A robust implementation of the Carathéodory-Fejér method

Joris Van Deun and Lloyd N. Trefethen

June 8, 2010

Abstract

Best rational approximations are notoriously difficult to compute. However, the difference between the best rational approximation to a function and its Carathéodory-Fejér (CF) approximation is often so small as to be negligible in practice, while CF approximations are far easier to compute. We present a robust and fast implementation of this method in the chebfun software system and illustrate its use with several examples. Our implementation handles both polynomial and rational approximation and substantially improves upon earlier published software.

1 Introduction

The problem of obtaining the best rational (m, n) approximant to a given real continuous function f on a finite interval with respect to the supremum norm is a remarkably elusive one. The *characterization* of such an approximant is straightforward enough: the error equioscillates at least $m+n+2-\delta$ times, where δ is known as the *defect* of f. This result is due to de la Vallée Poussin [8, Theorem 98]. The actual computation of this approximant, however, is a much more complex matter. In particular, the Remez algorithm is very sensitive to the initial guess and often diverges.

Nevertheless, for many practical applications there is a perfectly viable alternative to best rational approximation, namely *Carathéodory-Fejér (CF) approximation*. CF approximation was introduced by the second author and Martin Gutknecht in a series of papers published mostly in the early 1980's [4, 5, 6, 17, 18, 19, 20, 22]. The foundations, however, were laid almost a century ago by Constantin Carathéodory and Lipót Fejér in [3], which deals with polynomial approximation, and by Issai Schur in [12, 13]. Their

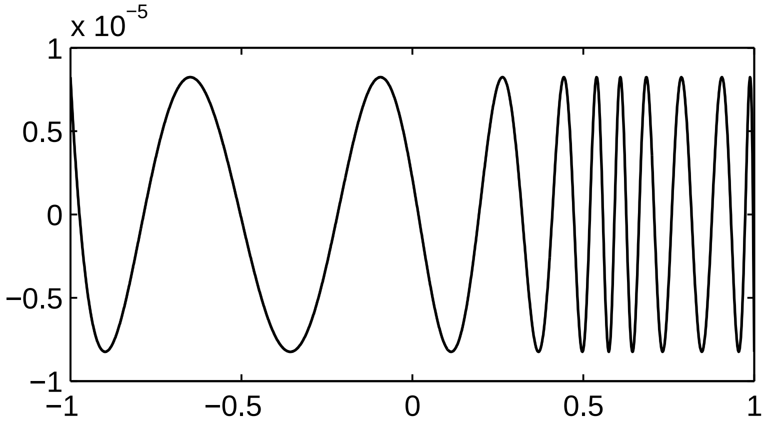

Figure 1: Error in $(10, 10)$ CF approximation to $f(x) = \log(1.2 + \cos(e^{2x}))$.

theory was later extended to the rational case by Takagi [15, 16] and further generalized by Adamjan, Arov and Krein [1]. Some recent applications of CF approximation include computing the gamma function using complex contours [11] and Talbot quadrature [23] for inverse Laplace transforms and solution of partial differential equations.

Before giving a brief overview of the theory behind CF approximation in Section 2, let us first illustrate the power of this method and its usage in the chebfun system. For information about chebfun, the reader is referred to [2, 10] and `www.maths.ox.ac.uk/chebfun`.

Suppose we wish to compute the $(10, 10)$ CF approximant to the function

$$f(x) = \log(1.2 + \cos(e^{2x}))$$

and plot the error curve. This is done by the following three lines of code:

```
>> f = chebfun('log(1.2+cos(exp(2*x)))');
>> [p,q] = cf(f,10,10);
>> plot(f-p./q)
```

Figure 1 shows that the error curve equioscillates 22 times, indicating that the CF approximation is indistinguishable from the best approximation.

2 CF approximation

The original CF method dealt with polynomial approximation on the complex unit circle. In [17] it is shown how to compute near-best polynomial approximants to analytic functions on the unit circle in the sense that the error curves approximate perfect circles about the origin. The procedure involves finding the principal singular value and corresponding singular vector of a complex Hankel matrix containing the Taylor series coefficients of the function.

