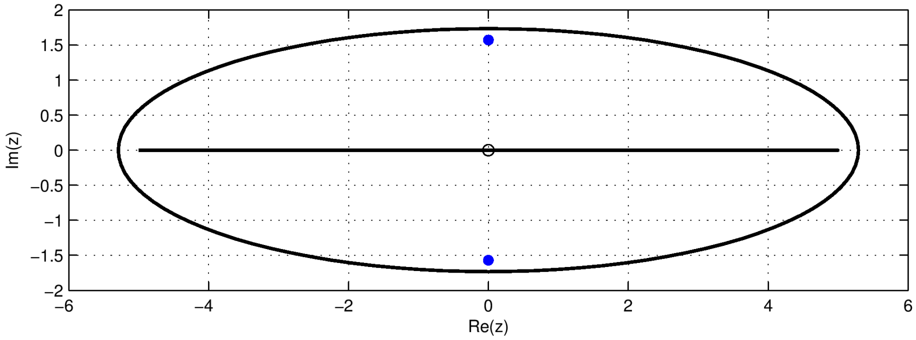

FIG. 4.1. *Schematic of the numerical analytic continuation of the solution to* (4.1) *on* $[-5,5]$*, showing simple poles at* $\pm\frac{\pi}{2}i$ *(blue dots) and a zero at* 0 *(circle). The Chebfun ellipse, computed by Chebfun, is slightly larger than the actual Bernstein ellipse associated with tanh.*

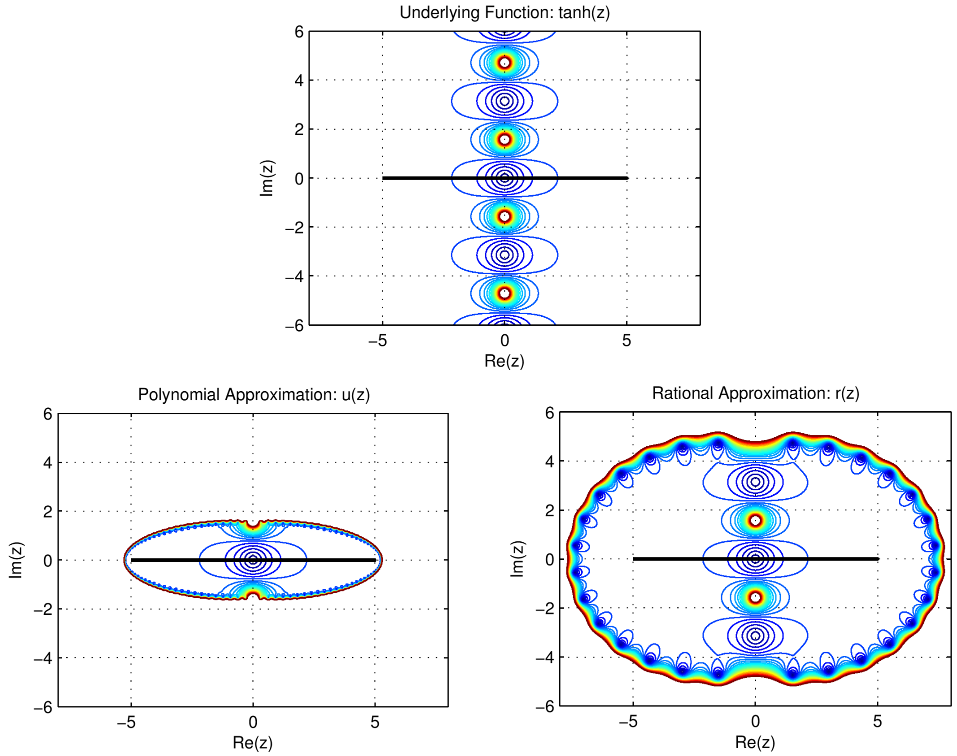

FIG. 4.2. *Contour plots of the absolute value of* tanh(z) *(top), the chebfun numerical solution to* (4.1) *u (left) and its* **ratinterp** *approximant r (right). The contours are coloured from blue to red on* $[0,5]$*. The chebfun u is useless as an approximation of* tanh(z) *outside of the Bernstein ellipse whereas the ratinterp approximant r reaches much further.*

4.2. Logarithmic Branch Points. The following example is a first order ODE with solution $f(t) = \log(1+(t-\gamma)^2)$ for an arbitrary constant γ (given an appropriate initial value):

$$\frac{df}{dt} + 2(\gamma - t)\exp(-f) = 0. \tag{4.2}$$

The exponential term in (4.2) may seem contrived, but an example from combustion theory, the time independent Frank-Kamenetskii blowup equation, takes a similar form: $d^2f/dx^2 + A\exp(f) = 0$ [17].

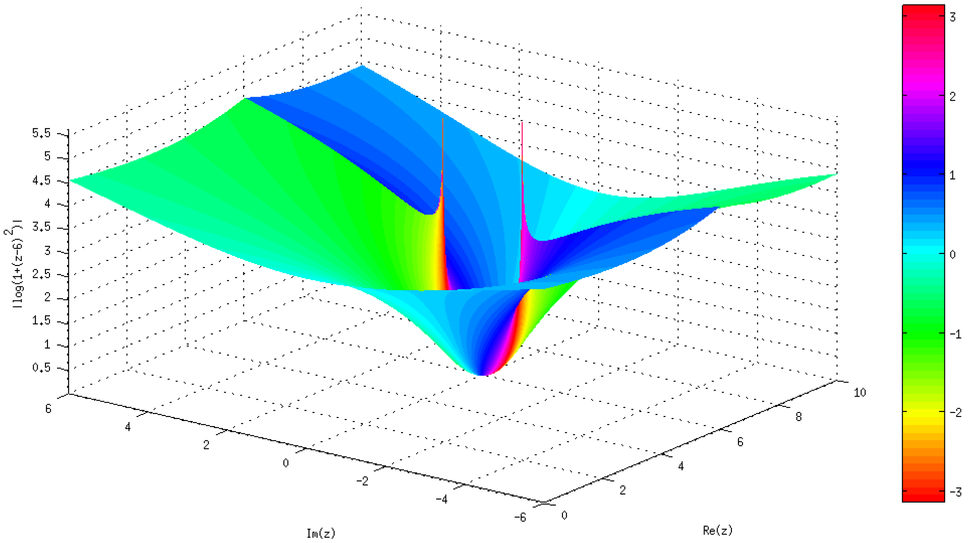

Fig. 4.3. *3D plot of the absolute value of* $f(z) = \log(1 + (z - 6)^2)$ *on* $[0, 10] \times [-6, 6] \subseteq \mathbb{C}$. *The plot is coloured by argument ranging from* $-\pi$ *to* π. *We can clearly see the two logarithmic singularities and the branch cuts behind them, across which the argument of* f *is discontinuous.*

We consider the case $\gamma = 6$, for which we have plotted the surface generated by the absolute value of the analytical solution. The surface is coloured by the argument of the complex value in order to see that along the branch cuts the solution is not continuous. We solved (4.2) using chebops on the interval $[0, 10]$ with initial data $f(0) = \log(37)$. The solution u has degree $N = 137$ so as in the previous example, we used this information and computed a type $(\lfloor \frac{1}{2}N \rfloor, 10) = (68, 10)$ rational approximant r on 138 Chebyshev points in $[0, 10]$. `ratinterp` returned a type $(57, 4)$ rational function with poles at $t = 6.0017 \pm 1.0429i, 6.0109 \pm 1.2515i$.

