LA-UR -77-633

TITLE: SOFTWARE FOR APPROXIMATIONS OR APPROXIMATION THEORY AS AN EXPERIMENTAL SCIENCE

AUTHOR(S): L(ouis) Wayne Fullerton

SUBMITTED TO: Proceedings of the Rational Approximation Conference, Academic Press

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87545

An Affirmative Action/Equal Opportunity Employer

Form No. 836
St. No. 2629
1/75

UNITED STATES
ENERGY RESEARCH AND
DEVELOPMENT ADMINISTRATION
CONTRACT W-7405-ENG. 36

SOFTWARE FOR APPROXIMATIONS
OR
APPROXIMATION THEORY AS AN EXPERIMENTAL SCIENCE

L. Wayne Fullerton

Numerical analysis and approximation theory, in particular, can be an experimental science. This experimental nature is illustrated with several more-or-less new results. In the first half of this paper techniques for estimating the accuracy and significance of approximations are given. In the second half several generalizations of Chebyshev series that lead to nearly best approximations with respect to almost arbitrary weight functions and basis sets are presented.

1. Introduction

Conversational references to the experimental nature of numerical analysis usually emphasize the trial-and-error aspects of research. Certainly I do not dispute the trial-and-error nature of numerical analysis research, but I am most anxious to avoid illustrating the errors I have made. I call numerical analysis an experimental science in the same way that we all call physics or chemistry experimental sciences. There are two essential facets to an experimental science. First, theory or hypothesis suggests experiments that should be carried out. And second, experiments (conducted perhaps with computer programs) suggest new theoretical results. I wish primarily to emphasize this latter facet. In the next section, it is shown how computational experience can dictate the kind of numerical analysis that should be done. And in the third section, it is shown how experiments conducted with computer software can lead to new theoretical results.

2. Numerical Analysis for Software

Anyone who has used an approximation program probably has been annoyed by its inability to detect user errors. In order to compute an approximation, the user must supply function values that are somewhat more accurate than the approximation he desires. The more accurate values are often computed with a convenient ascending series for some argument values and an asymptotic series for other argument values. It is not uncommon to estimate incorrectly the number of terms needed in one of the series, so that the two series fail to match to the required accuracy. Alternatively, the user may incorrectly estimate the stability against roundoff of one of the series, so that it is inaccurate even though enough terms are used.

Now when a user requests a very accurate approximation with inaccurate function values, some approximation programs will do a great deal of work and possibly fail to derive any approximation. Even if the user supplies accurate function values, his approximation form may be so unstable that the approximation (if it can be derived) is not useful. These common experiences with approximation software dictate that the troublesome situations be detected so that perplexed users can be warned.

2.1 Input Function Accuracy

We wish to assess the error of a user-supplied function. The general methods in this section may be used to derive, for example, the relative error but in this case Generalized Chebyshev Series discussed in Section 3.2 must also be used. Let us, therefore, restrict consideration to the estimation of absolute errors and simply note that extension of the results here to arbitrarily weighted errors is straightforward.

Suppose we compute a high-order Chebyshev series approximation to the user-supplied function. Even though the series may contain 50 terms, only 10 terms may be significant. In such a case the error of the 10-term series would be nearly the same as

the full 50-term series, and the magnitude of the last 40 terms would all be nearly the same. We can determine how many terms are significant by observing that an N^{th} order series

$$F(x) \approx \sum_{i=0}^{N} f_i \, T_i(x)$$

is not only a near minimax approximation but also a discrete least squares approximation over the Chebyshev points $x_j = \cos \frac{j\pi}{N}$. Our strategy, then, is to estimate the number of terms to keep in the Chebyshev series in the same way that we estimate the number of terms to keep in any least squares approximation (cf. Ralston [5]).

