Algorithms and Transformations

1. Numerical Sequences. The integral $I_a^b(f)$, $-\infty < a \le b < \infty$, can be approximated by the trapezoidal rule:

$$\int_a^b f(x)\,dx \approx T_n(f) = h_n \sum_{k=0}^{n}{}'' f\left(a + k h_n\right), \tag{1.1}$$

where $h_n = (b-a)/2$. This rule is derived by partitioning the interval $[a,b]$ into n subintervals $(1 \le n < \infty)$, and using the approximation $I_{a'}^{b'}(f) \approx \dfrac{b'-a'}{2}\left\{f(a') + f(b')\right\}$ over each subinterval $[a', b']$ (see §3.3). This method yields two types of sequences of numbers depending on whether the interval $[a,b]$ is subdivided into $1, 2, 3. \dots$ subintervals or into $1, 2, 4, \dots$ subintervals. We discuss these cases separately.

Case 1. The sequence $\{T_n(f)\}$ is obtained by setting $T_0 = 0$ and $n = 1, 2, \dots$ in formula (1.1). This yields

$$T_0 = 0, \quad T_{n+1} = \sum_{k=0}^{n} E_k, \quad E_n = T_{n+1} - T_n \quad \text{for } n = 1, 2, \dots. \tag{1.2}$$

Example 1.1. Consider $I_1 = \int_0^1 1/(x+1)\,dx$ and $I_2 = \int_0^1 e^{-x}\,dx$, with exact values $I_1 = \ln(2) \approx 0.69314718$ and $I_2 = 1 - e^{-1} \approx 0.63212055$. The values of T_n and E_n are presented in Table 1.1 for $n = 0(1)8$.

Table 1.1. Values of T_n and E_n for $n = 0, 1, \dots, 8$.

	Table 1.1a for I_1		Table 1.1b for I_2	
n	T_n	E_n	T_n	E_n
0	0.000000	0.750000	0.000000	0.683940
1	0.750000	−0.041667	0.683940	−0.038705
2	0.708333	−0.008333	0.645235	−0.007272
3	0.700000	−0.002976	0.637963	−0.002553
4	0.697024	−0.001389	0.635409	−0.00118343
5	0.695635	−0.000758	0.634226	−0.000643
6	0.694877	−0.000458	0.633583	−0.000388
7	0.694420	−0.000298	0.633195	−0.000252
8	0.694122	−0.000204	0.632943	−0.000173

Case 2. The sequence $\{T_n\}$ is obtained by successively halving the interval $[a,b]$, for which we set $T_0 = 0$, and take $n = 1, 2, 4, \dots$ in formula (1.1). This yields

$$\hat{T}_0 = 0, \quad \hat{T}_{n+1} = \sum_{k=0}^{n} \hat{E}_k, \quad \hat{E}_n = \hat{T}_{n+1} - \hat{T}_n \quad \text{for } n = 0, 1, 2, \dots. \tag{1.3}$$

Thus in the notation of formula (1.2) we have $\hat{T}_0 = T_0$, and $\hat{T}_n = T_{2^n-1}$ for $n = 1, 2, \dots$. The values of $\hat{T}_n$ and $\hat{E}_n$ for the integrals I_1 and I_2 are presented in Table 1.2.

Table 1.2. Values of $\hat{T}_n$ and $\hat{E}_n$ for $n = 0, 1, \dots, 8$.

	Table 1.2a for I_1		Table 1.2b for I_2	
n	T_n	E_n	T_n	E_n
0	0.000000	0.750000	0.000000	0.683940
1	0.750000	−0.041667	0.683940	−0.038705
2	0.708333	−0.011310	0.645235	−0.009826
3	0.697024	−0.002902	0.635409	−0.002446
4	0.694122	−0.000731	0.632943	−0.000617
5	0.693391	−0.000183	0.6323266	−0.000154
6	0.693208	−0.000046	0.632172	−0.000039
7	0.693162	−0.000011	0.632133	−0.000010
8	0.693151	−0.000003	0.632124	−0.000002

2. Acceleration Methods. Given a sequence of numbers $\{S_k\}_{k=0}^{\infty}$ and a finite number S, we have two cases: (i) if this sequences converges to S, then S is its limit; (ii) if the sequences $\{S_k\}$ diverges, then we say that S is the divergent limit of this sequence (e.g., S may be the formal sum of of a divergent power series whose partial sums are S_k). The problem we seek is to obtain an estimate of the value of S from a limited number of initial elements of the sequence $\{S_k\}$. We will discuss different approaches to this problem.