This is typical of rational approximants; in practice branch cuts are approximated by lines of poles in the rational function as in Figure 4.4 and the actual location of the branch point itself is not very accurate (see Table 4.2).

Table 4.2

Error data for rational approximation of the solution to (4.2). The error in the location of the branch points (poles(r)) is much worse than that for the poles in the previous example.

Quantity	Max Error	Mean Error
u on $[0, 10]$	3.7×10^{-12}	1.1×10^{-12}
r on $[0, 10]$	1.9×10^{-9}	2.0×10^{-10}
poles(r)	4.3×10^{-2}	4.3×10^{-2}

What relevance do the residues of the poles of r have for f at branch point singularities? For this example the poles of r have residues of absolute value 1.3×10^{14}

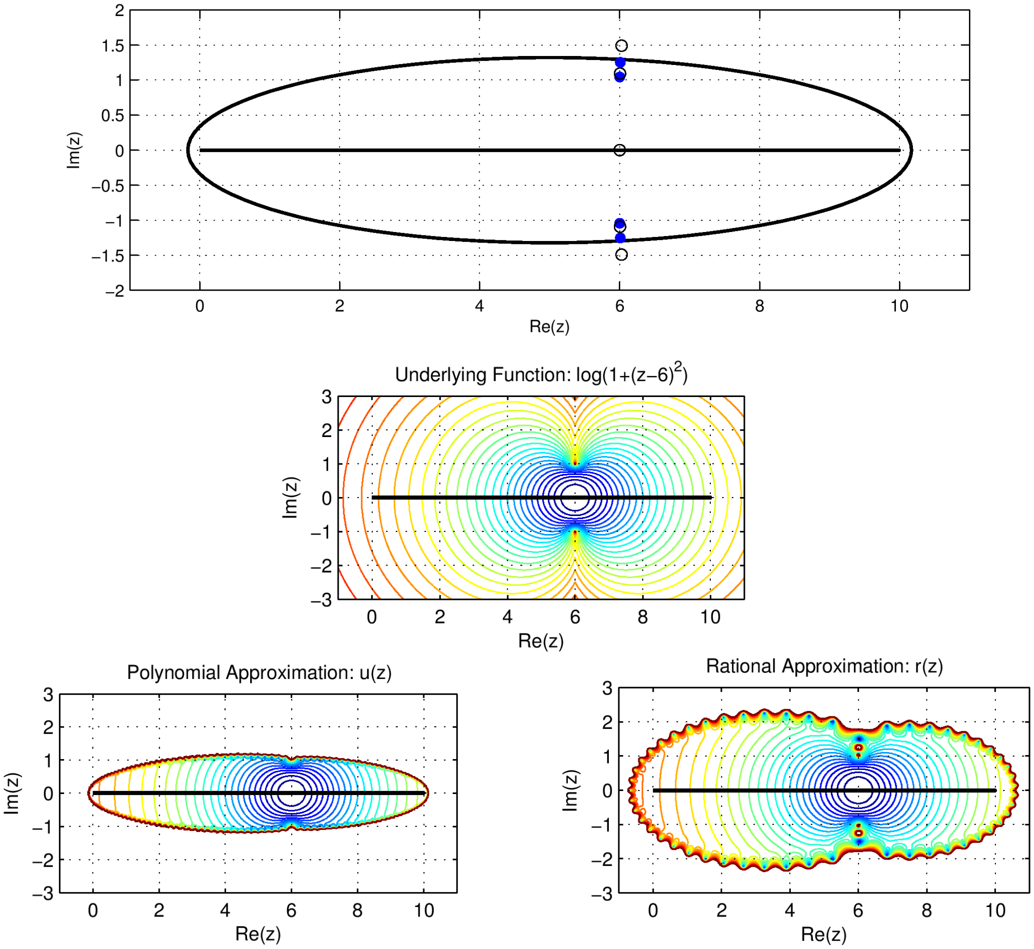

FIG. 4.4. *Top: Schematic of the poles (blue dots) and roots (circles) of the* `ratinterp` *approximant of the numerical solution to* (4.2) *on* $[0, 10]$. *Middle: Contour plot of the absolute value of the underlying solution* $log(1 + (z − 6)^2)$. *Left: polynomial approximation of the solution to* (4.2)*, which blows up outside of its Bernstein ellipse. Right:* `ratinterp` *approximant* r *which approximates the solution further into the complex plane. The contours are coloured blue to red on* $[0, 5]$.

and 1.4×10^{14}, which are surprisingly large. This could be a feature of our method for calculating the residues, MATLAB's `residue` command [5]. The documentation warns that the method is unstable in some circumstances. It could also be a property of the rational interpolation and least squares approximation for branch point singularities.

4.3. Lorenz Attractor. Viswanath and Şahutoğlu 2010 [21] is a fascinating paper, which puts forward the point of view that although the Lorenz attractor is a well known example in applied mathematics, relatively little is known about the mathematical analysis of its solutions. The authors present an analytic treatment where they consider time as a complex variable and show that a certain class of solutions respresented by so-called Ψ-series are singular, with complex logarithmic singularities close to the real line.

The system was originally studied by Lorenz, who derived it from the simplified equations of convection rolls arising models of the atmosphere:

$$\begin{aligned}\frac{dx}{dt} &= 10(y-x),\\ \frac{dy}{dt} &= 28x - y - xz, \\ \frac{dz}{dt} &= -8z/3 + xy.\end{aligned} \tag{4.3}$$

It is an early example of a chaotic dynamical system, in which small changes in initial data can produce wildly different results in the solution. Lorenz used this in his 1963 paper to argue that accurate long-range weather prediction may be impossible [12].

Definition 4.1. *A logarithmic Ψ-series centered at t_0 is a series of the form*

$$\sum_{j=-J}^{\infty} P_j(\eta)(t-t_0)^j, \quad \eta = \log(b(t-t_0)), \tag{4.4}$$

where J is an integer, P_j is a polynomial and b is a complex number with $|b| = 1$.

