AN INTRODUCTION TO THE ACCELERATION OF SCALAR SEQUENCES

By

MARK ALLAN TOWNSEND

Bachelor of Science
Bethany Nazarene College
Bethany, Oklahoma
1965

Master of Arts
University of Oklahoma
Norman, Oklahoma
1969

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF EDUCATION
May, 1983

AN INTRODUCTION TO THE ACCELERATION

OF SCALAR SEQUENCES

Thesis Approved:

John P. Chandler
Thesis Adviser

Sharilyn A. Thoreson

Marvin S Keener

Jeanne L. Agnew

Norman N Durham
Dean of the Graduate College

ACKNOWLEDGMENTS

I would like to thank some of the many people who have been instrumental in my finishing this project. First, my committee: Professors Agnew, Chandler, Karman, Keener, and Thoreson. They have been as cooperative as anyone could possibly ask. Extra thanks go to Drs. Agnew and Chandler. If Professor Agnew had not given me a friendly thumb in the back, I would probably still be trying to write the definitive masterpiece on acceleration methods. Professor Chandler has been most patient with his delinquent thesis student, who seemed always to be having more and more office hours and less and less thesis hours. Most professors would have given up on me long before now; but he did not.

Other O.S.U. professors have been of great encouragement to me. Professors Choike, Jobe, Bertholf, and Blose were the best "bosses" any graduate assistant could have. I was very fortunate that Professor Duvall was on the O.S.U. Graduate Council when I had to ask them for an extension to finish this project.

My office mates--Dan, Neil, George, and Tom--have been ideal people to work with. (George, you can have your desk back, now.) Other graduate students have certainly added spice to my life while I have been here; I think especially of Evangelos, Joan, both Larrys, Paul, Eddie, and Mathew.

I could never have afforded to take a year off from teaching except for the fact that I have had the most considerate landlady in town, Agnes

Greiner. My rent would have been a good "deal" in the 1950's. Barbara Newport, who holds the Math Department together (with some help from Faye), has done her usual "top-notch" job in typing this entire paper. She has not complained a bit about all my "nitpicking" changes.

Some of the best people in the world have gone to church with me while I was a student here. Special thanks are due to the singles class at the First Christian Church for all their support in recent years. And many people who worked with me earlier at the Nazarene Student Center will be remembered fondly.

The next group I think of at this moment is composed of the hundreds (thousands?) of students I have had at O.S.U. in my years as a teaching assistant. They have convinced me beyond a doubt that there is no other job in the world which is so much fun as teaching.

When I think about all the people who have been inspirations to me since I was a boy, the list seems to have cardinality "c", at least. But last of all, and most of all, I would like to thank my brother Gary and my parents, for the boundless love and support they have given me over my entire life. If there is anything about me that merits admiration, it is largely due to their encouragement and example.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION AND BACKGROUND

A. Purpose of the Study

One of the most practical (as well as one of the most interesting) areas in mathematics is acceleration methods. The strategy of such methods is to use some kind of transformation on the partial sums of a slowly convergent series (or sequence). The goal of the methods is to produce new sequences which converge to the same limit as the old sequence, but faster.

The list of methods is quite extensive; each one is tailored to some particular assumption about the manner in which the sequence is approaching its limit. Smith and Ford (1979, 1982) did comparison testing on a number of the more important methods: Euler's method, the Wimp-Salzer method, Toeplitz arrays, Aitken's method, the epsilon, rho, and theta algorithms, and the u, v, and t transforms of Levin. Other methods covered in Wimp's (1981) book are Richardson and Romberg integration, power series methods, Lubkin's W transform and other variations of Aitken's method, the Q-D algorithm, the eta algorithm, and the G transform. Covering all these algorithms in this thesis was out of the question. A somewhat arbitrary subset has been chosen for discussion: Euler's method is easy to motivate and illustrates the basic acceleration concepts well, so we begin with it. Then will come a chapter on Richard-

son extrapolation. It is discussed briefly in the textbooks, but the reader has probably not seen some of the material we will present. Aitken extrapolation comes next--again, a textbook algorithm, but with more material here than in the usual text. We then move on to the epsilon algorithm, a generalization of Aitken's algorithm which is well-known for its power. The epsilon algorithm has had a vast number of articles written about it, but it is still not in the textbooks. The notation and the difficulty of the proofs for this algorithm are not conducive to inclusion in the undergraduate curriculum. But the main features of the epsilon "landscape" are quite accessible to any post-Calculus II student, if the notational difficulties and intricate proofs can be postponed for later studies. We finish with a brief introduction to the rho and theta algorithms, both offshoots of the epsilon algorithm. The rho algorithm complements the epsilon algorithm; and the theta algorithm, as an all-purpose accelerator, surpasses either of its "relatives". Smith and Ford (1979) decided that if each method is judged only as a "solo" performer, u, v, and θ are the winners. In their later (1982) article, looking at things from a slightly different point of view, they elevated epsilon and u (taken as a pair) to the top ranking. Therefore, though we do not discuss all acceleration methods, we have chosen some of the most important ones.

The overruling principle in our discussions will be clarity, not rigor. If something is easy to motivate, we motivate it. But we will often take the approach, "It can be shown that . . ." This is not done because the proofs are thought to be unimportant; rather, it is done because the sheer mass of such details would be likely to discourage the average reader from ever pursuing the subject. For the reader who wishes

to go through the proofs, we give the appropriate references. (There are enough proofs in this subject to give plenty of reading for many years.) We seek, for now, to complement the "precise" style of the current literature, in hopes of making the area more accessible.

B. An Illustration of the Problem

Consider the series

$$\pi = 4(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \ldots), \tag{1}$$

which results from setting x = 1 in the Maclaurin series

$$\tan^{-1}(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \ldots. \tag{2}$$

The next question is, assume we had the right side of (1) come up in a problem and we did not know the answer. How could we find it? (We thus are simulating a "real-world" problem, where we often would <u>not</u> know the sum of a particular series.) Any Calculus II student should answer, "Just add up some partial sums, and they will approach pi--even if you don't <u>know</u> they should approach pi." Good answer, but not too practical in this situation.

Let us assume we want to take partial sums until the distance from pi is less than $5 \cdot 10^{-9}$; our demand is thus for eight decimal place accuracy, less than most calculators have. If we let

$$S_0 = 0, \quad S_1 = 4, \quad S_2 = 4(1 - \frac{1}{3}), \quad \ldots \tag{3}$$

then the last term in S_n is $\pm 4/(2n - 1)$. If we require S_n and S_{n+1} to be within $5 \cdot 10^{-9}$ of the correct answer, they have to be within 10^{-8} of each other. Therefore, we must require

$$\frac{4}{2n+1} < 10^{-8}$$

$$n > 199,999,999.5 \tag{4}$$

$$n \geq 200,000,000.$$

Adding this many terms is out of the question, even on a large computer. And note, we have not necessarily insured that S_n and S_{n+1} are close to pi; just because they are close together does not force them to be close to the limit or even insure that a limit exists. But we have shown no n less than 200,000,000 has any chance of being large enough.

In fact, a good Calculus II student might say that the n of (4) should be doubled. The standard theorem on alternating series with terms going to zero monotonically says this: the size of the error is less than the size of the first omitted term. That theorem would <u>insure</u> that S_n is good enough, if

$$\frac{4}{2n+1} < 5 \cdot 10^{-9}$$

$$n > 399,999,999.5 \tag{5}$$

$$n \geq 400,000,000.$$

Nevertheless, the 200,000,000 is closer to being ideal than the 400,000,000. This follows from an easy result which should be in the textbooks but is not. The result, surprisingly, was first proven by a beginning Calculus student, Calabrese (1962); his theorem depends on an additional hypothesis which most alternating series satisfy anyway. Namely, not only the terms but also the gaps between the terms go to zero monotonically. The result is as follows: assume

$$S_n = a_1 - a_2 + \ldots + (-1)^{n-1} a_n,$$

$$a_1 > a_2 > a_3 > \ldots > 0, \qquad a_i \to 0 \tag{6}$$

$$a_1 - a_2 > a_2 - a_3 > a_3 - a_4 > \ldots > 0$$

$$S = S_n + r_n.$$

Then

$$\frac{a_{n+1}}{2} < |r_n| < \frac{a_n}{2}. \tag{7}$$

Not being able to resist giving the gist of a proof considerably shorter than Calabrese's, we show r_4 is greater than $a_5/2$:

$$r_4 = a_5 - a_6 + a_7 - a_8 + \ldots$$

$$= (a_5 - a_6) + (a_7 - a_8) + \ldots. \tag{8}$$

$$2r_4 = (a_5 - a_6) + (a_5 - a_6) + (a_7 - a_8) + (a_7 - a_8) + \ldots$$

$$a_5 = (a_5 - a_6) + (a_6 - a_7) + (a_7 - a_8) + (a_8 - a_9) + \ldots.$$

By comparing the two series term-wise and using the additional hypothesis, the result is immediate. Writing a_4 as a series leads to the other half of (7). Calabrese's result can easily be applied to show that to get an error less than $5 \cdot 10^{-9}$ on the pi series, it is sufficient to take

$$\frac{a_n}{2} \leq 5 \cdot 10^{-9}, \quad n \geq 200{,}000{,}001; \tag{9}$$

and it is necessary to take

$$\frac{a_{n+1}}{2} < 5 \cdot 10^{-9}, \quad n \geq 200{,}000{,}000. \tag{10}$$

Thus Calabrese has indeed saved us an added 200,000,000 partial sums that we thought might be necessary; unfortunately, we are still left with the first 200,000,000! Since that number is obviously prohibitive, this series is a prime candidate for an acceleration method of some type. In fact, series (1) is easily conquered by most of the methods we will discuss in this thesis.

C. Selecting the Best Approximation

When the different methods are applied to the partial sums, we will end up not with one new sequence, but a triangular array (taking the original sums as the left side). The question then arises, which of the entries is the best approximation? The answer, if things are working even half-way, should be "somewhere on the last row". Those entries will be influenced by the later, more accurate partial sums. A natural candidate will be the right end of the last row, since that entry is influenced by all the information available (to that point). Unfortunately, this candidate does not always deserve "election". Fortunately, the table entries themselves can sometimes be used to give us a clue as to when the best entry is not on the end: if the best entry is somewhere in the middle of the row, the approximations will generally get closer and closer together until we pass the best approximation. Then they will start spreading out as we continue along the row. By monitoring the horizontal differences and watching for them to increase in size, one can often pick out the best entry within a couple of columns. In fact, something like this can be easily proven in the case where we assume we have an oscillating series with the errors monotonically decreasing and then monotonically increasing.

For example, let

$$x_1 = S + e_1, \quad x_2 = S - e_2, \quad x_3 = S + e_3, \quad x_4 = S - e_4,$$
$$x_5 = S + e_5, \quad x_6 = S - e_6, \quad x_7 = S + e_7, \quad x_8 = S - e_i, \tag{11}$$

with all e_i positive and

$$e_1 > e_2 > e_3 > e_4 \quad \text{and} \quad e_4 < e_5 < e_6 < e_7 < e_8. \tag{12}$$

Then

$$e_1 + e_2 > e_2 + e_3 > e_3 + e_4,$$
$$e_4 + e_5 < e_5 + e_6 < e_6 + e_7 < e_7 + e_8, \tag{13}$$

which translates into

$$|x_1 - x_2| > |x_2 - x_3| > |x_3 - x_4|,$$
$$|x_4 - x_5| < |x_5 - x_6| < |x_6 - x_7| < |x_7 - x_8|. \tag{14}$$

The smallest difference has to involve x_4, the best approximation. We shall discuss this topic most in the chapter on Euler acceleration, but it is a good standard to keep in mind when using any of the algorithms. Fortunately, the best answers <u>are</u> often on the right end of the last row; this alleviates the problem somewhat.

D. The Role of Divergent Series

The modern student has generally been trained to think of divergent series as worthless for giving any information. The truth of the matter is, as we shall show repeatedly, there often is enough information hidden

in the divergent partial sums to determine quite precisely what function value the series was unsuccessfully "trying" to converge to. And if a powerful enough acceleration method is chosen to apply to the sums, it may very well generate a sequence which converges to what the divergent series "should have" converged to!

Euler (1707-1783) was certainly the leader in the use of divergent series to arrive at correct answers. For him, if a certain function f gave rise to a series, then the "sum" of that series for any x should be taken as f(x), even when the series diverged. He was of course aware that he was using the word "sum" in an extended sense but asserted there was no great problem since "the new definition. . .coincides with the ordinary meaning when a series converges. . ." (Bromwich, 1926, p. 322). This amounts to a belief, which Euler explicitly affirmed, that it was impossible for more than one function to produce a given (power) series. Bromwich adds that, although this "rule" is not completely correct, Euler used his definition "almost exclusively" in the form of defining

$$\sum u_n = \lim_{x \to 1} \left(\sum_{n=0}^{\infty} u_n x^n \right), \tag{15}$$

when that limit exists. If restricted to this case, Bromwich notes that the rule will always give the "correct" answer for the series (in the modern terms of analytic continuation). Of course, Euler did not have access to that later development. Of Euler's rule, Hardy (1949, p. 8) remarked, "No mathematician of his period could possibly have expressed himself on such a subject without very serious ambiguity," because of the "inadequacy of the current theory of functions."

Naturally some other early mathematicians tried to use divergent

series and came up with erroneous results. Knopp (1951, p. 459) comments that "It was only Euler's unusual instinct for what is mathematically correct which in general served him from false conclusions in spite of the copious use which he made of divergent series."

One example of the way Euler used divergent series was in obtaining the values of slowly convergent series. He was able to transform (deliberately!) the very slowly convergent series

$$1 + \frac{1}{2^p} + \frac{1}{3^p} + \frac{1}{4^p} + \ldots \qquad (p > 1) \tag{16}$$

into a sequence of divergent series. Each of the series eventually diverged; but each one approached more closely, and more rapidly, to the correct answer before diverging. By noticing where the differences started to increase (as in the previous section), Euler could pick a best approximation from each divergent series. The best approximations converged to the correct answer much more rapidly than the partial sums of the original convergent series!

A simpler use of divergent series, but still impressive, is given by Bromwich (1926, p. 320) from the works of Fourier (1768-1830). Fourier was obtaining a sine series for a particular function and found a series expression for the coefficient of sin (nx); namely,

$$(-1)^{n-1}\left(\frac{1}{n^1} - \frac{1}{n^3} + \frac{1}{n^5} - \ldots\right). \tag{17}$$

When $n > 1$, the series reduces to the function

$$f(n) = \frac{(-1)^{n-1} n}{1 + n^2} . \tag{18}$$

But when $n = 1$, the series is

$$1 - 1 + 1 - 1 + \ldots, \tag{19}$$

which Fourier does not hesitate to evaluate by using (18), which came only via (17)! The answer obtained is 1/2, which can be shown by more conventional techniques to be correct. Presumably it was this sort of maneuver which caused some of the early mathematicians to have severe doubts about the value of Fourier's work; but Euler would have agreed with him completely. After all, (19) can also be obtained by setting $x = -1$ in

$$1 + x + x^2 + x^3 + \ldots, \tag{20}$$

which was "always" $1/(1-x)$. Note: although we have here one numerical series coming from two functions, we do not have one power series from two functions; (17) and (20) are quite dissimilar. And, as Euler would have expected, when the two functions do give the same numerical series at $x = 1$, it is only because the functions themselves have identical values at that x--namely, 1/2, which Euler thought of as the "intrinsic" value of (19).

Cauchy and Abel renounced the use of any non-convergent series in the 1820's, but they did this with some hesitation (Knopp, 1951, p. 459). In modern times, since the theory of complex variables has become more developed, a lot of the "tricks" Euler succeeded with can be explained in terms of analytic continuation and asymptotic series. We discuss those two topics briefly before moving on to begin our study of acceleration methods.

Because some of the readers may not have any background in complex variables, we will restrict ourselves to a simple example on analytic

continuation. Let

$$f(z) = \frac{1}{1 - z}. \tag{21}$$

Except for having a "pole" at $z = 1$, which corresponds to a "vertical asymptote" if we keep z real, this function is perfectly well-behaved. All its derivatives are continuous away from $z = 1$. In the language of complex variables, f is "analytic" except at $z = 1$. Now let us consider the function g(z) defined by

$$g(z) = 1 + z + z^2 + z^3 + \ldots. \tag{22}$$

Notice that since g(z) is defined only by the power series, g(z) is meaningless outside the disk of unit radius centered at 0. E.g., for $z = -2$, $+3$, or $2 + 2i$, the ratio test implies divergence of the defining series. When g(z) is defined, it is identical to f(z); but f(z) is defined on the entire plane (except $z = 1$) while the domain of g is severely restricted. Nevertheless, g does have the power of determining <u>all</u> the function values of f, in the following sense: suppose you want to find <u>another</u> function h, <u>besides</u> f, which is analytic on the (punctured) plane and agrees with g where g(z) is defined. Complex variable theory says you can't find such an h; by making the function satisfy the stated conditions, you have eliminated every function except f(z); f therefore has a legitimate right to be called "<u>the</u> analytic continuation" of g to the rest of the plane. We will later apply our acceleration methods to the partial sums for "$g(z_0)$" where z_0 is <u>outside</u> the unit disk. The partial sums will diverge since $g(z_0)$ is not actually defined. But often the acceleration method can, to put it very roughly, "unscramble" the deficient information and produce the natural extension value, $f(z_0)$.

When this happens, it is of course quite exciting to observe! The "miracle" is caused by the fact that the acceleration method is transforming the (divergent) partial sums of the power series into the "convergents" of some other representation for f(z). Of course we are most familiar with the power series representation of f. But there are other representations, and sometimes they represent f on a much wider region than the power series does. If our acceleration method translates the power series into one of these "better" forms, we often get convergence to the correct value as the result. This is called "analytic continuation" of the power series function.

We turn now, briefly, to asymptotic series. Again, we will work via a standard example; it has the additional advantage of showing Euler's genius for violating rules and still coming up with correct results. Let

$$f(x) = \int_0^\infty \frac{e^{-w}}{1 + xw}\, dw. \tag{23}$$

First, let us ignore the fact that xw ranges up to ∞, and expand $(1/(1+xw)$ as a geometric series; we now have

$$f(x) = \int_0^\infty (e^{-w} - xwe^{-w} + x^2w^2e^{-w} - x^3w^3e^{-w} + \ldots)\, dw. \tag{24}$$

The Advanced Calculus veteran should recognize that the next step requires uniform convergence, whereas we have no reason to think we have any convergence. Nevertheless, forging ahead, we get

$$f(x) = \int_0^w e^{-w}\, dw \; - \; x\int_0^\infty we^{-w}\, dw \; + \; x^2\int_0^\infty w^2e^{-w}\, dw \; - \; \ldots \tag{25}$$

The reader who has studied probability will recognize the gamma integrals in (25); they do legitimately give factorials. We thus have

$$f(x) = 0! - 1!x + 2!x^2 - 3!x^3 + \ldots. \tag{26}$$

We previously talked about doing an analytic continuation of a power series that converged on just a small disk in the complex plane. But this power series is considerably worse: the ratio test shows the series in (26) converges only for $x = 0$. Is there really any chance of a legitimate connection between $f(x)$ and such a crazy series? We shall soon see there is.

Euler (1755) showed that the integral for $f(x)$ could be rewritten as a finite integral:

$$f(x) = \frac{1}{x} e^{1/x} \int_0^x \frac{e^{-1/t}}{t}\, dt, \tag{27}$$

so that, for example,

$$f(1) = e\int_0^1 \frac{1}{te^{1/t}}\, dt. \tag{28}$$

By the trapezoidal rule or some other rule, this integral can be evaluated as precisely as desired. It is approximately .596347. Euler therefore concluded

$$.596347 = 0! - 1! + 2! - 3! + 4! - \ldots. \tag{29}$$

We are greatly surprised, but Euler was probably not, when he repeatedly applied his acceleration method to the right side of (29) and obtained a sequence of divergent series which indicated the correct answer was about .596347. Euler had several other ways of attacking the series; and they all indicated about the same answer. He therefore concluded that the "natural" value of Wallis' series (the right side of (29)) was indeed $f(1)$. A very complete account of Euler's work on this series is given

by Barbeau (1979).

The function f(x) is analytic as a function of a complex variable, in the entire plane. Hardy (1949, p. 26) even gives a series representation for it, though not a power series. But what is the connection between the series in (26) and the nice function f(x)? The connection will require a bit of explanation.

We are used to thinking of keeping z (or x) fixed and letting the partial sum index n go to ∞. But for this situation we must switch gears: keep n constant and let z approach zero. Then, no matter what n we choose, f(z) is not "very" far from the n^{th} partial sum of (26) when z gets close to zero. More precisely, let z be kept in any "wedge" in C which does not contain the negative real axis. Hardy (1949, p. 27) shows there is then a K which depends on neither n nor z, such that for any n and z,

$$|f(z) - (0! - 1!z + 2!z^2 - \ldots + (-1)^n n!z^n| \leq K(n+1)!|z|^{n+1}. \tag{29}$$

Denote the partial sum by $f_n(z)$. For any n, we have

$$\frac{|f(z) - f_n(z)|}{|z|^n} \leq K(n+1)!|z|, \tag{30}$$

which goes to zero as $|z| \longrightarrow 0$.

Now it would seem fair to claim the f_n are "good" representations for f(z) when |z| is small, in a modified sense. For let n be any number; choose it very large, to make $|z|^n \longrightarrow 0$ very fast as $|z| \longrightarrow 0$. Then (30) asserts that the difference between f(z) and $f_n(z)$ goes to zero even faster as $|z| \longrightarrow 0$. In this situation, the series in (26) is said to be "asymptotic" to f(z) as $z \longrightarrow 0$. We indicate this by replacing "=" in

(26) with "$\sim$". Acceleration methods are sometimes able to recover f(z) from a merely "asymptotic" series for f(z). Several of the methods we discuss will successfully recover f(z) on Wallis' series. Hardy (1949) gives more information on asymptotic series.

At long last, we now begin our study of the acceleration methods proper. It seems appropriate to begin with Euler's method, for obvious reasons.

CHAPTER II

EULER ACCELERATION

A. Introduction

The simplest and oldest acceleration method will be discussed first. It was published by Euler (1755, p. 281) but is still in use. We will examine standard results concerning its application to both convergent and divergent series; then we will discuss some minor but possibly new refinements found by the present author.

Recall a series mentioned earlier:

$$\pi = 4(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \ldots). \tag{1}$$

Let

$$S_0 = 0, \quad S_1 = 4, \quad S_2 = 4(1 - \frac{1}{3}), \quad \ldots. \tag{2}$$

(The reason for the initial "0" will be clear later.) By a well-known theorem, the errors of the partial sums alternate between being positive and being negative. Form a new sequence of averages:

$$T_1 = (S_0 + S_1)/2, \quad T_2 = (S_1 + S_2)/2, \quad \ldots. \tag{3}$$

Examination will show that the new sequence is converging to pi faster than the original partial sums, with errors still alternating in sign. So repeat the averaging process, again and again. This gives the Euler

method. The results obtainable by using $S_0,\ldots,S_6$ are shown in Table I, with the best entry in each row underlined. (The table is better constructed by row than by column, if the S_i are difficult to obtain for a particular problem.)

TABLE I

EULER ACCELERATION OF PI SERIES

S_i	T_i					
0						
4.000	2.000					
2.667	3.333	2.667				
3.467	3.067	3.200	2.933			
2.895	3.181	3.124	3.162	3.048		
3.340	3.117	3.149	3.137	3.149	3.098	
2.976	3.158	3.138	3.143	3.140	3.145	3.122

Note, the best answer on each row is not obtained by continuing the process as far as possible. See Table II. Again, the position of the best entry in each row is indicated by underlining. The best answers are consistently about two-thirds of the way across the row, instead of at the end. The present author has not seen this sort of rule elsewhere, and it leads to several (apparently) new results and conjectures, which will be mentioned later. The best entry in Table I, 3.140, is more precisely 3.13997. It can be easily shown that to attain an answer of this accuracy by ordinary summation would require over six-hundred partial

sums. The Euler method has used seven.

TABLE II

NEGATIVE OF EXPONENTS ON ERRORS IN EULER ACCELERATION FOR PI SERIES WHEN ERRORS ARE IN SCIENTIFIC NOTATION

0											
1	0										
1	1	1									
1	2	2	1								
1	2	2	2	2							
1	2	3	3	3	2						
1	2	3	3	3	3	2					
1	2	3	4	4	4	3	3				
1	3	3	4	4	4	4	4	3			
1	3	4	4	5	5	5	5	4	3		
2	3	4	4	5	5	5	5	5	5	3	
2	3	4	5	5	5	6	6	6	5	5	4

B. An Alternate Presentation

The presentation of the Euler method in terms of repeated averaging is apparently well known. Wynn (1971) mentions that it was used as an example in the original report on ALGOL 60. But this common knowledge does not seem to find its way into the numerical analysis textbooks. For example, Hardy (1949, p. 21ff) gives an extended discussion of the Euler method; but he never mentions that averaging is equivalent. He does men-

tion averaging for the partial sums of a divergent series, but he attributes it only to Hutton in 1812. Most books do not mention averaging, even when Euler's method is included. This seems unfortunate; nevertheless, the standard presentation does aid in understanding the algorithm's performance. For that reason, we will now re-formulate the algorithm in terms more similar to those used in the texts.

We proceed in a purely formal way. Write

$$\begin{aligned} S &= A_1 - A_2 + A_3 - A_4 + A_5 - A_6 + \ldots \\ S &= \quad A_1 - A_2 + A_3 - A_4 + A_5 - \ldots. \end{aligned} \tag{4}$$

Adding term-wise,

$$\begin{aligned} 2S &= A_1 - (A_2 - A_1) + (A_3 - A_2) - (A_4 - A_3) + \ldots \\ S &= A_1/2 - (1/2)(\Delta A_1 - \Delta A_2 + \Delta A_3 - \ldots). \end{aligned} \tag{5}$$

Repeat the process:

$$\begin{aligned} S &= A_1/2 - (1/2)(\Delta A_1 - \Delta A_2 + \Delta A_3 - \ldots) \\ S &= A_1/2 - (1/2)(\quad \Delta A_1 - \Delta A_2 + \ldots). \end{aligned} \tag{6}$$

Adding term-wise, and dividing by 2,

$$S = (1/2)(A_1 - (1/2)\Delta A_1) + (1/4)(\Delta^2 A_1 - \Delta^2 A_2 + \ldots). \tag{7}$$

This process can be continued. The infinite series we are proceeding toward is

$$S = (1/2)(A_1 - (1/2)\Delta A_1 + (1/4)\Delta^2 A_1 - \ldots), \tag{8}$$

the standard representation for the basic Euler transform.

On the other hand, we could rewrite (4):

$$S = S_1 - (A_2 - A_3 + A_4 - A_5 + \ldots) \tag{9}$$

then apply (8) to the abridged sum in "( )" to obtain

$$S = S_1 - (1/2)(A_2 - (1/2)\Delta A_2 + (1/4)\Delta^2 A_2 - \ldots). \tag{10}$$

This is what Wynn (1971) calls the "delayed" Euler transformation, with a delay of one. It is easy to show that the entries along any diagonal of the Euler table are partial sums in the Euler series with an appropriate delay. See Table III.

TABLE III

REPRESENTATION OF THE EULER TABLE IN TERMS OF SERIES

0			
S_1	$0 + (1/2)A_1$		
S_2	$S_1 - (1/2)A_2$	$0 + (1/2)(A_1 - (1/2)\Delta A_1)$	
S_3	$S_2 - (1/2)A_3$	$S_1 - (1/2)(A_2 - (1/2)\Delta A_2)$	...
S_4	$S_3 - (1/2)A_4$	$S_2 + (1/2)(A_3 - (1/2)\Delta A_3)$	...

The standard presentation considers a one-dimensional string of approximations which corresponds to one diagonal. The method of calculation is typically a table of forward differences instead of a table of averages. See Table IV and the accompanying calculations for the diagonal through S_3 (3.46667) in our previous example.

TABLE IV

FORWARD DIFFERENCES FOR COMPUTING THE S_3 DIAGONAL OF THE PI SERIES

$A_4 = .5714$		
$A_5 = .4444$	$\Delta A_4 = -.1270$	
$A_6 = .3636$	$\Delta A_5 = -.0808$	$\Delta^2 A_4 = .0462$

We can use the top diagonal of Table IV to compute the next three entries in the S_3 diagonal; as in Table I,

$$\begin{aligned} S_3 - (1/2)A_4 &= 3.181 \\ S_3 - (1/2)(A_4 - (1/2)\Delta A_4) &= 3.149 \\ S_3 - (1/2)(A_4 - (1/2)\Delta A_4 + (1/4)\Delta^2 A_4) &= 3.143. \end{aligned} \tag{11}$$

Note that what appears to be an alternating series is in fact monotonic because the differences alternate in sign.

C. Keys to Successful Use

Knopp (1951, p. 243) says Euler gave his results "without any considerations of convergence." Ames (1901) was the first to prove that if <u>any</u> sequence of S_i converges to a limit S, then the columns and diagonals of the associated Euler table must also converge to S. However, the convergence rate for what we now call "totally oscillatory" alternating series had been established much earlier by Poncelet (1835, pp. 1-15). (Note that Table IV suggests the pi series is totally oscillatory, since the signs on the differences alternate.) Ames (1901) reached equivalent

results; they essentially state that if the series is totally oscillatory, then the upper bound on error is divided by a factor of 2 for each step down any diagonal in the Euler table.

Pinsky (1978) essentially halved the upper bound on errors that had been given earlier, by using the results of Calabrese (1962) given in Chapter I. Assume that we do have a totally oscillatory series; i.e., suppose the diagonals are monotone, in alternating directions. (See Table III.) Then, although Pinsky evidently did not see this, his result immediately implies: an upper bound on the size of the error of any entry in the table is the difference between the entry above it and the entry beside it; i.e., a difference on the next higher diagonal.

To see how a lack of "total" oscillation can destroy the usual success, see Table V, based on the series

$$\pi + 1 = (4 + 1/2) - (4/3 - 1/4) + (4/5 + 1/8) - (4/7 - 1/16) + \ldots . \qquad (12)$$

TABLE V

APPLICATION OF EULER'S METHOD TO AN ALTERNATING SERIES WITH MONOTONE COMPONENT (SUM = PI + 1)

0						
4.500	2.250					
3.417	3.958	3.104				
4.342	3.879	3.919	3.511			
3.833	4.087	3.983	3.951(*)	3.731		
4.308	4.071	4.079	4.031	3.491(*)	3.861	
3.960	4.134	4.103	4.091(*)	4.061	4.026(*)	3.926

The best entry on each row is underlined, and (*) denotes a violation of the required monotonicity along diagonals. Series (12) is an alternating series, with differences going to zero monotonically. The diagonals are still converging to pi, but quite slowly. The later columns seem to be a waste of time. The difficulty comes from the flawed monotonicities of the diagonals, indicating wrong signs on $\Delta^2 A_2$, $\Delta^3 A_2$, $\Delta^4 A_2$, and $\Delta^2 A_4$.

If we did not know the correct limit for Table V, which entries should we accept as the best? As we shall discuss later, a row of the Euler table will often oscillate about the limit, with the errors first decreasing and then increasing in magnitude. In such a case, monitoring the differences between consecutive entries of a row seems a good idea. The best entry is likely to be near the smallest difference. For Table V, the differences do locate the best entries fairly well, as we discussed in Chapter I. See Table VI. Best entry positions and smallest differences are underlined.

TABLE VI

HORIZONTAL DIFFERENCES FOR THE ENTRIES IN TABLE V

x												
x	-2.250	x										
x	.541	x	-.854	x								
x	-.463	x	.040	x	-.408	x						
x	.254	x	-.104	x	-.032	x	-.220	x				
x	.237	x	.008	x	-.048	x	-.022	x	-.130	x		
x	.174	x	-.031	x	-.012	x	-.030	x	-.035	x	-.100	x

It should be fairly clear that Euler's method is useful mostly for oscillating sequences, not for monotone ones. For a monotone sequence, each column will be worse than the previous one. However, it should be noted that on some occasions, a monotonic series can be rewritten in terms of an alternating one. For example, Ames (1901) sums the series for Zeta(p),

$$\text{Zeta}(p) = 1 + 2^{-p} + 3^{-p} + 4^{-p} + \ldots \qquad (p > 1) \tag{13}$$

by rewriting Zeta(p) as

$$\text{Zeta}(p) = \frac{2^{p-1}}{2^{p-1} - 1}\,(1 - 2^{-p} + 3^{-p} - \ldots), \tag{14}$$

and then applying the Euler method. See Table VII, for the case p = 2. The true sum is 1.645 to three decimal places. Again, the best entry in each row is underlined.

TABLE VII

ZETA(2) VIA EULER'S METHOD

0						
2.000	1.000					
1.500	1.750	1.375				
1.722	1.611	1.681	1.528			
1.597	1.660	1.635	1.658	1.593		
1.677	1.637	1.648	1.642	1.650	1.621	
1.622	1.649	1.643	1.646	1.649	1.647	1.634

D. "Summing" Divergent Series

To this point, we have examined the use of Euler's method on convergent alternating series. Although a monotone component will destroy the method's effectiveness, the method will work beautifully on totally oscillating series. We have seen that the table entries can easily be used to calculate the maximum error of any entry. But now we turn to what may seem less orthodox: the use of the method to "sum" divergent series.

We have discussed, in Chapter I, Euler's rule about how the "sum" of a divergent power series should be defined. His rule would give (in the extended sense), <u>for all x</u>,

$$1/(1 - x) = 1 + x + x^2 + \ldots \tag{15}$$

$$1/(1 - x)^2 = 1 + 2x + 3x^2 + 4x^3 + \ldots. \tag{16}$$

His opinion was strengthened by the experimental evidence that his transform, when applied to a divergent series, often converged to the "right" answer. For example, setting $x = -2$ in (15) and then applying the transform gives the following results on our top diagonal (Knopp, 1951, p. 468).

$$1 - 2 + 4 - 8 + \ldots \longrightarrow 1/2 - 1/4 + 1/8 - 1/16 = 1/3 = 1/(1 - x)]_{x=-2} \tag{17}$$

Likewise, setting $x = -1$ in (16) gives

$$1 - 2 + 3 - 4 + \ldots \longrightarrow 1/2 - 1/4 + 0 + 0 + \ldots = 1/4 = 1/(1 - x^2)]_{x=-1} \tag{18}$$

On the other hand, the transform does not always succeed so well. With

$x = 2$ in (15), we obtain

$$-1 = 1 + 2 + 4 + 8 + \ldots. \tag{19}$$

It is obvious that no amount of averaging of the partial sums is going to give -1 as a limit. This raises the question of when, exactly, is the method going to give the "right" answer for a divergent series? We turn now to that question, which amounts to discussing the use of the method in analytic continuation.

To explore this topic, we need to allow repeated applications. This consists in taking a diagonal, generally the top one, and using it as a new "initial" column for another table. We will use E_p to denote the operation of applying the transform p times in this manner.

Knopp (1951, p. 508 ff.) gives several results along this line. The results for even a geometric series are revealing; he obtains

$$\sum_{n=0}^{\infty} Z^n \ (Z \in \mathbb{C}) \xrightarrow{E_p} 2^{-p} \sum_{n=0}^{\infty} \frac{(2^p - 1 + Z)^n}{2^{np}} . \tag{20}$$

The original series has a radius of convergence "1". The tranformed series is also geometric and also sums to $1/(1-Z)$, but it converges whenever $|Z - (1 - 2^p)| < 2^p$. The region of convergence increases as p increases. See Figure 1.

The picture itself suggests the proper conclusion: If $\mathrm{Re}(Z) < 1$, then the series $1 + Z + Z^2 + \ldots$ is "E_p-summable" to $1/(1-Z)$ for sufficiently large p. For example, $Z = -4$ gives $1 - 4 + 16 - 64 + \ldots$, which will be outside the E_1 circle. The top diagonal will diverge. But if we use that top diagonal as the basis for a new table, the next top diagonal will converge to the "proper" sum, 1/5, since $Z = -4$ is inside the E_2

circle.

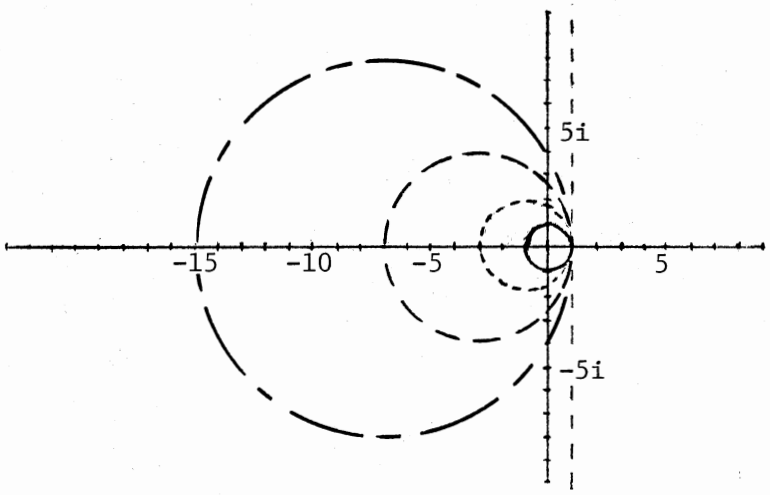

Figure 1. Regions of Convergence for the Geometric Series $\sum_{n=0}^{\infty} Z^n$ and its First Three Euler Transforms

On the other hand, notice that none of the circles in Figure 1 reach to the right of the line Re(Z) = 1. That means that no E_p will sum a geometric series like $1+2+4+8+\ldots$ (Z = 2). This is true in general: if the partial sums of a series "converge" to either plus or minus infinity, the transformed sums will likewise go toward the same infinite limit. It is only when the partial sums diverge by way of oscillation that the Euler transforms may give a finite limit (Hardy, 1949, p. 10).

The first systematic work on the E_p transforms was in two papers by Knopp in the early 1920's. The papers are in German, but two of the more

interesting theorems are given by Hardy (1949, p. 179):

Theorem: If a series is E_q-summable and $p > q$, then the series is also E_p-summable to the same limit.

Thus, just as in Figure 1, the transforms always do form a "nested" sequence of methods, with bigger and bigger convergence regions.

Theorem: If $\sum_{n=1}^{\infty} a_n$ is E_q-summable, then $a_n = 0((2q + 1)^n)$.

This theorem simply puts a limit on how fast a_n can grow without destroying the chances of the E_q transform to converge.

The most beautiful theorem on the use of the transform is also by Knopp (1949, p. 508). It gives an explicit algorithm for drawing the circles of consequence (as in Figure 1) for much more general series than the geometric series:

Theorem: Assume $\sum_{n=0}^{\infty} c_n z_n = f(z)$ has a finite positive radius of convergence in $\mathbb{C}$. Given a natural number p, this is how to find where (in $\mathbb{C}$) the series is E_p-summable to f(z): for each a, $0 \leq a < 2\pi$, move in the "a" direction from 0 until reaching the first singular point of f(z) in that direction. Call that point z_a. (If no singularity lies in the "a" direction, that direction need not be considered in what follows.) Define k_a to be the circle in C given by $|z/z_a + 2^p - 1| < 2^p$. Take the intersection of all k_a as G_p. Then the series is (absolutely) E_p-summable to the analytic extension of f on the interior of G_p. Outside G_p the E_p transform of the series is divergent.

For another example, consider the well-known series, which converges when $-1 < x \leq +1$:

$$\ln(1 + x) = x - (1/2)x^2 + (1/3)x^3 - (1/4)x^4 + \ldots. \qquad (21)$$

The convergence regions given by Knopp's theorem are as shown in Figure 2, along with the original circle of convergence.

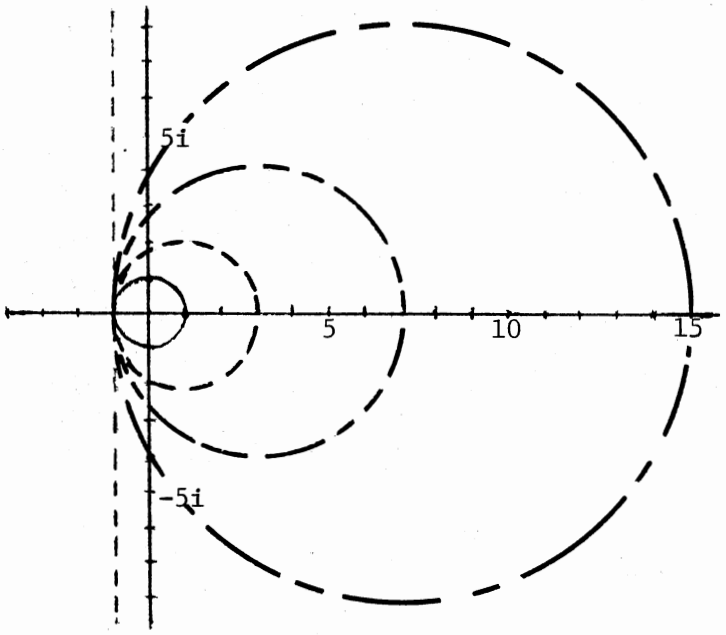

Figure 2. Convergence Regions for $\ln(1+x)$ Series and its First Three Euler Transforms

According to Figure 2, we can let $x = 2$ and use (21), apply Euler's method to the partial sums, and obtain a top diagonal converging to $\ln(3)$. For $x = 4$, Figure 2 indicates we will have to use E_2 to obtain convergence of (21) to $\ln(5)$, 1.609... This is in fact the case; see Table VIII.

TABLE VIII

RESULTS OF APPLYING E_2 TO THE DIVERGENT SERIES FOR LN(5) = 1.609...

0	0	0
4.000	2.000	1.000
-4.000	1.000	1.250
17.333	2.167	1.396
-46.667	.917	1.474
158.133	2.442	1.522
-524.533	.546	1.551
1816.038	3.003	1.570
-6375.962	-.208	1.583
⋮	⋮	⋮
Partial Sums	E_1 Diagonal	E_2 Diagonal

The gains may not be so considerable when the original function f(z) has "unfortunately" placed singularities. For example, let

$$g(z) = 1/(1-z) + 1/(2+z). \tag{22}$$

The series can be obtained easily and converges for $|z| < 1$. The convergence regions for the series itself, along with the convergence regions for E_1, E_2, E_3, are shown in Figure 3. On the real axis, E_2, E_3,... give us no additional extension of the convergence region.

