A general Chebyshev complex function approximation procedure
and an application to beamforming

R. L. Streit and A. H. Nuttall
Naval Underwater Systems Center, New London Laboratory, New London, Connecticut 06320
(Received 22 December 1981; accepted for publication 24 March 1982)

A new computational technique is described for the Chebyshev, or minimax, approximation of a
given complex valued function by means of linear combinations of given complex valued basis
functions. The domain of definition of all functions can be any finite set whatever. Neither the
basis functions nor the function approximated need satisfy any special hypotheses beyond the
requirement that they be defined on a common domain. Theoretical upper and lower bounds on
the accuracy of the computed Chebyshev error are derived. These bounds permit both a priori and
a posteriori error assessments. Efforts to extend the method to functions whose domain of
definition is a continuum are discussed. An application is presented involving "re-shading" a 50-
element antenna array to minimize the effects of a 10% element failure rate, while maintaining
full steering capability and mainlobe beamwidth.

PACS numbers: 43.60.Gk, 43.30.Vh

LIST OF SYMBOLS

$\mathbf{f}$ the given complex valued function to be approximated

$h_1, \ldots, h_s$ the given basis functions; linear combinations of these functions are used to approximate $\mathbf{f}$

$Q_m$ the given finite point set; approximations to $\mathbf{f}$ are constructed on the $m$ elements of this set. (Ordinarily, $Q_m$ is a set of complex numbers; however, $Q_m$ can be any finite set on which $\mathbf{f}$ and $h_k$ are defined.)

$z_1, \ldots, z_m$ the elements of $Q_m$

$(a_1, \ldots, a_n) = a$ any vector of complex numbers used as coefficients of the basis functions $h_1, \ldots, h_s$

$e_n(z,a)$ the complex error “curve” of the approximation to $\mathbf{f}$ afforded by the coefficient vector $a$; defined by Eq. (A1)

$E_n(f)$ the actual maximum magnitude error committed by the “best” (i.e., Chebyshev, or minimax) approximation to $\mathbf{f}$ by linear combinations of $h_1, \ldots, h_s$; defined by Eq. (A2)

$\hat{a}$ any coefficient vector for which $E_n(f)$ is actually attained; $\hat{a}$ is essentially an approximation to the vector $a$ (see above)

$W_n(f)$ the maximum magnitude error committed using the coefficient vector $\hat{a}$; defined in Theorem A2

$\mathbf{a}(z), \mathbf{a}(z)$ the real and imaginary parts, respectively, of $\mathbf{f}(z)$

$r_k(z), s_k(z)$ the real and imaginary parts, respectively, of the basis function $h_k$ (see above)

$\mathbf{R}, \mathbf{S}$ real $m \times n$ matrices whose entries in the $j$th row and $k$th column are $r_k(z_j)$ and $s_k(z_j)$, respectively. Used to construct matrix $\mathbf{B}$ (see below)

$b_k, c_k$ the real and imaginary parts, respectively, of the coefficient $a_k$ of basis function $h_k$

$\mathbf{Bx} = \mathbf{g}$ the real overdetermined system on $mp$ equations in $2n$ unknowns, whose Chebyshev solution yields a solution $\hat{a}$ to the problem $M_{mp}(f)$; see the paragraph containing Eq. (A6) for details of construction

$\hat{\mathbf{Bx}} = \mathbf{g}$ analogous to $\mathbf{Bx} = \mathbf{g}$ when the solution vector $\hat{a}$ is forced to be a vector of real numbers; see Eq. (A7) for details

all notations not in this glossary are understood to be “local”; that is, they are used only in the context of the particular paragraphs which contain them
INTRODUCTION

The approximation of desired or given functional behavior by finite sets of simpler or specified basis functions is a recurrent problem in many fields. For example, in the mathematical field, we might wish to approximate a (desired) complex integral by a set of (simpler) sinusoidal components. In an antenna array processing application, we often want to realize a (given) low side-lobe behavior by means of an array with (specified) element locations which are not under our control.

For the case where the given functional behavior and the specified basis functions are all real valued and defined on a finite discrete data set, and where the approximation is afforded by a real-weighted linear combination of these basis functions, the optimum solution for minimizing the maximum magnitude error, i.e., the Chebyshev norm is in very good shape due to a fine algorithm given in Barrodale and Phillips. Specifically, this algorithm solves the following mathematical problem: given real constants \([f_i], [h_k]\)

where \(1 < i < m, 1 < k < n, m \geq n\), the real quantities \([a_i]\) are determined that minimize the maximum absolute value of the error residuals

\[ e_i = f_i - \sum_{k=1}^{n} a_k h_k, \quad \text{for} \quad 1 < i < m. \tag{1} \]

This algorithm has recently been used to good advantage in an antenna processing application to design some real symmetric weighting functions with very good side-lobe behavior, subject to constraints on the rate of decay of the distant side lobes.

Here we wish to employ the algorithm, as described above for real variables in Eq. (1), for the minimization of the Chebyshev norm of

\[ e_i(z_i) = |f(z_i) - \sum_{k=1}^{n} a_k h_k(z_i)|, \tag{2} \]

where \(f(z)\) and \([h_k(z)]\) are complex, and \(z\) can take values in an arbitrary finite discrete point set. The weighting coefficients \([a_i]\) may be complex, or alternatively, they may be restricted to be real. Applications are afforded by an antenna array with arbitrarily specified element locations, but employing weights that are restricted to be real, or alternatively by array weights that are also allowed to be phased (complex). Numerical examples and applications of the technique, some efforts attempted for extending the method to a continuum of values of \(z\), and a discussion constitute the rest of the main body of the paper. In the Appendix the basic mathematical theory and algorithm for the minimization of Eq. (2) is developed. Streit and Nuttall present a FORTRAN program in a form which should be useful to readers interested in applying this technique to their own particular applications; unfortunately the listing is too long to include here. A brief study of the appendix, especially with regard to Eq. (A6), should enable interested readers to write their own program.