2.1. The ϕ-Algorithm. One approach to solve this problem is as follows. Let us assume that the members of the sequence $\{S_k\}$ behave like successive values of a function $\beta(k)$ that is known to have a limit (or divergent limit) S' as $n \to \infty$. Also, let $\beta(k)$ be a function of a finite number i of free parameters, which can be determined by equating $\beta(k)$ to S_k for $k = m, m+1, \dots, m+i-1$, where $m > 0$ is a smaller integer. Then the value S' is accepted as a suitable approximation to S. Specifically, let $\beta(k)$ be taken as a polynomial of degree n in $(\sigma + k)^{-1}$, i.e.,

$$\beta(k) = \phi + \sum_{j=1}^{n} A_j \, (\sigma + k)^{-j}, \tag{2.1}$$

which has $n+1$ quantities, namely ϕ and A_j $(j = 1, 2, \dots, n)$. These $n+1$ quantities can be determined by equating $\beta(k)$ to S_k for $k = m, m+1, \dots, m+n$. The value of ϕ so determined is the estimate for S. This method is due to Wynn (1956a) who also suggested to use the Lagrangian interpolation to determine ϕ. Since this estimate depends on n, which is the number of coefficients A_j, and m which is the point at which we equate the values of

$\beta(k)$ and S_k, we will denote by $\phi_n^{(m)}$ the number just determined. Then by varying n and m we obtain a double sequence of estimates of S.

The problem of obtaining the value of ϕ in formula (2.1) is basically an interpolation problem: We wish to determine this value of the polynomial $\beta(k)$ when $(\sigma+k)^{-1}=0$ under the condition that $\beta(k)$ must assume the values $S_m, S_{m+1}, \dots, S_{m+n}$ when $k = m, m+1, \dots, m+n$. This interpolation problem is solved by using the Aitken-Neville algorithm, which in this case yields a simple recursion involving the numbers $\{\phi_n^{(m)}\}$, and is given by

$$\phi_0^{(m)} = S_m, \quad m = 0, 1, \dots, \tag{2.2}$$

$$\phi_{n+1}^{(m)} = (n+1)^{-1}\left[(\sigma+m+n+1)\phi_n^{(m+1)} - (\sigma+m)\phi_n^{(m)}\right], \qquad n, m = 0, 1, \dots. \tag{2.3}$$

This is known as the *ϕ-algorithm*. Table 2.1 shows a part of the ϕ-array obtained with $\sigma=0$ from the sequence

$$S_m = \sum_{k=0}^{m-1} (k+1)^{-2}, \quad m = 0, 1, \dots, \tag{2.4}$$

where $\lim_{m\to\infty} S_m = \pi^2/6 \approx 1.644934066848$.

Table 2.1.

m	$n=0$	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
0	0.000000	1.000000	1.500000	1.625000	1.643519	1.644965	1.644951
1	1.000000	1.500000	1.625000	1.643519	1.644965	1.644951	
2	1.250000	1.583333	1.638889	1.644676	1.644954		
3	1.361111	1.611111	1.642361	1.644861			
4	1.423611	1.623611	1.643611				
5	1.463611	1.630278					
6	1.491389						

