

ROBUST PADÉ APPROXIMATION VIA SVD*

PEDRO GONNET[†], STEFAN GÜTTEL[‡], AND LLOYD N. TREFETHEN[§]

Abstract. Padé approximation is considered from the point of view of robust methods of numerical linear algebra, in particular the singular value decomposition. This leads to an algorithm for practical computation that bypasses most problems of solution of nearly-singular systems and spurious pole-zero pairs caused by rounding errors; a Matlab code is provided. The success of this algorithm suggests that there might be variants of Padé approximation that would be pointwise convergent as the degrees of the numerator and denominator increase to ∞ , unlike traditional Padé approximants, which converge only in measure or capacity.

Key words. Padé approximation, SVD, regularization, Froissart doublet, epsilon algorithm, Nuttall–Pommerenke theorem

AMS subject classifications. 41A21, 65F22

1. Introduction. Padé approximants, like rational approximants more generally, are well known to be fragile. One complication is that degeneracies may occur in which the numerator and denominator have less than the allowed degree, which leads to several entries in the Padé table being identical, some of them matching the Taylor series of the function being approximated to less than the expected order. Another complication is that even in theory, in the absence of rounding errors on a computer, Padé approximants are subject to the appearance of seemingly spurious pole-zero pairs or “Froissart doublets” in arbitrary locations that prevent pointwise convergence of the kind one might hope for. When rounding errors or other forms of noise are brought into the picture, such anomalies become almost ubiquitous.

In this paper we shall propose a theoretical variation on Padé approximation, and a corresponding numerical algorithm, that go a good way toward eliminating these anomalies. Philosophically speaking, our approach is one of regularization of an ill-posed problem, and perhaps it is no surprise that it relies upon the singular value decomposition (SVD). The reason Padé approximation is ill-posed is that it is related to analytic continuation, since the aim is typically to gain information about a function in a region of the complex plane based on information at a single point.

This paper is an outgrowth of earlier work with Pachón, which proposed an algorithm for robust rational interpolation in roots of unity [15]. The present problem is a limiting case in which the points of interpolation degenerate to a single point. Here one can take advantage of the exceptionally clean theory of square blocks of degeneracies in the Padé table. Our algorithm runs in standard floating point computer arithmetic and treats the mathematically classical problem of computing the type (m, n) Padé approximant from data given as a sequence of $m + n + 1$ Taylor coefficients. This is in contrast to related algorithms in the signal processing and model reduction literatures, mentioned in the discussion at the end, which attempt to extract low-rank signals from potentially much longer sequences of noisy data.

* P. G. was supported by Swiss National Science Foundation Individual Support Fellowship No. PBEZP2-127959, and S. G. was supported by Deutsche Forschungsgemeinschaft Fellowship No. GU 1244/1-1.

[†]Oxford University Mathematical Institute, 24–29 St Giles, Oxford OX1 3LB, UK

[‡]Oxford University Mathematical Institute, 24–29 St Giles, Oxford OX1 3LB, UK

[§]Oxford University Mathematical Institute, 24–29 St Giles, Oxford OX1 3LB, UK

2. Basic algorithm: Padé approximation in exact arithmetic. Let f be a function analytic in a neighborhood of $z = 0$ with Taylor series

$$f(z) = c_0 + c_1z + c_2z^2 + \cdots;$$

alternatively, f could be a formal power series. Given $n \geq 0$, let P_n denote the set of polynomials of degree at most n , and given $m \geq 0$ and $n \geq 0$, let R_{mn} be the set of rational functions of *type* (m, n) , that is, functions that can be written as quotients p/q with $p \in P_m$ and $q \in P_n$. If $\mu \leq m$ and $\nu \leq n$ denote the smallest integers such that $r \in R_{\mu\nu}$, then we say that r is of *exact type* (μ, ν) and has *defect* $\delta = \min\{m - \mu, n - \nu\} \geq 0$ with respect to R_{mn} . (In the special case $r = 0$ we define $\mu = -\infty, \nu = 0$.) The *type* (m, n) *Padé approximant* to f is the function $r_{mn} \in R_{mn}$ whose Taylor series at $z = 0$ matches that of f as far as possible:

$$(2.1) \quad r_{mn}(z) - f(z) = O(z^{\text{maximum}}).$$

It is known that r_{mn} exists and is unique and has the following characterization: if $r \in R_{mn}$ has defect δ with respect to R_{mn} , then $r = r_{mn}$ if and only if

$$(2.2) \quad r(z) - f(z) = O(z^{m+n+1-\delta}).$$

Here the “big O” notation has its usual meaning: the first nonzero term in the Taylor series of $r(z) - f(z)$ is Cz^k for some $k \geq m + n + 1 - \delta$. If $\delta = 0$, then p and q are uniquely determined up to a scale factor, whereas if $\delta > 0$, then p and q are only determined up to a scale factor if we make the additional assumption that they have degrees μ and ν , or equivalently, that they have no common roots. For extensive information about Padé approximation, see the book by Baker and Graves-Morris [1]. However, that monograph uses an alternative definition according to which a Padé approximant only exists if f can be matched to order z^{m+n+1} or further, and in fact the present paper is mathematically closer to the landmark review of Gragg [16], which uses the definition (2.1). For historical developments, see [5].

Equation (2.1) is nonlinear, but multiplying through by the denominator gives the linear condition

$$(2.3) \quad p(z) = f(z)q(z) + O(z^{\text{maximum}}).$$

By itself, this condition is vacuous, since matching to all orders could be achieved by taking p and q identically zero. The condition becomes meaningful when q is required to satisfy $q \not\equiv 0$. With this requirement, it is known that the matching condition can always be satisfied through degree $m + n$ or higher,

$$(2.4) \quad p(z) = f(z)q(z) + O(z^{m+n+1}),$$

as we shall confirm after (2.8).

Finding p and q to satisfy (2.4) is a linear algebra problem. Suppose \mathbf{a} and \mathbf{b} are $(m + 1)$ - and $(n + 1)$ -vectors of coefficients of polynomials $p \in P_m$ and $q \in P_n$, respectively,

$$\mathbf{a} = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix},$$

$$p(z) = \sum_{k=0}^m a_k z^k, \quad q(z) = \sum_{k=0}^n b_k z^k.$$

Then (2.4) can be written in matrix form, and it is here that our treatment of Padé approximation begins to depart from the usual. Usually one normalizes by a coefficient condition such as $b_0 = 1$, whereupon what remains in (2.4) is a system of linear equations that may be highly ill-conditioned or singular. Instead, following [15], we shall normalize by the condition

$$(2.5) \quad \|\mathbf{b}\| = 1,$$

where $\|\cdot\|$ is the vector 2-norm. This normalization will help to eliminate problems of singularity and ill-conditioning.

