MATHEMATICS OF COMPUTATION
VOLUME 51, NUMBER 184
OCTOBER 1988, PAGES 721-739

A Fast Algorithm for Linear Complex Chebyshev Approximations*

By Ping Tak Peter Tang

Abstract. We propose a new algorithm for finding best minimax polynomial approximations in the complex plane. The algorithm is the first satisfactory generalization of the well-known Remez algorithm for real approximations. Among all available algorithms, ours is the only quadratically convergent one. Numerical examples are presented to illustrate rapid convergence.

1. Introduction. Given a complex function F analytic on a specified domain in the complex plane, how can one construct the polynomial, of prescribed degree, that best approximates F in the Chebyshev (minimax) sense? Applications for Chebyshev polynomial approximations include

1. approximate numerical conformal mapping [11], [20], [22],
2. discrete antenna problems [24],
3. matrix iterative processes [6], and
4. system and control theory and digital filter design [12], [7].

In the past, complex Chebyshev polynomial approximation has been far less well understood than its real analogue. In particular, the quadratically convergent Remez algorithm ([3], [21]) for real approximation has not been satisfactorily generalized to complex approximation. Although a few generalized Remez algorithms have been proposed, some do not always converge and none converges quadratically. One difficulty in the generalization is that a major step in the iterative Remez algorithm is solving a best approximation problem on a small, finite set of points. While in real approximation the correct small number of points is known, and the solution to that subproblem is readily obtainable, neither of these happens in the complex case.

We have developed a way to choose a finite set of points together with another set of parameters (angles associated with the points) for which a best complex approximation subproblem is easily solvable. This generalizes the corresponding subproblem in the Remez algorithm for real approximations. The two sections following this introduction explain our algorithm.

Carrying the generalization further, we have also proved that our Remez algorithm for complex approximation converges quadratically under a certain condition similar to the corresponding one for real approximation [27], namely, that the optimal error graph has a sufficient number of alternations of sign. Our proof turns

Received September 30, 1987; revised February 11, 1988.
1980 *Mathematics Subject Classification* (1985 *Revision*). Primary 30E10, 41A10, 41A50; Secondary 65D15, 65K05.
*This work was supported in part by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, under contract W-31-109-Eng-38.

out to be similar to the one for real approximation. Statements of our results on convergence are given in Sections 4 and 5.

At this point, an interesting difference between real and complex approximation emerges naturally. Whereas the condition guaranteeing quadratic convergence is almost always satisfied for real approximation, we do not know yet how often it is satisfied for complex approximation. Discussions on related issues are presented in Section 6. Fortunately, computational experience shows that a weaker form of the condition is almost always satisfied, though no theoretical proof exists. Hence, we have extended our Remez algorithm to converge quadratically under that weaker condition. This is the subject of Section 7.

In Section 8, we compare our algorithm with two others recently proposed for the same problem. We conclude the paper by discussing a few applications and summarizing our results.

2. Formulation. Since the given function F is analytic, its best polynomial approximation on the given domain is identical to its best approximation on the domain's boundary. We will assume that this boundary is the range of a periodic function with domain [0,1]. For simplicity's sake, we further assume that the periodic function is smooth (cf. the discussion in Section 6). Since our algorithm also works for linear approximations with nonpolynomial basis functions, we will reformulate our problem in terms of a set of general basis functions.

Let f, $\varphi_1, \varphi_2, \ldots, \varphi_n$ be given smooth functions mapping the unit interval $[0,1]$ into the complex plane C.

Problem P. Find n real parameters $\lambda_1^*, \lambda_2^*, \ldots, \lambda_n^*$ such that

$$\max_{t\in[0,1]} \left| f(t) - \sum_{l=1}^{n} \lambda_l^* \varphi_l(t) \right| \le \max_{t\in[0,1]} \left| f(t) - \sum_{l=1}^{n} \lambda_l \varphi_l(t) \right|$$

for all $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_n]^T$ in $\mathbb{R}^n$.

To compress the notation, we define the function

$$E(\boldsymbol{\lambda}, \cdot) := f(\cdot) - p(\cdot), \qquad p(\cdot) = \sum_{l=1}^{n} \lambda_l \varphi_l(\cdot)$$

and use $\| \quad \|$ to denote the maximum norm taken over the interval [0,1]:

Problem P. Find $\boldsymbol{\lambda}^* \in \mathbb{R}^n$ such that

$$\|E(\boldsymbol{\lambda}^*, \cdot)\| \le \|E(\boldsymbol{\lambda}, \cdot)\| \quad \text{for all } \boldsymbol{\lambda} \in \mathbb{R}^n.$$

A simple example will be illustrative. Suppose we want to best approximate a function $F(z)$ over the unit disk by a linear polynomial in z with complex coefficients; then we can define $f(t) := F(e^{i2\pi t})$, $n := 4$, $\varphi_1(t) := 1$, $\varphi_2(t) := e^{i2\pi t}$, $\varphi_3(t) := i$, and $\varphi_4(t) := ie^{i2\pi t}$. Consequently, if solving Problem P yields λ_1^*, λ_2^*, λ_3^*, and λ_4^* as the optimal real parameters, the desired best polynomial approximation to $F(z)$ will be $(\lambda_1^* + i\lambda_3^*) + (\lambda_2^* + i\lambda_4^*)z$.

Our first step towards solving Problem P is to examine its dual (cf. [23]).

Problem D. Find $n+1$ points $t_1, t_2, \ldots, t_{n+1} \in [0,1]$, $n+1$ angles $\alpha_1, \alpha_2, \ldots,$ $\alpha_{n+1} \in [0, 2\pi]$, and $n+1$ nonnegative weights $r_1, r_2, \ldots, r_{n+1} \in [0,1]$ so as to maximize the inner product

$$\sum_{j=1}^{n+1} r_j \operatorname{Re}(f(t_j)e^{-i\alpha_j})$$

subject to the constraints

$$\sum_{j=1}^{n+1} r_j = 1$$

and

$$\sum_{j=1}^{n} r_j \operatorname{Re}(\varphi_l(t_j)e^{-i\alpha_j}) = 0 \quad \text{for } l = 1, 2, \ldots, n.$$

Problem D can be restated in more compact notation:

Problem D. Find $\mathbf{t} \in [0,1]^{n+1}$, $\boldsymbol{\alpha} \in [0, 2\pi]^{n+1}$, and $\mathbf{r} \in [0,1]^{n+1}$ so as to maximize

$$h = \mathbf{c}(\mathbf{t}, \boldsymbol{\alpha})^T \mathbf{r}$$

subject to the constraints

$$A(\mathbf{t}, \boldsymbol{\alpha})\mathbf{r} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix},$$

where

$$\mathbf{c}^T(\mathbf{t}, \boldsymbol{\alpha}) := [\operatorname{Re}(f(t_1)e^{-i\alpha_1}), \operatorname{Re}(f(t_2)e^{-i\alpha_2}), \ldots, \operatorname{Re}(f(t_{n+1})e^{-i\alpha_{n+1}})],$$

and the jth column of the $n+1$ by $n+1$ matrix $A(\mathbf{t}, \boldsymbol{\alpha})$ is

$$[1, \operatorname{Re}(\varphi_1(t_j)e^{-i\alpha_j}), \operatorname{Re}(\varphi_2(t_j)e^{-i\alpha_j}), \ldots, \operatorname{Re}(\varphi_n(t_j)e^{-\alpha_j})]^T.$$