2.2. Romberg Extrapolation. In the ϕ-algorithm theory it is natural to take the points at which $\beta(k)$ is equated to S_k, and that such points can be chosen from a subset of the index set k. Since the subset $k = 1, 2, 4, \dots$ arises in numerical integration where the interval of integration is divided by successive halving of the subintervals into $1, 2, 4, \dots$ intervals, the estimate $\hat\phi_n^{(m)}$ of S is derived from the values of S_k for $k = 2^m, 2^{m+1}, \dots, 2^{m+n}$ with $n = 0, 1, \dots$ and $m = 0, 1, \dots$. For $\sigma = 0$ the recursion for the numbers $\{\hat\phi_n^{(m)}\}$ is given by

$$\hat\phi_0^{(0)} = 0, \quad \hat\phi_0^{(m)} = S_{2^{m-1}} \quad m = 1, , 2, \dots,$$

$$\hat\phi_{n+1}^{(m)} = \left(2^{n+1}\,\hat\phi_n^{(m+1)} - \hat\phi_n^{(m)}\right)\left[2^{n+1}-1\right]^{-1}, \quad n, m = 0, 1, \dots. \tag{2.5}$$

This is known as *Romberg extrapolation*. The numbers $\{\hat\phi_n^{(m)}\}$ are set in array similar to the ϕ-array, and computed recursively. Part of the $\hat{\phi}_n^{(m)}$-array obtained for the sequence

$$\hat{S}_0 = 0, \quad \hat{S}_m = \sum_{k=0}^{2^{m-1}-1} (k+1)^{-2}, \quad m = 1, , 2, \dots , \tag{2.6}$$

is presented in Table 2.2.

Table 2.2.

m	$n=0$	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
0	0.000000	2.000000	1.333333	1.671958	1.642583	1.645018	1.644933
1	1.000000	1.500000	1.629630	1.644419	1.644942	1.644934	
2	1.250000	1.597222	1.642570	1.644910	1.644935		
3	1.426311	1.631233	1.644617	1.644933			
4	1.527422	1.641271	1.644894				
5	1.584347	1.643988					
6	1.614167						

2.3. The ρ-Algorithm. The limit of the rational function of the form

$$\beta(k) = \frac{\sum_{j=0}^{n} p_j\, k^j}{\sum_{j=0}^{n} q_j\, k^j} \tag{2.7}$$

as $k \to \infty$ exists and, therefore, can be used as the basis for extrapolating the limit. The estimates of S in this case are denoted by $\left\{\rho_{2n}^{(m)}\right\}$, where $\rho_{2n}^{(m)}$ is derived from $S_m, S_{m+1}, \dots , S_{m+2n}$. These estimates are constructed by using the Thiele's interpolation method (see Norlund 1937, Ch. 15). The recursion involves the numbers $\left\{\rho_{2n}^{(m)}\right\}$ and is given by

$$\begin{aligned}
&\rho_{-1}^{(m)} = 0, \quad m = 1, 2, \dots , \\
&\rho_{0}^{(m)} = S_m, \quad m = 0, 1, \dots , \\
&\rho_{n+1}^{(m)} = \rho_{n-1}^{(m+1)} + (n+1)\left[\rho_n^{(m+1)} - \rho_n^{(m)}\right]^{-1}, \quad n, m = 0, 1, \dots .
\end{aligned} \tag{2.8}$$

This is known as the ρ-*algorithm* (Wynn 1956b). The numbers $\left\{\rho_{2n}^{(m)}\right\}$ are placed in a two-dimensional array where the superscript m denotes a forward diagonal and the subscript n a column. This array is constructed from left to right as in the preceding tables. Table 2.3

shows a part of the ρ-array for the sequence (2.4).

Table 2.3.

m	$n=0$	$n=1$	$n=2$	$n=3$
1	1.000000	1.666667		
2	1.250000	1.650000	1.644737	
3	1.361111	1.646825	1.644895	1.644936
4	1.423611	1.645833	1.644923	
5	1.463611	1.645429		
6	1.491389			