We have been discussing problems where the Euler method takes a divergent series and finds the "correct" sum by converting the divergent series into a convergent one which converges to the correct limit. But the next (and last) example we shall introduce shows that the production

of a convergent series is not absolutely necessary to the successful use of Euler's method!

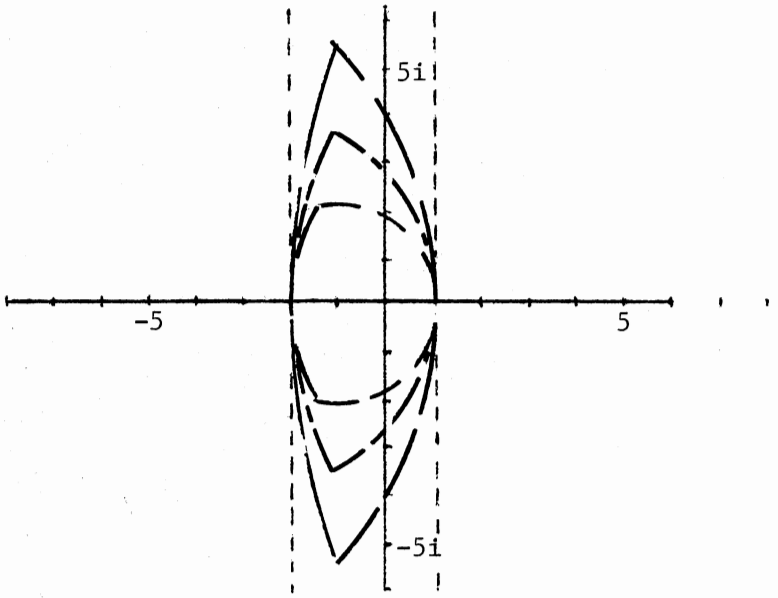

Figure 3. Convergence Regions for the g(z) Series and its First Three Euler Transforms

E. "Almost Summing" Divergent Series

Wallis' series was introduced in Chapter I:

$$0! - 1! + 2! - 3! + 4! - \ldots \qquad (23)$$

has the "natural" associated value .596347..., as we discussed earlier. Barbeau (1979) gives a complete study of the different ways Euler used to justify the value obtained. (See Chapter I for one way.) One of the

methods Euler used is the one we have been discussing. We will work with the related series

$$1! - 2! + 3! - 4! + 5! - \ldots, \tag{24}$$

which should clearly have the value .403653 if .596347 is correct for (23). Hardy (1949, p. 26) describes the efforts of La Croix, working in 1819, to assign a proper value to (24). His best estimate was .4008. We follow his basic attack: apply the Euler method repeatedly and hope that something good happens! After the first application we will (with La Croix) always use a delay of two, which means the first two entries on the diagonal are not used in producing the next table. For the resulting upper diagonals, see Table IX. The best entry in each diagonal is underlined.

TABLE IX

THE INITIAL PARTIAL SUMS OF LA CROIX' SERIES, AND THE RESULTS OF APPLYING E_1, E_2, E_3

0			.3594
1			.4005
-1	.250	.3594	.3967
+5	.625	.4414	.4016
-19	-.062	.3447	.3999
101	1.594	.4949	.4024
-619	-3.234	.2263	.4009
4421	13.320	.7845	.4034
-35,899	-51.863	-.5212	.4007
⋮	⋮	⋮	⋮

Note that each transformed sequence eventually diverges; in fact Hardy (1949, p. 28) comments that each generated sequence eventually diverges almost as wildly as the original partial sums. However, each new sequence approaches nearer the correct value and stays there longer than did the earlier sequences. When the differences in a given sequence begin to grow, we know we are leaving the "best" part of the sequence. This makes it possible to estimate the "correct" sum without ever obtaining a convergent series. Bromwich (1926, p. 326) mentions that Euler sometimes obtained answers correct to eighteen decimal places using such techniques! But it should be noted that, as a matter of fact, there are other methods, to be discussed later, which will "sum" Wallis' series much more quickly than does Euler's method.

F. Some (Possibly) New Results and Conjectures

Having discussed the standard facts about Euler's method, we return to a question suggested by Table I: under what conditions might we be able to predict the location of the best entry on a particular row of the Euler table, even before we construct the table? This question does not appear to be approached in the recent literature, probably because the algorithm is usually discussed in terms of one diagonal. However, Poncelet (1835, pp. 1-17) did notice a fact which is extremely relevant: given two consecutive entries in a column which oscillates about the limit, the average of those entries will fail to be an improvement on both of them only if the error in the better of the two entries is less than one-third the error in the worse of the two entries. See Figure 4.

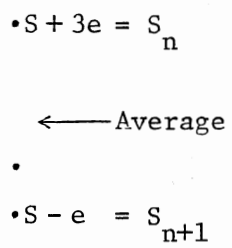

Figure 4. The Fastest Improvement Which Euler's Method Can Allow Without Giving a Worse Average

Poncelet did not put the question quite in our terms; but to find the best entry in a row, we therefore just proceed along the row until we reach an element "x" which is three times as close to the limit as is the element just above "x". Next question: how can we know when the error ratio reaches one-third, since we do not know the correct limit? At this point, we shall not discuss proofs; on some points the present author would not claim yet to have completely rigorous proofs, anyway. But there do seem to be some things one can predict about series which approach their limit in such a way that the error has a specified form. Namely, assume that in the original sequence, for $i = 1,2,3,\ldots$,

$$S_i = S + (-1)^i(K_1/i^p + K_2/i^{p+1} + \ldots) \tag{25}$$

then the following things are apparently true in the Euler table formed from the SI:

(1) The columns and rows both oscillate regularly about the correct answer: positive error, then negative, then positive, etc.

(2) In the first column $|S_i - S|$ goes to zero at the same rate as i^{-p} goes to zero. In the next column, $|T_i - S|$ goes to zero at

the same rate as $i^{-(p+1)}$ goes to zero. And so forth.

(3) For large values of i, the best approximation in row i will occur about two-thirds of the way across the row, regardless of K_1, K_2,... and p.

(4) By assuming that when we reach two-thirds of the way across the row the error ratio is about 1/3, we can calculate an extrapolation based on that assumption. The extrapolation tends to be much better than any entry in the Euler table to that point.

The author hopes to publish explanations and proofs for these results someday; but they are too technical to fit into the present thesis.

G. Conclusions

We have spent so much time on the Euler algorithm because it illustrates so many of the things that happen with all extrapolation algorithms. As a competitive method, it falls short on some problems when compared with more recent algorithms. Nevertheless, the method is still used for problems that are not "too" wild. And nothing can possibly be more satisfying than finding the sum of a slowly convergent series or a divergent one, without ever doing anything more complicated than averaging pairs of numbers! For convergent series where the ratio of consecutive errors approaches -1, and for divergent series which oscillate back and forth (but not too wildly), the more modern methods are not apt to make much improvement on Euler's method.

CHAPTER III

RICHARDSON/ROMBERG EXTRAPOLATION

A. Introduction and Overview

At this point, we switch to a slightly different context. In the discussion on Euler's method, the approximations used to generate the table were assumed to be partial sums of some series. But in the present discussion, the approximations used will not be partial sums; instead, they will be function values A(h), where h is a varying positive step size; the goal is to use the A(h) values to estimate $\lim_{h \to 0} A(h)$.

The two algorithms discussed in this chapter have generated an enormous amount of literature. Their histories have been interwoven and have been so interesting that, rather than simply presenting the algorithms in their modern forms immediately, we shall discuss the evolution of the methods to their present forms.

Richardson (1910, 1923, 1927) introduced his extrapolation idea briefly in his two earlier articles, but the 1927 article was completely dedicated to extrapolation. The reason for the later attachment of Richardson's name to the method might lie in the fact that the Philosophical Transactions had such a broad readership. However, we shall see that Richardson was, at best, only the fifth person to use "his" method in some sense. And, in fact, the algorithm did not reach its present form until Romberg's work (1955).

Romberg's contribution was a new implementation of Richardson's method in general, though Romberg was interested in the application to numerical integration. That application is now called Romberg integration. Because of the advantages of Romberg's implementation, Richardson extrapolation is now always done in the Romberg style, even for problems not involving integration. The reader of most texts might assume that Romberg was (just) the first person to apply Richardson's method to integration. This is not the case, as we shall now see: the "Richardson" method was first used in integration in 1900!

B. The Earlier Writers on Extrapolation

W. F. Sheppard (1900) used the Euler-Maclaurin series for the exact error made by the trapezoidal rule when using m equal sub-intervals on [a,b]; we give the theorem involved.

Theorem: Let $h = (b-a)/m$, $x_i = a + ih$ for $i = 0,\ldots,m$; let $f_i = f(x_i)$. Assume f and its first 2k derivatives are continuous on [a,b]. Let $A_j = B_j/j!$ for $j = 2,4,6,\ldots$, where the B_j are the Bernoulli numbers. Then there is a c in [a,b] such that

$$\int_a^b f(x)bx = h(\tfrac{1}{2}f_0 + f_1 + \ldots + f_{m-1} + \tfrac{1}{2}f_m) - A_2h^2(f'(b) - f'(a)) - A_4h^4(f^{(3)}(b) - f^{(3)}(a)) - A_6h^6(f^{(5)}(b) - f^{(5)}(a)) - \ldots - A_{2k}h^{2k}f^{(2k)}(c)(b-a). \quad (1)$$

Formula (1) assumes the notation for the Bernoulli numbers given in the CRC Standard Mathematical Tables (1974, p. 485). For example,

$$B_2 = 1/6, \quad B_4 = -1/30, \quad B_6 = 1/42, \quad B_8 = -1/30. \quad (2)$$

Krylov (1962, p. 216) gives a proof of the theorem.

If all derivatives of f are continuous, it seems natural to write

$$I = T(h) - Dh^2 - Eh^4 - Fh^6 - \ldots. \tag{3}$$

Strictly speaking, "=" is not always correct; but "$\sim$" is. We shall, however, use (3) with the proviso that more will be said on this topic, later.

We now discuss the main results of Sheppard (1900). Assume the trapezoidal sums T(h) and T(2h) are available. Then we can find a linear combination of them which has an error series beginning with h^4. For small h, such an approximation should be better than either T(2h) or T(h). The combination needed is easily obtained from

$$\begin{aligned} T(2h) &= I + 4Dh^2 + 16Eh^4 + 64Fh^6 + \ldots \\ T(h) &= I + Dh^2 + Eh^4 + Fh^6 + \ldots. \end{aligned} \tag{4}$$

Thus,

$$\frac{4T(h) - T(2h)}{3} = I - 4Eh^4 - 20Fh^6 - \ldots. \tag{5}$$

Sheppard noted some other interesting facts besides the elimination of the h^2 term. First, if the number m of subintervals is even, then it is easy to show

$$\frac{4T(h) - T(2h)}{3} = S(h), \tag{6}$$

where S(h) is Simpson's approximation with m subintervals and m/2 parabolas to approximate the curve over [a,b]. We have thus written a more sophisticated area approximation as a linear combination of trapezoidal approximations. Sheppard showed this could be done for a number of so-

phisticated approximations. Another fact that he noted concerned the price paid for the elimination of the h^2 term: the coefficients on the remaining terms are magnified. Consequently, if D were zero, the dominant error term in S(h) would be four times larger than the corresponding term of T(h). In such a case, we would do better to use T(h) than S(h), even though S(h) is "normally" more accurate for small h. Referring back to (1), we thus see that T(h) should be considered preferable to S(h) if $f'(b) = f'(a)$.

Sheppard also dealt with the simultaneous removal of many of the error terms. E.g., suppose T(4h), T(2h), and T(h) are available. Then

$$\begin{aligned} I &= T(4h) + 16Dh^2 + 256Eh^4 + 4096Fh^6 + \ldots \\ I &= T(2h) + 4Dh^2 + 16Eh^4 + 64Fh^6 + \ldots \\ I &= T(h) + Dh^2 + Eh^4 + Fh^6 + \ldots. \end{aligned} \tag{7}$$

Using p, q, and r as undetermined coefficients gives

$$\begin{aligned} (p+q+r)I = {} & [pT(4h) + qT(2h) + rT(h)] + Dh^2[16p + 4q + r] + \\ & Eh^4[256p + 16q + r] + Fh^6[4096p + 64q + r] + \ldots. \end{aligned} \tag{8}$$

Set the coefficient of I equal to 1 and the h^2 and h^4 coefficients to 0 to obtain

$$\begin{bmatrix} 1 & 1 & 1 \\ 16 & 4 & 1 \\ 256 & 16 & 1 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \tag{9}$$

Solve (9) to obtain

$$I = \frac{T(4h) - 20T(2h) + 64T(h)}{45} + \frac{652}{9}Fh^6 + \ldots. \tag{10}$$

Sheppard noted that this approximation is equivalent to what is called

Boole's integration rule, ordinarily obtained by using fourth-degree interpolating curves. (In our present day tables, the third column will be approximations given by Boole's rule.)

Sheppard extended our system (7) to allow elimination of any number of powers of h, given an appropriate number of trapezoidal approximations. Unfortunately, if you decide at the end of calculations that it would be better to eliminate another power of h, you essentially have to set up a new linear system and begin again. But no one before 1955, including Richardson, made any improvement on Sheppard's presentation of "Richardson" extrapolation, so far as the present author knows. It is not difficult to see why Joyce (1971, p. 477) considers it unfortunate that Sheppard's article has not been more widely known.

It is puzzling that the 1900 article was apparently never read by Richardson, though he does cite (1910, pp. 308, 310) one of Sheppard's articles published in the same Proceedings a year earlier. The only early writer who built on Sheppard's foundation was Buchanan (1902).

Milne (1903) used similar operations to reduce errors in a completely different problem. It concerned the approximation of the length of a circular arc by taking inscribed polygons. This process was of ancient interest, because it gives approximations for pi if the arc is a semi-circle with unit radius. We next summarize the basic idea for that special case.

Let the semi-circle be divided into n equal arcs with h as the measure of the central angles in radians. Therefore $h = \pi/n$. Let P(h) denote the total length of the corresponding inscribed polygonal approximation to the semi-circle. It is trivial to show from elementary geometry and trigonometry that

$$P(h) = \frac{2\pi}{h} \sin(\frac{h}{2}) = \pi - ah^2 + bh^4 - \ldots. \qquad (11)$$

Therefore, if we have P(h) and P(2h) available, we can use the same sort of approach as before, with a better approximation for pi given by

$$Q(h) = \frac{4P(h) - P(2h)}{3} \qquad (12)$$

It should be noted that this is a modern restatement of Milne's work; he never actually obtained anything quite so simple as (11) and (12), though what he did is logically equivalent. Using a larger number of P values and then truncating the series so as to obtain a system of K equations in K unknowns, he found even better approximations. However, his presentation is defective when compared with Sheppard's, because Milne did not see the new approximations had an error series with the lower powers of h removed. He apparently just felt, accurately, that the approximation obtained by solving the truncated linear system would be better than the approximations used in forming the system. Milne's claim to predating Richardson lies mainly in the fact that he did obtain the same results as Richardson's method gives, though his explanation was lacking a bit.

S. A. Corey (1906) likewise obtained some "Richardson-like" results before Richardson's first article; again, the method was truncation, which prevented his noticing the form of the error for the new approximation.

In his 1927 paper, Richardson did make some important points that had not been made before. For example, he noted that when physical problems were modeled by differential equations and solved approximately by central differences, the error series for the approximation generally

has no odd powers of h. He was the first to apply the method to problems of general scientific interest; even the titles of his articles indicate an interest in applied mathematics. But only in one example (1927, pp. 308-311) did he eliminate both the h^2 and h^4 terms. He showed no interest in proceeding further. He never used his method for improving estimates of a definite integral value.

C. The Modern Algorithm

As we have seen, the early writers always eliminated several error terms by setting up a linear system which eliminated all of them at once. As we have seen, this would be a bit inconvenient if one decided at the end that it would have been a good idea to eliminate a few more powers. Essentially you have to start over. Romberg (1955) was the first to see how to avoid this difficulty imposed by having to solve a linear system. We show now how his procedure works. It is so simple that it seems incredible no one devised it earlier.

Suppose we want to find a quantity X which is the limit, as h approaches zero, of some more easily accessible quantity A(h). Suppose there are constants D, E, F,..., such that, for every h,

$$X = A(h) + Dh^{2q} + Eh^{2q+2} + Fh^{2q+4} + \ldots. \tag{13}$$

We do not need to know the constants involved, except q. Then

$$\begin{aligned} X &= A(2h) + D4^q h^{2q} + E4^{q+1} h^{2q+2} + F4^{q+2} h^{2q+4} + \ldots \\ X &= A(h) + Dh^{2q} + Eh^{2q+2} + Fh^{2q+4} + \ldots. \end{aligned} \tag{14}$$

Take a linear combination as before, to obtain

$$X = \frac{4^q A(h) - A(2h)}{4^q - 1} + \frac{4^q(1 - 4)}{4^q - 1} Eh^{2q+2} + \frac{4^q(1 - 4^2)}{4^q - 1} Fh^{2q+4} + \ldots$$

$$\equiv B(h) - E^* h^{2q+2} - F^* h^{2q+4} - \ldots. \tag{15}$$

The sturcture of the error series for B(h) is the same as the error series for A(h); we can thus repeat the pattern if we have B(2h) available, computed from A(2h) and A(4h):

$$X = B(2h) - 4^{q+1} E^* h^{2q+2} - 4^{q+2} F^* h^{2q+4}$$

$$X = B(h) - E^* h^{2q+2} - F^* h^{2q+4} \tag{16}$$

Take a linear combination, to obtain

$$X = \frac{4^{q+1} B(h) - B(2h)}{4^{q+1} - 1} + \frac{4^{q+1}(1 - 4)}{4^{q+1} - 1} F^* h^{2q+4}$$

$$\equiv C(h) + F^{**} h^{2q+4} + \ldots. \tag{17}$$

The process can be continued. How far depends on our knowledge about the powers in the initial error series, and how many step sizes we are able to use in the initial stage of finding "A" values. If we halve the step size for each new A and call the first step size h, we have the following dependencies as set up by Romberg (see Figure 5).

Romberg's paper had to do with definite integral approximations based on the trapezoidal and mid-point rules. He did not restrict himself to step-halving, realizing that other patterns had possible advantages. But we will discuss only the use of the trapezoidal approximations; and at present we will also restrict ourselves to step-halving, since that is still the best known method.

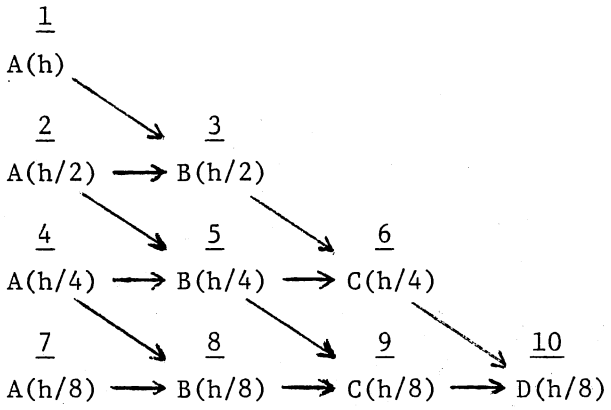

Figure 5. Dependencies in the Romberg Implementation of Richardson Extrapolation

D. Initial Illustrations on Romberg Integration

We now examine the algorithm's performance on several definite integral problems. The pattern of calculations is as in (13) - (17), since the hypotheses of the Euler-Maclaurin theorem will be satisfied. First, an example where the method succeeds well: the integral of 1/x from 1 to 2. The correct answer is, of course, ln(2), .693147181.... For the results obtained from the first five trapezoidal sums, see Table X. Correct digits are underlined.

The entry in the right-most positions has an error series beginning with h^{10} and is about 179,000 times as accurate as the best trapezoidal sum used, .693391.

Unfortunately there are some other fairly simple integrals where the extrapolation process leads to less satisfying results. For example, consider the results in Table XI, of applying the method to an example given by Bauer, Rutishauser, and Stiefel (1963). The best entry in each

row is underlined.

TABLE X

ROMBERG PERFORMANCE ON $\int_1^2 (1/x)\ dx = .693147818$

n					
1	.750000				
2	.708333	.694444			
4	.697024	.693254	.693175		
8	.694122	.693155	.693148	.693147478	
16	.693391	.693148	.693147194	.693147183	.693147182

TABLE XI

ROMBERG PERFORMANCE ON $\int_0^{10} 1/(1+x^2)\ dx =$
$\tan^{-1}(10) = 1.47113...$

n						
1	5.04950					
2	2.71706	1.93958				
4	1.74703	1.42368	1.38929			
8	1.49162	1.40649	1.40534	1.40560		
16	1.47120	1.46439	1.46825	1.46925	1.46950	
32	1.47111	1.47108	1.47153	1.47158	1.47159	1.47159
64	1.47112	1.47113	1.47113	1.47112	1.47112	1.47112

Bauer et al. (1963) imply that the optimal column moves toward

the right as we extend the table. This is true, but the movement is slow. The accuracy of the first few columns does become spectacular, but some deterioration still takes place at the end of the rows.

E. A Search for Factors Determining Performance

1. Rate of Divergence

Removing many error terms seemed to help greatly in the first example, but the same process seemed counter-productive in the second example. Stroud (1974) makes the enigmatic remark that when we extend (1) into the infinite series (3), the series is known to diverge for "most" functions. The present author then guessed that the series for the 1/x example converged nicely, while the series for the $1/(1+x^2)$ error diverged. This turns out to be quite incorrect, as we shall now see.

First, what could cause the series to diverge for "most" functions? Could it be the A_j constants in (1)? No. Using a known asymptotic expression for the Bernoulli numbers, it can be shown that for large j, $A_{j+1} \doteq (-1/4\pi^2)A_j$ (Engels, 1978, p. 389). Thus the A_j go to zero and the divergence must be caused by the derivative differences in (1) growing fast enough to cause divergence. But why would that generally happen? The reader who knows any complex variables theorems can show that if a function is analytic at "a" but has some singularities somewhere, and if a certain "often existing" limit does exist, then $|f^k(a)|$ does indeed have to go to infinity as k becomes large (Conway, 1973, pp. 30ff).

Let $f_1(x) = 1/x$ and $f_2(x) = 1/(1+x^2)$. It seemed promising to construct the terms of the actual error series for f_1 and f_2. This required

calculating high derivatives of the two functions. Higher derivatives of f_2 are nontrivial to calculate; but it does turn out to be possible to calculate the value of $f_2^k(x)$ without knowing the exact albebraic formula for $f_2^k(x)$. The reader with considerable manipulative skills might enjoy trying to obtain the following recursion:

$$f_2^{(n-1)}(x) = Q_n(x)/(1 + 1/x)^n, \text{ where}$$
$$Q_1(x) = 1/x, \quad Q_2(x) = -2/x, \quad \text{and} \tag{18}$$
$$Q_{n+1}(x) = -n[(n+1)(1 + 1/x^2)Q_{n-1}(x) + 2Q_n(x)].$$

The surprise is that the supposedly "nice" f_1 function has derivatives which grow much more rapidly than the supposedly "bad" f_2 function! The error series for f_1 diverges much more rapidly than the error series for f_2. After the first twenty-three terms, the coefficient of h^{46} is -4.598E+19 for the f_1 example, and -3.628E-27 for the f_2 example.

The question then arose, is having a rapidly divergent error series somehow helpful in the extrapolation process? This seemed attractive partly because it is obvious that the extrapolation process greatly encourages divergence as we move to the right. The later terms are magnified enormously. For example, it is not difficult to use the pattern in (15) to show that by the time we eliminate the h^2, h^4, h^6, and h^8 terms, the h^{48} coefficient is magnified by a factor of .9E+52.

However, when the error series (including the h values in the computations) were computed for the entries throughout Tables X and XI, the hypothesis, that the rate of divergence was crucial, was not supported. For example, the twentieth terms of the error series for the two tables were almost interchangeable, position for position. And in the success-

ful f_1 table, the rate of improvement slowed down as we move to the right and the more rapid divergences.

We must conclude that our evidence about the connection between rapid divergence and successful extrapolation seems rather mixed. In the first column, rapid divergence and good extrapolating went together; but the later columns do not support such a connection. It is tempting to guess that, in itself, the rapidity of divergence may simply be irrelevant.

2. The Removed Terms as an Error Estimate

However, it is clear that the performance is controlled by how good an approximation the removed term was for the true error. Let us be more precise: assume

$$\begin{aligned} T(2h) &= I + a(2h)^{2k}P_{2h} \\ T(h) &= I + ah^{2k}P_h. \end{aligned} \tag{19}$$

Then it follows quickly that

$$\begin{aligned} \frac{4^kT(h) - T(2h)}{4^k - 1} &= I + \frac{4^k}{4^k - 1}\left(1 - \frac{P_{2h}}{P_h}\right)(T(h) - I) \\ &= I + \frac{1}{4^k - 1}\left(\frac{P_h}{P_{2h}} - 1\right)(T(2h) - I) \end{aligned} \tag{20}$$

Thus, if the errors were exactly $100a(2h)^{2k}$ and $100ah^{2k}$, a perfect extrapolation would result. Thus it is not quite correct to say that the success of extrapolation depends on ah^k and $a(2h)^k$ being good approximations to the actual error. What is crucial is that those two terms should be about the same percentage of the respective errors. Of course, the easiest way to force this is to have both P_{2h} and P_h close to 1. Formula

(20) can be used to establish that the extrapolation will be an improvement on T(h) if and only if

$$\frac{1}{4^k} < \frac{P_{2h}}{P_h} < 2 - \frac{1}{4^k}, \tag{21}$$

with 1 as the ideal value for P_{2h}/P_h. The required interval is always contained in (0,2). Similarly, the extrapolation will be better than T(2h) if and only if a much less stringent requirement is met; namely,

$$2 - 4^k < \frac{P_h}{P_{2h}} < 4^k. \tag{22}$$

Notice also that if T(h) and T(2h) are on opposite sides of I, then P_{2h} and P_h have opposite signs and (21) insures that improvement is impossible. An oscillating column is a good sign in an Euler table, but not in a Richardson/Romberg table.

The calculation of the P_h values for the entries in the f_1 table (1/x) and the f_2 table $(1/(1+x^2))$ shows how the f_1 table attains its superiority. For example, consider the P_h values in the row with n = 8, for f_1: .9981, .9634, .7559, and .2932. The corresponding values in the f_2 table are -80, +20,870, 1.243E+5, and 9.369E+4. But we do still get some improvement for some P_h values faraway from 1: for example, in the f_2 table, the P_h value below the 1.243E+5 is 3.477E+5. This gives P_{2h}/P_h = .36, which is in the acceptable range (.03,1.97) for that column. And in the f_1 table, the P_h value below the .2932 is .6341, giving a ratio of .46, well within the desired range, (.02,1.98). Of course the resulting improvement is not nearly so good as the corresponding one in the second column, which results from having a .9995 P_h value under the .9981 value.

The extrapolation process itself tends to force the P_h values away

from "1" as you move along a row, because of the magnification of the later terms in the series. For example, if we began with unit coefficients on all the powers ($h^2, h^4, \ldots$), then the ratio of the second term to the first increases as we move along the row: from $1h^2$ to $5h^2$ to $21h^2$ to $85h^2$ to $341h^2$, in the first five columns. The result is a deteriorating rate of improvement as we move along the row, even in the f_1 table: e.g., in the row with $n = 16$, the error is multiplied by .0019, then .0291, then .1831, then .5397.

The author has not seen the P_h idea elsewhere, but it does seem to help in understanding what is happening. We now move on from f_1 and f_2 to examples which illustrate other points about the process.

F. Examples on Other Points

An illustration given by Davis (1959) makes clear what the infinite series in (3) signifies. Let us compute the Romberg table with

$$f_3(x) = \frac{.095}{1.81 - 1.80\cos(x)}$$
$$\int_0^{2\pi} f_3(x)\, dx = \pi \qquad (23)$$

All requirements of the Euler-Maclaurin theorem are satisfied. But f_3 has a period of 2π, making all the terms in (3) equal to zero. Does this mean that the error is zero for any h? Absolutely not. See Table XII.

One aspect of the behavior in Table X is made clear by referring back to (1), the original Euler-Maclaurin series. No matter how far we choose to extend the series, all the weight is in the last term, the one that is never included in (3). The extrapolation process is continually trying to remove terms that are not there. This hinders rather than

helps. Suppose we have (for f_3, any k is appropriate)

$$\begin{aligned} I &= T(2h) - A\, 4^k h^{2k} f^{2k}(c) \\ I &= T(h) - A\, h^{2k} f^{2k}(c_1). \end{aligned} \tag{24}$$

Then the first extrapolation will give

$$I = \frac{4T(h) - T(2h)}{3} - A\, h^{2k} f^{2k}(c_1)\left(\frac{4}{3} - \frac{4^k f^{2k}(c)}{3 f^{2k}(c_1)}\right) \tag{25}$$

The 4^k is a very bad sign of things to come. And sure enough, the extrapolations in Table X are not improvements over the elements used in computing them.

TABLE XII

PERFORMANCE OF ROMBERG INTEGRATION ON $\int_0^{2\pi} f_3(x)\, dx = \pi$

n							
1	59.69						
2	29.93	20.00					
4	15.13	10.20	9.54				
8	7.89	5.48	5.16	5.09			
16	4.57	3.46	3.33	3.30	3.29		
32	3.37	2.96	2.93	2.92	2.92	2.92	
64	3.15	3.08	3.08	3.09	3.09	3.09	3.09

The other main aspect of Table X is much more encouraging and shows what the infinite series (3) does imply when all the terms are zero:

namely, the error function $I - T(h)$ is asymptotic to the zero function and thus goes to zero (as $h \to 0$) faster than any polynomial in h. The column-one entry for $n = 512$, $h = 2\pi/512$, has an error of only .222E - 13. In a way, the failure of extrapolation causes no great concern here, because the trapezoidal sums themselves converge very rapidly to the answer. They do have difficulty at first because the vast majority of the area under the curve is at the ends of the interval.

Another example which shows the importance of one of the Euler-Maclaurin hypotheses is

$$f_4(x) = \frac{3}{2}x^{\frac{1}{2}}$$
$$\int_0^1 f_4(x)\, dx = 1. \tag{26}$$

We almost have the requirements of the theorem, but the derivatives fail to be continuous at 0. We can no longer be sure that the error series for $T(h)$ is an h^2 series; in fact it is not, though the h^2, h^4,... terms are present and it does help a bit to remove them. The rows of the Romberg table improve slightly as we move to the right, but each new "9" in the approximation requires about four times as many function evaluations as the previous decimal place required. The rate of improvement along the top diagonal is no greater than the rate of improvement for the trapezoidal sums themselves. We shall see later how to modify the Romberg algorithm to make the extrapolation on this example much more successful. But the point for now is simply that if the hypotheses assumed by the Euler-Maclaurin series are not completely satisfied, then the extrapolations based on that series are also not likely to succeed.

G. Quality Estimates on Extrapolations

By this point it seems reasonable to approach the question of how we can detect whether extrapolation is justified. The answer is not very complicated. If the first remaining term in the error series for a sequence of approximations V(4h), V(2h), V(h) is an h^{2k} term, and if the P_h, P_{2h}, P_{4h} are used as before,

$$V(4h) = A + C16^k h^{2k} P_{4h} \tag{27}$$

$$V(2h) = A + C4^k h^{2k} P_{2h}$$

$$V(h) = A + Ch^{2k} P_j$$

Then

$$\frac{V(4h) - V(2h)}{V(2h) - V(h)} = 4^k \left(\frac{4^k P_{4h} - P_{2h}}{4^k P_{2h} - P_h}\right). \tag{28}$$

If P_{4h}, P_{2h}, P_h are all approximately equal, which is all we need for successful extrapolation, then

$$\frac{V(4h) - V(2h)}{V(2h) - V(h)} \doteq 4^k \tag{29}$$

If we are proposing to use the V column for extrapolation and assume the first term in the error is an h^{2k} term, we can thus get a measure of how good the extrapolation based on that assumption is likely to be, by noting how close (30) is to a true equality:

$$\frac{V(h) - V(2h)}{V(2h) - V(4h)} 4^k \doteq 1. \tag{30}$$

(29) is itself often used as a quality estimate (Conte and deBoor, 1972,

p. 316). But if one wishes to use variations beside step-halving, it seems most convenient to set up quality estimators which always approach "1" when the procedure is working correctly. See Table XIII, where the quality estimators were calculated assuming the usual h^2, h^4, ... series. Note that if the quality estimate is not close to "1", virtually anything may happen to the error via extrapolation. But when the estimator is near 1, the extrapolation invariably multiples the error by a number considerably less than 1. Joyce (1971) attributes the first use of (30) to Lynch (1965).

TABLE XIII

QUALITY ESTIMATES FOR THE FIRST FIVE COLUMNS OF ROW SIX (n=64), WITH THE ACTUAL MULTIPLICATIVE EFFECT ON ERROR VIA EXTRAPOLATION FROM THAT COLUMN'S ELEMENTS IN ROWS FIVE AND SIX

f_1	1.0005	1.0076	1.0626	1.3744	2.7764
	(.0001)	(.0019)	(.0160)	(.0911)	(.3053)
f_2	.54707	.10781	-7.7725	-50.125	-228.92
	(.0007)	(-1114.)	(-1.104)	(1.540)	(1.089)
f_3	.71660	-3.6377	-24.776	-111.05	-456.70
	(-8.708)	(.8824)	(.9569)	(.9882)	(.9970)
f_4	1.4398	5.6570	22.627	90.510	36.204
	(.4006)	(.8781)	(.9710)	(.9928)	(.9982)

H. Theorems and Implications in Use

Having seen the kinds of success and failure that the Richardson/Romberg method can produce, the question arises as to what hypotheses will always guarantee success. Most of the basic theorems on Romberg integration are to be found in the important paper by Bauer, Rutishauser, and Stiefel (1963). We will now informally discuss the main ones and their implications.

The first crucial theorem insures that if f is continuous on [a,b], then all columns of the Romberg table must converge to the correct value of the definite integral. This is certainly a believable result, regardless of the difficulty of the proof.

To see what the second main theorem asserts, use system (19) to show

$$\frac{T(2h) - I}{T(h) - I} = 4^k \frac{P_{2h}}{P_h}. \tag{31}$$

The theorem asserts, translating into our terms, that if enough derivatives of f are continuous to let us extend the removal process to the h^{2k} column, then (for small h), P_{2h}/P_h will be close to 1 and we must have successful extrapolation. More precisely, if we proceed deeply enough into the column where the error series begins with h^{2k}, any entry will have an error about 4^k times as large as the error of the entry underneath it.

The third theorem we shall discuss requires f, considered as a function of a complex variable, to be analytic in an open domain of $\mathbb{C}$ which contains [a,b]. The conclusion is that as we proceed down any diagonal, the ratio of consecutive errors goes to zero. Again, reinterpreting the theorem in our terms is easy. Referring back to (20), the crucial ratio

is, for the h^{2k} column,

$$\frac{1}{4^k - 1}\left(\frac{P_h}{P_{2h}} - 1\right). \tag{32}$$

The theorem implies that, under the stated conditions, P_h/P_{2h} is not too far from 1. So the limit of (32) will be zero.

Unfortunately, as often happens in mathematics, these limit theorems are not as conclusive in practical computing as we might hope. We shall soon see an example where round-off error will actually result in the columns getting worse as we proceed down, no matter how far we proceed. As for the other two theorems, note that $1/(1+x^2)$, which causes some difficulty in extrapolation, satisfies the hypotheses of both the theorems. We are guaranteed by the theorems that if we are able to proceed far enough down any column or diagonal, and if round-off error remains insignificant, then the stated limits will <u>eventually</u> be approached. The problem with the f_2 example is that the conclusions are just beginning to take effect when we reach the limits of our finite precision arithmetic and the limits of how much computer time we are willing to use. (Each new row added essentially doubles the total number of function evaluations needed.) For example, as we proceed down the second column of the f_2 table, through the row with $n = 512$, the consecutive error ratios are essentially 10, 1, 10, 149, 16703, 57, 16, and 16. In the later columns, the error ratios are not yet close to converging at this point in the table.

The same sort of thing is happening on the diagonals. The ratio of consecutive errors is decreasing steadily but is only down to .001 (on the top diagonal) by the time we reach the tenth row of the table.

The last theorem we shall discuss was not in the 1963 paper, but it

was in an earlier paper by Rutishauser and Stiefel (1961). This theorem essentially says that if f is an arbitrary polynomial of degree 2m or 2m+1, then m extrapolations should be required to produce the exact integral. All the entries in the $m^{\underline{th}}$ extrapolation column should be identical and should be absolutely correct. This points out the main fault of Romberg integration. To integrate a fifteenth degree polynomial precisely, eight rows and columns are necessary in the table. This requires 129 function evaluations, even if we can avoid evaluating f(x) more than once for any x used in the trapezoidal sums. Based on this criterion alone, the Romberg method would be very inferior to the Newton-Cotes and Gaussian methods of integration. They require only fifteen and eight function values, respectively, for exact integration of any fifteenth degree polynomial. But Bauer et al. (1963) point out some other considerations which are important. Romberg integration is the only one of the three methods which permits easy, iterative calculation of the higher order approximations, based on the lower level approximations already calculated. And the higher order Newton-Cotes methods have a rather frightening feature: mathematicians have constructed continuous functions such that the Newton-Cotes methods do <u>not</u> converge to the proper answer. Davis (1962, pp. 482-483) gives such a function. Stroud (1974, pp. 106-140) gives a good discussion of these two alternatives to Romberg integration.

At this point, we should perhaps answer a question that may have occurred to the reader: why measure a method by its ability to integrate a polynomial, anyway? Certainly we do not need a numerical method to integrate polynomials. The answer involves a "big" theorem in analysis: any continuous function on [a,b] can "essentially" be "followed" by a

polynomial, if we make the degree high enough. Bartle (1964, pp. 177-182) gives a more precise statement. Thus integrating arbitrary continuous functions is a lot like integrating polynomials.

Before we move on to other things, we should also answer another question we have raised: how do you arrange to calculate the successive trapezoidal sums so that f is never evaluated at any x more than once? Staying on [0,1] for simplicity, the scheme is this: with S_1, S_2, S_3, ... as the trapezoidal sums, compute via

$$\begin{aligned} S_1 &= (f(0) + f(1))/2 \\ S_2 &= S_1/2 + f(1/2)/2 \\ S_3 &= S_2/2 + (f(1/4) + f(3/4))/4 \\ S_4 &= S_3/2 + (f(1/8) + f(3/8) + f(5/8))/8 \\ &\vdots \end{aligned} \tag{33}$$

I. Maximizing Accuracy of the Initial Sums

Before discussing other variations of Romberg integration besides step-halving, we shall discuss a refinement which may be used either with step-halving or some other method. This refinement was suggested by Rutishauser (1967) for getting more accurate sums in the initial column of the Romberg table. Wallick (1970) found that if one hopes to reach full machine accuracy with the extrapolations, then Rutishauser's summation algorithms may be needed to attain the last digit or two. We now explain this method, which could be used in summing any list of numbers.

The difficulty in adding a long string of numbers on the computer is that, by the time we approach the end, we may be adding comparatively small terms onto comparatively large partial sums. In finite precision

arithmetic, this can result in a considerable error. Rutishauser tries to minimize this problem by having three levels of summation. Call our sum variables PP (for "pre-pre-sum"), P (for "pre-sum"), and S (for "sum"). We will choose another variable "NG", for "number in a group". The idea is to divide the collection of numbers into groups of NG terms, each. The groups are themselves divided into collections of NG groups, each. The summation goes thus: after NG terms are added up in PP, the collection sum P is updated via PP and PP is reset to zero to begin the next group sum. When P has been updated NG times (i.e., after $(NG)^2$ terms have been taken into account) the sum S is updated via P, and both P and PP are reset to zero to begin the next collection sum. Assume all the initial terms to be added together are of approximately equal size. Then it is only on the final "S" level that a quantity will ever be added onto another one which is more than about NG times as large. Raising NG will cut down on this effect in the "S" level, at the price of less balanced summations on the earlier levels. The reader might find it instructive to follow the process in adding up the numbers from 1 to 32--first with NG = 2, then NG = 3. Rutishauser suggests NG = 16, though most of the sums for this thesis were done with NG = 8. To show that all this does make a difference on the computer, the trapezoidal sums for x^3, from -2 to 0, were computed repeatedly with different values of NG. To make sure the error being measured was essentially due only to the summing, all the variables were kept in double precision except for the variables PP, P, and S, used in summing the new function values each time. If the summing process itself adds no extra error, the trapezoidal sums obtained should be correct to between seven and eight digits. The amount of error introduced by the summing can reasonably be measured in units of

error in the eighth place. See Table XIV, which indicates that for this problem, setting NG = 4 is the best strategy. The correct sum for n subdivisions of [-2,0] is easily calculated to be $-4(1+1/n^2)$; this gave the standards for comparison. (The number n was taken to be powers of two and the triples of those numbers, rather than just powers of two. This is immaterial for the point at hand, however. That subdivision scheme will be discussed next.)

TABLE XIV

UNITS OF ERROR IN THE EIGHTH PLACE, FOR x^3 SUMS WITH VARIOUS NG VALUES AND N SUBDIVISIONS

	N:	96	128	192	256
	1	79	26	88	112
	2	28	0	30	19
	4	25	4	21	16
NG	8	34	14	35	25
	16	56	28	52	48
	32	79	35	65	55
	64	79	74	88	76

J. Generalized Step Sequences

1. Overview and the Use of Interpolation Theory

At this time, we will begin working toward the goal of alleviating the main weakness of "step-halving" Romberg integration: the large num-

ber of function evaluations needed to add a row to the table. As we shall see, there is a price to be paid, both in simplicity and susceptibility to round-off error in the calculations. Nevertheless, on balance, one of the variations to be examined is possibly preferable to step-halving in most problems. The other variation we will discuss, while previously rejected out of hand because of susceptibility to round-off error, has been recognized lately as a legitimate method of economizing when high accuracy is not required. However, the reader will have to be patient for a while as we develop the (less obvious) formulas for the arbitrary step sequences. The argument followed for step-halving (equations (13) - (17)) seems to be impossibly difficult to implement in this new context. But help will come from taking a different approach, based on the connection between Romberg integration and interpolation.