How is Problem P related to Problem D? The classical paper by Rivlin and Shapiro [23] shows that $\|E(\boldsymbol{\lambda}^*, \cdot)\|$, the optimal (minimized) value of $\|E(\boldsymbol{\lambda}, \cdot)\|$ in Problem P, equals h^*, the optimal (maximized) value of h in Problem D. Moreover, if the optimal value h^* of Problem D is achieved at some $\mathbf{t}^*$, $\boldsymbol{\alpha}^*$, and $\mathbf{r}^*$ such that

$$A^{-1}(\mathbf{t}^*, \boldsymbol{\alpha}^*) \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} > \mathbf{0},$$

then h^*, $\boldsymbol{\lambda}^*$, $\mathbf{t}^*$, and $\boldsymbol{\alpha}^*$ satisfy

$$[h^*, \boldsymbol{\lambda}^{*T}] \cdot A(\mathbf{t}^*, \boldsymbol{\alpha}^*) = \mathbf{c}^T(\mathbf{t}^*, \boldsymbol{\alpha}^*),$$

thus allowing us to calculate h^* and $\boldsymbol{\lambda}^*$ from $\mathbf{t}^*$ and $\boldsymbol{\alpha}^*$.

Indeed, when the problem of real approximation is cast in this language, the quadratically convergent real Remez algorithm is an ascent algorithm that solves Problem D (instead of P itself) and yields $\boldsymbol{\lambda}^*$ also as a result. Therefore, we ask the following questions: Can we devise a similar ascent algorithm to solve Problem D in the complex case? Will such an algorithm converge? If so, will it converge quadratically under some moderate assumptions? We answer all these questions affirmatively in the rest of the paper.

3. A Remez Algorithm. Our ascent algorithm to solve Problem D above goes roughly as follows:

Step 0. Pick $S = (\mathbf{t}, \boldsymbol{\alpha})$ such that

$$\mathbf{r} := A^{-1}(\mathbf{t}, \boldsymbol{\alpha}) \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \geq \mathbf{0}.$$

Step 1. Terminate the algorithm if the value $h = \mathbf{c}^T(\mathbf{t}, \boldsymbol{\alpha})\mathbf{r}$ is optimal.

Step 2. Update $S = (\mathbf{t}, \boldsymbol{\alpha})$ and $\mathbf{r}$ (without violating any of the constraints) to increase the value of the scalar product h. Go back to Step 1.

In the rest of this section, we will describe our algorithm and present two examples. First we will explain why each of Steps 0 through 2 is possible.

Step 0. Suppose we can find $\mathbf{t}$ and $\boldsymbol{\alpha}$ such that the matrix $A(\mathbf{t}, \boldsymbol{\alpha})$ is nonsingular (which is almost always the case if we generate $\mathbf{t}$ and $\boldsymbol{\alpha}$ at random); then one can construct $\boldsymbol{\alpha}'$, where $\alpha'_j = \alpha_j$ or $\alpha_j + \pi$, $j = 1, 2, \ldots, n+1$, such that

$$A^{-1}(\mathbf{t}, \boldsymbol{\alpha}') \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \geq \mathbf{0}.$$

Step 1. Given $\mathbf{t}$, $\boldsymbol{\alpha}$, and $\mathbf{r}$, we can define $h \in \mathbb{R}$ and $\boldsymbol{\lambda} \in \mathbb{R}^n$ by the equations

$$[h, \boldsymbol{\lambda}^T] A(\mathbf{t}, \boldsymbol{\alpha}) = \mathbf{c}^T(\mathbf{t}, \boldsymbol{\alpha}).$$

It can be shown (see [25] for details) that

$$h = \mathbf{c}^T(\mathbf{t}, \boldsymbol{\alpha})\mathbf{r} \quad \text{and} \quad h \leq \|E(\boldsymbol{\lambda}^*, \cdot)\| \leq \|E(\boldsymbol{\lambda}, \cdot)\|,$$

where $\boldsymbol{\lambda}^*$ is the best approximation parameter we seek. Consequently, if $h = \|E(\boldsymbol{\lambda}, \cdot)\|$, we know that optimality has been reached. Moreover, if the relative distance $(\|E(\boldsymbol{\lambda}, \cdot)\| - h)/h$ between $\|E(\boldsymbol{\lambda}^*, \cdot)\|$ and h is small enough, the parameter $\boldsymbol{\lambda}$ at hand can be taken as the best parameter for practical purposes. For example, $(\|E(\boldsymbol{\lambda}, \cdot)\| - h)/h < .01$ means that the approximation is as good as the best to within 1%.

To calculate $\|E(\boldsymbol{\lambda}, \cdot)\|$, one can presumably perform a sampling of the values $|E(\boldsymbol{\lambda}, t)|$ for a large number of points t in $[0, 1]$. However, in most practical situations, the derivative of the function $|E(\boldsymbol{\lambda}, t)|$ with respect to t is computable explicitly. Hence, solving

$$\frac{d}{dt}|E(\boldsymbol{\lambda}, t)| = 0$$

for critical points t in $[0, 1]$ is a better method than dense sampling, both in accuracy and speed.

Step 2. This step is the heart of the algorithm. To avoid getting mired in algebraic details, we will use the basic theory of the simplex algorithm in linear programming.**

Suppose that $\|E(\boldsymbol{\lambda}, \cdot)\| > h$; then we could find some $(x, \vartheta) \in [0, 1] \times [0, 2\pi]$ such that

$$h < \|E(\boldsymbol{\lambda}, \cdot)\| = E(\boldsymbol{\lambda}, x)e^{-i\vartheta}.$$

For these fixed $\mathbf{t}$, $\boldsymbol{\alpha}$, x, and ϑ, consider the following linear programming problem: *Find a nonnegative vector* $\mathbf{r}' \in \mathbb{R}^{n+1}$ *and a nonnegative scalar* s *so as to maximize*

$$\mathbf{c}^T(\mathbf{t}, \boldsymbol{\alpha})\mathbf{r}' + \mathrm{Re}(f(x)e^{-i\vartheta})s$$

subject to the constraints

$$[A(\mathbf{t}, \boldsymbol{\alpha}) | \mathbf{v}(x, \vartheta)] \begin{bmatrix} \mathbf{r}' \\ s \end{bmatrix} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix},$$

**For background information, refer to any standard text on linear programming; see [18] for example. For a self-contained derivation of the results to follow, see [25, Chapter 3].

where

$$\mathbf{v}(x,\vartheta) := \operatorname{Re}([1, \varphi_1(x)e^{-i\vartheta}, \varphi_2(x)e^{-i\vartheta}, \ldots, \varphi_n(x)e^{-i\vartheta}]^T).$$

The facts

- $A^{-1}(\mathbf{t},\boldsymbol{\alpha})[{}^1_{\mathbf{0}}] \geq \mathbf{0}$,
- $\mathbf{c}^T(\mathbf{t},\boldsymbol{\alpha})\mathbf{r} = h$, and
- $\operatorname{Re}(E(\boldsymbol{\lambda},x)e^{-i\vartheta}) > h$

imply that $(\mathbf{t},\boldsymbol{\alpha})$ is a nonoptimal feasible basis. Consequently, a new $(\mathbf{t},\boldsymbol{\alpha})$ can be obtained by swapping an appropriate (t_j, α_j) in $(\mathbf{t},\boldsymbol{\alpha})$ with (x,ϑ) so that the following holds:

$$\text{new } \mathbf{r} := A^{-1}(\text{new } (\mathbf{t},\boldsymbol{\alpha})) \cdot \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \geq \mathbf{0},$$

and

$$\mathbf{c}^T(\text{new } (\mathbf{t},\boldsymbol{\alpha})) \cdot \text{new } \mathbf{r} \geq h.$$

In fact, the last "$\geq$" is actually "$>$" except for rare situations that need not concern us for the moment.