In [18] this method is extended to rational approximation. The approach is very similar, only now non-principal singular values and vectors of the same matrix are needed. Both results are based on a theorem by Carathéodory and Fejér and its extensions.

The transition to a finite interval is made using the Joukowski function $x = \frac{1}{2}(z + z^{-1})$, which maps the complex unit circle to the interval $I = [-1, 1]$. A simple linear transformation then maps I to any finite interval $[a, b]$. To keep the exposition simple we work with the interval $[-1, 1]$, but the actual implementation can handle arbitrary finite intervals.

Real rational CF approximation on I is discussed in detail in [22]. We begin with a real function $F(x)$ continuous on I. For any finite $M \geq 0$, F possesses a partial Chebyshev expansion

$$F(x) = F_M(x) + G_M(x) = \sum_{k=0}^{M}{}' a_k T_k(x) + G_M(x)$$

where the prime indicates that the $k = 0$ term should be halved. Here a_k is defined by

$$a_k = \frac{2}{\pi} \int_{-1}^{1} F(x) T_k(x) \frac{dx}{\sqrt{1 - x^2}} \ . \tag{1}$$

If F is represented by a chebfun, we obtain the coefficients a_k as follows:

```
>> a = chebpoly(F);
>> a = a(end:-1:end-M);
>> a(1) = 2*a(1);
```

In keeping with Matlab tradition, `chebpoly` returns the coefficients from high to low degree, which is why we need the second command above. The last command is necessary because chebfun uses a slightly different definition for the coefficient a_0.

Next we want to relate F_M to an analytic function on the unit circle in order to apply the results of Carathéodory, Fejér, Schur and Takagi. This is done using the Joukowski transformation defined above. Putting $a_{-k} = a_k$ we have

$$F_M(x) = \frac{1}{2} f_M(z) = \frac{1}{2}[f^+(z) + f^+(z^{-1}) + f^0(z)]$$

where

$$f_M(z) = \sum_{k=-M}^{M} a_k z^k, \quad f^+(z) = \sum_{k=m-n+1}^{M} a_k z^k,$$

$$f^0(z) = \begin{cases} \displaystyle\sum_{k=n-m}^{m-n} a_k z^k & \text{if } m \geq n, \\ \displaystyle -\sum_{k=m-n+1}^{n-m-1} a_k z^k & \text{if } m < n. \end{cases}$$

The idea is to approximate the function $f^+(z)$ on the unit circle by an *extended* rational function of the form

$$\tilde{r}(z) = \frac{\displaystyle\sum_{k=-\infty}^{m} d_k z^k}{\displaystyle\sum_{k=0}^{n} e_k z^k}$$

where the terms of negative degree in the numerator converge to a bounded analytic function in $|z| > 1$ and the denominator has no zeros in $|z| \leq 1$. The space of all these functions is denoted by $\tilde{R}_{mn}$. Once this approximation is found, we can map back to the interval and then truncate to a true rational function in R_{mn}, the space of rational functions of degree at most m in the numerator and at most n in the denominator.

Let H be the real symmetric Hankel matrix

$$H = \begin{pmatrix} a_{m-n+1} & a_{m-n+2} & \cdots & a_M \\ a_{m-n+2} & & & 0 \\ \vdots & & & \vdots \\ a_M & 0 & \cdots & 0 \end{pmatrix}$$

and assume that its eigenvalues are ordered by absolute magnitude $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_{M+n-m}|$. Let λ abbreviate λ_{n+1} and let $(u_1, \cdots, u_{M+n-m})^T$ be the corresponding eigenvector. Then we have the following theorem proved by Carathéodory and Fejér for polynomials [3] and extended to the rational case by Takagi [15].