The sum of the squares of the errors for an ℓ-th order series is

$$V_\ell = \sum_{j=0}^{N} \left[F(x_j) - \sum_{i=0}^{\ell} f_i \, T_i(x_j) \right]^2 .$$

If we estimate the value of $F(x_j)$ by the N^{th} order series and if we make use of orthogonality relations to eliminate cross products, we obtain

$$V_\ell = \sum_{j=0}^{N} \sum_{i=\ell+1}^{N} f_i^2 \, T_i^2 \, (x_j) = \frac{N+1}{2} \sum_{i=\ell+1}^{N} f_i^2 .$$

The standard error of one function value for an ℓ-th order series is given by

$$\sigma_\ell^2 = \frac{V_\ell}{N-\ell} = \frac{N+1}{2} \, \frac{1}{N-\ell} \sum_{i=\ell+1}^{N} f_i^2 .$$

We now compute these values for all ℓ. In order to evaluate the sum accurately, we start at $\ell=N$ for which the sum is zero and progressively decrease ℓ. Next we check in a forward direction

for some $\sigma_{k+1} \geq \sigma_k$. We then have an estimate of the number of terms, k, to keep and also an estimate of the error, σ_k , of the user-supplied function.

The scheme we have described can be used to detect both random errors and discontinuities. The scheme works because we know the true function being approximated must have only very low-amplitude high "frequencies" and that it must have no discontinuities. Otherwise, a low-order polynomial approximation would be inappropriate. We have found an efficient method for assessing the accuracy of input functions as well as output Chebyshev series approximations. The requirement for such an accuracy estimate was dictated by computational experience, and well known numerical techniques fortunately provided the solution.

2.2 Stability of Approximation Form

Knowing only the accuracy of an approximation is insufficient, because the approximation may be unstable against roundoff. A ten-digit approximation is of little use if 100-digit accuracy is needed to evaluate the approximation. A significance loss of 90 digits is, of course, uncommon; however, even a loss of one digit of significance may be unacceptable. Anyone who derives an approximation for use in a full machine-precision special function routine will be most distraught to learn the approximation is unstable against roundoff error while he is testing the special function routine. He should be warned about the instability of the approximation when the approximation is derived. Once again, experience (or experiment) dictates the need for some numerical analysis research. The results are just as easily obtained as in the previous subsection.

The significance loss incurred during the evaluation of an approximation can be easily estimated when the approximation itself is derived, provided we do not try to do too much. A simple way of measuring the stability of an approximation is to calculate the number of significant digits that should be kept in each

of the coefficients of the approximation so that the extra error introduced by rounding the coefficients is no larger than the weighted error of the approximation. Because every major computer represents floating point numbers with a nearly constant relative error, we need to calculate only one number, namely the number of significant digits to keep in each coefficient.

Suppose now we are given an approximation

$$A_n = \sum_{i=0}^{n} f_i \, \phi_i(x)$$

whose weighted error

$$\varepsilon = \max |\varepsilon(x)| = \max |w(x) \, [F(x) - A_n(x)]|$$

is nearly minimax. We require the orthogonal functions ϕ_i to be normalized so that $w^2(x) \, \phi_i^2(x) \leq 1.0$ as in Section 3.2. In the special case $w(x) \equiv 1$, the ϕ_i are just Chebyshev polynomials. We have chosen to analyze orthogonal series, because they presumably are the most stable form and, moreover, the easiest form to derive.

Assume the errors introduced by arithmetic operations and by evaluating the ϕ_i are negligible. Further assume the absolute error of the rounded coefficient f_i is Gaussian distributed with standard deviation σ_i. Of course, the errors are not really Gaussian distributed, but we need only an estimate of the required significance. An error of 50 percent in our estimate corresponds to only 0.3 significant figures and is perfectly acceptable. The standard deviation of the absolute error of the approximation evaluated with rounded coefficients is given by

$$\sigma_A^2(x) = \sum_{i=0}^{n} \left(\frac{\partial A_n}{\partial f_i} \right)^2 \sigma_i^2 \quad .$$

Now let δ be the standard deviation of the relative error of each rounded coefficient so that $\sigma_i^2 = f_i^2 \, \delta^2$. Furthermore,

recall that we want the weighted error introduced by the rounded coefficients to be less than the weighted error of the approximation, ε . Then we find

$$\varepsilon^2 = \max w^2(x)\, \sigma_A^2(x) = \max \left\{ \sum_{i=0}^{n} f_i^2\, w^2\, \phi_i^2 \right\} \delta^2 \quad .$$