We can now restate the algorithm.

REMEZ ALGORITHM

Step 0 (*Initialization*). Pick a stopping threshold $\varepsilon > 0$. Find $\mathbf{t}$, $\boldsymbol{\alpha}$ such that

$$\mathbf{r} := A^{-1}(\mathbf{t},\boldsymbol{\alpha}) \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \geq \mathbf{0}.$$

Step 1 (*Check for optimality*). Find an updating element $(x,\vartheta) \in [0,1] \times [0,2\pi]$ such that $E(\boldsymbol{\lambda},x)e^{-i\vartheta} = \|E(\boldsymbol{\lambda},\cdot)\|$. If $(\|E(\boldsymbol{\lambda},\cdot)\| - h) \leq h\varepsilon$, terminate the algorithm. Otherwise, move on.

Step 2 (*Exchange*). By swapping (x,ϑ) with an appropriate entry in $(\mathbf{t},\boldsymbol{\alpha})$, obtain a new $(\mathbf{t},\boldsymbol{\alpha})$, and go back to Step 1.

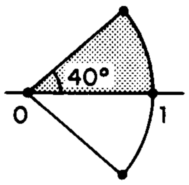
We present two simple examples to give the flavor of the proposed algorithm. The calculations were performed on a DEC VAX in double precision (D-format) with 56 significant bits, roughly 16 significant decimals.

We will count the number of iterations in terms of sweeps, where

$$1 \text{ sweep } = n+1 \text{ iterations.}$$

Measurement in sweeps thus gives us an idea of the algorithm's performance independent of the problem's size. Because each iteration involves inverting a rank-one perturbation of a matrix whose inverse we have already computed, the work per sweep is approximately the same as inverting one full $n+1$ by $n+1$ matrix.

Example 1 [19]. Approximate z^3 on the circular sector (the little circles indicate the extrema of the optimal error curve $E(\boldsymbol{\lambda}^*,\cdot)$)



by quadratics

$$a_1 + a_2 z + a_3 z^2.$$

By symmetry, the problem is equivalent to approximating z^3 on the upper half of the sector by quadratics with real coefficients. Thus the number of real coefficients n equals 3, and 1 sweep equals 4 iterations. The convergence behavior of the algorithm is shown in the following table:

No. of Sweeps	$(\|E(\boldsymbol{\lambda},\cdot)\| - h)/h$
1	9.2×10^{-2}
2	3.3×10^{-4}
3	1.3×10^{-6}
4	5.1×10^{-9}
5	3.0×10^{-11}

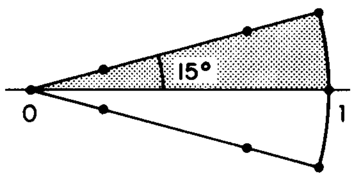
The optimal parameters (8 significant digits) are

$$h^* = 1.8375669 \times 10^{-2},$$
$$\boldsymbol{\lambda}^* = \begin{bmatrix} 1.8479253 \times 10^{-1} \\ -1.1847921 \\ 1.8152371 \end{bmatrix}.$$

Finally, we note that

- the number of extrema of $E(\boldsymbol{\lambda}^*,\cdot)$ is 3, which is less than $n+1$; and
- the rate of convergence seems to be fast but linear.

Example 2 [19]. Approximate z^4 on the sector (the little circles indicate the extrema of the optimal error curve $E(\boldsymbol{\lambda}^*,\cdot)$)



by cubic polynomials

$$a_1 + a_2 z + a_3 z^2 + a_4 z^3.$$

By symmetry, the problem is equivalent to approximating z^4 on the upper half of the sector by cubic polynomials with real coefficients. Thus the number of real coefficients n equals 4, and 1 sweep equals 5 iterations. The convergence behavior of the algorithm is shown in the following table:

No. of Sweeps	$(\|E(\boldsymbol{\lambda},\cdot)\| - h)/h$
1	1.1
2	3.0×10^{-7}
2.6	1.6×10^{-14}

The optimal parameters (8 significant digits) are

$$h^* = 2.1196498,$$

$$\lambda^* = \begin{bmatrix} -2.1196498 \times 10^{-2} \\ 4.8103001 \times 10^{-1} \\ -1.8208555 \\ 2.3398254 \end{bmatrix}.$$

These two examples suggest that

1. the algorithm proposed seems satisfactory (more examples confirming this observation will be exhibited); and

2. when the number of extrema equals $n+1$, the convergence rate may be quadratic.

The next two sections substantiate these observations.

4. Convergence of the Remez Algorithm. The kth iteration of the Remez algorithm generates, among other quantities, the parameter $\boldsymbol{\lambda}^{(k)}$ in $\mathbb{R}^n$ and the value $h^{(k)}$. Does the sequence

$$(\|E(\lambda^{(k)}, \cdot)\| - h^{(k)})/h^{(k)}$$

converge to 0? Without further assumptions on the best approximation of f, the answer is no. However, a weak assumption ensures that a subsequence of $(\|E(\boldsymbol{\lambda}^{(k)}, \cdot)\| - h^{(k)})/h^{(k)}$ does converge to 0.

THEOREM 1. *Suppose that* $\mathbf{r}^{(k)} > \mathbf{0}$ *for all* $k = 1, 2, 3, \ldots$. *Then*

$$\liminf_{k\to\infty} \left(\frac{\|E(\boldsymbol{\lambda}^{(k)}, \cdot)\| - h^{(k)}}{h^{(k)}} \right) = 0.$$

A complete, self-contained proof can be found in [25]. Theorem 1 suffices for practical purposes, because it means the iterations will terminate in a finite number of steps for any positive stopping threshold. Moreover, a stronger assumption implies not only that the whole sequence

$$(\|E(\boldsymbol{\lambda}^{(k)}, \cdot)\| - h^{(k)})/h^{(k)}$$

converges to 0, but that it does so quadratically, as will be explained momentarily. Laurent and Carasso [16] proposed a convex programming algorithm whose convergence proof is almost identical to ours for Theorem 1. However, their algorithm is too general to admit any estimate of its rate of convergence.

5. Quadratic Convergence. We now state the theorem of quadratic convergence and present an example. The necessary assumptions are first presented. The significance of these assumptions will be discussed in detail in the next section. The assumptions are as follows:

Uniqueness. The function f has a unique best approximation.

