ON NUMERICAL APPROXIMATION IN SYNTHESIS FOR PRESCRIBED AMPLITUDE RESPONSE

by

WALTER JAMES HENRY HARRIS

B.A.Sc., U.B.C., 1962

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Applied Science

in the Department of

Electrical Engineering

We accept this thesis as conforming to the required standard

Members of the Department

of Electrical Engineering

THE UNIVERSITY OF BRITISH COLUMBIA

NOVEMBER, 1964

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the Head of my Department or by his representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical Engineering

The University of British Columbia,
Vancouver 8, Canada

Date November 30, 1964

ABSTRACT

This thesis describes a new method of obtaining suitable equal-ripple rational functions for use in the synthesis of networks. It has two principal advantages over other methods used for this purpose. Firstly, it is capable of finding the best approximation to a given magnitude function, phase function, or both simultaneously. The error of approximation may be weighted as desired at any frequency in the range of approximation. Secondly, it is easily adapted for use on an automatic computer. This enables quick comparison of the approximations produced by using rational functions of different orders of complexity.

ACKNOWLEDGEMENT

The author would like to thank Dr. A. D. Moore for his supervision and guidance throughout this research project and the staff of the computing centre for their generous assistance.

The principal part of this research was carried out with the support of the National Research Council with additional assistance from a B. C. Telephone Co. Scholarship.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1. INTRODUCTION

Historically, suitable rational functions for the synthesis of networks have been obtained by two general classes of methods: one has been the use of an analogue of the network which allowed easier manipulation of poles and zeroes than the actual network, and the other has been the use of various analytic functions and truncated series. Probably the best example of the former is the potential analogy. For the latter, the Butterworth and Bessel approximations to the low-pass filter characteristic are among the better known. For many applications these methods produce sufficiently good results. However, when filters are required to have precisely specified magnitude and phase responses, as for example in long telephone networks, it is often desirable to be able to specify the maximum deviation from the desired function and to make this value as small as possible for a given degree of network complexity. To begin the synthesis, it is first necessary to obtain either the positions of the poles and zeroes or, equivalently, the coefficients of the rational function. Approximations obtained by using the potential analogy in the form of an electrolytic tank give the pole and zero positions directly while classical analytic approaches usually result in one of these sets of quantities expressed by an explicit formula in terms of tabulated functions. However, in the case of approximations obtained by directly reducing the maximum error, the amount of calculation is, generally, considerably greater.

The potential analogue has the advantage of permitting a better intuitive feeling for the sensitivity of the final response to small deviations in pole and zero positions, but the disadvantage of being capable of only limited accuracy. Analytic approximations may result in an easily characterized response; for example, with the Chebyshev approximations to a constant, the maximum passband ripple is directly available. A disadvantage is that these methods have only been developed for simple forms of responses. The numerical methods to be considered in this paper exploit the capabilities of modern high-speed automatic computers; this allows direct reduction of the maximum error of response with respect to practically any arbitrary function, with accuracy limited only by the number of significant figures retained during calculations.

Although questions of the existence and uniqueness of a "best" rational function approximating a desired function will be dealt with later, it is necessary first to consider the limitations imposed for the sake of physical realizability.

1-1. Properties of Impedances

The network functions to be approximated can be represented as either transfer impedances or driving-point impedances. The following requirements must be met by a physically-realizable driving-point impedance (or admittance) produced by a network of lumped, linear, passive, time-invariant elements:

1. $Z(s)$ is a rational function of the complex frequency s, and can be written in the two equivalent forms:

$$Z(s) = \frac{\sum_{i=0}^{n} a_i s^i}{\sum_{i=0}^{m} b_i s^i} = K \cdot \frac{\prod_{i=1}^{n} (s - s_i)}{\prod_{i=1+n}^{n+m} (s - s_i)}$$

2. Poles and zeroes of Z(s) are either real or occur in complex conjugate pairs or, equivalently, all the coefficients a_i and b_i are real.
3. The real parts of all poles and zeroes of Z(s) must be non-positive.
4. Poles and zeroes of Z(s) must be simple with real, positive residues if the real part is zero.
5. The constant multiplier K must be real and positive.
6. $|m - n|$ must be either 0 or 1.

The requirements for an open circuit transfer impedance are the same except that the real parts of the zeroes of Z(s) are not restricted and $m - n \geqslant -1$.

Usually specifications for response are given for real frequencies, i.e., for $s = j\omega$. In some cases, the impedance is separated into real and imaginary components. If the impedance is written in the form

$$Z(s) = \frac{m_1 + n_1}{m_2 + n_2}$$

where the m's and n's denote respectively the even and odd parts of the polynomials, then, for $s = j\omega$, the even parts are real and the odd parts are imaginary, and

$$\mathrm{Re}\left[\mathbf{Z}(s)\right] = \frac{m_1 m_2 - n_1 n_2}{(m_2)^2 - (n_2)^2}$$

$$\mathrm{Im}\left[\mathbf{Z}(s)\right] = \frac{m_2 n_1 - m_1 n_2}{(m_2)^2 - (n_2)^2}$$

Assuming that $\mathbf{Z}(s)$ is analytic in the right half-plane, that is, for all $s \in \mathrm{Re}(s) > 0$, then either the real part or the imaginary part can be obtained from the other, to within an added resistance or reactance respectively.

For some purposes it is preferable to use polar co-ordinates and describe the impedance in terms of its magnitude and phase angle. For theoretical purposes, it is more common to use the squared magnitude or the logarithm of the magnitude rather than the magnitude itself. The "squared-magnitude" function is non-negative for all ω, and may be obtained from

$$\mathbf{Z}(s)\cdot\mathbf{Z}(-s) = \frac{(m_1)^2 - (n_1)^2}{(m_2)^2 - (n_2)^2}$$

by setting $s = j\omega$, so that

$$\left|\mathbf{Z}(j\omega)\right|^2 = \mathbf{Z}(j\omega) \cdot \mathbf{Z}(-j\omega)$$

The phase angle is an odd function having the principal value,

$$\theta(\omega) = \tan^{-1}\left[\frac{m_2 n_1 - m_1 n_2}{j(m_1 m_2 - n_1 n_2)}\right]_{s = j\omega}$$

This function is continuous except at frequencies corresponding to poles or zeroes lying on the $j\omega$ axis, at which points the phase changes discontinuously by an integer multiple $\pm \pi$. Again assuming no right-half-plane poles or zeroes are present, the magnitude or the phase may be obtained from each other. By factoring the numerator and denominator of $Z(s).Z(-s)$ and retaining only the left-half-plane poles and zeroes, $Z(s)$ can be obtained. Thus, knowing any one of $Z(s)$, $\mathrm{Re}\left[Z(s)\right]$, $\mathrm{Im}\left[Z(s)\right]$, $\theta(\omega)$, $Z(s).Z(-s)$, or $\left|Z(j\omega)\right|^2$, any one of the others can, in theory, be obtained, subject to the stated assumptions.

In this thesis, only approximations to specified magnitude functions will be treated in detail, although the algorithm to be developed can be readily extended for approximating other functions derived from $Z(s)$.

1-2. Types of Functions to be Approximated

Network functions to be approximated may be separated broadly into two categories: those appropriate for describing band filters and those appropriate for equalizers. Often the specifications for transmission through such networks are given in terms of the attenuation. This quantity is proportional to the logarithm of the magnitude of the transfer impedance, i.e.,

$$A = -k \cdot \log |Z| \text{, where k is real and positive.}$$

Band filters are characterized by distinct passbands (intervals of ω over which the attenuation is small) and stopbands (intervals of ω over which the attenuation is large)

alternating along the ω-axis. Specifications for this type of filter are often given in terms of:

(i) permissible passband ripple, i.e., the ratio of maximum to minimum transmission within a passband;

(ii) minimum attenuation in the stop-bands;

and (iii) width of transition- or guard-bands.

Equalizers generally do not require sharp transitions, but often require careful control of either magnitude or phase, or both. Examples are networks used to shape the frequency characteristic of the feedback loop of an amplifier, to equalize the frequency-dependent loss in a transmission line, or to simulate transmission-line impedances.

Following a discussion of some methods which have been used for finding rational-function approximations, a new method will be shown which seems more suitable for equalizer applications.

2. MATHEMATICAL BACKGROUND

2-1. Notation

For convenience, the independent variable will be denoted by x in place of ω or ω^2. Let f(x) be a specified squared-magnitude function and let the rational function approximation to f(x) on the interval $a \leqslant x \leqslant b$ be given by

$$F(x) = \frac{P(x)}{Q(x)} = \frac{\sum_{i=0}^{n} p_i x^i}{\sum_{i=0}^{m} q_i x^i}$$

$$\sum_{i=0}^{m} q_i x^i \neq 0, \quad x \in [a,b]$$

$$\sum_{i=0}^{m} (q_i)^2 = 1$$

The error of approximation will be defined by

$$\varepsilon(x) = f(x) - F(x) .$$

Since the number of coefficients of the rational function occurs rather often, let it be represented by

$$n_c = n + m + 2 .$$

When it is convenient to refer to the full set of numerator and denominator coefficients, this point in n_c-dimensional space will be called R.

In order to provide an exact mathematical statement of the problem to be solved, it is necessary to define a precise

measure of the closeness of the approximation to the desired function. This will be done in terms of the norm of the "error" function. A norm is a real scalar quantity defined on a function space. It is denoted by $\| \quad \|$ and must satisfy the following three conditions:

1. $\| g(x) \| \geqslant 0$, and $\| g(x) \| = 0$ if and only if $g(x) \equiv 0$
2. $\| cg(x) \| = |c| \cdot \| g(x) \|$ for any real constant c
3. $\| h(x) + g(x) \| \leqslant \| g(x) \| + \| h(x) \|$

for all g(x) and h(x) in the function space. The two linear spaces and their norms which will be of interest are:

1. <u>The Space L^p $(p \leqslant 1)$</u>

   The space L^p consists of the totality of all functions measurable in the interval (a,b) whose absolute value to the p-th power is integrable in the sense of Lebesgue. The norm in L^p, called the least p-th norm, is defined by

$$\| g \| = \left\{ \int_a^b |g(t)|^p \, dt \right\}^{1/p}$$

2. <u>The Space C</u>

   The space C consists of the totality of all continuous functions of the points P of a bounded, closed set S in Euclidean space of any dimensionality. The norm in C, called the Chebyshev norm, is defined by

$$\| g \| = \max_{P \in S} | g(P) |$$

Setting $g = \varepsilon(x) = f(x) - F(x)$, the norm of the error function, $\|\varepsilon\|$, is a suitable measure of the closeness of the approximation to the specified function.

2-2. Statement of the Approximation Problem

Given $f(x)$, a real-valued continuous function defined on a set S, find a rational function $F(R^*,x)$ with $R^* \epsilon P$, a bounded set, such that

$$\|\varepsilon\ (R^*)\| \leqslant \|\varepsilon(R)\| \qquad \text{for all } R \epsilon P$$

Such a rational function will be called a best approximation.

2-3. Existence and Uniqueness

For the Chebyshev norm, existence and uniqueness for the case of approximation of a continuous function over a set containing no isolated points are assured by the following theorem attributed to Chebyshev:

Theorem: Given a closed (finite or infinite) interval $[a,b]$ on the real-number axis and a real, single-valued function, $f(x)$, continuous in $[a,b]$, there then exists at least one function

$$F_{n,m}(x) = \frac{P_n(x)}{Q_m(x)}$$

such that

$$\|\varepsilon\| = \max_{x \epsilon [a,b]} |\ f(x) - F_{n,m}(x)\ |$$

for this function is not greater than for any other rational function of the same order. Moreover, this function $F_{n,m}(x) = F^*_{n,m}(x)$ is unique, if we consider

two rational functions as identical when they coincide after being reduced to their lowest terms. The function ε takes on its maximum absolute value for at least $n_c - \min(\mu,\nu)$ points in $[a,b]$, where $n-\mu$ and $m-\nu$ are the highest powers of x with non-zero coefficients occurring in $P^*_n(x)$ and $Q^*_m(x)$ respectively after reduction to lowest terms. Proofs of the above may be found in books of Achieser [1] or Rice [15].