Although the above algorithm is limited to a discrete set of points, it has been used fruitfully to minimize the continuous error [Eq. (2)] over a real variable \(z\) in the interval \([a, b]\), when \(f(z)\) and \([h_k]\) are real, in the following manner. First, an initial set of \(m \geq n\) real points \([z_i]\) was specified and the Chebyshev norm minimized in the usual fashion, resulting in the coefficient set \([a_i]\). For this set of optimum coefficients, the locations \([z_i]\) of the largest peaks of \([e_i(z)]\) were located, by setting the derivative \([e_i'(z)]\) to zero and solving numerically for \([z_i]\); the number \(l\) of such peaks will generally be less than \(m\), but larger than \(n\). This approach produces the availability of computable expressions for \([f'(z)]\) and \([h_k'(z)]\). Then the modified set of points \([z_i]\) were used for another Chebyshev minimization, resulting in coefficient set \([a_i]\). Repetition of this procedure stabilized after a few trials with a unique set of \([z_i]\); at which the maximum errors were equal and irreducible. In the examples tried in Nuttall, the number of peaks \(l\) at which the magnitude error \([e_i(z)]\) was largest and equal turned out to be \(n + 1\). Further discussion of this recursive approach is given in Sec. II.

Our method, as presented in the Appendix, is not inherently restricted to arrays of any particular geometry, but does assume that interelement effects (mutual coupling) can be ignored. In the most general case of a spatial or volumetric array, the method proposed here can still be applied. All the functions in Eq. (1) are then functions of spherical coordinates \((\theta, \phi)\), so the finite domain of approximation becomes an appropriately chosen finite set \((\theta_0, \phi_0)\) instead of a set of complex numbers. This difference does not in any way affect the mathematical properties of our method; rather it affects the size of the numerical problem to be solved and consequently, the computer effort required for its solution. For large enough arrays, such effort ultimately becomes prohibitive; where that point lies depends upon the designer and the application.

Although our method is applied only to single-frequency design problems for arrays, it can also be applied to broadband frequency design by sampling in frequency space as well. This again adds to the computer effort of solution, but does not affect the basic mathematical method.

We use no weighting function in Eq. (2), and so the resulting farfield beam patterns have a level side-lobe structure. For example, the classical Dolph--Chebyshev array design can be reproduced by our method. If such a level side-lobe structure is not desired, then use of an appropriate weighting function in Eq. (2) is easily incorporated into our method without altering the algorithm in any essential way.

I. APPLICATION TO ARRAY DESIGN WITH A CONSTRAINT

Consider a linear antenna array with \(N\) elements, located at arbitrary fixed positions \([x_i]\), receiving a plane-wave arrival of wavelength \(\lambda\) from direction \(\theta_\alpha = -\pi/2 < \theta < \pi/2\), relative to a normal to the array. If the array is steered to look in direction \(\theta_\beta = -\pi/2 < \theta < \pi/2\), then the complex transfer function of the beamformer is given by

\[ T(u) = \sum_{k=1}^{N} w_k \exp(-id_k u), \tag{3} \]

where \([w_k]\) are the element weights, and
\[ T(k) = 2 \pi x_k / \lambda, \quad \text{for } 1 \leq k \leq N, \]
\[ u = \sin \theta_k - \sin \theta_l. \]

Observe that the total range of \( u \) depends on the look direction \( \theta_l \); for example, if \( \theta_l = 0 \), then the range of \( u \) is the closed interval \([-1, 1]\). The peak response of \( T(u) \) should occur at \( u = 0 \), so we normalize without loss of generality according to

\[ T(0) = 1 = \sum_{k=1}^{N} w_k. \]

To realize small side lobes, we must minimize \( |T(u)| \) for all \( u \) values in some subset \( U \) of the total range of \( u \). For example, if \( \theta_l = 0 \), the total range of \( u \) is \([-1, 1]\), and \( U \) could be the union of intervals \([-1, -u_0] \) and \([u_0, 1]\), where \( u_0 > 0 \) is chosen small relative to 1. For the special case of real weights, since from Eq. (3), \( T(-u) = T^*(u) \), we could confine attention to \( U = [u_0, 1] \). The normalization constraint is most easily accounted for by solving for \( w_u \) and eliminating it; we obtain then

\[ T(u) = \exp(-idNu) - \sum_{k=1}^{N} w_k \exp(-id_k u). \]

This problem now fits the framework of Eq. (A.1) in the appendix if we identify

\[ z = u, \quad n = N - 1, \quad e(z; a) = T(u), \]
\[ f(z) = \exp(-id_k u), \quad a_k = w_k, \]
\[ h(z) = \exp(-id_k u) - \exp(-id_k u). \]
\[ Q_u = \text{finite subset of } U. \]

There has been no statement thus far as to the real or complex nature of the weights \( \{w_k\} \). This distinction depends upon the application and the capability of the beamformer. Both cases fit the above framework; the only difference is that the number of unknowns to be solved for will be twice as large for the complex weights as for the real weights.

If the array is half-wavelength equispaced, then the computed element weights will be identical to the classical Dolph–Chebyshev weights and can, in this instance, be computed analytically. The general case of arbitrary spacings, however, cannot be computed analytically; yet the algorithm presented in this paper can always be applied.