If $m \geq n$, equation (2.4) takes the Toeplitz form

$$(2.6) \quad \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \\ \vdots \\ a_m \\ \hline a_{m+1} \\ \vdots \\ a_{m+n} \end{pmatrix} = \begin{pmatrix} c_0 & & & & \\ c_1 & c_0 & & & \\ \vdots & & \ddots & & \\ c_n & c_{n-1} & \cdots & c_0 & \\ \vdots & \vdots & & \vdots & \\ c_m & c_{m-1} & \cdots & c_{m-n} & \\ \hline c_{m+1} & c_m & \cdots & c_{m+1-n} & \\ \vdots & \vdots & \ddots & \vdots & \\ c_{m+n} & c_{m+n-1} & \cdots & c_m & \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix}$$

coupled with the condition

$$(2.7) \quad a_{m+1} = \cdots = a_{m+n} = 0.$$

In other words, \mathbf{b} must be a (right) null vector of the $n \times (n+1)$ matrix displayed below the horizontal line. The coefficients a_0, \dots, a_m of p are then obtained by multiplying out the matrix-vector product above the line.

If $m \leq n$, the essence of the matter remains the same, though it is worth displaying the matrix anew to make it clear exactly what its form is:

$$(2.8) \quad \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \\ \hline a_{m+1} \\ \vdots \\ a_n \\ \vdots \\ a_{m+n} \end{pmatrix} = \begin{pmatrix} c_0 & & & & \\ c_1 & c_0 & & & \\ \vdots & & \ddots & & \\ c_m & c_{m-1} & \cdots & c_0 & \\ \hline c_{m+1} & c_m & \cdots & c_1 & c_0 \\ \vdots & \vdots & & \ddots & \ddots \\ c_n & c_{n-1} & & \ddots & c_1 & c_0 \\ \vdots & \vdots & & & \vdots & \\ c_{m+n} & c_{m+n-1} & \cdots & & c_m & \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix}.$$

Again we require (2.7) to hold, meaning that \mathbf{b} should again be a null vector of the matrix below the line, with \mathbf{a} again obtained by multiplying above the line. Since an

$n \times (n + 1)$ matrix always has a nontrivial null vector, we have confirmed (2.4), as promised. There is no assurance that \mathbf{a} and \mathbf{b} are unique, and we shall spell out the details of nonunique solutions in the next section.

Let us now focus on the portion of the linear algebra below the line, involving the $n \times (n + 1)$ matrix. This equation takes the form

$$(2.9) \quad \mathbf{0} = \tilde{C}\mathbf{b},$$

where \tilde{C} is the $n \times (n + 1)$ Toeplitz matrix

$$(2.10) \quad \tilde{C} = \begin{pmatrix} c_{m+1} & c_m & \cdots & c_{m+1-n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m+n} & c_{m+n-1} & \cdots & c_m \end{pmatrix}.$$

For simplicity in cases with $n > m + 1$, we have adopted the convention that $c_k = 0$ for $k < 0$. Let C denote the square $n \times n$ matrix obtained by deleting the first column of \tilde{C} :

$$(2.11) \quad C = \begin{pmatrix} c_m & \cdots & c_{m+1-n} \\ \vdots & \ddots & \vdots \\ c_{m+n-1} & \cdots & c_m \end{pmatrix}.$$

Many treatments of Padé approximation work with this matrix, solving a linear system of equations if its determinant is nonzero and bypassing it if its determinant is zero. Often C is flipped so that its structure is Hankel rather than Toeplitz.

We shall make use of the singular value decomposition of \tilde{C} , a factorization

$$(2.12) \quad \tilde{C} = U\Sigma V^*,$$

where U is $n \times n$ and unitary, V is $(n + 1) \times (n + 1)$ and unitary, and Σ is an $n \times (n + 1)$ real diagonal matrix with diagonal entries $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$.

Suppose $\sigma_n > 0$. Then \tilde{C} has rank n , and the final column of V provides a unique nonzero null vector \mathbf{b} of \tilde{C} up to a scale factor. This null vector defines the coefficients of q , with two subcases of special interest. If the square submatrix C is singular, then necessarily $b_0 = 0$. From (2.6) or (2.8) we see that this implies also $a_0 = 0$. Thus p and q share a common factor z , or possibly z^λ for some $\lambda > 1$, and this factor can be divided out at the end. If C is nonsingular, then b_0 must be nonzero, and the defect is $\delta = 0$.

On the other hand, suppose $\sigma_n = 0$. In this case \tilde{C} has rank $\rho < n$, with zero singular values $\sigma_{\rho+1} = \cdots = \sigma_n = 0$. Then C must have rank ρ or $\rho - 1$, so it is singular. In particular, the submatrix of \tilde{C} consisting of its last $\rho + 1$ columns must be rank-deficient, implying that \tilde{C} has a nonzero null vector that is zero in its first $n - \rho$ positions. The defect of the corresponding rational function is at least $n - \rho$, and we can reduce n to ρ and m to $m - (n - \rho)$ (or to 0 in the special case $r = 0$), and start again.

These observations suggest the following SVD-based algorithm for the calculation of the unique type (m, n) Padé approximant to a function f defined by its Taylor series.

ALGORITHM 1. PURE PADÉ APPROXIMATION IN EXACT ARITHMETIC

Input: $m \geq 0$, $n \geq 0$, and Taylor coefficients c_0, \dots, c_{m+n} of a function f .

Output: Polynomials $p(z) = a_0 + \cdots + a_\mu z^\mu$ and $q(z) = b_0 + \cdots + b_\nu z^\nu$, $b_0 = 1$, of the minimal degree type (m, n) Padé approximation of f .

1. If $c_0 = \cdots = c_m = 0$, set $p = 0$ and $q = 1$ and stop.
2. If $n = 0$, set $p(z) = c_0 + \cdots + c_m z^m$ and $q = 1$ and go to Step 8.
3. Compute the SVD (2.12) of the $n \times (n+1)$ matrix \tilde{C} . Let $\rho \leq n$ be the number of nonzero singular values.
4. If $\rho < n$, reduce n to ρ and m to $m - (n - \rho)$ and return to Step 2.
5. Get q from the null right singular vector \mathbf{b} of \tilde{C} and then p from the upper part of (2.6) or (2.8).
6. If $b_0 = \cdots = b_{\lambda-1} = 0$ for some $\lambda \geq 1$, which implies also $a_0 = \cdots = a_{\lambda-1} = 0$, cancel the common factor of z^λ in p and q .
7. Divide p and q by b_0 to obtain a representation with $b_0 = 1$.
8. Remove trailing zero coefficients, if any, from $p(z)$ or $q(z)$.

This algorithm produces the unique Padé approximant r_{mn} in a minimal-degree representation of type (μ, ν) with $b_0 = 1$. We state this result as a theorem, whose proof is part of a fuller discussion in the next section.