We can take $b = \pm i$ without loss of generality. As $t \to t_0$ a Ψ-series behaves asymptotically like a pole of Jth order, as the polynomials in η are "overpowered" by the $(t-t_0)^{-J}$ term (Hille calls them *pseudopoles* [10]). We expect pseudopoles to be approximated well by rational functions because of this asymptotic approximation.

Ψ-series solutions of the Lorenz attractor described in [21] are of the form

$$\begin{aligned} x(t) &= \frac{P_{-1}(\eta)}{t-t_0} + P_0(\eta) + P_1(\eta)(t-t_0) + P_2(t-t_0)^2 + \dots, \\ y(t) &= \frac{Q_{-2}(\eta)}{(t-t_0)^2} + \frac{Q_{-1}(\eta)}{t-t_0} + Q_0(\eta) + Q_1(\eta)(t-t_0) + Q_2(t-t_0)^2 + \dots, \\ z(t) &= \frac{R_{-2}(\eta)}{(t-t_0)^2} + \frac{R_{-1}(\eta)}{t-t_0} + R_0(\eta) + R_1(\eta)(t-t_0) + R_2(t-t_0)^2 + \dots, \end{aligned} \tag{4.5}$$

in the disc $|t-t_0| \leq r$ for some $r > 0$ but with the singular point $t = t_0$ and a branch cut deleted from the disc. It should be borne in mind that it has not yet been proved that all singular solutions of the Lorenz attractor take this form.

The chebop methods in Chebfun struggle to solve the Lorenz system without a good initial starting point as it is nonlinear and highly oscillatory, so we use Chebfun's overload of `ode113`. The same applies to examples in subsections 4.4 and 4.5. For this experiment we set our initial conditions to be

$$x(0) = -14, \quad y(0) = -15, \quad z(0) = 20, \tag{4.6}$$

which gives the beautiful butterfly shaped trajectory in 3-dimensional space shown in the second plot in Figure 4.5. We solved the Lorenz system on $[0,5]$ with the above initial conditions, and `ode113` returned three chebfuns u_x, u_y and u_z with degrees $N_x = 462$, $N_y = 509$ and $N_z = 498$. We used our general strategy and computed rational approximants of types $(231,20)$, $(255,20)$ and $(249,20)$ and `ratinterp` returned rational functions of types $(173,10)$, $(227,10)$ and $(221,10)$ respectively.

We can see from (4.5) that x, y and z have precisely the same singularities, t_0, so we should expect our rational approximants r_x, r_y and r_z to have the same singularities (assuming the initial conditions give a solution of this Ψ-series form). Table 4.3 lists the locations of the 10 computed singularities for each component and we do

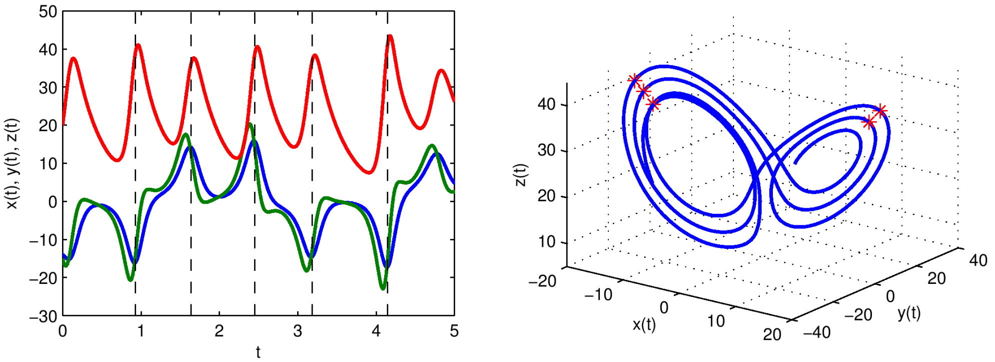

FIG. 4.5. *An* `ode113` *solution to the Lorenz system solved on the interval* $[0, 5]$ *with initial conditions* (4.6)*. For each singular point computed by* `ratinterp`*,* $t \in \mathbb{C}$ *with* $Re(t) = t_0$*, we have plotted a dotted line on at time* t_0 *and a star at point* $(x(t_0), y(t_0), z(t_0))$*.*

see some agreement with this hypothesis, but the agreement is only to 1 or 2 decimal places. Just as in the elementary branch point example, the singularities are not approximated to a very high degree of accuracy, even if in this case we are approximating pseudopoles.

TABLE 4.3

Poles of the `ratinterp` *approximation to our numerical solution to the Lorenz attractor. The* 3 *components should have the same poles, but we only agree to* 1 *or* 2 *decimal places.*

r_x	r_y	r_z	Max Separation
$0.9301 \pm 0.1642i$	$0.9295 \pm 0.1564i$	$0.9297 \pm 0.1565i$	7.6×10^{-2}
$1.6394 \pm 0.1873i$	$1.6384 \pm 0.1773i$	$1.6382 \pm 0.1796i$	1.0×10^{-2}
$2.4530 \pm 0.1669i$	$2.4519 \pm 0.1572i$	$2.4520 \pm 0.1570i$	9.9×10^{-3}
$3.1834 \pm 0.1816i$	$3.1836 \pm 0.1717i$	$3.1818 \pm 0.1715i$	1.0×10^{-2}
$4.1492 \pm 0.1520i$	$4.1484 \pm 0.1446i$	$4.1484 \pm 0.1445i$	7.5×10^{-3}

In Figure 4.6 we have plotted the locations of the singularities and zeros of the `ratinterp` approximations to the chebfun solutions to the Lorenz system. The singularity structure is similar to that of the *periodic* solution computed by Viswanath and Şahutoğlu in Figure 1.1 of [21].

Figure 4.7 shows an important feature of robust rational interpolation and least squares on an interval. The clustering of the contours implies the existence of complex singularities in the underlying solution with real parts about 0.2 and 4.8, which have been missed by the rational approximation. In these experiments, this has been normal behaviour: singularities with real part close to the edges of the interval do not often appear in the rational approximant, but are lost behind the oval shaped barrier of high-degree-polynomial increase.

The residues of the poles are huge, of magnitude around 10^{100}, even bigger than those of the previous example. Some investigation should be made into whether the calculated figures for the residues are a facet of our rational approximation of branch points or due to the noted instability of the `residue` command.