2.4. The $\hat{\rho}$-Algorithm. We can take the points at which the rational function $\beta(k)$ defined in (2.7) is equated to S_k from the subset $k = 1, 2, 4, \dots$. The estimates $\hat{\rho}_{2n}^{(m)}$ of S are then derived from the values of S_k for $k = 2^m, 2^{m+1}, \dots, 2^{m+2n}, \dots$ with $n = 0, 1, \dots$ and $m = 0, 1, \dots$. The recursion involving the numbers $\hat{\rho}_{2n}^{(m)}$ is given by

$$\begin{aligned}
&\hat{\rho}_{-1}^{(m)} = 0, \quad m = 1, 2, \dots, \\
&\hat{\rho}_{0}^{(0)} = S_0, \quad \hat{\rho}_{0}^{(m)} = S_{2^{m-1}}, \quad m = 1, 2, \dots, \\
&\hat{\rho}_{n+1}^{(m)} = \hat{\rho}_{n-1}^{(m)} + 2^m \left(2^{n+1} - 1\right) \left[\hat{\rho}_{n}^{(m+1)} - \hat{\rho}_{n}^{(m)}\right]^{-1}, \quad n, m = 0, 1, \dots.
\end{aligned} \tag{2.9}$$

The numbers $\left\{\hat{\rho}_{2n}^{(m)}\right\}$ are set in an array similar to the ρ-array and recursively computed as before. Table 2.4 shows part of the $\hat{\rho}_{2n}^{(m)}$-array for the sequence

$$S_0 = 0, \quad S_m = \sum_{k=0}^{2^{m-1}-1} (k+1)^{-2}, \quad m = 1, 2, \dots. \tag{2.10}$$

This algorithm requires very large summation indices. For computational purposes see `rhohat.nb`.

2.5. The ϵ-Algorithm. Another form of the function $\beta(k)$, which is used as a basis for a system of recursions for extrapolated limits, is defined by

$$\beta(k) = S' + \sum_{j=1}^{i} \gamma_j^k \sum_{l=0}^{i_j-1} \alpha_{j,l} k^l, \tag{2.11}$$

where S' is assumed constant, the factors γ_j are nonzero and distinct, $\alpha_{j,i_j-1} \neq 0$ for $j = 1, 2, \dots, i$, and in the case when the set $\{\gamma_j\}$ contains a unit member, say $\gamma_l = 1$, then $\alpha_{l,0} = 0$, which ensures that the representation (2.11) cannot be reduced to another form of $\beta(k)$ containing fewer disposable parameters. The constant S', the exponential terms $\{\gamma_j\}$, the integer distribution $\{i_j\}$ and the coefficients $\{\alpha_{j,l}\}$ can be evaluated by equating $\beta(k)$ to S_k for $k = m, m+1, \dots, m+2n$ when $\sum_{j=1}^{i} t_j = 2n$. Note that the coefficients of the nonhomogeneous linear difference equations $\sum_{k=0}^{n} c_k \, S_{m+k+k'} = \left(\sum_{k=0}^{n} c_k\right) S'$ for $k' = 0, 1, \dots, n$, can be determined, and the zeros of the polynomial $\sum_{k=0}^{n} c_k \, \gamma^k$ are $\{\gamma_j\}$ with multiplicity $\{i_j\}$. Thus, S' which is the formal limit (or actual limit) of $\beta(k)$ as $k \to \infty$ is the only quantity in the representation (2.11) which must be considered since it is a rational function of S_k for $k = m, m+1, \dots, m+2n$. The various values of S' so obtained are denoted by $\left\{\epsilon_{2n}^{(m)}\right\}$ such that $\epsilon_{2n}^{(m)}$ is obtained from S_k, $k = m, m+1, \dots, m+2n$. The sequence $\left\{\epsilon_{2n}^{(m)}\right\}$ is constructed by means of the recursion

$$\begin{aligned} &\epsilon_{-1}^{(m)} = 0, \quad m = 1, 2, \dots; \quad \epsilon_0^{(m)} = S_m \quad m = 0, 1, \dots, \\ &\epsilon_{n+1}^{(m)} = \epsilon_{n-1}^{(m+1)} + \left[\epsilon_n^{(m+1)} - \epsilon_n^{(m)}\right]^{-1} \quad n, m = 0, 1, \dots. \end{aligned} \tag{2.12}$$

This is known as the ϵ-*algorithm* (Wynn 1956c). The numbers $\epsilon_{2n}^{(m)}$ are set in an array and recursively computed as in previous algorithms. Table 2.5 presents the part of the ϵ-array from the sequence

$$S_m = \sum_{k=0}^{m-1} (-1)^k \, (k+1)^{-1}, \quad m = 0, 1, \dots. \tag{2.13}$$

Note that $\lim_{m \to \infty} S_m = 0.69314718056$.