Theorem 1. *The analytic function f^+ has a unique best approximation $\tilde{r}^*$ on the unit circle $|z| = 1$ in $\tilde{R}_{mn}$ given by*

$$f^+ - \tilde{r}^* = b,$$

where b is the finite Blaschke product

$$b(z) = \lambda z^M \frac{u_1 + \cdots + u_{M+n-m} z^{M+n-m-1}}{u_{M+n-m} + \cdots + u_1 z^{M+n-m-1}} \; .$$

The approximation error is

$$\|f^+ - \tilde{r}^*\|_{|z|=1} = |\lambda|,$$

and the error curve $(f^+ - \tilde{r}^*)_{|z|=1}$ is a perfect circle about the origin whose winding number is $m+n+1$ if $|\lambda_n| > |\lambda| > |\lambda_{n+2}|$.

We can map back to I by defining

$$\tilde{R}(x) = \frac{1}{2}[f_M(z) - b(z) - b(z^{-1})].$$

It then follows that

$$F_M(x) - \tilde{R}(x) = \mathrm{Re}\, b(z),$$

which indeed equioscillates $m+n+2$ times on I. However, $\tilde{R}$ is in general not a rational function in R_{mn}, so we have to truncate it. This is done as follows. Let q be the polynomial of degree n or less whose zeros are the finite poles of $\tilde{r}^*$ outside the unit circle. Define

$$Q(x) = cq(z)q(z^{-1})$$

where c is such that the constant term of Q is 1. Then Q is a polynomial of degree n or less and it is the denominator of $\tilde{R}$. We now define the CF approximant as

$$R^{\mathrm{cf}}(x) = \frac{P(x)}{Q(x)} \; ,$$

with P defined by

$$\tilde{R}(x) = R^{\mathrm{cf}}(x) + O(T_{m+1}(x))$$

and $O(T_{m+1}(x))$ means that the Chebyshev expansion of $\tilde{R} - R^{\mathrm{cf}}$ starts with a term of order $m+1$. To compute P, we need the Chebyshev coefficients c_k of $\tilde{R}$ and γ_k of $1/Q$. Then the Chebyshev coefficients β_k of P satisfy the Toeplitz system

$$\begin{pmatrix} \gamma_0 & \gamma_1 & & \cdots & \gamma_{2m} \\ \gamma_1 & \ddots & \ddots & & \vdots \\ & \ddots & & & \\ \vdots & & & & \gamma_1 \\ \gamma_{2m} & \cdots & & \gamma_1 & \gamma_0 \end{pmatrix} \begin{pmatrix} \beta_m \\ \vdots \\ \beta_0 \\ \vdots \\ \beta_m \end{pmatrix} = 2 \begin{pmatrix} c_m \\ \vdots \\ c_0 \\ \vdots \\ c_m \end{pmatrix}$$

which can be reduced to a system of size $m+1$ instead of $2m+1$.

The condition $|\lambda_n| > |\lambda| > |\lambda_{n+2}|$ at the end of the Carathéodory-Fejér-Takagi theorem is related to the structure of the CF table. If for given F_M we organize the approximants in a table in the (m,n)-plane, then it follows from [4], [6] and [20] that this table consists of disjoint square blocks. In each block $\tilde{r}^*$ and $|\lambda|$ are fixed. All entries along any diagonal $m-n = const.$ correspond to a single Hankel matrix H, and the magnitudes of the absolute values of this matrix are the approximation errors as we descend along this diagonal. The length of the intersection of the diagonal with a square block equals the multiplicity of the corresponding eigenvalue (in magnitude). Furthermore, it is shown in [6] that the CF operator (the map from function to approximation) is continuous if and only if (m,n) lies in the lower-left or upper-right corner of such a square block.