It has been proved that for solutions on the Lorenz attractor, any singularity $t_0 \in \mathbb{C}$ must have $|\text{Im}(t_0)| > 0.037$ [21]. This is important, because if in general a

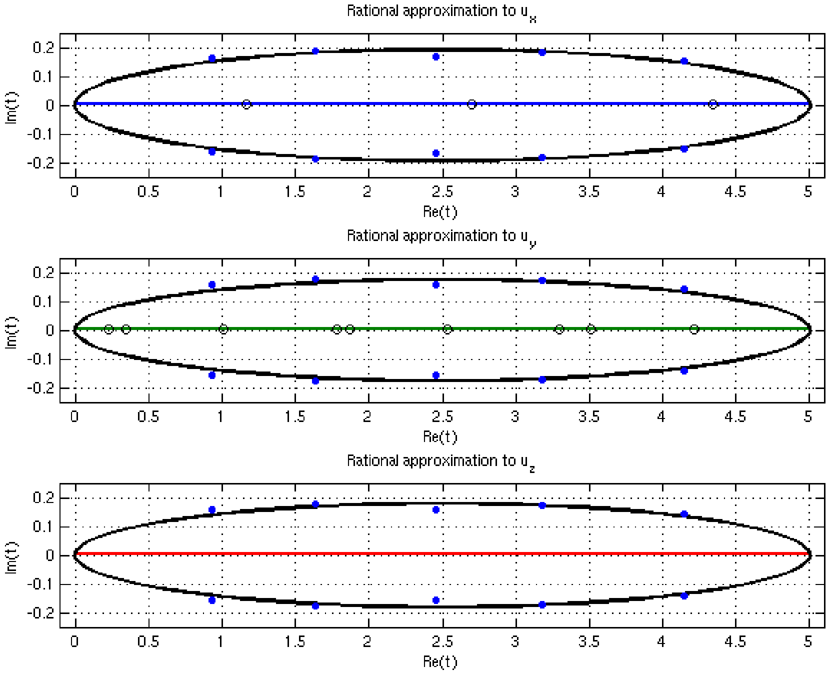

FIG. 4.6. *A schematic of the rational approximation of u_x (blue), u_y (green) and u_z (red) of the* **ode113** *solution to* (4.3) *on* $[0, 5]$. *The singularity structure is similar to that of the* periodic *solution computed by Viswanath and Şahutoğlu in Figure 1.1 of [21].*

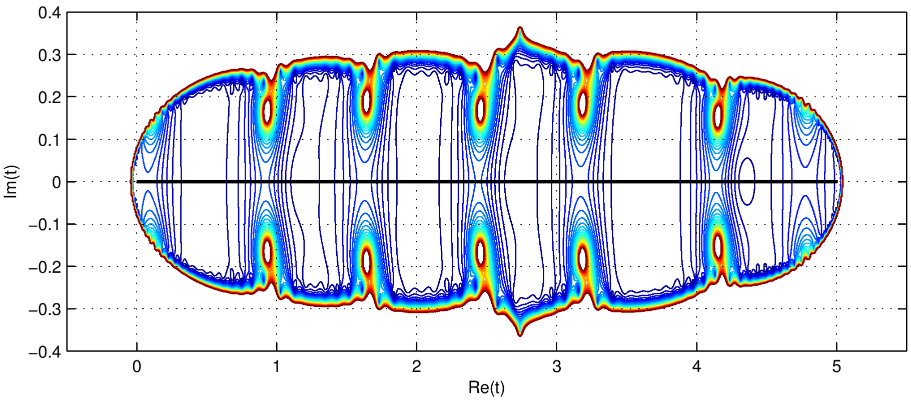

FIG. 4.7. *A contour plot of the absolute value of the* **ratinterp** *approximant $r_x(t)$ of the numerical solution to the Lorenz attractor* (4.3). *The contours are coloured blue to red on* $[0, 80]$

solution to a differential equation satisfies $|\mathrm{Im}(t_0)| > \tau$, the transformation

$$\zeta = \frac{\exp(\pi t/2\tau) - 1}{\exp(\pi t/2\tau) + 1} \tag{4.7}$$

maps the strip where the solution is analytic to the unit disc. The solution must therefore have a globally convergent expansion in powers of ζ. Numerical experiments such as the ones we have done here, and those done by Viswanath and Şahutoğlu for [21] can help inform the analysis in this fashion.

4.4. Lotka-Volterra Predator-Prey Model. Our next example is from mathematical ecology. A simple population model for interacting species is the Lotka-Volterra predator-prey model

$$\begin{aligned} \frac{dx}{dt} &= \alpha x - \beta x y, \\ \frac{dy}{dt} &= -\gamma y + \delta x y, \end{aligned} \tag{4.8}$$

where $\alpha, \beta, \gamma, \delta$ are positive constants with $\gamma < \beta$. In the model, y represents the population of a predator and x represents the population of its prey as they vary over time [11]. The coefficient α encodes the rate of reproduction in prey irrespective of predators, β describes the rate at which predators kill the prey, γ represents the starvation of predators in the absence of prey, and δ expresses the rate at which predators reproduce given enough prey to feed on.

If we restrict ourselves to the physically meaningful case where $x, y > 0$, the system can be integrated exactly. Dividing the two equations and using the chain rule gives the separable equation,

$$\frac{\delta x - \gamma}{x} \frac{dx}{dy} = \frac{\alpha - \beta y}{y}, \tag{4.9}$$

and we obtain an implicit solution by integrating with respect to y:

$$H = (\alpha \log y - \beta y) + (\gamma \log x - \delta x). \tag{4.10}$$

H can be considered as the first integral, Hamiltonian or energy of the system and is a constant that depends only on the initial conditions. We plot this trajectory (x, y) in Figure 4.9 for different values of H. The parameters and initial conditions we use here are $\alpha = \beta = 0.5, \gamma = \delta = 1, x(0) = 2, y(0) = 3$.

There are still many open problems for the analysis of the Lorenz attractor, which is a three-dimensional autonomous system [21]. In contrast, the analysis of the Lotka-Volterra system, a plane autonomous system, is further developed. Hille proved that the plane quadratic system

$$\frac{dx}{dt} = x(a_0 + a_1 x + a_2 y), \tag{4.11}$$

$$\frac{dy}{dt} = y(b_0 + b_1 x + b_2 y), \tag{4.12}$$

has a logarithmic Ψ-series singularity of order 1 (i.e. $J = 1$ in the expression (4.4)) if $(a_1 - b_1)(a_2 - b_2)/(a_1 b_2 - a_2 b_1)$ is a positive integer [10, Sec. 12.5,12.6]. For the Lotka-Volterra system, we have

$$\frac{(a_1 - b_1)(a_2 - b_2)}{a_1 b_2 - a_2 b_1} = \frac{(0 - \delta)(-\beta - 0)}{0 \cdot 0 - (-\beta)(\delta)} = 1. \tag{4.13}$$

Hence the Lotka-Volterra system always has a simple logarithmic Ψ-series singularity in the complex plane.