In the remainder of this section, we presume that the elements are equispaced at half-wavelength. Then \( x_k = k \lambda / 2 \) and Eq. (3) becomes

\[ T(u) = \sum_{k=1}^{N} w_k \exp(-iku). \]

Observe now that \( T(u) \) in Eq. (6) has period 2 in \( u \), regardless of whether the weights \( \{w_k\} \) are real or complex, or whether some elements have failed, i.e., zero weight values. This means that we can study and control \( T(u) \) in Eq. (6) over any convenient \( u \) interval of length 2, and need not confine our investigation to \([-1, 1]\). In particular, we concentrate on the \( u \) interval \([0, 2]\) in the following.

As an illustration of the capability of the minimization technique of this paper, a 50-element, half-wavelength, equispaced linear array was initially designed for peak side lobes of \(-30 \text{ dB} \) relative to the main peak. This is of course a standard Dolph–Chebyshev case, and gives \(-30 \text{ dB} \) side lobes throughout the \( u \) range \([u_0, 2 - u_0]\), where \( u_0 = 0.0538117 \). Then 10% of the elements were randomly eliminated from the array, but the remaining weights were unchanged; this corresponds to five elements failing in the array. The relative response of this particular array, with elements 7, 22, 40, 43, 50 failed, is illustrated in Fig. 1. The peak side lobe has increased from \(-30 \text{ to } -21.58 \text{ dB} \), a degradation of \(8.4 \text{ dB} \), and there is a large variety of different size peaks.

When our method with \( p = 2 \) and \( m = 251 \) is applied to this defective array and the remaining 45 elements are weighted with real coefficients, subject to the constraints that the mainlobe width be the same as the ideal 50-element array and that the steering range in \( u \) be the same, the resultant array pattern is as displayed in Fig. 2. The peak side lobe is now \(-23.62 \text{ dB} \), an improvement of \(2.04 \text{ dB} \) over Fig. 1; however, there is still a significant variation in the values of the side lobes due to an insufficient number of phase controls, namely only \( p = 2 \).

The best real weights resulting from an increase in the parameter values to \( p = 8 \) and \( m = 501 \) are displayed graphically in Fig. 3, and the corresponding array pattern is given in Fig. 4. The gaps in Fig. 3 at locations 7, 22, 40, 43, and 50 correspond to zero weighting at the failed elements. The general character of the weights is a bell-shaped one of all pos-

---

FIG. 1. Relative pattern for five elements failed

FIG. 2. Relative pattern for \( p = 2, m = 251 \), real weights.

R. L. Streit and A. H. Nuttall: Chebyshev approximation

---

always exists a set of real weights. Thus it is not necessary to use complex weights in the case of line arrays to achieve best possible side-lobe levels.)

The use of linear programming to design antenna arrays is not entirely new. In McMahon et al. and Wilson, linear programming was used to synthesize desired complex transfer functions to within 3 dB of the best possible side-lobe level. Their method corresponds identically to taking $p = 2$ in the method presented in this paper, i.e., treating only the real and imaginary parts of Eq. (2).

The computation of the real weights of Fig. 2 (where $p = 2$, $m = 251$, and $n = 44$) and of Fig. 4 (where $p = 8$, $m = 501$, and $n = 44$) required 1.2 min/205 simplex iterations and 38.4 min/402 simplex iterations, respectively. On the other hand, when the weights were allowed to be complex (replacing $n = 44$ by $n = 88$, but leaving $p$ and $m$ unchanged in both cases), the computations required 7.0 min/657 simplex iterations and 179 min/1262 simplex iterations, respectively. The two of these four cases requiring the smallest CPU times encountered almost no system overhead due to program size. However, the two cases requiring the largest CPU times encountered very significant system overhead because their large memory requirements caused significant usage of the virtual memory feature of the DEC VAX 11/780. The 38.4-min case required over 3.6 million page faults, while the 179-min case required over 11 million page faults. It is important to bear in mind that the DEC VAX 11/780 is essentially a minicomputer, and that without virtual memory, only the largest mainframe computers could have solved either of these two problems.

II. EFFORTS TO EXTEND THE METHOD

Our basic problem is to minimize the maximum magnitude of complex error

$$e_z(z; \phi) = f(z) - \sum_{k=1}^{n} a_k e_k(z)$$

over a continuum of values of $z$, when $f$, $\{e_k\}$, and $\{a_k\}$ are complex. We immediately approximate this desired problem by discretizing the $z$ variable to a finite number of values, in order to make the problem computable. Furthermore, at any $z$ value of interest, we additionally discretize the number of phase errors we are willing to consider. To be specific, since the algorithm in Barrodale et al. applies only to real quantities, we consider the "projection" of a rotated version of the complex error:

$$\rho_z(z; \phi) = \text{Re}\{\exp(\phi) e_z(z)\}.$$  

Then, since the argument of complex error [Eq. (7)] is unknown a priori, we let $\phi$ take on a finite set of values spread over any $\pi$ radian interval, and minimize the magnitude of projection [Eq. (8)] over all these selected $\phi$ values. This is equivalent to the method of the Appendix.

In an effort to eliminate this second discretization process in $\phi$, a perturbation method was put forth that claimed guaranteed convergence to the optimum weights for any given finite discrete set of $z$ values. When applied to the examples in Barrodale et al., the proposed perturbation technique did indeed converge. However, when applied to the

FIG. 3. Best real weights for $p = 8$, $m = 501$.

FIG. 4. Relative pattern for $p = 8$, $m = 501$.
following example, of approximation of \(\exp(i3x)\) by the three
basis functions \(1, \exp(ix), \exp(i2x)\), over 100 equispaced
points in the domain \([0, \pi/4]\) in \(x\), it sometimes failed to con-
verge, depending on the initial weights employed. The rea-
son for this failure is that the "direction of the minimum"
furnished by the perturbation is often totally irrelevant, and
the best scale factor to apply to this perturbation is very
small. Thus there occurs a small random meander in the
coefficient space, and occasional convergence to a nonopti-
mum point. A modification of this technique was attempted
wherein the magnitude of the perturbation was bounded.
Although this improved the situation somewhat, conver-
geance to the optimum was not always obtained.