For the least p-th norm, the proof of existence of a best rational function approximation to a function on an interval can be established as follows. In [1], pp. 10-11, it is shown that the least p-th norm satisfies the Condition E of Young for a polynomial approximating function. Thus Theorem 1-4 of Rice [15] proves existence for the polynomial case. Then from an argument parallel to that of Lemma 3-5 of Rice [15], the result can be extended to rational functions. Moreover, Achieser [1] proves that the best approximation is unique for any strictly normalized function space which establishes uniqueness for L^p if $1 < p < \infty$.

For polynomial approximating functions a well-known theorem of Polya and Jackson (see [15], p. 8) shows that the sequence of best least p-th approximations as $p \to \infty$ contains a convergent subsequence. Further the limit of this subsequence is a best approximation using the Chebyshev norm. Unfortunately, there is little published material concerning rational function approximation using the least p-th norm, particularly the possibility of convergence for increasing p.

However, existence cannot be assured in the case of a rational-function approximation to a function defined on a discrete set for either the Chebyshev or the least p-th norm. Moreover, even if the rational function does exist, it may have poles within the range of approximation. Examples of these conditions may be found in [3] .

3. METHODS OF RATIONAL APPROXIMATION

The statement of the problem immediately suggests one type of approach to a solution; namely, to consider $\|\varepsilon\|$ as a function of its coefficients, R, and to attempt to find a minimum of this function. For the least p-th norm, the space defined by

$$\|\varepsilon\| \leqslant k; \quad \varepsilon^* < k < \infty$$

is strictly convex, a fact which greatly simplifies the task of finding such a minimum. For the Chebyshev norm, the space as defined above is a polytope; thus the methods of convex programming can be used to obtain a best approximation. Minimization of a function can be accomplished on a digital computer by testing the value of this function on a discrete set of points to provide information about the location of a global minimum or local minima. These procedures may be classified by their method of choosing test points as either sequential or nonsequential.

Most sequential methods for convex functions use the gradient of the function as an indication of the direction toward the minimum. Let $r^{(k)}$ be the values of the coefficients excluding q_o, evaluated at the k-th iteration, $h^{(k)}$ be the step size of the k-th step and $D^{(k)}$ be an n_c-1 dimensional direction vector providing the direction of the change of R. The gradient methods to be described all use the iterative equation

$$r^{(k+1)} = r^{(k)} + h^{(k)} \cdot D^{(k)}$$

but differ in their choise of h and D.

(i) Univariate or Relaxation Method

As the name suggests, only one coefficient is changed during each step; hence $D^{(k)}$ is a column vector of the form

$$D^{(k)} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow j^{th} \text{ row}$$

The index j identifying the coefficient to be changed at each step is the one for which $\frac{\partial f}{\partial r_j}$ or $\left(\frac{\partial f}{\partial r_j}\right)^2 \Big/ 2 \frac{\partial^2 f}{\partial r_j^2}$ is the largest in magnitude, where $f = \|\varepsilon\|$. The step size is

$$h^{(k)} = - \frac{\left.\frac{\partial f}{\partial r_j}\right|_{r^{(k)}}}{\left.\frac{\partial^2 f}{\partial r_j^2}\right|_{r^{(k)}}}$$

(ii) Steepest Descent, Newton's and Optimum Gradient Methods

These methods change all the independent variables at each iteration. $D^{(k)}$ can be written in the form

$$D^{(k)} = -\left[B^{-1}\right]\left[\Delta^{(k)}\right]$$

where B is a positive-definite weighting matrix and

$$\Delta^{(k)} = \left[\frac{\partial f}{\partial r_1}\Big|_{r(k)}, \frac{\partial f}{\partial r_2}\Big|_{r(k)}, \cdots, \frac{\partial f}{\partial r_{n_c-1}}\Big|_{r(k)} \right]^t$$

For the Method of Steepest Descent, B is the unit matrix of order n_c-1. Newton's Method uses the Hessian matrix for B. It is defined by

$$b_{ij} = \frac{\partial^2 f}{\partial r_i \partial r_j}$$

The step size may be chosen in a number of ways. One of these is to satisfy the inequality $\|\varepsilon\|^{(k+1)} < \|\varepsilon\|^{(k)}$ for each step. Another is to choose $h^{(k)}$ so that the new point is at the intersection of the co-ordinate axes and the tangent hyperplane to the function at $R^{(k)}$. From analytic geometry, the value of $h^{(k)}$ to achieve this can be calculated from the formula

$$h^{(k)} = \frac{\|\varepsilon\|^{(k)}}{(B^{-1}\,\Delta^{(k)})^t \cdot (B^{-1}\,\Delta^{(k)})}$$

The step size for the optimum gradient method is chosen to minimize $\|\varepsilon\|^{(k+1)}$ along the line $B^{-1}\,\Delta^{(k)}$. An approximation is usually used for this step size [5] since its calculation is often lengthy.

An example of a nonsequential method of minimizing a function is the method of directly testing the value of the function at every point of an n-dimensional grid. The location of the minimum is taken to be the point for which the smallest

value of the function is observed.

Using the purely random method, a certain number of test points are chosen from a volume containing the minimum, with their locations determined by an (n_c-1)-dimensional probability-density function. The smallest value achieved in this sample is considered to be at the location of the minimum. If S is the confidence level that a point of the random distribution falls within a hypercube of side d_1 and the search volume is a hypercube of side d_2, then define

$$a = (d_1/d_2)^{n_c-1}$$

The number of points, p, which should be tested to ensure the minimum found to be within d_1 of the position of the true minimum with confidence level S, assuming a uniform density function, is

$$p = \frac{\log (1 - S)}{\log (1 - a)}$$

More complex methods have been proposed which reduce the calculation time by starting the search on a coarse grid over a large volume (or randomly within this volume), then systematically reduce the volume of the search. Such methods attempt to combine the best features of sequential and nonsequential methods. The amount of calculation required using nonsequential methods increases exponentially with the number of independent variables and the precision of location of the minimum. In return for this increase of calculation time, the function to be minimized by these methods is not required to be differentiable or even continuous throughout the volume.

For the Chebyshev norm, the above methods cannot be used since the required derivatives are not available. Instead, one of the algorithms based on convex programming may be used.

1. The first algorithm to be described was proposed by Loeb. To start this iterative procedure, choose an arbitrary rational function $F(R^{(0)},x)$; then at the k-th step select coefficients $R^{(k)}$ which minimize

$$\max_{x \in [a,b]} \left| \frac{1}{Q^{(k-1)}(x)} \right| \left| f(x)Q^{(k)}(x) - P^{(k)}(x) \right|$$

The trivial solution, $R \equiv 0$, is avoided by fixing one coefficient.

2. A similar algorithm described by Cheney and Loeb [4] differs only in that the function to be minimized is

$$\max_{x \in [a,b]} \left\{ \left| f(x)Q^{(k)}(x) - p^{(k)}(x) \right| - \|\varepsilon\|^{(k-1)} Q^{(k)}(x) \right\}$$

under the restriction that $\left| R_i^{(k)} \right| \leqslant 1$, $1 \leqslant i \leqslant n_c$.

3. The third method of this type was proposed by Loeb [13] and in a slightly modified form by Goldstein [7]. Consider the system of inequalities

$$\left.\begin{aligned} P(x) - (f(x) + M)Q(x) &< 0 \quad \text{all } x \in [a,b] \\ -P(x) + (f(x) - M)Q(x) &< 0 \quad \text{"} \\ \|R\| &= 1 \end{aligned}\right\} I'(M)$$

This system is inconsistent if $M \leqslant M^*$, where M^* is the value of M associated with the best approximation. Choose $b_i^{(0)}$ so that $\sum_{i=0}^{m} b_i^{(0)} x^i \neq 0$, $x \in [a,b]$;

$L^{(0)} = 0$; $H^{(0)} = f(x)$; and $a_i = 0$, $i = 0, 1, \ldots, n$.

At the k-th step

$$M^{(k)} = \frac{L^{(k-1)} + H^{(k-1)}}{2}$$

If the system $I'(M^{(k)})$ is inconsistent, then $L^{(k)} = M^{(k)}$, $H^{(k)} = H^{(k-1)}$ and $R^{(k)} = R^{(k-1)}$. If the system is consistent, then choose $R^{(k)}$ to satisfy it; $L^{(k)} = L^{(k-1)}$ and $H^{(k)} = M^{(k)}$.

This algorithm provides upper and lower bounds for the error of the approximation and these converge monotonically to the value of the best approximation.

To conclude this section, two direct approaches to obtaining a best rational function approximation will be presented.

1. Iteration using the zeroes of the error

This method, developed by Maehly [14], is started by choosing n_c-1 points, x_i, such that $a \leqslant x_1 < x_2 < \ldots x_{n_c-1} \leqslant b$. From this set of points, the coefficients of the rational function which is equal to $f(x)$ at the points x_i are found by solving the system of linear equations

$$f(x_i) \cdot Q(x_i) - P(x_i) = 0 \qquad 1 \leqslant i \leqslant n_c-1$$

Next, modify the x_i so that the maxima of $|\varepsilon(x)|$ will be more nearly equal. One method of adjusting these zeroes is based upon the assumption that the maximum deviation between two adjacent zeroes is approximately proportional to the distance between these zeroes. Hence, the zeroes should be moved closer to the points which have the largest deviation.

2. Iteration using the maxima of the error

Choose an arbitrary set of n_c initial points for the extrema of the error curve such that $a \leqslant x_1 < x_2 < \ldots x_{n_c-1} \leqslant b$, and a value of $E^{(0)}$ for the deviation at these points. An example of an algorithm of this type is Remes Second Algorithm. At the k-th step, attempt to find a rational function which deviates by $E^{(k)}$ from $f(x)$ with alternating sign at the points x_i. The value of $E^{(k)}$ for each succeeding step is found by solving an iterative equation. The error curve is then examined for the location of the largest deviation between adjacent points x_i. These are then taken as the set of points for the next step of the iteration.

4. A NEW METHOD

4-1. Description

The methods which have been described for obtaining best rational function approximations have several limitations for use in filter synthesis. Firstly, they cannot be used to approximate a given phase response since this is a transcendental function of the coefficients of F(s). Secondly, they cannot be used for the simultaneous approximation of two or more functions, for example in minimizing the error in approximation for given magnitude and phase responses. Finally, there is no convenient way to introduce the requirements of physical realizability into the algorithm.

In an attempt to overcome these limitations, a new algorithm was developed which successfully circumvents the first two objections but still does not handle the realizability constraints. A similar method devised by Linvill [11] has the advantage of exposing realizability conditions for simple control but does not lend itself easily to automatic computation, a serious disadvantage because the required calculation can easily be excessive if done manually.