THEOREM 2.1. *Algorithm 1 (in exact arithmetic) converges in a finite number of steps to the unique normalized minimal degree representation of the type (m, n) Padé approximant $r_{mn} \in R_{mn}$ to f : polynomials p and $q \neq 0$ of exact degrees μ and ν with no common factors and $b_0 = 1$. The number of times that Step 3 is executed is no greater than $2 + \log_2(\delta + 1)$.*

3. Square blocks and proof of Theorem 2.1. An understanding of how Algorithm 1 works requires a discussion of block structure in the *Padé table*, by which we mean the array of Padé approximants r_{mn} for various $m, n \geq 0$ associated with a given function f :

$$\begin{pmatrix} r_{00} & r_{10} & r_{20} & \cdots \\ r_{01} & r_{11} & r_{21} & \cdots \\ r_{02} & r_{12} & r_{22} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

Suppose r is a rational function of exact type (μ, ν) that is the type (m, n) Padé approximant to f for at least one pair (m, n) . Then it is known that there is an integer $k \geq 0$ such that r is the type (m, n) Padé approximant to f if and only if $\mu \leq m \leq \mu + k$ and $\nu \leq n \leq \nu + k$ [16, Sec. 3], [29]. In other words, r is the Padé approximant to f precisely in the following $(k+1) \times (k+1)$ square block of the Padé table:

$$(3.1) \quad \begin{pmatrix} r_{\mu\nu} & \cdots & r_{\mu+k,\nu} \\ \vdots & & \vdots \\ r_{\mu,\nu+k} & \cdots & r_{\mu+k,\nu+k} \end{pmatrix}.$$

We have already defined the defect δ of r as a function of type (m, n) with $m \geq \mu$ and $n \geq \nu$. Showing the case $k = 5$ for illustration, δ takes these values within a

block:

$$(3.2) \quad \text{defect } \delta: \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 2 & 2 & 2 \\ 0 & 1 & 2 & 3 & 3 & 3 \\ 0 & 1 & 2 & 3 & 4 & 4 \\ 0 & 1 & 2 & 3 & 4 & 5 \end{pmatrix}.$$

Similarly we define the *deficiency* λ of r as the distance below the cross-diagonal in the square block, $\lambda = \max\{0, (m - \mu) + (n - \nu) - k\}$, with the following pattern:

$$(3.3) \quad \text{deficiency } \lambda: \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 2 & 3 & 4 & 5 \end{pmatrix}.$$

Finally, the *rank deficiency* χ of r is defined by the formula

$$(3.4) \quad \chi = \delta - \lambda,$$

and corresponds to the distance from the border of the square block:

$$(3.5) \quad \text{rank deficiency } \chi: \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 2 & 1 & 0 \\ 0 & 1 & 2 & 2 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The following characterization of Padé approximants, which explains why χ is called the rank deficiency, is equivalent to results derived in Section 3 of [16].

THEOREM 3.1. *Let f and $m, n \geq 0$ be given, let $\mu, \nu, k, \delta, \lambda, \chi$ be the parameters defined above for the type (m, n) Padé approximant r_{mn} to f , and let \hat{p} and $\hat{q} \neq 0$ be polynomials of exact degrees μ and ν with $r_{mn} = p/q$. Then the matrix \tilde{C} of (2.10) has rank $n - \chi$, and two polynomials $p \in P_m$ and $q \in P_n$, $q \neq 0$, satisfy (2.4) if and only if*

$$(3.6) \quad p(z) = z^\lambda w(z) \hat{p}(z), \quad q(z) = z^\lambda w(z) \hat{q}(z)$$

for some $w \in P_{\delta-\lambda}$.

It is worth summarizing in words some of the implications of (3.6). The minimal degree polynomials \hat{p} and $\hat{q} \neq 0$ of a Padé approximant r are unique apart from a scalar multiple. The defect δ determines how many additional degrees of freedom there are in the polynomials p and q of degrees $m \geq \mu$ and $n \geq \nu$ that represent r if the condition (2.3) is used. In the upper-left half of a square block, this is the same as the number of degrees of freedom in the polynomials p and q representing r by (2.4). In the lower-half of the block, however, representations (2.4) are more constrained than those of (2.3), with the number of degrees of freedom being equal to $\chi = \delta - \lambda$.

We are now equipped to prove Theorem 2.1. The reader may find it helpful to review the example summarized in Figure 5.2 in parallel with reading this proof.

Proof of Theorem 2.1. Each time Step 3 of Algorithm 1 is executed, either $\rho = n$ or $\rho < n$. In the first instance the algorithm executes Steps 5–8 and then stops, whereas in the second it returns to Step 2 with smaller values of m and n . Thus it terminates in at most δ steps. We must show that it terminates with p and $q \neq 0$ of exact degrees μ and ν , that is, at the upper-left corner of the square block.

One way the algorithm can terminate is with $n = 0$ at Step 2, followed by Step 8. In this case $q = 1$, with exact degree $\nu = 0$, and p must have exact degree μ since trailing zeros are removed at Step 8.

The other way the algorithm can terminate is with $\rho = n$ at Step 5 followed by Steps 6–8. In this case \tilde{C} is of full rank n , so $\chi = 0$ and (m, n) lies along the edge of the square block. If it lies in the left column or top row, then Step 8 brings it to the upper-left corner as required. If it lies in the right column or bottom row, then the cancellation of the common factor z^λ in Step 6 moves it to the top row or left column, respectively, and Step 8 brings it to the upper-left corner again.

This completes the proof except for the final assertion that the number of executions of Step 3 is at most $2 + \log_2(\delta + 1)$. If (m, n) lies in the top row or left column, then the number of such executions is 0 or 1, and since $\delta = 0$ this is less than $2 + \log_2(\delta + 1)$. If (m, n) lies elsewhere in the upper-left half of the square block, then the number of steps is at most 2, which again is less than $2 + \log_2(\delta + 1)$. Higher numbers become possible if (m, n) begins in the lower-right half of the square block, below the cross-diagonal and away from the boundary (hence always with $\delta > 0$). Here successive steps in the worst case might make (m, n) hop from one position to the next with $\chi = 1, 2, 4, 8, \dots$ until eventually the upper-left half of the block is reached; since χ can attain values no greater than $\delta/2$, the maximum number of such steps is $\log_2 \delta$. At this point two further steps must complete the process, giving at most $2 + \log_2 \delta$ steps in total. \square

4. Modified algorithm for noisy data. It is well known that the singular values of a matrix are well-conditioned functions of its entries. Specifically, if \tilde{C} is perturbed by a matrix of 2-norm at most ε , then each singular value is perturbed by at most ε . Thus Algorithm 1 immediately suggests a variant for the case in which the approximation problem is subject to noise, whether intrinsic to the data or introduced by rounding errors. The variant algorithm is this: carry out the operations as before, but treat a singular value as zero if it is less than $\text{tol} \cdot \|\mathbf{c}\|_2$, where $\mathbf{c} = (c_0, \dots, c_{m+n})$ and tol is a relative tolerance. The tolerance is applied also in the detection of zero coefficients in Steps 1, 6, and 8. For most purposes involving problems perturbed just by rounding errors, we take $\text{tol} = 10^{-14}$.

In addition, there is another matter to consider for practical computation. Algorithm 1 is implicitly tied to a scaling associated with the unit disk through the use of the 2-norm to measure the coefficient vector \mathbf{b} . In exact arithmetic this does not matter, but in practical applications it means the algorithm will be most effective when the coefficients $\{c_j\}$ of f are of roughly comparable sizes, neither decreasing nor increasing at a rapid geometric rate. Thus it may be beneficial first to pick a parameter $\gamma > 0$ and compute the Padé approximant $\hat{r}_{mn}(z) = r_{mn}(z/\gamma)$ to $\hat{f}(z) = f(z/\gamma)$. An algorithm for automatic determination of such parameters can be found in [9].

ALGORITHM 2. ROBUST PADÉ APPROXIMATION FOR NOISY DATA OR FLOATING POINT ARITHMETIC

Input: $m \geq 0, n \geq 0$, Taylor coefficients c_0, \dots, c_{m+n} of a function f , and relative tolerance $\text{tol} \geq 0$.