Smoothness. The functions f, φ_1, $\varphi_2, \ldots, \varphi_n$ are twice continuously differentiable in $[0, 1]$. Furthermore, we assume that whenever

$$E(\boldsymbol{\lambda}, x)e^{-i\vartheta} = \|E(\boldsymbol{\lambda}, \cdot)\|,$$

then

$$\frac{\partial}{\partial t} \operatorname{Re}(E(\boldsymbol{\lambda}, t)e^{-i\alpha})\bigg|_{(x,\vartheta)} = \frac{\partial}{\partial \alpha} \operatorname{Re}(E(\boldsymbol{\lambda}, t)e^{-i\alpha})\bigg|_{(x,\vartheta)} = 0$$

for all $\boldsymbol{\lambda} \in \mathbb{R}^n$.

Nondegeneracy. We assume that the optimal error function $E(\boldsymbol{\lambda}^*, \cdot)$ has exactly $n+1$ extreme points $t_1^*, t_2^*, \ldots, t_{n+1}^*$. Moreover,

$$A^{-1}(\mathbf{t}^*, \boldsymbol{\alpha}^*) \cdot \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} > \mathbf{0},$$

where $\alpha_j^* :=$ argument of $E(\boldsymbol{\lambda}^*, t_j^*)$.

Concavity. We assume that at each of the $n+1$ extreme points, although the first derivative of the function $|E(\boldsymbol{\lambda}^*, t)|$ with respect to t vanishes, the second derivative remains strictly negative.

THEOREM 2. *Under the assumptions above, the sequences* $\{\|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^{(k)}\|\}$ *and* $\{(\|E(\boldsymbol{\lambda}^{(k)}, \cdot)\| - h^{(k)})/h^{(k)}\}$ *converge to zero quadratically in sweeps. Precisely, there exist a sequence* $\{\sigma_k\}$ *and two constants* K *and* M *such that, for all* $k \geq K$,

$$\|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^{(k)}\| \leq M\sigma_k,$$

$$(\|E(\boldsymbol{\lambda}^{(k)}, \cdot)\| - h^{(k)})/h^{(k)} \leq M\sigma_k,$$

and

$$\sigma_{k+n+1} \leq M\sigma_k^2 \to 0.$$

Example. Approximate z^8 on the ellipse $\{(x, y) | x^2 + 4y^2 \leq 1\}$ by lower-degree polynomials

$$a_1 + a_2 z + a_3 z^2 + \cdots + a_8 z^7.$$

By symmetry, the problem is equivalent to approximating z^8 on the upper half of the ellipse by even-degree polynomials

$$a_1 + a_2 z^2 + a_3 z^4 + a_4 z^6$$

with real coefficients a_j's. The convergence behavior of the algorithm is shown in the following table:

No. of Sweeps	$(\|E(\boldsymbol{\lambda}, \cdot)\| - h)/h$
1	1.6×10^{-5}
1.8	2.6×10^{-16}

The optimal parameters (8 significant digits) are

$$h^* = 1.0012817 \times 10^{-1},$$

$$\boldsymbol{\lambda}^* = \begin{bmatrix} -2.4719238 \times 10^{-3} \\ -1.0546875 \times 10^{-1} \\ -7.0312500 \times 10^{-1} \\ -1.5000000 \end{bmatrix}.$$

6. Discussion of the Assumptions. How often are the assumptions leading to Theorem 2 satisfied? The uniqueness assumption is always satisfied for polynomial approximation on a continuum. The smoothness condition is satisfied whenever the boundary of the domain of approximation is smooth. Moreover, in most situations we have come across, the curve in which the approximation takes place is either smooth or piecewise smooth. Our result can be extended to the latter case by techniques similar to those employed in Veidinger's work [27] to handle endpoints. The other two assumptions, nondegeneracy and concavity, need more discussion.

6.1. *The Concavity Assumption and Circular Error Graph.* The concavity assumption is a standard one for convergence proofs of the Remez algorithm for real approximations (cf. [21], [27]). Moreover, even if the concavity assumption is violated, at each extreme point there must be a first nonzero higher-order (even) derivative. The reason is that the optimal error graph must change sign. In such cases, the Remez algorithm for real approximations can be proved to converge superlinearly [15]. The situation, however, can be very different in complex approximation.

Perfectly Circular Error Graph. Suppose we are to find the best approximation to $f(z) = z$ on the unit circle among the complex scalars. It is easy to see that zero is the best approximation. In this case, the optimal error $E(\boldsymbol{\lambda}^*, t)$ satisfies

$$|E(\boldsymbol{\lambda}^*, t)| = 1 \quad \text{and} \quad \frac{\partial}{\partial t}|E(\boldsymbol{\lambda}^*, t)| = 0$$

for all $t \in [0, 1]$. The concavity is clearly violated totally. In general, we can consider those examples with a circular optimal error graph. In those cases,

$$|E(\boldsymbol{\lambda}^*, t)| = \text{constant} \quad \text{and} \quad \frac{\partial}{\partial t}|E(\boldsymbol{\lambda}^*, t)| = 0$$

for all $t \in [0, 1]$. Will the complex Remez algorithm converge at all? If it does, what will the rate be, and what do the optimal parameters $\mathbf{t}^*$ and $\boldsymbol{\alpha}^*$ mean? Athough we have not been able to establish rigorous mathematical results in this situation, in what follows we will present a typical example and offer partial explanations of why neither the convergence rate nor the update procedure of our algorithm is affected by circular error graphs.

Consider the following problem. Approximate $1/[z - (2 + \imath)]$ by a quadratic on the unit circle. We parametrize the circle by $e^{\imath 2\pi t}$, $0 \le t \le 1$. The convergence property we observed is as follows:

No. of Sweeps	$(\|E(\boldsymbol{\lambda}, \cdot)\| - h)/h$
1.3	1.4
2	2.0×10^{-2}
3	1.5×10^{-6}
3.7	1.3×10^{-14}

The optimal parameters (5 significant digits) are

$$h^* = 5.0000 \times 10^{-2},$$

$$\boldsymbol{\lambda}^* = \begin{bmatrix} -4.0000 \times 10^{-1} + \imath 2.0000 \times 10^{-1} \\ -1.2000 \times 10^{-1} + \imath 1.6000 \times 10^{-1} \\ -2.0000 \times 10^{-2} + \imath 1.1000 \times 10^{-1} \end{bmatrix},$$

and

$$(\mathbf{t}^*, \boldsymbol{\alpha}^*) = \begin{bmatrix} 0.0000, & 6.4350 \times 10^{-1} \\ 9.1370 \times 10^{-2}, & -3.0969 \\ 1.8740 \times 10^{-1}, & -6.3090 \\ 3.1621 \times 10^{-1}, & 1.8206 \\ 4.7860 \times 10^{-1}, & -1.8943 \\ 6.7178 \times 10^{-1}, & 1.0094 \\ 8.5035 \times 10^{-1}, & -2.4231 \end{bmatrix}.$$