First, recall the "truncation" approach which would have been used by some of the older writers to eliminate the h^2 and h^4 errors and obtain a new approximation I*; solve

$$\begin{aligned} S_0 &= I^* + ah_0^2 + bh_0^4 \\ S_1 &= I^* + ah_1^2 + bh_1^4 \\ S_2 &= I^* + ah_2^2 + bh_2^4 \end{aligned} \tag{34}$$

The I* obtained is the same approximation Sheppard obtained with his more sophisticated technique; and I* can be interpreted in terms of interpolation: I* is $P_2(0)$, where $P_2(x)$ is the unique second degree polynomial satisfying

$$P_2(h_0^2) = S_0, \quad P_2(h_1^2) = S_1, \quad P_2(h_2^2) = S_2. \tag{35}$$

So far as the Romberg table is concerned, I* is the entry two columns to

the right of S_2. Adopt the notation of Stroud (1974, p. 150) and denote I* by $P_2(0: h_0^2,h_1^2,h_2^2)$. A similar interpretation can be put on all the entries of the table. We then obtain Table XV, as in Stroud (1974).

TABLE XV

THE ROMBERG TABLE IN TERMS OF INTERPOLATING POLYNOMIALS

$S_0 = P_0(0: h_0^2)$		
$S_1 = P_0(0: h_1^2)$	$P_1(0: h_0^2,h_1^2)$	
$S_2 = P_0(0: h_2^2)$	$P_1(0: h_1^2,h_2^2)$	$P_2(0: h_0^2,h_1^2,h_2^2)$

This lends some light on deterioration along the rows, if the reader is familiar with interpolation characteristics (Shampine and Allen, 1973, p. 42).

We can now use a recursion discovered by Neville (1932),

$$P_n(x: x_0,\ldots,x_n) = \frac{x - x_0}{x_n - x_0} P_{n-1}(x: x_1,\ldots,x_n) + \frac{x_n - x}{x_n - x_0} P_{n-1}(x: x_0,\ldots,x_{n-1}). \tag{36}$$

Let T_n^i denote the entry in column n and diagonal i (i,n = 0,1,2,...); then set x = 0 in (36) to give

$$T_n^i = \frac{(\frac{h_i}{h_{i+n}})^2 T_{n-1}^{i+1} - T_{n-1}^i}{(\frac{h_i}{h_{i+n}})^2 - 1} . \tag{37}$$

It is routine to show that when step-halving is used we obtain the more common formula.

For FORTRAN implementation, relabel the table with $T_{j,k}$ being the entry on row j and column k, with j and k = 1,2,3,.... Using N_i as the number of subintervals corresponding to h_i, we obtain

$$T_{n,k} = \frac{(\frac{N_n^2}{N_{n-k+1}^2}) T_{n,k-1} - T_{n-1,k-1}}{(\frac{N_n^2}{N_{n-k+1}^2}) - 1} . \tag{38}$$

The extrapolation (38) is equivalent to assuming there is a constant A_k such that

$$T_{n-1,k-1} = I + A_k h_{n-1}^2 \ldots h_{n-k+1}^2 \tag{39}$$

$$T_{n,k-1} = I + A_k h_n^2 \ldots h_{n-k+2}^2 .$$

As before, we can calculate a quality estimate which will be near 1 if the extrapolation is justified. Namely,

$$(\frac{T_{n,k-1} - T_{n-1,k-1}}{T_{n-1,k-1} - T_{n-2,k-1}})(\frac{N_n^2}{N_{n-k}^2})(\frac{N_{n-1}^2 - N_{n-k}^2}{N_n^2 - N_{n-k+1}^2}) \doteq 1. \tag{40}$$

The present author has not seen (40) elsewhere, but it is an obvious extension of (30), based on the known form of error in (39), which was

given by Bulirsch (1964) and also Gragg (1971). The use of (40) will be illustrated later. Engels (1980, pp. 376-380) gives a proof that correct convergence of each column is "mathematically" assured if the h_i approach zero faster than some geometric null sequence. (As we shall see, "computational" assurance is not implied in finite precision arithmetic.)

Each additional h_i makes it possible to add one more column to the table. We therefore desire to add as many h_i as possible, but with no more new function evaluations than necessary.

2. Two Alternatives to Step-Halving

The first scheme that comes to mind is likely to be exchanging the N_i sequence 1,2,4,8,16,32,... of step-halving for the N_i sequence 1,2,3,4,5,6,..., which gives the h_i sequence $1,\frac{1}{2},\frac{1}{3},\frac{1}{4},\frac{1}{5},\frac{1}{6},\ldots$. We will call this method the "harmonic" one for the obvious reason. Implementation of this method requires a new scheme for economy in the trapezoidal sums. For each h_i (N_i) we will need to store intermediate sums as we progress. See (41), where we again assume [a,b] = [0,1].

$$
\begin{aligned}
F_1 &= (f(0) + f(1))/2 \\
F_2 &= f(1/2) \\
F_3 &= f(1/3) + f(2/3) \\
F_4 &= f(1/4) + f(3/4) \\
&\vdots \\
F_N &= \sum_{\substack{k=1 \\ (k,N)=1}}^{N-1} f(k/N).
\end{aligned}
\tag{41}
$$

Here (k,N) denotes the greatest common divisor of k and N. Just for an

example, the twelfth sum will be formed via

$$S_{12} = \frac{1}{12}\,[F_1 + F_2 + F_3 + F_4 + F_6 + F_{12}]. \tag{42}$$

The Euclidean division algorithm can be used to decide whether (k,N) = 1 (Agnew, 1972, p. 32).

The other method we shall discuss in detail was suggested by Bulirsch (1964): let N_i = 1,2,3,4,6,8,12,16,...; i.e., all powers of two and the triples of those numbers. This is called the "Q" method of subdivision. This method is going to include the trapezoidal sums obtained by step-halving, but with an extra sum between consecutive "step-halving" sums. Again, a suitable scheme has to be developed for use in the initial column. Define the F_i as in (43):

$$\begin{aligned}
F_2 &= f(1/2)/2 \\
F_3 &= (f(1/3) + f(2/3))/3 \\
F_4 &= (f(1/4) + f(3/4))/4 \\
F_5 &= (f(1/6) + f(5/6))/6 \\
F_6 &= (f(1/8) + f(3/8) + f(5/8))/8 \\
F_7 &= (f(1/12) + f(5/12) + f(7/12) + f(11/12))/12,
\end{aligned} \tag{43}$$

etc. The general pattern becomes clear from S_4, on:

$$\begin{aligned}
N = 1,\ & S_1 = (f(0) + f(1))/2 \\
N = 2,\ & S_2 = S_1/2 + F_2 \\
N = 3,\ & S_3 = S_1/3 + F_3 \\
N = 4,\ & S_4 = S_2/2 + F_4 \\
N = 6,\ & S_5 = S_3/2 + F_2/3 + F_5 \\
N = 8,\ & S_6 = S_4/2 + F_6 \\
N = 12,\ & S_7 = S_5/2 + F_4/3 + F_7 \\
& \vdots
\end{aligned} \tag{44}$$

Surely (41) and (44) are well-known to the experts in this area; but the present author has not yet seen the formulations in print.

In spite of the slightly more complicated implementation, the two variations we have discussed have a great economy advantage insofar as reaching the later rows (and thus, the desirable later columns) is concerned. See Table XVI, based on one given by Bulirsch (1964).

TABLE XVI

NUMBER OF FUNCTION EVALUATIONS NEEDED TO ELIMINATE A SPECIFIED NUMBER OF TERMS FROM THE ERROR

Degree of Last Term Removed From Error	Halving	Q	Harmonic
2	3	3	3
4	5	5	5
6	9	7	7
8	17	9	11
10	33	13	13
12	65	17	19
14	129	25	23
16	257	33	29
18	513	49	34
20	1025	65	44

Unfortunately, Table XVI is rather misleading when taken by itself. Two factors are at work to cut into the advantages of the more economical methods. The first factor, often not mentioned, is that any given column

of the "halving" table is more accurate than the same column of the other two tables. The superiority begins even in the initial column: the seventh trapezoidal sum in the "halving" table is based on sixty-four subdivisions of [a,b], compared with twelve and seven subdivisions in the other two methods. Bulirsch (1964) showed that for integration on [0,1], the accuracy along the top diagonal can be expressed by

$$\int_0^1 f(x)\,dx - T_m^0 = K_m f^{2m+2}(c), \qquad m = 0,1,2,... \tag{45}$$

for some c in (0,1), where K_m is independent of f but depends on which of the three methods we choose. Bulirsch gives the exact form of the various K_m and calculates their exact values. See Table XVII. The most desirable K's are the smaller ones, of course. On this basis, the order of preference would be step-halving, Q, and then the harmonic method.

TABLE XVII

$|K_m|$ FOR THE THREE METHODS

Degree of Last Term Removed From Error	Halving	Q	Harmonic
2	3E-4	3E-4	3E-4
4	5E-7	1E-6	1E-6
6	2E-10	1E-9	1E-9
8	2E-14	1E-12	1E-12
10	5E-19	4E-16	1E-15
12	3E-24	7E-20	5E-19
14	5E-30	7E-24	2E-22

For the definite integral of 1/x from 1 to 2, the new variations produce more accurate answers for a set number of function evaluations--at least, when the number is small. However, as the number grows larger, the harmonic method falls behind quickly and the advantage of "Q" gradually vanishes. See Table XVIII. It would suggest that if the values of f(x) are difficult to obtain, the new variations might be quite helpful.

TABLE XVIII

ABSOLUTE VALUES OF ERRORS IN THE TABLES
FOR $\int_1^2 \frac{1}{x}\,dx$

row		33 f(x) evaluations				
(6)	halving	.6E-4	.3E-7	.2E-9	.1E-10	.4E-11
		.2E-11				
(9)	Q	.1E-3	.5E-7	.2E-9	.3E-11	.1E-12
		.2E-13	.5E-14	.2E-14	.1E-14	
(10)	harmonic	.6E-3	1.E-6	.7E-8	.1E-9	.7E-11
		.6E-12	.9E-13	.2E-13	.8E-14	.5E-14
row		65 f(x) evaluations				
(7)	halving	.2E-4	.2E-8	.4E-11	.6E-13	.5E-14
		.2E-14	.1E-14			
(11)	Q	.3E-4	.3E-8	.3E-11	.1E-13	0
		.3E-15	.4E-15	.4E-15	.4E-15	.4E-15
(14)	harmonic	.3E-3	.2E-6	.8E-9	.7E-11	.1E-12
		.3E-13	.5E-13	.8E-13	.12E-12	.14E-12
		.15E-12	.15E-12	.16E-12		
row		129 f(x) evaluations				
(8)	halving	.4E-5	.1E-9	.6E-13	0	.3E-15
		.3E-15	.3E-15	.3E-15		
(13)	Q	.7E-5	.2E-9	.4E-13	.3E-15	.4E-15
		.5E-15	.6E-15	.6E-15	.6E-15	.7E-15
		.8E-15	.8E-15	.8E-15	.8E-15	

The deterioration at the ends of the rows in Table XVI, especially for the harmonic method, suggests that another factor besides the K_m values is affecting the accuracies: namely, round-off error. We now examine how this propensity to deterioration is different for each of the three methods.

Consider the integral

$$\int_{-15}^{20} \frac{21}{5}\left(\frac{x}{5} - 3\right)^{20} dx = 1. \tag{46}$$

For each of the three methods, every number in the tenth extrapolation column "should" be 1. This in fact does not happen, and the differences between the three columns are the result of the three methods having different susceptibilities to round-off error. Further, continuing downward in the columns increases the error due to round-off. See Table XIX.

TABLE XIX

MAGNITUDE OF ERRORS IN A COLUMN THAT WOULD BE EXACT WITHOUT ROUND-OFF ERROR (NUMBER OF FUNCTION EVALUATIONS IN PARENTHESES)

Halving		Q		Harmonic	
(1024)	.9E-15	(65)	.27E-13	(43)	.16E-11
(2048)	.9E-15	(97)	.22E-13	(47)	.31E-11
		(129)	.28E-13	(59)	.54E-11
		(193)	.24E-13	(65)	.10E-10

Thus, the variations on step-halving may not be able to attain as high an accuracy as step-halving attains, regardless of how far the table is extended. Nevertheless, Table XIX should not be allowed to obscure the fact that the variations attain the less demanding accuracies faster than step-halving: The .27E-13 error attained by "Q" is not matched by any entry in the halving table based on less than 256 function evaluations. The .16E-11 error for the harmonic method is not matched by halving until 128 function evaluations are used.

Another way to measure the susceptibilities of the three methods to round-off error is to find the maximum magnification on a prescribed perturbation in the initial column. This is mentioned, for example, by Gragg (1971). Certainly every entry, in whichever table we use, is a linear combination of trapezoidal sums in the first column. Define the coefficients $C^m_{k,j}$ as follows:

$$T_{n,k} = \sum_{j=1}^{k} C^n_{k,j} \, S_{n-k+j}, \qquad 1 \leq k \leq n. \tag{47}$$

(The reader would probably find it helpful to write out the sums for several elements in the $T_{n,k}$ table.)

Suppose that each S_i is perturbed by a small quantity X_i, where all X_i have a magnitude less than some positive X. Then, for any $T_{n,k}$, the sum of absolute values of the associated coefficients is a measure of the possible resulting perturbation $Y_{n,k}$ in $T_{n,k}$; thus

$$T_{n,k} + Y_{n,k} = \sum_{j=1}^{k} C^n_{k,j}(S_{n-k+j} + X_{n-k+j}), \tag{48}$$

so that

$$|Y_{n,k}| \leq \left(\sum_{j=1}^{k} |C^n_{k,j}| \right) X. \tag{49}$$

To find the recursion for the C's, use (47) to rewrite (38), clear the fractions and equate corresponding S coefficients on both sides. One will be led to extend the sums on one side by defining

$$C_{k-1,0} = 0 \quad \text{and} \quad C_{k-1,k} = 0. \tag{50}$$

The result is finally, for $k \geq 2$ and $1 \leq j \leq k$,

$$C^m_{k,j} = \frac{\left(\frac{N_m^2}{N_{m-k+1}^2}\right) C^m_{k-1,j-1} - C^{m-1}_{k-1,j}}{\left(\frac{N_m^2}{N_{m-k+1}^2}\right) - 1} \tag{51}$$

This a natural extension of a formula obtained by Bauer et al. (1963); the superscripts were unneeded for step-halving, since the row number makes no difference. We are here using a slightly different notation than those authors.

The implementation of (51) is quite similar to the implementation of (38), though there are some slight complications due to having the superscripts. See Figure 6 for a schematic diagram.

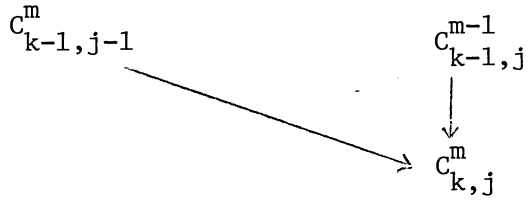

Figure 6. Pattern of Calculation for the C's, by Rows

The C's were calculated for each entry in the early rows of the Romberg table, for each of the three methods, and the absolute values were summed to give an "error magnification" bound for each entry. These bounds indicate clearly the differences in susceptibility to round-off error, as we vary from table to table or even from one position to another in the same table. See Table XX.

TABLE XX

ERROR MAGNIFICATION BOUNDS IN THE SECTION OF THE TABLES FROM $T_{4,5}$ to $T_{9,5}$ to $T_{9,10}$
($T_{0,1} = T_0^0$)

			Halving		
1.9641					
"	1.9680				
"	"	1.9692			
"	"	"	1.9692		
"	"	"	"	1.9693	
"	"	"	"	"	1.9693
			Q		
6.3					
8.2	8.4				
6.8	7.3	7.4			
8.2	8.7	9.0	9.1		
6.8	7.3	7.5	7.6	7.7	
8.2	8.7	9.0	9.1	9.2	9.2
			Harmonic		
13					
25	26				
48	54	56			
84	106	116	119		
140	195	233	251	256	
222	343	446	513	545	553

The entries in Table XX agree well with the upper bounds on the magnifications given in the literature: Bauer et al. (1963) give the upper bound for step-halving as 1.969.... Bulirsch (1964) gives the upper bound for "Q" as less than 9.3. And virtually everyone mentions that there is no upper bound for the harmonic method. (E.g., Bauer et al. (1963).)

Bulirsch, according to Joyce (1971), chose his own "Q" as the best all-around method, reasoning that the reduction in work justified accepting four times as much susceptibility to round-off error. And for some time, everyone assumed the unbounded magnifications for the harmonic method made it useless. For example, in their very influential paper, Bauer et al. (1963, p. 204) said that the harmonic method "causes severe numerical instability and therefore cannot be used practically."

Laurie (1975, p. 277) noted that the 1963 article had given no examples, and he proceeded to show that the instability is "only mild", resulting in a loss, at worst, of about two-fifths of a significant decimal digit for each new row or column added.

Laurie's mathematical arguments are quite involved, but we can experimentally reach comparable results without much effort. The harmonic section of Table XX indicates that, on any row, the last entry is the one most susceptible to round-off error magnification. Therefore we restrict ourselves to consideration of the top diagonal for our worst-case analysis. Assume that we have n digits for each number in the initial column, and we permit a perturbation of up to 1 in the kth digit of each of them. Then the numbers in the last part of Table XX can all be taken as the possible resulting perturbations, measured in units of variation in the kth digit. When we reach a possible magnification of 10, we have lost

10 units in the kth place; i.e., we have lost an extra significant digit. When the magnification reaches 100, we have lost two digits more than were lost in the initial column. In general, a reasonable way to measure the loss of digits is thus to take $\log_{10}$ of the magnification factors. If we do this for the entries along the top diagonal, the difference between consecutive logarithms is the number of additional digits lost by adding the new row. As it turns out, each new diagonal element loses a bit more accuracy than the preceeding one. However, taking the second differences indicates that the increase in losses is steadily declining. See Table XXI.

TABLE XXI

MAXIMUM LOSS OF SIGNIFICANT DIGITS ALONG THE TOP DIAGONAL OF THE TABLE FOR THE HARMONIC METHOD

Magnification Factors	Total Additional Digits Lost	New Loss	Increase in Loss
1	0		
1.6667	.22186	.2219	
3.1333	.49600	.2741	.0522
6.2127	.79328	.2973	.0232
12.694	1.1036	.3103	.0130
26.441	1.4223	.3187	.0084
55.823	1.7468	.3245	.0058
119.03	2.0757	.3289	.0044
255.73	2.4078	.3321	.0032
552.76	2.7425	.3347	.0026
1200.6	3.0794	.3369	.0022
2618.5	3.4181	.3387	.0018
5730.1	3.7582	.3401	.0014

The behavior of the last two columns makes it easy to believe Laurie's assertion that the loss per additional extrapolation will never go above .3574 digits. He tested several examples to confirm his predictions and concluded that the harmonic method is a plausible choice if we do not need too much accuracy. Havie (1977) agrees with Laurie's conclusions and extends them somewhat.

It seems appropriate to mention here that the quality estimate (40) appears to be a fairly good indicator of when the round-off error begins dominating the calculations. See Table XXII.

TABLE XXII

QUALITY ESTIMATES TOGETHER WITH THE ACTUAL MULTIPLICATIVE FACTOR APPLIED TO THE SIZE OF THE ERROR BY EXTRAPOLATION, FOR THE HARMONIC METHOD ON $\int_1^2 \frac{1}{x}\,dx$

Function Evaluations					
65	1.00023	1.00155	1.00504	1.01084	
	(.0007)	(.003)	(.009)	(.02)	
	.857530	-2.8	-10	-18	...
	(.2)	(1.7)	(1.6)	(1.3)	...
129	1.00007	1.00048	1.00144	1.00933	
	(.0003)	(.002)	(.004)	(.03)	
	-.18	-.72	-.14	-2.1	...
	(.4)	(10)	(5)	(3)	...

The use of horizontal differences in locating the best entry in a

row is also a possibility, but it has not been tested.

K. Extrapolation in Initial Value Problems

We have spent a great deal of time on the use of the Richardson/Romberg technique as applied to definite integrals where the error series has only even integral powers of h. But it would be unfortunate not to discuss, at least briefly, other situations where the same techniques can be used with slight modification.

First, assume we are given an initial value problem as follows: Find y(b), where y(x) satisfies $y' = f(x,y)$ and $y(a) = c$. The simplest method to get an approximation of y(b) is to take small steps of size h along the x axis, using Euler's method to proceed from the estimate $\bar{y}(x)$ to a new estimate $\bar{y}(x+h)$:

$$\bar{y}(x + h) = \bar{y}(x) + h \cdot f(x,\bar{y}(x)). \tag{52}$$

As Bauer et al. (1963) pointed out, the error in the final approximation $\bar{y}(b)$ can be represented by an error series in powers of h. But this time the odd powers are present as well as the even powers. The only modification needed in the Richardson/Romberg technique is to use the power sequence 1,2,3,..., instead of 2,4,6,..., in formulas (13) - (17). For step-halving, the "4" in the previous recursion formula becomes a "2": thus,

$$T_m^k = \frac{2^m T_{m-1}^{k+1} - T_{m-1}^k}{2^m - 1}. \tag{53}$$

Bauer et al. gave a theorem asserting that if f(x,y) is continuous and does not change too rapidly as y changes (more precisely, satisfies a

Lipschitz condition in y), then the diagonals and columns of the table constructed with (53) will converge to the desired value, y(b).

They applied (88) to the problem

$$y' = y, \quad y(0) = 1, \quad y(.2) = ? \tag{54}$$

The correct value is, of course, $e^{.2}$, 1.221 402 758 .... Using step sizes of .2,.1,.05,...,.00625, and using the six approximations for y(.2) thus obtained, they obtained the results shown in Table XXIII. (We round each entry as its accuracy seems to make appropriate.)

TABLE XXIII

STEP-HALVING APPLIED TO AN INITIAL VALUE PROBLEM
WITH SOLUTION 1.221 402 758...

1.2000				
1.2100	1.220			
1.2155	1.2210	1.221350		
1.2184	1.2213	1.221395	1.221 401 7	
1.2199	1.221 376	1.221 401 7	1.221 402 68	1.221 402 747
1.2206	1.221 396	1.221 402 6	1.221 402 752	1.221 402 756

Of course the same idea can be used to improve the estimates given by more sophisticated methods for solving differential equation. Joyce (1971) mentions that, as early as 1912, Runge had suggested a Richardson type extrapolation for use in improving the estimates obtained by the Runge-Kutta fourth-order method.

L. Modifications for a Broader Class of Problems

To discuss other modifications of the Romberg technique, we now return to an example mentioned earlier: Romberg integration in the "usual" sense does not work well on the problem

$$\int_0^1 \frac{3}{2}\sqrt{x}\,dx = 1. \tag{55}$$

Presumably the reason is the discontinuity in the first derivative at 0, which prevents application of the Euler-Maclaurin error series theorem. At first glance, it is not clear whether there is any way to modify our procedure slightly so as to solve this type of problem more successfully. But in fact, there is a good amount of literature dealing with possible modifications of the basic procedure, so as to solve a larger class of problems. Problem (55), for example, belongs to a class of integration problems which Fox (1967) studied. He showed that the only thing which causes ordinary Romberg integration to fail on (55) is that the error series contains an $h^{3/2}$ term as well as all the even powers of h. Knowing this, it is easy to show that the extrapolation T(h) based on S(h) and S(2h) in the initial column should be

$$T(h) = \frac{2^{3/2}\,S(h) - S(2h)}{2^{3/2} - 1}\ . \tag{56}$$

From that point on, we can return to the ordinary power elimination, dealing with $h^2, h^4, \ldots$. On the other hand, Fox showed that the required procedural changes could be considerably greater on other problems. For example, the integration problem

$$I = \int_0^1 \sqrt{x(1-x)}\,dx \tag{57}$$

has derivative problems at both ends of the interval, and the resulting error series has no even powers of h at all. Instead, the powers are $h^{3/2}, h^{5/2}, h^{7/2}, \ldots$. But again, the extrapolation procedure can be easily modified to remove those powers instead of the even ones. As one can see in equations (13) - (17), the crucial thing is not to have any particular "kind" of powers. It is, rather, to know what the power sequence is. Another more ominous example is given by Fox. For the problem

$$I = \int_0^1 \sqrt{x} \ln x \, dx , \tag{58}$$

the error series begins with an $h^{3/2} \ln(h)$ term! Then come $h^2, h^4, \ldots$, as usual. Fox does show how to obtain a modification of the Romberg method which will conquer this problem, but it depends heavily on knowing the first error term is $h^{3/2} \ln(h)$. It should be clear why this example is unsettling: it makes clear that if we do not have the assumptions of the Euler-Maclaurin theorem, then the error may not be expressible as a series in powers of h, at all. And such an error representation is crucial to everything we have done.

Nevertheless, assuming that we are fortunate enough to be integrating a function which does have an associated error series containing only powers of h, we can use the approximations in each column of the table to help us guess which power of h should be removed next. At least for step-halving, this is not difficult. Returning to equation (29), with the power k instead of 2k, we obtain

$$k \doteq \frac{\log\left(\frac{V(4h) - V(2h)}{V(2h) - V(h)}\right)}{\log(2)} \tag{59}$$

The power to detect the correct first term for elimination was tested on several problems where the form of the first term was known. See Table XXIV.

TABLE XXIV

TEST ON ALGORITHM TO FIND THE LEADING POWER IN THE ERROR SERIES

N	$\int_0^1 \sqrt{x}\, dx$	$\int_0^1 \sqrt{x} \ln x\, dx$	$\int_0^{2\pi} \frac{.095}{1.81-1.8\cos x} dx$	$\int_0^1 \sqrt{x} \sin x\, dx$
4	1.38	1.12	1.01	1.84
8	1.42	1.18	1.03	1.88
16	1.45	1.23	1.12	1.92
32	1.46	1.26	1.46	1.94
64	1.47	1.29	2.48	1.96
128	1.482	1.32	4.87	1.97
256	1.487	1.33	9.73	1.98
512	1.491	1.35	19.46	1.99
Appropriate Limit	3/2	None, or 3/2	None	2

The $\sqrt{x} \ln x$ column of Table XXIV might be unexpected, but it is not difficult to show that the quantity inside $\log_{10}$ in the numerator of (59) does slowly approach $2^{3/2}$, even though the first term in the error series is $h^{3/2} \ln(h)$. And, in this case, things go unexpectedly well if we do go through the operations indicated by the 3/2: the $h^{3/2} \ln(h)$ term is replaced precisely by an extra $h^{3/2}$ term, for any h! But this is apparently just a coincidence and we would probably not deduce the 3/2 from the approximations obtained, anyway. As for the $\sqrt{x} \sin x$, it has no dis-

continuity in the first derivative, so we can begin with the first (h^2) term of the Euler-Maclaurin series. Fox (1967) did show that for any integral

$$I = \int_0^1 x^{1/2} g(x)\, dx, \tag{60}$$

where g(x) is analytic, the error series can be no more complicated than $ah^{3/2} + bh^2 + ch^{5/2} + dh^3 + eh^{7/2} + fh^4 + \ldots$.

M. Conclusions

This chapter has not come close to covering the vast amount of literature on Richardson-like procedures; we have touched only on the more accessible portions. The reader who wishes to pursue the matter further should refer to Joyce (1971) for a good beginning. When any sequence of numbers is approaching its limit slowly but monotonically, Richardson extrapolation is one of the first methods that should be tried--preferably in an interactive mode, a column at a time; so that the user can estimate the next power to be removed, based on (59). If, on the other hand, one wishes to do an integration of a known function which satisfies the Euler-Maclaurin hypotheses, doing the table by rows would seem to be a better idea. Interaction would not be so necessary. If the assumption of a series in h is valid, and if the powers can be established, no method short of Gaussian integration is likely to outperform the methods of this chapter.

CHAPTER IV

AITKEN EXTRAPOLATION

A. Introduction

The subject of this chapter is perhaps the most powerful method which is elementary to motivate. Todd (1962) sees an invention of the method in the work of Kummer (1837); however, the present author is unconvinced on that point. (The impressive formulas of Kummer would reduce to the later method only by a drastic simplification: he works quite hard to solve for two variable coefficients which would be 1's in the modern method.) It seems better to attribute the original invention to the author whose name the method now bears: A. C. Aitken (1926). The algorithm was originally invented to help accelerate the convergence of a standard method for solving polynomial equations; but it can be used in many other contexts. (Aitken's "Δ^2" method has more flexibility than the previous methods discussed: it can reasonably be tried as an accelerator for either an oscillating sequence or a monotone sequence.

B. Motivation and Implementation

One motivation for the method rests on a certain expression for the "remainder" after n partial sums of a geometric series. Assume for some r (any r is permitted except $r = 1$)

$$A_n = a + ar + ar^2 + \ldots + ar^n, \qquad n \geq 0, \tag{1}$$

then it is easy to show, for any $r \neq 1$, that

$$A_n = A + cr^n, \tag{2}$$

where

$$A = \frac{a}{1-r} \quad \text{and} \quad c = -ar/(1-r). \tag{3}$$

In other words, there are constants A and r such that the distance from A_n to A is always multiplied by r when we increase n by one. It is not difficult to show that there can be only one "A" having that property. If $|r| < 1$, then A is the limit of the A_n and the sum of the series. If $r = -1$ or $|r| > 1$, then A is still unique but is called the anti-limit of the A_n. "A" then has a natural meaning in terms of analytic continuation: if we let $z \in \mathbb{C}$ and take

$$A(z) = a/(1-z), \tag{4}$$

then the function A(z) gives rise to a geometric series with partial sums as in (1) (z replaces r). That series converges to A(z) only if $|z| < 1$. But if we can use the partial sums $A_n(z)$, as in (1), to find A(z) even when $|z| > 1$, we will have succeeded in computing the analytic continuation of the power series at z.

Can this computation of A be done irrespective of convergence? Yes, and we do not need to know the values of c, r, or n; we only need three consecutive sums. Assume

$$\begin{aligned} A_n &= A + cr^n \\ A_{n+1} &= A + cr^{n+1} \\ A_{n+2} &= A + cr^{n+2}. \end{aligned} \tag{5}$$

It is simple to show

$$\frac{1}{r} = \frac{A_n - A}{A_{n+1} - A} = \frac{A_{n+1} - A}{A_{n+2} - A} , \tag{6}$$

which then gives various forms

$$A = \frac{A_n A_{n+2} - A_{n+1}^2}{(A_{n+2} - A_{n+1}) - (A_{n+1} - A_n)} \tag{7}$$

$$= A_n - \frac{(\Delta A_n)^2}{(\Delta^2 A_n)} \tag{8}$$

$$= A_{n+2} - \frac{(\Delta A_{n+1})^2}{(\Delta^2 A_n)} . \tag{9}$$

It is also obvious that we could have begun with (6), without ever mentioning geometric series; i.e., we need only assume that the ratio of consecutive errors is constant, independent of n. This amounts to assuming "perfect" linear convergence; that is, of course, characteristic of convergent geometric series.

The Aitken method consists in using one of the right sides of (7), (8), or (9) to approximate the limit of any sequence. Each trio in the original sequence gives an approximation. As we move down the original sequence, we thus generate a sequence of approximations. If we then use this new sequence for the basis of another application of the method, we are doing "repeated" Aitken extrapolation. It should not be surprising that Aitken extrapolation sums any geometric series perfectly, given only three sums. We planned it that way. E.g., if we use A_1, A_2, and A_3 from

the partial sums of $1+2+4+8+\ldots$, we obtain -1, the appropriate value for $1/(1-x)$ when $x=2$.

Of course, we need no help in summing geometric series. But just as Euler's method works well when the ratio of consecutive errors approaches -1, Aitken's method would seem to have a good chance of success when the ratio of consecutive errors approaches any constant $r \neq 0$. We thus have considerably more latitude than in the Euler method. But it can be shown that the denominators in (7), (8), and (9) go to zero if $r = 1$. This would suggest that if the sum is finite but the ratio of consecutive errors goes to 1, the Δ^2 method is not going to work well.

C. Successful Applications

Certainly the Aitken method should work well on the pi series, since the Euler method did. And that is in fact the case. The best extrapolation from the first seven entries in the initial column is 3.14156, compared with Euler's best approximation of 3.140 through that point. Also, the best answers are always on the right end in the Aitken table, which is what we would hope.

Now let us move on to other problems, for which the Euler method might not have been well-suited. For example, let us look at the kind of problem the Aitken method was originally designed for: aiding in accelerating Bernoulli's method for finding the largest root of a polynomial equation. Suppose we want to solve the equation

$$x^4 + 3x^3 - 32x^2 - 12x + 112 = 0. \tag{10}$$

The roots are $-\sqrt{2}$, $+\sqrt{2}$, 4, and -7; and Bernoulli's method gives a sequence of numbers which will approach the -7. The algorithm goes thus:

begin with

$$x_0 = x_1 = x_2 = 0 \quad \text{and} \quad x_3 = 1. \tag{11}$$

Then compute x_4, x_5, x_6,..., from

$$x_n = -3x_{n-1} + 32x_{n-2} + 12x_{n-3} - 112x_{n-4}. \tag{12}$$

The ratio x_{n+1}/x_n will approach the -7, but not very rapidly. It can be shown that the ratio of consecutive errors does approach a constant for a polynomial of this "type". Henrici (1964, pp. 146ff) gives more details. Thus Aitken's method should help. It does. See Table XXV, where we use the method repeatedly. Each entry in the table is the extrapolation based on the entry to its left and the two entries just above that one. The leftmost column is the sequence generated via the x_{n+1}/x_n ratios.

TABLE XXV

AITKEN'S METHOD USED TO ACCELERATE THE BERNOULLI RATIO SEQUENCE

-3.00000			
-13.66667			
-5.04878	-8.8999438		
-8.62319	-7.57528		
-6.24706	-7.19588	-7.04360	
-7.48785	-7.06219	-6.98946	
-6.73939	-7.02100	-7.00265	-7.00007

The reader might wish to verify that the Euler method does not do as well on this problem as Aitken's method does; the ratio of one error to the next error approaches -7/4, not -1.

The next two examples will show the Δ^2 method routinely mastering problems which the Euler method found either very difficult or not quite possible. For example, recall the $\ln(1+x)$ series. As Figure 2 in Chapter II indicated, the Euler method would require four tables to be computed before convergence to ln(17) would be achieved. See Table XXVI below, for the results obtained in the (single) Aitken table.

TABLE XXVI

AITKEN CONVERGENCE ON THE DIVERGENT SERIES
FOR LN(17), 2.83321...

Row	Column One Entry	Last Column	Last Column Entry
10	.713437E+10	5	2.85438
11	-.102817E+12	6	2.83031
12	.149647E+13	6	2.83654
13	-.219598E+14	7	2.83290
14	.324471E+15	7	2.83363
15	-.482250E+16	8	2.83319

Another problem we saw where Euler's method was never able to produce convergence (although each new table had a top diagonal better than the previous table's) was related to the asymptotic sum of Wallis' series,

$$.59634736... \sim 0! - 1! + 2! - 3! + 4! - .... \qquad (13)$$

Aitken's method produces rather spectacular convergence along the diagonals, though the columns still diverge. See Table XXVII; correct digits are underlined.

TABLE XXVII

THE AITKEN RESULTS ON WALLIS' SERIES

Row	Column One Entry	Last Column	Last Column Entry
10	.3590E+5	5	.<u>5965</u>
11	-.3270E+6	6	.<u>596337</u>
12	.3302E+7	6	.<u>596363</u>
13	-.3661E+8	7	.<u>5963469</u>
14	.4424E+9	7	.<u>59634849</u>
15	-.5785E+10	8	.<u>59634742</u>

D. Unsuccessful Applications

We hope the reader is now convinced that Aitken had a very good idea. But in order to give a realistic view of matters, we shall now show that our new-found power on some problems is accompanied by the possibility of Aitken's method doing strange things on some problems--e.g., not converging even when the initial column does, or converging to the wrong answer! Wimp (1981, p. x) explains that this is generally the price of power in acceleration methods: if you insist on a method that will always give only the correct answer, the method will generally never be spectacular because it is trying to be too versatile. Euler's method

would never do the bad things we are about to demonstrate; but it also was unable to compete with Aitken's method on the problems of the previous section.

The first example we shall discuss was given by Lubkin (1952). The series is

$$1 + 1/2 - 1/3 - 1/4 + 1/5 + 1/6 - \ldots =$$

$$\pi/4 + (\ln 2)/2 = 1.13197.... \qquad (14)$$

See Table XXVIII below.

TABLE XXVIII

FAILURE OF AITKEN'S METHOD ON LUBKIN'S SERIES,
SUM = 1.13147...

0			
1			
1.5	2.0000		
1.1667	1.3000		
.9167	.1667	3.1308	
1.1167	1.0278	.6560	
1.2833	2.1167	-3.0888	7.9530
1.1405	1.2064	1.6209	-1.0027
1.0155	.1405	7.4390	-23.098

A continuation of the table would show that the second column is in fact approaching three different limit points separated by gaps of 1:

.13197..., 1.13197..., and 2.13197.... The method is unable to analyze this behavior and the following columns are worthless. The source of the difficulty is, of course, in the error pattern in column one: the signs of the errors are -, -, +, +, -, -, +, +,.... This makes the ratio of consecutive errors alternately positive and negative. Such a sequence of ratios is not going to converge in the manner assumed by the Δ^2 method. The method therefore fails: the "accelerated" columns do not converge to the proper answer though the initial column did.

The next example was invented by Shanks (1955). In a way this example is worse than Lubkin's because in Shanks' example the method converges nicely--to the wrong answer. The function involved can be written as the sum of two geometric series:

$$f(z) = \frac{2}{(1-z)(2-z)} = 2\left(\frac{1}{1-z}\right) - \frac{1}{1-\left(\frac{z}{2}\right)} =$$

$$1 + \frac{3}{2}z + \frac{7}{4}z^2 + \frac{15}{8}z^3 + \ldots. \qquad (15)$$

If we let $z = 4$, we will get a rapidly divergent series. But this is no cause for alarm: if Aitken's method can "sum" one divergent geometric series easily, we would perhaps expect the method to find the "sum" of (15) without much difficulty. Unfortunately, this is not the case. See Table XXIX. The value of f(4) is 1/3. The later columns converge to 7/27 (.25926...) instead.

Shanks does show that this mistaken limit is going to occur only at $x = 4$, which is somewhat encouraging. However, though this particular series was his only numerical example, his analysis did show that there is going to be such an x for the sum of virtually any two geometric series. This is not encouraging; but from a probabilistic view, one might

think that, since there is only one x that causes the mistaken limit, we can generally expect the Aitken method to do well for most x's. On the other hand, the existence of even one such x might be an indication that something is wrong, in general. Shanks mentions that the convergence is nonuniform near the crucial x; but he gives no examples. It seems reasonable to try a number of x's on both sides of 4, to see what happens. We will compare relative errors to get a uniform standard for the different x's. The basic approach was to record the smallest relative error attained by the Δ^2 method in the tenth row of the Aitken table. See Table XXX. The relative errors grow toward -.222 as we approach 4 and are really never very small past 4.0.

TABLE XXIX

CONVERGENCE OF AITKEN'S METHOD TO THE WRONG ANSWER ON SHANKS' EXAMPLE

0				
1				
7	-.2000			
35	-.6364			
155	-1.5217	.2241		
651	-3.2979	.2437		
2667	-6.8526	.2519	.2579	
10,795	-13.9634	.2557	.2589	
43,435	-28.1854	.2575	.25920	.25932

TABLE XXX

AITKEN CONVERGENCE ON ROW TEN FOR SHANKS' EXAMPLE (FOUR EXTRAPOLATIONS ALLOWED)

x	Best Relative Error
.6	.98E - 9
1.4	.38E - 4
2.2	.12E - 2
3.0	-.80E - 2
3.8	-.14E + 0
4.6	-.17E + 0
5.4	.12E - 1
6.2	-.36E - 1
7.0	-.63E - 1
7.8	-.93E - 1
8.6	-.12E + 0
9.4	-.14E + 0
10.2	-.14E + 0
11.0	-.13E + 0

The larger x's are, of course, associated with more rapidly divergent partial sums. Thus, some of the method's difficulties are from severe round-off error in the later rows, as opposed to intrinsic weakness in the method itself. Nevertheless, the poor performance can not be attributed mostly to round-off errors: a method to be introduced in the next chapter takes exactly the same partial sums and slightly more complicated calculations based on them and does much better than the Δ^2 method. For large x's, the Δ^2 method finds this simple problem more difficult than summing Wallis' series. See Table XXXI, based on the sums

at x = 10. All partial sums can easily be verified to be correct to sixteen decimal places. The correct sum is .027777777.... The other method mentioned above produces a table where all the entries beyond column two and above row eleven are at least as good as .027777777.

TABLE XXXI

AITKEN'S METHOD ATTEMPT TO SUM SHANKS' EXAMPLE
AT x = 10 TO .277...E - 1

0				
1				
.16E + 2	-.71E - 1			
.19E + 3	.41E + 0			
.21E + 4	-.20E + 1	.17E - 1		
.21E + 5	-.99E + 1	.64E - 2		
.22E + 6	-.49E + 2	-.19E - 1	.23E - 1	
.22E + 7	-.24E + 3	-.83E - 1	.24E - 1	
.22E + 8	-.12E + 4	-.24E + 0	.24E - 1	.24E - 1
.22E + 9	-.60E + 4	-.64E + 0	.23E - 1	.24E - 1

The monotonicity of the partial sums is not the major difficulty: at x = -10, where the sums are oscillating, Aitken's method is not able to obtain more than three correct digits in the first ten rows. It seems reasonable to guess that the Δ^2 method is often not going to work well on the sum of (even) two geometric series, even when it does eventually approach the correct answer.