At each step of the procedure a set of "correction functions", $C(\omega)$, is obtained. The correction functions used are the linear terms of a Taylor series expansion of the function of F(s) of interest with respect to all the coefficients of F(s) except the constant term in the denominator. For simplicity in testing this on only the squared-magnitude

function, the derivatives used are taken with respect to the coefficients of $|F(j\omega)|^2$, and F(s) is later obtained by factoring. That linear combination of corrections is found which minimizes the norm of the difference between the linear combination and the error function. To avoid singular equations, the correction function with respect to the term b_o is not used. The linear combination is found most conveniently by considering the given function and the correction functions at a discrete set of frequencies. In this case it is necessary to find a "best" solution to the overdetermined set of linear equations

$$\sum_{j=1}^{n_c-1} C_j(\omega_i)\Delta_j = f(\omega_i) - F(\omega_i) + \beta_i; \quad 1 \leqslant i \leqslant m$$

$$= \varepsilon(\omega_i) + \beta_i$$

where m is the number of discrete frequencies chosen, β is the vector of residuals and Δ is the vector of corrections to the rational function coefficients. If the importance of errors of approximation is greater at some frequencies than at others, a weighting function, $W(\omega)$, may be introduced so that the system of equations becomes

$$W(\omega_i) \sum_{j=1}^{n_c-1} C_j(\omega_i)\Delta_j = W(\omega_i).\varepsilon(\omega_i) + \beta_i; \quad 1 \leqslant i \leqslant m$$

Proof of the existence and uniqueness of a "best" solution of the set of equations which minimizes $\|\beta\|$ for the Chebyshev norm may be found in [8] while it is stated in [9] that the solution also exists for the least p-th norm and that the

series of solutions for increasing p have a convergent subsequence whose limit is the solution for the Chebyshev norm.

After determining the corrections Δ_i, these are then added to the original coefficients R_i. If the corrections are small enough the process is considered to have converged; otherwise the matrices $[C]$ and $[\varepsilon]$ are recalculated with the new coefficients, R_i, and the process is repeated.

4-2. Solution of a Subsidiary Problem

The method just described requires that a "best" solution to a set of overdetermined linear equations be found, that is, a set of values, Δ_j, which will minimize $\|\beta\|$. If the norm is the least-squares norm, the solution is easily calculated by an explicit formula which may be derived in the following way. If $S = \sum_{i=1}^{m} (\beta_i)^2$ is a minimum, then the partial derivatives

$$\frac{\partial S}{\partial \Delta_j} \qquad j = 1,2, \ldots, n_c - 1$$

must be zero. Evaluating these conditions explicitly gives the set of linear equations

$$\sum_{i=1}^{m} (\beta_i) \frac{\partial \beta_i}{\partial \Delta_i} \qquad j = 1,2, \ldots, n_c - 1$$

which may be expanded to

$$\sum_{i=1}^{m} \left[\sum_{k=1}^{n_c - 1} C_k(\omega_i)\Delta_k - \varepsilon(\omega_i) \right] C_j(\omega_i)\Delta_j = 0 \qquad j = 1,2, \ldots, n_c - 1$$

hence

$$\sum_{i=1}^{m}\left[\sum_{k=1}^{n_c-1} C_k(\omega_i)\Delta_k - \varepsilon(\omega_i)\right]C_j(\omega_i) = 0 \qquad j = 1,2,\ldots,n_c-1$$

$$\sum_{i=1}^{m} C_j(\omega_i)\left[\sum_{k=1}^{n_c-1} C_k(\omega_i)\Delta_k\right] = \sum_{i=1}^{m} C_j(\omega_i)\varepsilon(\omega_i) \qquad "$$

which may be rewritten in matrix notation as

$$\left[C^t \cdot C\right]\left[\Delta\right] = \left[C^t \cdot \varepsilon\right]$$

The method which is used for minimizing $M = \max_{1 \leqslant i \leqslant m} |\beta_i|$ is an exhange process proposed by Stiefel [17]. It is based on a theorem originally proved by de la Vallée Poussin which states that the best Chebyshev solution of a system of m linear equations in n unknowns is the best solution of the subset of n+1 of the m equations which maximizes the deviation $\|\beta\|$. The exchange process is started by picking an arbitrary subset of n+1 equations. After finding the best solution for this subset a search is made for a residual β_i whose magnitude is greater than the deviation of the solution for the subset. If none is found, the solution to the subset must be the solution for the entire set of m equations. If one is found, the corresponding equation is systematically substituted for one of the n+1 equations of the original subset. The substitution is done so that at every step the deviation of the best solution increases. The process then starts again with the new subset. Since the maximum deviation is bounded above, and the number of combinations of equations is limited, the algorithm must converge in a finite number of steps to the best Chebyshev solution.

4-3. The Computer Program

For ease of writing, this program has been written in a number of independent segments. The mainline program, called **MASTER**, was written to provide proper logical flow and communication between three subroutines which perform four distinct tasks. The first two, providing an initial approximation by matching the rational function to the desired function at arbitrary points, and evaluating the response of a given rational function on a predetermined set of points, are performed by the subroutine **PMEV**. Subroutine MAIN adjusts the coefficients of the initial rational function so as to reduce the norm (either least-squares or Chebyshev) of the deviation from the given function with the desired weighting. Finally subroutine POLES finds the positions of the poles and zeroes of the resulting rational function and decides upon its physical realizability. Finally the mainline program controls some printing so that the output will appear more readable and in logical order.

i) PMEV

The point matching section of this subroutine is developed as follows. Calling the order of the numerator of the rational function N0 and the order of the denominator N1, the number of free coefficients is N0+N1+1 = N8 since one coefficient can be assumed to be specified in advance for scaling. Therefore if the rational function is equated to the desired function at this number of points the coefficients may be uniquely determined. The equations are derived as follows:

$$f(\omega_i^2) = \frac{\sum_{j=0}^{n} a_j \omega_i^{2j}}{\sum_{j=0}^{m} b_j \omega_i^{2j}}; \quad i = 1,2, \ldots, N8$$

Now assume $b_o = 1$ and multiply by the denominator of the rational function, which gives

$$\sum_{j=0}^{n} a_j \omega_i^{2j} = f(\omega_i^2) \cdot \sum_{j=0}^{m} b_j \omega_i^{2j}; \quad i = 1,2, \ldots, p$$

This may be written in matrix form as

$$A = \begin{bmatrix} 1 & \omega_1^2 & \ldots & \omega_1^{2n} & -f(\omega_1^2)\omega_1^2 & \ldots & -f(\omega_1^2)\,\omega_1^{2m} \\ 1 & \omega_2^2 & \ldots & \omega_2^{2n} & -f(\omega_2^2)\omega_2^2 & \ldots & -f(\omega_2^2)\,\omega_2^{2m} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & \omega_p^2 & \ldots & \omega_p^{2n} & -f(\omega_p^2)\omega_p^2 & \ldots & -f(\omega_p^2)\,\omega_p^{2m} \end{bmatrix}$$

$$[A] \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_m \end{bmatrix} = \begin{bmatrix} f(\omega_1^2) \\ f(\omega_2^2) \\ \vdots \\ f(\omega_p^2) \end{bmatrix}$$

The parts of the program which perform output editing

and evaluation of a given rational function are straightforward and should require no further explanation.

(ii) MAIN

This subroutine is used to minimize the norm of the deviation of the rational function from the desired function. The method used has been described previously. The first step is to calculate the partial derivatives

$$\left.\begin{array}{ll} \dfrac{\partial}{\partial a_j}\left(\left| F(j\omega)\right|^2 \right) & j = 0,1, \ldots, NO \\ \dfrac{\partial}{\partial b_j}\left(\left| F(j\omega)\right|^2 \right) & j = 1,2, \ldots, N1 \end{array}\right\} \quad i = 1,2, \ldots, NOPR$$

where NOPR is the number of points of the discrete set of frequencies. The linear equations are formed by equating the total differentials to the deviations. Both sides of the equation are then multiplied by the weighting factor (a function of frequency). In the program the array of rational function coefficients is called X and the array of the corrections to these coefficients is called DX.

The task of finding the best solution to the above set of equations occupies most of the remainder of subroutine MAIN. When the solution has been found, the corrections are added to the rational-function coefficients. The new rational function is used to calculate a new set of equations and the process is repeated until all the corrections become sufficiently small. Optionally, if a record of the rate of convergence is desired, the rational-function coefficients may be printed out at each step before the new partial derivatives are calculated.

(iii) POLES

Subroutine **POLES** calculates the location of the poles and zeroes of a rational function and stores these in the common area for later printing. The work is divided among three subroutines. The first, called **ROOT**, finds the zeroes in ω^2 of first the denominator and then the numerator of the rational function, both of which are polynomials in ω^2. The second subroutine, called **ACCEPT**, decides if the zeroes found correspond to a physically realizable impedance. This information is also stored in the common storage area for logical routing in the mainline program. Finally, to obtain the actual pole and zero locations, it is necessary to take the square root of the complex numbers found. This is done by subroutine **SQROOT**.

(iv) Auxiliary Programs

APPROX	calculates the squared magnitude function
WEIGHT	calculates the weighting function
DSOLTN	solves a set of linear equations in double precision

5. EXAMPLES

To test the algorithm developed, it has been programmed in Fortran IV for an IBM 7040 computer. The essential parts of this program are included in the Appendix for reference. The results of applying this program to three different magnitude functions and for varying orders of rational function complexity are shown in the accompanying diagrams. With the exception of the one illustrating least-squares fit, the error curves show the alternation of equal positive and negative deviations characteristic of the Chebyshev fit.

Two features of the resulting approximations using the Chebyshev norm are apparent when comparing approximations to amplitude functions with sharp breaks in response with approximations to smooth functions. Firstly, for the same degree of complexity, the maximum error of approximation is usually greater for those functions with abrupt changes of slope than for smooth amplitude functions. Secondly, the reduction of the error of approximation with increasing order of complexity of the rational function is much slower if the function has abrupt changes of slope than otherwise.