Table 2.5.

m	$n = 0$	$n = 2$	$n = 4$	$n = 6$
1	1.00000	0.66667		
2	0.50000	0.70000	0.69231	
3	0.83333	0.69048	0.69333	0.69312
4	0.58333	0.69444	0.69309	
5	0.78333	0.69242		
6	0.61667			

Note that we cannot freely apply the ϵ-algorithm repeatedly, i.e., we cannot take any progression of numbers from an even order ϵ-array and form a new initial sequence which can again be transformed by means of the ϵ-algorithm. However, there are two kinds of repetitions of the ϵ-algorithm, which are as follows.

2.5.1. The ϵ-numbers $\left\{{}_0\epsilon_n^{(m)}\right\}$ are produced from a prescribed sequence $\{S_m\}$ and then successive sets of numbers $\left\{{}_i\epsilon_n^{(m)}\right\}$ are computed by taking the step-like sequence

$${}_{i-1}\epsilon_0^{(0)}, \; {}_{i-1}\epsilon_0^{(1)}, \; {}_{i-1}\epsilon_2^{(0)}, \; {}_{i-1}\epsilon_2^{(1)}, \; {}_{i-1}\epsilon_4^{(0)}, \; \dots$$

as the initial sequence

$${}_i\epsilon_0^{(0)}, \; {}_i\epsilon_0^{(1)}, \; {}_i\epsilon_0^{(2)}, \; {}_i\epsilon_0^{(3)}, \; {}_i\epsilon_0^{(4)}, \; \dots,$$

from which the next ϵ-array is constructed for $i = 1, 2, \ldots$. This is known as the *corresponding repeated application* of the ϵ-algorithm (Wynn 1956d). This kind of repetition is applied to the sequence (2.13) and Table 2.6 shows the numbers ${}_i\epsilon_0^{(m)}$ for $i = 0, 1, 2$ and $m = 0, 1, \ldots, 8$.

Table 2.6.

m	$n = 0$	$n = 1$	$n = 2$
0	0.0000000000	0.0000000000	0.0000000000
1	1.0000000000	1.0000000000	1.0000000000
2	0.5000000000	0.6666666667	0.7500000000
3	0.8333333433	0.7000000036	0.6969696999
4	0.5833333433	0.6923076987	0.6931046866
5	0.7833333462	0.6933333425	0.6931653724
6	0.6166666746	0.6931217032	0.6931517595
7	0.7595238239	0.6931524654	0.6931473196
8	0.6345238239	0.6931464286	0.6931471917

As another example, we consider the sequence

$$S_m = \sum_{k=0}^{m-1} \binom{\alpha}{k} (k+1)^{-1}, \quad m = 0, 1, \ldots. \tag{2.14}$$

Note that $\lim_{m\to\infty} S_m = \dfrac{2^{\alpha+1} - 1}{\alpha + 1}$, which for $\alpha = -0.5$ has the value 0.8284271247461903. The numbers ${}_i\epsilon_0^{(m)}$ for $i = 0, 1, 2$ and $m = 0, 1, \ldots, 6$ for the sequence (2.14) are presented in Table 2.7.