We saw in Lubkin's example how a failure of the consecutive error

ratios to converge can destroy the Δ^2 method's effectiveness completely. But even when the ratios converge, success may be postponed "indefinitely" if the convergence is too slow. For an example of this, we used the "pi + 1" series which destroyed the effectiveness of the Euler algorithm (Chapter II, equation 12). The Aitken method will succeed perfectly if and only if the ratio of consecutive error ratios is 1. For the pi series, the ratio of ratios from S_9 through S_{14} has values 1.016, 1.012, 1.010, 1.008, 1.007, 1.006. The result is good extrapolation: an absolute error of -.12E - 10 after five extrapolations, in the S_{14} row. For the "pi + 1" series, the corresponding ratios are .886, 1.092, .968, 1.032, .994, and 1.013. The monotone component in the series has retarded the convergence. The result is fairly poor extrapolation: an absolute error of .92E - 3 after five extrapolations, in the S_{14} row.

The next example shows that when the ratio of consecutive error ratios is not 1, how close it needs to be to 1 for good success of the Δ^2 method may depend on what the ratio of errors approaches. Regroup the pi series as follows:

$$\pi = 4(1 - 1/3) + 4(1/5 - 1/7) + 4(1/9 - 1/11) + \ldots. \qquad (16)$$

The ratios of ratios from S_9 through S_{14} are as good as for the original pi series: 1.016, 1.012, 1.010, 1.008, 1.007, and 1.006. But now those are not good enough: the limit of consecutive errors is now +1 instead of -1. We have turned our friendly pi series, for which every acceleration method seems to work, into a logarithmically convergent series. Those are known to cause difficulty for most acceleration methods; in particular, we saw earlier that troubles were to be expected for Aitken's method when the ratio of errors approaches 1. The Aitken method in fact

does do very poorly on the regrouped pi series. See Table XXXII.

TABLE XXXII

Δ^2 ATTEMPT AT SUMMATION OF THE RE-GROUPED "PI" SERIES

0			
2.67			
2.90	2.92		
2.98	3.02		
3.02	3.06	3.08	
3.04	3.08	3.10	
3.06	2.09	3.11	3.12

Construction of the corresponding rows for the original pi series would give a 3.14156... in place of the 3.12 above, in spite of using only half as many terms in forming the partial sums.

We have seen how various misbehaviors of the error ratios can destroy the effectiveness of the Δ^2 method. But there is one other situation, actually extremely desirable, which can make the Δ^2 method quite useless. Suppose that, in the original sequence, the ratio of one error to the error preceding it goes to zero. This is no longer linear convergence, and the Aitken method will customarily give "accelerations" which in fact are not as good as the best number used in the computations. Suppose, for example, that we have the sequence 1/2, 1/4, 1/16, 1/256,.... Instead of

$$S_n - S \doteq K(S_{n-1} - S), \tag{17}$$

with K non-zero and independent of n, we have

$$\begin{aligned} &S_n - S = (S_{n-1} - S)(S_{n-1} - S), \\ &\lim_{n\to\infty} \left(\frac{S_n - S}{S_{n-1} - S}\right) = 0. \end{aligned} \tag{18}$$

What will the Aitken method conclude from the first three numbers, for example? It is not difficult to use system (5) to show that for a geometric series we always have

$$\frac{A_{n+2} - A_{n+1}}{A_{n+1} - A_n} = r. \tag{19}$$

From the differences -1/4 and -3/16, the assumption of a geometric series forces the conclusion r = 3/4. The Aitken estimate of the limit will therefore assume that for all n,

$$\frac{S_{n+2} - S_{n+1}}{S_{n+1} - S_n} = \frac{3}{4}. \tag{20}$$

In fact, however, the skips beyond the 1/16 are going to zero much faster than in (20); in fact,

$$\lim_{n\to\infty} \frac{S_{n+2} - S_{n+1}}{S_{n+1} - S_n} = \lim_{n\to\infty} \left(\frac{S_{n+2} - S}{S_{n+1} - S}\right) = 0. \tag{21}$$

Consequently, the Aitken method drastically underestimates how rapidly the skips are diminishing and calculates the S_n will approach -1/2 instead of the true limit, 0. Note that the -1/2 is not as good an approximation as two of the three numbers used in calculating it. This kind of

occurrence extends through the Aitken table. See Table XXXIII, where the method consistently overshoots the limit and is producing answers not as good as the best number used in calculating them.

TABLE XXXIII

AITKEN'S METHOD ON A SEQUENCE CONVERGING SUPER-LINEARLY TO ZERO

.500E + 0		
.250E + 0		
.625E - 1	-.500E + 0	
.391E - 2	-.227E - 1	
.153E - 4	-.262E - 3	+.848E - 3
.233E - 9	-.598E - 7	+.302E - 5

However, the failure of the method is not very critical in such an instance: the original sequence is converging quite rapidly and we were really being a bit greedy to try to accelerate it more.

E. Prediction and Measurement of Success

By now we have seen the Δ^2 method succeed in a quite spectacular way, and we have also seen it produce rather dismal failures. It seems appropriate to address briefly the questions of how to tell whether the method is likely to succeed on a given sequence, and how to estimate the quality of the extrapolations when we do not know the correct limit.

The answer to the first question is that for a geometric series, we

obtain from (19) that

$$\frac{A_{n+3} - A_{n+2}}{A_{n+2} - A_{n+1}} = r = \frac{A_{n+2} - A_{n+1}}{A_{n+1} - A_n}. \tag{22}$$

Therefore, if we have a sequence of numbers $S_1, S_2, \ldots$, we can conclude that it is a good candidate for Aitken extrapolation if, for large n, and some positive c,

$$\frac{(S_{n+3} - S_{n+2})(S_{n+1} - S_n)}{(S_{n+2} - S_{n+1})^2} \doteq 1, \tag{23}$$

with

$$\left| \frac{S_{n+2} - S_{n+1}}{S_{n+1} - S_n} - 1 \right| > c > 0. \tag{24}$$

For example, on Wallis' series,

$$S_0 = 0!, \quad S_1 = 0! - 1!, \quad S_2 = 0! - 1! + 2!, \ldots. \tag{25}$$

It is not hard to show that the quotients corresponding to the left sides of (23) and (24) are

$$\frac{n+3}{n+2} \doteq 1 \quad \text{and} \quad -(n+2) << +1. \tag{26}$$

For the pi sequence, let

$$S_0 = 0, \quad S_1 = 4, \quad S_2 = 4(1 - 1/3), \quad S_3 = 4(1 - 1/3 + 1/5), \ldots. \tag{27}$$

The ratios obtained are

$$\frac{4n^2 + 12n + 9}{4n^2 + 12n + 5} \doteq 1 \quad \text{and} \quad -\frac{2n+1}{2n+3} \doteq -1 \neq +1. \tag{28}$$

The first eight ratios corresponding to (23) are thus 1.80, 1.19, 1.09, 1.05, 1.03, 1.02, 1.02, 1.01. On the other hand, the corresponding ratios for the troublesome "pi + 1" sequence are 3.55, .64, 1.70, .78, 1.24, .92, 1.08, .98. On this basis alone, we could predict that the Δ^2 method is not going to work well. (But if we go deeply enough into the sums, the monotone component does die out and Aitken's method would eventually succeed as well as on the pi series.)

The second question, concerning how to estimate quality of the extrapolations when the limit is unknown, is usually answered in the following way: assume for now that the limit is ℓ, $\ell \neq 0$. Then the relative error of any entry x in the Aitken table is given naturally by

$$r = \frac{x - \ell}{\ell} = \frac{x}{\ell} - 1. \tag{29}$$

(We vary here from the usual convention, $(\ell - x)/\ell$, because it seems backwards to the present author.) For an estimate of r when ℓ is unknown, use the entry y to the right of x in the table as an approximation for ℓ. I.e., the relative error in x is approximated by

$$r' = \frac{x - y}{y} = \frac{x}{y} - 1 . \tag{30}$$

We thus estimate the relative accuracy of the entries by their relative variations. This may seem to be a desperate measure, but it generally works quite well: normally, when the table entries are close to each other, they are also close to the limit.

Some examples of the relative error approximation are given in Table XXXIV. They are based on the approximations calculated for the entries in row ten, for various tables.

TABLE XXXIV

ACCURACY OF THE RELATIVE ERROR ESTIMATES IN THE AITKEN TABLES, ROW TEN

pi:	estimates	.351E - 1	.149E - 3	.306E - 5	.165E - 6
	true	.353E - 1	.153E - 3	.324E - 5	.184E - 6
Wallis:	estimates	.597E + 3	.686E + 2	.436E + 0	.611E - 2
	true	.602E + 5	.996E + 2	.446E + 0	.638E - 2
pi with ( ):	estimates	-.782E - 2	-.430E - 2	-.240E - 2	.142E - 2
	true	-.177E - 1	-.993E - 2	-.566E - 2	.327E - 2
Shanks:	estimates	-.367E + 5	.949E + 4	-.285E + 2	-.342E - 1
(x=10)	true	.798E + 10	-.217E + 6	-.239E + 2	-.167E + 0

The worst thing that can happen is, of course, for the method to stall far away from the limit. For example, the last error estimate on row fifteen of the Shanks example (with x=10) is -.257 E - 5, while the true relative error is +.105E - 1. But this sort of thing is usually temporary; for example, at the end of the sixteenth row of the same table, the relative error estimate (-.429E - 1) is quite close to the actual values (-.535E - 1). If one is setting up a termination criterion for using the Aitken method in an automatic mode, it would be a very good idea to base the termination criterion on several error estimates, not just one. In case the limit is zero, one branch of the program should allow for concluding a zero limit on the basis of absolute value rather than relative error.

F. Theorems Concerning the Method

We have by now, hopefully, a fairly good idea of when the Δ^2 method

is likely to work and when it is likely to fail. Our study has been built on examples rather than theorems. But there are some relevant theorems which should be mentioned before we conclude our discussion. Wimp (1981) has gathered several in his book; we now discuss them informally, along with a few others given elsewhere.

First, the reader may have noticed that we used, in (24), the natural assumption that for large n,

$$\frac{S_{n+2} - S}{S_{n+1} - S} \doteq \frac{S_{n+2} - S_{n+1}}{S_{n+1} - S_n}. \tag{31}$$

Wimp (pp. 6-7) proves that if the limit of error ratios converges to some number other than -1, 0, or +1, then (31) is always justified. The proof if not as easy as one might think, but the result seems quite plausible.

Most of the results he gives are in his Chapter 7 (pp. 149-151). For example, if $(S_{n+2} - S_{n+1})/(S_{n+1} - S_n)$ is bounded away from "1" and the S_n converge to S, then the (single) application of extrapolation will produce a new column which converges to S, also. (Lubkin's example obviously must not have satisfied the first hypothesis.) Lubkin (1952) did show that if any two consecutive columns of the Aitken table both converge, then they converge to the same limit. (But recall Shanks' example with $x = 4$: the first column converged, the second diverged, and the rest converged to <u>another</u> limit.) Tucker (1967) proved that, in a sense, things can not be too much worse than in Lubkin's example: if a column converges to S, then the next column at least has a subsequence converging to S. In Lubkin's example, we had three limit points in the second column. But one of them was the true answer, as Tucker's result would predict. Note that none of these theorems are concerned with accelera-

tion, only convergence.

One of Lubkin's results, not mentioned by Wimp, essentially says that if the ratio of consecutive differences goes to -1 (excluded by the first theorem, concerning (31)), and if our formula (23) holds, then convergence will be preserved and accelerated by the next column (Theorem 5, p. 231). Lubkin gives several theorems on when acceleration will occur, but the hypotheses seem to be rather complex except in a few of the theorems. We mention only two for that reason. His Theorem 11 says that if the S_n converge and the ratio $(S_n - S_{n-1})/(S_{n+1} - S_n)$ can be written as a power series in 1/n, with leading constant c_0, then the next column will accelerate convergence if $c_0 \neq 1$; but if $c_0 = 1$, then the convergence will not be accelerated. (For the regrouped pi series, c_0 was obviously 1.) His Theorem 17 concerns the case where the S_n diverge. In this case, if the ratio above can be written as mentioned above, with leading constant -1, then using an appropriate number of columns will eventually give convergence to the right answer, which of course involves an analytic continuation. We leave it to the reader to pursue the details in Lubkin's paper.

Of course the one fact which is the most important is that if any sequence converges linearly, then the Aitken method will accelerate the convergence. Johnson and Riess (1977) give a formal theorem and proof; but notice that the statement given is misleading for the case when the limit $(S_{n+1} - S)/(S_n - S)$ is zero. (The standard of comparison for acceleration should be the error in the last of the three numbers used, not the first.)

G. Conclusions

We have seen that the Aitken method, while easy to motivate, is considerably more powerful than Euler's method. The Δ^2 method can sum some divergent series which are difficult or impossible for Euler's method; and it can accelerate the convergence of any convergent series which has consecutive error ratios approaching any non-zero number except +1; Euler's method needed a -1 limit. The Δ^2 method can be used on both monotone and oscillating sequences, while the previous methods were restricted to oscillating sequences (Euler) or monotone sequences (Richardson/Romberg).

In their second survey article on extrapolation methods, based on extensive testing of many problems, Smith and Ford (1982) included Aitken's method in their tests. The method did not "win" any of the categories, but was competitive in almost all categories. The main lapses arose with logarithmic series (like our regrouped pi series) and series where the error signs did not fit into the acceptable patterns (like Lubkin's example). We have also seen a case where the slow convergence of the error ratios delayed the effectiveness of the Δ^2 method considerably (the "pi + 1" series). Finally, there was Shanks' simple example which was not handled well at all by the Δ^2 method. The method to be introduced in the next chapter will, in some of these areas, make considerable improvements on the Δ^2 method. But on some problems, Δ^2 will actually be slightly superior to the more "sophisticated" method (e.g., on Wallis' series and the ln(17) series).

CHAPTER V

THE EPSILON ALGORITHM

A. Introduction and Historical Overview

The algorithm we shall study in this chapter has a history somewhat similar to that of Richardson extrapolation: even as early as Jacobi (1846) and then Froebinius (1881), explicit expressions had been given for every approximation in what is now the "epsilon" table. The only problem was that the expressions involved determinants; and if anyone had considered setting up a table as is used now, he would have been quickly discouraged: the matrices would have become larger and larger, and the determinant calculations increasingly inconvenient. Also, each entry would have been necessarily calculated from the original column, somewhat in the same way the earlier writers removed several powers of h in Richardson extrapolation. The computations needed made the expressions, while useful from a theoretical standpoint, quite uninviting for actual implementation. The formulas thus lay unused and forgotten until Schmidt (1941) and Shanks (1955), without any knowledge of the earlier work or each other, rediscovered them and implemented them in spite of the determinants. It seems unlikely that their example would have ever been followed by many, even in the coming Computer Age. However, Wynn (1956) made the next crucial step: he discovered a way to find the entries of the table very simply, with no determinants required. Since this advance,

the method has become increasingly popular, with good reason. Smith and Ford (1982), after extensive testing on most of the main acceleration methods, found that the combination of "epsilon" and Levin's "u" (1973) was unsurpassed on their test problems. In the next section, we will begin by showing one motivation for the kind of expressions obtained by the writers before Wynn. (There is probably no way to motivate Wynn's ingenious determinant manipulations in deriving the new formulas. Wimp (1981, pp. 244-247) gives some details Wynn (1956) omitted.)

B. Motivation of Shanks' Transforms

Let us return briefly to the geometric series with partial sums

$$A_n = c + cr + \ldots + cr^n, \qquad n \geq 0, \tag{1}$$

and limit (or anti-limit)

$$A = c/(1 - r) . \tag{2}$$

Then, in a step reminiscent of Richardson extrapolation, and Aitken extrapolation, write

$$\begin{aligned} A_n &= A - \frac{cr^{n+1}}{1 - r} \\ A_{n+1} &= A - \frac{cr^{n+2}}{1 - r} . \end{aligned} \tag{3}$$

So we see that in a geometric series, the correct limit is always a weighted average of the last two partial sums, with the weights not depending on n. I.e., if we know we have a geometric series, then even without knowing c, n, or r, we can be sure there exists constants a and b such that $a + b = 1$ and, for all n,

$$aA_n + bA_{n+1} = A \tag{5}$$

The next question would be, how do we find A? Use the fact that a and b work for any n, to set up the system,

$$\begin{bmatrix} 1 & 1 & 0 \\ A_n & A_{n+1} & -1 \\ A_{n+1} & A_{n+2} & -1 \end{bmatrix} \begin{bmatrix} a \\ b \\ A \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \tag{6}$$

Cramer's Rule quickly gives

$$A = \begin{vmatrix} 1 & 1 & 1 \\ A_n & A_{n+1} & 0 \\ A_{n+1} & A_{n+2} & 0 \end{vmatrix} \Big/ \begin{vmatrix} 1 & 1 & 0 \\ A_n & A_{n+1} & -1 \\ A_{n+1} & A_{n+2} & -1 \end{vmatrix} \tag{7}$$

$$= \frac{A_n A_{n+2} - A_{n+1}^2}{(A_{n+2} - A_{n+1}) - (A_{n+1} - A_n)},$$

which is, of course, Aitken's formula. Note that although we could have also used (6) to find the weights in terms of the A_i, that was not necessary in calculating A. We only needed (4) to guarantee (6) could be set up independently of n. Now let us see if we can extend this procedure.

Suppose that our series is the sum of two geometric series: let A be the limit (or anti-limit), as $n \longrightarrow \infty$, of

$$A_n = c + d + cr + ds + \ldots + cr^n + ds^n, \tag{8}$$

and let

$$x_n = -\frac{cr^{n+1}}{1 - r} \quad \text{and} \quad y_n = -\frac{ds^{n+1}}{1 - s}. \tag{9}$$

Then, as in (3),

$$A_n = A + x_n + y_n$$

$$A_{n+1} = A + rx_n + sy_n \tag{10}$$

$$A_{n+2} = A + r^2x_n + s^2y_n .$$

As before, take a, b, and e as coefficients in a linear combination so as to end with only "A" left on the right side. This could be accomplished by solving for a, b, e, in

$$a + b + e = 1$$

$$a + br + er^2 = 0 \tag{11}$$

$$a + bs + es^2 = 0.$$

We conclude that the desired coefficients do exist, and that the "A" can be written as a weighted average of any three (not two, as before) consecutive partial sums, with the weights independent of n. But solving (11) requires that r and s be known; can we avoid this requirement? And we would prefer not to have to get the formulas for a, b, and e. We really want A, and in terms of the A_i only. Our goals can be attained by solving for A below:

$$\begin{aligned} a \quad &+ b \quad + e \quad = 1 \\ aA_n \quad &+ bA_{n+1} + eA_{n+2} - A = 0 \\ aA_{n+1} \quad &+ bA_{n+2} + eA_{n+3} - A = 0 \\ aA_{n+2} \quad &+ bA_{n+3} + eA_{n+4} - A = 0. \end{aligned} \tag{12}$$

Once again, Cramer's rule could be used to solve for A; this time the matrices would be 4×4 instead of 3×3, with twenty-four terms in the determinant formulas instead of six. We conclude that if we have a sum of two geometric series, then the sum is the weighted average of any three consecutive partial sums, with the required weights not depending where we are in the summing process. Note that the determinant formula is going to require A_n through A_{n+4}.

Somewhat similar arguments (though our presentation is more in line with the later one of Levin (1973)), led Shanks to this conclusion: if we have a series which is obtained by adding k geometric series together, then the sum A is always the weighted average of any A_n, A_{n+1},..., A_{n+k}, with the weights independent of n. Further, the desired limit (or anti-limit) can be calculated using only the A_i and an appropriate pair of (k+1)×(k+1) determinants, involving A_n through A_{n+2k}.

Shanks then defined what he called the e_1, e_2, e_3,..., transformations. The progression is like this: $e_1(A_n)$ is the implied limit, assuming that A_n, A_{n+1}, and A_{n+2} are partial sums in a geometric series. If these assumptions are correct, then $A = e_1(A_1) = e_1(A_2) = \ldots$; otherwise, we get a new non-constant sequence. Thus, e_1 is identical with Aitken's Δ^2 process. $e_2(A_n)$ is the implied limit, assuming that A_n, A_{n+1}, A_{n+2}, A_{n+3}, A_{n+4} are partial sums in a "double" geometric series. If these assumptions are correct, then $A = e_2(A_1) = e_2(A_2) = \ldots$; otherwise, we get a new non-constant sequence. From the poor attempt we saw Aitken's method make on a series of this type, e_2 must not be the same as applying Aitken's method twice. The difference will be discussed later. The process can be continued indefinitely: e_3 assumes a "triple" geometric series, etc. Some of the series can be null, of course: if we have a

single geometric series, then e_2, e_3,..., will sum it as well as e_1 does.

The dependencies on the A_i can be indicated in Figure 7, each $e_k(A_n)$ depends on the A_i's between the two diagonals through $e_k(A_n)$.

A_0			
A_1	$e_1(A_0)$		
A_2	$e_1(A_1)$	$e_2(A_0)$	
A_3	$e_1(A_2)$	$e_2(A_1)$	$e_3(A_0)$
A_4	$e_1(A_3)$	$e_2(A_2)$	
A_5	$e_1(A_4)$		
A_6			

Figure 7. Configuration for the e_i Transforms

C. Wynn's Epsilon Table

Now Wynn (1956) entered the picture. He showed that the entries in Figure 7 did not all have to be calculated directly from the A_i, using determinants. The entries could actually be calculated from the previous column, if some auxiliary numbers were saved along the way. Wynn used epsilons with both subscripts (column number) and superscripts (diagonal number) to denote the entries of Shanks' table together with his own auxiliary numbers. See Figure 8 for the arrangement used in the "epsilon"

algorithm.

$$
\begin{array}{lllll}
 & A_0 = \varepsilon_0^0 & & & \\
\varepsilon_{-1}^1 = 0 & & \varepsilon_1^0 & & \\
 & A_1 = \varepsilon_0^1 & \varepsilon_2^0 = e_1(A_0) & & \\
\varepsilon_{-1}^2 = 0 & & \varepsilon_1^1 & \varepsilon_3^0 & \\
 & A_2 = \varepsilon_0^2 & \varepsilon_2^1 = e_1(A_1) & & \varepsilon_4^0 = e_2(A_0) \\
\varepsilon_{-1}^3 = 0 & & \varepsilon_1^2 & \varepsilon_3^1 & \\
 & A_3 = \varepsilon_0^3 & \varepsilon_2^2 = e_1(A_2) & & \\
\varepsilon_{-1}^4 = 0 & & \varepsilon_1^3 & & \\
 & A_4 = \varepsilon_0^4 & & &
\end{array}
$$

Figure 8. Shanks' Arrangement, as Supplemented by Wynn

The numbers in the odd subscript ε columns customarily diverge to $\pm\infty$; but the even subscript columns will (generally) converge to the limit of the A_n--more quickly than the A_n, if the assumptions made for that column are essentially correct.

To compute any entry in the table, consider it as the right side of a lozenge with four corners. That right side entry can be calculated by the formula

$$\text{Right} = \text{Left} + \frac{1}{\text{Bottom} - \text{Top}} \quad . \tag{13}$$

For example, assuming ε_0^3, ε_1^2, and ε_1^3 are available,

$$\varepsilon_2^2 = \varepsilon_0^3 + \frac{1}{\varepsilon_1^3 - \varepsilon_1^2}; \tag{14}$$

the most common procedure is to calculate and then print one rising diagonal at a time (printed as a row often, for convenience) to avoid unnecessary use of storage. However, the procedure must allow for the fact that most of the entries in the previous diagonal will be needed twice before they are "lost". Thus, ε_1^3 can not be written over ε_0^3 immediately, because ε_0^3 is also needed in calculating ε_2^2. Wynn (1965) suggested the use of auxiliary variables, so as to allow the final storage process to lag one step behind the calculation process. The process can be implemented as shown in Figure 9.

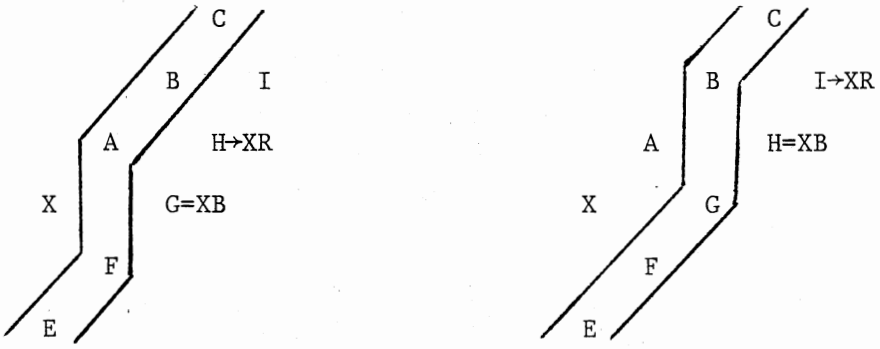

Figure 9. Changes in Storage Allocation for One Step Along the Diagonal

The enclosed area represents the vector used for storing the diag-

onal entries. Entries above the enclosed area are now lost, having already been printed out and written over by entries in later diagonals. Entries below the enclosed area have either not yet been computed, or at least have not been placed in their final storage location. The first portion of the figure assumes we have already calculated E, F, and G along the new diagonal, with G not yet placed in the vector which will hold (only) the new diagonal at exit time. First, A, B, and G are used to calculate H, which is stored temporarily in XR. Now A is no longer needed, so its position in the diagonal vector is filled with G. Next, use XR to reset XB to contain H, leaving the XR slot available for the next computation. This brings us back to the beginning of the process, with everything now ready for the calculation of I. This process continues until you have calculated all the entries in the new diagonal; at the end, you have to add on one additional step for final storage of the last element.

This is as good a time as any to mention the possibility of providing an additional step in the epsilon calculations, so as to permit the user the alternative of computing the Aitken table. The -1, 0, +1, and +2 subscripted columns are identical in the two tables. But when you wish to proceed to the ε_3^i column, do not use ε_1^0, ε_1^1, ε_1^2,... as left sides of the lozenges. Throw them away and use zeros in their place, just as we had zeros on the left for calculating the ε_1^i. And so forth: the odd column entries are calculated and are eligible to serve as lozenge tops and bottoms--but not as left sides. This will give the Aitken table instead of the epsilon table. All the Aitken runs for this thesis were carried out in this way, rather than by using a separate Aitken program.

D. Successful Applications

We will now look at some examples of the epsilon algorithm in action. For the first example only, we shall include the odd columns and use the lozenge arrangement. This example is from Wynn (1956).

Let

$$S_0 = 0, \quad S_{n+1} = \frac{1}{4}(S_n^2 + 2) \quad \text{for } n \geq 0. \tag{15}$$

The reader can easily verify that if the S_n converge, the limit must be $2 \pm \sqrt{2}$. In fact, the lower choice is correct, .5857864.... The convergence is rather slow, but the epsilon algorithm speeds it up considerably. See Table XXXV; correct digits are underlined. (From now on, we will write out the rising diagonals as rows, and the "odd" columns will be omitted.)

TABLE XXXV

THE COMPLETE EPSILON TABLE FOR WYNN'S ITERATION SEQUENCE

	0				
0		2.000			
	.5000		.5714		
0		16.000		89.111	
	.5625		.5851		.5874
0		60.235		1658.7	
	.5791		.58573		.5857857
0		211.06		20262.	
	.5838		.585781		
0		725.94			
	.5852				

The reader might want to verify that, on this particular problem, Aitken's method does just as well. Naturally, we are primarily interested in finding examples which show that the epsilon algorithm is able to handle problems not responsive to the earlier methods.

The obvious place to begin our search is with Shanks' "double" geometric series, which gave the Aitken method great difficulty when x was fairly large. Recall the function could be written as

$$2(1 + x + x^2 + \ldots) - 1(1 + \frac{x}{2} + \frac{x^2}{4} + \ldots). \qquad (16)$$

The e_2 transform should "sum" (16) perfectly, regardless of x, given any five consecutive partial sums. Set $x = 10$ since we know that series gave the Aitken method great difficulty. Part of the errors were due to finite precision in the calculations involving the large partial sums; and that factor still remains to prevent perfect summation by e_2. But e_2 is able to obtain at least eight (out of sixteen) digits for all entries in rows five through ten. Naturally, the entries in the e_2 column which are least susceptible to the calculation difficulties are the entries based on the smaller partial sums. We might expect that the later columns would quickly drift further from the correct answer. But in fact, the drift is quite slow: most of the later entries are about as good as the best number used in computing them. This self-protection against error growth seems rather incredible to the present author. No explanation presents itself at this time; conceivably the good behavior is by chance, but that seems unlikely. See Table XXXVI for the absolute errors of the table entries.

TABLE XXXVI

ABSOLUTE ERRORS IN THE EPSILON TABLE FOR
SHANKS' DOUBLE GEOMETRIC SERIES,
WITH x = 10

	e_1	e_2	e_3	e_4
-.277E-1				
.972E-2				
.160E+2	-.992E-1			
.191E+3	-.434E+0			
.207E+4	-.204E+1	-.47 E-15		
.214E+5	-.992E+1	-.387E-14		
.218E+6	-.489E+2	.230E-13	-.85 E-15	
.220E+7	-.243E+3	.193E-12	-.893E-14	
.221E+8	-.212E+4	.730E-11	.188E-13	-.115E-14
.222E+9	-.604E+4	.773E-10	-.611E-12	-.4-2E-14

For our next example, let us return to Lubkin's series, which completely confused the repeated Aitken method because the ratio of consecutive errors kept switching signs. Recall that the series was

$$1 + \frac{1}{2} - \frac{1}{3} - \frac{1}{4} + \frac{1}{5} + \frac{1}{6} - \ldots = \frac{\pi}{4} + \frac{1}{2}\ln(2) = 1.13197175.... \qquad (17)$$

The epsilon table begins with the same two first columns as the Aitken table because e_1 is just Aitken's method. However, whereas the Aitken method was unable to make any real progress at all, the epsilon method soon approaches the correct limit in the later columns. See Table XXXVII.

Another example where the Aitken method had great difficulty (at the beginning) was the "pi+1" series, which has an alternating component and a monotone component. The monotone component, though it finally becomes insignificant, temporarily kept the Aitken method from making much progress. After fifteen rows, the closest Aitken's method had come to 4.14159265... was 4.142.... By that same point in the partial sums, the

epsilon algorithm will produce 4.14159264... and several other approximations of comparable quality. We see that the new algorithm seems less sensitive to the presence of competing components in the series being summed. This is probably not very surprising, considering the original motivation was built on assuming several components. In their testing, Smith and Ford (1982) found the epsilon algorithm by far the best method for accelerating slowly convergent series with irregular sign patterns. The test problems involved Fourier series; for example, for $0 < x < 2\pi$,

$$\sin x + \frac{\sin 2x}{2} + \frac{\sin 3x}{3} + \ldots = \frac{\pi - x}{2} \tag{18}$$

was one of their test problems.

TABLE XXXVII

THE EPSILON TABLE FOR LUBKIN'S SERIES,
SUM = 1.13197

0				
1.00				
1.50	2.00			
1.17	1.30			
.917	.167	1.0755		
1.12	1.03	1.1248		
1.28	2.12	1.1420	1.1504	
1.14	1.21	1.1333	1.1359	
1.02	.140	1.1285	1.1226	1.1300
1.13	1.07	1.1315	1.1304	1.1317

The epsilon algorithm has also been found useful in definite integral problems which do not fit well into the Romberg scheme. For example,

recall an example from Chapter III; the integral was

$$\int_0^1 \sqrt{x(1-x)}\, dx = \frac{\pi}{8}. \tag{19}$$

As we mentioned, the singularities in the derivatives at both ends of the interval prevent successful application of the Romberg scheme; the powers in the error series for the trapezoidal sums are not h^2, h^4, h^6, ..., but $h^{3/2}$, $h^{5/2}$, $h^{7/2}$,.... If we know that, we can modify the Romberg algorithm accordingly and obtain success in that way. But requiring the user to have such knowledge is obviously undesirable. Chisholm, Genz, and Rowlands (1972) were able to show that the epsilon algorithm will succeed well, given the (step-halving) trapezoidal sums for any problem of the type

$$I = \int_0^1 x^{\alpha}(1-x)^{\beta}\, dx \tag{20}$$

with α and β not integers. Applying the epsilon algorithm to (19) gives us Table XXXVIII. Correct digits are underlined.

TABLE XXXVIII

EPSILON PERFORMANCE ON AN INTEGRAL WITH END-POINT SINGULARITIES

0			
.2500			
.3415	.39433		
.3745	.39302		
.3862	.39276	.39269233	
.3904	.39271	.39269856	
.3919	.39270	.39269904	.392699086

Chisholm et al. (1972) note, however, that the algorithm is not likely to do well on problems with well-behaved integrands. No one has proposed the epsilon algorithm as a substitute for Romberg integration in general; but it is a valuable supplement.

It should be mentioned that the Aitken method also does well on (19); but the analysis of the behavior of the Δ^2 method in later columns has not been carried out as successfully as for the epsilon algorithm. In short, Δ^2 sometimes works well when we can not show why it should. This may seem rather paradoxical, since the Aitken calculations can be done as a slight simplification of the epsilon algorithm. But the additional structure in the epsilon algorithm evidently makes its theory more manageable. The epsilon algorithm also is not able quite to match the Aitken method on the ln (17) series we used earlier, or on Wallis' series. These cases may be indicative of general patterns: Smith and Ford (1982) did find in their testing that the Δ^2 method was superior to the epsilon algorithm on alternating divergent power series and most of the asymptotic series tested. Nevertheless, from the standpoint of being able to understand what is going on, many users would probably prefer the epsilon algorithm even when it is slightly less effective than the Δ^2 method. Certainly Smith and Ford, along with most other writers, regard the epsilon algorithm as more powerful than the Aitken method, over-all.

E. Unsuccessful Applications

By now, the reader may have concluded that the epsilon algorithm is the answer to everything. Unfortunately, no acceleration method is <u>that</u> good. One of our earlier examples was the simple "regrouped" pi series,

$$\pi = 4(1 - \tfrac{1}{3}) + 4(\tfrac{1}{5} - \tfrac{1}{7}) + 4(\tfrac{1}{9} - \tfrac{1}{11}) + \ldots. \qquad (21)$$

As we mentioned earlier, the grouping has given us a logarithmic series, with the ratio of consecutive errors going to 1. The Aitken method did poorly; the epsilon algorithm does no better. See Table XXXIX.

TABLE XXXIX

EPSILON ALGORITHM ATTEMPT ON THE REGROUPED PI SERIES

0			
2.67			
2.90	2.92		
2.98	3.02		
3.02	3.06	3.08	
3.04	3.08	3.10	
3.06	3.09	3.11	3.12

An additional rather disconcerting feature of this example is the following: a reasonable way to estimate the relative error of an entry is to compare its relative "difference", compared to the next entry on its right. (See Chapter IV.) But the epsilon algorithm "stalls" to such an extent that the relative error estimates are often more than one hundred times too low at the end of rows eleven through sixteen. See Table XL.

Another simple example from the first article of Smith and Ford (1979) is the "p" series with p = 2,

$$\frac{\pi^2}{6} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \ldots. \tag{22}$$

The epsilon table is again disappointing; after sixteen rows the smallest absolute error in the table is -.146E - 1. And once again, the error estimates are often considerably too low, though not as bad in the previous example. (The Aitken error estimates on the last two problems, while low, are consistently better than the estimates in the epsilon table. No reason is apparent.)

TABLE XL

ACCURACY OF RELATIVE ERROR ESTIMATES FOR REGROUPED PI SERIES

Row	Estimate	True Relative Error	True/Estimate
11	.389E - 4	.526E - 2	135
12	.145E - 3	.452E - 2	31
13	.201E - 4	.376E - 2	187
14	.785E - 4	.330E - 2	42
15	.115E - 4	.282E - 2	245
16	.461E - 4	.251E - 2	54

As the reader could have guessed by now, Smith and Ford found the epsilon algorithm to be consistently ineffective on logarithmically convergent series. That class of series was one of the very few classes where other methods were found to work much better. In general, given many slowly convergent monotone sequences, the epsilon algorithm tends to be unstable, with relative errors due to finite precision becoming magnified increasingly as the calculations continue. But this is not the case for all monotone sequences. For example, recall the integration

problem we worked with the epsilon algorithm. Some analysis of how well the algorithm may be expected to work on certain classes of series is given by Wynn (1966).

F. The Connection with Older Mathematics

We have previously mentioned the advantage that the epsilon algorithm has over the Δ^2 method, from the standpoint of mathematicians being able to analyze the behavior. Part of the advantage comes from the fact that the (even column) entries in the epsilon table can be related to much older areas of mathematics, areas well developed before the algorithm itself was invented. Those areas are "Padé approximants" and "continued fractions". We now discuss these connections briefly.

1. Padé Approximants - Assume f(x), a given fraction, can be represented by

$$f(x) = a + bx + cx^2 + dx^3 + ex^4 + \ldots . \tag{23}$$

Suppose we want to approximate f(x) by a rational function; for example, suppose we allow a first degree polynomial in both numerator and denominator:

$$Q(x) = \frac{A'x + B'}{C'x + D'} = \frac{Ax + B}{x + D} . \tag{24}$$

We want to choose Q(x) so that it is the "best" rational approximation for f(x), given the allowed degrees. By "best", we mean that we want the power series for Q(x) to coincide with the power series of f(x) as far as possible. Since (24) gives us three parameters to adjust, it seems reasonable to guess that we might be able to match the first three terms of (23), but not four.

The solution process is not hard, though it is a little tedious. We want

$$f(x) - \frac{Ax + B}{1x + D} = Ex^3 + Fx^4 + \ldots, \tag{25}$$

$$f(x)(1x + D) - (Ax + B) = (1x + D)(Ex^3 + Fx^4 + \ldots)$$
$$= E'x^4 + F'x^5 + \ldots. \tag{26}$$

Substituting (23) for f(x), we easily obtain a power series for the left side of (26). Set the coefficients of x^0, x^1, and x^2 equal to zero and we will obtain a linear system which (barring zero determinants) allows solution for A, B, D in terms of the early coefficients in (23).

The same sort of procedure can be done with other degree restrictions. For example, the best R(x) of the form

$$R(x) = \frac{Ax + B}{1x^2 + Dx + E} \tag{27}$$

Would be called the [1/2] Padé approximant to f(x), while our Q(x) would be the [2/2] approximant. The 0/0 , 1/0 , 2/0 ,... approximants of f(x) are exactly the partial sums in (23). If we then put all the rational functions together, we obtain Padé table for f(x), normally arranged as in Figure 10. (For any set value of x, all the rational functions reduce to numbers, of course.)

$[0/0]_x$	$[1/0]_x$	$[2/0]_x$	...
$[0/1]_x$	$[1/1]_x$	$[2/1]_x$	...
$[0/2]_x$	$[1/2]_x$	$[2/2]_x$	...
⋮	⋮	⋮	

Figure 10. The Padé Table Arrangement

The connection of this with the e_i transforms and the corresponding epsilon table is easy: Shanks (1955) showed that if the A_0, A_1,... sequence is being formed from the partial sums of the power series for f(x), evaluated at x, then the $e_k(A_n)$ table consists of <u>half</u> of the Padé table, evaluated at x. The "half" table is now on its side; see Figure 11. (The top "zero" does not fit in this time, since even the $[0/0]_x$ entry is normally not zero.)

$$
\begin{array}{llll}
A_0 = [0/0]_x & & & \\
A_1 = [1/0]_x & & & \\
A_2 = [2/0]_x & [1/1]_1 = e_1(A_0) & & \\
A_3 = [3/0]_x & [2/1]_x = e_1(A_1) & & \\
A_4 = [4/0]_x & [3/1]_x = e_1(A_2) & [2/2]_x = e_2(A_0) &
\end{array}
$$

Figure 11. The Relation Between the Padé Table and the e_k Table

Figure 11 should make clear that if we use a power series of a rational function to generate the initial column of our table, then the epsilon algorithm will find the exact limit--after only a finite number of partial sums. For example, assume

$$g(x) = \frac{5x^3 + 3x + 2}{6x + 7} . \tag{28}$$

If the power series for g(x) is being used to generate the initial column, it is obvious that

$$e_1(A_2) = [3/1]_x = g(x), \tag{29}$$

and all later entries in the e_1 column should be exact, also. (The epsilon algorithm would require some slight modification to avoid division by zero in moving to the following columns; but with that change, all the entries beyond $[3/1]_x$ would be exact, except for round-off error.)

Why is the connection between the epsilon algorithm and the Padé table important? A main reason is that Padé approximation theory had been nicely developed much earlier than the work of Shanks and Wynn. In essence, as soon as the e_k transforms were invented, there were already some nice theorems which described how they should work. For one main example, Montessus de Balloire (1902) had shown under fairly general conditions, the rows of the Padé table had to converge nicely to f(x). Shanks could translate the statement into terms of his e_k table and have the following theorem: if f(z) is analytic for $|z| \leq R$ except for p poles within this circle, and if $\{A_n\}$ is the sequence of partial sums of the power series for f(z), evaluated at some z_0, then the e_p transform will converge uniformly to $f(z_0)$ in the domain obtained from $|z| \leq R$ by removing the interiors of small circles with centers at the poles. For example, let

$$f(z) = \frac{(z - 2)(z + 3)}{(z + 1)(z - 4)} . \tag{30}$$

If we agree to keeping

$$|z + 1| \geq .01, \quad |z - 4| \geq .01, \quad |z| \leq 1000 \tag{31}$$

and generate an e_k table using the epsilon algorithm and the partial sums of the power series for $f(z_0)$, and if we require an accuracy of $5 \cdot 10^{-8}$,

then there will be an integer N such that if n > N, then the $e_2(A_n)$ entry will be within $5 \cdot 10^{-8}$ of $f(z_0)$, regardless of what z_0 is chosen.

When the epsilon algorithm succeeds, it is because the Padé approximations are often better representations of f(x) than a power series in x, in two senses: a sequence of Padé approximants (the $[n/n]_x$ sequence, for example) will often converge to f(x) when the power series diverges. And even if both converge, the Padé approximants may converge much more rapidly.