Output: Polynomials $p(z) = a_0 + \dots + a_\mu z^\mu$ and $q(z) = b_0 + \dots + b_\nu z^\nu$, $b_0 = 1$, of the minimal degree type (m, n) Padé approximation of a function close to f .

1. Rescale $f(z)$ to $f(z/\gamma)$ for some $\gamma > 0$ if desired to get a function whose Taylor coefficients c_0, \dots, c_{m+n} do not vary too widely.
2. Define $\tau = \mathbf{tol} \cdot \|\mathbf{c}\|_2$. If $|c_0| = \dots = |c_m| \leq \tau$, set $p = 0$ and $q = 1$ and stop.
3. If $n = 0$, set $p(z) = c_0 + \dots + c_m z^m$ and $q = 1$ and go to Step 7.
4. Compute the SVD (2.12) of the $n \times (n+1)$ matrix \tilde{C} . Let $\rho \leq n$ be the number of singular values of \tilde{C} that are greater than τ .
5. If $\rho < n$, reduce n to ρ and m to $m - (n - \rho)$ and return to Step 3.
6. Get q from the null right singular vector \mathbf{b} of \tilde{C} and then p from the upper part of (2.6) or (2.8).
7. If $|b_0|, \dots, |b_{\lambda-1}| \leq \mathbf{tol}$ for some $\lambda \geq 1$, zero the first λ coefficients of p and q and cancel the common factor z^λ .
8. If $|b_{n+1-\lambda}|, \dots, |b_n| \leq \mathbf{tol}$ for some $\lambda \geq 1$, remove the last λ coefficients of q .
9. Divide p and q by b_0 to obtain a representation with $b_0 = 1$.
10. Undo the scaling of Step 1 by redefining $\gamma^j a_j$ as a_j and $\gamma^j b_j$ as b_j for each j .

A Matlab code `padeapprox` implementing Algorithm 2 is shown in Figure 4.1 and is available at www.maths.ox.ac.uk/~trefethen. Three lines of this code, marked by comments beginning and ending with the word “reweighting,” go beyond Algorithm 2. These lines compute the final null vector by QR factorization of a column-reweighted matrix rather than by the SVD, an alternative that does a better job of taking advantage of sparsity in Toeplitz matrices. The effect is that the blocks produced in regions of a table corresponding to approximation accuracies close to `tol` more often come out exactly square.

5. Examples of computed Padé tables and noise removal. Figure 5.1 shows Padé tables computed by `padeapprox` with `tol` = 10^{-14} for the functions $\exp(z)$, $\cos(z)$, $(z^5 - 1)/(z^5 + 1)$, and $\log(5 + z^5)$. Each figure is based on the computation of 441 distinct Padé approximants and took about 1 second to produce in Matlab on a 2010 desktop machine. As described in the caption, the images show the block structures for the various functions clearly. Since the blocks arise from 441 independent computations for various (m, n) , this is a visual confirmation of the speed and reliability of Algorithm 2.

A departure from the theoretically expected block structure is apparent in the lower-right part of the tables for the first two functions. Here, m and n are larger than needed for resolution to machine precision, and the algorithm automatically reduces them by equal amounts, moving up and left along a diagonal to smaller values of m and n (Step 5 of Algorithm 2). Thus these diagonal stripes are indications of the robustness of Algorithm 2 in the presence of rounding errors.

Figure 5.2 shows another Padé table as in Figure 5.1, but now with some additional numbers displayed. The purpose of this figure is to illustrate how the algorithm works, as explained in the caption.

Figure 5.3 shows three more Padé tables, all corresponding to approximation of the same noisy Taylor series but with different levels of the parameter `tol`. This example is mentioned at the beginning of [13]. When `tol` is above the noise level, `padeapprox` detects the underlying rational function reliably.

6. Examples of the elimination of Froissart doublets. As we have mentioned, rounding errors or other perturbations commonly introduce Froissart doublets

```

function [r,a,b,mu,nu,poles,residues] = padeapprox(f,m,n,tol)
% Input: Function handle f or vector of coefficients f_0,...,f_(m+n).
%        Numerator and denominator degrees m>=0 and n>=0.
%        An optional 4th argument specifies relative tolerance tol.
%        If omitted, tol=1e-14. Use tol=0 to turn off robustness.
% Output: Function handle r of exact type (mu,nu) approximant to f
%        with coeff vectors a and b and optional poles and residues.
% P. Gonnet, S. Guettel, and L. N. Trefethen, October 2011

if nargin<4, tol = 1e-14; end % default rel tolerance 1e-14
if ~isfloat(f) % compute coeffs if necessary
    N = 2048; z = exp(2i*pi*(0:N-1)'/N); % sample at many roots of unity
    f = fft(f(z))/N; % Fast Fourier Transform
    tc = 1e-15*norm(f); f(abs(f)<tc) = 0; % discard near-zero coeffs
    if norm(imag(f),inf)<tc, f = real(f); end % make real functions real
end
c = [f(:); zeros(m+n+1-length(f),1)]; % make sure c is long enough
c = c(1:m+n+1); % but not longer than necessary
ts = tol*norm(c); % absolute tolerance
if norm(c(1:m+1),inf)<=tol*norm(c,inf) % special case r=0
    a = 0; b = 1; mu = -inf; nu = 0;
else
    row = [c(1) zeros(1,n)]; col = c; % 1st row/col of Toeplitz matrix
    while true % diagonal hopping across block
        if n==0, a = c(1:m+1); b = 1; break, end % special case n=0
        Z = toeplitz(col(1:m+n+1),row(1:n+1)); % Toeplitz matrix
        C = Z(m+2:m+n+1,:);
        rho = sum(svd(C)>ts); % numerical rank
        if rho==n, break, end
        m = m-(n-rho); n = rho; % decrease m,n if rank-deficient
    end
    if n>0 % hopping finished; compute b,a
        [U,S,V] = svd(C,0);
        b = V(:,n+1); % null vector gives b
        D = diag(abs(b)+sqrt(eps)); % reweighting preserves zeros better
        [Q,R] = qr((C*D).'); % so does final computation via QR
        b = D*Q(:,n+1); b = b/norm(b); % compensate for reweighting
        a = Z(1:m+1,1:n+1)*b; % multiplying gives a
        lam = find(abs(b)>tol,1,'first')-1; % count leading zeros of b
        b = b(lam+1:end); a = a(lam+1:end); % discard leading zeros of b,a
        b = b(1:find(abs(b)>tol,1,'last')); % discard trailing zeros of b
    end
    a = a(1:find(abs(a)>ts,1,'last')); % discard trailing zeros of a
    a = a/b(1); b = b/b(1); % normalize
    mu = length(a)-1; nu = length(b)-1; % exact numer, denom degrees
end
r = @(z) polyval(a(end:-1:1),z)... % function handle for r
    ./polyval(b(end:-1:1),z);
if nargin>5 % only compute poles if necessary
    poles = roots(b(end:-1:1)); % poles
    t = max(tol,1e-7); % perturbation for residue estimate
    residues = t*(r(poles+t)-r(poles-t))/2; % estimate of residues
end

```

FIG. 4.1. *Matlab code padeapprox for robust Padé approximation, based on Algorithm 2. The input function can be either a vector of Taylor coefficients or a function handle. The code is available at www.maths.ox.ac.uk/~trefethen.*