As in the previous subsection, we used `ode113` to solve the Lotka-Volterra system on $[0, 45]$, which returned two chebfuns u and v with degrees $N_x = 743$ and $N_y = 737$.

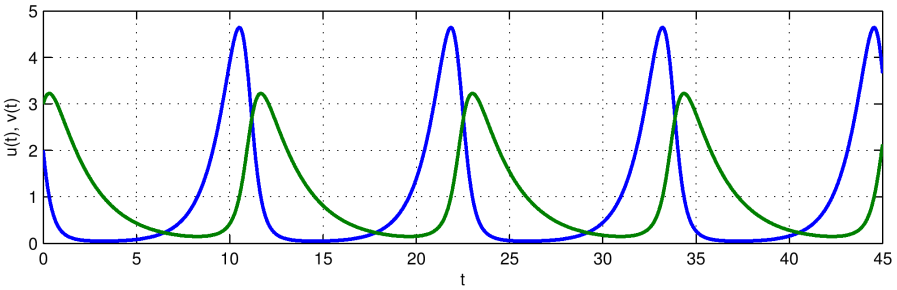

FIG. 4.8. *`ode113` solutions u and v to* (4.8) *with* $\alpha = \beta = 0.5, \gamma = \delta = 1, u(0) = 2, v(0) = 3$ *in blue and green respectively.*

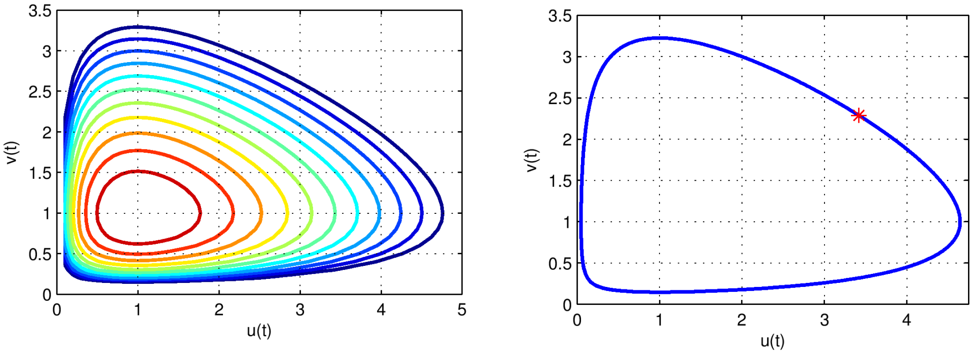

FIG. 4.9. *Left: Implicit solutions to* (4.8) *with* $\alpha = \beta = 0.5, \gamma = \delta = 1$, *for various values of H. Right: A numerically computed trajectory for the same problem with initial conditions* $u(0) = 2, v(0) = 3$. *The point on the trajectory corresponding to u and v at the time that is the real part of the (periodic) complex singularities has been marked* $((u, v) = (3.44, 2.27))$.

TABLE 4.4

Singularities of the Lotka-Volterra equations for $\alpha = \beta = 0.5, \gamma = \delta = 1$. *They should, by periodicity, have the same imaginary part, but we can see that the accuracy in this respect is quite low.*

r_x	r_y	Separation
$11.0204 \pm 0.9166i$	$11.0244 \pm 0.9170i$	4.0×10^{-3}
$22.3637 \pm 0.9169i$	$22.3692 \pm 0.9171i$	5.5×10^{-3}
$33.7086 \pm 0.9168i$	$33.7128 \pm 0.9168i$	4.2×10^{-3}

For the benefit of the reader, these two chebfuns have been plotted in Figure 4.8. Following the heuristics discussed in Section 3, we used `ratinterp` with target types $(371, 20)$ and $(366, 20)$ on 372 and 366 Chebyshev points in $[0, 45]$, which returned rational approximants of types $(297, 6)$ and $(287, 6)$.

In Figure 4.10, we see that the periodicity of the solution is expressed somewhat in the rational approximant, but the appoximant still blows up outside of an oval shaped barrier. What should plainly be singularities just off the real line near time 0 and time 45 are lost behind the barrier, just as in the Lorenz attractor example. To approximate these singularities, can solve the system on $[-15, 15]$ and $[30, 60]$.

Although the user can see where the barrier is by inspection, and that there may be further singularities behind it, if an automated procedure for locating singularities

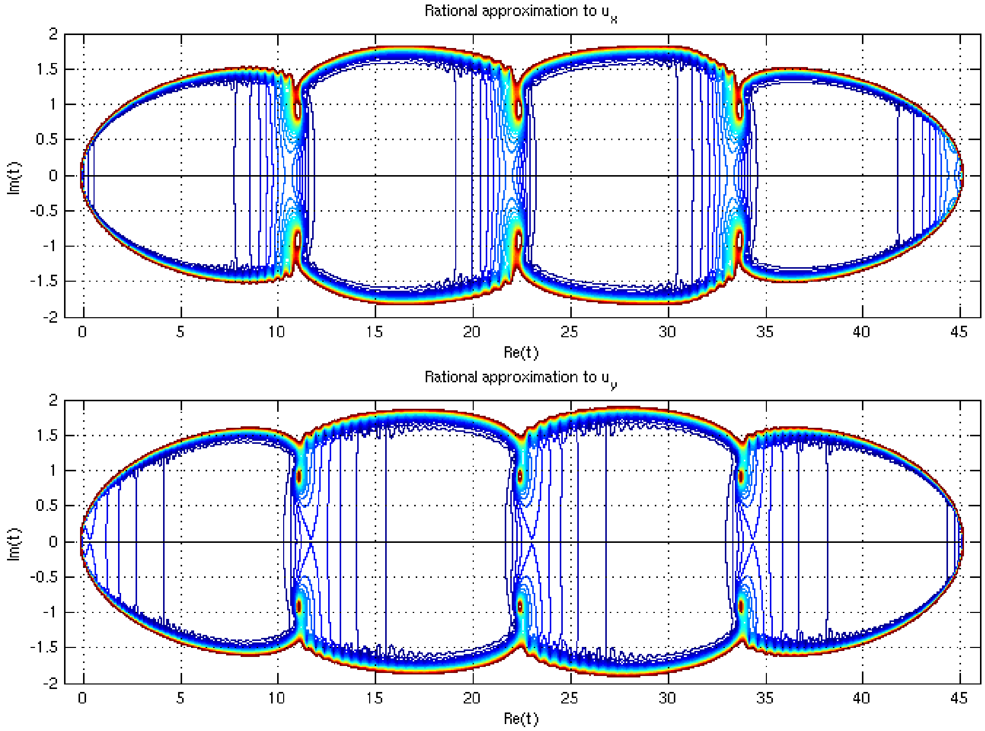

FIG. 4.10. *We solved the Lotka-Volterra system on the interval* $[0, 45]$ *using* `ode113`*. These are contour plots of rational approximants* $r_x(t)$ *(top) and* $r_y(t)$ *(bottom) to the numerical solution, in the complex* t*-plane. We can see the periodic nature of the solutions in the complex plane, with period around* 11.3*, and the branch cut associated with each singularity, the beginnings of which protrude outwards from the real line. The contours are coloured blue to red on* $[0, 20]$

is what we desire, estimates for the size and shape of the oval barrier for the rational approximant would be necessary.