We first note that h^* and λ^* are correct (cf. [1]) and that the rate of convergence seems to be quadratic. What do the parameters $(\mathbf{t}^*, \boldsymbol{\alpha}^*)$ mean? Certainly, $\mathbf{t}^*$ represents seven of the continuum of extrema of the optimal error graph and $\boldsymbol{\alpha}^*$ represents the arguments of the error function at those seven places. That $\mathbf{r}^*$ (not shown) is strictly positive also implies strong uniqueness of the best approximation [10]. But since we have a continuum of extrema, it seems that $(\mathbf{t}^*, \boldsymbol{\alpha}^*)$ are nonunique. (One can prove that is the case.) Indeed, we are able to execute the algorithm for the example here with different starting values and obtain the same h^* and λ^* at the same convergence rate; but the $(\mathbf{t}^*, \boldsymbol{\alpha}^*)$ is different from before. The algorithm seems to be able to zoom in on one possible pair of $(\mathbf{t}^*, \boldsymbol{\alpha}^*)$ automatically. Thus, nonunique optimal $(\mathbf{t}^*, \boldsymbol{\alpha}^*)$ do not seem to affect the convergence rate. In fact, if there are only a finite number of optimal $(\mathbf{t}^*, \boldsymbol{\alpha}^*)$, we are able to prove that observation. Technical difficulties prevented us from doing the same when there is a continuum of extrema.

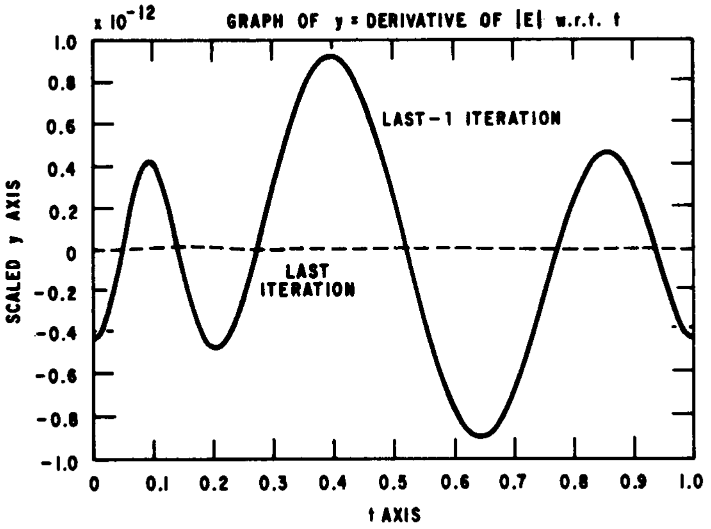
How does circularity affect the update procedure? After all, the optimality check of the Remez algorithm searches for the extrema by solving

$$\frac{\partial}{\partial t}|E(\lambda, t)| = 0$$

for t. To get a feeling for the iteration, we plotted the graph

$$\frac{\partial}{\partial t}|E(\lambda, t)|$$

for $t \in [0, 1]$ during the last two iterations (not sweeps) of the example in question.



The behavior shown here is typical. The graphs of the error functions prior to the optimal one (relative to machine precision) are so noncircular that extrema are typically well distinguishable. For strongly unique approximations, as in this example, one can offer a mathematical explanation. Suppose λ matches λ^* to half the machine precision; then $\|\lambda^* - \lambda\|$ is usually small enough that

$$\min_t |E(\lambda, t)| = \min_t |E(\lambda^*, t) - p(\lambda - \lambda^*, t)| < \|E(\lambda^*, \cdot)\|.$$

On the other hand, strong uniqueness means that there is a $c > 0$ such that

$$\|E(\lambda, \cdot)\| \geq \|E(\lambda^*, \cdot)\| + c\|\lambda^* - \lambda\|.$$

Thus, $|E(\lambda, t)|$ would deviate from circularity significantly with respect to the machine precision. In the presence of quadratic convergence, λ would usually match λ^* to only roughly 80% of the machine precision before the last iteration, the case here.

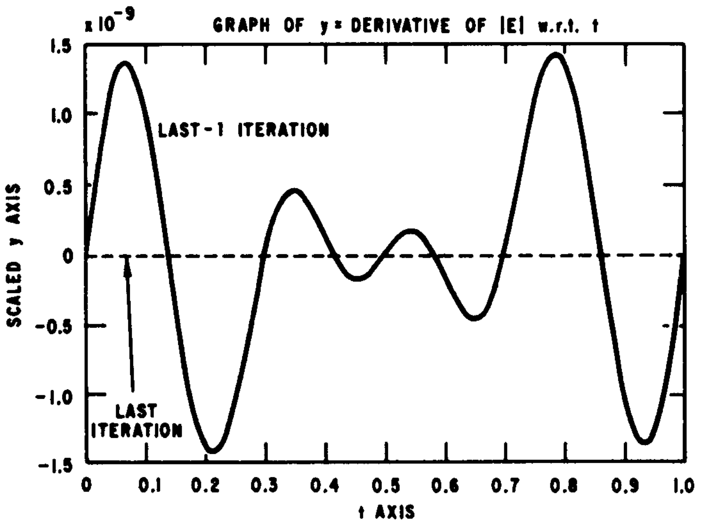
Nearly Circular Error Graph. Next we present two approximation problems with optimal error graphs circular to within machine precision. They are the approximations on the unit circle of

$$\cos(z) \text{ by } a_1 + a_2 z^2 + \cdots + a_4 z^6, \qquad a_j \in \mathbf{R},$$

and

$$\sin(z) \text{ by } a_1 z + a_2 z^3 + \cdots + a_5 z^9, \qquad a_j \in \mathbf{R}.$$

(The optimal error graphs cannot be perfectly circular because sin and cos are not rational functions. For a succinct proof, see Trefethen [26].) Note that the effective machine precisions for the two problems are 11 and 9 digits, respectively, because the optimal errors have magnitudes of the order 10^{-5} and 10^{-7}, respectively. The graph of $\frac{\partial}{\partial t}|E(\lambda, t)|$ during the last two iterations of the cos example is given below. The graph for the sin example is similar.



Like the perfectly circular examples, near circularity affects neither the algorithm nor its speed.

6.2. *Degeneracy.* Unlike the assumption about concavity, which in our experience fails only in special and rare situations, the assumption on nondegeneracy is violated frequently. Experimentally, somewhere between 60%–70% of the examples we tried were degenerate. Moreover, whenever degeneracy occurs, the convergence rate the algorithm exhibits is always linear only.

At present, we are not aware of any satisfactory explanation of the cause of degeneracy. The only work we know is that by Blatt [2] which discusses how often degeneracy occurs. Blatt shows that in the set of continuous complex-valued functions on a compact set with a limited number of isolated points, the subset of functions with nondegenerate best approximations is dense. However, Blatt left open the question of whether a similar result would hold if the word "continuous" were replaced by "analytic." Clearly, our experiments strongly suggest the negative. Indeed, we are able to prove such is the case; the proof is the subject of another paper.

Although degenerate problems show up frequently, they fortunately exhibit a typical behavior that we can exploit to restore quadratic convergence. This is the subject of the next section.