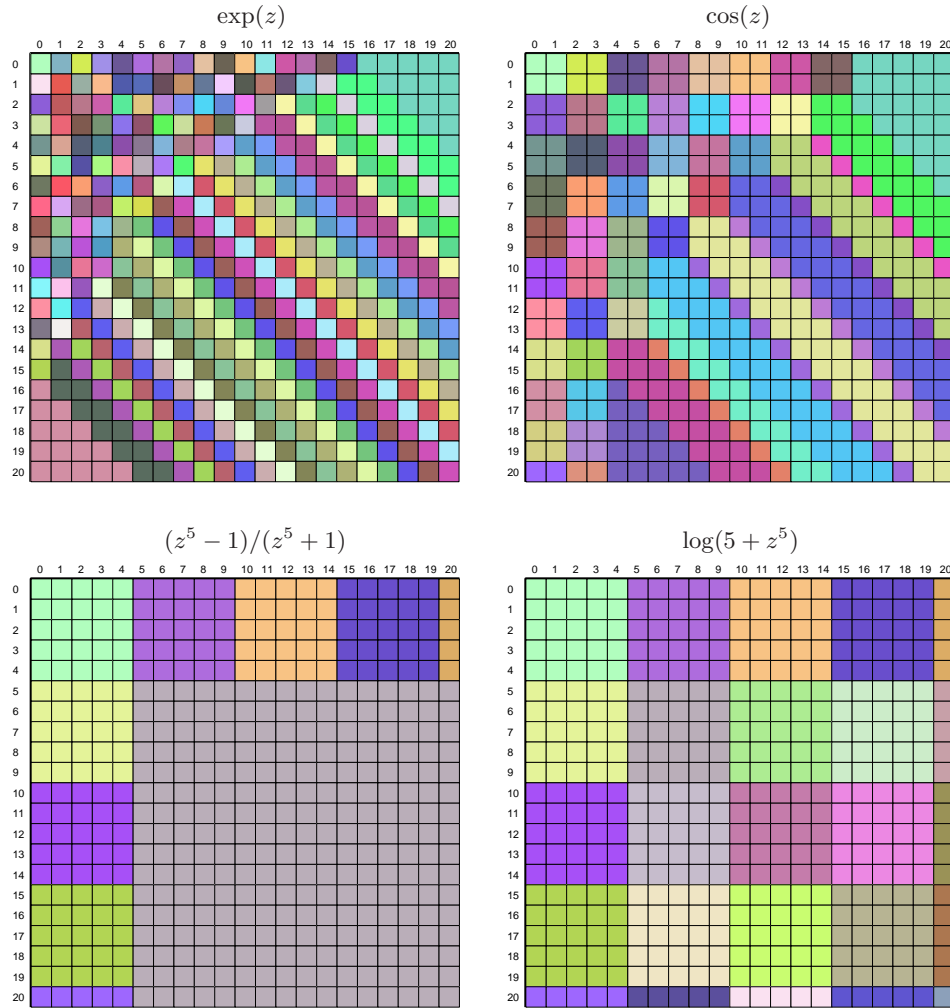


FIG. 5.1. Padé tables computed numerically by Algorithm 2, with m and n on the horizontal and vertical axes, respectively. Each square (m, n) is marked by a color determined by the exact type (μ, ν) of the corresponding approximant, so that each square block appears in a single color. For $\exp(z)$, all the entries lie in 1×1 blocks until the function is resolved to machine precision, after which the numerator and denominator degrees are systematically reduced as far as possible, causing diagonal stripes (Step 5 of Algorithm 2). For the even function $\cos(z)$, 2×2 square blocks appear. For $(z^5 - 1)/(z^5 + 1)$, we get an infinite square block since the function is resolved exactly for $m, n \geq 5$. For $\log(5 + z^5)$, there are 5×5 blocks all the way down.

in computed Padé approximations that do not reflect genuine information about f nor contribute to the quality of the approximation. We now give a few examples to illustrate how Algorithm 2 removes such effects by reducing the degrees m and n . Following a pattern employed in [15], each example is presented in the form of a two-part figure showing results from Algorithm 2 in its non-robust form with $\text{tol} = 0$ on the left, and in its robust form with the default value $\text{tol} = 10^{-14}$ on the right. The unit circle is marked by a dotted line. The pair (m, n) is listed on the upper-right. The lower-left of each plot lists the exact type (μ, ν) returned by `padeapprox` and the

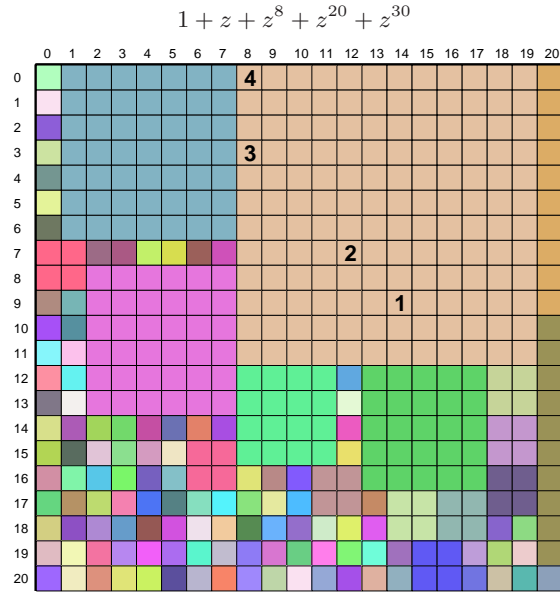


FIG. 5.2. The numerically computed Padé table for $f(z) = 1 + z + z^8 + z^{20} + z^{30}$, with numbers showing the path taken by Algorithm 2 for the particular case $(m, n) = (14, 9)$. Starting at this position of the table (label 1), the algorithm finds that the 9×10 Toeplitz matrix (2.10) has rank deficiency $\chi = 2$, so it moves to position (12, 7) (label 2, Step 5 of Algorithm 2). This 7×8 matrix has rank deficiency 4, causing a move to (8, 3) (label 3, Step 5). At this point three trailing zeros of \mathbf{b} are discarded (Step 8), bringing us to the final position (8, 0) (label 4).

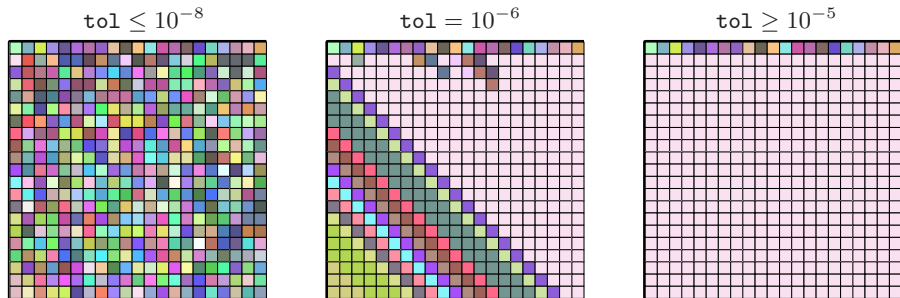


FIG. 5.3. An explicit example of noise removal by Algorithm 2. Here the Taylor series is defined by coefficients $c_j = 1 + 10^{-6}s_j$, where $\{s_j\}$ are independent samples from the standard normal distribution. This corresponds to the function $1/(1-z)$ plus noise on a scale of 10^{-6} . If `padeapprox` is run with `tol = 10-8` or a lower value such as the default 10^{-14} , the noise has the effect of making the Padé approximants distinct, and we see 1×1 blocks. With a tolerance above the noise level, `tol $\geq 10^{-5}$` , the code detects that this is essentially a rational function of type (0, 1).

elapsed time for computing this approximation on a 2010 desktop computer.