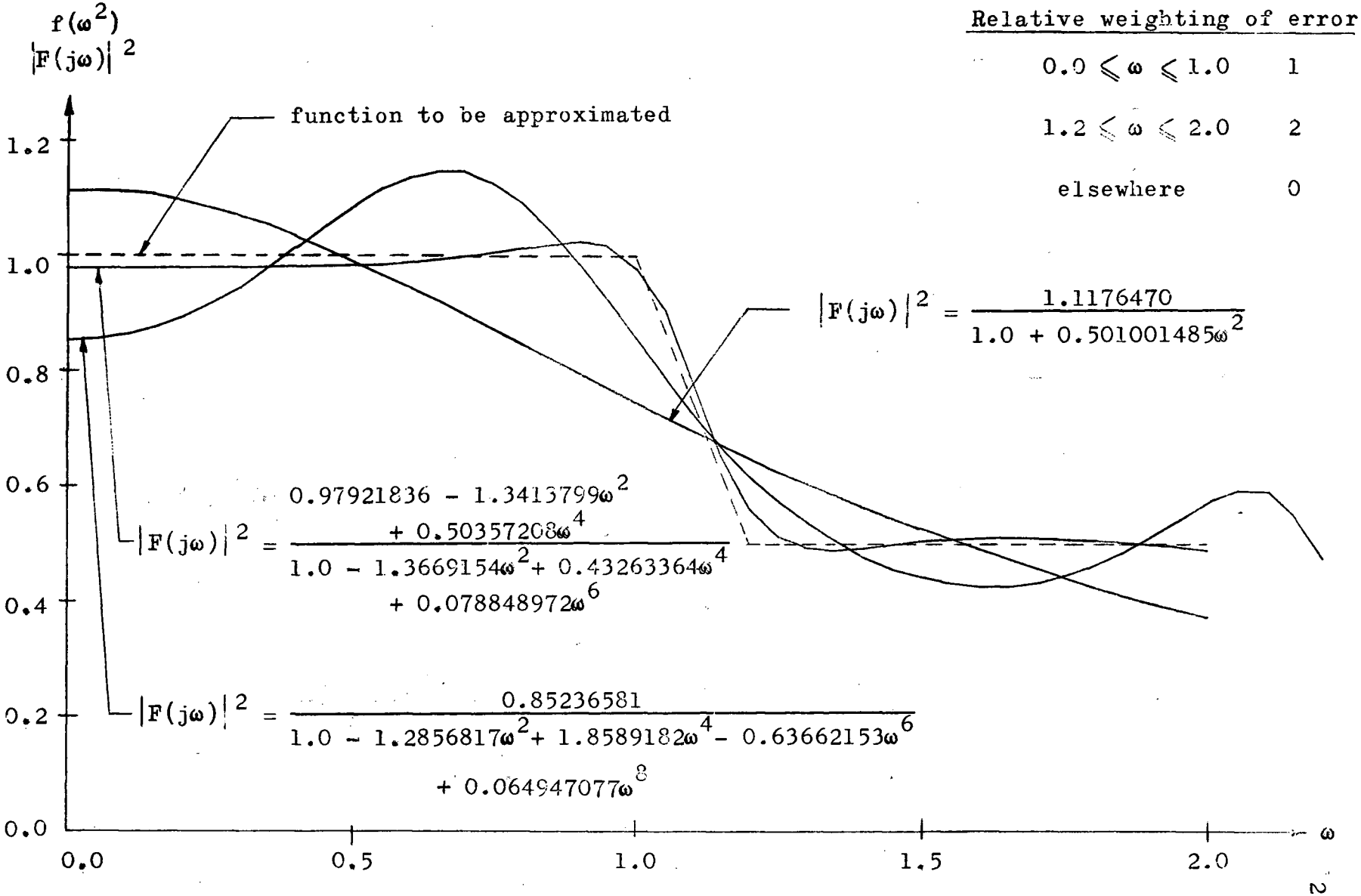

Figure 1. Approximation to Illustrated Response

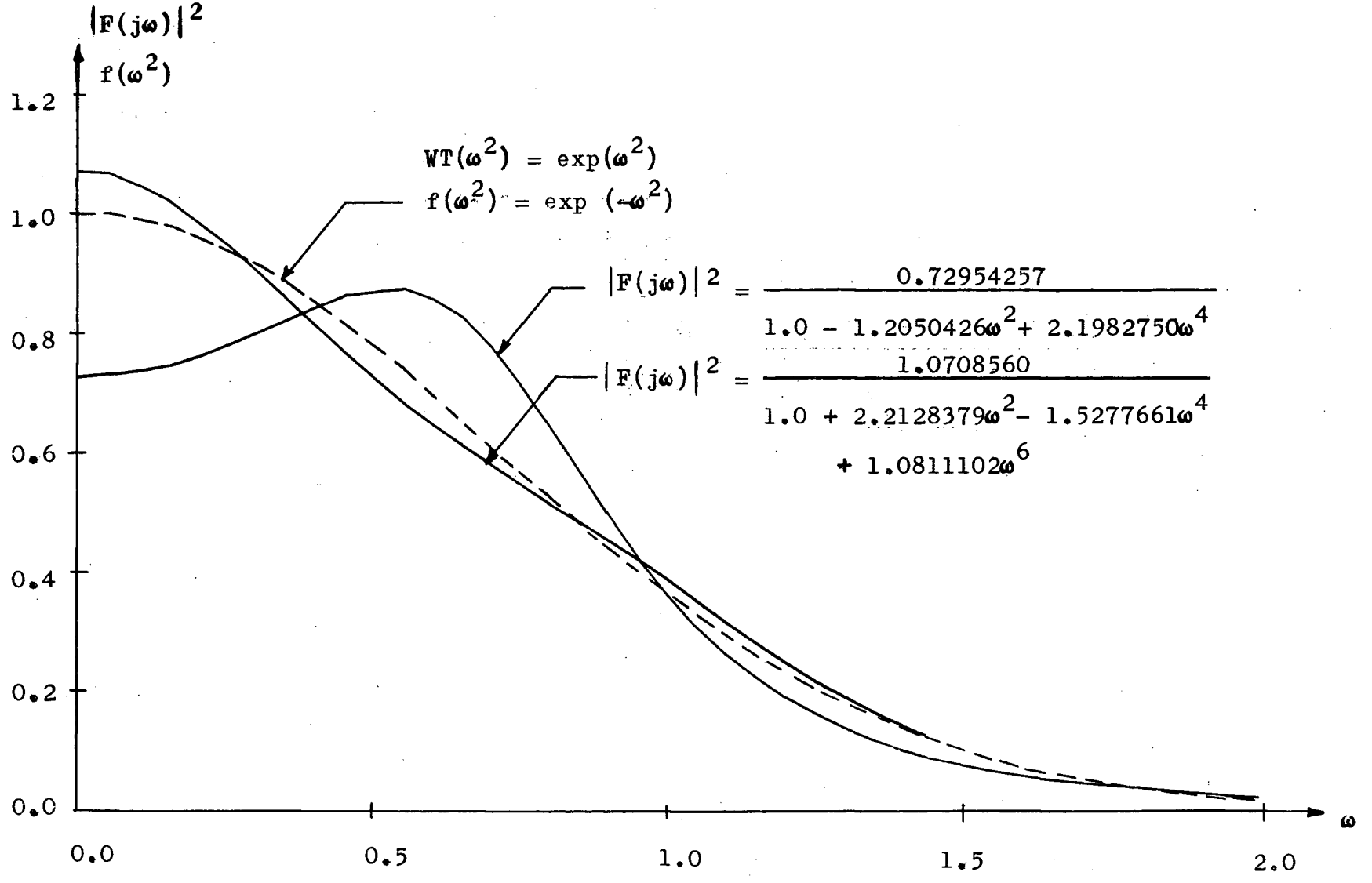

Figure 2. Approximation to Gaussian Response

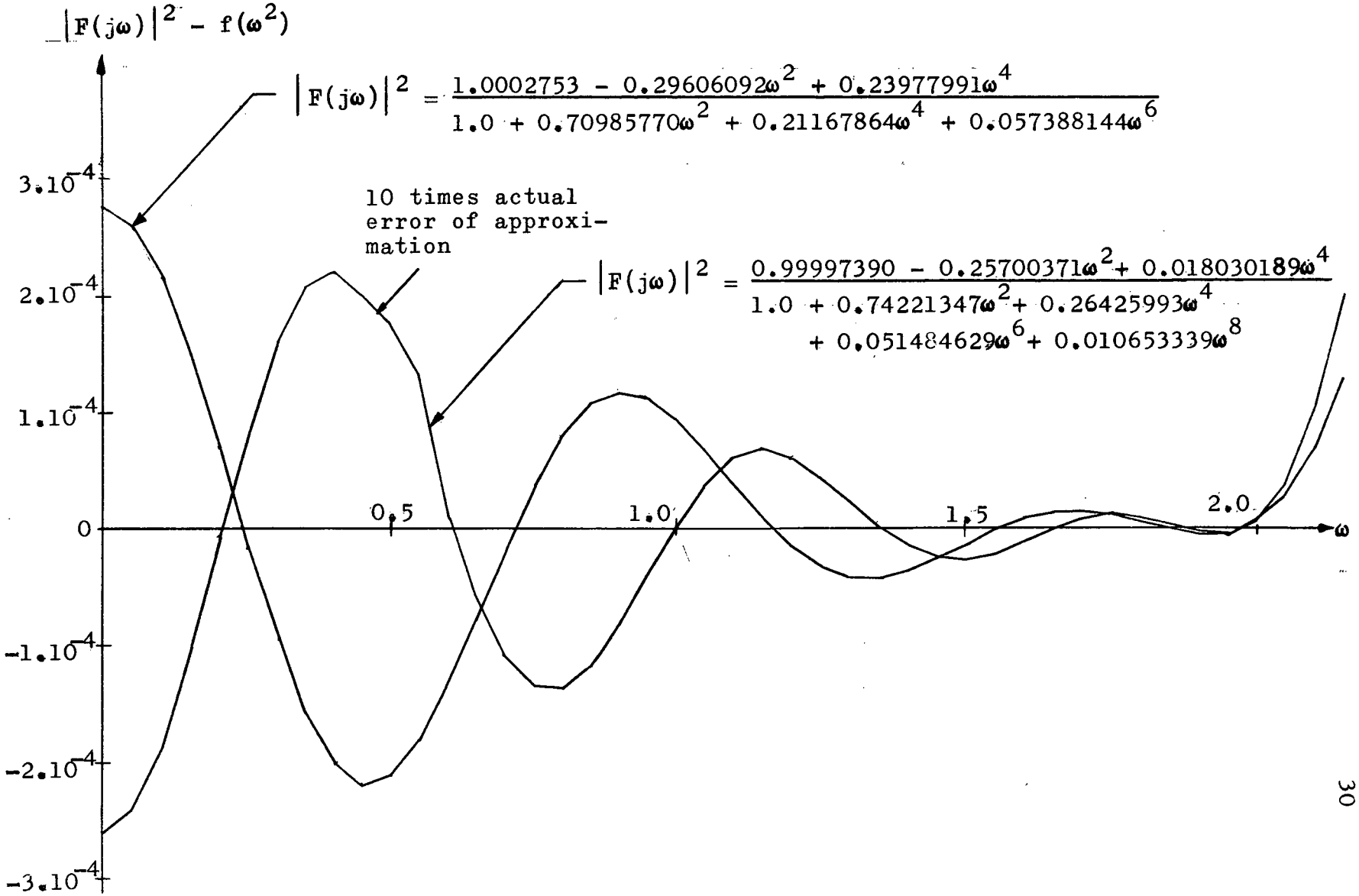

Figure 3. Error in Approximation to Gaussian Response with minimum percentage error for $0 \leqslant \omega \leqslant 2$