It was thought that this meander in coefficient space
might be eliminated by tracking the exact \(z\) values at which
Eq. (7) is a maximum. Recall that in the real case discussed in
the Introduction, convergence to the absolute optimum over
a continuum of real \(z\) values was achieved in a practical ex-
ample by re-evaluating the \(z\) points of maximum error and
using these in a recursive approach. When this idea was ex-
tended to the two continuous variables \(z, \varphi\) in Eq. (8), and
only the \(2n + 1\) largest error points were retained, conver-
geance was not obtained. When, however, the single "point"
of a maximum, i.e., a pair of values \((z_k, \varphi_k)\), was replaced by a
"patch", i.e., a set of values \([z_{ik}, \varphi_{ik}]\) covering the maxi-
mum point \((z_k, \varphi_k)\), the convergence to the absolute opti-
mum for the examples considered was apparently achieved.
The patch width in \(\varphi\) was of the order of a degree in most
cases. The problem with this latter modification is that a
large number of computations of the error function and its
derivative must be evaluated, and the improvement over the
method of the appendix is insignificant when \(p\) is large.

If the final error in Eq. (7), after application of the meth-
ood of the Appendix, is inadequate due to inadequate sampling
in \(z\) and/or \(\varphi\), it is possible, for a given coefficient set \([a_k]\),
to locate the point \((z_k, \varphi_k)\) at which Eq. (8) is largest, and then
use a gradient approach to decrease this maximum error at
\((z_k, \varphi_k)\). Of course, the particular point of maximum will
jump around as the set \([a_k]\) is perturbed; nevertheless, the
technique does converge (although slowly) and does lead to
smaller errors at the maximum of Eq. (8) in a continuum for \(z\)
and \(\varphi\).

III. DISCUSSION AND SUMMARY

It has been observed that two of the locations of maxi-
mum magnitude error often occur at the endpoints, if the
specified domain in Eq. (2) is a real interval. (For example,
see Figs. A1 and A2. The example of real coefficients in Fig.
A1 had one of the maximum error points at one endpoint,
but not the other. However, if we had specified domain
\([-\pi/4, \pi/4]\) in that example, we would have observed four
peak-error points, two of which would have been at end-
points, due to the conjugate property of the desired function
and the basis functions.) Since the endpoints may be the only
ones we can anticipate a priori and specify as locations of
maximum error, an obviously useful procedure is to use
more values of phase shift \(\varphi\) in Eq. (8) [alternatively, the
angles \(\theta_{j}\) in Lemma A2] at the endpoints than in the inte-
rior, so as to better control these very likely locations of
maximum error. For example, we might use \(p = 6\) in the
interior of a specified real interval domain of \(x\) and use
\(p = 12\) or 20 at the two endpoints. This does not add greatly
to the total computation, since there are generally far more
interior points than (two) endpoints. The program in Streit
and Nuttall may be readily used with different values of \(p\)
at different data points.

The \(p\) different phase shifts \(\psi\) selected in Eq. (8) have
been chosen here to be equally spaced over a 180° span (along
with their 180° mates). This is the most reasonable selection
in the absence of a priori knowledge of the complex error
magnitude and phase because it gives the best upper bound
in Lemma A2 of any set of phases. However, one could select
any value of \(\psi\) to investigate the error; for example, different
sets of values of \(\psi\) could be used at various values of abscissa
\(z\). The program in Streit and Nuttall may be used with any
desired set of phases at any, or all, of the data points.

The potential for significant round-off error accumula-
tion is always present in linear Chebyshev complex function
approximation. For example, in approximating
\(f(x) = \cos(12x) + i \sin(3x)\) by a complex linear combination
of the 12 basis functions \(1, \exp(i\omega), \exp(i2\omega)\) on the interval
\([0, \pi/4]\), the complex coefficients of best approximation were
observed to be large in magnitude and to lie in all quadrants
of the complex plane; therefore significant numerical round-
off error occurred during computation of the residuals with-
in algorithm ACM495. Even if the coefficients of best ap-
proximation had happened to be better behaved, serious
cancellation error may still occur in some problems because
of the very nature of complex arithmetic. It might, therefore,
be wise to use a double precision version of algorithm
ACM495 routinely in complex Chebyshev approximation
problems to alleviate such cancellation errors.

A sensitivity analysis on the optimum coefficients may
be in order in some applications to determine their utility.
This consideration is completely independent of their nu-
merical accuracy. For example, in an antenna array design
problem where some elements are spaced significantly less
than a half-wavelength apart, it might well turn out that the
optimum coefficients need to be specified with a relative er-
rors of better than \(10^{-6}\). Then, although the mathematical
results may be correct and accurate, practical usage is pre-
cluded. This sensitivity can be determined by perturbing the
optimum weights a few percent and observing if a drastic
change occurs on the desired side-lobe behavior. (Such ar-
rays are referred to as super-directive arrays.)