2. Continued Fractions - The other well-developed theory that was waiting for the invention of the epsilon algorithm was the theory of continued fractions. This theory is related to the Padé approximants but is not a subset of that area. Just as we can represent functions by power series and Padé approximants, continued fraction representations can be derived. A continued fraction is of the form

$$a + \cfrac{b}{c + \cfrac{d}{e + \cfrac{f}{g + \ldots}}} \tag{32}$$

where the division process continues forever, somewhat like the partial sums of an infinite series. The value of (32) is the limit, if it exists, of the "convergents"

$$\frac{0}{1}, \frac{a}{1}, a+\frac{b}{c}, a+\cfrac{b}{c+d/e}, \ldots. \tag{33}$$

There is a difficulty not involved with partial sums: having the n^{th} convergent is not obviously helpful in computing the $(n+1)^{\text{st}}$ convergent. However, there is a way to compute the convergents from the top, down. First, let us agree to write (32) in the (standard) form to save space:

$$a + \frac{b}{c} + \frac{d}{e} + \frac{f}{g} + \cdots . \tag{34}$$

Let P_n/Q_n be the n^{th} convergent,

$$b_0 + \frac{a_1}{b_1} + \frac{a_2}{b_2} + \frac{a_3}{b_3} + \ldots + \frac{a_n}{b_n} = \frac{P_n}{Q_n} . \tag{35}$$

Then P_n and Q_n can be generated directly from the previous ones, for $n \geq 2$, by

$$\begin{aligned} P_n &= b_n P_{n-1} + a_n P_{n-2} \\ Q_n &= b_n Q_{n-1} + a_n Q_{n-2} . \end{aligned} \tag{36}$$

Khovanskii (1963, pp. 2-3) gives a proof. However, this method of computation tends to be unstable numerically (International Dictionary of Applied Mathematics, 1960, p. 374).

Just as we have a standard form for representing f(x) as a series, so there is a standard form for the continued fraction representation for f(x). Namely, the fraction should be of the form

$$f(x) \longrightarrow b_0 + \frac{a_1 x}{b_1} + \frac{a_2 x}{b_2} + \ldots , \tag{37}$$

with the n^{th} convergent having a power series representation agreeing with the f(x) power series through x^n. Such a fraction is called the "corresponding" continued fraction for f(x). There is an algorithm for obtaining the corresponding continued fraction from the power series (Khovanskii, 1963, pp. 27-28).

As it turns out, the corresponding continued fractions are an excellent vehicle for giving new insight into what happens as we move along the diagonals of the epsilon table. To take a typical example from Shanks (1955), the continued fraction representation for $\ln(1+z)$ has

long been known to be

$$\ln(1+z) = 0 + \frac{z}{1} + \frac{1^2 z}{2} + \frac{1^2 z}{3} + \frac{2^2 z}{4} + \frac{2^2 z}{5} + \frac{3^2 z}{6} + \frac{3^2 z}{7} + \ldots . \tag{38}$$

And it has also been widely known that, unlike the power series for $\ln(1 + z)$, which diverges when $|z| > 1$, (38) converges for any complex z except the real $z \leq -1$. Fortunately, the convergents of this continued fraction are exactly the entries of the two top diagonals of the e_k table when we use the power series in the initial column; e.g., let

$$k_0 = 0, \quad k_1 = 0 + \frac{z}{1}, \quad k_2 = 0 + \frac{z}{1} + \frac{1^2 z}{2}, \ldots . \tag{39}$$

Using the power series for $\ln(1 + z)$ to generate the initial column in the e_k table, the convergents in (39) are located in Figure 12.

k_0			
k_1			
x	k_2		
x	k_3		
x	x	k_4	
x	x	k_5	
x	x	x	k_6

Figure 12. Continued Fraction Convergents in the e_k Table

A similar type of thing happens in general. For example, we have earlier seen the function

$$f(x) = \int_0^\infty \frac{e^{-t}}{1 + xt}\, dt. \tag{40}$$

This function is well-behaved for $x \geq 0$; but the power series expansion around zero is woefully inadequate as a good representation, converging for no x except zero:

$$f(x) \sim 1! - 1!x + 2!x^2 - 3!x^3 + 4!x^4 - \ldots. \tag{41}$$

But Sullivan (1978) gives a continued fraction whose convergents will be found in the ε-table (include the top "0") based on (41):

$$f(x) = 0 + \frac{1}{1} + \frac{x}{1} + \frac{x}{1} + \frac{2x}{1} + \frac{2x}{1} + \frac{3x}{1} + \frac{3x}{1} + \frac{4x}{1} + \frac{4x}{1} + \ldots. \tag{42}$$

This gives some insight as to why the epsilon algorithm is able to "sum" the series (41) well; along the top two diagonals, it is converting the wild partial sums of (41) into the nicely behaved convergents of (42).

G. The Special Rules

We have previously mentioned that, due to finite precision, the epsilon algorithm sometimes does not do as well as the theorems (assuming infinite precision) would predict. Difficulties are also inherent in certain cases, regardless of precision. We will end our discussion of the epsilon algorithm by describing how these undesirable situations can occur, and how Wynn (1963), Cordellier (1977), and Brezinski (1978) have suggested dealing with them. As Brezinski says, so much is lost in terms of simplicity that one may often prefer to use the "basic" epsilon algorithm. Nevertheless, the refinements we are about to discuss are crucial

to accuracy in some problems.

Wynn (1963) describes the basic difficulty and lays out the basic pattern for evading it. He gives the first part of the epsilon table generated by the partial sums of the power series for e^x. He then tries setting $x = 2$. One section of the resulting table is shown in Figure 13.

$$-1 = N$$

$$\frac{1}{2} = NW \qquad \frac{1}{2} = NE$$

$$3 = W \qquad \pm\infty = C \qquad 5 = E$$

$$\frac{1}{2} = SW \qquad \frac{1}{2} = SE$$

$$9 = S$$

Figure 13. An Indeterminancy in the Epsilon Table

What has happened is that two entries in the ε_1^i column have accidently become equal. There is no problem in seeing that the 1/2's in the next odd column are appropriate. But the 5 can hardly be obtained using the usual formula. There is no question, however, that the appropriate answer is 5, based on the rational function in that position. Wynn cleverly evades the difficulty by deriving another formula for the 5. He shows that when the $\pm\infty$ occurs in the middle of a "large" lozenge, the correct formula to use is

$$E = S + N - W. \tag{43}$$

Unfortunately, when NW, SW, NE, and SE are <u>almost</u> equal, then (43) is no longer very accurate. And that is the more common case. The $\pm\infty$ is now replaced by a very large "C". Since

$$E = C + \frac{1}{SE - NE} = W + \frac{1}{SW - NW} + \frac{1}{SE - NE}\ , \tag{44}$$

E is highly susceptible to the loss of significant digits via subtraction when SW $\doteq$ NW. The first question at this point might be, how can we write a program to recognize when subtractive cancellation is occurring? The second question would be, what can we do about it?

The first question is very easy. Assume that we have SW and NW to nine significant digits, with SW equal to 98765.4321 and NW equal to 98764.3245. Then SW $-$ NW is 1.1076; even C will lose four significant digits. The way to have a program "discover" this is to estimate the number of digits lost by comparing the difference to the size of one of the original numbers. E.g., if SW is about ten times as large as the difference, you have essentially lost one significant digit. In general, let the estimate of digits lost by C be given by

$$L = \mathrm{Log}_{10}\left(\frac{|SW|}{|SW - NW|}\right). \tag{45}$$

This is the estimator used by Brezinski (1978) to detect when emergency measures are required. He sets a cut-off value for L; if L gets larger than that value, special formulas somewhat similar to (43) are used when we get to E in the large lozenge. We shall next describe those formulas and their implementation, though not their derivation. We shall use the special rules of Cordellier (1977) because they extend to the "vector"

case, which is not true with Wynn's rules. Also, the program needed is slightly less complicated than Wynn's though the final formulas look rather intimidating:

$$E = \frac{\dfrac{N}{(C - N)^2} + \dfrac{S}{(C - S)^2} - \dfrac{W}{(C - W)^2} + \ell C}{\dfrac{1}{(C - N)^2} + \dfrac{1}{(C - S)^2} - \dfrac{1}{(C - W)^2} + \ell} \tag{46}$$

where

$$\ell = \frac{(N - W)^2}{(C - W)^2(C - N)^2} + \frac{(W - S)^2}{(C - W)^2(C - S)^2} - \frac{(S - N)^2}{(C - N)^2(C - S)^2} . \tag{47}$$

The program used in this thesis is, in all major respects, the EPS2 program of Brezinski (1978). However, a few minor revisions have been made: for example, the variables were given more descriptive names. Also, the exit operations were simplified to bring them into line with the simple relative error estimate prescribed by Brezinski himself on page 330. (It is the same type of estimate we have used before, except that he compares, for example, $e_2(A_1)$ with $e_1(A_2)$ instead of $e_1(A_3)$. Refer back to Figure 7.) The present author is convinced that the relative error estimates of many of the (excellent) programs in Brezinski's book are slightly faulty. (The interested reader who has access to the 1978 book may verify this by printing out A1, A2, and E values in the last section of EPS2.) A couple of other potential difficulties in some of the programs are the use of automatic initialization of variables to zero, and assumed preservation of local variables between subroutine calls. It is also puzzling that, while the programs insert a zero at the top of each table, the extra entries which result (and are computed) are never printed out. The program used in this thesis inserts the zero, but has been modified so that the extra entries are eligible for printing. Revisions of this sort

are very minor, of course; the programs will be of great value to anyone who is wanting to learn the details of implementing the algorithms discussed.

The routine implemented by Brezinski will allow handling several of the instability occurrences at once, so long as we do not have three consecutive entries of one column close together by accident. (That would result in N - C and C - S being close to zero, which would create obvious problems in (46) and (47).) Such an occurrence causes termination of the calculations. Queues are maintained for the W and N values which will be needed in the future, as well as for the column numbers where the cancellation occurred. (This last queue lets us know when we are ready for the next E calculation.) The program prints out only the last (permitted) even column entry on each rising diagonal of the epsilon table.

Brezinski (1978, p. 319) gives a nice example to illustrate the difference the special measures can make. Let

$$\begin{aligned} &S_1 = 1.5999999, \quad S_2 = 1.2, \quad S_3 = 1.0 \\ &S_{n+3} = \tfrac{1}{2}S_{n+2} + \tfrac{1}{4}S_{n+1} + \tfrac{1}{8}S_n, \qquad n = 1,2,.... \end{aligned} \tag{48}$$

It is clear zero can be written as a weighted average of any four consecutive partial sums; namely,

$$0 = \frac{8S_{n+3} - 4S_{n+2} - 2S_{n+1} - S_n}{8 - 4 - 2 - 1} . \tag{49}$$

This should remind us of the sum of three geometric series, with zero as the sum. Therefore, it is not difficult to believe Brezinski when he asserts that the e_3 column should be all zeroes. (The top entry is probably an exception, if we begin the table with the initial zero as

usual. The recursion in (48) does not hold for 0, 1, 1.2 and 1.5999999.) If we set the upper bound on the L of (45) at 20 and use EPS2, the special rules will never come into effect. EPS2 thus uses the usual epsilon calculations, which give an e_3 column as shown in the left half of Table XLI. If we set the upper limit at 4 instead of 20, the special rules are used only in computing two entries in the first thirteen rows. However, the carried over effect in the e_3 column is quite striking; see the right half of Table XLI.

TABLE XLI

THE EFFECTS OF THE SPECIAL RULES IN BREZINSKI'S EXAMPLE

Usual Rule	Special Rules (Twice)
.247E - 1	-.200E - 6
.215E - 1	.888E - 15
.318E - 2	-.125E - 14
.319E - 1	-.289E - 14
.149E - 1	-.430E - 15
.241E - 2	.928E - 16
.278E - 15	.278E - 15

It seems appropriate to examine in some detail what has happened in this example. The two entries which were calculated by the special E rule are located as indicated in Figure 14. The later entries in the table which are influenced by them are enclosed by the dotted line.

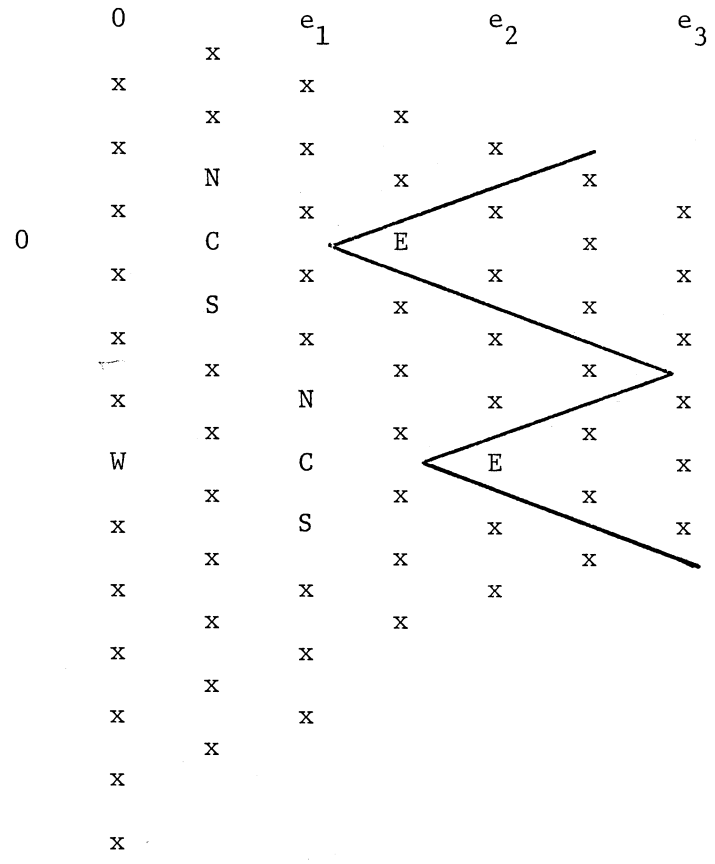

Figure 14. The Position of the E Entries in Brezinski's Example, and Their Areas of Influence

The upper E in Figure 14 was -.174999995...; the simple algorithm obtained -.125678... in that position, off by about 28%. For the upper E, the respective values were -14.999999... and -14.92139..., a bit closer. The NW and SW entries in the upper large lozenge show how the special rules were triggered. See Figure 15.

Trying to attain even more accuracy by setting the upper limit on digit loss at 1 will result in three consecutive entries being close enough to trigger the abortion process before the e_3 column is ever

reached.

N = -5.00000000

NW = 1.00000000 NE = .99999999

W = 0 C = -.800E+8 E = -14.99999938

SW = .99999999 SE = 1.00000000

S = -10.0000006

Figure 15. The Upper E Lozenge in Brezinski's Example

H. Conclusions

We have seen the epsilon algorithm is in essence a generalization of Aitken's method, dealing with the sum of several geometric series instead of one. This generalization gave many new powers not enjoyed by the Δ^2 method, although the Δ^2 method does tend to be slightly superior on some classes of problems, for reasons that are not always clear. The epsilon algorithm, like many other acceleration methods, does have difficulty with logarithmic convergence. Therefore, one would rather have an alternating sequence to accelerate via the epsilon method, though the results are fine for some monotonic sequences.

We have seen that the epsilon algorithm's success can be made more understandable by its connections with Padé approximants and continued fractions. And finally, since a great deal of subtractive cancellation sometimes takes place, we have seen how to counteract the effects of this

by using other formulas which, while more complicated, are able to minimize the impact of C and avoid the subtraction SE - NE completely.

The reader who is interested in pursuing the study of the epsilon algorithm will have no difficulty in finding enough material to occupy anyone for twenty years. Wynn has virtually built a career writing on his invention. Anyone who can read French will certainly want to have Brezinski's books (1977, 1978) on hand. The bibliography of the earlier one is, by itself, worth the price of the book. The programs and examples in the latter one are indispensable, and mostly understandable even to the person who does not read French. Wimp (1981) has also written a book which is an excellent reference on the epsilon algorithm, as well as well as on many other methods.

CHAPTER VI

A GLIMPSE OF SOME "RELATIVES" OF THE EPSILON ALGORITHM

A. The Rho Algorithm

We saw in the previous chapter that the epsilon algorithm, while very powerful, tends to have trouble with logarithmically convergent series. Evidently, assuming a sum of geometric series does not give a good model for the logarithmic convergence. At almost the same time that Wynn invented the epsilon algorithm, he also designed the "rho" algorithm as a complementary algorithm (Wynn, 1956 b). The rho algorithm is based on Thiele's reciprocal differences, which are well-explained by Milne-Thomson (1933). Instead of assuming the initial column is approaching its limit as a sum of geometric series would, the assumption is that the limit is being approached in the same manner as a rational fraction approaches its horizontal asymptote. Such an approach is eventually monotone, of course. Thus, the rho algorithm is predisposed to good performance on monotone sequences. And in particular, if the initial columns are actually being generated by the rule

$$S_n = \frac{Sn^k + a_1 n^{k-1} + \ldots + a_k}{n^k + b_1 n^{k-1} + \ldots + b_k}, \tag{1}$$

then the $k^{\underline{th}}$ even column of the ρ table will be all S's (Brezinski,

1977, p. 104). Naturally the method is useless on oscillating sequences.

Fortunately, the rho algorithm (at least in the original "simple" form used by Wynn, which is the only one we will discuss) requires only a trivial change in the calculations made for the epsilon algorithm. If we number the columns starting with "zero" in the initial column, then the rule for calculating entries in the $k^{\underline{th}}$ column is

$$\text{Right} = \text{Left} + \frac{k}{\text{Bottom-Top}}, \tag{2}$$

which requires only an additional branch in the epsilon subroutine. All the rho tables for this thesis were thus done with the same program that implemented the epsilon and Aitken algorithms, with the program notified by a numerical code as to which algorithm was desired.

Let us begin with the rho algorithm's performance on the regrouped pi series. The best answer Aitken was able to produce in eight rows was 3.1265; the epsilon algorithm was even worse, attaining only 3.1108. the rho algorithm does considerably better. See Table XLII. Correct digits are underlined in the last two columns.

TABLE XLII

RHO PERFORMANCE ON THE REGROUPED PI SERIES

0			
2.667			
2.895	3.167		
2.976	3.145		
3.017	3.143	3.14139	
3.042	3.142	3.14156	
3.058	3.142	3.141585	3.1415929
3.070	3.142	3.141590	3.1415929

For another example, let us use the new algorithm on the p series with p = 2, Zeta(2) = $\pi^2/6$ = 1.6449341.... In eight rows, the Aitken method could only reach 1.618; the epsilon algorithm fell behind again, with 1.590. Once again, the rho algorithm excels. See Table XLIII.

TABLE XLIII

RHO PERFORMANCE ON THE LOGARITHMIC ZETA(2) SERIES

0			
1.000			
1.250	1.667		
1.361	1.650		
1.424	1.647	1.64474	
1.464	1.646	1.64489	
1.491	1.645	1.644923	1.644936
1.512	1.645	1.644929	1.6449344

By now, it may be appearing that the rho algorithm is "the" answer for logarithmic series. Unfortunately this is not the case. Smith and Ford (1979) found that on six of the eight logarithmic test problems they used, the rho algorithm did do slightly better than even the best of the competing algorithms. However, on the last two problems, the rho algorithm completely failed, while some of the competitors continued to perform quite well. One of the failures was caused by the series

$$(1 + e^{1})^{-\sqrt{2}} + (2 + e^{1/2})^{-\sqrt{2}} + (3 + e^{1/3})^{-\sqrt{2}} + \ldots = 1.7137967\ldots \quad (3)$$

After fifteen rows, the rho algorithm has reached only 1.592.... (It is

"stalling" enough that the relative error estimates are off by a factor of ten, but this does not seem too bad.) Because there were some methods that did quite well on all eight test problems, Smith and Ford concluded that the slight advantage of "rho" on most of the problems was outweighed by its failure on the last two. Brezinski (1977, p. 106) comments that the rho method has not been used too much because of the lack of theorems concerning its convergence. Nevertheless, he has found a slightly modified rho algorithm superior to Romberg integration in some situations.

B. The Theta Algorithm

Brezinski (1971) invented the "theta" algorithm, which is in a sense a hybrid of the epsilon and rho algorithms. He comments (1977, p. 123) that the method is more versatile than either the epsilon or rho methods: it may not quite match the better of its two relatives on a particular problem; but it tends to do quite well both on "epsilon-type" problems and "rho-type" problems. Smith and Ford (1979) ranked theta as one of the top three methods, although they later decided (1982) to rate the epsilon method above it for general use, if the "u" algorithm of Levin (1973) is available as an alternative to epsilon. (We will not try to discuss "u" in this thesis, unfortunately.)

The theta algorithm is essentially the epsilon algorithm with an acceleration parameter involved in the even column calculations to make those columns converge more rapidly. The odd columns are calculated as in the epsilon algorithm. The entries of the table which are used in the acceleration parameters do not all lie in the lozenge of the epsilon algorithm. See Figure 16 for the configuration of the entries used in the calculations for the even columns.

XT

XL XR

XB

X2BCK

XSUB

Figure 16. The Entries Needed in Calculating XR (Even Column)

The formula is

$$XR = \frac{X2BCK(XSUB - XB) - XL(XB - XT)}{(XSUB - XB) - (XB - XT)}. \qquad (4)$$

The presence of XSUB in Figure 16 forces abandonment of our usual movement along diagonals. The calculations by Brezinski's THETA on one cell fall along what could be called a "weave", rather than along a diagonal. The calculations for a weave require the accessability of the two previous weaves. See Figure 17 for the appearance of the weaves and one of the even column XR configurations.

The program used in this thesis was, in all important aspects, Brezinski's THETA (1978, p. 369). We have taken the liberty of making the same sort of changes we earlier made in EPS2. THETA is designed to print out only the sequence elements, the last even column entry in the

weave, and a relative error estimate.

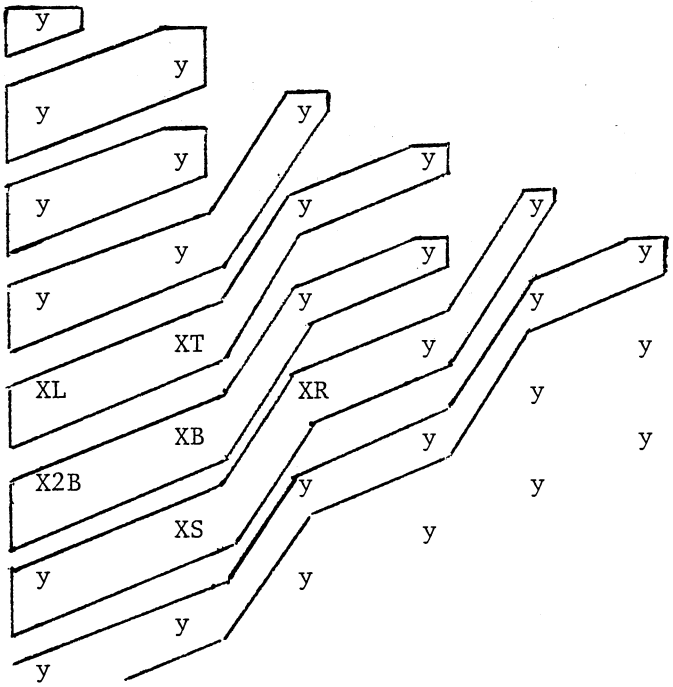

Figure 17. The Order of Calculation for Theta Algorithm

First, let us apply THETA to the pi and regrouped pi series, recalling that epsilon succeeds only on pi and rho succeeds only on the regrouped series. After twelve rows on the pi series, epsilon had 3.14159267; after twelve rows on the regrouped series, rho had 3.14159265360. (Recall, we progress more rapidly into the later terms on the regrouped series.) By the same point in the partial sums, THETA gives 3.14159265261 on the pi series and 3.14159265341 on the regrouped series. At least on these two problems, THETA is not excelled on either

problem by either of the other two methods.

An even more impressive demonstration is given on (3), the logarithmic series which caused the rho algorithm to fail. The correct limit is 1.7137967.... The best entry on row fifteen is 1.46 for Aitken, 1.33 for epsilon, and 1.59 for rho. See Table XLIV for the numbers given at the end of the weaves by THETA through the same partial sums. It is not difficult to see why the method is highly regarded.

TABLE XLIV

THETA PERFORMANCE ON SERIES (3)

Weave Number	Last Partial Sum	Approximation
10	.805	1.83
11	.838	1.73
12	.867	1.718
13	.894	1.7142
14	.917	1.71366
15	.939	1.713766

With the inclusion of these last two algorithms, we have what might be called the epsilon family of methods. Taken together, they form a very potent arsenal of acceleration methods. The theory is most developed for the epsilon algorithm itself; but the theta algorithm is currently the focus of much research, for obvious reasons.

CHAPTER VII

SUMMARY AND AREAS FOR FURTHER STUDY

The reader may have hoped at the beginning of this paper to be shown some acceleration method which would excel on every slowly converging sequence. But by now, this hope must surely have faded: there are obviously too many different manners for slowly convergent sequences to approach their limit. As Wimp (1982, p. x) points out, the only methods which can accomodate virtually any manner of approach pay the price of well-rounded mediocrity. At this stage of development of the theory, the choice of methods is, generally speaking, an art as much as a science. There are general rules of thumb: for example, the epsilon algorithm "often" does very well on oscillating sequences, and "often" does not do well on monotone sequences. But these statements are not very precise. Often you just try a method on a sequence; and if it seems to be converging to something, you "hope" it is the correct limit! If two methods approach the same limit, so much the better.

Two acceleration researchers in the French "school" (led by Brezinski) are Delahaye and Germain-Bonne. In a 1980 article they showed that, for example, it is impossible for any method (now or in the future) to accelerate every logarithmic sequence. This is a rather amazing thing to prove, to say the least. But it indicates that future research can, at best, hope to find methods which succeed a higher percentage of the time. The current situation of having a method work well and then fail

miserably is, therefore, somewhat intrinsic to the subject.

Brezinski (1980) has recently invented a "super-algorithm" which includes many of the others as special cases, depending on the choice of parameters. He has now published a corresponding computer code (Brezinski, 1982). This may help bring the theory together a bit more.

It would have been nice to include in this paper a study of the "u" transform of Levin (1973), which Smith and Ford (1979, 1982) regard so highly. But including it did not turn out to be feasible. Professor Smith has graciously provided a preprint of an article that will soon appear in the ACM Transactions on Mathematical Software. It discloses a nice iterative scheme for u which has not been previously available in print.

There are two advantages of the epsilon and rho algorithms which the present author (and, no doubt, many others) would like to see extended to some of the other algorithms, such as Euler's, Aitken's, theta, and u. Namely, we know precisely what kind of sequence in the initial column will produce "perfect" answers in the e_2, e_3,..., columns. But, to take one example, what kind of sequence do you have to start with in order for the Δ^2 method to produce perfect answers in two or three columns? The present author worked on this for a while, but with no results. If this sort of question could be answered, it would be of great benefit in understanding exactly what the Aitken method is doing. Cordellier (1977) has shown some preliminary results for the theta algorithm: he has characterized the sequences which one application of theta solves exactly; it turns out that this collection contains all sequences which either epsilon or rho solves in one application. Smith and Ford (1979) also give some "exactness" results for u.

There will undoubtedly be great advances in the future. This field really began growing only with the invention of the epsilon algorithm. Shanks (1955, p. 40) commented that "the literature on non-linear transforms is not very large." The situation is completely different now; indeed, one is just about half-way through a new important paper on the subject when another one is published. There is perhaps no area of mathematics that is more exciting at the present moment than acceleration methods.

BIBLIOGRAPHY

Agnew, J. Explorations in Number Theory. Monterey, California: Brooks/Cole, 1972.

Aitken, A. C. "On Bernoulli's Numerical Solution of Algebraic Equations." Proceedings of the Royal Society of Edinburgh, 46 (1926), 289-305.

Ames, L. D. "Evaluation of Slowly Convergent Series." Annals of Mathematics, 2nd Series, 3 (1901), 185-192.

Barbeau, E. J. "Euler Subdues a very Obstreperous Series." American Mathematical Monthly, 86 (1979), 356-372.

Bartle, R. G. The Elements of Real Analysis. New York: John Wiley and Sons, 1964.

Bauer, F. L., H. Rutishauser, and E. Stiefel. "New Aspects in Numerical Quadrature." Proceedings of Symposia in Applied Mathematics, 15 (1963), 198-218.

Boas, R. P. "Estimating Remainders." Mathematics Magazine, 51 (1978), 83-89.

Brezinski, C. "Accélération de Suites à Convergence Logarithmic." Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences, 273A (1971), 772-774.

Brezinski, C. Accélération de la Convergence en Analyse Numérique. New York: Springer-Verlag, 1977.

Brezinski, C. Algorithmes d'Accélération de la Convergence: Étude Numérique. Paris: Éditions Technip, 1978.

Brezinski, C. "A General Extrapolation Algorithm." Numerische Mathematik, 35 (1980), pp. 1980.

Brezinski, C. Padé-Type Approximation and General Orthogonal Polynomials. Boston: Birkhaüser-Verlag, 1980.

Brezinski, C. "Algorithm 585 - a Subroutine for the General Interpolation and Extrapolation Problems." ACM Transactions on Mathematical Software, 8 (1982), 290-301.

Bromwich, T. J. I'A. An Introduction to the Theory of Infinite Series. 2nd Edition, Revised. London: MacMillan and Company, 1926.

Buchanan, J. "The Errors in Certain Quadrature Formula." Proceedings of the London Mathematical Society, 34 (1902), 335-345.

Bulirsch, R. "Bemerkungen zur Romberg-Integration." Numerische Mathematik, 6 (1964), 6-16.

Calabrese, P. "A Note on Alternating Series." American Mathematical Monthly, 69 (1962), 215-217.

Chisholm, J. S. R., A. C. Genz, and G. E. Rowlands. "Accelerated Convergence of Sequences of Quadrature Approximations." Journal of Computational Physics, 10 (1972), 284-307.

Conte, S. D. and C. de Boor. Elementary Numerical Analysis: an Algorithmic Approach. 2nd Edition. New York: McGraw-Hill, 1972.

Conway, J. B. Functions of One Complex Variable. New York: Springer-Verlag, 1973.

Cordellier, F. "Particular Rules for the Vector ε-algorithm." Numerische Mathematik, 27 (1977), 203-207.

Cordellier, F. "Caracterisation des Suites que la Première Étape du θ-Transforme en Suites Constants." Comptes Rendus Hebdomadaires des Séances de l'Academie des Sciences, 284A (1977), 389-392.

Corey, S. A. "A Method of Approximation." American Mathematical Monthly, 13 (1906), 137-140.

Davis, P. J. "On the Numerical Integration of Periodic Analytic Functions." On Numerical Approximation. Editor R. E. Langer. Madison: University of Wisconsin Press, 1959.

Davis, P. J. "Errors of Numerical Approximations for Analytic Functions." Survey of Numerical Analysis. Editor John Todd. New York: McGraw-Hill, 1962.

Delahaye, J. P. and B. Germain-Bonne. "Resultats Négatifs en Accélération de la Convergence." Numerische Mathematik, 35 (1980), 443-457.

Engles, H. Numerical Quadrature and Cubature. New York: Academic Press, 1980.

Erdelyi, A. Asymptotic Expansions. New York: Dover Publications, Inc., 1956.

Euler, L. Institutiones Calculi Differentials. St. Petersburg: Academiae Imperialis Scientiarum, 1755.

Fox, L. "Romberg Integration for a Class of Singular Integrands." Computer Journal, 10 (1967), 87-93.

Froebinius, G. "Ueber Relationen Zwischen den Näherungsbrüchen von Potenzreihen." Journal fur die reine und angewandte Mathematik, 90 (1881), 1-17.

Gragg, W. B. "Lecture Notes on Extrapolation Methods." (Paper presented at the SIAM National Meeting, Seattle, Washington, June, 1971.)

Hardy, G. H. Divergent Series. Oxford: Clarendon Press, 1949.

Havie, T. "Romberg Integration as a Problem in Interpolation Theory." BIT, 17 (1977), 418-429.

Henrici, P. Elements of Numerical Analysis. New York: John Wiley and Sons, 1964.

Hildebrand, F. B. Introduction to Numerical Analysis. New York: McGraw-Hill, 1956.

International Dictionary of Applied Mathematics. New York: D. Van Nostrand, 1960.

Jacobi, G. C. I. "Uber die Darstellung einer Reice gegebener Werte durch eine gibrochene Rationale Funktion." Journal für die reine uber angewandte Mathematik, 30 (1846), 127-156.

Johnson, L., and R. Riess. Numerical Analysis. Reading, Massachusetts: Addison-Wesley, 1977.

Joyce, D. C. "Survey of Extrapolation Processes in Numerical Analysis." SIAM Review, 13 (1971), 435-486.

Khovanskii, A. N. The Application of Continued Fractions and their Generalization to Problems in Approximation Theory. Translator P. Wynn. Groningen: P. Noordhoff N. V., 1963.

Knopp, K. Theory and Applications of Infinite Series. 4th Edition. Translator R. C. H. Young. New York: Hafner, 1951.

Krylov, V. I. Approximate Calculation of Integrals. Translator A. H. Stroud. London: MacMillan, 1962.

Kummer, E. "Eine neue Methods die numerische Summen langsam convergierender Reihen zu berechnen." Journal für die reine und angewandte Mathematik, 16 (1837), 206-214.

Laurie, D. P. "Propagation of Initial Rounding Error in Romberg-like Quadrature." BIT, 15 (1975), 277-282.

Levin, D. "Development of Non-linear Transformations for Improving Convergence of Sequences." International Journal of Computer Mathematics, B3 (1973), 371-388.

Lubkin, S. "A Method of Summing Infinite Series." Journal of Research, National Bureau of Standards, B48 (1952), 228-254.

Lynch, R. E. "Notes on Romberg Quadrature." (Report TNN-56, University of Texas Computation Center, Austin, Texas, 1965.)

MacDonald, J. R. "Accelerated Convergence, Divergence, Iteration, Extrapolation, and Curve Fitting." Journal of Applied Physics, 35 (1964), 3034-3041.

Milne, R. M. "Extension of Huygen's Approximation to a Circular Arc." Mathematical Gazette, 2 (1903), 309-311.

Milne-Thomson, L. M. Calculus of Finite Differences. London: MacMillan, 1933.

Moler, C. "Extrapolation to the Limit." (Paper delivered to the Engineering Summer Conference on Numerical Analysis, Ann Arbor, Michigan, 1967.)

Montessus de Balloire, R. "Sur les fractions continues algébriques." Bulletin de la Société Mathematique de France, 30 (1902), 28-36.

Neville, E. H. "Iterative Interpolation." Proceedings of the International Congress of Mathematicians. Zurich: Orell-Fussli-Verlag, 1932.

Poncelet, J. V. Application de la Méthode des Moyennes á la Transformation, au Calcul Numerique et a la Détérmination des Limites du Reste des Series." Journal für die reine und angewandte Mathematik, 13 (1835), 1-54.

Pinsky, M. "Averaging an Alternating Series." Mathematics Magazine, 51 (1978), 235-237.

Richardson, L. F. "The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations, with an Application to the Stresses in a Masonry Dam." Royal Society of London Philosophical Transactions, A210 (1910), 307-357.

Richardson, L. F. "Theory of the Measurement of Wind by Shooting Spheres Upward." Royal Society of London Philosophical Transactions, A223 (1923), 345-382.

Richardson, L. F. "The Deferred Approach to the Limit." Royal Society of London Philosophical Transactions, A226 (1927), 299-349.

Romberg, W. "Vereinfachte Numerische Integration." Det Kongelige Norske Videnskabers Selskabs Forhandlinger, 28 (1955), 30-36.

Rutishauser, H., and E. Stiefel. "Remarques concernant 1'Integration Numérique." Comptes Rendus Hebdomadaires des Séances de 1'Académie des Sciénces B252 (1961), 1899-1900.

Rutishauser, H. Description of ALGOL 60. Berlin, Springer-Verlag, 1967.

Schmidt, R. J. "On the numerical solution of linear simultaneous equations by an iterative method." The Philosophical Magazine and Journal of Science, 7th Series, 32 (1941), 369-383.

Shampine, L. F., and R. C. Allen, Jr. Numerical Computing: an Introduction. Philadelphia: W. B. Saunders, 1973.

Shanks, D. "Non-linear Transformations of Divergent and Slowly Convergent Series." Journal of Mathematics and Physics, 34 (1955), 1-42.

Sheppard, W. F. "On Some Quadrature Formula." Proceedings of the London Mathematical Society, 34 (1900), 335-345.

Smith, D., and W. Ford. "Acceleration of Linear and Logarithmic Convergence." SIAM Journal on Numerical Analysis, 16 (1979), 223-240.

Smith, D., and W. Ford. "Numerical Comparisons of Nonlinear Convergence Accelerators." Mathematics of Computation, 38 (1982), 481-499.

Smith, D., T. Fessler, and W. Ford. "HURRY: An Acceleration Algorithm for Scalar Sequences and Series." ACM Transactions on Mathematical Software, in press.

Standard Mathematical Tables. 22nd Edition. Cleveland: The CRC Press, 1974.

Stroud, A. H. Numerical Quadrature and Solution of Ordinary Differential Equations. New York: Springer-Verlag, 1974.

Sullivan, J. "Padé Approximants via the Continued Fraction Approach." American Journal of Physics, 46 (1978), 489-494.

Todd, J. "Motivation for Working in Numerical Analysis." Survey of Numerical Analysis. Editor J. Todd. New York: McGraw-Hill, 1962.

Tucker, R. R. "The Δ^2-Process and Related Topics." Pacific Journal of Mathematics, 22 (1967), 349-359.

Widder, D. The Laplace Transform. Princeton: Princeton University Press, 1946.

Wimp, J. Sequence Transformations and Their Applications. New York: Academic Press, 1981.

Wynn, P. "On a Device for Computing the $e_m(S_n)$ Transformation." Mathematical Tables and Other Aids to Computation, 10 (1956), 91-96.

Wynn, P. "On a Procrustean Technique for the Numerical Transformation of Slowly Convergent Sequences and Series." Proceedings of the Cambridge Philosophical Society, 52 (1956), 663-671.

Wynn, P. "Singular Rules for Certain Nonlinear Algorithms." BIT, 3 (1963), 175-195.

Wynn, P. "A Note on Programming Repeated Application of the ε-algorithm." Revue Francaise de Traitement de 1'Information. Chiffres, 8 (1965), 23-62.

Wynn, P. "On the Convergence and Stability of the Epsilon Algorithm." SIAM Journal on Numerical Analysis, 3 (1966), 91-122.

Wynn, P. "A Note on the Generalized Euler Transform." Computer Journal, 14 (1971), 437-440.

APPENDICES

APPENDIX A

A TEXTBOOK SUPPLEMENT ON

ACCELERATION METHODS

Infinite series and sequences are an important part of both pure and applied mathematics. In Calculus II you learned various theorems which help to determine whether the partial sums S_n of a series are going to converge to a limit or diverge. But if we have any practical tendencies at all, showing only that the limit does exist is not completely satisfying. The natural next question is, "How do we actually find what the sum is?" Of course, adding up an infinite number of terms is not really an option; but on some series something "close" to an infinite number of partial sums is required if we want good accuracy. The topic we are going to look at involves various methods used to accelerate the process of finding the limit with a good deal of accuracy. In some cases we will get good answers when no number of partial sums could get close!

For example, take the standard series

$$\ln(2) = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \ldots, \tag{1}$$

which results from setting $x = 1$ in a Maclaurin series,

$$\ln(x+1) = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \frac{1}{4}x^4 + \ldots. \tag{2}$$

Assume that we don't know where the right side of (1) came from; we can still assert, via the alternating-series test, that the partial sums do converge to something. And in fact a standard theorem tells us that the error of any partial sum is, in this situation, less than the size of the first omitted term. Let us suppose we demand an error less than $5 \cdot 10^{-9}$, which amounts to eight decimal-place accuracy. How many terms are needed? It would be possible to answer this quite precisely by using a theorem discovered by a beginning calculus student in 1962 (Calabrese, 1962). However we choose to take a less rigorous approach, to get an approximate answer to our question. If S_n and S_{n+1} are both within $5 \cdot 10^{-9}$ of the time limit, they must be within 10^{-8} of each other. Therefore, the term a_{n+1} which is added (or subtracted) from S_n to obtain S_{n+1} has a magnitude less than 10^{-8}. But a_{n+1} is $1/(n+1)$, which means it is necessary (though perhaps not sufficient) to require

$$\frac{1}{n+1} < 10^{-8}$$

$$n > 99{,}999{,}999. \qquad (3)$$

Adding that many terms is obviously out of the question, even on a large computer. If we do not already know the limit is ln(2), we are not going to be able to find that limit to the required tolerance, unless we can do something besides simply adding up partial sums. That is what acceleration methods are designed for, in hopes of taking the earlier partial sums and "coaxing" some more information out of them.

Exercise 1. A standard series is

$$\pi = 4(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \ldots); \tag{4}$$

Assume that we are far enough into the summing so that S_{n+1} and S_n are correct to nine decimal digits. What is the smallest n that has a chance of making this happen?

Exercise 2. If we take the harmonic series

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \ldots, \tag{5}$$

we can use an argument similar to the one above to show that using $n > 99{,}999{,}999$ will keep S_{n+1} and S_n within 10^{-8} of each other. Does it follow that both of them are within $5 \cdot 10^{-9}$ of the "limit?" What is different?

Every acceleration method, no matter how sophisticated, depends on making some assumption about the manner in which the partial sums approach the limit. If we apply the particular method to a series which behaves (more or less) in the assumed manner, the results can be spectacular. If the assumptions are <u>not</u> close to being appropriate for that series, the acceleration method may hinder rather than help. Fortunately, it is often possible to tell easily which of these situations is occurring; this will become clear.

Exercise 3. Use a calculator to compute the first six partial sums of (1); graph them on graph paper as y values, at x = 1, 2, . . . 6. Also graph the line y = ln(2). Can you see a way to use the partial sums in

pairs to obtain five new approximations which give a considerably tighter fit around the line y = ln(2)? Calculate those five new approximations and graph them. Explain how this process might be continued.