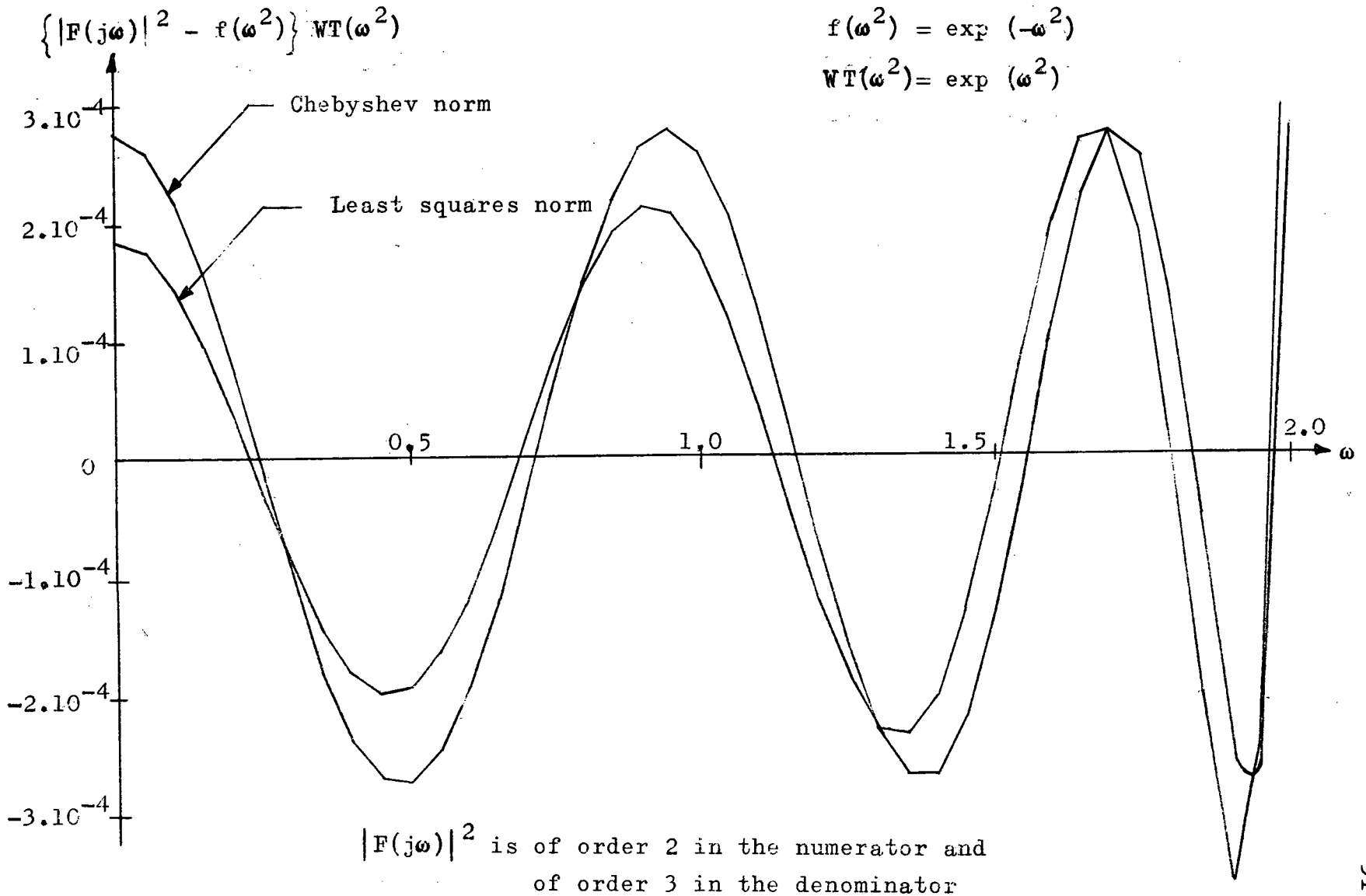

Figure 4. Comparison of Weighted Error in Approximation to Gaussian Response

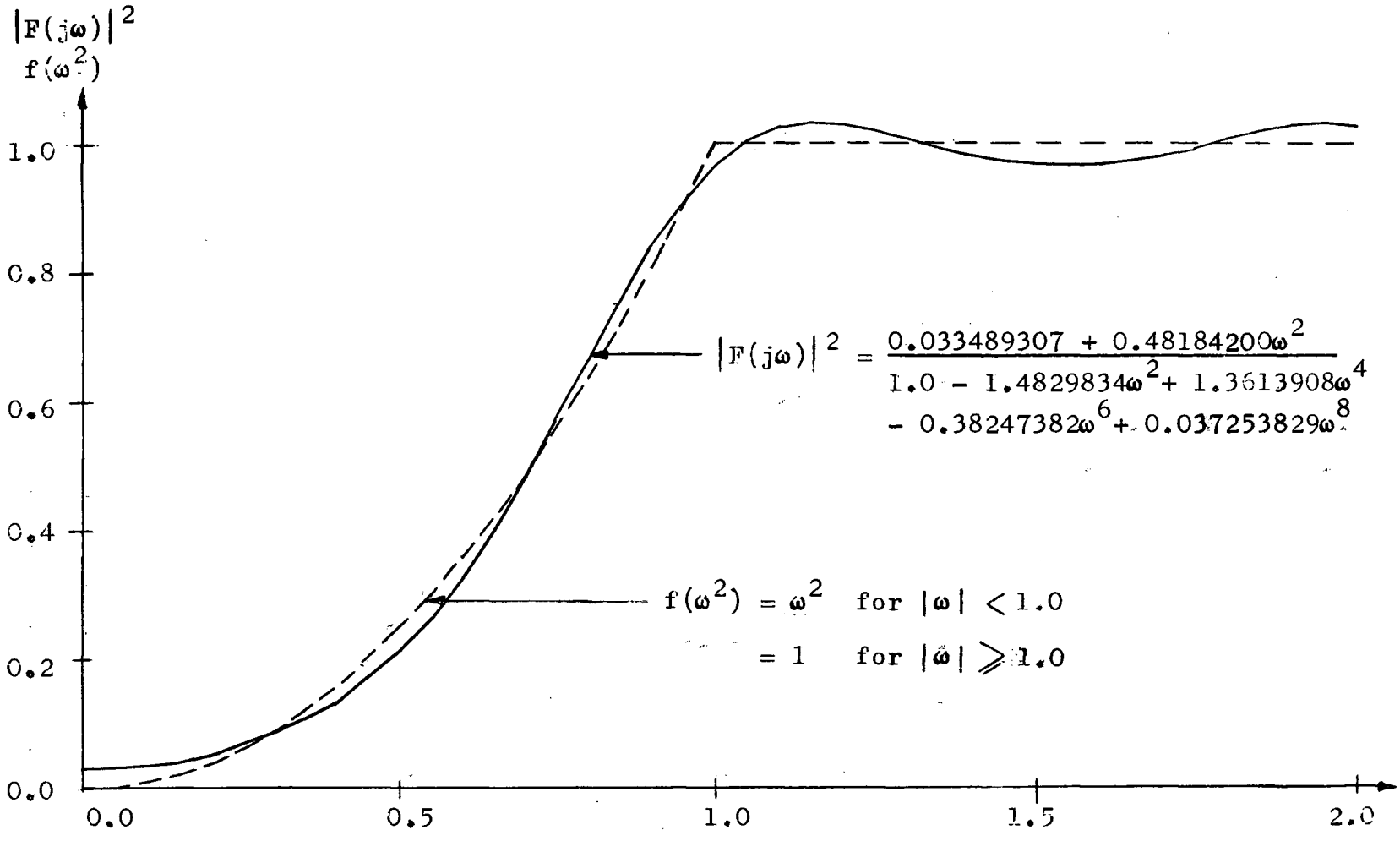

Figure 5. Approximation to Illustrated Response

6. CONCLUSION

The new algorithm for finding "best" rational function approximations has been tested on several different magnitude functions and found to be entirely practical for use on a computer such as the IBM 7040, taking about half a minute to find a best rational function of given degree of complexity and test this for physical realizability. Because of difficulty of programming, the ability of the algorithm to find best simultaneous approximations to given magnitude and phase responses has not yet been tested.

Since this algorithm produces a best fit to a given function on a given set of frequencies, it frequently happens that the resultant is not physically realizable. Probably the most interesting extension of this work would be to reformulate the realizability conditions in such a form that they could be introduced as constraints, allowing calculation of the best realizable approximation with a rational function of specified degree.

APPENDIX

COPY OF FORTRAN IV COMPUTER PROGRAM

FORTRAN SOURCE LIST MASTER

SOURCE STATEMENT

```
C     LOAD SUBROUTINES   PMEV,MAIN,POLES,ACCEPT,ROOT,SQROOT,APPROX,
C                        WEIGHT,DSOLTN
      DOUBLE PRECISION X(16),RR(30),RI(30),RTR(15),RTI(15),
     1     WR1,WR2,WA1,WA2,WA3,WA4,WA5
      LOGICAL ISW1,ISW2,YES,ERROR,SW2,SW5,SWNUM,SWDEN
      COMMON NO,N1,WR1,WR2,WA1,WA2,WA3,WA4,WA5,NOPR,ISW1,ISW2,YES,X,RR,
     1     RI,RTR,RTI,ERROR,N3,N4,N5,N6,N7,N8,N9,SWNUM,SWDEN
C     NO IS THE ORDER OF THE NUMERATOR, N1 IS THE ORDER OF THE DENOM-
C     INATOR.  WR1 AND WR2 ARE THE ENDS OF THE RANGE OF APPROXIMATION.
C     WA1 AND WA2 ARE THE ENDS OF THE RANGE OF INITIAL POINT MATCHING.
C     WA3 AND WA4 ARE THE ENDS OF THE RANGE OF EVALUATION OF THE RESULT
C     AND WA5 IS THE INTERVAL OF THIS EVALUATION.  NOPR IS THE NUMBER OF
C     POINTS FOR BEST FIT.  IF ISW1=.TRUE. THEN LEAST SQUARES FIT, ELSE
C     MINMAX FIT.  ISW2=.FALSE. FOR SUPPRESSION OF INTERMEDIATE PRINT.
C     IF SW2 = .TRUE.  INITIAL PM IS EVALUATED.  IF SW5 = .TRUE.  THEN
C     POLES AND FINAL EVALUATION PRINTED.  IF SW5 = .FALSE.  THESE ARE
C     PRINTED ONLY FOR A PHYSICALLY REALIZABLE FUNCTION.
C     EVALUATE FINAL RATIONAL FUNCTION IF SW5=.TRUE. OR YES=.TRUE.
  100 START = CLOCK(0.)
      READ(5,1) NO,N1,WR1,WR2,WA1,WA2,WA3,WA4,WA5,NOPR,ISW1,ISW2,SW2,SW5
      CALL SKIP TO (1)
      WRITE(6,3)NO,N1,WR1,WR2,WA1,WA2,WA3,WA4,WA5,NOPR,ISW1,ISW2,SW2,SW5
      YES=.FALSE.
C     N3 IS THE NUMBER OF NUMERATOR COEFFICIENTS OF RATIONAL FUNCTION.
      N3=NO+1
C     N5 IS THE POSITION OF THE FIRST  DENOMINATOR COEFFICIENT
      N5=N3+1
C     N6 IS THE POSITION OF THE SECOND DENOMINATOR COEFFICIENT
      N6=N5+1
C     N7 IS THE NUMBER OF DENOMINATOR COEFFICIENTS OF RATIONAL FUNCTION.
      N7=N1+1
      N8=NO+N1
      N9=N8+1
C     N4 IS THE TOTAL NUMBER OF COEFFICIENTS IN THE RATIONAL FUNCTION.
      N4=N9+1
      SWNUM=NO.EQ.0
      SWDEN=N1.EQ.0
C     INITIAL APPROXIMATION BY POINT MATCHING.
      CALL PMEV (.TRUE.,SW2)
      IF (ERROR) GO TO 104
C     REDUCE NORM OF ERROR.
      CALL MAIN
      IF (ERROR) GO TO 104
      IF(X(N3)*X(1).LT.0.) GO TO 101
      IF (X(N4).LT.0.) GO TO 101
C     FIND THE LOCATION OF POLES AND ZEROS AND CHECK FOR PHYSICAL
C     REALIZABILITY.  IF THE RATIONAL FUNCTION IS REALIZABLE, YES=.TRUE.
  103 CALL POLES
      GO TO 102
  101 IF (SW5) GO TO 103
  102 SW2=SW5.OR.YES
C     EVALUATE FINAL RATIONAL FUNCTION IF SW5=.TRUE. OR YES=.TRUE.
      CALL PMEV (.FALSE.,SW2)
C     OUTPUT REALIZABILITY DECISION.
      IF (.NOT.YES) WRITE (6,2)
```

FORTRAN SOURCE LIST MASTER

```
      SOURCE STATEMENT

      IF (YES)      WRITE (6,11)
      IF (.NOT.SW2) GO TO 104
      L=0
C     BEGIN PRINTING POLES AND ZEROS.
      IF (SWNUM) GO TO 105
      WRITE (6,9)
      WRITE (6,7)
      L=2*NO
      WRITE (6,8) (RI(I),RR(I),I=1,L)
  105 IF (SWDEN) GO TO 104
      WRITE (6,6)
      WRITE (6,7)
      M=L+2*N1
      L=L+1
      WRITE (6,8) (RI(I),RR(I),I=L,M)
  104 TOTAL=CLOCK(START)/60.
      WRITE (6,10) TOTAL
C     CHECK FOR MORE INPUT DATA.
      GO TO 100
    1 FORMAT (2I3,7F5.2,I4,4L3)
    2 FORMAT (44HOTHE IMPEDANCE IS NOT PHYSICALLY REALIZABLE.//)
    3 FORMAT (80H  NO   N1   WR1    WR2    WA1    WA2    WA3    WA4    WA5    NUP
     1R   ISW1   ISW2    SW3    SW5//1X,I3,I4,7F6.3,I5,4L6//)
    6 FORMAT (/39X,5HPOLES)
    7 FORMAT (18X,5HSIGMA,35X,5HOMEGA/)
    8 FORMAT (D33.16,D40.16)
    9 FORMAT (/39X,5HZEROS)
   10 FORMAT (17HOELAPSED TIME WAS,F7.2,8H SECONDS)
   11 FORMAT (40HOTHE IMPEDANCE IS PHYSICALLY REALIZABLE.//)
      END
```