But the $\phi_i(x)$ are normalized so that $w^2(x)\, \phi_i^2(x) \leq 1$, and so

$$\delta^2 \geq \frac{\varepsilon^2}{\sum_{i=0}^{n} f_i^2} \quad .$$

Finally, the number of significant figures, S, required to insure the effect of the rounding errors does not exceed the error of the approximation is

$$S = -\log_{10} \delta \quad .$$

Stable approximations are those for which δ is a large number compared with ε, that is, the required number of significant figures should be small. Thus, stable approximations will have small leading coefficients -- the higher order coefficients are unimportant if the series converge reasonably quickly.

The extension of the analysis in this subsection to rational orthogonal series is straightforward, but the resulting expression for δ is not as elegantly simple as the result above.

3. Software for Numerical Analysis

In the previous section, the importance of numerical analysis applications to approximation programs used in a production mode was emphasized. Naturally, these programs become at the same time more useful and reliable as research tools. In this section, we emphasize the use of carefully designed programs to conduct numerical experiments that may lead to new theoretical

results. Like a true experimental science, these theoretical results may immediately suggest new numerical experiments. Two (almost) new theoretical results are used to illustrate the utility of computer programs as research tools in the next two subsections.

3.1. Leveled Truncated Chebyshev Series

Truncated Chebyshev series are well known to be nearly best absolute error approximations in the uniform norm. Because Chebyshev series are near minimax approximations and because they are quite stable against roundoff errors, it is natural to express true minimax approximations in terms of Chebyshev polynomials. It is also natural to wonder what the error of a minimax approximation looks like in terms of Chebyshev polynomials. The Chebyshev series of the error is almost trivially calculated, especially if one is already expressing minimax approximations in terms of Chebyshev polynomials.

Consider, therefore, the dominant error terms of a second order polynomial minimax approximation to the exponential function on the interval [-1, +1]:

$$\begin{aligned}\varepsilon_2(x) = \ldots &+ .00013\ T_1 - .00553\ T_2 \\ &+ .04434\ T_3 \\ &+ .00547\ T_4 + .00054\ T_5 + \ldots\end{aligned}$$

The main error term is, as expected, T_3. Note, though, that the neighboring error terms are of the same magnitude but opposite sign. If this happens only once or twice, it must be an accident. But it happens over and over. It even occurs for rational minimax approximations. Consider the Chebyshev series for the absolute error of a second order divided by a second order rational minimax approximation to the exponential:

$$\varepsilon_{2,2}(x) = \ldots + .000009\ T_3 - .000038\ T_4 + .000067\ T_5 + .000037\ T_6 + .000011\ T_7 + \ldots$$

As anticipated, T_5 is the dominant error term. And again neighboring error terms are of the same magnitude but opposite sign. Because the behavior we observe for these two cases occurs very frequently, we should consider an explanation.

A truncated Chebyshev series is ironically guaranteed to have a nonuniform error curve. If, for example, we truncate a Chebyshev series at fourth order, then the dominant error term will ordinarily be T_5. The next error term will be T_6, and this error term (if nonzero) will constructively interfere with T_5 in some places and destructively interfere in other places. We truncate a Chebyshev series to obtain a nearly best approximation, but at the same time we insure the error curve is nonuniform.

From the above numerical results we know what to do about the interference of higher order error terms with the dominant error term: we modify the truncated Chebyshev series so that lower order error terms of the same magnitude but opposite sign are introduced in the error expansion. This procedure works because the sum of the high and low order terms have zeroes exactly where the dominant error term has extremae. To see this effect, make the transformation $x = \cos\theta$. The dominant error term is then $T_m(x) = \cos m\theta$, and furthermore

$$T_{m-\ell}(x) - T_{m+\ell}(x) = \cos(m-\ell)\theta - \cos(m+\ell)\theta = 2 \sin \ell\theta \ \sin m\theta \ .$$

The nonzero low order error term aliases the high order error term and, therefore, reduces interference effects.