3 Chebfun implementation

The first implementation of the CF method for real rational approximation appeared in [21], which so far as we know may have been the first research article ever published containing a Matlab program! Our current implementation is substantially different in several ways. A quick overview of the algorithmic innovations is given in the following bullet list, which we discuss in more detail in the rest of this section:

- Formulation within the chebfun system
- Arbitrary (finite) intervals
- Approximation of piecewise smooth functions
- Correct handling of block structure in the CF table, including even and odd functions
- Special treatment of (numerically) rational functions using Chebyshev-Padé approximation
- Detection of ill-conditioning
- Unified approach to polynomial and rational approximation
- Computation of only a few eigenvalues in case of non-smooth functions, with Matlab's `eigs` command
- Use of Henrici's method to extract the co-analytic part of $b(z)$

- Construction of $1/Q(x)$ from explicit knowledge of $Q(x)$
- Safety margins and argument checks to make the code more robust
- Reduction of the Toeplitz system of size $2m+1$ to a system of size $m+1$

The complete syntax of the CF command in chebfun is

```
[p,q,r_handle,s] = cf(f,m,n,M)
```

but some of the input and output parameters are optional, as explained in the help text. Here `p` and `q` are chebfuns representing the numerator and denominator polynomials of the approximant, `r_handle` is a function handle to a pointwise evaluator of the rational approximant itself, and `s` is the absolute value of the eigenvalue that measures the approximation error. The input parameter `f` is the chebfun we wish to approximate and `m` and `n` are the numerator and denominator degrees. The optional parameter `M` indicates the number of Chebyshev coefficients of `f` that we want to use.

3.1 Chebfun system

This CF implementation is part of the chebfun system, which means we can take advantage of the built-in chebfun methods. This is especially useful in computing the Chebyshev coefficients of the various functions discussed in Section 2 with the `chebpoly` command. The only exception here is in the computation of the coefficients c_k of $\tilde{R}(x)$, which we compute from the Laurent coefficients of the Blaschke product $b(z)$ using the FFT as explained in [22]. Using `chebpoly` here would be too slow.

Dealing with intervals different from I is also made easy in chebfun. The interval on which `f` is defined can be obtained with the command

```
>> d = domain(f);
```

and giving `d` as an extra argument to the chebfun constructor will produce a new chebfun defined on `d`.

Finally, `f` may consist of several pieces; see [10]. In this case `M` should be provided; a typical value might be in the low hundreds. This makes it possible to quickly approximate functions such as $|x|$.

3.2 Block structure of CF table

A major difference from previous implementations of the CF method, crucially important in practice, is that ours takes into account the structure of the CF table. This is done by the subfunction `getblock`. This function constructs the Hankel matrix for given `m`, `n` and `M`, computes the eigenvalues, and returns k and l such that $|\lambda_{n-k+1}| = \cdots = |\lambda| = \cdots = |\lambda_{n+l+1}|$. If either $k > 0$ or $l > 0$, we are in a nontrivial square block and should move to the upper-right or lower-left corner. The program first tries the upper-right corner by calling `getblock` again but now with the parameters `m+l`, `n-k`. If it turns out that we were below the main diagonal of the square, then this will not work and `getblock` is called a third time to move to the lower-left corner.

If the function to be approximated is itself a rational function in R_{mn}, then there is an infinite square block in the CF table. This can of course not be detected from a finite number of eigenvalues, but if $|\lambda| = |\lambda_{n+2}| = \cdots = |\lambda_{M+n-m}|$ then the program assumes that the block is infinite. Moving to the upper-right or lower-left corner of an infinite block is impossible, so in this case the CF method cannot be used. Instead we use `chebpade` to compute a Chebyshev-Padé approximation. This should reproduce the rational function exactly. However, if m or n are large, the results may be incorrect because of ill-conditioning.

A common case of predictable block structure is when `f` is an odd or even function, which leads to a tiling of the CF table by 2×2 blocks. To avoid having to compute the eigenvalues of a Hankel matrix several times, we test in the beginning whether `f` is odd or even by looking at the Chebyshev coefficients.

3.3 Ill-conditioning

The CF method works remarkably well for smooth functions, but it is known that the method is less useful for more difficult functions (e.g. when there are singularities close to the interval). This difficulty manifests itself in different ways.