FORTRAN SOURCE LIST PMEV

```
      SOURCE STATEMENT

      SUBROUTINE PMEV (SW1,SW2)
      DOUBLE PRECISION X(16),RR(30),RI(30),RTR(15),RTI(15),WR1,WR2,WA1,
     1     WA2,WA3,WA4,WA5,A(15,15),W(15),WSQ(15),Y(15),WT(1),FUNC(1),
     2     XN,NUM,DEN,F,DEV,WTDEV
      LOGICAL ISW1,ISW2,YES,ERROR,SWNUM,SWDEN,SW1,SW2,INDEX
      COMMON NO,N1,WR1,WR2,WA1,WA2,WA3,WA4,WA5,NOPR,ISW1,ISW2,YES,X,RR,
     1     RI,RTR,RTI,ERROR,N3,N4,N5,N6,N7,N8,N9,SWNUM,SWDEN
C     IF SW1=.TRUE. PROGRAM DOES POINT MATCHING, OTHERWISE BYPASSES THIS
C     PART OF THE PROGRAM.  IF SW2=.TRUE. PROGRAM DOES EVALUATION OF
C     RATIONAL FUNCTION.
      IF (.NOT.SW1) GO TO 200
C     THIS SECTION FINDS A RATIONAL FUNCTION WHICH IS EQUAL TO A GIVEN
C     FUNCTION (APPROX) AT (NO+N1+1) POINTS.  EQUATING THE TWO AT THIS
C     NUMBER OF POINTS AND ASSUMING A VALUE (1.) FOR THE CONSTANT TERM
C     IN THE DENOMINATOR PROVIDES A SET OF LINEAR EQUATIONS WHOSE
C     SOLUTION GIVES THE DESIRED RATIONAL FUNCTION COEFFICIENTS.
      XN=N8
      XN=(WA2-WA1)/XN
      W(1)=WA1
      WSQ(1)=WA1**2
      DO 150 I=2,N9
      W(I)=W(I-1) + XN
  150 WSQ(I)=W(I)**2
      CALL APPROX (N9,W,X)
      WRITE (6,3) (W(I),I=1,N9)
      DO 152 I=1,N9
      A(I,1)=1.D0
      IF (SWNUM) GO TO 101
      DO 153 J=2,N3
  153 A(I,J) = A(I,J-1)*WSQ(I)
  101 IF (SWDEN) GO TO 152
      A(I,N5) = -X(I)*WSQ(I)
      IF (N1.EQ.1) GO TO 152
      DO 154 J=N6,N9
  154 A(I,J) = A(I,J-1)*WSQ(I)
  152 CONTINUE
      CALL DSOLTN (A,X,N9,15,XN)
      IF (XN .EQ.0.) GO TO 300
C     IF THE ORDER OF THE DENOMINATOR IS GREATER THAN ZERO, SHIFT THE
C     COEFFICIENTS FOUND AND INSERT THE ASSUMED VALUE OF 1. FOR X(N5).
      IF (SWDEN) GO TO 103
      I=N4
  155 X(I)=X(I-1)
      I=I-1
      IF (I.GE.N6) GO TO 155
  103 X(N5)=1.D0
C     BEGIN EDITING FOR OUTPUT.
  200 IF (NO.GT.N1) GO TO 201
      LESS=N3
      INDEX=.TRUE.
      GO TO 202
  201 LESS=N7
      INDEX=.FALSE.
  202 DO 250 I=1,N7
      J=I+N3
```

FORTRAN SOURCE LIST PMEV

SOURCE STATEMENT

```
  250 Y(I)=X(J)
      WRITE (6,4) (X(I),Y(I),I=1,LESS)
      LESS=LESS+1
      IF (.NOT.INDEX)          WRITE (6,6) (X(I),I=LESS,N3)
      IF (NO.NE.N1.AND.INDEX) WRITE (6,5) (Y(I),I=LESS,N7)
      IF (.NOT.SW2) RETURN
C     BEGIN EVALUATION OF RATIONAL FUNCTION.
      NOP=(WA4-WA3)/WA5 + 1.DO
      WRITE (6,8)
      DO 251 I=1,NOP
      XN=I-1
      W(1)=WA3+XN*WA5
C     APPROX GIVES DESIRED VALUE IN ORDER TO CALCULATE DEVIATION.
      CALL APPROX (1,W,FUNC)
      CALL WEIGHT (1,W,WT)
      XN=W(1)**2
      NUM=X(N3)
      IF (SWNUM) GO TO 203
      DO 252 J=2,N3
      K=N3+1-J
  252 NUM=NUM*XN+X(K)
  203 DEN=Y(N7)
      IF (SWDEN) GO TO 204
      DO 253 J=2,N7
      K=N7+1-J
  253 DEN=DEN*XN+Y(K)
  204 F=NUM/DEN
      DEV=F-FUNC(1)
      WTDEV=DEV*WT(1)
  251 WRITE (6,9) W(1),F,DEV,WTDEV
      RETURN
  300 WRITE (6,10)
      ERROR=.TRUE.
      RETURN
    3 FORMAT (18X,43HPOINT MATCHING AT THE FOLLOWING FREQUENCIES/
     1      (D53.16))
    4 FORMAT (//14X,22HNUMERATOR COEFFICIENTS,12X,24HDENOMINATOR COEFFIC
     1IENTS/(2D36.16))
    5 FORMAT (D72.16)
    6 FORMAT (D36.16)
    8 FORMAT (/5X,5HOMEGA,14X,8HRESPONSE,18X,9HDEVIATION,9X,
     1      18HWEIGHTED DEVIATION/)
    9 FORMAT (1X,F9.3,3D27.16)
   10 FORMAT (14HOERROR IN PMEV)
      END
```

FORTRAN SOURCE LIST MAIN

```
      SOURCE STATEMENT

      SUBROUTINE MAIN
      REAL SIGMA(16)
      DOUBLE PRECISION X(16),RR(30),RI(30),RTR(15),RTI(15),WR1,WR2,WA1,
     1     WA2,WA3,WA4,WA5,A(201,15),B(201),DX(16),W(240),FUNC(240),
     2     WT(240),R(16,16),WR3,NUM,DEN,D,BIG,SIG,F1,F2,SM,BI,DET,E(201)
     3     ,LAMBDA(16),MU(16)
      DOUBLE PRECISION DABS
      INTEGER IA(16)
      LOGICAL ISW1,ISW2,YES,ERROR,JY,SWNUM,SWDEN
      COMMON NO,N1,WR1,WR2,WA1,WA2,WA3,WA4,WA5,NOPR,ISW1,ISW2,YES,X,RR,
     1     RI,RTR,RTI,ERROR,N3,N4,N5,N6,N7,N8,N9,SWNUM,SWDEN
      ERROR=.FALSE.
C     IRUN1 COUNTS THE NUMBER OF LINEAR EQUATION ADJUSTMENTS
      IRUN1=0
      NNP=N4
      WR3=NOPR-1
      WR3=(WR2-WR1)/WR3
      N1=N1-1
      N4=N4-1
      W(1)=WR1
      DO 450 I=2,NOPR
  450 W(I)=W(I-1)+WR3
C     GENERATE SQUARED MAGNITUDE FUNCTION AND WEIGHTING FUNCTION
      CALL APPROX (NOPR,W,FUNC)
      CALL WEIGHT (NOPR,W,WT)
  600 IRUN1=IRUN1+1
C     IRUN2 COUNTS THE NUMBER OF EXCHANGES REQUIRED TO FIND CHEBYSHEV
C     SOLUTION TO OVERDETERMINED EQUATIONS.
      IRUN2=0
      NOP=0
C     BEGIN ITERATIVE PROCEDURE.  DO LOOP 650 CALCULATES THE EQUIVALENT
C     OVERDETERMINED LINEARIZED EQUATIONS
      DO 650 I=1,NOPR
C     IF WEIGHTING IS ZERO DO NOT FORM THE ASSOCIATED TRIVIAL EQUATION.
      IF (WT(I).EQ.0.) GO TO 650
C     RESTORE N1 AND N4 TO THEIR ORIGINAL VALUES FOR THIS SECTION.
      N1=N1+1
      N4=N4+1
      NOP=NOP+1
      WSQ=W(I)**2
      NUM=X(N3)
      IF (SWNUM) GO TO 601
      DO 651 J=1,NO
      K=N3-J
  651 NUM=NUM*WSQ+X(K)
  601 DEN=X(N4)
      IF (SWDEN) GO TO 602
      DO 652 J=1,N1
      K=N4-J
  652 DEN=DEN*WSQ+X(K)
C     THE MATRIX A(I,J) CONSISTS OF THE PARTIAL DERIVATIVES OF THE
C     RATIONAL FUNCTION AT FREQUENCY W(I) WITH RESPECT TO THE
C     COEFFICIENTS X(J) ALL MULTIPLIED BY WEIGHT WT(I) EXCEPT THAT
C     DERIVATIVES WITH RESPECT TO X(N5) ARE OMITTED
  602 A(NOP,1)=WT(I)/DEN
```

FORTRAN SOURCE LIST MAIN

```
      SOURCE STATEMENT

           N1=N1-1
           N4=N4-1
           IF (SWNUM) GO TO 603
           DO 653 J=2,N3
       653 A(NOP,J)=A(NOP,J-1)*WSQ
       603 IF (SWDEN) GO TO 604
           A(NOP,N5)=-WT(I)*WSQ*NUM/DEN**2
           IF (N1.EQ.0) GO TO 604
           DO 654 J=N6,N4
       654 A(NOP,J)=A(NOP,J-1)*WSQ
      C    THE MATRIX B(I) CONSISTS OF THE DIFFERENCE BETWEEN THE DESIRED
      C    RESPONSE AND THE RESPONSE OF THE CURRENT RATIONAL FUNCTION
      C    AT OMEGA = W(I) ALL MULTIPLIED BY THE WEIGHTING FUNCTION.
       604 B(NOP)=WT(I)*(FUNC(I)-NUM/DEN)
       650 CONTINUE
      C    ON THE FIRST PASS, FIND LEAST SQUARES FIT TO EQUATIONS AND
      C    CORRESPONDING LARGEST RESIDUALS.  ON SUBSEQUENT PASSES, USE THE
      C    VALUES OF IA( ) FROM THE PREVIOUS PASS.
           IF (IRUN1.NE.1.AND..NOT.ISW1) GO TO 200
           IF (NOP.GT.N4) GO TO 606
           DO 655 I=1,N4
           DX(I)=B(I)
           DO 655 J=1,N4
       655 R(I,J)=A(I,J)
           CALL DSOLTN (R,DX,N4,16,DET)
           IF (DET .EQ. 0.) GO TO 820
           GO TO 800
       606 DO 656 I=1,N4
           DO 656 J=1,N4
           IF (I.GT.J) GO TO 607
           R(I,J)=0.
           DO 657 K=1,NOP
       657 R(I,J)=R(I,J)+A(K,I)*A(K,J)
           GO TO 656
       607 R(I,J)=R(J,I)
       656 CONTINUE
           DO 658 I=1,N4
           DX(I)=0.
           DO 658 J=1,NOP
       658 DX(I)=DX(I)+A(J,I)*B(J)
           CALL DSOLTN (R,DX,N4,16,DET)
           IF (DET.EQ.0.) GO TO 820
           IF (.NOT.ISW1) GO TO 608
           IRUN2=-1
           GO TO 800
       608 DO 659 I=1,NOP
           E(I)=-B(I)
           DO 660 J=1,N4
       660 E(I)=E(I)+A(I,J)*DX(J)
           IF (I.GT.NNP) GO TO 610
           IF (I.GT.1) GO TO 609
           IA(1)=1
           GO TO 659
       609 K=I-1
           GO TO 611
```