Each plot also lists a number `Err`, equal to the maximum of $|f(z) - r(z)|$ over the discrete grid of 1976 points in the disk $|z| \leq 0.5$ whose real and imaginary coordinates are odd multiples of 0.01. This gives a rough indication of the success of the algorithm in computing an effective approximation to f . The radius is chosen less than 1 to stay away from Froissart doublets clustered near the unit circle [13], but there is no

special significance to the particular choice 0.5.

Finally, each plot also shows the poles of r , marked by dots, following a scheme suggested by Grady Wright of Boise State University. The absolute value of the residue at each pole of r , evaluated by a finite difference, is indicated by a color code:

$$|\text{residue}| \in \begin{cases} [10^{-3}, \infty) & \text{blue} \\ [10^{-12}, 10^{-3}) & \text{green} \\ [0, 10^{-12}) & \text{red} \end{cases}$$

Thus a blue or green pole has a good chance of being genuine and useful for approximation, whereas red poles are likely to be artifacts introduced by rounding errors.

Figures 6.1–6.3 show Padé approximations to the function $f(z) = \tan(z^4)$, which has rows of poles outside the unit circle lying along eight rays emanating from the origin. For larger m and n , the removal of Froissart doublets by Algorithm 2 is striking.

Figure 6.4 shows results for a function with a branch point, $f(z) = \log(1.2 - z)$. According to a theory of Stahl, most of the poles of Padé approximants to such functions line up along certain branch curves determined by a capacity-minimization condition in the z^{-1} -plane [26]. (There is no assurance that *all* the poles must lie near the branch curves, merely a fraction of them approaching 1.) In this case the curve in question is the interval $[1.2, \infty)$, and the figure shows blue poles lining up as expected.

Figure 6.5 shows results for a function with an essential singularity, $f(z) = \exp((z + 1.5)^{-2})$.

7. Ill-posedness and stability. It is easy to show that Padé approximation problems are sometimes ill-posed. For example, here is a function and its type $(1, 1)$ Padé approximant:

$$f(z) = 1 + z^2, \quad r_{11}(z) = 1.$$

The approximant matches f only through power z^1 , but that is enough, according to (2.2), since the defect is 1. An arbitrarily small perturbation of f , however, changes r completely:

$$f(z) = 1 + \varepsilon z + z^2, \quad r_{11}(z) = \frac{1 - (1 - \varepsilon^2)z/\varepsilon}{1 - z/\varepsilon} \quad (\varepsilon \neq 0).$$

This perturbed function, which now matches f through order z^2 , has a Froissart doublet with a pole at $z = \varepsilon$ of residue $-\varepsilon^3$ and a zero at $\varepsilon/(1 - \varepsilon^2)$.

The example points the way to the general case: a Padé approximation problem is ill-posed if and only if r_{mn} has defect $\delta > 0$. The reason is that an arbitrarily small perturbation could fracture the block, forcing r_{mn} to match f to a higher order than before. For details see [29, 32].

Our point of view as numerical analysts is that it is not reasonable to ask an algorithm to find nearly the exact solution of an ill-posed problem. A more reasonable expectation is that an algorithm should be *stable*, finding nearly the exact solution of a slightly perturbed problem [31]. We hope to discuss stability of Algorithm 2 in a future publication. Note that a stable algorithm need not always produce square blocks, since different choices of m and n will correspond to different perturbations of the data.

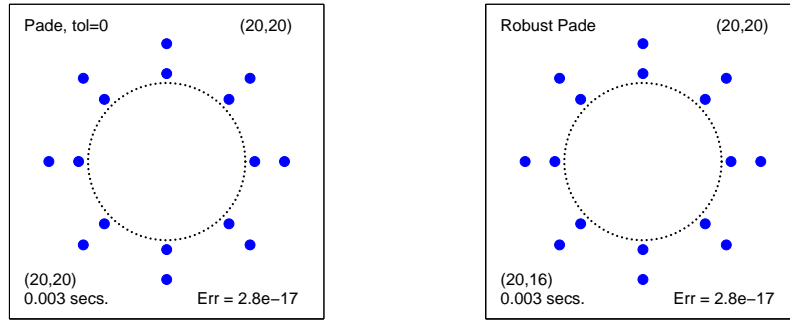


FIG. 6.1. Type $(20, 20)$ approximation of $\tan(z^4)$, a function with poles along eight rays emanating from the origin. Both the non-robust and robust algorithms place poles near the innermost 16 poles of f , the inner 8 matching the poles of f to six digits and the next 8 to two digits. There is little difference between the two algorithms except that the robust one reduces the type from $(20, 20)$ to $(20, 16)$, removing four poles with absolute value about 2×10^5 .

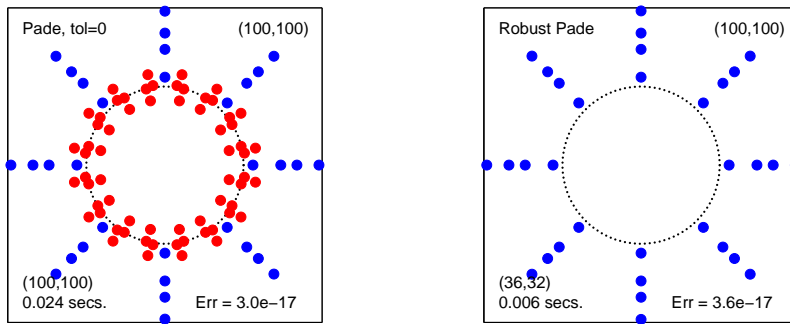


FIG. 6.2. Type $(100, 100)$ approximation of $\tan(z^4)$. Now each algorithm places four poles along each of the eight rays, which match the poles of f to approximately 14, 6, 3, and 2 digits (from inside out). These very accurate agreements of computed poles show how powerful Padé approximation can be for extracting information about a Taylor series beyond its circle of convergence. The non-robust algorithm also produces 64 Froissart doublets near the unit circle, as well as four additional doublets of absolute value about 3×10^3 , all of which are removed by the robust algorithm.

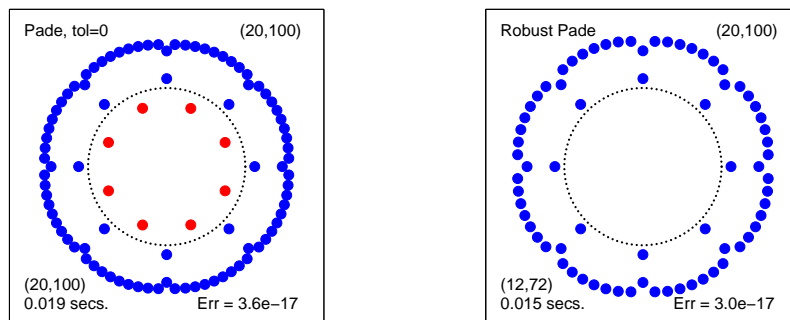


FIG. 6.3. The type $(20, 100)$ approximation of $\tan(z^4)$ shows the kind of effects that may arise with $n > m$. Here both the robust and non-robust approximations place some poles along a ring outside the unit disk.