Theoretically, the Blaschke product $b(z)$ can have at most n poles outside the complex unit circle. However, because of ill-conditioning, it may happen that there are poles inside the unit disk too. If this happens, the program discards those poles and outputs a warning message.

Another source of ill-conditioning is the Toeplitz system for computing the numerator polynomial P. If the denominator Q has zeros close to the interval, then the Toeplitz matrix becomes ill-conditioned and digits are inevitably lost. This only becomes a problem if we lose so many digits that the equioscillation property is lost, specifically, if $\kappa/|\lambda| > 1/\varepsilon$, where κ is the condition number of the Toeplitz matrix and ε is machine precision. To be on the safe side, we take $1/\varepsilon = 1e13$, a choice which works well in practice. Also here the program generates a warning message when it detects ill-conditioning.

3.4 Polynomial case

Although we have only discussed the case of rational approximation, the chebfun `cf` program can also be used for the special case of polynomial approximation. The computations, however, are much simpler and there are no problems of block structure or ill-conditioning. Our program takes special steps to handle the polynomial case efficiently, but we will not go into the details of the implementation. The interested reader is referred to [5].

3.5 Miscellaneous optimizations

If `f` is non-smooth it will have a long Chebyshev series representation and thus M will be large. Computing the eigenvalues of a large Hankel matrix is very time-consuming, especially in Matlab where no advantage is taken of the special structure of the matrix. Since we do not need all the eigenvalues, but only the $(n+1)$st, we can use `eigs` to return only the largest eigenvalues and speed up the computations. This makes a huge difference in cases where M is in the thousands. But in order to study the block structure, we may need more than $n+1$ eigenvalues, so the program computes $n+10$ eigenvalues and assumes that will be enough to draw conclusions. We only use `eigs` when $M > 1024$, otherwise we use `eig` to compute all the eigenvalues.

The polynomial $q(z)$ can be computed in two ways: by factoring the denominator polynomial of $b(z)$ with the `roots` command, or using a method described by Henrici in [7, §3.2] based on the FFT. If M is large then factoring the complete denominator polynomial becomes very expensive. Hence the program uses Henrici's method to obtain the small polynomial $q(z)$. In fact we could construct $Q(x)$ directly from $q(z)$ without even computing the roots, but for reasons of stability we nevertheless still factor $q(z)$ and then construct $Q(x)$ with the command

```
>> q = chebfun(@(x) real(prod(x-z)/prod(-z)),'vectorize');
```

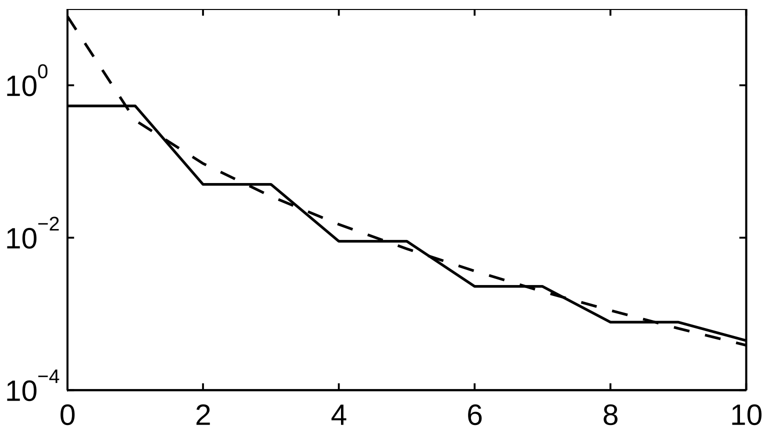

Figure 2: Approximation error (solid) for the (n,n) CF approximation to $|x|$ and the asymptotic error (dashed) of Varga, Carpenter, Ruttan and Stahl.

where z contains the roots of $q(z)$ transformed to the x-plane using the Joukowski map.