We have, then, derived a technique for leveling truncated Chebyshev series -- a technique suggested solely by Chebyshev

series expansions of true minimax approximation errors. The leveled Chebyshev series should be regarded only as first order modifications to truncated Chebyshev series, because the introduction of the lower order error terms simply avoids the addition of more error at the extremae of the main error term. Nonetheless, the improvement is obtained at essentially no cost, and while a truncated Chebyshev series may deviate from a minimax approximation by perhaps 20 or 30 percent, the deviation of a leveled Chebyshev series is more likely to be only a few percent.

Economization of a power series [2] is a commonly employed method of obtaining a good approximation from a power series. In effect, the power series is converted to a Chebyshev series, then the small amplitude high order terms are dropped. One then obtains an economical approximation with fewer terms, but with little additional error. The results in this subsection could, however, be used to obtain a still better approximation with the same number of terms. Rather than truncating the Chebyshev series, the Chebyshev series should be leveled.

3.2. Generalized Chebyshev Series

Truncated Chebyshev series are nearly best approximations in the uniform norm. Unfortunately, they are only nearly best polynomial approximations and only in the sense of absolute error. It is natural to wonder about generalizations that would be good for arbitrary weight functions and non-polynomial bases. Originally this problem was motivated by the need for good starting values for the rational Remez iteration. However, before the rational problem is studied, we should solve the polynomial case.

Consider first the problem of finding an approximation $A_n(x)$ to the function $F(x)$ on $[-1, +1]$ with weight function $W(x) = 1$, such that the error

$$\varepsilon(x) = W(x)\ [F(x) - A_n(x)]$$

is near minimax. We know, of course, the solution is first to

define some polynomials -- Chebyshev polynomials -- from the orthogonality condition

$$\int_{-1}^{1} \frac{T_m(x)\, T_n(x)}{\sqrt{1-x^2}}\, dx = 0 \ , \ m \neq n,$$

with the $T_n(x)$ normalized so that their extreme value is unity. Next we expand $F(x)$ in a series

$$F(x) = \sum f_i\, T_i(x)$$

with

$$f_i = \frac{1}{h_n} \int_{-1}^{1} \frac{F(x)\, T_i(x)}{\sqrt{1-x^2}}\, dx \ ,$$

where

$$h_n = \int_{-1}^{1} \frac{T_i^{\,2}(x)}{\sqrt{1-x^2}}\, dx \ .$$

When this series is truncated at n-th order, we obtain the desired approximation A_n.

In generalizing to arbitrary weights, it is reasonable to suppose a simple function of the weight must be included in the orthogonality condition. I incorrectly conjectured that the weight in the orthogonality condition might be $W(x)/\sqrt{1-x^2}$ or perhaps $\sqrt{W(x)} \,/\, \sqrt{1-x^2}$. The problem of finding the appropriate orthogonal polynomials and expansion coefficients can be posed essentially as a Gauss-Christoffel quadrature problem. Because a good Gauss-Christoffel quadrature program was available to me, I quickly learned that these conjectures did not lead to nearly best approximations. I did observe, however, that the quadrature weight containing $\sqrt{W(x)}$ was the worse, so I tried

$W^2(x) / \sqrt{1-x^2}$. That choice I found to be the correct one. Now that the correct generalization is known, it is easy -- embarrassingly easy -- to explain why.

We will be expanding $F(x)$ in a series of some orthogonal polynomials

$$F(x) = \sum f_i \phi_i(x) \quad ,$$

and when we truncate the series at n-th order, the weighted error will be roughly $W(x) \phi_{n+1}(x)$. We want this error to be an equal ripple curve, just like $T_{n+1}(x)$ would be. Thus, the analogue of the Chebyshev polynomial T_i is $W \phi_i$. And when we substitute this result in the orthogonality condition, we find the ϕ_i are given by

$$\int_{-1}^{1} \frac{W^2(x) \ \phi_m(x) \ \phi_n(x)}{\sqrt{1-x^2}} \ dx = 0 \quad , \ m \neq n.$$

See Gautschi [4] for a discussion of the derivation of orthogonal polynomials. We choose to normalize these polynomials so that the extremum of $W(x) \phi_i(x)$ is unity. Such a normalization allows one to assess readily the accuracy of a truncated series in these polynomials. The weighted error bound is simply the sum of the absolute values of all the coefficients dropped from the series, and this bound is usually close to the true weighted error.