7. An Extension of the Remez Algorithm.

Let f be a function that satisfies the "uniqueness," "smoothness," and "concavity" conditions stated in the previous section; and let $E(\boldsymbol{\lambda}^*,\cdot)$ be the optimal error function. However, the function $|E(\boldsymbol{\lambda}^*,t)|$ has only $n+1-d$ extreme points t_1^*, $t_2^*,\ldots,t_{n+1-d}^*$. The question is: can we still find the best approximation quickly? For argument's sake, suppose we were given the last d entries λ_{n-d+1}^*, $\lambda_{n-d+2}^*,\ldots,\lambda_n^*$ of $\boldsymbol{\lambda}^*$. Then, to find the best approximation to f, it would suffice to find the best approximation to the new function

$$f-(\lambda_{n-d+1}^*\varphi_{n-d+1}+\lambda_{n-d+2}^*\varphi_{n-d+2}+\cdots+\lambda_n^*\varphi_n)$$

from the approximants

$$\mathscr{W}:=\{\mu_1\varphi_1+\mu_2\varphi_2+\cdots+\mu_{n-d}\varphi_{n-d}\mid\mu_j\in\mathbf{R}\}.$$

Clearly, the Remez algorithm will converge quadratically when applied to the new approximation problem just defined. The trouble is, however, that we do not know those values λ_{n-d+1}^*, $\lambda_{n-d+2}^*,\ldots,\lambda_n^*$ a priori.

Although the λ_j^*'s are unavailable for $n-d+1\le j\le n$ a priori, we can obtain some approximate values:

$$[x_1,x_2,\ldots,x_d]^T\approx[\lambda_{n-d+1}^*,\lambda_{n-d+2}^*,\ldots,\lambda_n^*]^T.$$

We can, for example, use the Remez algorithm with a crude stopping threshold of 0.5, say, to find the best approximation of f from the original set of approximants. Using those x_j's, we can devise a plausible scheme to find $\boldsymbol{\lambda}^*$ as follows:

1. Use the Remez algorithm to find an approximant in the set

$$\{\mu_1\varphi_1+\mu_2\varphi_2+\cdots+\mu_{n-d}\varphi_{n-d}\mid\mu_j\in\mathbf{R}\}$$

that best approximates the function

$$f-(x_1\varphi_{n-d+1}+x_2\varphi_{n-d+2}+\cdots+x_d\varphi_n).$$

2. Derive a method to improve x based upon the results obtained in (1) above.
3. Repeat (1) and (2).

As it turns out, provided the x_j's are close to the corresponding λ_j^*'s to begin with, then

- the process in (1) converges quadratically;
- $\mathbf{x}^*:=[\lambda_{n-d+1}^*,\lambda_{n-d+2}^*,\ldots,\lambda_n^*]^T$ can be characterized as the root of a certain function $\mathbf{H}$ from R^d to R^d. Hence, Newton's iteration can be used to correct the

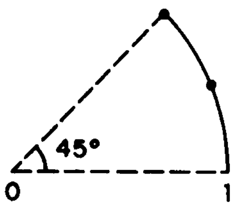
x_j's; and

• the Jacobian of the function **H** just mentioned is nonsingular at $\mathbf{x}^*$. Thus, the iterations (1)–(2) converge quadratically.

A complete proof can be found in [25].

To summarize, the method requires a few initial sweeps of Remez iterations. Then, some nested iterations are performed. The outer iteration of the nest is a Newton's iteration on a $d \times d$-system (d = deficiency); and the inner loop requires a number of sweeps of Remez iterations applied to a problem with $n+1-d$ parameters. Let us illustrate the scheme by a problem with deficiency 2.

Example. Approximate $F(z) = z^3$ by a real quadratic on the circular arc



After 3.3 sweeps, $\mathbf{t}^{(k)}$ looks like

$$[.4793, .4918, 1, 1]^T,$$

suggesting that the deficiency is 2. So we use the extended algorithm:

Outer Loop Newton's Iter. No.	Inner Loop No. of sweeps needed	$(\|E(\lambda,\cdot)\| - h)/h$
1	3.7	1.9×10^{-3}
2	1.3	1.5×10^{-4}
3	1.0	2.4×10^{-8}
4	1.0	2.2×10^{-12}

8. Comparison with Two Other Algorithms.

Two recently proposed algorithms ([24] and [9]) for complex polynomial approximation are so similar to our Remez algorithm that we would like to compare them. (For comparison with other algorithms such as [19], [28], [17], and [5], see [25].)

These two algorithms try to solve Problem D in a way different from our Remez algorithm. Recall Problem D:

Problem D. Find $\mathbf{t} \in [0,1]^{n+1}$, $\boldsymbol{\alpha} \in [0, 2\pi]^{n+1}$ and $\mathbf{r} \in [0,1]^{n+1}$ so as to maximize

$$\mathbf{c}(\mathbf{t}, \boldsymbol{\alpha})^T \mathbf{r}$$

subject to the constraints

$$A(\mathbf{t}, \boldsymbol{\alpha})\mathbf{r} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}.$$

Both [24] and [9] discretize the domain $[0,1] \times [0, 2\pi]$ for $(\mathbf{t}, \boldsymbol{\alpha})$ in Problem D and hence change it to a standard linear programming problem. This new problem,

Problem D′, can then be solved by a standard simplex algorithm. But how can Problem D be solved by solving Problem D′? This is where [24] and [9] differ.

8.1. *Repeated Discretization.* If the set $[0,1] \times [0,2\pi]$ is replaced by a discrete set

$$\mathscr{D}_N = \{(t_1, \alpha_1), (t_2, \alpha_2), \ldots, (t_N, \alpha_N)\} \subset [0,1] \times [0,2\pi],$$

then Problem D′ is an ordinary linear programming problem with $n+1$ constraints and N variables. As $N \to \infty$, it can be proved (under moderate assumptions) that the optimal parameter $\boldsymbol{\lambda}^{\text{discrete}}$ to the discrete problem converges to $\boldsymbol{\lambda}^*$. Hence Streit and Nuttall [24] suggest solving Problem D simply by solving Problem D′ with some N large enough.

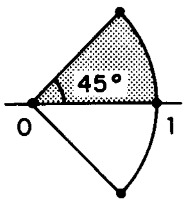
If the domain $[0,1]$ were discrete, then in order to obtain an approximant close to the best to within a given threshold, [24] shows that a discretization on $[0,2\pi]$ can be chosen a priori. That will be the case, for example, when the original approximation problem is defined on a discrete domain in the complex plane. When, as in our situation, the domain of approximation is a continuum, then the discretization cannot be chosen a priori. Consequently, the problem may have to be resolved with a finer discretization whenever the initial one turns out to be too coarse. Some observations are in order.

• Combining the results in [24] and [4] shows that the convergence rate of repeatedly refining the discretization is of second order $O(1/N^2)$.

• Although the number of simplex iterations required to solve each discretized problem is roughly 2 to 3 times the dimension (independent of N) of the problem, the cost of solving each discretized problem does not remain the same as the grid gets refined. The reason is that the number of function evaluations at each simplex iteration is proportional to N and, when this number is much larger than the dimension of the problem, the cost will be apparent.

• Suppose we would like to have $\|E(\boldsymbol{\lambda}^{\text{discrete}}, \cdot)\|$ to within 1% of $\|E(\boldsymbol{\lambda}^*, \cdot)\|$; how large should N be? This cannot be determined a priori. Hence we have to solve at least two discretized problems.