4.5. Three-Body Problem. The three-body problem is the term used to refer to the system of ODEs modelling the motion of three points of prescribed masses under mutual Newtonian gravitation in three dimensions:

$$\begin{aligned}
\frac{d^2x_1}{dt^2} &= m_2\frac{x_2 - x_1}{\|x_2 - x_1\|^3} + m_3\frac{x_3 - x_1}{\|x_3 - x_1\|^3},\\
\frac{d^2x_2}{dt^2} &= m_1\frac{x_1 - x_2}{\|x_1 - x_2\|^3} + m_3\frac{x_3 - x_2}{\|x_3 - x_2\|^3},\\
\frac{d^2x_3}{dt^2} &= m_1\frac{x_1 - x_3}{\|x_1 - x_3\|^3} + m_2\frac{x_2 - x_3}{\|x_2 - x_3\|^3}.
\end{aligned} \tag{4.14}$$

Here x_1, x_2 and x_3 are the positions of three particles with masses m_1, m_2 and m_3 respectively, in space at time $t \in \mathbb{R}$. The problem is a classic that has fascinated some of the greatest mathematicians: Newton, Euler, Lagrange, Laplace, Poincaré and even those of today.

In 2000 Chenciner and Montgomery published a paper proving the existence of a "remarkable periodic solution to the three-body problem in the case of equal masses"

[4], where the three particles travel around in a planar figure of eight shape as in Figure 4.11. This particular solution was first discovered numerically by C. Moore in 1993 [14]. Such solutions to the general n-body problem with smooth periodic trajectories have since been called *choreographies* and it is now an active area of research [13].

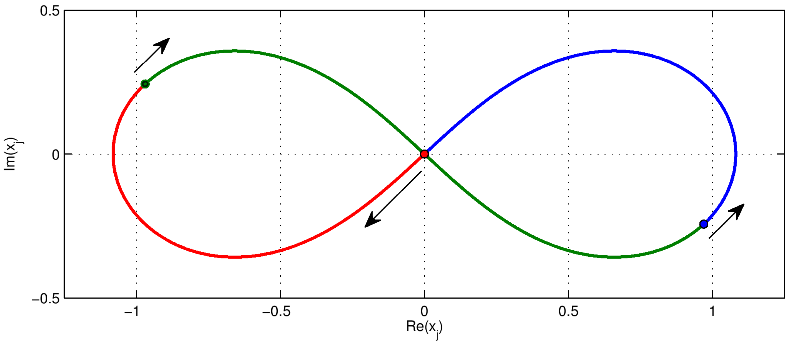

FIG. 4.11. *The paths of the three complex-valued bodies x_1, x_2 and x_3 solving* (4.14) *travel periodically in a figure of eight shape in the complex plane. We colour the bodies and their paths for the first third of their periods here in blue, green and red respectively.*

Since this special case of the three-body problem is planar, we can use complex arithmetic to represent the positions x_1, x_2 and x_3[1]. Without loss of generality we may assume the masses are all equal to 1. The initial conditions for the figure of eight are given in the paper (computed by Carles Simó) and are as follows:

$$\begin{aligned} x_1 &= -x_2 = 0.97000436 - 0.24308753i, \quad x_3 = 0, \\ \dot{x}_1 &= \dot{x}_2 = 0.466203685 + 0.43236573i, \quad \dot{x}_3 = -0.93240737 - 0.86473146i, \end{aligned} \tag{4.15}$$

The resulting choreography has period $T = 6.32591398$ (approx.), so we used `ode113` to solve the system on the interval $[0, 12.65182796]$ (two periods) which returned u_1, u_2 and u_3, chebfuns of degree 330, 330 and 335 respectively. We used `ratinterp` to compute type $(165, 20)$, $(165, 20)$, and $167, 20)$ rational approximations on 331, 331 and 336 Chebyshev points on the interval, which returned rational functions r_1, r_2 and r_3 of types $(157, 7), (157, 7)$ and $(159, 6)$.

The resulting singularity structure shown in Figure 4.12 is different from all the other examples: Because x_1, x_2 and x_3 are complex valued, the singularities do not necessarily come in conjugate pairs. Now, let us turn to the complex t-plane (as opposed to the complex x_i plane) on which the solution should be T-periodic throughout and let us assume that x_3 has a singularity at complex time $t = \alpha T + \beta i$. Our argument will be simplest if we use x_3 because its initial position is the origin. It follows from the four-fold symmetry of the solution that x_3 is singular for values of t with real parts: $\pm\alpha T, T/2 \pm \alpha T$ (modulo T), and imaginary part $\pm\beta$. This is precisely what we see in our numerical analytic continuation (see Figure 4.12 and Table 4.5), four singularities for each body per period, with two bodies participating in each.

[1] To be clear, each body x_i takes complex values in a figure eight shape for real time t, but we also consider complex time, at which x_i can take any value in the complex plane