APPENDIX: MATHEMATICAL THEORY AND
ALGORITHM

Let \(f\) and \(h_1, \ldots, h_n\) be complex valued functions defined
on the finite discrete point set \(Q_n = \{z_1, \ldots, z_n\}\). For a com-
plex vector \(a = [a_1, \ldots, a_n] \in \mathbb{C}^n\), define the complex error
function
\[ f(x) = \sum_{k=1}^{n} a_k h_k(x) = e_k(x), \quad x \in Q_n. \]  \hspace{1cm} (A1)

The discrete linear Chebyshev approximation problem is to
find a complex vector \(\hat{a} = [\hat{a}_1, \ldots, \hat{a}_n] \in \mathbb{C}^n\) so that

The quantity \( E_\sigma(f) \) is called the discrete Chebyshev, or minimal, error of the approximation on the point set \( Q_m \). The restriction of \( \delta \) to real values is discussed below.

We do not solve this problem exactly. An algorithm presented in Barrodale et al.\(^9\) for its solution is erroneous; we have discovered examples (see Sec. II) such that the recursive \( M_\sigma(f) \) need not converge to a solution of Eq. (A2). The error in these approximate solutions to Eq. (A2) can be determined and, in fact, made arbitrarily small, using the results we prove below; see Theorems A1 and A2.

It can be shown by standard mathematical methods\(^10\) that a vector \( \delta \) satisfying Eq. (A2) exists, although it may not be unique. Sufficient conditions are known that result in unique \( \delta \), but we do not need these conditions here. Therefore no further assumptions on \( f, h, a, \) or the point set \( Q_m \) are made. In order to proceed, we need the following results.

Proofs of all these results are given in Streit and Nuttall.\(^4\)

Lemma A1. If \( z = x + iy \), where \( x \) and \( y \) are real, then

\[
|z| = \max_{-\infty < \theta < \infty} (x \cos \theta + y \sin \theta).
\]

Lemma A2. Let \( \theta_j = \pi(j - 1)/p, j = 1, 2, \ldots, 2p \), where \( p \geq 2 \), and let \( \theta_j = \pi/2 + j \pi/2p \), then

\[
M = \max_{j=1}^{2p} (x \cos \theta_j + y \sin \theta_j).
\]

Then

\[
M < |z| < M \sec \left( \pi/(2p) \right).
\]

We are now in a position to describe a problem that we can solve exactly and that is related to the given discrete linear Chebyshev approximation problem (A2). Let the real and imaginary parts of the complex error \( e_\sigma(x, \alpha) \) be denoted by \( R_\sigma(x, \alpha) \) and \( I_\sigma(x, \alpha) \), respectively. For notational convenience, we define, for any complex vector \( \alpha \in \mathbb{C}^n \),

\[
G_j(x, \alpha) = R_j(x, \alpha) \cos \theta_j + I_j(x, \alpha) \sin \theta_j, \quad j = 1, \ldots, 2p,
\]

where \( \theta_0 = \theta_2p = \pi \). The \( \theta_j \) are the angles given explicitly in Lemma A2. We seek a complex vector \( \delta = (\delta_1, \ldots, \delta_n) \in \mathbb{C}^n \) satisfying

\[
M_\sigma(f) = \min_{\alpha \in \mathbb{C}^n} \max_{1 \leq j \leq 2p} \max_{-\infty < \delta_j < \infty} G_j(x, \alpha).
\]

With standard mathematical methods, it is easy to see that at least one such vector \( \delta \in \mathbb{C}^n \) exists. The connection between the problem (A4) and the problem (A2) is explored in the next few results.

Theorem A1. Let \( p \geq 2 \) be an integer, and let \( \theta_j = \pi(j - 1)/p, j = 1, 2, \ldots, 2p \). Then

\[
M_\sigma(f) < E_\sigma(f) \leq M_\sigma(f) \sec \left( \pi/(2p) \right)
\]

Theorem A2. Let \( p \geq 2 \) be an integer, and let \( \theta_j = \pi(j - 1)/p, j = 1, 2, \ldots, 2p \). Let

\[
0 < E_\sigma(f) - E_\sigma(f) < E_\sigma(f) \sec \left( \pi/(2p) \right) - 1
\]

where the complex vector \( \delta \in \mathbb{C}^n \) is any vector satisfying (A4).

Then

\[
E_\sigma(f) < E_\sigma(f) < E_\sigma(f) \sec \left( \pi/(2p) \right).
\]

Corollary A2.1. Under the conditions of Theorem A2,

\[
M_\sigma(f) < E_\sigma(f) < M_\sigma(f) \sec \left( \pi/(2p) \right)
\]

The preceding corollary evidently gives excellent upper and lower bounds on the discrete linear Chebyshev approximation error \( E_\sigma(f) \), and these bounds are readily available after the numerical computation of \( \delta \in \mathbb{C}^n \) and \( M_\sigma(f) \) has been completed. We point out that the above two theorems substantially generalize results in Barrodale et al.,\(^9\) p. 854.

Using the Maclaurin series for \( \sec x \) in Theorem A2 gives the relative discrepancy

\[
0 < \frac{E_\sigma(f) - E_\sigma(f)}{E_\sigma(f)} < \sec \left( \pi/(2p) \right) - 1
\]

Note that this upper bound on the relative error is independent of \( f \), the point set \( Q_m \), the basis functions \( \{ h_k \} \), and \( n \). We will now explicitly formulate an overdetermined system of real linear equations to be solved in the Chebyshev norm (to be defined) which is equivalent to solving the problem (A4). Referring to the choice of \( \theta_j \)'s in Lemma A2, we observe that \( \theta_{j+1} = \pi + \theta_j \), \( j = 1, \ldots, p \), and so, from Eq. (A3), we have

\[
G_j(x, \alpha) = G_{p+j}(x, \alpha), \quad j = 1, \ldots, p.
\]

Therefore, we may rewrite Eq. (A4) as

\[
M_\sigma(f) = \min_{\alpha \in \mathbb{C}^n} \max_{1 \leq j \leq 2p} |G_j(x, \alpha)|.
\]