Since we have now computed the roots of $Q(x)$ explicitly, we can use this to our advantage when constructing a chebfun representation for $1/Q(x)$ (needed to set up the Toeplitz system). The root closest to the interval determines the exact ellipse of analyticity for $1/Q$, hence the convergence rate of the Chebyshev expansion, and hence the length of the chebfun representation. This length is passed as an extra argument to the chebfun constructor to speed up the computations. If we do not do this, it often happens (especially when there are roots close to the interval) that the computed Chebyshev coefficients reach a plateau which is still above the chebfun tolerance, and the constructor keeps on generating coefficients until one (accidentally) falls below the tolerance.

4 Numerical experiments

Both the software and hardware scenes have changed tremendously over the last 30 years, and computations that were unfeasible or far too time-consuming in 1984 have become almost trivial in 2010. In this section we illustrate our new CF implementation with several examples.

We first look at (n,n) rational approximations to $f_1(x) = |x|$. Following Newman's original discovery [9] it was conjectured in [24] and proved in [14]

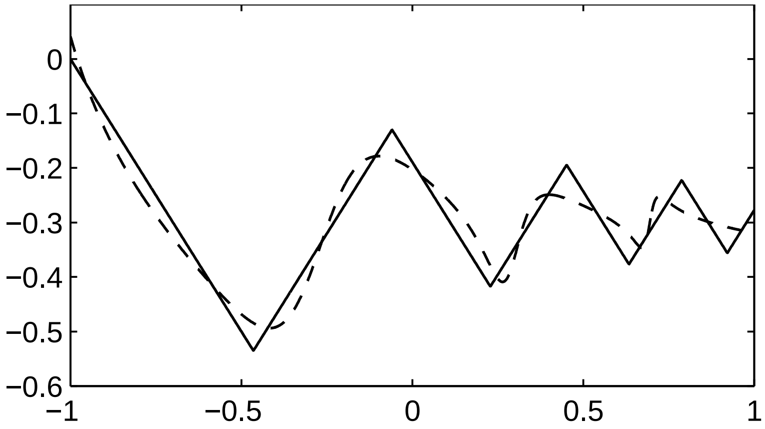

Figure 3: f_2 (solid) and its $(7,7)$ CF approximant using 70 Chebyshev coefficients (dashed).

that

$$\lim_{n\to\infty} e^{\pi\sqrt{n}} E_{nn}(|x|;[-1,1]) = 8,$$

where $E_{nn}(|x|;[-1,1])$ denotes the error of best uniform approximation to $|x|$ on $[-1,1]$ by (n,n) rational functions. Figure 2 plots the asymptotic estimate $E_{nn} \sim 8e^{-\pi\sqrt{n}}$ along with the approximation errors of an (n,n) CF approximant, computed using `M` = 1000 Chebyshev coefficients. Note how close both curves are to each other, even though n is small and CF approximants are only *near-best*.

To illustrate the use of `cf` for more difficult piecewise smooth functions, let us try and approximate the function

$$f_2(x) = \int_{-1}^{x} \operatorname{sign}\left(\sin(10e^t)\right) dt.$$

A chebfun representation for this function and the $(7,7)$ CF approximant based on 70 Chebyshev coefficients are produced in Matlab with the commands

```
> x = chebfun('x');
> f2 = cumsum(sign(sin(10*exp(x))));
> [p,q] = cf(f2,7,7,70);
```

The function and its approximant are plotted in Figure 3. Using more than 70 Chebyshev coefficients yields no visible difference.