Truncated generalized Chebyshev series often are within 20 or 30 percent of the corresponding true weighted minimax approximations. Because each approximation will usually have a unique weight function, the use of a general Gauss-Christoffel quadrature routine is not the best way to obtain the orthogonal polynomials and expansion coefficients. The integrals needed can be done efficiently by an automated Gauss-Chebyshev scheme, and the expansion coefficients can be derived at the same time as the recurrence coefficients for the orthogonal polynomials.

Generalization to non-polynomial bases is now straightforward, in principle. Instead of constructing the orthogonal functions ϕ_i from the basis x^i, it is necessary to orthogonalize the desired basis functions. In practice, the integrals are not so easily evaluated, and one must be certain that the basis functions form a Chebyshev set. For a brief discussion of the applications of these approximations see Fullerton [3].

Further generalization to orthogonal-Padé approximations, where we require

$$\frac{\sum_{i=0}^{m} p_i \phi_i}{\sum_{i=0}^{n} q_i \phi_i} = \sum_{i=0}^{m+n} f_i \phi_i(x) \quad ,$$

with $q_0 \equiv 1$, are now easily obtained. We first discretize at the zeroes of $\phi_{m+n+1}(x)$, and solve the resulting linear equations for the p_i and q_i. As with Chebyshev-Padé approximations [1], we obtain either a degenerate approximation or a good one. If it is degenerate, we are not interested in the approximation because the next lowest order rational approximation will be nearly as good as the one we are trying to derive. If the discretized approximation is good, we can then use it as a starting value for finding differential corrections to the rational coefficients in the above nonlinear equation. The discretized solution usually is so close to the orthogonal-Padé solution, that only one or two iterations of the linearized version of the above equation are necessary. The orthogonal-Padé approximations always seem to be more accurate than the corresponding discretized approximations. Although the solution for orthogonal-Padé approximations outlined here is not nearly as elegant as the solution for Chebyshev-Padé approximations given by Clenshaw and Lord [1], at least we have obtained a solution.

4. Conclusions and Acknowledgements

The need for quality software dictates to some extent the type of numerical analysis that should be done, and several examples were given in Section 2. The results there led to programs that were not only much more useful production tools, but also much more useful research tools. Employing computer programs to conduct numerical experiments was shown in Section 3 to be an effective means of testing conjectures and deriving new theoretical results. I am confident that I would never have obtained the results in Section 3 unless I had had quality software tools available. Approximation theory and numerical analysis can be an experimental science.

Dr. D. D. Warner suggested the solution to the input function accuracy problem given in Section 2.1, and I am grateful to him for the suggestion. I am also grateful to Warner for numerous, long conversations that certainly led to some of the other results presented in this paper.

References

1. Clenshaw, C. W. and K. Lord, Rational approximations from Chebyshev series, Studies in Numerical Analysis (B. K. P. Scaife, editor), Academic Press, London, 1974, pp. 95-113.

2. Dahlquist, G. and A. Björck, Numerical Methods, Prentice - Hall, Englewood Cliffs, N. J., 1974, pp 125-126, (Translated by N. Anderson).

3. Fullerton, L. W., Portable special function routines, Proc. Workshop on Portability of Numerical Software, (W. Cowell, editor), Springer - Verlag, New York, (in Press).

4. Gautschi, W., Construction of Gauss-Christoffel quadrature formulae, Math. Comp., 22 (1968), pp. 251-270.

5. Ralston, A., A First Course in Numerical Analysis, McGraw - Hill, New York, 1965, pp. 234-235.

L. W. Fullerton
Group C3, MS 265
Los Alamos Scientific Laboratory
Los Alamos, New Mexico 87545