Now, breaking the following quantities into their real and imaginary components

\[
f(x) = u(x) + iv(x),
\]

\[
h_k(x) = r_k(x) + ir_k(x),
\]

\[
\alpha_k = b_k + ic_k,
\]

we may write

\[
R_k(x, \alpha) = u(x) - \sum_{k=1}^{n} b_k r_k(x) + \sum_{k=1}^{n} c_k r_k(x),
\]

\[
I_k(x, \alpha) = v(x) - \sum_{k=1}^{n} b_k s_k(x) - \sum_{k=1}^{n} c_k r_k(x),
\]

\[
G_j(x, \alpha) = u(x) \cos \theta_j + iv(x) \sin \theta_j
\]

\[
- \sum_{k=1}^{n} b_k \, r_k(x) \cos \theta_j + s_k(x) \sin \theta_j
\]

\[
- \sum_{k=1}^{n} c_k \, r_k(x) \sin \theta_j - s_k(x) \cos \theta_j.
\]

Note that \( G_j(x, \alpha) \) is a real linear equation in the 2n variables \( \{ b_k \} \) and \( \{ c_k \} \), and that all the coefficients of this equation are computable directly from known data.

Define the \( m = 2n \times m \) real matrix \( B \) in the partitioned form
G_j(z;a) arranged in a special order. Therefore the problem (A5) can be solved by computing a solution to the overdetermined linear system (A6) in the Chebyshev norm; i.e., the largest magnitude component of the residual vector g - Bx is minimized over all choices of the vector x.

This equivalent problem in linear algebra can, in principle, be solved exactly and in a finite number of steps using linear programming methods. Solutions of Eq. (A6) are not required to be unique; every solution of Eq. (A6) is a solution of Eq. (A5).

An excellent algorithm, which we will refer to as ACM 495, is available in the literature for solving the overdetermined system of equations Ax = b. A linear program is set up and solved by the algorithm, so that knowledge of linear programming techniques is not necessary to use the algorithm in practice. The computational procedure, internal to the algorithm, actually solves the dual of the primal linear program using a modification of the simplex method. The dual formulation of this problem is available. We will not discuss the details of the linear programming technique in this paper.

A very simple modification of ACM 495 yields an algorithm for solving any real overdetermined system of linear equations in the Chebyshev norm subject to the additional constraints that all the residuals be non-negative. For a general system Ax = b, this problem takes the form

\[
\text{minimize } \max \left( b_j - \sum_{k=1}^{r} \alpha_k x_k \right)
\]

subject to the r constraints

\[
b_j - \sum_{k=1}^{r} \alpha_k x_k \geq 0, \quad j = 1, \ldots, r.
\]

The solution \(x_1, \ldots, x_r\), returned by this modified algorithm is correct, even though the residuals returned may be in error. The correct residuals, if desired, must be calculated directly from the solution. Alternatively, if the residuals are required to be non-positive, then the same modified algorithm will work with A and b replaced by \(-A\) and \(-b\), respectively.

Requiring non-negative residuals in the overdetermined system (A6) has interesting geometrical interpretations. For example, if we take \(p = 2\) in Lemma A2, then \(\theta_1 = 0\) and \(\theta_2 = \pi/2\). Thus \(G(z;a)\) and \(G(z;2a)\) are merely the real and imaginary parts of the complex error \(e_j(z;a)\), and the 2m components of the residual vector \(g = Bx\) are precisely the real and imaginary parts of \(e_j(z;a)\) evaluated at all m data points. Therefore, if the system (A6) is required to have non-negative residuals, we have forced the error curve to lie entirely in the first quadrant of the complex plane. More generally, we may always constrain \(e_j(z;a)\) to lie in a given convex wedge-shaped sector of the complex plane with vertex at the origin, by making different, but appropriate, choices of the angles \(\theta_1\) and \(\theta_2\).

Suppose, finally, that the complex solution vector \(aeC^m\) of problem (A4) is required to be strictly real, while f and \([h_k]\) are complex. Then, in the vector x of Eq. (A6), \(c_1 = \ldots = c_m = 0\). Thus the overdetermined system \(Bx = g\) of \(m p\) equations in 2n unknowns can be replaced by a smaller system \(Bx = g\) of \(m p\) equations in only n unknowns,
TABLE AI. Coefficients for the real weight case.  

<table>
<thead>
<tr>
<th>m</th>
<th>p</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>2</td>
<td>0.364737</td>
<td>0.954343</td>
<td>2.518388</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.320404</td>
<td>0.123951</td>
<td>2.396455</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>0.385547</td>
<td>2.518185</td>
<td>2.425096</td>
</tr>
<tr>
<td></td>
<td>54</td>
<td>0.844146</td>
<td>2.301461</td>
<td>2.410611</td>
</tr>
<tr>
<td>101</td>
<td>2</td>
<td>0.367671</td>
<td>0.942223</td>
<td>2.518458</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.321214</td>
<td>2.395252</td>
<td>2.425033</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>0.365131</td>
<td>2.423033</td>
<td>2.425006</td>
</tr>
<tr>
<td></td>
<td>54</td>
<td>0.853823</td>
<td>2.315031</td>
<td>2.425006</td>
</tr>
</tbody>
</table>

TABLE AII. Coefficients for the complex weight case.  