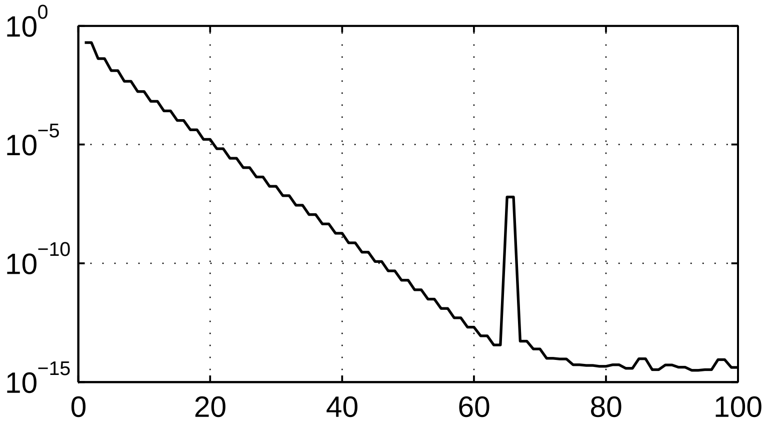

Figure 4: Error in $(n, 2)$ CF approximation to $f_3(x) = \tanh(10x)$.

In the next example we compute $(n, 2)$ approximants to

$$f_3(x) = \tanh(10x).$$

Figure 4 shows the approximation error for increasing values of n. This example clearly illustrates the block structure of the CF table. Since f_3 is an odd function, the $(n, 2)$ and $(n + 1, 2)$ approximants are equal when n is odd, hence the 'staircase' shape of the plot.

The strange artifact at $n = 64$ and $n = 65$ is due to ill-conditioning: `cf` writes out a warning message that is has detected this. From $n = 66$ on `cf` decides that f_3 looks close to rational and it uses Chebyshev-Padé approximation to compute the approximant. The difference between the (rational) approximant and the function f_3 is indeed close to machine precision. It might have been better to use Chebyshev-Padé also for $n = 64$ and $n = 65$, but detecting this automatically is an inherent difficulty of CF approximation close to machine epsilon.

Our final example is more complicated. For $\omega \ll 1$ and $-1 < c < 1$ define

$$p(x) = \frac{\pi}{\omega}(x^2 - c^2), \quad f_4(x) = \frac{xp(x)}{\sinh p(x)}\ .$$

This function has two sharp spikes at $x = \pm c$ caused by poles in the complex plane, and is difficult to approximate by polynomials. For $c = 0.6$ and $\omega = 0.02$ the chebfun representation of this function has length $M = 1682$. We compute a $(100, 10)$ CF approximation and plot the error as a function of x, together with the lines $\pm\lambda$ (the eigenvalue that measures the approximation

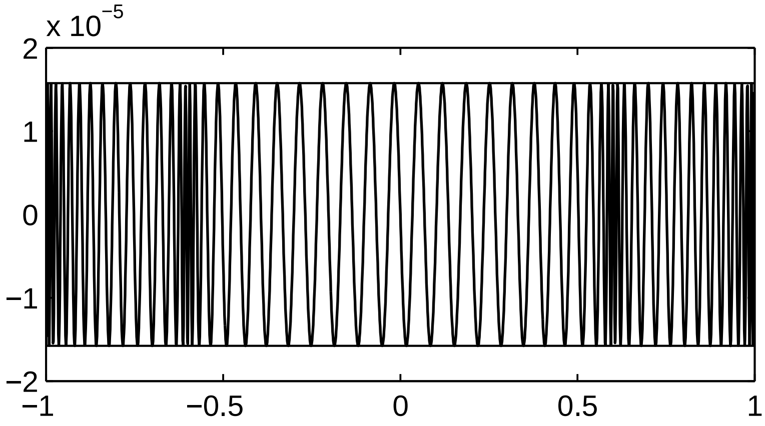

Figure 5: Error in $(100, 10)$ CF approximation to $f_4(x)$ for $c = 0.6$ and $\omega = 0.02$.

error). The result is shown in Figure 5. Note how the error equi-oscillates beautifully between the two bounds $\pm\lambda$.

References

[1] V. Adamjan, D. Arov, and M. Krein. Analytic properties of Schmidt pairs for a Hankel operator and the generalized Schur-Takagi problem. *Math. USSR Sb.*, 15:31–73, 1971.

[2] Zachary Battles and Lloyd N. Trefethen. An extension of MATLAB to continuous functions and operators. *SIAM J. Sci. Comput.*, 25(5):1743–1770, 2004.