Example [19]. Approximate z^4 on the sector



by real cubic polynomials. We first tabulate the results of repeated discretization.

Streit/Nuttall

No. of Points in $[0,1] \times [0,2\pi]$	No. of Sweeps	Cumulative No. of Function Evaluations	$(\|E(\lambda,\cdot)\| - h)/h$
17×32	2.4	204	1.5×10^{-2}
33×64	2.8	462	1.2×10^{-3}
65×128	3.6	1170	8.5×10^{-4}
129×256	4.2	2709	1.8×10^{-4}
257×512	4.0	5140	2.9×10^{-5}
513×1024	4.0	10260	1.5×10^{-5}
1025×2048	5.0	25625	1.2×10^{-6}

Using the Remez algorithm to solve the same problem, we get the following:

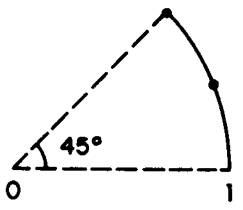
Remez

No. of Sweeps	Cumulative No. of Function Evaluations	$(\|E(\lambda,\cdot)\| - h)/h$
1.6	189	5.0×10^{-2}
2.2	267	7.9×10^{-3}
2.6	323	4.9×10^{-4}
3.2	372	1.0×10^{-5}
3.6	401	4.2×10^{-6}
4.4	428	8.8×10^{-7}
5.0	466	2.8×10^{-8}
5.4	502	1.7×10^{-9}

The example illustrates that the two methods are comparable if we need only an approximant within 1% to the best. It is possible, though, the algorithm in [24] has to be applied more than once. If an approximant that is much closer to the best is desired, then the size of the discretization has to be quite large. In that case, the Remez algorithm is much more efficient.

8.2. *Discretization and Newton's Iteration.* Similar to the previous scheme is the one suggested by Glashoff and Roleff [9]. It is not hard to show that the best approximation can be characterized locally by a system of $4 \times (n+1)$ nonlinear equations (see, for example, [25]). Hence, provided a first approximation to the solution can be found, one may apply Newton's iteration to those $4 \times (n+1)$ equations. Indeed, Glashoff and Roleff suggest that the solution to a discretized problem (Problem D$'$) be used as an initial guess (starting vector) for the Newton's iteration. Thus, [9] consists of two phases. Phase 1 is equivalent to the previous scheme of Streit and Nuttall, and Phase 2 involves a number of inversions of $4(n+1) \times 4(n+1)$ matrices. Because Phase 2 is expensive, it is necessary only when an approximant very close to the best is needed. We will therefore compare the Remez algorithm with [9], assuming this to be the case. Let us consider an example. (More examples can be found in [25].)

Example [24]. Approximate z^3 on the arc



by a real quadratic. Using Glashoff and Roleff's algorithm, we first solve a discretized problem.

Glashoff/Roleff
Phase 1: discretized problem

No. of pts. in $[0, 1] \times [0, 2\pi]$	No. of Sweeps	$(\|E(\lambda, \cdot)\| - h^*)/h^*$
33×64	3.5	1.9×10^{-3}

Glashoff/Roleff
Phase 2: Newton's iteration on 9 equations

Newton's Iteration No.	$(\|E(\lambda, \cdot)\| - h^*)/h^*$	No. of Sweeps-equivalents
1	1.3×10^{-3}	45
2	5.3×10^{-6}	45
3	1.5×10^{-9}	45
4	1.6×10^{-12}	45

We can also use the extended version of the Remez algorithm to solve this problem. After 3.3 sweeps, $(\|E(\lambda, \cdot)\| - h)/h$ is 8.5×10^{-3}, and $\mathbf{t}^{(k)}$ is

$$[.48, .49, 1, 1]^T,$$

suggesting that the degeneracy is 2. Hence, only 2 (compared to 9) equations need to be solved by Newton's iteration.

Remez: Newton's iterations for 2 equations

Newton's Iteration No.	No. of Sweeps	$(\|E(\lambda, \cdot)\| - h)/h$
1	3.7	1.9×10^{-3}
2	1.3	1.5×10^{-4}
3	1.2	2.5×10^{-8}
4	0.8	1.6×10^{-12}

This and the previous examples, among others, sustain the following observations:

• When the problem is nondegenerate, the Remez algorithm is much more economical than the two-stage method of Glashoff and Roleff. On the average, Remez takes 4 to 5 sweeps to reduce the number $(\|E(\boldsymbol{\lambda},\cdot)\| - h)/h$ to the rounding error level. On the other hand, one Newton's iteration during Glashoff and Roleff's second stage involves inverting a matrix roughly four times the dimension of the problem. Hence, each of these iterations can cost as much as 4^3 sweeps.

• The competition between the two algorithms becomes more interesting when the problem is degenerate, in which case an extension of the Remez algorithm is needed to ensure quadratic convergence. Despite this extension, the Remez algorithm is still much more economical because the work of each correction step is roughly 1 to 1.5 sweeps, as opposed to 4^3 sweeps in Glashoff and Roleff's algorithm.

• Practically, it seems that an approximation whose $\|E(\boldsymbol{\lambda},\cdot)\|$ lies within a tenth of a percent of the best approximation is sufficient most of the time. Thus, the Remez algorithm without any extension seems to be the most straightforward and economical algorithm to use.

9. Some Applications.

9.1. *Approximate Conformal Maps.* Suppose we want to map the ellipse $x^2 + (4y/3)^2 = 1$ conformally to a disk. It is known that the conformal map is the analytic function $g(z)$ in the set

$$\{g \mid g(0) = 0 \text{ and } g'(0) = 1\}$$

with the smallest maximum magnitude taken on the whole ellipse. Motivated by this observation, we can find an approximate conformal map by calculating the polynomial function in the set

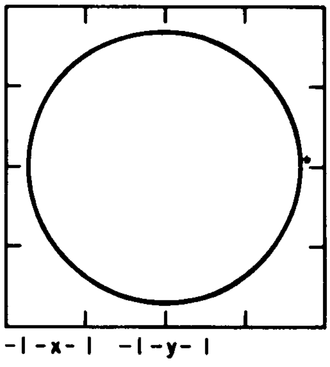
$$\{z + a_1z^2 + a_2z^3 + \cdots + a_nz^{n+1} \mid a_j \in \mathbb{C}\}$$

with the smallest norm on the ellipse (cf. [11], [20]). This is equivalent to finding the best approximation to the function $F(z) = -z$ from the set

$$\{a_1z^2 + a_2z^3 + \cdots + a_nz^{n+1}\}$$

over the ellipse.

We use our Remez algorithm to solve the problem with $n = 8$. (Because of symmetry, the best coefficients are real.) After 2 sweeps, the approximation is close to within 1% of the best. The ellipse's image under this approximate conformal map, as shown in the figure below, is a circle to within 0.01%.



9.2. *Voigt Profile.* The function

$$F(z) = e^{-z^2}\left(1 + \frac{2i}{\sqrt{\pi}}\int_0^z e^{t^2}\,dt\right)$$

is of interest in spectroscopy and astrophysics because of its relationship to the Voigt function ([14], [13], [8]). We approximated this function over the square $[0,1]\times[0,1]$ by a complex polynomial of degree 12 (13 complex coefficients). After 3 sweeps, the approximation is well within 1% of the best, and its error $\|E(\lambda,\cdot)\| = 4.8\times 10^{-8}$.