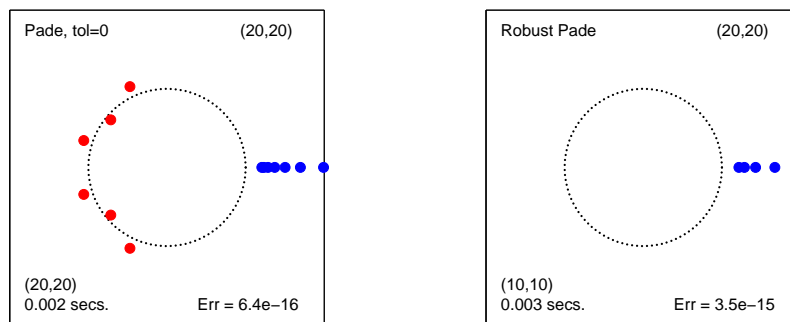


FIG. 6.4. Type (20, 20) approximation of $\log(1.2 - z)$. This function has a branch cut $[1.2, \infty)$ that attracts poles in keeping with the theory of Stahl [26]: 13 such in the non-robust approximation, 10 in the robust one. The non-robust algorithm scatters six additional spurious poles near the left half of the unit circle as well as two other negative real poles off the scale of the plot at about -6 and -8 .

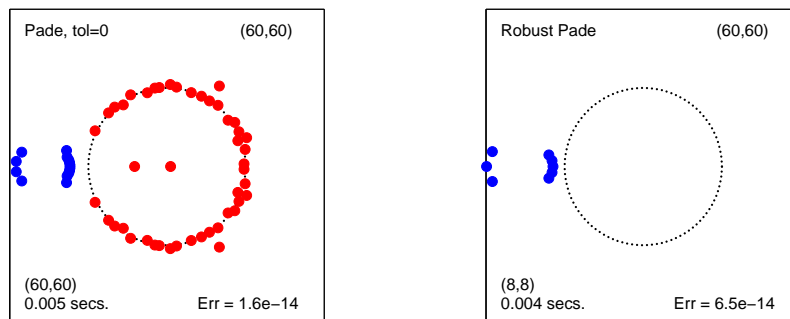


FIG. 6.5. Type (60, 60) approximation of $f(z) = \exp((z + 1.5)^{-2})$, a function with an essential singularity at $z = -1.5$.

The idea of regularization, which applies across a wide range of ill-posed problems, involves a balance between the accuracy of a solution and its other properties such as smoothness [18]. In eliminating Froissart doublets, we are slightly reducing the accuracy of our solution to a certain matrix problem, but the benefit is that the resulting function r_{mn} will be pole-free in most regions where f itself is pole-free. Consequently, in an application, it may be a better solution to the scientific problem that lies behind the ill-posed linear algebra.

8. A modified Padé approximant with pointwise convergence. The main purpose of this paper has been to propose a regularized algorithm for computing Padé approximations in floating point arithmetic or for problems with noise. However, one of the interesting features of the SVD-based approach is that it may also be applicable in the theoretical setting of exact arithmetic for problems with no noise, because this situation too is afflicted by seemingly spurious pole-zero pairs.

How can an exactly correct pole-zero pair be spurious? The phenomenon in question has been recognized for a century, going back at least to Perron in 1913 [24], and a formal definition of spurious poles is given in [27]. In the simplest case, suppose f is a meromorphic function in the complex plane \mathbb{C} , and consider the behavior of $f - r_{mn}$ as $n \rightarrow \infty$. A natural hope might be that on any compact set E disjoint from the poles of f , the supremum norm of $f - r_{mn}$ over E , $\|f - r_{mn}\|_E$, should converge to

zero. However, this does not happen in general. The Padé approximants r_{mn} can have poles in arbitrary locations in \mathbb{C} , and as $n \rightarrow \infty$, although the residues of these poles will decrease, they need never disappear entirely. In fact it may even happen that the type (n, n) Padé approximants to a fixed entire function f have so many spurious poles that the sequence of approximants is unbounded at every nonzero point in the complex plane [35].

In the face of this fact about Padé approximation, theorems have been developed that assert a weaker kind of convergence: *convergence in measure* or *convergence in capacity*. For meromorphic functions there is the Nuttall–Pommerenke Theorem [1, 20, 25], and for functions with branch points, there is a beautiful generalization by Stahl mentioned in the last section [26]. For the simpler problem of approximations of type (m, n) with $m \rightarrow \infty$ while n remains fixed, there is the de Montessus de Ballore Theorem [1, 19], and even in this case one must be careful to exclude spurious poles.

Our SVD-based algorithm suggests another possibility. Can one define a modified Padé approximant \tilde{r}_{mn} that is guaranteed to converge, with no exceptional sets? For each n , one could define \tilde{r}_{nn} to be the rational function computed in exact arithmetic by an algorithm like Algorithm 2 with tolerance $\text{tol} = \text{tol}_n$, where tol_n is a function of n that decreases to 0 as $n \rightarrow \infty$. Might such an approximation scheme be pointwise convergent? This is a challenge for theorists.

9. Discussion. Padé approximations arise in many applications, and Algorithm 2 will not be appropriate for all of them. Our framework is numerical approximation theory, in the spirit of [30] and [33]. Applications we have in mind include analytic continuation and extrapolation of sequences and series, both based on data at a single point in the complex plane. For data on a circle, we recommend the least-squares approach of [15], and for data on an interval, see the `ratinterp` command in Chebfun [33].

Many methods that have been proposed for extrapolation, such as the Aitken, Shanks, eta and epsilon methods, are mathematically equivalent to the evaluation of certain Padé approximants but have traditionally been formulated algorithmically by fast recurrence relations related to continued fractions, in the interests of speed. Indeed, the Toeplitz structure of the Padé problem makes possible some remarkably fast algorithms, in theory [4]. This focus on fast recurrences may be the right strategy for some applications, but we suspect that for many problems nowadays, it is better to use more robust algorithms, whose $O(n^3)$ complexity will rarely be a problem.

A central feature of our algorithm is that it removes spurious Froissart doublets as a by-product of the use of numerical ranks computed with the SVD. Other methods for removing doublets have also been proposed, typically based on explicit computation of zeros and poles, sometimes with the use of extended precision arithmetic [2, 3, 6, 8, 12, 13, 14, 28]. On the other hand Perotti, Vrinceanu, and Bessis argue that Froissart doublets can actually be beneficial in the sense that perturbations in their distribution from a regular pattern along the unit circle may reveal locations of underlying genuine poles [23].

An example of a different set of applications related to Padé approximation are problems of signal processing or model reduction, where the aim is to extract a low-order model from a sequence of hundreds of possibly noisy data values [2, 7, 17, 21]. In such problems the sampled data are typically assumed to come from a stable system, which would correspond for our problem to poles only outside the unit disk, and low-rank fits to sufficiently long data sequences will inherit this stability property. That is a different setting from the one we have focussed on here, which starts from arbitrary

Taylor coefficients, without an a priori assumption of scaling to the unit disk.