The simple process which you recently discovered in Exercise 3 was first used by the great mathematician Leonard Euler in 1755. It is mainly for use with oscillating sequences which appear to be oscillating in a fairly symmetrical way about the limit. At each step of the process, the calculation is based on the assumption of exactly opposite errors in consecutive partial sums. That isn't exactly a true assumption for the ln(2) series; but it is close enough to being true that the method works well. To get the nicest connections with Euler's theory, an initial zero should be placed at the beginning. We then can have the arrangement in Figure 18 which we will call the Euler table.

$$
\begin{array}{llllll}
0 = S_1 & & & & & \\
S_2 & t_1 = (S_1+S_2)/2 & & & & \\
S_3 & t_2 = (S_2+S_3)/2 & u_1 = (t_1+t_2)/2 & & & \\
S_4 & t_3 = (S_3+S_4)/2 & u_2 = (t_2+t_3)/2 & v_1 = (u_1+u_2)/2 & & \\
S_5 & t_4 = (S_4+S_5)/2 & u_3 = (t_3+t_4)/2 & v_2 = (u_2+u_3)/2 & w_1 = (v_1+v_2)/2 &
\end{array}
$$

Figure 18. The Euler Table

Before we get you involved in implementing the algorithm, some explanation is necessary about the pattern we are going to follow in

studying all the algorithms. They all are going to give tables that are similar in structure. Therefore, to make it possible for you to spend less time on things like FORMAT statements and more time on studying the algorithms themselves, the subroutine ANLYZ has been written to take care of all output for all the methods. You will find more details in the program listing itself, but here is the general idea: you will have to write a subroutine to calculate one row of the table per call. You will also use either a supplied function subprogram which calculates one new partial sum per call, or write one yourself. The procedure is then to call ANLYZ, supplying it with the relevant information about what kind of table is desired. ANLYZ will then follow this pattern repeatedly: obtain a partial sum from the function subprogram, call your row-generating subroutine, and print out the results. After all desired rows have been thus processed, ANLYZ returns control to the calling program.

Some explanation is also needed for the procedures used by ANLYZ in its error analysis. For use in problems when we don't know the correct limit, we would like to have estimators for the relative and absolute error of a particular entry. The estimates should be calculated using only the table entries themselves, not the actual limit. For example, in Figure 18, the actual values for the absolute and relative error of u_3 would be

$$\begin{aligned} \text{True A.E.} &= u_3 - \text{CORRECT} \\ \text{True R.E.} &= (u_3 - \text{CORRECT})/\text{CORRECT} \end{aligned} \tag{6}$$

Although our next step may appear rather desperate, it generally works fairly well: estimate how close u_3 is to the correct answer by how close

u_3 is to v_2! Thus, the estimates for u_3 are

$$\begin{aligned} \text{Est. A.E.} &= u_3 - v_2 \\ \text{Est. R.E.} &= (u_3 - v_2)/v_2. \end{aligned} \tag{7}$$

ANLYZ will complete and print all the numbers in (6) and (7). You will thus see, not only how well the algorithm is actually performing, but also how well our "blind" estimators of the performance are working. Note that our procedures will give no estimates of the accuracy of the last entry on a row.

Exercise 4. Write a subroutine which will use the following information: the row number, the new partial sum, the entries in the previous row of the Euler table, and the maximum number of row entries allowed by the user. The subroutine should then furnish the following information on return: the contents of the new row (written over the old row) and the number of entries in the new row. See ANLYZ for the exact structure required of METHOD, the "place-holder" name for your subroutine. (The second form shown is appropriate for the Euler method). You may assume the row numbers begin with "1." You will need some "local" storage but ask your teacher whether local vectors are permitted.

Exercise 5. Write a (short) program to call ANLYZ three times to build three Euler tables using your subroutine and the supplied subprogram LN1PX; the calling program should store x = 1.D0 in COMMON/XLINK/, so that the correct limit is DLOG(2.D0). The tables are to have the following features:

(1) Twelve rows, but never more than six columns. Error analysis is to be included.

(2) Same as (1), but no error analysis.

(3) Same as (1), but "twenty" specified as the maximum number of columns allowed.

Exercise 6. Discuss your table from the third part of Exercise 5, on the following points:

(1) What is the ratio of the A.E. in the tenth partial sum to the A.E. in the best entry in row ten?

(2) Can you see any pattern as to where the best entry on any row is apt to lie?

(3) Where on the rows are the horizontal differences (absolute error estimates) the smallest?

(4) Like big cities, the table has some one-way streets along the diagonals. Explain.

(5) Pick any three entries in the interior of the table. Show that the A.E. of each x chosen is smaller than the distance between the entry to the right of x and the entry above x.

(6) How are our "blind" estimators working?

Exercise 7. Repeat Exercise 5, part (3), and Exercise 6 with the supplied function program PI. (CORRCT = 4.D0*DATAN(1.D0))

Exercise 8. Repeat Exercise 7 with a geometric series program GEO written by you. First use $r = -\frac{1}{4}$, then $r = -\frac{9}{10}$. How many of the previous conclusions still hold?

The previous examples were in the class of "totally oscillating" series, the type of alternating series most likely to arise in applications: the terms go to zero in a "regular" way which we shall not try to make more precise. The next alternating series has a small monotone component.

Exercise 9. Repeat Exercise 7, but with the program PIP1, which generates partial sums of the series

$$\pi + 1 = (4 + \tfrac{1}{2}) - (\tfrac{4}{3} - \tfrac{1}{4}) + (\tfrac{4}{5} + \tfrac{1}{8}) - \ldots. \tag{8}$$

Now we are going to see some great stuff; if you like to break the "rules" and still get the right answer, you'll love this. Every Calculus II student knows that the equation

$$\frac{1}{1-r} = 1 + r + r^2 + \ldots \tag{9}$$

is true if and only if $|r| < 1$. And everybody knows that the right side of (9) is useless if, for example, $r = -2$. We then get $1 - 2 + 4 - 8 + \ldots$ which certainly has no connection with $1/(1-(-2)) = 1/3$; everybody knows that, right?

Exercise 10. Apply Euler's method to the series generated by your GEO subprogram. Try $r = -2$, as discussed above.

The results of the previous exercise are explainable only in terms of what is called "analytic continuation." There is nothing bad about the function $1/(1-r)$ at $r = -2$; but the series is just not capable of representing the function at that r value. On Exercise 10, the Euler method was able to take the "deficient" series and convert it to a new series which converged to what the old series "should" have converged to.

Exercise 11. Verify (for at least four or five terms) that, while the initial column was formed from partial sums of $1 - 2 + 4 - \ldots$, the

Euler transform has given, on the top diagonal, the partial sums of $\frac{1}{2} - \frac{1}{4} + \frac{1}{8} - \frac{1}{16} + \ldots$, which converges to $\frac{1}{3}$.

Sometimes, even when the method doesn't work, using it more than once gives success. Doing the table as we have corresponds to the E_1 transform. If we take the top diagonal and use it for a new initial column for another table, this corresponds to the E_2 transform. And so forth, for E_3, E_4, ... ANLYZ is equipped to allow this procedure without difficulty: while the table is being built, ANLYZ saves the last entry on each row, in a vector. The subprogram REDO was written to generate its partial "sums" out of that vector. You thus can call the original series program to build the original table, and then call ANLYZ to use REDO for more tables.

Exercise 12. Set r = -8 in GEO and call ANLYZ to try to sum the series. (Hint: you need not ask for error analysis). Call ANLYZ three more times, with instructions to use the "artificial" series generator REDO instead of GEO. Discuss the four top diagonals thus generated; verify that each top diagonal is the initial column of the next table.

By now, we hope the reader is suitably intrigued; however, the Euler method is the weakest one we are going to study! We certainly would not be prone to use the Euler method on a monotone sequence, for example: averaging consecutive partial sums doesn't sound very attractive in that situation. The next method, due to A.C. Aitken in 1926, will master any series the Euler method will master, and a lot more. And, while it isn't quite as simple as averaging, it is still not difficult to motivate.

Suppose we have a geometric series (either convergent or divergent, r positive or negative, so long as $r \neq 1$).

$$S_1 = 0,\ S_2 = c,\ S_3 = c(1+r),\ S_4 = c(1 + r + r^2),\ \ldots \quad (10)$$

Define S as $c/(1 - r)$. If $|r| > 1$, then we can reasonably say S is the "anti-limit" of the S_n, as will become clear. In either case, by an easy identity,

$$S_n = c(1 + \ldots + r^{n-2}) = \frac{c - cr^{n-1}}{1 - r} =$$

$$S - \frac{cr^{n-1}}{1 - r} = S + pr^{n-1}, \quad (11)$$

where p is independent of n and is given by

$$p = \frac{-c}{(1 - r)}. \quad (12)$$

Thus, for any n,

$$S_{n+1} - S = pr^n = r(pr^{n-1}) = r(S_n - S). \quad (13)$$

Regardless of convergence or divergence, we see that there is a relationship between the behavior of the S_n and the special number S: the difference between the partial sum and S is multiplied by r in moving to the next partial sum.

Now let's assume we have an ordered triplet of numbers A, B, and C which we know are consecutive partial sums of some geometric series, with r unknown. Suppose also that we know only that there is <u>some</u> n such that $A = S_n$, $B = S_{n+1}$, and $C = S_{n+2}$, but we don't know n. Assuming that we do have "some" S_n, S_{n+1}, S_{n+2}, we claim it is easy to use (13) to set

up an equation where r does not occur, n is unimportant, and S is the only "unknown."

Exercise 13. Use (13) (twice) to set up an equation which can be solved for S, in terms of S_n, S_{n+1}, and S_{n+2}. Carry out the solution process.

Exercise 14. Use your answer to Exercise 13 to show that the formula gives the same answer if S_n and S_{n+2} are exchanged, thus reversing the assumed "time" sequence of the numbers.

For the next exercise, we need some notation. Given any sequence $A_1, A_2, \ldots$, let

$$\Delta A_n = A_{n+1} - A_n$$

$$\Delta^2 A_n = \Delta(\Delta A_n) = \Delta A_{n+1} - \Delta A_n = (A_{n+2} - A_{n+1}) - (A_{n+1} - A_n) \tag{14}$$

Exercise 15. Show that your answer to exercise (13) can be written as the following. (Do the four parts in any order.)

$$S = S_{n+2} - \frac{(\Delta S_{n+1})^2}{\Delta^2 S_n} \tag{15}$$

$$= S_{n+2} - \frac{(S_{n+2} - S_{n+1})^2}{(S_{n+2} - S_{n+1}) - (S_{n+1} - S_n)} \tag{16}$$

$$= S_n - \frac{(S_{n+1} - S_n)^2}{(S_{n+2} - S_{n+1}) - (S_{n+1} - S_n)}. \tag{17}$$

$$= S_n - \frac{(\Delta S_n)^2}{\Delta^2 S_n}. \tag{18}$$

You can now see why this is called Aitken's "delta-square" process. The formulas you will want to use in the subroutine are (16) and/or (17).

Actually, all of the results can also be obtained from a different viewpoint, with no reference to geometric series. Suppose we have any sequence A_1, A_2, A_3, . . . and there is a number A and a non-zero number x such, that, for all large n

$$\frac{A_{n+1} - A}{A_n - A} \doteq x \tag{19}$$

Such a sequence is said to "converge linearly" to A, if $|x| < 1$ and we thus obtain convergence. Of course, "linear convergence" is really identical with the errors behaving approximately like the errors of a geometric series, at least when n is large. A geometric series might be said to converge in a "perfectly" linear way, since no "approximation" is needed in (19).

Exercise 16. Explain why, if we have linear convergence to S, then all the previous results (beginning with number 13) hold, except that "=" is replaced by "$\doteq$".

Aitken extrapolation, like Euler's method, has its distinctive assumption: namely, given any ordered triplet, it assumes the sequence is behaving "perfectly" linearly, and uses (15) - (18) to calculate the implied limit. Generally speaking, this assumption will not be exactly true; but if it is close to being true as n increases, then the Aitken sequences generated from the successive triplets will converge more rapidly to the limit S then the S_n do. We can then use the method on the original Aitken sequence, generating a third column, etc. The Δ^2 table

will have a slightly different form than the Euler table, since we need three numbers, instead of two, to use the algorithm. See Figure 19, where each entry is to be generated from the entry on its left and the two entries above that. We thus will need to have the two previous rows available in computing a new row. In the Euler method, recall, we only needed one row of entries.

$$
\begin{array}{llll}
S_1 & & & \\
S_2 & & & \\
S_3 & T_1 & & \\
S_4 & T_2 & & \\
S_5 & T_3 & U_1 & \\
S_6 & T_4 & U_2 & \\
S_7 & T_5 & U_3 & V_1 \\
S_8 & T_6 & U_4 & V_2
\end{array}
$$

Figure 19. Configuration of the Aitken Table

Exercise 17. Use the result of Exercise 14 to show that if we turn the first column of Figure 19 upside down, thus running a finite portion of the sequence "backward", all of the other columns are turned upside down also.

Exercise 18. Explain why allowing any non-zero "x" in (19), as Aitken does, is a less restrictive condition than the Euler method assumes.

In view of the derivation of the Δ^2 method, the next exercise should not be surprising; we essentially designed the method to work "perfectly" on geometric series, didn't we?

Exercise 19. Let $c(1 + r + r^2 + \ldots)$ be any geometric series, with $r \neq 1$, so that $A_n = c(1 + r + \ldots + r^n)$. Show the result of applying the Aitken transformation to any three consecutive partial sums A_n, A_{n+1}, A_{n+2} is $c/(1 - r)$. Now verify this by using Aitken on any three partial sums of $0 + 1 + 2 + 4 + 8 + 16 + \ldots$!

The next exercise shows that if we could be sure the error in A_n was exactly the type assumed by Richardson extropolation (only one power of h), then we would not even need to know the power of h involved, though using the Richardson formula would require it. We could instead just use the Aitken formula on any three consecutive approximations and the limit would be attained exactly.

Exercise 20. Assume that

$$\begin{aligned} A_n &= A + c(4h)^p \\ A_{n+1} &= A + c(2h)^p \\ A_{n+2} &= A + ch^p. \end{aligned} \tag{20}$$

Then the usual Richardson steps give

$$A = \frac{2^p A_{n+2} - A_{n+1}}{2^p - 1} = \frac{2^p A_{n+1} - A_n}{2^p - 1}; \tag{21}$$

but assume we don't have immediate knowledge of p. Show that, in fact,

(21) implies that

$$2^p = \frac{A_{n+1} - A_n}{A_{n+2} - A_{n+1}} \quad . \tag{22}$$

Further, substituting (22) in (21) and simplifying leads to

$$A = \frac{A_n A_{n+2} - A_{n+1}^2}{(A_{n+2} - A_{n+1}) - (A_{n+1} - A_n)} \tag{23}$$

which we could have done without ever worrying about p!

Exercise 21. Show that we may often verify linear convergence on the basis of the partial sums alone: if

$$\frac{S_{n+2} - S}{S_{n+1} - S} \longrightarrow r, \text{ with } r \neq 1, \ r \neq 0, \tag{24}$$

then

$$\frac{S_{n+2} - S_{n+1}}{S_{n+1} - S_n} \longrightarrow r . \tag{25}$$

Hint: rewrite (25) as

$$\frac{(S_{n+2} - S) - (S_{n+1} - S)}{(S_{n+1} - S) - (S_n - S)}; \tag{26}$$

now divide everything by $(S_{n+1} - S)$ and let $n \to +\infty$.

Exercise 22. Write a subroutine to calculate and store a new row of the Aitken table, given the row number, a new partial sum, the previous two

rows of the table, and the upper limit specified by the user for the number of columns to be filled. Note that the routine must actually update two rows on each call; and it should also return the number of entries in the row. See ANLYZ for the exact structure required. Use the top structure for METHOD shown, since METHOD is to be 2 for this routine. Ask your teacher if local vectors are allowed.

Exercise 23. Use your routine to test performance on the PI subprogram. Where are the best answers on the rows?

Exercise 24. Test your routine with the LN1PX program to try to find ln(17). (So let x = 16). How do the columns behave differently from the diagonals? This is a very difficult series, diverging so rapidly that the Euler method is able to bring it under control only after doing four tables.

The next problem is wilder still. The Euler method is never "quite" able to produce a convergent series out of it. The way we "show" what the appropriate value of the series "should" be is really quite illegal; but you will probably find it fairly plausible. Define f(x) by

$$f(x) = \int_0^\infty \frac{e^{-w}}{1+xw}\, dw . \tag{27}$$

Euler was able to show that f(x) can be rewritten as a finite integral,

$$f(x) = \frac{1}{x} e^{1/x} \int_0^\infty \frac{e^{-(1/t)}}{t}\, dt. \tag{28}$$

For a given x, (28) can be evaluated by the trapezoidal rule to any desired accuracy. The true value of f(1) is .59634736.... Everything through this point is legitimate. But now Euler expanded the 1/(1+xw)

in (27) as a geometric series; this ignores the fact that xw is varying up to ∞. Oh, well . . .

$$f(x) = \int_0^\infty e^{-w}(1 - xw + x^2w^2 - x^3w^3 + \ldots)dw. \qquad (29)$$

The next step demands what is called uniform convergence of the series. We do not have any convergence when w is large. But integrate term by term, anyway. This gave Euler

$$f(x) = \int_0^\infty w^0 e^{-w}\,dw - x\int_0^\infty w^1 e^{-w}\,dw + x^2\int_0^\infty w^2 e^{-w}\,dw - \ldots. \qquad (30)$$

The integrals probably don't look familiar, but anybody who has had probability theory knows that

$$\int_o^\infty w^k e^{-w}\,dw = k! \qquad (31)$$

We thus get

$$f(x) \longleftrightarrow 0! - 1!x + 2!x^2 - 3!x^3 + 4!x^4 - \ldots \qquad (32)$$

This series is perfectly horrible: the ratio test shows it doesn't converge for any x besides x = 0. Nevertheless, forging ahead, we set x = 1 in (32) and assert that all this seems to indicate

$$.596347\ldots. \text{ "=" } 0! - 1! + 2! - 3! + 4! - 5! + \ldots. \qquad (33)$$

Exercise 25. Try your Aitken routine on the program WALLIS (the name of the first mathematician associated with (33)). The main program should initialize FACT as 1.DO in COMMON before calling ANLYZ. Again, note the diagonals versus the columns.

By now, you might be thinking the Aitken method can sum anything effectively. Sorry, but no acceleration method is that good. The price of our new-found power is that, while Aitkens' Δ^2 method can handle some

extremely difficult problems, it can also do some strange things when confronted with fairly simple problems which don't match its assumptions. The next problem concerns a series proposed by Lubkin in 1952:

$$1.13197 = \frac{\pi}{4} + \frac{1}{2}\ln(2) = 1 + \frac{1}{2} - \frac{1}{3} - \frac{1}{4} + \frac{1}{5} + \frac{1}{6} - \ldots . \quad (34)$$

Exercise 26. Try the Δ^2 method on the partial sums generated by LUBK. You will see the second column can't decide between 1.13197, .13197, and 2.13197! What happens in the later columns? Use the result of exercise (21) to show that the error ratios are "confusing" the Aitken method by not converging. Note also the pattern on error signs in the first column to show the same thing.

This next example is even more diabolical; it was invented by Shanks in 1955. Aitken "wipes out" any geometric series problem immediately. How about a series which is the sum of <u>two</u> geometric series? Should be easy, right? The series implemented by DGEO is

$$\frac{2}{(1-x)(2-x)} = 2(1 + x + x^2 + \ldots) - 1(1 + \frac{x}{2} + \frac{x^2}{4} + \ldots). \quad (35)$$

<u>Exercise 27</u>. Use the Δ^2 method on DGEO; the main program should set X = 4.D0 and XD2 = 2.D0 in COMMON/DZ/ before calling ANLYZ. The "correct" answer for this divergent series is $\frac{1}{3}$, from (35). What happens in the Δ^2 table?

Shanks did show that the "disaster" in Exercise 26 was going to happen at (only) one point for just about any "double" geometric series. But "disasters" at one point often indicate poor performance elsewhere.

Exercise 28. Try Aitken on DGEO with X = 7.DO and XD2 = 3.5DO. How does this compare with the performance on the difficult ln(17) problem?

Before you do the next exercise, you should note that much of what we did in equations (10) - (13) wouldn't make much sense with r = 1. This would seem to suggest that Aitken might have difficulty with the regrouped pi series

$$\Pi = 4(1 - \frac{1}{3}) + 4(\frac{1}{5} - \frac{1}{7}) + 4(\frac{1}{9} - \frac{1}{11}) + \ldots. \tag{36}$$

This is now what is called a "logarithmically" convergent series, a type that creates difficulties for most accelerators. The ratio of consecutive terms (and errors) goes to +1.

Exercise 29. Try your Δ^2 subroutine on the partial sums generated by PIGRP. Pretty impressive performance?

We have mentioned that the Δ^2 method accelerates "linear" convergence. On the other hand, there are faster kinds of convergence. The next exercise will demonstrate what Aitken gives us if we get greedy and use it on a sequence that already converges as rapidly as anyone could reasonably ask. For example, consider the sequence which follows the pattern

$$X_1 = \frac{1}{2},\ X_2 = \frac{1}{4},\ X_3 = \frac{1}{16},\ \ldots,\ X_n = X_{n-1}^2$$

$$\frac{X_{n+1} - 0}{X_n - 0} = X_n \longrightarrow 0. \tag{37}$$

This is superlinear convergence to 0.

Exercise 30. Write a function subprogram which generates the successive elements in (37), and try Aitken on it. Discuss what happens. Can you give an intuitive reason why, in terms of what the Aitken method assumes?

We arrive finally at one of the most popular algorithms presently available: the epsilon alogrithm. Smith and Ford (1982) did a lot of testing on the various acceleration methods available, and they concluded that a "team" made up of the epsilon algorithm and another recent method ("u," invented in 1973 by Levin) is virtually unbeatable.

Just as the Aitken method assumes the numbers are partial sums in a geometric series, the epsilon algorithm assumes that we are dealing with the sum of several geometric series. The algorithm generates what amount to the "e_k" transforms invented by Shanks. (See his 1955 paper). But Shanks' formulation required the constant use of larger and larger determinant calculations; this made his "invention" quite unattractive for actual use. Fortunately, Wynn in 1956 published an article showing how to calculate the same "e_k" approximations with completely trivial calculations. The epsilon table has a different structure then the earlier tables we have done. See Figure 20. The $e_k(A_i)$ are the approximations; the x's are auxiliary numbers which, while absolutely necessary to the process, will always diverge to $\pm\infty$ when the $e_k(A_i)$ columns are converging nicely!

The reason for the name "epsilon" alorithm is that Wynn denoted the entries of the table by epsilons with both subscripts (column number) and superscripts (diagonal number). We won't do that, however.

The dependencies in Figure 20 are like this: each $e_k(A_i)$ depends on all the A_i between the two diagnols going through $e_k(A_i)$. Thus $e_3(A_i)$

depends on A_1 through A_7. First, you can take $A_1 - A_3$, assume a single geometric series, and calculate the implied limit, $e_1(A_1)$. Thus $e_1(A_1)$, $e_1(A_2)$,... will be the same as the second column of an Aitken table: $e_1 = \Delta^2$. Or you can take $A_1 - A_5$, assume the sum of <u>two</u> geometric series (like DGEO) and e_2 column entries should all be exactly correct. (Round-off may prevent this from quite happening, however.) From the Δ^2 performance on DGEO, e_2 is obviously not "Δ^2 twice". Similarly $e_3(A_1)$ is the limit implied by assuming that $A_1 - A_7$ are partial sums of a "triple" geometric series.

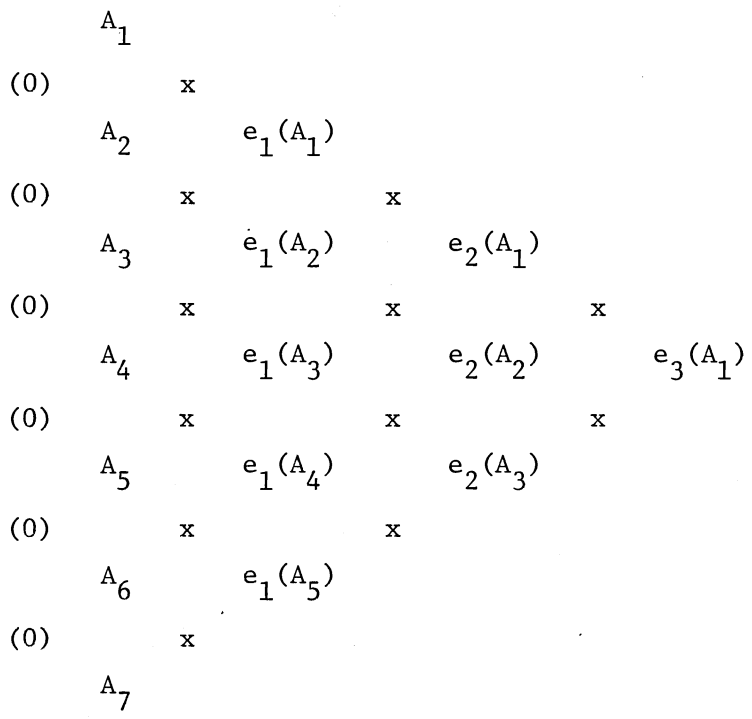

Figure 20. Configuration of the Epsilon Table

But how do we calculate the entries? It's simple, believe it or not! Regard the table as a set of parallelograms, generally called "lozenges" in the literature. On each lozenge, the right side entry can be calculated from the formula

$$\text{Right} = \text{Left} + \frac{1}{\text{Bottom} - \text{Top}} \tag{37}$$

For example, if we replace the x between $e_2(A_1)$ with "y", then

$$e_3(A_1) = e_2(A_2) + \frac{1}{x - y} \tag{38}$$

Thus, any entry can be calculated from the entry just behind it on its own (rising) diagonal and two entries on the previous diagonal. Your epsilon routine will therefore calculate one rising diagonal. You might as well think of it as a row, since we will print the rising diagonals as rows. See Figure 21. (We generally won't print the other columns.) The configuration with the even columns omitted is the same as an Aitken Table, though the entries are different after the e_1 column.

A_1			
A_2			
A_3	$e_1(A_1)$		
A_4	$e_1(A_2)$		
A_5	$e_1(A_3)$	$e_2(A_1)$	
A_6	$e_1(A_4)$	$e_2(A_2)$	
A_7	$e_1(A_5)$	$e_2(A_3)$	$e_3(A_1)$

Figure 21. The Usual Printing of the e_k Table

Exercise 31. Refer to the earlier figure of the "complete" epsilon table and formula (37). Your epsilon subroutine will be given, as usual, "row" (diagonal) number, and the previous "row" (diagonal), along with the new partial sum and a maximum on the number of columns to be used. You will need some local storage; but ask your teacher if local vectors are allowed. Note that the second column calculations use some "virtual" zeros which will not actually be in the row vector. Your subroutine should replace any "ultra-small" denominator by something like 1.D-30, to avoid overflow possibilities. NINROW must also be adjusted before exit.

Exercise 32. Use the epsilon routine and the PI program with ANLYZ. Set METHOD = 3 to see all the columns and then switch to METHOD = 4 to get rid of the auxiliary columns in the display.

Exercise 33. Try the epsilon routine on LUBK, the Lubkin series whose confusing error ratios were too much for the Δ^2 method. Discuss what happens. Why should epsilon handle "competing" components better than Δ^2?

Exercise 34. Try the routine on DGEO with X = 7.0D0 and XD2 = 3.5D0. The e_2 column (and all following ones) should "theoretically" be perfect. What actually happens? What happens as you go more deeply into the e_2 column? Do you have any ideas why? How does the performance compare with the Aitken method's performance?

The next problem comes from Smith and Ford's testing; it gave most methods "fits" because of the irregular sign patterns typical of Fourier series.

$$\frac{\sin(2)}{1} + \frac{\sin(4)}{2} + \frac{\sin(6)}{3} + \frac{\sin(8)}{4} + \ldots = \frac{\pi}{2} - 1. \qquad (39)$$

Exercise 35. Write a program for (39) and apply the epsilon algorithm. Then try the Aitken method on it.

Exercise 36. Write a function subprogram TRAP to find the (step-halving) trapezoidal sum approximations for the integral

$$\frac{\pi}{8} = \int_0^1 \sqrt{x(1-x)}\ dx \qquad (40)$$

This integral does not "yield" to Romberg integration; because of singularities in the derivatives at 0 and 1, the error series is not in powers of h^2. But use the epsilon algorithm on it, and then Aitken's Δ^2 method.

Exercise 37. The epsilon routine can be modified slightly to allow computation of the Aitken table as an alternative: you need a "key" in labelled COMMON which is specified by the main program and available in your epsilon routine. If the request is for "Aitken via modified Epsilon," all that has to be done is to compute all the even columns the way the second one is: whenever an even column is being computed, the left side of the lozenge is thrown away and replaced by zero. Make this slight addition to your epsilon routine and then test against some old Aitken table to see that they agree.

Exercise 38. Show that the Aitken method (slightly) outperforms the epsilon algorithm on WALLIS. Nevertheless, the epsilon algorithm has a bit of an advantage from the standpoint of our being able to analyze what is going on. The algorithm is doing something along the diagonals that

can be connected with the theory of continued fractions. The "nice" fraction f(x) in (28) is woefully misrepresented by the horrible series (32), but it can be represented nicely by a well-behaved "continued" fraction, as explained by Sullivan (1978).

$$f(x) = 0 + \cfrac{1}{1 + \cfrac{x}{1 + \cfrac{x}{1 + \cfrac{2x}{1 + \ \ldots.}}}} \tag{41}$$

The zeroth through the fourth "convergents" of f(x) are

$$0,\ \frac{1}{1},\ \cfrac{1}{1 + \cfrac{x}{1}},\ \cfrac{1}{1 + \cfrac{x}{1 + \cfrac{x}{1}}},\ \cfrac{1}{1 + \cfrac{x}{1 + \cfrac{x}{1 + \cfrac{2x}{1}}}} \tag{42}$$

Verify that with x = 1, these expressions reduce to the numbers on the right end of the first five rows of the e_k table. When the epsilon algorithm succeeds spectacularly, it is because it is using the partial sums of the power series to generate the convergents of the continued fraction representation of f(x). As we have seen, the new representation may converge to f(x) much faster then the power series, or it may converge when the power series diverges. The epsilon algorithm is also intimately connected with what is called the Padé table; but we shall not go into that.

But even the epsilon algorithm has trouble with logarithmic series. One such series, for zeta (2), was used by Smith and Ford:

$$\frac{\pi^2}{6} = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \ \ldots. \tag{43}$$

Exercise 39. Verify that the epsilon algorithm is not able to succeed well on the sums generated by PIGRP and ZETA2.

Wynn designed the "rho" algorithm to make up for the often noted poor performance of the epsilon algorithm on logarithmic series. Your epsilon routine only needs one more small "branch" to allow it to do the "rho" algorithm: when calculating an entry in column K, replace the usual "1" in the numerator of the lozenge formula by "K-1" This changes the "assumption" made by the algorithm. It now assumes the sequence is approaching its limit in the same way that a rational function approaches its horizontal asymptote. This allows handling some monotone series not susceptible to the epsilon algorithm.

Exercise 40. Add the modification to your epsilon routine so that it can perform the rho algorithm on request. Show that it works well on the PIPGRP and ZETA2 programs, but is very poor on the ordinary PI program. However, it also does poorly on the monotone sequence generated by TRAP; so even on monotone sequences it is not the answer to everything.

The reader may have initially hoped we would demonstrate a method which would always work well. Unfortunately, no such method exists. The subject of acceleration methods is very much a current research topic and the use of the methods is still as much an art as a science. It is pretty well agreed that the epsilon algorithm is the strongest method we have discussed, over-all. But for divergent power series and asymptotic series, (e.g., ln(17) and the Wallis series), Smith and Ford found Aitken slightly preferable. The best method we have not discussed, the other one ranked by Smith and Ford at the "top" when paired with ε, is the u

transform of Levin. It has only existed since 1973 and is still the subject of intense study by many mathematicians. Also, the θ algorithm of Brezinski is a powerful hybrid of the epsilon and rho algorithms; it, too, is a research topic of great current interest.

APPENDIX B

COMPUTER PROGRAMS

```
      SUBROUTINE EULER(N,SUMN,ROW,NINROW,MAXKL)
C   THIS SUBROUTINE DOES THE HEART OF THE CALCULATIONS NEEDED FOR
C   A EULER TABLE.  WHEN COMBINED WITH THE SUBROUTINES ANLYZ AND REDO
C   (BOTH SHOWN LATER) AND A USER-WRITTEN SEQUENCE PROGRAM, EULER CAN
C   PRODUCE TABLES FOR THE E1,E2,E3,... TRANSFORMS.  SEE ANLYZ AND REDO
C   TOWARD THE END OF THE APPENDIX, FOR MORE DETAILS.
      DOUBLE PRECISION ROW(1),SUMN,XR,PX
      NINROW=N
      IF(NINROW.GT.MAXKL) NINROW=MAXKL
      NM1=NINROW-1
      XR=SUMN
      IF(NINROW.EQ.1) GO TO 10
      DO 5 J=1,NM1
      PX=XR
      XR=(ROW(J)+PX)/2.DO
      ROW(J)=PX
    5 CONTINUE
   10 ROW(NINROW)=XR
      RETURN
      END
```

```
      SUBROUTINE ROMBRG(A,B,F,KOLMAX,IMAX,METHOD,LAUOPT,TRUE)
C       THIS PACKAGE PERFORMS ROMBERG INTEGRATION, USING STEP-HALVING,
C   THE HARMONIC METHOD, OR THE Q METHOD OF BULIRSCH.  ROMBRG
C   CALLS HALVER, HARMON, OR BULIRQ TO CALCULATE THE TRAPEZOIDAL SUMS IN
C   COLUMN ONE AND GENERATE THE APPROPRIATE N SEQUENCE.  THOSE THREE
C   ROUTINES CALL RSUM, WHICH IMPLEMENTS THE SUMMING PROCEDURE OF RUTI-
C   SHAUSER TO MAXIMIZE ACCURACY OF THE TRAPEZOIDAL SUMS.  RSUM USES
C   THE SUBROUTINE LOCATR TO HELP IT FIND THE X-S WHERE F NEEDS TO BE
C   EVALUATED.  THE FUNCTION SUBPROGRAM FOR F(X) MUST BE SUPPLIED BY THE
C   USER.  SEE REQUIRED FORM BELOW.  THE CORRECT ANSWER IS ASSUMED KNOWN
C   BUT ANY NUMBER COULD BE SUPPLIED FOR TRUE IF THE ANSWER IS NOT KNOWN.
C   IN THAT CASE THE ERRORS PRINTED OUT WILL BE MEANINGLESS.
C       THE ACTUAL EXTRAPOLATIONS ARE DONE BY ROMBRG ITSELF, USING A
C   FORTRAN IMPLEMENTATION OF THE GENERALIZED STEP SEQUENCE FORMULA OF
C   LAURIE.  (TWO ARRANGEMENTS OF THE FORMULA ARE AVAILABLE.)  QUALITY
C   ESTIMATES ARE ALSO CALCULATED FOR THE TABLE ENTRIES.
C
C   PAGE NUMBERS AND FORMULA NUMBERS GIVEN IN THE COMMENTS REFER TO THE
C   ED.D. THESIS BY MARK TOWNSEND.  THE PRECISE BIBLIOGRAPHIC REFERENCES
C   ARE GIVEN IN THAT THESIS.
```

ROMBRG (CONTINUED)

```
C         INPUT PARAMETERS-
C              A      -THE LEFT ENDPOINT OF THE INTEGRATION INTERVAL.
C              B      -THE RIGHT ENDPOINT.
C              F      -THE USER-WRITTEN DOUBLE PRECISION FUNCTION TO BE
C                      INTEGRATED.  IT SHOULD INCLUDE A COMMON/AB/ICOUNT
C                      STATEMENT AND INCREASE ICOUNT BY 1 EACH TIME F IS
C                      CALLED.  ROMBRG WILL INITIALIZE ICOUNT.
C              KOLMAX-THE LAST COLUMN ALLOWED IN THE ENTIRE TABLE.  THE
C                      TRAPEZOIDAL SUMS FORM COLUMN 1.  IF KOLMAX.GT.25,
C                      THE VECTORS WILL NEED TO BE LENGTHENED.
C              IMAX   -THE NUMBER OF ROWS ALLOWED.  THE INITIAL TRAPE-
C                      ZOIDAL SUM IS ROW 1.
C              METHOD-USE +1 FOR BULIRQ, 0 FOR HALVER, -1 FOR HARMON.
C              LAUOPT-USE 0 FOR USE OF THE ORIGINAL FORMULA, 1 FOR THE
C                      RE-ARRANGEMENT RECOMMENDED BY LAURIE AS BEING LESS
C                      SUSCEPTIBLE TO ROUND-OFF ERROR.  (IN THE THESIS
C                      PROBLEMS, THE CHOICE MADE VIRTUALLY NO DIFFERENCE.)
C              TRUE   -THE TRUE VALUE OF THE INTEGRAL.  IF THAT VALUE IS
C                      UNKNOWN, USE ANY VALUE BUT IGNORE THE ERRORS
C                      PRINTED.
C         OUTPUT PARAMETERS-NONE.
C
C   LOCAL VARIABLES IN THE ORDER OF APPEARANCE-
C         ICOUNT-KEEPS COUNT OF FUNCTION EVALUATIONS.
C         KOUT   -UNIT NUMBER FOR OUTPUT.
C         KENDI  -THE COLUMN NUMBER OF THE LAST COLUMN ALLOWED ON ROW I.
C         KENIM1-DITTO, EXCEPT IN REFERENCE TO ROW (I-1).
C         QUAL   -QUAL(K) IS AN ESTIMATE OF THE QUALITY OF THE ENTRY JUST
C                 OBTAINED IN COLUMN K OF ROW I.  IT IS BASED ON THE 3
C                 LAST ENTRIES IN THE PREVIOUS COLUMN.  (QUAL(1) IS A
C                 DUMMY VALUE, THOUGH IT IS PRINTED OUT.)  A QUALITY ESTI-
C                 MATE NEAR 1 INDICATES THE NEW ENTRY IS PROBABLY BETTER
C                 THAN THE TWO NUMBERS USED IN COMPUTING IT.
C         K      -THE COLUMN NUMBER OF THE ENTRY BEING COMPUTED.
C         ROWIM1-HOLDS ALL ENTRIES OF THE (I-1)ST ROW.
C         ROWIM2-HOLDS ALL ENTRIES OF THE (I-2)ND ROW.
C         NI     -THE NUMBER OF SUBDIVISIONS OF (A,B) USED IN COMPUTING
C                 THE ITH TRAPEZOIDAL SUM.  NI IS ADJUSTED BY HARMON, ETC.
C         SI     -THE ITH TRAPEZOIDAL SUM, THE ONE WHICH BEGINS ROW I.
C         ROWI   -HOLDS ALL ENTRIES OF THE ITH ROW.
C         RN2    -HOLDS ALL THE NI**2 VALUES, SINCE MANY ARE NEEDED IN
C                 THE EXTRAPOLATIONS.  IF MORE THAN 50 ROWS ARE USED, RN2
C                 WILL HAVE TO BE LENGTHENED.
C         ERR    -HOLDS ALL THE ERRORS FOR THE ITH ROW.
C         KM1    -K-1
C         KENIM2-SAME AS KENDI, BUT IN REFERENCE TO ROW (I-2).
      DOUBLE PRECISION ROWI(25),ROWIM1(25),ROWIM2(25),ERR(25),RN2(50),
     '     TRUE,SI,F,A,B,DENOM
      REAL QUAL(25)
      COMMON /AB/ICOUNT
      EXTERNAL F
C
C        INITIALIZATIONS AND HEADINGS.
      KOUT=6
      KENDI=1
      KENIM1=0
      QUAL(1)=987654.E25
      ICOUNT=0
C             THE FOLLOWING LOOP IS NEEDED TO PREVENT ABORTION IN THE
C             SHIFT-LOOP AT STATEMENT 90.
      DO 5 K=1,KOLMAX
      ROWIM1(K)=0.D0
      ROWIM2(K)=0.D0
    5 CONTINUE
      WRITE(KOUT,1000) KOLMAX,IMAX,TRUE
      IF(LAUOPT.EQ.1) WRITE(KOUT,1100)
```