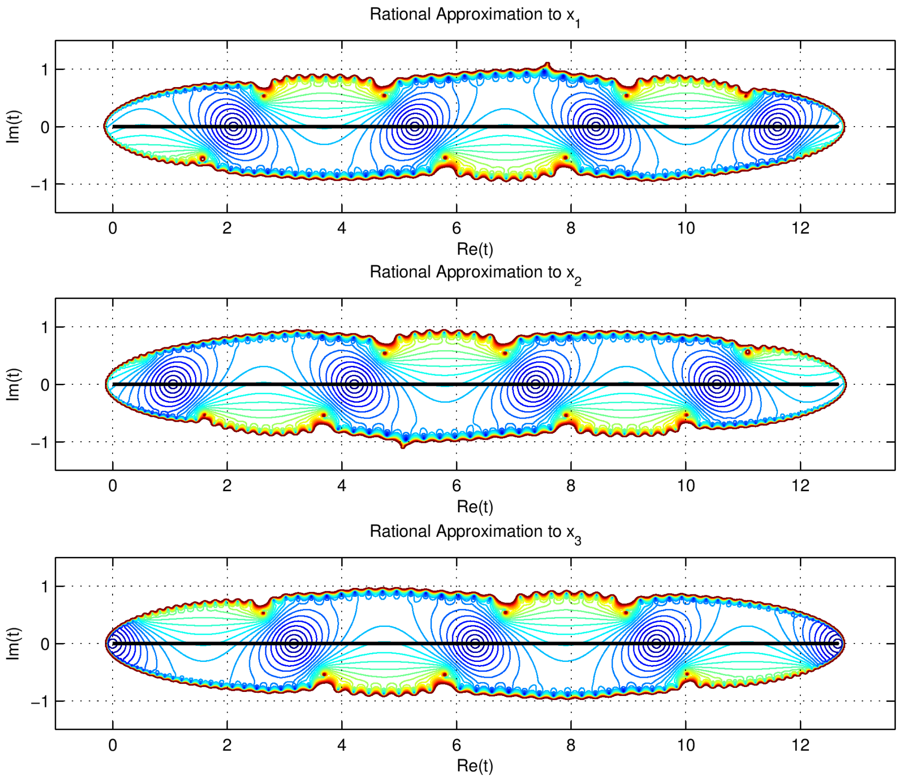

Fig. 4.12. *Contour plots of the figure-of-eight solution solved using* `ode113` *on the time interval* $[0, 12.6518]$, *extended into the complex plane using rational interpolation. Each red dot is a singularity in the system for complex time, with two bodies participating in each. The contours are coloured from blue to red on the interval* $[0, 2]$

Now, by construction the particles divide the circuit into three by time, so $x_1(t) = x_2(t + T/3) = x_3(t + 2T/3)$. Therefore if x_3 and x_2 (or x_1) participate in the singularity at αT, x_3 also has a singularity with real part $\alpha T - T/3$ (or $\alpha T - 2T/3$).

We can use these two symmetrical observations to find equations involving α modulo T; solving for α, we can find all possibilities for its numerical value. For example, $\alpha T + T/3 = T/2 - \alpha T$ implies $\alpha = 1/12$. One finds that the only possibilities are $\alpha = 1/12, 5/12, 7/12, 11/12$.

This gives us good reason to postulate that the real parts of the singularities of the figure of eight solution are (modulo T and to 4 decimal places),

$$\alpha T = \begin{cases} 1.5815, 2.6358, 4.7444, 5.7988 & (\text{for } x_1), \\ 0.5272, 1.5815, 3.6901, 4.7444 & (\text{for } x_2), \\ 0.5272, 2.6358, 3.6901, 5.7988 & (\text{for } x_3), \end{cases} \tag{4.16}$$

and our numerical results agree with this to 1 or 2 decimal places (see Table 4.5 and Figure 4.13). This is, as with the previous three examples, not very accurate, but nonetheless `ratinterp` did not produce any spurious poles.

The singularity structure of the figure of eight solution was brought to the author's attention in a private communication with Divakar Viswanath, for which the author is very grateful. The symmetry argument above for the possible locations of the

TABLE 4.5
Singularities of three bodies for the figure-of-eight solution to the three body problem. Two bodies participate in each singularity.

r_1	r_2	r_3	Separation
$1.5776 - 0.5735i$	$1.6141 - 0.5387i$		5.0×10^{-2}
$2.6331 + 0.5447i$		$2.6309 + 0.5416i$	3.8×10^{-3}
	$3.6898 - 0.5511i$	$3.6886 - 0.5484i$	3.0×10^{-3}
$4.7462 + 0.5518i$	$4.7440 + 0.5533i$		2.7×10^{-3}
$5.7981 - 0.5532i$		$5.8002 - 0.5527i$	2.2×10^{-3}
	$6.8537 + 0.5532i$	$6.8516 + 0.5527i$	2.2×10^{-3}
$7.9078 - 0.5533i$	$7.9057 - 0.5518i$		2.6×10^{-3}
$8.9621 + 0.5511i$		$8.9633 + 0.5484i$	3.9×10^{-3}
	$10.0187 - 0.5447i$	$10.0210 - 0.5416i$	3.9×10^{-3}
$11.0378 + 0.5387i$	$11.0743 + 0.5735i$		5.0×10^{-2}

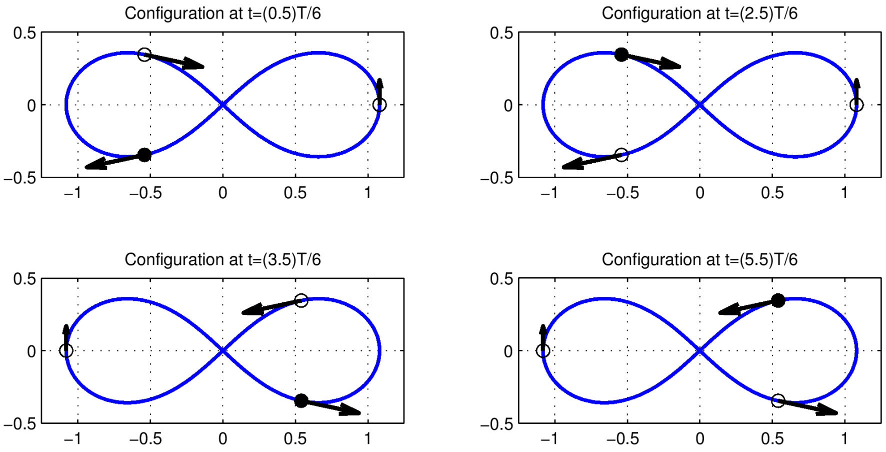

FIG. 4.13. *The configuration of the particles at times* $t = 0.5272, 2.6358, 3.6901, 5.7988$*, the real parts of the singular points of* x_3*. Coloured black is* $x_3(t)$ *with* x_1 *and* x_2 *shown as empty circles. The bodies form an isosceles triangle at these times.*

singularities is due him as well.

5. Discussion.

5.1. Computing Complex Singularities of Differential Equations. We have performed numerical experiments using Chebfun's `ratinterp` for computing complex singularities of solutions to some interesting ODE problems. We used a strategy developed in preliminary experiments and demonstrated that we can successfully find the singularities of difficult problems while avoiding spurious poles. The claim of Gonnet et al. [9], that the algorithm is robust, is evidently justified.

The robust algorithm `ratinterp` has potential for the numerical study of parabolic PDEs and parametrised problems, because implementation of our strategy allows the computation of the singularities at each time step or parameter variation to be automated. It is however, not suitable for applications which require a high level of accuracy. As was pointed out in [9] the error in the rational approximant is increased by the inclusion of the robustness procedure, dependent on the *tol* parameter. But if `ratinterp` is used to *reliably* find the approximate locations of the singularities, other methods such as the method of steepest ascent can be used to find precise locations [7].