FORTRAN SOURCE LIST MAIN

SOURCE STATEMENT

```
  610 K=NNP
  611 DO 661 L=1,K
      LD=IA(L)
      IF (DABS(E(LD)).LE.DABS(E(I))) GO TO 612
  661 CONTINUE
      IF (I.LE.NNP) IA(I)=I
  659 CONTINUE
      GO TO 200
  612 M=K+1
      IF (I.GT.NNP) M=M-1
  613 IA(M)=IA(M-1)
      M=M-1
      IF (M.GT.L) GO TO 613
      IA(L)=I
      GO TO 659
C     SECTION CALCULATES BEST FIT TO NNP GIVEN EQUATIONS.  LOCATION OF
C     EQUATIONS IS GIVEN BY THE VECTOR IA( ), SIGN OF DEVIATIONS GIVEN
C     BY VECTOR SIGMA( ), THE COEFFICIENTS OF THE LINEAR COMBINATION OF
C     ROWS ARE LAMBDA( ).  THIS SECTION FOLLOWS RICE, PG. 174.
C     FIRST CALCULATE REFERENCE DEVIATION, D.
  200 ID=IA(NNP)
      DO 250 J=1,N4
      JD=IA(J)
      LAMBDA(J)=-A(ID,J)
      DO 250 I=1,N4
  250 R(I,J)=A(JD,I)
      CALL DSOLTN (R,LAMBDA,N4,16,DET)
      IF (DET.EQ.0.) GO TO 820
      LAMBDA(NNP)=1.
  201 IRUN2=IRUN2+1
      NUM=0.
      DEN=0.
      DO 251 I=1,NNP
      SIGMA(I)=1.
      IF (LAMBDA(I).LT.0.) SIGMA(I)=-1.
      ID=IA(I)
      NUM=NUM-B(ID)*LAMBDA(I)
  251 DEN=DEN+DABS(LAMBDA(I))
      D=NUM/DEN
C     USING THIS VALUE FOR THE REFERENCE DEVIATION, CALCULATE THE
C     CORRESPONDING SOLUTION, DX( ), GIVING THE BEST CHEBYSHEV FIT TO
C     THE NNP EQUATIONS OF IA( ).
      DO 252 I=1,N4
      ID=IA(I)
      DX(I)=B(ID)+SIGMA(I)*D
      DO 252 J=1,N4
  252 R(I,J)=A(ID,J)
      CALL DSOLTN (R,DX,N4,16,DET)
      IF (DET.EQ.0.) GO TO 820
C     CALCULATE VECTOR E = A . DX - B  AND FIND VALUE AND POSITION OF
C     ELEMENT OF LARGEST MAGNITUDE.
      BIG=0.
      DO 253 I=1,NOP
      E(I)=-B(I)
      DO 254 J=1,N4
```

FORTRAN SOURCE LIST MAIN

SOURCE STATEMENT

```
  254 E(I)=E(I)+A(I,J)*DX(J)
      IF (DABS(E(I)).LE.BIG) GO TO 253
      BIG=DABS(E(I))
      MAX=I
  253 CONTINUE
C     CHECK IF THIS LARGEST ELEMENT IS CONTAINED IN THE SET IA( ).
      DO 255 I=1,NNP
      IF (MAX.EQ.IA(I)) GO TO 800
  255 CONTINUE
C     IF THE LARGEST ELEMENT IS NOT IN THE SUBSET IA( ), THIS MAY HAVE
C     OCCURRED BECAUSE OF ROUNDING ERRORS.  THEREFORE, IF DEVIATION IS
C     CLOSE TO D, THEN ACCEPT THIS AS A SOLUTION.
      IF (BIG.LE.1.000001D0*D) GO TO 800
C     IF NOT A SOLUTION, IT WILL BE NECESSARY TO TRY A NEW SET OF
C     EQUATIONS.IA( ).  THIS IS DONE BY EXCHANGING ONE OF THE ORIGINAL
C     SET BY THE LARGEST CURRENT DEVIATION.  FOLLOWING RICE, EQUATION
C     (6-8.12) OF PG. 174 FIND COEFFICIENTS MU( ).
      SIG=1.
      IF (E(MAX).LT.0.) SIG=-1.
      ID=IA(NNP)
      DO 256 J=1,N4
      JD=IA(J)
      MU(J)=-SIG*A(MAX,J)-A(ID,J)
      DO 256 I=1,N4
  256 R(I,J)=A(JD,I)
      CALL DSOLTN (R,MU,N4,16,DET)
      IF (DET.EQ.0.) GO TO 820
      MU(NNP)=1.
      F1=-SIG*B(MAX)
      F2=0.
      DO 257 I=1,NNP
      ID=IA(I)
      F1=F1-MU(I)*B(ID)
  257 F2=F2+LAMBDA(I)*B(ID)
C     NOW EVALUATE THE ABSOLUTE VALUE OF THE DEVIATIONS THAT WOULD
C     RESULT FOR THE NEXT STEP WITH EACH POSSIBLE EXCHANGE AND SELECT
C     THE LARGEST OF THESE.
      BIG=0.
      DO 258 J=1,NNP
      SM=MU(J)/LAMBDA(J)
      DEN=1.
      DO 259 I=1,NNP
      IF (I.NE.J) DEN=DEN+DABS(MU(I)-LAMBDA(I)*SM)
  259 CONTINUE
      BI=DABS((F1+F2*SM)/DEN)
      IF (BI.LE.BIG) GO TO 258
      BIG=BI
      MIN=J
  258 CONTINUE
      IA(MIN)=MAX
C     TO OBTAIN LAMBDA( ) FOR OTHER THAN THE FIRST STEP IT IS NOT
C     NECESSARY TO SOLVE A SET OF LINEAR EQUATIONS, BUT INSTEAD USE
C     EQUATION (6-8.15) WITH J = MIN.
      NUM=MU(MIN)/LAMBDA(MIN)
      DO 260 I=1,NNP
```

FORTRAN SOURCE LIST MAIN

```
      SOURCE STATEMENT

      IF(I.NE.MIN) LAMBDA(I)=LAMBDA(I)*LAMBDA(MIN)*(MU(I)/LAMBDA(I)-NUM)
  260 CONTINUE
      LAMBDA(MIN)=SIG*LAMBDA(MIN)
C     NORMALIZE COEFFICIENTS LAMBDA( ) TO PREVENT FLOATING POINT TRAPS.
      BIG=0.
      DO 261 I=1,NNP
      BI=DABS(LAMBDA(I))
  261 IF (BI.GT.BIG) BIG=BI
      DO 262 I=1,NNP
  262 LAMBDA(I)=LAMBDA(I)/BIG
      GO TO 201
C     OUTPUT SECTION.
  800 J=N4+1
  850 DX(J)=DX(J-1)
      J=J-1
      IF (J.GE.N6) GO TO 850
      N4=N4+1
      DX(N5)=0.
      JY=.FALSE.
C     CHECK IF THE ABSOLUTE VALUES OF THE CORRECTIONS TO THE RATIONAL
C     FUNCTION COEFFIECIENTS ARE CONVERGING.  IF NOT, REVISE THE SET OF
C     LINEAR EQUATIONS.
      DO 851 I=1,N4
      IF (DABS(DX(I)).GT.1.D-6*DABS(X(I))) JY=.TRUE.
  851 X(I)=X(I)+DX(I)
      IF (.NOT.JY) GO TO 802
      IF (.NOT.ISW2) GO TO 801
      WRITE (6,1) IRUN2
      WRITE (6,5) (X(I),I=1,N3)
      WRITE (6,2)
      WRITE (6,5) (X(I),I=N5,N4)
  801 N4=N4-1
      GO TO 600
  802 N1=N1+1
      D=100.*D
      WRITE (6,3) D,IRUN1
      IF (ISW2) WRITE (7,4) NO,N1,(X(I),I=1,N4)
      RETURN
  820 ERROR=.TRUE.
      RETURN
    1 FORMAT (23H-NUMERATOR COEFFICIENTS,8X,8HIRUN2 = ,I4/)
    2 FORMAT (25H-DENOMINATOR COEFFICIENTS/)
    3 FORMAT (22HOWEIGHTED DEVIATION IS,F8.3///20X,14HFINAL SOLUTION,
     1     16X,8HIRUN1 = ,I4)
    4 FORMAT (2I3/(6D12))
    5 FORMAT (D24.16)
      END
```

FORTRAN SOURCE LIST POLES

```
      SOURCE STATEMENT

      SUBROUTINE POLES
      DOUBLE PRECISION X(16),RR(30),RI(30),RTR(15),RTI(15),WR1,WR2,WA1,
     1     WA2,WA3,WA4,WA5,A(16),B(16),S(16)
      LOGICAL ISW1,ISW2,YES,ERROR,SWNUM,SWDEN
      COMMON NO,N1,WR1,WR2,WA1,WA2,WA3,WA4,WA5,NOPR,ISW1,ISW2,YES,X,RR,
     1     RI,RTR,RTI,ERROR,N3,N4,N5,N6,N7,N8,N9,SWNUM,SWDEN
      IDD=2*N1
C     PLACE COEFFICIENTS IN TEMPORARY LOCATION.
      DO 150 I=1,N4
  150 S(I)=X(I)
C     PLACE NUMERATOR COEFFICIENTS IN REVERSE ORDER IN A( ).
      DO 151 I=1,N3
      J=N5-I
  151 A(J)=S(I)
C     PLACE DENOMINATOR COEFFICIENTS IN REVERSE ORDER IN B( ).
      DO 152 I=1,N7
      J=N7+1-I
      K=I+N3
  152 B(J)=S(K)
      IF (SWDEN) GO TO 100
C     SUBROUTINE ROOT FINDS ZEROS OF OMEGA SQUARED OF DENOMINATOR
C     POLYNOMIAL.
      CALL ROOT (N1,B)
C     CHECK PHYSICAL REALIZABILITY.
      CALL ACCEPT (N1)
C     FIND ZEROS OF OMEGA BY TAKING COMPLEX SQUARE ROOT OF ZEROS ABOVE.
      CALL SQROOT (N1)
C     CHECK IF NECESSARY TO FIND ZEROS OF RATIONAL FUNCTION.
  100 IF (SWNUM) GO TO 101
C     IF SO, SHIFT POLE POSITIONS TO ALLOW ROOM FOR ZERO POSITIONS.
      J=IDD+2*NO
      I=IDD
  102 RR(J)=RR(I)
      RI(J)=RI(I)
      J=J-1
      I=I-1
      IF (I.GT.0) GO TO 102
C     BEGIN ZERO LOCATION AND REALIZABILITY TEST OF NUMERATOR.
      CALL ROOT (NO,A)
      CALL ACCEPT (NO)
      CALL SQROOT (NO)
  101 RETURN
      END
```