ROMBRG (CONTINUED)

```
C        CALCULATIONS AND OUTPUT FOR FOR EACH OF THE IMAX ROWS.
C
      DO 110 I=1,IMAX
      IF(METHOD) 10,20,30
   10 CALL HARMON(A,B,I,NI,F,SI)
      GO TO 40
   20 CALL HALVER(A,B,I,NI,F,SI)
      GO TO 40
   30 CALL BULIRQ(A,B,I,NI,F,SI)
   40 ROWI(1)=SI
      RN2(I)=NI*NI
      ERR(1)=SI-TRUE
C             IF ON ROW 1, WE ARE READY FOR OUTPUT ALREADY.
      IF(I.LT.2) GO TO 80
C
C          DO THE CALCULATIONS FOR ALL THE KENDI ENTRIES IN ROW I.
C
      DO 70 K=2,KENDI
      KM1=K-1
      IF(LAUOPT.EQ.1) GO TO 50
C             THE NEXT FORMULA USED IS NUMBER 38 OR PAGE 63.
      ROWI(K)=(RN2(I)/RN2(I-K+1)*ROWI(KM1) -ROWIM1(KM1)) /
     '        (RN2(I)/RN2(I-K+1)-1.D0)
      GO TO 60
C             THIS OPTIONAL RE-ARRANGEMENT IS RECOMMENDED BY LAURIE AND
C             SHOULD BE LESS SUSCEPTIBLE TO ROUND-OFF ERROR.  (NEW=OLD+
C             A SMALL CORRECTION.)
   50 ROWI(K)=ROWI(KM1) + RN2(I-K+1)/(RN2(I)-RN2(I-K+1))*
     '                    (ROWI(KM1)-ROWIM1(KM1))
   60 ERR(K)=ROWI(K)-TRUE
C             NO QUALITY ESTIMATE IS POSSIBLE UNLESS THERE ARE AT LEAST
C             3 ENTRIES ALREADY COMPUTED IN THE PREVIOUS COLUMN.
      IF(KENIM2.LT.KM1) GO TO 70
C             THE FORMULA USED BELOW IS NUMBER 40 ON PAGE 63.
      DENOM=ROWIM1(KM1)-ROWIM2(KM1)
      IF(DABS(DENOM).LT.1.D-16) DENOM=1.D-16
      QUAL(K)=RN2(I)*(RN2(I-1)-RN2(I-K))/RN2(I-K)/(RN2(I)-RN2(I-K+1))*
     '        (ROWI(KM1)-ROWIM1(KM1))/DENOM

   70 CONTINUE
C
C             OUTPUT FOR ROW I.
   80 WRITE(KOUT,1200) I,NI,ICOUNT
      WRITE(KOUT,1300)(ROWI(K),K=1,KENDI)
      WRITE(KOUT,1400)(ERR(K),K=1,KENDI)
      IF(I.LT.3) GO TO 90
      LASTQ=KENIM2+1
      IF(LASTQ.GT.KOLMAX) LASTQ=KOLMAX
      WRITE(KOUT,1500) (QUAL(K),K=1,LASTQ)
   90 DO 100 K=1,KENDI
      ROWIM2(K)=ROWIM1(K)
      ROWIM1(K)=ROWI(K)
  100 CONTINUE
C        SHIFT IN PREPARATION FOR THE NEXT ROW.
      KENIM2=KENIM1
      KENIM1=KENDI
      KENDI=KENDI+1
      IF(KENDI.GT.KOLMAX) KENDI=KOLMAX
  110 CONTINUE
      RETURN
 1000 FORMAT(17H1COLUMNS ALLOWED,,I3,5X,14H ROWS ALLOWED,,I4,5X,11H TRUE
     ' ANS.=,D25.16)
 1100 FORMAT(33H LAURIE RE-ARRANGEMENT IS IN USE.)
 1200 FORMAT(//4H ROW,I4,20H RESULTS FOLLOW.  N=,I5,5X,21HFUNCTION EVALU
     'ATIONS,,I5)
 1300 FORMAT(/10H   ENTRIES,/4(  1X,D23.16,6X))
 1400 FORMAT(/ 9H   ERRORS,/4(  1X,D23.16,6X))
 1500 FORMAT(/20H   QUALITY ESTIMATES,/4(  1X,E23.6,6X))
      END
```

ROMBRG SATELLITE PROGRAMS

```
      SUBROUTINE HALVER(A,B,I,NI,F,SI)
C  THIS SUBROUTINE SUPERVISES CALCULATION OF ONE NEW TRAPEZOIDAL SUM
C  PER CALL, WITH NO FUNCTION VALUES EVER RECALCULATED ON LATER CALLS.
C  THE NI SEQUENCE IS 1,2,4,8,16,....  THE FORMULAS IMPLEMENTED ARE
C  ON P. 58.  HALVER CALLS RSUM TO DO THE ACTUAL SUMMATIONS OF THE NEW
C  FUNCTION VALUES.  FOR DESCRIPTION OF THE PARAMETERS, SEE ROMBRG
C  ABOVE.
      DOUBLE PRECISION A,B,F,SI,SIM1,FI,HI,RSUM,BMA
C        SOME VALUES NEED TO BE SAVED BETWEEN CALLS.
      COMMON/SAV2/BMA,SIM1,NO3S
      EXTERNAL F
      IF(I.GT.1) GO TO 10
C        INITIALIZATIONS AND FIRST SUM.
      NI=1
      BMA=B-A
      SI=BMA*(F(A)+F(B))/2.D0
C        OUR SUMS WILL BE OF F(X) VALUES, WHERE X=A+J*HI AND J=
C        1,3,5,7,9,11,....
       NO3S=0
      GO TO 20
C        SUMS PAST THE FIRST BEGIN HERE.
   10 NI=2*NI
      HI=BMA/DFLOAT(NI)
C              RSUM WILL FIND THE X VALUES, F(X) VALUES, AND THE NEEDED
C              SUM.
      FI=HI*RSUM(NI,HI,NO3S,F,A)
      SI=SIM1/2.D0 + FI
   20 SIM1=SI
      RETURN
      END
      SUBROUTINE BULIRQ(A,B,I,NI,F,SI)
C  THIS SUBROUTINE SUPERVISES CALCULATION OF ONE NEW TRAPEZOIDAL SUM
C  PER CALL, WITH NO FUNCTION VALUES EVER RECALCULATED ON LATER CALLS.
C  THE BULIRSCH Q SEQUENCE IS USED, WITH NI=1,2,3,4,6,8,12,16,24,32,....
C  THE FORMULAS IMPLEMENTED ARE IN EQUATIONS 43 AND 44 ON PAGE 65.
C  BULIRQ CALLS RSUM TO DO THE ACTUAL SUMMATIONS OF THE NEW FUNCTION
C  VALUES.  FOR DESCRIPTION OF THE PARAMETERS, SEE ROMBRG ABOVE.
C  LOCAL VARIABLES ARE SELF-EXPLANATORY.
      DOUBLE PRECISION A,B,SI,SIM1,SIM2,FI,FIM1,FIM2,FIM3,HI,BMA,RSUM,F
C        SOME OF THE LOCAL VARIABLES MUST BE PRESERVED BETWEEN CALLS.
      COMMON/SAV/ SIM1,SIM2,FI,FIM1,FIM2,FIM3,BMA
      EXTERNAL F
      IF(I.GE.4) GO TO 40
      IF(I-2) 10,20,30
C        ROW 1 INITIALIZATIONS AND SUM
   10 NI=1
      BMA=B-A
      HI=BMA
      SI=HI*(F(A)+F(B))/2.D0
C              THE FOLLOWING VALUES ARE ARBITRARY BUT WILL PREVENT ABORT-
C              TION IN THE 60-BLOCK DURING THE FIRST FEW CALLS.
      SIM1=0.D0
      FI=0.D0
      FIM1=0.D0
      FIM2=0.D0
      GO TO 60
C        ROW 2 SUM
   20 NI=2
      HI=BMA/2.D0
      FI=HI*F((A+B)/2.D0)
```

ROMBRG SATELLITE PROGRAMS (CONTINUED)

```
      SI=SIM1/2.DO +FI
      GO TO 60
C        ROW 3 SUM
   30 NI=3
      HI=BMA/3.DO
      FI=HI*(F(A+HI)+F(A+2.DO*HI))
      SI=SIM2/3.DO +FI
      GO TO 60
   40 IF(I/2*2.NE.I) GO TO 50
C        ROW 4 OR 6 OR 8 OR ... SUM
      NI=2**(I/2)
      HI=BMA/DFLOAT(NI)
C              FOR SUCH A SUM, WE WANT RSUM TO ADD F(X) VALUES AT X=
C              A+J*HI, WHERE J=1,3,5,7,9,...,(NI-1).
      NO3S=0
      FI=HI*RSUM(NI,HI,NO3S,F,A)
      SI=SIM2/2.DO + FI
      GO TO 60
C        ROW 5 OR 7 OR 9 OR ... SUM
   50 NI=3*2**((I-3)/2)
      HI=BMA/DFLOAT(NI)
C              FOR SUCH A SUM, WE WANT RSUM TO ADD F(X) VALUES AT X=
C              A+J*HI, WHERE J=1,5,7,11,13,17,...,(NI-1).
      NO3S=1
      FI=HI*RSUM(NI,HI,NO3S,F,A)
      SI=SIM2/2.DO + FIM3/3.DO + FI
C        SHIFT IN PREPARATION FOR THE NEXT CALL.
   60 SIM2=SIM1
      SIM1=SI
      FIM3=FIM2
      FIM2=FIM1
      FIM1=FI
      RETURN
      END
      SUBROUTINE HARMON(A,B,I,NI,F,SI)
C  THIS SUBROUTINE SUPERVISES CALCULATION OF ONE NEW TRAPEZOIDAL SUM
C  PER CALL, WITH NO FUNCTION VALUES EVER RECALCULATED ON LATER CALLS.
C  THE NI SEQUENCE IS 1,2,3,4,5,....  THE FORMULAS IMPLEMENTED ARE IL-
C  LUSTRATED IN EQUATIONS 41 AND 42 ON PP. 64 AND 65.  HARMON CALLS RSUM
C  TO DO THE ACTUAL SUMMATIONS OF THE NEW FUNCTION VALUES.  IF ONE
C  WISHES TO GO BEYOND THE 51ST TRAPEZOIDAL SUM, FISAV WILL HAVE TO BE
C  LENGTHENED.  FOR DESCRIPTION OF THE PARAMETERS,  SEE ROMBRG ABOVE.
C  LOCAL VARIABLES ARE SELF-EXPLANATORY EXCEPT AS NOTED.
      DOUBLE PRECISION FISAV(25),HI,BMA,SI,A,B,FI,RSUM,F,SUMFI
      EXTERNAL F
C        SOME LOCAL VARIABLES MUST BE PRESERVED BETWEEN CALLS.
      COMMON /SAVE/BMA,FISAV,NO3S
      IF(I.GT.1) GO TO 10
C        ROW 1 INITIALIZATIONS AND SUM
      BMA=B-A
      FISAV(1)=(F(A)+F(B))/2.DO
      SI=BMA*FISAV(1)
C              SET NO3S TO TELL RSUM HARMON IS THE CALLING SUBROUTINE
C              FOR THIS TABLE.
      NO3S=-1
      NI=1
      GO TO 30
C        ROW 2 OR 3 OR 4 OR ... SUM
   10 NI=I
      HI=BMA/DFLOAT(NI)
C              RSUM WILL CALCULATE THE SUM OF F(X) VALUES, WITH X=A+J*HI,
C              WHERE 1.LE.J.LE.(NI-1) AND NI,J ARE RELATIVELY PRIME.
      FI=RSUM(NI,HI,NO3S,F,A)
      SUMFI=FI
      IF(I.LE.25) FISAV(I)=FI
      LASTRY=NI/2
```

ROMBRG SATELLITE PROGRAMS (CONTINUED)

```
C              SUM ALL FI SUMS WHICH CORRESPOND TO DIVISORS OF NI.
      DO 20 J=1,LASTRY
      IF(NI/J*J.EQ.NI) SUMFI=SUMFI+FISAV(J)
   20 CONTINUE
      SI=SUMFI*HI
   30 RETURN
      END
      DOUBLE PRECISION FUNCTION RSUM(NI,HI,NO3S,F,A)
C  THIS SUBROUTINE DIRECTS THE COMPUTATION AND SUMMATION OF THE NEW F(X)
C  VALUES NEEDED FOR THE COMPUTATION OF THE ITH TRAPEZOIDATL SUM IN HAL-
C  VER, HARMON, OR BULIRQ.  THE 3-LEVEL SUMMATION PROCESS OF RUTI-
C  SHAUSER IS AS DESCRIBED ON PP. 58-59.  LOCATR IS CALLED BY RSUM WHEN
C  A NEW X IS NEEDED.  LOCATR RETURNS NSTEPS, WHICH IMPLICITLY LOCATES
C  THE NEW X.  FOR A DESCRIPTION OF THE PARAMETERS OF RSUM, SEE ROMBRG,
C  ABOVE.
      DOUBLE PRECISION F,PPSUM,PSUM,HI,A,XIJ
C        INITIALIZATIONS
      NG=8
      PPSUM=0.D0
      PSUM=0.D0
      RSUM=0.D0
      NG2=NG*NG
      J=1
      LASTEP=NI-1
C        FIND THE XIJ VALUES NEEDED AND SUM THE F(XIJ) VALUES.
   10 CALL LOCATR(NO3S,J,NI,NSTEPS)
      XIJ=A+DFLOAT(NSTEPS)*HI
      PPSUM=PPSUM+F(XIJ)
C        IF J IS NOT A MULTIPLE OF NG, DO NOT UPDATE THE UPPER LEVELS.
      IF(J/NG*NG.NE.J) GO TO 20
C         TIME TO UPDATE THE SECOND LEVEL AND REINITIALIZE THE FIRST.
      PSUM=PSUM+PPSUM
      PPSUM=0.D0
C        IF J IS NOT A MULTIPLE OF NG**2, DO NOT UPDATE THE 3RD LEVEL.
      IF(J/NG2*NG2.NE.J) GO TO 20
C        TIME TO UPDATE THE 3RD LEVEL AND REINITIALIZE THE 2ND.
      RSUM=RSUM+PSUM
      PSUM=0.D0
C        IF WE HAVE NOT YET REACHED THE LAST X NEEDED, GO BACK FOR
C        ANOTTHER ONE, AFTER INCREASING THE INDEX J.
   20 IF(NSTEPS.NE.LASTEP) GO TO 30
C        LAST SWEEP-UP IS NOT ALWAYS NECESSARY BUT DOES NOT EVER HURT.
      RSUM=PPSUM+PSUM+RSUM
      RETURN
   30 J=J+1
C        BACK WE GO FOR THE NEXT F(XIJ).
      GO TO 10
      END
      SUBROUTINE LOCATR(NO3S,J,NI,NSTEPS)
C  THIS SUBROUTINE IS CALLED BY RSUM TO COMPUTE THE NUMBER OF STEPS
C  TO THE JTH NEW X NEEDED FOR THE ITH TRAPEZOIDAL SUM.  THE VALUE
C  OF NO3S IMPLICITLY TELLS LOCATR WHICH SCHEME TO USE, SINCE THERE ARE
C  3 ALTERNATIVE PATTERNS USED IN HALVER, BULIRQ, AND HARMON.  (THE
C  STEPS ARE OF LENGTH HI, THOUGH LOCATR DOES NOT USE THAT INFORMATION.)
      IF(NO3S) 30,20,10
C        WHEN NO3S=+1, WE WANT TO GENERATE NSTEPS=1,5,7,11,13,17,...
C        ON SUCCESSIVE CALLS WITH J=1,2,3,4,5,6,....
   10 NSTEPS=3*J-1
      IF(J/2*2.NE J) NSTEPS=NSTEPS-1
      GO TO 70
C        WHEN NO3S=0, WE WANT TO GENERATE NSTEPS=1,3,5,7,9,11,13,....
C        ON SUCCESSIVE CALLS WITH J=1,2,3,4,5,6,....
   20 NSTEPS=2*J-1
      GO TO 70
C        WHEN NO3S=-1, WE WANT NSTEPS TO TAKE ON ALL VALUES RELATIVELY
C        PRIME TO NI AS WE MAKE SUCCESSIVE CALLS WITH J=1,2,3,4,5,...
```

ROMBRG SATELLITE PROGRAMS (CONTINUED)

```
   30 IF(J.GT.1) GO TO 40
      NSTEPS=1
      GO TO 70
C        WE WILL INCREASE NSTEPS ONE UNIT AT A TIME UNTIL WE FIND THE
C        NEXT NUMBER RELATIVELY PRIME TO NI.
   40 NSTEPS=NSTEPS+1
C        WE USE THE EUCLIDEAN ALGORITHM TO SEE IF GCD(NI,NSTEPS)=1.
      IDIVSR=NSTEPS
      IDIVD=NI
   50 IREM=IDIVD-IDIVD/IDIVSR*IDIVSR
      IF(IREM.EQ.0) GO TO 60
      IDIVD=IDIVSR
      IDIVSR=IREM
      GO TO 50
   60 IF(IDIVSR.EQ.1) GO TO 70
C        BACK TO TRY THE NEXT VALUE FOR NSTEPS.
      GO TO 40
   70 RETURN
      END
```

ROMBERG-RELATED PROGRAMS

```
      SUBROUTINE ERRSER(A,B,NROWS,MAXEX,NTERMS)
C  THIS SUBROUTINE GIVES THE FIRST FEW TERMS AND THE FIRST FEW PARTIAL
C  SUMS FOR THE ERROR SERIES CORRESPONDING TO VARIOUS ENTRIES IN A
C  ROMBERG TABLE FOR A DEFINITE INTEGRAL.  ORIGINAL COEFFICIENTS OF THE
C  COLUMN 0 ERROR SERIES AND ALL MAGNIFICATION FACTORS ARE ALSO PRINTED
C  OUT.  THE USER MUST PROVIDE A ROUTINE DFILL WHICH PLACES THE DERIVA-
C  TIVE DIFFERENCES OF THE EULER-MACLAURIN SERIES INTO THE VECTOR DDIFF.
C  (SEE DDIFF DESCRIPTION BELOW.)
C
C        INPUT PARAMETERS-
C             A      -LEFT ENDPOINT OF THE INTEGRATION INTERVAL.
C             B      -RIGHT ENDPOINT OF THE SAME INTERVAL.
C             NROWS -NUMBER OF ROWS TO BE CONSIDERED IN THE ROMBERG
C                    TABLE.  THE TOP ENTRY IS ROW 1.
C             MAXEX -NUMBER OF EXTRAPOLATION COLUMNS TO BE CONSIDERED.
C                    (THE TRAPEZOIDAL SUMS DO NOT COUNT.)  IF MAXEX.GT.
C                    5, THE FIRST DIMENSION OF MAG WILL HAVE TO BE
C                    RAISED.
C             NTERMS-THE NUMBER OF POWERS OF H**2 TO BE CONSIDERED IN
C                    THE ORIGINAL ERROR SERIES, COLUMN 0.  IF NTERMS.GT.
C                    25, THE DIMENSIONS OF 25 WILL HAVE TO BE RAISED.
C  LOCAL VARIABLES IN ORDER OF APPEARANCE-
C        DDIFF -DDIFF(J) WILL HOLD F2JM1(B)-F2JM1(A), WHERE F2JM1 IS
C               THE (2J-1)TH DERIVATIVE OF F, THE FUNCTION TO BE INTE-
C               GRATED.
```

```
ROMBERG-RELATED PROGRAMS (CONTINUED)
C        ORICOF-ORICOF(J) WILL HOLD THE ORIGINAL COEFFICIENT OF H**2J,
C               IN THE COLUMN O ERROR SERIES FOR I(F)-T(H,F).
C        AJ     -AJ(J) WILL HOLD A-SUB-2J AS IN FORMULA 1 ON PAGE 37 OF
C               THE THESIS.  THESE ARE INITIALIZED WITH A DATA STATE-
C               MENT.
C        KOUT   -UNIT NUMBER FOR OUTPUT.
C        NLEFT  -THE NUMBER OF NON-ZERO TERMS LEFT, OUT OF THE ONES ORI-
C               GINALLY CONSIDERED.
C        K      -EXTRAPOLATION NUMBER CORRESPONDING TO THE COLUMN.
C        J      -TERM NUMBER.  (ONLY EVEN POWERS OF HI ARE COUNTED.)
C        MAG    -MAG(K,M) HOLDS THE TOTAL MAGNIFICATION FACTOR WHICH IS
C                . APPLIED TO THE MTH TERM OF THE ORIGINAL ERROR SERIES
C               IN REMOVING THE FIRST K TERMS VIA ROMBERG INTEGRATION-
C               IE., IN K EXTRAPOLATIONS.  IF M.LE.K, O RESULTS.  ALL
C               ENTRIES IN MAG ARE INITIALLY SET TO O IN A DATA STATE-
C               MENT.
C        I      -THE ROW NUMBER.  TOP ENTRY IS ROW 1.
C        NI     -THE NUMBER OF SUBDIVISIONS OF (A,B) USED IN COMPUTING
C               THE COLUMN O ENTRY ON ROW I.
C        OTERM  -OTERM(J) HOLDS THE JTH TERM FOR THE (ORIGINAL) ERROR
C               SERIES IN COLUMN O OF THE CURRENT ROW.
C        TERM   -TERM(J) HOLDS THE JTH TERM FOR THE ERROR SERIES IN THE
C               CURRENT COLUMN OF THE CURRENT ROW.
C        SUMS   -SUMS(J) HOLDS THE SUM OF THE FIRST J ENTRIES OF TERM.
      DOUBLE PRECISION MAG(5,25),AJ(25),ORICOF(25),TERM(25),OTERM(25),
     '                 DDIFF(25),SUMS(25),HI,A,B,FK
      DATA     AJ/8.333333333D-2,-1.388888889D-3, 3.306878307D-5,
     '-8.267195766D-7,2.087675699D-8,-5.284190138D-10,1.338253653D-11,
     '-3.389680296D-13,8.586062055D-15,-2.174868699D-16,5.509002827D-18,
     '-1.395446469D-19,3.534707041D-21,-8.953517428D-23,2.267952452D-24,
     '-5.744790671D-26,1.455172476D-27,-3.685994942D-29,9.336734257D-31,
     '-2.365022416D-32,5.990671762D-34,-1.517454884D-35,3.843758126D-37,
     '-9.736353073D-39,2.466247044D-40/,MAG /125*0.D0/
C
C        OBTAIN THE COEFFICIENTS OF THE ORIGINAL COLUMN O ERROR SERIES,
C        THROUGH ALL PRESCRIBED TERMS.
C
      CALL DFILL(A,B,DDIFF,NTERMS)
      KOUT=6
      DO 10 J=1,NTERMS
C               SEE THE FORMULA ON PAGE 37.
      ORICOF(J)=-DDIFF(J)*AJ(J)
   10 CONTINUE
      WRITE(KOUT,1000)
      WRITE(KOUT,1600) (ORICOF(J),J=1,NTERMS)
      NLEFT=NTERMS
C
C        FIND THE MAGNIFICATION FACTORS FOR ALL ALLOWED TERMS, FOR ALL
C        ALLOWED COLUMNS. PRINT ALL THE FACTORS OUT.
C
      DO 30 K=1,MAXEX
C               EACH COLUMN HAS ONE LESS ALLOWED TERM THAN THE LAST
C               COLUMN DID.
      NLEFT=NLEFT-1
      FK=4.D0**K
      DO 20 J=1,NLEFT
C               THE FOLLOWING FORMULAS ARE ILLUSTRATED ON PP. 42 F.
      IF(K.EQ.1) MAG(1,1+J)=4.D0*(1.D0-4.D0**J)/3.D0
      IF(K.GT.1) MAG(K,K+J)=MAG(K-1,K+J)*FK*(1.D0-4.D0**J)/(FK-1.D0)
   20 CONTINUE
   30 CONTINUE
      DO 40 K=1,MAXEX
      WRITE(KOUT,1100) K
      WRITE(KOUT,1600)(MAG(K,J),J=1,NTERMS)
   40 CONTINUE
```

ROMBERG-RELATED PROGRAMS (CONTINUED)

```
C
C         FOR EACH ALLOWED ROW, AND EACH ALLOWED COLUMN IN THAT ROW,
C         CALCULATE AS MANY TERMS OF THE ERROR SERIES AS POSSIBLE.
C         ALSO FIND THE CORRESPONDING PARTIAL SUMS.  PRINT IT ALL.
C
      DO 80 I=1,NROWS
      IF(I.EQ.1) NI=1
      IF(I.GT.1) NI=2*NI
      HI=(B-A)/DFLOAT(NI)
      WRITE(KOUT,1200) I,NI,HI
C             CALCULATIONS FOR THE INITIAL COLUMN IN ROW I.
      DO 50 J=1,NTERMS
      OTERM(J)=ORICOF(J)*HI**(2*J)
      IF(J.EQ.1) SUMS(1)=OTERM(1)
      IF(J.GT.1) SUMS(J)=SUMS(J-1) + OTERM(J)
   50 CONTINUE
      K=0
      WRITE(KOUT,1300) K
      WRITE (KOUT,1400)
      WRITE(KOUT,1600) (OTERM(J),J=1,NTERMS)
      WRITE(KOUT,1500)
      WRITE(KOUT,1600) (SUMS(J),J=1,NLEFT)
C             IF WE ARE IN ROW 1, NO EXTRAPOLATIONS.
      IF(I.EQ.1) GO TO 80
C             WE CAN NOT USE ALL COLUMNS IN ALL ROWS.
      MAXK=MIN0(I-1,MAXEX,NTERMS-1)
      NLEFT=NTERMS
C             CALCULATIONS FOR THE EXTRAPOLATION COLUMNS IN ROW I.
      DO 70 K=1,MAXK
      WRITE(KOUT,1300) K
      NLEFT=NLEFT-1
C             CALCULATIONS FOR THE ALLOWED NON-ZERO TERMS FOR THE
C             ERROR SERIES IN COLUMN K OF ROW I.
      DO 60 J=1,NLEFT
      TERM(J)=MAG(K,K+J)*OTERM(K+J)
      IF(J.EQ.1) SUMS(1)=TERM(1)
      IF(J.GT.1) SUMS(J)=SUMS(J-1) +TERM(J)
   60 CONTINUE
      WRITE(KOUT,1400)
      WRITE(KOUT,1600) (TERM(J),J=1,NLEFT)
      WRITE(KOUT,1500)
      WRITE(KOUT,1600) (SUMS(J),J=1,NLEFT)
   70 CONTINUE
   80 CONTINUE
C
C
 1000 FORMAT(91H1THE COEFFICIENTS OF H**2, H**4, H**6,... IN THE EULER-M
     'ACLAURIN SERIES FOR I(F)-T(H,F) ARE)
 1100 FORMAT(// 53H THE CUMULATIVE TERM MAGNIFICATION FACTORS FOR COLUMN
     ',I3,2X,3HARE)
 1200 FORMAT(////52H RESULTS FOR THIS PARTICULAR PROBLEM FOLLOW, FOR ROW
     ',I3,2X,8H,WITH N=,I3,2X,4H,H= ,D15.8)
 1300 FORMAT(//10X,22HRESULTS FOR THE COLUMN,I3,2X,20HERROR SERIES FOLLO
     'W.)
 1400 FORMAT(20X,30HTHE LEADING NON-ZERO TERMS ARE)
 1500 FORMAT(20X,27HTHE SUMS OF THOSE TERMS ARE)
 1600 FORMAT(6(1X,D15.8,5X))
      STOP
      END
```

ROMBERG-RELATED PROGRAMS (CONTINUED)

```
      SUBROUTINE COEFF(NROWS,KOLMAX,RN2,KOUT)
C  THIS PACKAGE IS AN AID IN EXAMINING THE CHARACTERISTICS OF ANY
C  GENERALIZED STEP SEQUENCE ROMBERG INTEGRATION.  THE OUTPUT SHOWS
C  HOW EACH ENTRY IN THE ROMBERG TABLE CAN BE WRITTEN AS A LINEAR COMBI-
C  NATION OF THE TRAPEZOIDAL SUMS IN COLUMN 1.  THE SUM OF THE ABSOLUTE
C  VALUES OF THE NEEDED COEFFICIENTS IS ALSO SHOWN, BECAUSE THAT CAN BE
C  REGARDED AS A MEASURE OF THE SUSCEPTIBILITY OF THE SPECIFIC ENTRY TO
C  ROUND-OFF ERROR.  (SEE P. 70 OF THE THESIS.)  NEWCS FINDS THE COEF-
C  FICIENTS OF EACH ENTRY IN ROW M OF THE ROMBERG TABLE.  THE SUBROUTINE
C  SHOWCS PRODUCES THE DESIRED OUTPUT.  COEFF SUPERVISES THE OPERATIONS.
C        INPUT PARAMETERS-
C              NROWS -NUMBER OF ROWS ALLOWED.  THE TOP ENTRY IN THE TA-
C                     BLE IS ROW 1.
C              KOLMAX-THE MAXIMUM NUMBER OF COLUMNS (INCLUDING THE TRAP-
C                     ZOIDAL SUMS).  IF KOLMAX.GT.13, THE ARRAYS WILL HAVE
C                     TO BE ENLARGED.
C              RN2   -MUST HOLD THE SQUARES OF THE NI, THROUGH
C                     ROW NROWS.  (E.G., FOR Q, 1, 4, 9, 16, 36,...)
C              KOUT  -UNIT NUMBER FOR OUTPUT
C  OTHER VARIABLES IN ORDER OF THEIR APPEARANCE
C        M     -THE CURRENT ROW NUMBER.  THE TOP ENTRY OF THE TABLE IS
C               ROW 1.
C        C1    -C1 HOLDS THE COEFFICIENTS FOR ROW (M-1) WHEN M IS ODD.
C               WHEN M IS EVEN, C1 RECEIVES THE COEFFICIENTS FOR THE
C               ROW M ENTRIES.
C        C2    -SAME AS C1, BUT REVERSE EVEN AND ODD.  THIS ALLOWS US TO
C               AVOID SHIFTING THE MATRIX ENTRIES THEMSELVES.  NEWCS
C               WILL ALWAYS BE WRITING THE ROW M COEFFICIENTS OVER THE
C               ROW (M-2) COEFFICIENTS.
      DOUBLE PRECISION C1(13,13),C2(13,13),RN2(10)
      DO 30 M=1,NROWS
      IF(M/2*2.EQ.M) GO TO 20
      CALL NEWCS(C2,C1,M,RN2,KOLMAX)
C        THERE IS NO ANALYSIS NEEDED ON ROW 1, EVEN THOUGH
C        WE DID NEED THE INITIALIZATIONS BY NEWCS.
      IF(M.EQ.1) GO TO 30
      CALL SHOWCS(C2,M,KOLMAX,KOUT)
      GO TO 30
   20 CALL NEWCS(C1,C2,M,RN2,KOLMAX)
      CALL SHOWCS(C1,M,KOLMAX,KOUT)
   30 CONTINUE
      RETURN
      END
      SUBROUTINE NEWCS(CM,CMM1,M,RN2,KOLMAX)
C  THIS SUBROUTINE IMPLEMENTS FORMULAS 50 AND 51 ON PAGE 71 OF THE THE-
C  SIS.  AT THE END OF THE MTH CALL, NEWCS RETURNS (IN THE TWO-DIMEN-
C  SIONAL ARRAY CM) ALL THE COEFFICIENTS FOR ALL THE ENTRIES IN ROW M.
C        INPUT PARAMETERS-
C             CMM1  -THE JTH ROW OF CMM1 WILL HOLD THE COEFFICIENTS FOR
C                    THE JTH COLUMN ROMBERG ENTRY IN ROW (M-1).  ORDER-
C                    CMM1(J,J) GOES WITH THE TRAP. SUM ON ROW(M-1),
C                    CMM1(J,J-1) GOES WITH THE ONE ON ROW (M-2),...,
C                    CMM1(J,1) GOES WITH THE FIRST ONE WHICH HAS ANY
C                    INFLUENCE ON THE JTH COLUMN OF ROW (M-1).
C                    CMM1(J,J+1) MUST BE 0.  (ALL THESE RESTRICTIONS
C                    ARE WAIVED FOR THE INITIAL CALL, HOWEVER.)
C             M     -NUMBER OF THE ROW FOR WHOSE ENTRIES WE WANT ALL THE COEF-
C                    FICIENTS.
C             RN2   -SEE COEFF.
C             KOLMAX-SEE COEFF.
```

ROMBERG-RELATED PROGRAMS (CONTINUED)

```
C         OUTPUT PARAMETERS-
C              CM     -THE SAME AS CMM1, BUT WITH (M-1) REPLACED BY M.
      DOUBLE PRECISION CM(KOLMAX,KOLMAX),CMM1(KOLMAX,KOLMAX),RN2(1),AMK
C         THE COEFFICIENTS FOR THE FIRST COLUMN ARE EASY.
      CM(1,1)=1.DO
      CM(1,2)=0.DO
      IF(M.EQ.1) RETURN
C         THE USER MAY NOT WANT ALL COLUMNS DONE.
      M1=MINO(M,KOLMAX)
C         FIND THE COEFFICIENTS FOR EACH OF THE COLUMNS IN TURN.
C         WE ALREADY DID THE FIRST COLUMN.
      DO 20 KOL=2,M1
C         AMK COMES FROM FORMULA 51 ON P. 71.
      AMK=RN2(M)/RN2(M-KOL+1)
C         WE SKIRT AROUND THE FACT FORTRAN WILL NOT ALLOW CM(K-1,0)=0 IN
C         THREE MORE LINES.  WE ARE COMPUTING THE COEFFICIENT OF
C         THE OLDEST RELEVANT TRAPEZOIDAL SUM, SEPARATELY.
      CM(KOL,1)=-CMM1(KOL-1,1)/(AMK-1.DO)
C         FIND THE COEFFICIENTS OF THE REMAINING TRAPEZOIDAL SUMS WHICH
C         AFFECT COLUMN KOL OF ROW M.  (FORMULA 51)
      DO 10 J=2,KOL
      CM(KOL,J)=(AMK*CM(KOL-1,J-1)-CMM1(KOL-1,J))/(AMK-1.DO)
   10 CONTINUE
      IF(KOL.EQ.KOLMAX) GO TO 20
C         IF WE DO NOT DO THE NEXT STEP,  WE WILL GET WIPED OUT AT THE
C         END OF THE THE NEXT PASS THROUGH THE 10-LOOP.
      CM(KOL,KOL+1)=0.DO
   20 CONTINUE
      RETURN
      END
      SUBROUTINE SHOWCS(CM,M,KOLMAX,KOUT)
      DOUBLE PRECISION CM(KOLMAX,KOLMAX), SUMN,SUMP
      WRITE(KOUT,1000) M
C         THE USER MAY NOT WANT THE ENTIRE ROW DONE.  THE ONLY COEF-
C         FICIENT OF THE COLUMN 1 ENTRY IS 1.  DO NOT PRINT IT OUT.
      M1=MINO(M,KOLMAX)
C
C         PROCESSING FOR EACH COLUMN, IN TURN.
C
      DO 20 KOL=2,M1
C         THE LEFTMOST COEFFICIENTS WILL GO WITH THE EARLIEST TRAPEZOI-
C         DAL SUMS.
      WRITE(KOUT,2000) KOL,(CM(KOL,L),L=1,KOL)
C         FOR EACH ENTRY IN THE ROW, SUM ITS COEFFICIENTS IN SUCH A WAY
C         AS TO MINIMIZE SUBTRACTIVE CANCELLATION.
      SUMN=0.DO
      SUMP=0.DO
      DO 10 L=1,KOL
      IF(CM(KOL,L).LT.0.DO) SUMN=SUMN+CM(KOL,L)
      IF(CM(KOL,L).GT.0.DO) SUMP=SUMP+CM(KOL,L)
   10 CONTINUE
      SUMP=SUMP-SUMN
      WRITE(KOUT,3000) SUMP
   20 CONTINUE
C
      RETURN
 1000 FORMAT(//25H COEFFICIENT DATA FOR ROW,I3)
 2000 FORMAT(11H0    COLUMN,I3/7(1X,D15.5))
 3000 FORMAT(78X,17HSUM OF MAGNITUDES,3X,D15.5)
      END
```

```
EXTRP PACKAGE, FOR AITKEN, EPSILON, RHO
      SUBROUTINE EXTRP(FN,CORRCT,KOUT,NMAX,KMAX,KEY)
C        THIS SUBROUTINE CALLS SHNK3 ONCE  FOR EACH DESIRED ROW OF THE
C        EPSILON, AITKEN, OR RHO TABLE.  EXTRP ALSO WRITES OUT THE ROWS,
C        ALONG WITH ERROR ANALYSIS.
C
C        INPUT PARAMETERS-
C             FN-       A USER-WRITTEN FUNCTION SUBPROGRAM WHICH COMPUTES
C                       PARTIAL SUMS.  USE EXTERNAL STATEMENT IN THE
C                       CALLING PROGRAM.  SEE REQUIRED STRUCTURE BELOW.
C             CORRCT-   OBVIOUS.
C             KOUT-     UNIT NUMBER FOR OUTPUT.
C             NMAX-     NUMBER OF ROWS IN THE DESIRED TABLE.
C             KMAX-     SEE SHNK3 BELOW.  KMAX MUST BE .LE. 12 WITH PRE-
C                       SENT DIMENSIONS IN EXTRP.
C             KEY-      SEE SHNK3 BELOW.
C        OUTPUT PARAMETERS- NONE
      DOUBLE PRECISION DIAG(25),REEST(25),TRUERE(25),ABERR(25),SNEW,
                       CORRCT,FN,DENOM,HDIFF(25)
      IBEGIN=1
      IF(KEY.EQ.-1) WRITE(KOUT,100)
      IF(KEY.EQ. 0) WRITE(KOUT,200)
      IF(KEY.EQ.+1) WRITE(KOUT,300)
      WRITE(KOUT,400) CORRCT
      DO 20 N=1,NMAX
      SNEW=FN(N, SNEW)
      CALL SHNK3(IBEGIN,KMAX,DIAG,KEY,SNEW,MAXLT,LOCEN)
      IF(LOCEN.LT.3) GO TO 15
      DO 10 KOL=3,LOCEN,2
      DENOM=DIAG(KOL)
      IF(DABS(DENOM).LT.1.D-30) DENOM=1.D-30
      REEST(KOL-2)=DIAG(KOL-2)/DENOM-1.D0
      HDIFF(KOL-2)=DIAG(KOL-2)-DIAG(KOL)
      DENOM=CORRCT
      IF(DABS(CORRCT).LT.1.D-30) DENOM=1.D-30
      TRUERE(KOL-2)=DIAG(KOL-2)/DENOM-1.D0
      ABERR(KOL-2)=DIAG(KOL-2)-CORRCT
   10 CONTINUE
      LOC=LOCEN
      IF(LOCEN/2*2.EQ.LOCEN) LOC=LOC-1
      ABERR(LOC)=DIAG(LOC)-CORRCT
      DENOM=CORRCT
      IF(DABS(DENOM).LT.1.D-30) DENOM=1.D-30
      TRUERE(LOC)=DIAG(LOC)/DENOM-1.D0
   15 WRITE(KOUT,500) N,(DIAG(KOL),KOL=1,LOCEN,2)
      LM2=LOCEN-2
      IF(LM2.LT.1) GO TO 20
      WRITE(KOUT,600)
      WRITE(KOUT,900) (REEST(KOL),KOL=1,LM2,2)
      WRITE(KOUT,700)
      WRITE(KOUT,900) (TRUERE(KOL),KOL=1,LOCEN,2)
      WRITE(KOUT,800)
      WRITE(KOUT,900) (ABERR(KOL),KOL=1,LOCEN,2)
      WRITE(KOUT,1000)
      WRITE(KOUT,900) (HDIFF(KOL),KOL=1,LM2,2)
   20 CONTINUE
      RETURN
  100 FORMAT(1H1,12HAITKEN TABLE)
```

```
EXTRP PACKAGE, FOR AITKEN, EPSILON, RHO (CONTINUED)
  200 FORMAT(1H1,13HEPSILON TABLE)
  300 FORMAT(1H1,9HRHO TABLE)
  400 FORMAT(15H CORRECT LIMIT=,D25.16/4HOROW)
  500 FORMAT(///,I4,3X,D23.16,3(7X,D23.16),/4(7X,D23.16))
  600 FORMAT(1HO,16HREL. ERR. ESTS.-)
  700 FORMAT(1HO,16HTRUE REL. ERRS.-)
  800 FORMAT(1HO,11HABS. ERRS.-)
  900 FORMAT(4(20X,D10.3))
 1000 FORMAT(1HO,12HHOR. DIFFS.-)
      END
      SUBROUTINE SHNK3(IBEGIN,KMAX,DIAG,KEY,SNEW,MAXLT,LOCEN)
C        THIS SUBROUTINE COMPUTES ONE ROW OF THE AITKEN, EPSILON, OR
C        RHO TABLE, DEPENDING ON THE VALUE OF THE INPUT PARAMETER KEY.
C        INPUT PARAMETERS-
C             IBEGIN-  VALUE OF 1 INDICATES THE CALLING PROGRAM WANTS
C                      TO BEGIN A NEW TABLE.
C             KMAX-    ALLOWED NUMBER OF EXTRAPOLATIONS.  DO NOT COUNT
C                      THE SEQUENCE COLUMN OR THE ODD COLUMNS OF THE
C                      EPSILON TABLE.
C             DIAG-    OBVIOUS.
C             KEY-     -1 FOR AITKEN, 0 FOR EPSILON, 1 FOR RHO.
C             SNEW-    OBVIOUS.
C             LOCEN-   AFTER THE FIRST CALL, THE LOCATION OF THE LAST
C                      ENTRY IN THE OLD ROW.
C        OUTPUT PARAMETERS-
C             IBEGIN-  SET TO 0 DURING THE CALL TO BEGIN A TABLE.
C             DIAG-    OBVIOUS.
C             MAXLT-   SET DURING THE FIRST CALL, TO SET UPPER LIMIT ON
C                      THE ROW LENGTH, CORRESPONDING TO KMAX SPECIFIED.
C             LOCEN-   THE LOCATION OF THE LAST ENTRY IN THE NEW ROW.
C                      IT MAY CORRESPOND TO EITHER AN EVEN OR AN ODD
C                      COLUMN IN THE EPSILON TABLE.
C
      DOUBLE PRECISION DIAG(1),XL,XR,XT,XB,DENOM,XNUM,SNEW,X2B
      IF(IBEGIN.EQ.0) GO TO 10
      IBEGIN=0
      MAXLT=2*KMAX+1
      LOCEN=1
      DIAG(1)=SNEW
      RETURN
   10 IF(LOCEN.LT.MAXLT) LOCEN=LOCEN+1
      XB=SNEW
      XL=0.D0
      DO 20 KOL=2,LOCEN
      XT=DIAG(KOL-1)
      XNUM=1.D0
      DENOM=XB-XT
      IF(DABS(DENOM).LT.1.D-30) DENOM=1.D-30
      IF(KOL.GE.3) XL=DIAG(KOL-2)
      IF((KEY.EQ.-1).AND.(KOL/2*2.EQ.KOL))XL=0.D0
      IF(KEY.EQ.+1) XNUM=DFLOAT(KOL-1)
      XR=XL  +XNUM/DENOM
      IF(KOL.GE.3) DIAG(KOL-2)=X2B
      X2B=XB
      XB=XR
   20 CONTINUE
      DIAG(LOCEN)=XR
      DIAG(LOCEN-1)=X2B
      RETURN
      END
```