9.3. *Gamma Function.* Suppose we want to calculate the value of the gamma function for z close to the real axis, say $|\mathrm{Im}(z)| \le 1/8$. It suffices to be able to calculate $\Gamma(z)$ for z where $|\mathrm{Re}(z)| \le 1/2$ and $|\mathrm{Im}(z)| \le 1/8$ because $\Gamma(z+1) = z\Gamma(z)$. Moreover, because $1/\Gamma(z)$ is an entire function, it is reasonable to approximate $\Gamma(z)$ by the inverse of a polynomial. Furthermore, for the sake of relative accuracy, we will approximate

$$\frac{1}{z\Gamma(z)} \quad \text{by } 1 + a_1 z + a_2 z^2 + \cdots + a_n z^n.$$

We have done this for $n = 7$, 8, and 9, and 3 to 4 sweeps are all we need to obtain an approximation to well within 1% of the best. We also compare the best approximation to the one obtained by simply truncating the Taylor series expansion of $1/(z\Gamma(z))$.

No. of Coefficients	$\|E(\lambda^{\mathrm{Taylor}},\cdot)\|$	$\|E(\lambda^{\mathrm{best}},\cdot)\|$
7	6.2×10^{-6}	3.4×10^{-7}
8	7.3×10^{-7}	2.5×10^{-8}
9	1.8×10^{-7}	4.7×10^{-9}

10. Conclusion.

We have shown that the Remez algorithm for real linear Chebyshev approximations can be generalized satisfactorily to work for complex approximation. Our algorithm seems to be the fastest available today for such problems and has given us hope that a fast algorithm for rational approximation may also be close at hand.

Acknowledgments. The author thanks Professor W. Kahan for his continual guidance on the problem and acknowledges the referee's insightful and constructive criticism. The bulk of this work was done while the author was a graduate student at the University of California at Berkeley, financially supported over a period of time by Air Force grant AFOSR-84-0158, a grant from ELXSI Inc., and an IBM pre-doctoral fellowship.

Mathematics and Computer Science Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois 60439-4844
E-mail: tang@anl-mcs.arpa

1. I. Barrodale, L. M. Delves & J. C. Mason, "Linear Chebyshev approximation of complex-valued functions," *Math. Comp.*, v. 32, 1978, pp. 853 863.

2. H.-P. Blatt, "On strong uniqueness in linear complex Chebyshev approximation," *J. Approx. Theory*, v. 41, 1984, pp. 159 169.

3. E. W. CHENEY, *Introduction to Approximation Theory*, 2nd ed., Chelsea, New York, 1986.

4. C. B. DUNHAM & J. WILLIAMS, "Rate of convergence of discretization in Chebyshev approximation," *Math. Comp.*, v. 37, 1981, pp. 135–139.

5. S. W. ELLACOTT & J. WILLIAMS, "Linear Chebyshev approximation in the complex plane using Lawson's algorithm," *Math. Comp.*, v. 30, 1976, pp. 35–44.

6. H. C. ELMAN & R. L. STREIT, *Polynomial Iteration for Nonsymmetric Indefinite Linear Systems*, Research report YALEU/DCS/RR-380, Department of Computer Science, Yale University, New Haven, Conn., March 1985.

7. B. FRANCIS, J. W. HELTON & G. ZAMES, "H^∞-optimal feedback controllers for linear multivariable systems," *IEEE Trans. Automat. Control*, v. 29, 1984, pp. 888–900.

8. W. GAUTSCHI, "Efficient computation of the complex error function," *SIAM J. Numer. Anal.*, v. 7, 1970, pp. 187–198.

9. K. GLASHOFF & K. ROLEFF, "A new method for Chebyshev approximation of complex-valued functions," *Math. Comp.*, v. 36, 1981, pp. 233–239.

10. M. H. GUTKNECHT, "Non-strong uniqueness in real and complex Chebyshev approximation," *J. Approx. Theory*, v. 23, 1978, pp. 204–213.

11. M. HARTMANN & G. OPFER, "Uniform approximation as a numerical tool for constructing conformal maps," *J. Comput. Appl. Math.*, v. 14, 1986, pp. 193–206.

12. J. W. HELTON, "Worst case analysis in the frequency domain: An H^∞ approach for control," *IEEE Trans. Automat. Control*, v. AC-30, 1985, pp. 1192–1201.

13. A. K. HUI, B. H. ARMSTRONG & A. A. WRAY, "Rapid computation of the Voigt and complex error functions," *J. Quant. Spectrosc. Radiat. Transfer*, v. 19, 1978, pp. 509–516.

14. J. HUMLÍČEK, "Optimized computation of the Voigt and complex probability functions," *J. Quant. Spectrosc. Radiat. Transfer*, v. 27, 1982, pp. 437–444.

15. W. KAHAN, *Lecture Notes on Superlinear Convergence of a Remes Algorithm*, Class Notes for Computer Science 274, University of California at Berkeley, April 1981.

16. P. LAURENT & C. CARASSO, "An algorithm of successive minimization in convex programming," *RAIRO Anal. Numér.*, v. 12, 1978, pp. 377–400.

17. C. L. LAWSON, *Contributions to the Theory of Linear Least Maximum Approximations*, Ph.D. thesis, University of California at Los Angeles, 1961.

18. D. G. LUENBERGER, *Linear and Nonlinear Programming*, 2nd ed., Addison-Wesley, Reading, Mass., 1984.

19. G. OPFER, "An algorithm for the construction of best approximation based on Kolmogorov's criterion," *J. Approx. Theory*, v. 23, 1978, pp. 299–317.

20. G. OPFER, "New extremal properties for constructing conformal mappings," *Numer. Math.*, v. 32, 1979, pp. 423–429.

21. M. J. D. POWELL, *Approximation Theory and Methods*, Cambridge Univ. Press, Cambridge, 1981.

22. L. REICHEL, "On polynomial approximation in the complex plane with application to conformal mapping," *Math. Comp.*, v. 44, 1985, pp. 425–433.

23. T. J. RIVLIN & H. S. SHAPIRO, "A unified approach to certain problems of approximation and minimization," *J. Soc. Indust. Appl. Math.*, v. 9, 1961, pp. 670–699.

24. R. L. STREIT & A. H. NUTTALL, "A general Chebyshev complex function approximation procedure and an application to beamforming," *J. Acoust. Soc. Amer.*, v. 72, 1982, pp. 181–190.

25. P. T. P. TANG, *Chebyshev Approximation on the Complex Plane*, Ph.D. thesis, Department of Mathematics, University of California at Berkeley, May 1987.

26. L. N. TREFETHEN, "Near circularity of the error curve in complex Chebyshev approximation," *J. Approx. Theory*, v. 31, 1981, pp. 344–366.

27. L. VEIDINGER, "On the numerical determination of the best approximation in the Chebyshev sense," *Numer. Math.*, v. 2, 1960, pp. 99–105.

28. J. WILLIAMS, "Numerical Chebyshev approximation in the complex plane," *SIAM J. Numer. Anal.*, v. 9, 1972, pp. 638–649.