FORTRAN SOURCE LIST ROOT

```
      SOURCE STATEMENT

      SUBROUTINE ROOT (M,B)
      DOUBLE PRECISION D(16),RR(30),RI(30),RTR(15),RTI(15),WR1,WR2,WA1,
     1     WA2,WA3,WA4,WA5,A(16,2),B(16),C(16),Q(16),TOL(2),R,S,R1,S1,T,
     2     X,Z,DR,DS
      DOUBLE PRECISION DABS,DSQRT
      LOGICAL ISW1,ISW2,YES,ERROR,SWNUM,SWDEN,IM,IND
      COMMON NO,N1,WR1,WR2,WA1,WA2,WA3,WA4,WA5,NOPR,ISW1,ISW2,YES,D,RR,
     1     RI,RTR,RTI,ERROR,N3,N4,N5,N6,N7,N8,N9,SWNUM,SWDEN
C     M IS THE ORDER OF THE POLYNOMIAL, N IS THE NUMBER OF TERMS IN
C     THE REDUCED POLYNOMIAL.
      N=M+1
      DO 150 I=1,N
      A(I,1)=B(I)
  150 A(I,2)=B(I)
C     TOL(1) IS THE TOLERANCE USED WITH THE REDUCED POLYNOMIAL AND
C     TOL(2) FOR FULL POLYNOMIAL.
      TOL(1)=1.D-6
      TOL(2)=1.D-12
      L=0
C     TO FIND ROOTS APPROXIMATELY IN ORDER OF INCREASING MODULUS, START
C     WITH INITIAL APPROXIMATION FOR FIRST ROOT OF ZERO.
      R1=0.
      S1=0.
      IM=.FALSE.
C     THE VALUE OF IN DETERMINES IF THE ROOT IS TO BE EXTRACTED FROM
C     THE ORIGINAL POLYNOMIAL OR THE REDUCED POLYNOMIAL, IN = 1 FOR
C     REDUCED POLYNOMIAL.  IM = .TRUE. WHEN THE LAST ROOTS ARE BEING
C     FOUND.
  100 IF (N.LT.4) GO TO 112
C     BEGIN BAIRSTOW*S METHOD WHICH FINDS QUADRATIC FACTORS, X**2 + R*X
C     + S, OF A POLYNOMIAL.
      IN=1
  115 K=0
      IND=IN.EQ.1
C     BEGIN SYNTHETIC DIVISION OF TRIAL FACTOR INTO POLYNOMIAL.
C     BEGIN SYNTHETIC DIVISION OF TRIAL FACTOR INTO POLYNOMIAL.
      R=R1
      S=S1
      Q(1)=A(1,IN)
      C(1)=Q(1)
  101 Q(2)=A(2,IN)-R*Q(1)
      C(2)=Q(2)-R*C(1)
C     RANGE OF LOOP DEPENDS UPON VALUE OF IN.
      IF (IND) NQ=N
      IF (.NOT.IND) NQ=M+1
      DO 151 I=3,NQ
      Q(I)=A(I,IN)-R*Q(I-1)-S*Q(I-2)
  151 C(I)=Q(I)-R*C(I-1)-S*C(I-2)
C     BEGIN CALCULATION OF CORRECTIONS TO ROOT POSITION.
      X=R*C(NQ-2)+S*C(NQ-3)
      Z=C(NQ-2)**2+X*C(NQ-3)
C     CHECK FOR ZERO DENOMINATOR.
      IF (Z.EQ.0.) GO TO 107
      DR=(C(NQ-2)*Q(NQ-1)-C(NQ-3)*Q(NQ))/Z
      DS=(X*Q(NQ-1)+C(NQ-2)*Q(NQ))/Z
```

FORTRAN SOURCE LIST ROOT

SOURCE STATEMENT

```
      IF (R.NE.0.) GO TO 102
      IF (S.EQ.0.) GO TO 103
C     CHECK FOR WILD CORRECTION.
  102 IF(((R+DR)**2+(S+DS)**2).GT.9.*(R**2+S**2)) GO TO 106
  103 R=R+DR
      S=S+DS
  104 K=K+1
C     TEST FOR EXCESSIVE ITERATIONS.
      IF (K.GE.50) GO TO 105
C     TEST FOR CONVERGENCE.
      IF (DABS(DR).GT.DABS(R*TOL(IN))) GO TO 101
      IF (DABS(DS).GT.DABS(S*TOL(IN))) GO TO 101
C     USE THE QUADRATIC FACTOR JUST FOUND AS THE INITIAL APPROXIMATION
C     FOR THE NEXT QUADRATIC FACTOR.
  105 R1=R
      S1=S
      IF (.NOT.IND) GO TO 109
      IN=2
      GO TO 115
  106 R=3.D0*R
      S=3.D0*S
      GO TO 104
  107 R=1.D-5
      S=1.D-5
      GO TO 104
C     NOW THAT A QUADRATIC FACTOR HAS BEEN REMOVED FROM THE POLYNOMIAL
C     ITS ROOTS ARE FOUND AND ADDED TO THE OUTPUT ARRAY.
  109 R=-.5D0*R
      S=R**2-S
      IF (S.GT.0.) GO TO 110
      S=DSQRT(-S)
      DO 152 I=1,2
      L=L+1
      RTR(L)=R
      RTI(L)=S
  152 S=-S
      GO TO 111
  110 S=DSQRT(S)
      L=L+1
      RTR(L)=R+S
      RTI(L)=0.
      L=L+1
      RTR(L)=R-S
      RTI(L)=0.
  111 IF (IM) RETURN
C     REDUCE N AND SET UP NEW REDUCED POLYNOMIAL.
      N=N-2
      DO 153 I=1,N
  153 A(I,1)=Q(I)
      GO TO 100
C     APPROACHING END OF JOB.  POLYNOMIAL HAS BEEN REDUCED TO DEGREE
C     TWO OR LESS.
  112 IF (N.LT.2) RETURN
      IF (N.GT.2) GO TO 113
C     LINEAR FACTOR TERMINATION.
```

FORTRAN SOURCE LIST ROOT

SOURCE STATEMENT

```
      L=L+1
      RTR(L)=-A(2,1)/A(1,1)
      RTI(L)=0.
      RETURN
C     QUADRATIC FACTOR TERMINATION.
  113 IM=.TRUE.
      R=A(2,1)/A(1,1)
      S=A(3,1)/A(1,1)
      IF (M.LT.3) GO TO 109
      IN=2
      GO TO 101
      END
```

FORTRAN SOURCE LIST ACCEPT

SOURCE STATEMENT

```
      SUBROUTINE ACCEPT (M1)
C     SUBROUTINE ACCEPT TESTS IF THE POLES AND ZEROS OF THE BEST
C     RATIONAL FUNCTION CORRESPOND TO A PHYSICALLY REALIZABLE FUNCTION.
      DOUBLE PRECISION X(16),RR(30),RI(30),RTR(15),RTI(15),WR1,WR2,WA1,
     1     WA2,WA3,WA4,WA5,FILE(15)
      DOUBLE PRECISION DABS
      LOGICAL ISW1,ISW2,YES,ERROR,SWNUM,SWDEN
      COMMON NO,N1,WR1,WR2,WA1,WA2,WA3,WA4,WA5,NOPR,ISW1,ISW2,YES,X,RR,
     1     RI,RTR,RTI,ERROR,N3,N4,N5,N6,N7,N8,N9,SWNUM,SWDEN
      INTEGER FILMRK
      FILMRK=0
      YES=.TRUE.
      DO 150 I=1,M1
C     IF THE IMAGINARY PART OF THE ROOT OF OMEGA SQUARED IS NOT ZERO
C     (A TOLERANCE OF 1.E-10 IS ALLOWED FOR ROUND OFF), THEN THERE WILL
C     BE A SET OF FOUR ROOTS OF OMEGA IN QUADRANTAL SYMMETRY.  HENCE
C     SUCH A ROOT IS REALIZABLE WITHOUT FURTHER EXAMINATION.
      IF (DABS(RTI(I)).GE.1.D-10) GO TO 150
C     IF THE REAL PART IS LESS THAN ZERO, THERE WILL BE A PAIR OF ROOTS
C     ON THE SIGMA AXIS, HENCE NO FURTHER EXAMINATION NEEDED.
      IF (RTR(I).LT.0.) GO TO 150
      IF (FILMRK.EQ.0) GO TO 103
C     ALL OTHER ROOTS WILL BE IN PAIRS ON THE OMEGA AXIS.  THESE WILL BE
C     REALIZABLE ONLY IF THEY ARE OF EVEN ORDER.  FILE( ) CONTAINS A
C     STACK OF SUCH UNPAIRED ROOTS.  FILMRK COUNTS THE NUMBER CURRENTLY
C     STORED IN THE STACK.
C     EXAMINE THE FILE TO SEE IF THE NEW ROOT IS SUFFICIENTLY CLOSE
C     TO A PREVIOUS MEMBER.
      DO 151 J=1,FILMRK
      IF (DABS(RTR(I)-FILE(J)).LE.1.D-5*DABS(FILE(J))) GO TO 104
  151 CONTINUE
      GO TO 103
C     THE NEW ROOT MATCHED A PREVIOUS MEMBER.  THE PREVIOUS MEMBER IS
C     REMOVED FROM THE STACK AND THE STACK COUNTER DECREMENTED.
  104 FILMRK=FILMRK-1
  102 IF (J.EQ.FILMRK+1) GO TO 150
      FILE(J)=FILE(J+1)
      J=J+1
      GO TO 102
  103 FILMRK=FILMRK+1
      FILE(FILMRK)=RTR(I)
  150 CONTINUE
C     IF THE FILE CONTENT IS NOT ZERO, THE RATIONAL FUNCTION IS NOT
C     PHYSICALLY REALIZABLE.
      IF (FILMRK.NE.0) YES=.FALSE.
      RETURN
      END
```

REFERENCES

1. Achieser, N. I. Theory of Approximation. Trans. C. J. Hyman. New York, Ungar, [1956.]

2. Cheney, E. W. and Goldstein, A. A. "Newton's Method for Convex Programming and Tchebycheff Approximation." Numerische Mathematik, vol. 1 (1959), pp. 253-268.

3. Cheney, E. W. and Loeb, H. L. "On Rational Chebyshev Approximation." Numerische Mathematik, vol. 4 (1962), pp. 124-127.

4. Cheney, E. W. and Southard, T. H. "A Survey of Methods for Rational Approximation with Particular Reference to a new method based on a formula of Darboux." S.I.A.M. Review, vol. 5 (1963), pp. 219-231.

5. Crockett, J. B. and Chernoff, H. "Gradient Methods of Maximization." Pacific J. Math., vol. 5 (1955), pp. 33-50.

6. Curry, H. B. "The Method of Steepest Descent for Non-Linear Minimization Problems." Quar. Appl. Math., vol. 2 (1944), pp. 258-260.

7. Goldstein, A. A. "On the Stability of Rational Approximation." Numerische Mathematik, vol. 5 (1963), pp. 431-438.

8. Goldstein, A. A. and Cheney, W. W. "A Finite Algorithm for the Solution of Consistent Linear Equations and Inequalities and for Tchebycheff Approximation of Inconsistent Linear Equations." Pacific J. Math., vol. 8 (1958), pp. 415-427.

9. Goldstein, A. A., Herreshoff, J. B., and Levine, N. "On the "best" and "least q-th" approximation of an overdetermined system of linear equations." J. Assoc. Comput. Mach., vol. 4 (1957), pp. 341-347.

10. Guillemin, E. A. Synthesis of Passive Networks. New York, Wiley, [1957.]

11. Linvill, J. G. "The Approximation with Rational Functions of Prescribed Magnitude and Phase Characteristics." Proc. I.R.E., vol. 40 (1952), pp. 711-721.

12. Loeb, H. L. On Rational Fraction Approximations at Discrete Points. San Diego, Calif., Convair Astronautics, 1957. (Mathematical Pre-Print Series No. 9).

13. Loeb, H. L. "Algorithms for Chebyshev Approximation using the ratio of Linear Forms." Jour. Soc. Indust. Math., vol. 8 (1960), pp. 458-465.

14. Maehly, H. J. Rational Approximation for Transcendental Functions. Yorktown Heights, New York, I.B.M. Corp., 1959. (Research Report RC-86).

15. Rice, J. R. The Approximation of Functions, vol. 1. Reading, Mass., Addison-Wesley, [1964.]

16. Spang, H. A. "A Review of Minimization Techniques for Nonlinear Functions." S.I.A.M. Review, vol. 4 (1962), pp. 343-365.

17. Stiefel, E. L. "Über Diskrete und lineare Tchebycheff-Approximationen." Numerische Mathematik, vol. 1 (1959), pp. 1-24.