<table>
<thead>
<tr>
<th>m</th>
<th>p</th>
<th>$a_{1+}$</th>
<th>$a_{1-}$</th>
<th>$a_{2+}$</th>
<th>$a_{2-}$</th>
<th>$a_{3+}$</th>
<th>$a_{3-}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>2</td>
<td>0.364737</td>
<td>0.954343</td>
<td>2.518388</td>
<td>2.518388</td>
<td>2.518388</td>
<td>2.518388</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.320404</td>
<td>0.123951</td>
<td>2.396455</td>
<td>2.396455</td>
<td>2.396455</td>
<td>2.396455</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>0.385547</td>
<td>2.518185</td>
<td>2.425096</td>
<td>2.425096</td>
<td>2.425096</td>
<td>2.425096</td>
</tr>
<tr>
<td></td>
<td>54</td>
<td>0.844146</td>
<td>2.301461</td>
<td>2.410611</td>
<td>2.410611</td>
<td>2.410611</td>
<td>2.410611</td>
</tr>
<tr>
<td>101</td>
<td>2</td>
<td>0.367671</td>
<td>0.942223</td>
<td>2.518458</td>
<td>2.518458</td>
<td>2.518458</td>
<td>2.518458</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.321214</td>
<td>2.395252</td>
<td>2.425033</td>
<td>2.425033</td>
<td>2.425033</td>
<td>2.425033</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>0.365131</td>
<td>2.423033</td>
<td>2.425006</td>
<td>2.425006</td>
<td>2.425006</td>
<td>2.425006</td>
</tr>
<tr>
<td></td>
<td>54</td>
<td>0.853823</td>
<td>2.315031</td>
<td>2.425006</td>
<td>2.425006</td>
<td>2.425006</td>
<td>2.425006</td>
</tr>
</tbody>
</table>

where the $m \times n$ real matrix $\hat{B}$ is defined in partitioned form by

$$
\hat{B} = \begin{pmatrix} B_{11} & \cdots & B_{1n} \\ \vdots & \ddots & \vdots \\ B_{m1} & \cdots & B_{mn} \end{pmatrix}
$$

(11)

where the $m \times n$ submatrices $B_{11}, \ldots, B_{mn}$, are unchanged from (6), and the real vector $\hat{x} = [b_1, \ldots, b_n]^T$. A solution of $\hat{B} \hat{x} = \hat{g}$ in the Chebyshev norm can be computed using linear programming and algorithm ACM 495 as before.

We illustrate the procedure by approximating the complex function $f(x) = \exp(i3x)$ by a weighted sum of the basis functions $\exp(x), \exp(-x), \exp(1x)$. That is, we seek to minimize the magnitude of the complex error curve

$$
e_{j}(x) = \exp(i3x) - \sum_{k=1}^{m} a_j \exp((k - 1)x)
$$

(12)

over interval $[0, \pi/4]$, by choice of $a_1, a_2, a_3$, by solving the problem $M_x(f)$ of Eq. (A4). Two cases are of interest; in the first, the coefficients $|a_j|$ are restricted to be real, whereas in the second, these coefficients can be complex. The number $m$, of equispaced $x$ values at which Eq. (A8) is sampled, is taken to be either 11, 101, or 1001, thereby ensuring that the smaller sample sizes are subsets of the larger sizes. The value of $p$, which is half the number of phase-shifted values of Eq. (A8) employed in the error minimization, is taken to be 2, 6, 18, 54, again ensuring the subset behavior of the smaller size cases. Note that $p$ and the phase shifts $\theta_j$ are as given in Theorem A1.

The optimum real coefficients in Eq. (A8) for the problem $M_x(f)$ are given in Table A1 for these choices of $m$ and $p$ and a plot of the magnitude of the error for several representative cases is given in Fig. A1. The best approximation of all cases considered is afforded by $m = 1001, p = 54$, and its error curve is plotted as a solid line; its maximum error is 0.1078, which is realized at two points in the interval $[0, \pi/4]$. The cases for smaller $m$ (less sampling of the abscissa) and smaller $p$ (less sampling of the phase of the complex error) are poorer; for example, the maximum error for $m = 11, p = 2$ is 0.1184, realized at only one point, namely $x = \pi/4$.

We have not plotted the other error curves with real coefficients for $m = 101$ and 1001, because they are indistinguishable from Fig. A1, as a perusal of Table A1 shows. For example, the coefficients for $m = 11, p = 2$ are very close to those for $m = 101, p = 2$ and $m = 1001, p = 2$. Thus our sampling in $x$ is already "fine enough" at $m = 11$. However, there is a significant change in the coefficients as $p$ is varied, for a fixed value of $m$; that is, $p = 2$ yields very coarse phase sampling of the error curve and should definitely be made larger.

The Chebyshev error curve ($m = 1001, p = 54$) in Fig. A1 realizes its maximum value at only $n + 1$ points, rather than at $n + 1$ points, where $n = 3$ is the number of coefficients for this example. This is probably related to the fact that we have minimized both the real and imaginary parts of the complex error, but have allowed ourselves to use only real coefficients.

The solution of the problem $M_x(f)$ for complex weights is given in Table AII for the same choices of $m$ and $p$ as above. Again, the change in coefficient values is more marked with $p$ than with $m$. Magnitude-error curves for $m = 11$ and 101 are given in Figs. A2 and A3, respectively; the curves for $m = 1001$ are indistinguishable from those for $m = 101$ and are not presented.

The Chebyshev error curve ($m = 1001, p = 54$) is now symmetric about the midpoint of the interval of interest and has four equal error peaks of value 0.0147. This error is 7.3 times smaller than that for the real coefficient case. Also, the number of equal error peaks now equals 1 plus the number of coefficients; whether this property holds generally is not known.
Upper and lower bounds on the discrete Chebyshev error $E_n(f)$ for the real and complex coefficient cases are given in Table AIII. These bounds are precisely those presented in Corollary A2.1. They correspond to sampling the complex error (A8) both in the abscissa $x$ and in the phase of $e_j(x)$. The lower bounds monotonically increase with increasing $m$ or $p$. The upper bounds decrease with increasing $p$, but increase with increasing $m$. All these trends follow from the fact that smaller sample sizes are subsets of the larger sizes.