EPS2, FOR EPSILON WITH THE SPECIAL RULES

```
      SUBROUTINE EPS2(SNEW,LOSDI,ICONT,LASKOL,XLAST,RELER)
C  THIS PROGRAM IS A SLIGHTLY MODIFIED VERSION OF THE ORIGINAL ONE.  FOR
C  THE ORIGINAL EPS2 BY BREZINSKI, SEE HIS ALGORITHMES D-ACCELERATION DE
C  LA CONVERGENCE, ETUDE NUMERIQUE, 1978, PP. 347FF.  EPS2 IMPLEMENTS
C  THE SPECIAL RULES OF CORDELLIER WHEN EXCESSIVE SUBTRACTIVE CANCELLA-
C  TION OCCURS IN THE EPSILON CALCULATIONS.  THIS VERSION AVOIDS THE
C  NEED FOR A FEW STEPS IN THE ORIGINAL EPS2, AND A FEW OTHER STEPS AND
C  VARIABLES ARE ADDED FOR CLARITY.  THE PARTS OF EPS2 WHICH HAVE BEEN
C  CHANGED APPRECIABLY ARE INDICATED IN THE COMMENTS.  (TRANSLATIONS OF
C  ORIGINAL BREZINSKI COMMENTS ARE PRECEDED BY *.)  ONE DIAGONAL OF THE
C  EPSILON TABLE IS DONE PER CALL.
C
C  INPUT PARAMETERS (BREZINSKI VARIABLE NAMES IN PARENTHESES)-
C        SNEW  -THE NEW PARTIAL SUM.  (TERM)
C        LOSDI -THE NUMBER OF LOST DECIMAL DIGITS WHICH WILL BRING THE
C               SPECIAL RULES INTO EFFECT.  (NC)
C        ICONT -SET TO 0 BEFORE BEGINNING A NEW TABLE.  EPS2 WILL RESET
C               TO 1 TO INDICATE A CONTINUING TABLE.  (IKK)
C        LASKOL-LAST COLUMN TO BE ALLOWED IN THE ENTIRE TABLE.  PARTIAL
C               SUMS ARE IN COLUMN 0.  (ICM)
C  OUTPUT PARAMETERS-
C        XLAST -LAST EVEN COLUMN ENTRY AT EXIT TIME.  (RES)
C        RELER -ESTIMATE FOR ABSOLUTE VALUE OF RELATIVE ERROR OF XLAST.
C               BASED ON XLAST AND THE ENTRY OF THE PREVIOUS DIAGONAL
C               WHICH SITS AT THE SAME HEIGHT IN EPSILON TABLE.  (PREC)
C  LOCAL VARIABLES, IN ORDER OF APPEARANCE-
C        MAXKOL-LARGEST COLUMN NUMBER EPS2 CAN ACCOMODATE.  CURRENTLY
C               49, THOUGH BREZINSKI ALLOWED 201.  (NMM PLUS 1)
C        SMALL -USED AS A RESET VALUE FOR ULTRA-SMALL DENOMINATORS. (ZE)
C        KOUT  -UNIT NUMBER FOR OUTPUT.  (NI)
C        MXNSI -MAXIMUM NUMBER OF SINGULARITIES WHICH EPS2 CAN PROCESS
C               AT ONCE.  CURRENTLY 20.  BUT SUCCESSIVE SINGULARITIES IN
C               A COLUMN CAUSE ABORTION.  (NSM)
C        XNXLS -NEXT-TO-LAST EVEN COLUMN ENTRY AT EXIT TIME.  (RA1)
C        ICANT -SET TO 1 BY EPS2 IF ONE OF THREE  ABORTION CONDITIONS
C               ARISES.  SEE MAXKOL AND MXNSI.  (I1,I2,I3)
C        CANCL -THE SMALLEST MAGNITUDE OF (BOTTOM-TOP)/TOP WHICH CORRE-
C               LATES TO A LOSS OF LESS THAN LOSDI DIGITS.  (ANC)
C        INPRS -0 WHEN NO INSTABILITY IS BEING PROCESSED.  1 IF AT LEAST
C               ONE IS STILL IN PROCESSING.  (LIP)
C        NOWE  -SET TO 1 AFTER XSE IS CALCULATED, TO INDICATE THE NEXT
C               ENTRY IS AN E.  0 AT OTHER TIMES.  (ICC)
C        NOWW  -SET TO 1 DURING THE CALCULATION FOR A NEW C, TO INDICATE
C               XLOZL SHOULD BE STORED IN THE W QUEUE AFTER C IS DONE.
C               0 AT OTHER TIMES.  (IAC)
C        NOWN  -SET TO 1 AT THE END OF THE NOWW PROCESSING.  INDICATES
C               THAT AFTER THE NEXT  ENTRY, NE, IS DONE, XLZTO SHOULD BE
C               STORED IN THE N QUEUE.  (IBC)
C        LOCEN -CELL OF DIAG WHICH HOLDS THE LAST ENTRY ON THE DIAGONAL
C               AT EXIT TIME.  (EXCEEDS BREZINSKI M BY 2)
C        DIAG  -HOLDS THE PREVIOUS DIAGONAL AT BEGINNING OF A CALL, AND
C               THE NEW DIAGONAL AT EXIT.  (E)
C        XLZBO -BOTTOM OF THE LOZENGE.  (A1)
C        KOL   -COLUMN NUMBER OF XLOZL.  (IS)
C        IQFRT -POINTS TO THE CELL IN KOLQU WHICH HOLDS THE W COLUMN
C               NUMBER FOR THE OLDEST SINGULARITY STILL IN PROCESSING.
C               WHEN KOL EXCEEDS THAT COLUMN NUMBER BY 1, EPS2 KNOWS
C               SE WAS JUST DONE.  S IS THEN STORED AND NOWE IS SET TO
C               1.  IQFRT ALSO POINTS TO THE LOCATIONS IN WQU AND NQU
C               WHICH HOLD THE CORRESPONDING W AND N VALUES.  IQFRT
C               BEGINS AT 1 AND IS INCREASED BY 1 EACH TIME AN E IS
C               FINISHED.  BUT IF NO MORE SINGULARITIES ARE IN LINE,
C               IQFRT RETURNS TO 1.  (LB)
```

```
EPS2. FOR EPSILON WITH THE SPECIAL RULES (CONTINUED)
C         IQBAC -SAME AS IQFRT, EXCEPT IT LOCATES INFORMATION ABOUT THE
C                NEWEST SINGULARITY.  IQBAC BEGINS AT 0 TO INDICATE NO
C                SINGULARITIES ARE BEING PROCESSED.  WHENEVER A NEW ONE
C                OCCURS, IQBAC IS INCREASED BY 1.  WHEN IQFRT EXCEEDS
C                IQBAC,  IT IS A SIGNAL THAT NO SINGULARITIES ARE IN PRO-
C                CESSING.  INPRS IS SET TO 0, WHICH WILL CAUSE RESETTING
C                THE TWO POINTERS TO THEIR INITIAL VALUES.  (LE)
C         XLZTO -TOP OF THE LOZENGE.  (NOT A BREZINSKI VARIABLE)
C         DENOM -DENOMINATOR OF THE EPSILON FORMULA, BOTTOM-TOP.  (A0)
C         KOLQU -HOLDS THE COLUMN NUMBERS OF THE W VALUES CORRESPONDING
C                TO THE ILL-DETERMINED C VALUES.  THOSE NUMBERS ARE  USED
C                TO TRIGGER RECOGNITION WHEN NEXT ENTRY IS AN E.  (IS1)
C         C,ETC -C, N, W, S, AND E ARE THE LARGE-LOZENGE ENTRIES WHEN THE
C                SPECIAL RULES ARE IN EFFECT.  (NOT BREZINSKI VARIABLES)
C         LAMDA -THE A OF BREZINSKI (1978,P.318).  CORDELLIER USED
C                LAMBDA.  (PA)
C         XLOZR -RIGHT SIDE OF THE LOZENGE.  (A0)
C         XLOZL -LEFT SIDE OF THE LOZENGE.  (BB)
C         WQU   -HOLDS THE W VALUES WHICH WILL BE NEEDED FOR LATER E
C                CALCULATIONS.  (A)
C         NQU   -SAME AS WQU, BUT FOR N VALUES.  (B)
C         X2BAC -THE DIAGONAL ENTRY WHICH WAS CALCULATED JUST BEFORE
C                XLZBO.  (A2)
C  THE FOLLOWING VARIABLES ARE NECESSARY ONLY BECAUSE BREZINSKI CHOSE
C  TO INVOLVE AN ENTRY FROM THE PREVIOUS DIAGONAL IN THE CALCULATIONS
C  FOR RELER.  THE ORIGINAL EPS2 SEEMS TO HAVE SOME ERRORS IN THE RELER
C  CALCULATIONS, AND THE TRANSLATIONS OF THE FOLLOWING TERMS ARE UN-
C  CERTAIN FOR THAT REASON.
C         XPRDIA-THE X ON THE PREVIOUS DIAGONAL WHICH SITS AT THE SAME
C                HEIGHT IN THE EPSILON TABLE AS XLAST.
C         PXLAST-THE XLAST ENTRY ON THE PREVIOUS DIAGONAL.
C         PXNXLS-THE XNXLS ENTRY ON THE PREVIOUS DIAGONAL.
C         LOCPEN-THE LOCATION OF THE END OF THE PREVIOUS DIAGONAL.
C
C *CARDS POSSIBLY TO BE CHANGED BEFORE USE.
C  (THROUGH MXNSI ASSIGNMENT)
C
      DOUBLE PRECISION SNEW,XLOZL,RELER,XLOZR,XLZBO,X2BAC,DIAG(50),
     *CANCL,WQU(20),NQU(20),DENOM,SMALL,C,W,S,N,E,XNXLS,XLAST,XLZTO,
     'LAMDA,XPRDIA,PXNXLS,PXLAST
      DIMENSION KOLQU(20)
C        LABELLED COMMON INSURES THE FOLLOWING VARIABLES WILL STILL
C        HAVE THEIR OLD VALUES ON THE NEXT CALL.
      COMMON /SAVE/DIAG,WQU,NQU,CANCL,SMALL,PXNXLS,PXLAST,KOLQU,ICANT,
     '             INPRS,NOWE,NOWW,NOWN,LOCEN,IQFRT,IQBAC,LOCPEN,KOUT,
     '             MAXKOL
C
C *INITIALIZATIONS
C
      IF(ICONT.EQ.1)GO TO 2
      KOUT=6
      SMALL=1.D-60
      MAXKOL=49
      MXNSI=20
      CANCL=10.D0**(-LOSDI)
      ICANT=0
      XLAST=SNEW
      XNXLS=0.D0
      RELER=1.D0/SMALL
      IF(LASKOL/2*2.NE.LASKOL) LASKOL=LASKOL-1
      ICONT=1
      INPRS=0
      NOWE=0
      NOWW=0
      NOWN=0
      LOCEN=2
      DIAG(1)=0.D0
```

```
EPS2, FOR EPSILON WITH THE SPECIAL RULES (CONTINUED)
C
C *EPSILON CALCULATIONS
C
    2 IF(ICANT.EQ.1)GO TO 20
C
C  INITIAL PROCESSING FOR THE NEW DIAGONAL.  THE POINTERS CAN BE
C  (RE)INITIALIZED ONLY HERE.
C
      XLZBO=SNEW
      KOL=0
      IF(INPRS.EQ.1)GO TO 3
      IQFRT=1
      IQBAC=0
C
C *SPECIAL RULES OF F. CORDELLIER.
C
    3 XLZTO=DIAG(KOL+1)
      DENOM=XLZBO-XLZTO
C
C  CHECK TO SEE IF A NEW INSTABILITY HAS OCCURRED.  FOUR WAYS TO AVOID
C  A YES- TOP VERY SMALL, THE POTENTIAL C IS AT THE END OF A DIAGONAL,
C  TIME TO CALCULATE AN E, OR (BOTTOM-TOP)/TOP IS NOT TOO SMALL.
C
      IF(DABS(XLZTO).LT.SMALL) GO TO 5
      IF(KOL+2.GE.LOCEN.OR.NOWE.EQ.1.OR.DABS(DENOM/XLZTO).GT.
     *   CANCL) GO TO 5
C
C  PERFORM PRELIMINARY PROCESSING FOR THE NEW SINGULARITY.  THE PROGRAM
C  THROUGH INPRS=1 HAS BEEN SHORTENED SOMEWHAT.
C
      IF(INPRS.EQ.0) GO TO 4
      IF(KOL.EQ.KOLQU(IQFRT)) GO TO 18
    4 NOWW=1
      IQBAC=IQBAC+1
      IF(IQBAC.GT.MXNSI)GO TO 17
      KOLQU(IQBAC)=KOL
      INPRS=1
    5 IF(NOWE.EQ.0) GO TO 7
C
C  THE CALCULATION OF E.
C
      XLOZL=DIAG(KOL)
      C=XLOZL
      N=NQU(IQFRT)
      W=WQU(IQFRT)
      LAMDA=((N-W)/(C-W)/(C-N))**2 + ((W-S)/(C-W)/(C-S))**2 -
     *      ((S-N)/(C-N)/(C-S))**2
      E=(N/(C-N)**2 + S/(C-S)**2 - W/(C-W)**2 + LAMDA*C) /
     *  (1.D0/(C-N)**2 + 1.D0/(C-S)**2 - 1.D0/(C-W)**2 + LAMDA)
      XLOZR=E
      NOWE=0
      IQFRT=IQFRT+1
      IF(IQFRT.GT.IQBAC)INPRS=0
      GO TO 12
C
C *NORMAL RULE OF THE EPSILON ALGORITHM
C  (BREZINSKI PUT THIS ONLY ON THE 11-BLOCK, WHICH WE OMIT.)
C
    7 XLOZL=0.D0
      IF(KOL.NE.0)XLOZL=DIAG(KOL)
C        DENOM WAS DONE BACK AT 3 FOR THE CHECK ON CANCELLATION.
      IF(DABS(DENOM).LT.SMALL)DENOM=SMALL
      XLOZR=1.D0/DENOM+XLOZL
C
```

```
EPS2, FOR EPSILON WITH THE SPECIAL RULES (CONTINUED)
C  THE NEXT STATEMENT LETS US AVOID THE NEED FOR THE 11-BLOCK, WHICH
C  DUPLICATED THE 7-BLOCK.
      IF(INPRS.EQ.O) GO TO 12
C
C  ONE OF THE FOLLOWING BLOCKS WILL BE USED, IF WE HAVE JUST CALCULATED
C  A C, AN XNE, OR AN XSE.
C
C  THE 8-BLOCK (WE OMIT 8 BECAUSE NO JUMP IS NOW MADE TO IT) FOLLOWS
C  THE C CALCULATION.
C
      IF(NOWW.EQ.O) GO TO 9
      WQU(IQBAC)=XLOZL
      NOWW=O
      NOWN=1
      GO TO 12
C
C  THE 9-BLOCK FOLLOWS THE XNE CALCULATION.
C
    9 IF(NOWN.EQ.O)GO TO 1O
      NQU(IQBAC)=XLZTO
      NOWN=O
      GO TO 12
C
C  THE 1O-BLOCK FOLLOWS THE XSE CALCULATION.
C
   1O IF(KOL.NE.(KOLQU(IQFRT)+1)) GO TO 12
      S=XLZBO
      NOWE=1
C
C  STORAGE IN DIAG AND SHIFTING OF THE LOZENGE.   RETURN TO 3 TO BEGIN
C  CALCULATION OF THE NEXT DIAGONAL ENTRY, IF APPROPRIATE.
C
   12 IF(KOL.NE.O)DIAG(KOL)=X2BAC
      X2BAC=XLZBO
      XLZBO=XLOZR
      KOL=KOL+1
      IF(KOL+2.LE.LOCEN) GO TO 3
C
C *EXIT OPERATIONS (BREZINSKI HAD THIS ABOVE THE 12-BLOCK.)
C
      DIAG(LOCEN-1)=X2BAC
      DIAG(LOCEN)=XLZBO
C        XLAST MAY OR MAY NOT BE THE LAST ENTRY ON THE DIAGONAL.
      IF(LOCEN.EQ.2) GO TO 16
      IF(LOCEN/2*2.EQ.LOCEN) GO TO 13
      XLAST=DIAG(LOCEN)
      XNXLS=DIAG(LOCEN-2)
      GO TO 14
   13 XLAST=DIAG(LOCEN-1)
      XNXLS=DIAG(LOCEN-3)
C
C  ERROR ESTIMATE AND SAVING XLAST, XNXLS, AND LOCEN FOR THE NEXT CALL.
C
   14 IF(DABS(XLAST).LT.SMALL) GO TO 15
C  THE ENTRY USED FROM THE PREVIOUS DIAGONAL ALTERNATES BETWEEN BEING
C  THE LAST EVEN COLUMN ENTRY OR THE NEXT-TO-LAST ONE, UNTIL WE REACH
C  THE LAST COLUMN ALLOWED BY THE USER.  (PRINT OUT A1 AND A2 IN THE
C  ORIGINAL EPS2, ALONG WITH THE REST OF THE DIAGONAL, TO SEE SOMETHING
C  IS WRONG.)
      XPRDIA=PXLAST
      IF(LOCEN/2*2.EQ.LOCEN.OR.LOCPEN.EQ.LASKOL+1) XPRDIA=PXNXLS
```

```
EPS2. FOR EPSILON WITH THE SPECIAL RULES (CONTINUED)
   RELER=DABS(1.DO-XPRDIA/XLAST)
   GO TO 16
15 RELER=DABS(XLAST)
16 PXLAST=XLAST
   PXNXLS=XNXLS
   LOCPEN=LOCEN
   IF(LOCEN.LT.LASKOL+1) LOCEN=LOCEN+1
   IF(LOCEN.LT.MAXKOL+1) GO TO 20
   WRITE(KOUT,101)
   GO TO 19
17 WRITE(KOUT,102)
   GO TO 19
18 WRITE(KOUT,100)
19 ICANT=1
20 RETURN
100 FORMAT(/34H NON-ISOLATED SINGULARITY IN EPS2,/
  *        23H IMPOSSIBLE TO CONTINUE/)
101 FORMAT(/32H INSUFFICIENT DIMENSIONS IN EPS2/)
102 FORMAT(/31H TOO MANY SINGULARITIES IN EPS2/)
   END
```

THETA

```
      SUBROUTINE THETA(SNEW,ICALL,LASCO,XLAST,RELER)
C  THIS IS A SLIGHTLY MODIFIED VERSION OF THE ORIGINAL.  SEE THE BREZIN-
C  SKI ORIGINAL IN HIS ALGORITHMES D-ACCELERATION DE LA CONVERGENCE,
C  ETUDE NUMERIQUE, 1978, P. 369F.  THETA IMPLEMENTS THE BREZINSKI ALGO-
C  RITHM OF THAT NAME.  THE CHANGES IN THIS VERSION ARE ALMOST PURELY
C  NOTATIONAL, EXCEPT FOR THE ADDITION OF EXPLANATION AT THE VERY BE-
C  GINNING.  (TRANSLATED) BREZINSKI COMMENTS ARE THE ONLY ONES IN THE
C  PROGRAM BODY, EXCEPT FOR THE COMMENTS PRECEDED BY *.
C
C  INPUT PARAMETERS (BREZINSKI NAMES IN PARENTHESES)-
C        SNEW  -THE NEW PARTIAL SUM OR SEQUENCE ELEMENT.  (S)
C        ICALL -COUNTER ON THE CALLS MADE TO THETA DURING A TABLE.  SET
C               TO 0 BEFORE THE FIRST CALL.  ICALL IS UPDATED BY THETA.
C               ICALL IS USED TO SIGNAL WHEN INITIALIZATION IS FINISHED,
C               AND ON WHICH CALLS THE LENGTH OF THE WEAVE CAN INCREASE.
C               (IK AND M PLUS 1)
C        LASCO -THE LAST COLUMN IN THE ENTIRE TABLE WHICH IS ALLOWED BY
C               THE USER.  THE PARTIAL SUMS ARE IN COLUMN 0.  (ICM)
C  OUTPUT PARAMETERS-
C        ICALL -SEE ABOVE.  AT END OF 1ST CALL, ICALL=1, ETC.
C        XLAST -THE LAST EVEN COLUMN ENTRY IN THE WEAVE AT EXIT TIME.
C               (R)
C        RELER -AN ESTIMATE OF THE RELATIVE ERROR OF XLAST, BASED ON
C               XLAST AND THE WEAVE ENTRY TWO COLUMNS BACK.  (PR)
C  OTHER LOCAL VARIABLES, IN ORDER OF APPEARANCE-
C        SMALL -A RESET VALUE FOR ULTRA-SMALL DENOMINATORS.  (ZE)
C        KOUT  -UNIT NUMBER FOR OUTPUT.  (NI)
C        MAXLE -THE MAXIMUM LENGTH WEAVE WHICH THETA CAN ACCOMODATE.
C               CURRENTLY 50, THOUGH BREZINSKI ALLOWED 100.  (MM)
C        WV2BK -THE WEAVE TWO BACK FROM THE WEAVE CURRENTLY BEING COM-
C               PUTED.  (P)
C        ICANT -SET TO 1 BY THETA IF THE WEAVE LENGTHS ACCOMODATED ARE
C               EXCEEDED.  THIS CAUSES ABORTION OF CALCULATIONS.
C        PWEAV -THE PREVIOUS WEAVE, ONE BACK OF THE CURRENT WEAVE.  (D)
C        XLZBO -BOTTOM OF THE LOZENGE.  (NOT A BREZINSKI VARIABLE)
C        XLZTO -TOP OF THE LOZENGE.  (NOT A BREZINSKI VARIABLE)
C        DENOM -DENOMINATOR OF THE EXPRESSION FOR THE THETA ENTRY.  DIF-
C               FERENT FORMS FOR EVEN AND ODD COLUMNS.  (PR)
C        XLOZR -RIGHT SIDE OF THE LOZENGE.  (NOT A BREZINSKI VARIABLE)
C        LENWV -THE LENGTH OF THE CURRENT WEAVE AT EXIT TIME.  (LD)
C        LNPWV -THE LENGTH OF THE PREVIOUS WEAVE.  (L)
C        ISWCH -USED TO SWITCH THE ALGORITHM BACK AND FORTH BETWEEN EVEN
C               AND ODD COLUMN CALCULATIONS.  (K)
C        LOCXR -THE CELL NUMBER OF WEAVE WHICH WILL HOLD XR AFTER XR IS
C               COMPUTED.  (I)
C        XLOZL -LEFT SIDE OF THE LOZENGE.  (R)
C        XSUBB -THE ENTRY RIGHT UNDER XLZBO.  XSUBB IS USED ONLY ON
C               EVEN-COLUMN  CALCULATIONS.  (NOT A BREZINSKI VARIABLE)
C        X2BAC -THE WEAVE ENTRY CALCULATED JUST BEFORE XLZBO  (NOT A
C               BREZINSKI VARIABLE)
C        LOXLS -LOCATION IN WEAVE OF THE LAST EVEN COLUMN ENTRY.  (L)
C        XNXLS -THE NEXT-TO-LAST EVEN COLUMN ENTRY IN THE WEAVE.  (PR)
C
C
C        CARDS POSSIBLY TO BE CHANGED BEFORE USE
C
```

```
THETA (CONTINUED)
      DOUBLE PRECISION SNEW,XLAST,WV2BK( 50),PWEAV( 50),WEAV( 50),RELER,
     'SMALL,XNXLS,DENOM,XLOZR,XLOZL,XLZTO,XLZBO,X2BAC,XSUBB
C        *THE NEXT LINE HAS BEEN ADDED TO INSURE PRESERVATION OF SOME
C        LOCAL VARIABLES BETWEEN CALLS TO THETA.
      COMMON /SAVE/ PWEAV,WV2BK,LENWV,ICANT
      SMALL=1.D-60
      KOUT=6
      MAXLE=100
C
C        INITIALIZATIONS *(3 POSSIBLE  SETS, DEPENDING ON ICALL.)
C
      IF(ICALL.NE.0) GO TO 1
      WV2BK(1)=SNEW
      ICALL=1
      XLAST=SNEW
      ICANT=0
      RELER=1.D0/SMALL
      GO TO 10
C
    1 IF(ICALL.GE.2) GO TO 2
      PWEAV(1)=SNEW
      XLZBO=PWEAV(1)
      XLZTO=WV2BK(1)
      DENOM=XLZBO-XLZTO
      IF(DABS(DENOM).LT.SMALL)DENOM=SMALL
      XLOZR=1.D0/DENOM
      PWEAV(2)=XLOZR
      ICALL=2
      LENWV=2
      XLAST=SNEW
      GO TO 10
C
    2 IF(ICANT.EQ.1)RETURN
      WEAV(1)=SNEW
      ICALL=ICALL+1
      LNPWV=LENWV
      IF(ICALL/3*3.NE.ICALL) LENWV=LENWV+1
      IF(LENWV.GT.(LASCO+1))LENWV=LASCO+1
      IF(LENWV.GT.MAXLE)GO TO 9
      ISWCH=1
C
C        THETA CALCULATION *(BREZINSKI HAD THIS COMMENT ABOVE THE
C        2-BLOCK.)
C
      DO 5 LOCXR=2,LENWV
      GO TO (3,4),ISWCH
C
C        ODD COLUMN
C
    3 XLOZL=0.D0
      IF(LOCXR.NE.2) XLOZL=WV2BK(LOCXR-2)
      XLZTO=PWEAV(LOCXR-1)
      XLZBO=WEAV(LOCXR-1)
      DENOM=XLZBO-XLZTO
      XLOZR=XLOZL+1.D0/DENOM
      WEAV(LOCXR)=XLOZR
      ISWCH=2
      GO TO 5
C
C        EVEN COLUMN
C
    4 XLZTO=WV2BK(LOCXR-1)
```

```
THETA (CONTINUED)
      XLZBO=PWEAV(LOCXR-1)
      XSUBB=WEAV(LOCXR-1)
      XLOZL=WV2BK(LOCXR-2)
      X2BAC=PWEAV(LOCXR-2)
      DENOM=(XSUBB-XLZBO)-(XLZBO-XLZTO)
      IF(DABS(DENOM).LT.SMALL) DENOM=SMALL
      XLOZR=( X2BAC*(XSUBB-XLZBO)-XLOZL*(XLZBO-XLZTO) ) / DENOM
      WEAV(LOCXR)=XLOZR
      ISWCH=1
    5 CONTINUE
C
C        EXIT OPERATIONS
C
      LOXLS=LENWV
      IF(ISWCH.EQ.2)LOXLS=LENWV-1
      XLAST=WEAV(LOXLS)
      IF(LENWV.EQ.2)GO TO 7
      XNXLS=WEAV(LOXLS-2)
      IF(DABS(XLAST).LT.SMALL)GO TO 6
      RELER=DABS((XLAST-XNXLS)/XLAST)
      GO TO 7
    6 RELER=DABS(XLAST)
    7 DO 8 LOCXR=1,LNPWV
      WV2BK(LOCXR)=PWEAV(LOCXR)
    8 PWEAV(LOCXR)=WEAV(LOCXR)
      IF(LENWV.GT.LNPWV) PWEAV(LENWV)=WEAV(LENWV)
      GO TO 10
    9 ICANT=1
      WRITE(KOUT,100)
   10 RETURN
  100 FORMAT(/33H INSUFFICIENT DIMENSIONS IN THETA/)
      END
```

PROGRAMS FOR APPENDIX I

```
      SUBROUTINE ANLYZ(MAXN,MAXKL,METCOD,METHOD,SERIES,CORRCT,IBRF,KOUT)
C        PROGRAMMER- MARK TOWNSEND, MARCH 1983
C        THIS SUBROUTINE ANALYZES THE OUTPUT OF A USER-WRITTEN
C        ACCELERATION SUBROUTINE, OPERATING ON A USER-WRITTEN
C        FUNCTION PROGRAM WHICH COMPUTES PARTIAL SUMS OF A SERIES.
C        SEE THE FORMAT STATEMENTS FOR THE METHODS ACCOMODATED.
C        ANLYZ TAKES CARE OF ALL OUTPUT FORMATTING FOR YOU.  HOORAY.
C
C
C        INPUT PARAMETERS-
C             MAXN-     NUMBER OF ROWS ALLOWED.
C             MAXKL-    MAXIMUM NUMBER OF COLUMNS ALLOWED.  INCLUDE
C                       PARTIAL SUMS COLUMN.  INCLUDE EVEN COLUMNS IN
C                       METHODS 3-6.  ANALYZ ALLOWS MAXKL THRUUGH 20.
C             METCOD-   SEE THE BEGINNING  STATEMENTS AND THEIR FORMATS.
C             METHOD-   NAME OF THE SUBROUTINE DOING THE CALCULATIONS
C                       OF THE TABLE ROWS.  USE EXTERNAL STATEMENT
C                       IN THE CALLING PROGRAM.  SEE STRUCTURE BELOW.
C             SERIES-   NAME OF THE FUNCTION SUBPROGRAM WHICH GENERATES
C                       THE PARTIAL SUMS.  USE EXTERNAL STATEMENT IN THE
C                       CALLING PROGRAM.  SEE STRUCTURE BELOW.
C             CORRCT-   OBVIOUS.
C             IBRF-     1 IF NO ERROR ANALYSIS IS DESIRED.
C             KOUT-     UNIT NUMBER FOR PRINTING.
C        OUTPUT PARAMETERS-NONE
      INTEGER MAXN,MAXKL,METCOD,IBRF,KOUT,IJUMP,L
      REAL HDIFFS(19),ABERR(20) ,RELEST(19),RELTRU(20)
      DOUBLE PRECISION SUMN,SERIES,CORRCT,ROW(20),PRVROW(20),DENOM,
                       SAVTOP(20)
      COMMON /TOP/ SAVTOP
      IF(METCOD.EQ.1) WRITE(KOUT,100)
      IF(METCOD.EQ.2) WRITE(KOUT,200)
      IF(METCOD.EQ.3) WRITE(KOUT,300)
      IF(METCOD.EQ.4) WRITE(KOUT,400)
      IF(METCOD.EQ.5) WRITE(KOUT,500)
      IF(METCOD.EQ.6) WRITE(KOUT,600)
      WRITE(KOUT,700) CORRCT
      IJUMP=1
C        FOR THE LAST THREE METHODS, THE EVEN COLUMNS ARE NOT PRINTED OR
C        USED IN ERROR ANALYSIS.
      IF(METCOD.GT.3) IJUMP=2
      DO 40 N=1,MAXN
C        NOTICE THE REQUIRED STRUCTURE FOR THE SERIES ROUTINE YOU WRITE.
C        ON THE LEFT SIDE, SUMN IS THE NEW PARTIAL SUM.  ON THE RIGHT
C        SIDE, SUMN IS THE PREVIOUS PARTIAL SUM.
      SUMN=SERIES(N,SUMN)
C        NOTICE THE REQUIRED STRUCTURE OF THE ACCELERATION ROUTINE
C        YOU WRITE.  IT MUST UPDATE ROW, PRVROW(IF USED),AND NINROW.
C        THE STRAIGHT AITKEN METHOD REQUIRES THE LAST TWO ROWS FOR
C        CALCULATIONS.
      IF(METCOD.EQ.2) CALL METHOD(N,SUMN,ROW,PRVROW,NINROW,MAXKL)
      IF(METCOD.NE.2) CALL METHOD(N,SUMN,ROW,NINROW,MAXKL)
C        SAVE THE TOP DIAGONAL FOR POSSIBLE RE-EXTRAPOLATION LATER.
      IF(METCOD.LT.3) SAVTOP(N)=ROW(NINROW)
      IF(METCOD.GE.3.AND.NINROW/2*2.EQ.NINROW) SAVTOP(N)=ROW(NINROW-1)
      IF(METCOD.GE.3.AND.NINROW/2*2.NE.NINROW) SAVTOP(N)=ROW(NINROW)
```

PROGRAMS FOR APPENDIX I (CONTINUED)

```
C         TWO WAYS TO SKIP ERROR ANALYSIS- BY REQUEST, OR BECAUSE ALL
C         COLUMNS OF THE EPSILON TABLE ARE PRINTED.
      IF(IBRF.EQ.1.OR.METCOD.EQ.3) GO TO 20
      DO 10 KOL=1,NINROW,IJUMP
      ABERR(KOL)=ROW(KOL)-CORRCT
      IF(KOL+IJUMP.GT.NINROW) GO TO 5
      HDIFFS(KOL)=ROW(KOL+IJUMP)-ROW(KOL)
C         RELATIVE ERROR ANALYSIS IS INAPPROPRIATE IF CORRECT LIMIT IS 0.
    5 IF(CORRCT.EQ.0.D0) GO TO 10
      RELTRU(KOL)=ROW(KOL)/CORRCT-1.D0
      IF(KOL+IJUMP.GT.NINROW) GO TO 10
      DENOM=ROW(KOL+IJUMP)
      IF(DABS(DENOM).LT.1.D-30) DENOM=1.D-30
      RELEST(KOL)=ROW(KOL)/DENOM-1.D0
   10 CONTINUE
C         OUTPUT TIME.
   20 WRITE(KOUT,800) N,(ROW(KOL),KOL=1,NINROW,IJUMP)
      IF(IBRF.EQ.1.OR.METCOD.EQ.3) GO TO 40
      L=NINROW-IJUMP
      IF(CORRCT.EQ.0.D0) GO TO 30
      IF(L.LT.1) GO TO 25
      WRITE(KOUT,900)
      WRITE(KOUT,1300) (RELEST(KOL),KOL=1,L,IJUMP)
   25 WRITE(KOUT,1000)
      WRITE(KOUT,1300) (RELTRU(KOL),KOL=1,NINROW,IJUMP)
   30 WRITE(KOUT,1100)
      WRITE(KOUT,1300) (ABERR(KOL),KOL=1,NINROW,IJUMP)
      IF(L.LT.1) GO TO 40
      WRITE(KOUT,1200)
      WRITE(KOUT,1300) (HDIFFS(KOL),KOL=1,L,IJUMP)
   40 CONTINUE
      RETURN
  100 FORMAT(1H1,11HEULER TABLE)
  200 FORMAT(1H1,12HAITKEN TABLE)
  300 FORMAT(1H1,31HEPSILON TABLE WITH EVEN COLUMNS)
  400 FORMAT(1H1,13HEPSILON TABLE)
  500 FORMAT(1H1,29HAITKEN USING MODIFIED EPSILON)
  600 FORMAT(1H1,9HRHO TABLE)
  700 FORMAT(15H CORRECT LIMIT=,D23.16/4HOROW)
  800 FORMAT(///I4,3X,D23.16,3(7X,D23.16),/4(7X,D23.16))
  900 FORMAT(1HO,16HREL. ERR. ESTS.-)
 1000 FORMAT(1HO,16HTRUE REL. ERRS.-)
 1100 FORMAT(1HO,11HABS. ERRS.-)
 1200 FORMAT(1HO,12HHOR. DIFFS.-)
 1300 FORMAT(4(20X,E10.3))
      END
```

PROGRAMS FOR APPENDIX I (CONTINUED)

```
C  FOR ALL THE FOLLOWING PARTIAL SUM PROGRAMS, ANLYZ SETS SOLD EQUAL
C  TO THE PARTIAL SUM JUST OBTAINED, IMMEDIATELY AFTER THE SUM PROGRAM
C  RETURNS CONTROL.  (SOLD IS NOT A SEPARATE VARIABLE IN ANLYZ, BUT
C  THE CONTEXT THERE MAKES THE SITUATION THERE QUITE CLEAR.)
C  ZERO IS ALWAYS TAKEN AS THE FIRST PARTIAL SUM.


      DOUBLE PRECISION FUNCTION LN1PX(N,SOLD)
C  THIS SUBROUTINE GIVES PARTIAL SUMS OF EQUATION 2 IN THE CHAPTER.  IF
C  X.GT.1, THE SERIES DIVERGES.  BUT THE APPROPRIATE LIMIT IS ALWAYS
C  DLOG(1.DO+X).  THE MAIN PROGRAM MUST PUT X IN COMMON /XLINK/ AND
C  INITIALIZE IT TO WHATEVER VALUE IS DESIRED.
      DOUBLE PRECISION SOLD,SGN,X
      COMMON/XLINK/ X
      IF(N.EQ.1) GO TO 10
      SGN=1.DO
      IF(N/2*2.NE.N) SGN=-SGN
      LN1PX=SOLD + SGN*X**(N-1)/DFLOAT(N-1)
      RETURN
   10 LN1PX=0.DO
      RETURN
      END


      DOUBLE PRECISION FUNCTION PI(N,SOLD)
C  THIS SUBROUTINE GIVES PARTIAL SUMS FOR EQUATION 4 IN THE CHAPTER.
C  CORRECT LIMIT IS 4.DO*DATAN(1.DO).
      DOUBLE PRECISION SOLD,SGN
      IF(N.EQ.1) GO TO 10
      SGN=1.DO
      IF(N/2*2.NE.N) SGN=-SGN
      PI=SOLD+SGN*(4.DO/DFLOAT(2*N-3))
      RETURN
   10 PI=0.DO
      RETURN
      END


      DOUBLE PRECISION FUNCTION PIP1(N,SOLD)
C  THIS SUBROUTINE GIVES PARTIAL SUMS AS IN EQUATION 8 OF THE CHAPTER.
C  THE CORRECT SUM IS 4.DO*DATAN(1.DO) + 1.DO.
      DOUBLE PRECISION SOLD,SGN
      IF(N.EQ.1) GO TO 10
      SGN=1.DO
      IF(N/2*2.NE.N) SGN=-SGN
      PIP1=SOLD+SGN*(4.DO/DFLOAT(2*N-3)+SGN/2.DO**(N-1))
      RETURN
   10 PIP1=0.DO
      RETURN
      END


      DOUBLE PRECISION FUNCTION REDO(N,SOLD)
C  THIS SUBROUTINE DOES NOT REALLY GIVE PARTIAL SUMS.  SOLD IS NOT USED.
C  IN CONSTRUCTING ANY TABLE, ANLYZ SAVES THE LAST EXTRAPOLATION ON EACH
C  ROW, IN THE VECTOR SAVTOP.  REDO SIMPLY RETRIEVES THOSE VALUES TO
C  ALLOW US TO DO REEXTRAPOLATION TABLES.
      DOUBLE PRECISION SOLD,SAVTOP(20)
      COMMON /TOP/ SAVTOP
      REDO=SAVTOP(N)
      RETURN
      END
```

PROGRAMS FOR APPENDIX I (CONTINUED)

```
      DOUBLE PRECISION FUNCTION WALLIS(N,SOLD)
C  THIS SUBROUTINE GIVES PARTIAL SUMS OF EQUATION 33 IN THE CHAPTER.
C  THE SERIES IS DIVERGENT BUT THE APPROPRIATE LIMIT IS .5963473621....
C  THE MAIN PROGRAM SHOULD PLACE FACT IN COMMON AND INITIALIZE IT TO
C  1.DO.
      DOUBLE PRECISION SOLD,SGN,FACT
      COMMON FACT
      IF(N-2) 20,30,10
   10 FACT=FACT*DFLOAT(N-2)
      SGN=1.DO
      IF(N/2*2.NE.N) SGN=-SGN
      WALLIS =SOLD+SGN*FACT
      RETURN
   20 WALLIS=0.DO
      RETURN
   30 WALLIS=1.DO
      RETURN
      END


      DOUBLE PRECISION FUNCTION LUBK(N,SOLD)
C  THIS SUBROUTINE GIVES PARTIAL SUMS OF THE SERIES IN EQUATION 34.
C  THE CORRECT LIMIT IS DATAN(1.DO) +.5DO*DLOG(2.DO).
      DOUBLE PRECISION SGN,SOLD
      COMMON /CT1/SGN
      IF(N.EQ.1) GO TO 10
      IF(N/2*2.EQ.N) SGN=-SGN
      LUBK=SOLD+SGN/DFLOAT(N-1)
      RETURN
   10 SGN=-1.DO
      LUBK=0.DO
      RETURN
      END


      DOUBLE PRECISION FUNCTION DGEO(N,SOLD)
C  THIS SUBPROGRAM GIVES THE PARTIAL SUMS FOR EQUATION 35 IN THE CHAP-
C  TER.  THE CORRECT SUM IS 2.DO/ (1.DO-X)/(2.DO-X).  X AND XD2(=X/2)
C  MUST BE PLACED IN COMMON/DZ/ AND SET BY THE MAIN PROGRAM.
      DOUBLE PRECISION SOLD,X,XD2
      COMMON /DZ/ X,XD2
      IF(N.EQ.1) GO TO 10
      DGEO=SOLD+2.DO*X**(N-2)-XD2**(N-2)
      RETURN
   10 DGEO=0.DO
      RETURN
      END
```

PROGRAMS FOR APPENDIX I (CONTINUED)

```
      DOUBLE PRECISION FUNCTION PIGRP(N,SOLD)
C  THIS SUBROUTINE GIVES PARTIAL SUMS FOR EQUATION 36 IN THE CHAPTER.
C  THE CORRECT LIMIT IS 4.DO*DATAN(1.DO).
      DOUBLE PRECISION SOLD,X
      IF(N.EQ.1) GO TO 10
      X=DFLOAT(4*N)
      PIGRP=SOLD+4.DO*(1.DO/(X-7.DO)-1.DO/(X-5.DO))
      RETURN
   10 PIGRP=0.DO
      RETURN
      END


      DOUBLE PRECISION FUNCTION ZETA2(N,SOLD)
C  THIS SUBROUTINE  GENERATES PARTIAL SUMS OF THE SERIES IN EQUATION
C  43 OF THE CHAPTER.  CORRECT LIMIT IS (4.DO*DATAN(1.DO))**2/6.DO.
      DOUBLE PRECISION SOLD
      IF(N.EQ.1) GO TO 10
      ZETA2=SOLD+1.DO/DFLOAT(N-1)**2
      RETURN
   10 ZETA2=0.DO
      RETURN
      END
```

VITA

Mark Allan Townsend

Candidate for the Degree of

Doctor of Education

Thesis: AN INTRODUCTION TO THE ACCELERATION OF SCALAR SEQUENCES

Major Field: Higher Education

Biographical:

Personal Data: Born in Temple, Texas, June 24, 1943, the son of F. Mark and Naomi Townsend.

Education: Graduated from Norman High School, Norman, Oklahoma, 1961; received the Bachelor of Science degree from Bethany Nazarene College, Bethany, Oklahoma, 1965; graduate study at Purdue University, West Lafayette, Indiana, 1965-1967; received the Master of Arts degree from University of Oklahoma, 1969; graduate study at University of Oklahoma, 1971-1972; completed requirements for Doctor of Education degree at Oklahoma State University, May, 1983.

Professional Experience: Graduate Teaching Assistant, Purdue University, 1965-1967; Graduate Teaching Associate, Oklahoma State University, 1975-1982.

Professional Organizations: Member of the American Mathematical Society and the Mathematical Association of America.