5.2. Stability of Barycentric Interpolation Formulas for Extrapolation. During our experiments, we noticed a surprising instability. When `ratinterp` returns the rational approximant, if we use Chebyshev points, equispaced points around the unit circle or some other specific types of points, it returns a rational barycentric interpolation formula. This is a formula of the form

$$r(x) = \sum_{j=0}^{N} \frac{w_j f(x_j)}{x - x_j} \bigg/ \sum_{j=0}^{N} \frac{w_j}{x - x_j}, \quad w_j = \frac{q(x_j)}{\prod_{i \neq j}(x_i - x_j)}, \tag{5.1}$$

where f is the function we are approximating in the interpolation points, $\{x_j\}$ (which throughout this paper have been Chebyshev points on $[a, b]$). It is a tricky calculation to show that for Chebyshev points the weights are $w_j = (-1)^j q(x_j)$, after cancelling out factors independent of j, with w_0 and w_N equal to half this formula [2].

The instability arises for x outside of the Chebfun ellipse for f (defined in the Section 2), and is due to the following formula:

$$\sum_{j=0}^{N} \frac{w_j}{x - x_j} = \frac{q(x)}{\prod_{i=0}^{N}(x - x_i)}. \tag{5.2}$$

As the absolute value of x increases, the right hand side of (5.2) decreases to machine precision, beyond which it cannot decrease any further. The result is a dramatic loss of accuracy in (5.1) until there are no accurate digits at all. We can see in Figure 4.1 for example, that a polynomial can be extrapolated throughout the Bernstein ellipse, but outside of that it is useless as an approximation of the underlying function, so if we cannot stably evaluate our *rational* approximant outside of the Bernstein ellipse we are doing no better than we can with a polynomial! More detail can be found in a short article written with Trefethen and Gonnet [22]. `ratinterp` has since been corrected to evaluate the rational function as the quotient of two polynomials, each evaluated by a numerically stable version of polynomial barycentric interpolation formula.

5.3. Further Work. There are some issues with the calculation of residues. In Gonnet et al.'s program `ratdisk` [9], the precursor to `ratinterp`, the residue is calculated using a trapezium rule approximation to the associated contour integral. This turns out not to be very accurate for all but the simplest of functions. The method used in Chebfun's `ratinterp` is implemented using MATLAB's `residue` function, which is much more accurate but represents an ill-posed problem; if the numerator is close to a polynomial with multiple roots then small changes in the data can make arbitrarily large changes in the resulting poles and residues [5].

We noted that the residues calculated in this way can be remarkably large, most probably because of this instability. A short study comparing different methods of calculating residues should be done to get to the bottom of the issue.

Gonnet, Güttel and Trefethen have produced a paper on a robust implementation of Padé approximation [8]. They achieve robustness using the SVD of a linearised problem, just as in the robust implementation of rational interpolation and least squares. The question of how the SVD of the linear system is related to the singularities in the Padé approximant should be explored too, just as for rational interpolation, and their comparison could lead to a more concrete theory of these SVD-based robust algorithms for rational approximation.

6. Acknowledgements. The author would like to thank Nick Trefethen for his guidance and supervision during the project. His advice and suggestions were invaluable, especially when he took the time to read and comment on the early drafts of this paper. Many thanks to André Weideman for his insight and stimulating discussion during his brief stay in the UK, Pedro Gonnet for his astounding expertise in scientific computing, Alex Townsend for the voice of reason in times of confusion, and everyone at Oxford University in the Numerical Analysis Group for their hospitality, especially the administrator, Lotti Ekert. This paper was part of a summer research project funded by an EPSRC Undergraduate Vacation Bursary.

REFERENCES

[1] G.A. Baker and P.R. Graves-Morris, *Padé approximants*, vol. 59, Cambridge Univ. Press, 1996.

[2] J.-P. Berrut and L.N. Trefethen, *Barycentric lagrange interpolation*, SIAM Review, 46 (2004), pp. 501–517.

[3] F. Bornemann, P. Clarkson, P. Deift, A. Edelman, A. Its, and D. Lozier, *Painlevé project on the web*, Physics Today, 63 (2010), p. 10.

[4] A. Chenciner and R. Montgomery, *A remarkable periodic solution of the three-body problem in the case of equal masses*, Annals of Mathematics, Second Series, 152 (2000), pp. 881–902.

[5] MATLAB documentation, `residue`.

[6] T.A. Driscoll, F. Bornemann, and L.N. Trefethen, *The chebop system for automatic solution of differential equations*, BIT Numerical Mathematics, 48 (2008), pp. 701–723.

[7] B. Fornberg and J.A.C. Weideman, *A numerical methodology for the Painlevé equations*, Journal of Computational Physics, (2011).

[8] P. Gonnet, S. Güttel, and L.N. Trefethen, *Robust Padé approximation via SVD*, SIAM Review, (2012).

[9] P. Gonnet, R. Pachón, and L.N. Trefethen, *Robust rational interpolation and least-squares*, Electronic Transactions on Numerical Analysis, 38 (2011), pp. 146–167.

[10] E. Hille, *Ordinary differential equations in the complex domain*, Dover Publications, 1997.

[11] F. Hoppensteadt, *Lotka-Volterra equation*, Scholarpedia.

[12] E.N. Lorenz, *Deterministic nonperiodic flow*, Atmos. J. Sci., 20 (1963), pp. 130–141.

[13] R. Montgomery, *N-body choreographies*, Scholarpedia.

[14] C. Moore, *Braids in classical dynamics*, Physical Review Letters, 70 (1993), pp. 3675–3679.

[15] R. Pachón, *Algorithms for polynomial and rational approximation in the complex domain*, PhD thesis, University Of Oxford, 2010.

[16] R. Pachon, P. Gonnet, and J. Van Deun, *Fast and stable rational interpolation in roots of unity and chebyshev points*, SIAM Journal on Numerical Analysis, 50 (2012), pp. 1713–1734.

[17] L.N. Trefethen, *PDE coffee table book: blow-up equation with exp(u) nonlinearity, http://people.maths.ox.ac.uk/trefethen/pdectb/blowup22.pdf.*

[18] ———, *Six myths of polynomial interpolation and quadrature*, Mathematics Today, August (2011), pp. 184–188.

[19] ———, *Approximation Theory and Approximation Practice*, SIAM, 2012.

[20] L.N. Trefethen and D. Bau, *Numerical linear algebra*, SIAM, 1997.

[21] D. Viswanath and S. Sahutoglu, *Complex singularities and the Lorenz attractor*, SIAM Review, 52 (2010), pp. 294–314.

[22] M. Webb, L.N. Trefethen, and P. Gonnet, *Stability of barycentric interpolation formulas for extrapolation*, SIAM J. Sci. Comput, (2012).

[23] J.A.C. Weideman, *Computing the dynamics of complex singularities of nonlinear PDEs*, SIAM J. Appl. Dyn. Syst, 2 (2003), pp. 171–186.