Padé approximants also appear in the study of Krylov subspace iterations for the solution of large matrix problems. Here the issue of degeneracies and square blocks has been of considerable interest, and “look-ahead” variants of some algorithms have been devised to jump around square blocks [10]. A feature of such problems is that since the scales are large, one is necessarily working with partial information obtained from recurrences, so the systematic use of the SVD as in Algorithm 2 is not an option.

The algorithm we have proposed works well, but we do not regard it as a last word on this subject. Other choices could have been made, and in particular, our method of hopping across a square block to find the upper-left corner is not the only reasonable one. For example, one could reduce just n rather than both m and n in cases of rank deficiency, and this would lead to vertical rather than diagonal stripes in the upper panels of Figure 5.1. (See [34] for an algorithm to find the upper-left corner of a square block in the case of Carathéodory–Fejér approximation.) The choices we have made seem to be effective, and one must remember that for most functions, few reductions of (m, n) will take place at all, so getting to the upper-left corner of a block by absolutely the shortest route possible is not a high priority. The true priority is robustness, and we believe Algorithm 2 meets this need effectively.

Acknowledgments. We have had fruitful discussions with a number of colleagues that have led to improvements in this article. We are grateful in particular for contributions from Daniel Bessis, Jaček Gilewicz, Ricardo Pachón, Ed Saff, Marc Van Barel, Joris Van Deun, and Grady Wright.

REFERENCES

- [1] G. A. BAKER, JR. AND P. R. GRAVES-MORRIS, *Padé Approximants*, 2nd ed., Cambridge Univ. Press, 1996.
- [2] D. BELKIĆ AND K. BELKIĆ, *Padé–Froissart exact signal-noise separation in nuclear magnetic resonance spectroscopy*, J. Phys. B, 44 (2011), 125003.
- [3] D. BESSIS, *Padé approximations in noise filtering*, J. Comput. Appl. Math., 66 (1996), pp. 85–88.
- [4] R. P. BRENT, F. G. GUSTAVSON, AND D. Y. Y. YUN, *Fast solution of Toeplitz systems of equations and computation of Padé approximants*, J. Algorithms, 1 (1980), pp. 259–295.
- [5] C. BREZINSKI, *History of Continued Fractions and Padé Approximants*, Springer, 1991.
- [6] C. BREZINSKI AND M. REDIVO-ZAGLIA, *Padé-type rational and barycentric interpolation*, manuscript, 2011.
- [7] P. FELDMANN AND R. W. FREUND, *Efficient linear circuit analysis by Padé approximation via the Lanczos process*, IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, 14 (1995), pp. 639–649.
- [8] C. FITZGERALD, *Confirming the accuracy of Padé table approximants*, in E. B. Saff and R. S. Varga, eds., *Padé and Rational Approximation: Theory and Applications*, Academic Press, 1976.
- [9] B. FORNBERG, *Numerical differentiation of analytic functions*, ACM Trans. Math. Software, 7 (1981), pp. 512–526.
- [10] R. W. FREUND, M. H. GUTKNECHT AND N. M. NACHTIGAL, *An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices*, SIAM J. Sci. Comput., 14 (1993), pp. 137–158.
- [11] G. FROBENIUS, *Über Relationen zwischen den Näherungsbrüchen von Potenzreihen*, J. Reine Angew. Math., 90 (1881), pp. 1–17.
- [12] M. FROISSART, *Approximation de Padé: application à la physique des particules élémentaires*, in RCP, Programme No. 25, v. 9, CNRS, Strasbourg (1969), pp. 1–13.
- [13] J. GILEWICZ AND M. PINDOR, *Padé approximants and noise: a case of geometric series*, J. Comp. ut Appl. Math., 87 (1997), pp. 199–214.
- [14] J. GILEWICZ AND M. PINDOR, *Padé approximants and noise: rational functions*, J. Comput. Appl. Math., 105 (1999), pp. 285–297.

- [15] P. GONNET, R. PACHÓN AND L. N. TREFETHEN, *Robust rational interpolation and least-squares*, Electron. Trans. Numer. Anal., 38 (2011), pp. 146–167.
- [16] W. B. GRAGG, *The Padé table and its relation to certain algorithms of numerical analysis*, SIAM Review, 14 (1972), pp. 1–62.
- [17] S. GUGERCIN AND A. C. ANTOULAS, *Model reduction of large-scale systems by least-squares*, Linear Algebra Appl., 415 (2006), pp. 290–321.
- [18] P. C. HANSEN, *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM, 1998.
- [19] R. DE MONTESSUS DE BALLORE, *Sur les fractions continues algébriques*, Bull. Soc. Math. France, 30 (1902), pp. 28–36.
- [20] J. NUTTALL, *The convergence of Padé approximants of meromorphic functions*, J. Math. Anal. Appl., 31 (1970), pp. 147–153.
- [21] E. A. O’SULLIVAN AND C. F. N. COWAN, *Modelling room transfer functions using the decimated Padé approximant*, IET Signal Process., 2 (2008), pp. 49–58.
- [22] H. PADÉ, *Sur la représentation approchée d’une fonction par des fractions rationnelles*, Thesis, Ann. École Nor. (3), 9 (1892), pp. 1–93, supplement.
- [23] L. PEROTTI, D. VRINCEANU, AND D. BESSIS, *Signal induced symmetry breaking in noise statistical properties of data analysis*, arXiv 1009:6021v1, 2010.
- [24] O. PERRON, *Die Lehre von den Kettenbrüchen*, BG Teubner, 1913.
- [25] CH. POMMERENKE, *Padé approximants and convergence in capacity*, J. Math. Anal. Appl., 41 (1973), pp. 775–780.
- [26] H. STAHL, *The convergence of Padé approximants to functions with branch points*, J. Approx. Theory, 91 (1997), pp. 139–204.
- [27] H. STAHL, *Spurious poles in Padé approximation*, J. Comp. Appl. Math., 99 (1998), pp. 511–527.
- [28] P. STOICA AND T. SÖDERSTRÖM, *Common factor detection and estimation*, Automatica, 33 (1997), pp. 985–989.
- [29] L. N. TREFETHEN, *Square blocks and equioscillation in the Padé, Walsh, and CF tables*, in P. R. Graves-Morris, E. B. Saff, and R. S. Varga, eds., *Rational Approximation and Interpolation*, Springer Lect. Notes in Math. 1105, 1984.
- [30] L. N. TREFETHEN, *Approximation Theory and Approximation Practice*, book in preparation.
- [31] L. N. TREFETHEN AND D. BAU, III, *Numerical Linear Algebra*, SIAM, 1997.
- [32] L. N. TREFETHEN AND M. H. GUTKNECHT, *On convergence and degeneracy in rational Padé and Chebyshev approximation*, SIAM J. Math. Anal., 16 (1985), pp. 198–210.
- [33] L. N. TREFETHEN ET AL., *Chebfun software package*, www.maths.ox.ac.uk/chebfun/.
- [34] J. VAN DEUN AND L. N. TREFETHEN, *A robust implementation of the Carathéodory–Fejér method*, BIT Numer. Math., 2011.
- [35] H. WALLIN, *On the convergence theory of Padé approximants*, in Linear Operators and Approximation, Internat. Ser. Numer. Math., 20 (1972), pp. 461–469.