[3] C. Carathéodory and L. Fejér. Über den Zusammenhang der Extremen von harmonischen Funktionen mit ihren Koeffizienten und über den Picard-Landauschen Satz. *Rend. Circ. Mat. Palermo*, 32:218–239, 1911.

[4] Martin H. Gutknecht. Rational Carathéodory-Fejér approximation on a disk, a circle, and an interval. *J. Approx. Theory*, 41(3):257–278, 1984.

[5] Martin H. Gutknecht and Lloyd N. Trefethen. Real polynomial Chebyshev approximation by the Carathéodory–Fejér method. *SIAM J. Numer. Anal.*, 19(2):358–371, 1982.

[6] Eric Hayashi, Lloyd N. Trefethen, and Martin H. Gutknecht. The CF table. *Constr. Approx.*, 6:195–223, 1990.

[7] Peter Henrici. Fast Fourier methods in computational complex analysis. *SIAM Rev.*, 21(4):481–527, 1979.

[8] G. Meinardus. *Approximation of functions: Theory and numerical methods.* Springer, 1967.

[9] D. J. Newman. Rational approximation to $|x|$. *Michigan Math. J.*, 11:11–14, 1964.

[10] Ricardo Pachón, Rodrigo B. Platte, and Lloyd N. Trefethen. Piecewise-smooth chebfuns. *IMA J. Numer. Anal.*, 2010. Advance Access published on July 28, 2009.

[11] T. Schmelzer and L.N. Trefethen. Computing the gamma function using contour integrals and rational approximations. *SIAM J. Numer. Anal.*, 45(2):558–571, 2007.

[12] I. Schur. Über Potenzreihen, die im Innern des Einheitskreises beschränkt sind. I. *J. Reine Angew. Math.*, 147:205–232, 1917.

[13] I. Schur. Über Potenzreihen, die im Innern des Einheitskreises beschränkt sind. II. *J. Reine Angew. Math.*, 148:122–145, 1918.

[14] Herbert Stahl. Uniform rational approximation of $|x|$. In *Methods of approximation theory in complex analysis and mathematical physics (Leningrad, 1991)*, volume 1550 of *Lecture Notes in Math.*, pages 110–130. Springer, Berlin, 1993.

[15] T. Takagi. On an algebraic problem related to an analytic theorem of Carathéodory and Fejér and on an allied theorem of Landau. *Japan J. Math.*, 1:83–93, 1924.

[16] T. Takagi. Remarks on an algebraic problem. *Japan J. Math.*, 2:13–17, 1925.

[17] Lloyd N. Trefethen. Near-circularity of the error curve in complex Chebyshev approximation. *J. Approx. Theory*, 31(4):344–367, 1981.

[18] Lloyd N. Trefethen. Rational Chebyshev approximation on the unit disk. *Numer. Math.*, 37(2):297–320, 1981.

[19] Lloyd N. Trefethen. Chebyshev approximation on the unit disk. In H. Werner et al., editor, *Computational aspects of complex analysis*, pages 309–323. D. Reidel Publishing Company, 1983.

[20] Lloyd 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, editors, *Rational Approximation and Interpolation*, volume 1105 of *Lecture Notes in Mathematics*. Springer-Verlag, 1984.

[21] Lloyd N. Trefethen. Matlab programs for CF approximation. In *Approximation Theory V*, pages 599–602. Academic Press, Boston, 1986.

[22] Lloyd N. Trefethen and Martin H. Gutknecht. The Carathéodory–Fejér method for real rational approximation. *SIAM J. Numer. Anal.*, 20(2):420–436, 1983.

[23] L.N. Trefethen, J.A.C. Weideman, and T. Schmelzer. Talbot quadratures and rational approximations. *BIT*, 46:653–670, 2006.

[24] R. S. Varga, A. Ruttan, and A. J. Carpenter. Numerical results on the best uniform rational approximations of the function $|x|$ on the interval $[-1, 1]$. *Math. USSR-Sb.*, 74(2):271–290, 1993.