However, the maximum magnitude error, evaluated over the continuum of $x$ values in the interval $[0, \pi/4]$ (actually computed on a dense discrete sampling space), obeys none of these monotonic relations, as Table AIV demonstrates. For example, the maximum error in the real case for $m = 11$, $p = 18$ is less than that for $m = 11$, $p = 54$. Also, the maximum error in the complex case for $m = 11$, $p = 6$ is greater than that for $m = 101$, $p = 6$. The reason for this behavior is that we have minimized a discrete approximation to our problem of interest, sampling both in the abscissa $x$ and in the phase values of the complex error. However, the numerical discrepancies are small, as they must be for reasonably fine sampling in both variables. (A recursive gradient procedure could be used with any of these coefficient sets to improve the final maximum magnitude error if desired.)

Efficiency and timing estimates for actual calculation of complex Chebyshev approximations by the method of this paper is an important consideration in some applications. If we define an operation as consisting of a multiplication followed by an addition, then it is known that the number of operations per simplex iteration required by algorithm ACM 495 is exactly the number of equations times the number of unknowns. In our case, the number of equations is $m p$, and the number of unknowns is $2n$ if the coefficients are complex, or $n$ if the coefficients are required to be real. Thus the operation count per iteration is either $2nmp$ or $nmp$. The number of iterations required is difficult to estimate, since it depends on the particular problem. However, in randomly generated problems, it has been observed that the number of iterations, $I$, is approximately the number of unknowns times some small constant $c$, where usually $1 < c < 3$. (Similar estimates have been observed in more general linear programs as well.) Thus, in our case, $I = cn$ if the coefficients are complex and $I = cn$ if they are real.

The CPU time should be proportional to the total operation count, which equals the product of the number of iterations and the number of operations per iteration. That is, we expect the CPU time to be proportional to $nm p$. For the particular example here, however, we obtain an excellent fit to the limited data in Table AV with the equation

### Table AIV. Maximum magnitude error, computed over 2001 equispaced points in $[0, \pi/4]$

<table>
<thead>
<tr>
<th>$m$</th>
<th>$p$</th>
<th>Real coefficients</th>
<th>Complex coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Lower bound</td>
<td>Upper bound</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>0.083718</td>
<td>0.118396</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.105074</td>
<td>0.107890</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>0.107307</td>
<td>0.107717</td>
</tr>
<tr>
<td></td>
<td>54</td>
<td>0.107612</td>
<td>0.107658</td>
</tr>
<tr>
<td>101</td>
<td>2</td>
<td>0.083731</td>
<td>0.118414</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.105192</td>
<td>0.108993</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>0.107556</td>
<td>0.107967</td>
</tr>
<tr>
<td></td>
<td>54</td>
<td>0.107767</td>
<td>0.107813</td>
</tr>
<tr>
<td>1001</td>
<td>2</td>
<td>0.083734</td>
<td>0.113418</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.105191</td>
<td>0.108991</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>0.107565</td>
<td>0.107976</td>
</tr>
<tr>
<td></td>
<td>54</td>
<td>0.107775</td>
<td>0.107821</td>
</tr>
<tr>
<td>$m$</td>
<td>$p$</td>
<td>Real coefficients</td>
<td>Complex coefficients</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Simplex CPU(s)</td>
<td>Simplex CPU(s)</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>0.02</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.08</td>
<td>0.16</td>
</tr>
<tr>
<td>18</td>
<td>11</td>
<td>0.23</td>
<td>0.58</td>
</tr>
<tr>
<td>54</td>
<td>13</td>
<td>0.81</td>
<td>2.25</td>
</tr>
<tr>
<td>101</td>
<td>11</td>
<td>0.20</td>
<td>0.40</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.73</td>
<td>1.60</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>2.65</td>
<td>5.78</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>11.39</td>
<td>24.27</td>
</tr>
<tr>
<td>1001</td>
<td>13</td>
<td>9.35</td>
<td>5.00</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>10.34</td>
<td>19.38</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>48.16</td>
<td>105.47</td>
</tr>
</tbody>
</table>

CPU time $= 0.128 \ n^{1.13} m^{1.18} p^{1.18}$

where $n = 6$ if the coefficients are complex, and $n = 3$ if they are real. This fit was obtained by letting the exponents of $n$, $m$, and $p$ vary separately. Other examples, however, lead us to anticipate that, more generally,

CPU time $\propto n^2 (m^3 p)^{1/3}$,

with a proportionality factor of the order of 0.01 - 0.03 ms, where $n$ is either twice the number of approximation coefficients if the coefficients are complex, or exactly the number of coefficients if they are required to be real.

The CPU time estimates apply, of course, only to the DEC VAX 11/780 computer on which the calculations were performed. The virtual memory feature of this system allows very large problems to be solved; however, for sufficiently large problems, the system overhead incurred (page faulting, and so on) may significantly and adversely affect these estimates.

One method of detecting the presence of significant round-off errors is supplied by the nature of the approximation problem itself. That is, it can be proven that

$$M_\alpha(f) < 1 < M_\beta(f) < M_\gamma(f) sec(\pi/2p)$$

Once $M_\alpha(f)$ and the coefficients have been computed in algorithm ACM 495, these bounds may be checked to see if significant numerical round-off error has occurred. In example [A8] above ($p = 6, m = 101$, complex coefficients), these inequalities were observed numerically to hold to five (but not six) significant digits. We conclude that the effects of round-off errors, although visible in the results, are not significant in this example. (Single precision numbers on the DEC VAX 11/780 have approximately seven significant decimal digits.)