Table 2.7.

m	$n = 0$	$n = 1$	$n = 2$
0	0.0000000000	0.0000000000	0.0000000000
1	1.0000000000	1.0000000000	1.0000000000
2	0.7500000000	0.8000000000	0.8333333333
3	0.8750000000	0.8333333333	0.8285714286
4	0.7968750000	0.8275862069	0.8284263959
5	0.8515625000	0.8285714286	0.8284271284
6	0.8105478750	0.8284023669	0.8284271247

3. Generalized Euler Transformation. (Hartree 1952, Ch. 12) We assume that the terms $\{u_k\}$ of the infinite series

$$\sum_{k=0}^{\infty} u_k \tag{3.1}$$

behaves like a geometric progression with ratio t so that we write $u_m = t^m\, v_m$ for $m = 0, 1, \ldots$, where the numbers $\{v_m\}$ are approximately constant. If Δ is the difference operator with respect to m, then the *generalized Euler transformation* is defined by

$$\sum_{k=0}^{\infty} u_k \longrightarrow \frac{1}{1-t} \sum_{k=0}^{\infty} \left(\frac{t}{1-t}\right)^k \Delta^k v_0. \tag{3.2}$$

It is expected that the successive differences $\{\Delta^k v_0\}$ will decrease rapidly in magnitude, and that the transformed series (3.1) converge numerically faster than the series from which it is derived.

In its delayed form the generalized Euler transformation is

$$\sum_{k=0}^{\infty} u_k \longrightarrow \sum_{k=0}^{m-1} u_k + \frac{t^m}{1-t} \sum_{k=0}^{\infty} \left(\frac{t}{1-t}\right)^k \Delta^k v_m. \tag{3.3}$$

Its partial sums, denoted by $S_n^{(m)}$ and defined by

$$S_n^{(m)} = \sum_{k=0}^{m-1} u_k + \frac{t^m}{1-t} \sum_{k=0}^{n-1} \left(\frac{t}{1-t}\right)^k \Delta^k v_m, \tag{3.4}$$

form a double sequence of approximations to the sum (or formal sum) of the original series (3.1). These sums are placed in a two dimensional array similar to the ϕ-array. The numbers $\{S_0^{(m)}\}$ are the successive partial sums of the original series (3.1), whereas the numbers $\{S_n^{(0)}\}$ are the successive partial sums of the transformed series (3.2). The numbers $\{S_n^{(m)}\}$ are constructed from the initial values $S_0^{(m)} = \sum_{k=0}^{m-1} u_k$ for $m = 0, 1, \ldots$ by using the recursion relations

$$S_{n+1}^{(m)} = p\, S_n^{(m+1)} + q\, S_n^{(m)}, \quad \text{for } n, m = 0, 1, \ldots, \tag{3.5}$$

where $p = 1/(1-t)$ and $q = -t/(1-t)$. This transformation is applied to the series whose terms are $u = (k+1)^{-1}\, t^k$, $k = 0, 1, \ldots$, with $t = -2$. Note that this series is the (divergent) expansion of the function $f(t) = -t^{-1} \ln(1+t)$, with the value $f(-2) = 0.549306$. In formula (3.4) we take $t = -2$, $v_k = (k+1)^{-1}$ for $k = 0, 1, \ldots$. The part of the resulting array of the values $\{S_n^{(m)}\}$ are presented in Table 3.1.

Table 3.1.

m	$n=0$	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$
1	1.0000	0.3333							
2	0.0000	0.6667	0.4444						
3	1.3333	0.4444	0.5926	0.4838					
4	−0.6667	0.6667	0.5185	0.5679	0.5185				
5	2.5333	0.4000	0.5777	0.5383	0.5580	0.5317			
6	−2.8000	0.7556	0.5185	0.5580	0.5449	0.5536	0.5390		
7	6.3429	0.2476	0.5862	0.5411	0.5524	0.5474	0.5515	0.5432	
8	−9.6571	1.0095	0.5016	0.5580	0.5467	0.5505	0.5484	0.5505	0.5456

4. Application to Sequences Generated by the Trapezoidal Rule. We consider the application of the above algorithms to the sequences of values generated by the trapezoidal rule to the integral I_1 and I_2 defined in §1.

4.1. Application of the ϕ-Algorithm. We apply the ϕ-algorithm, defined in §2.1 with $\sigma = 0$ to the initial values $\{S_n\}$ of Tables 1.1a and 1.1b, which yields Tables 4.1a and 4.1b, respectively.

Table 4.1a.

m	$n=0$	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
0	0.000000	0.750000	0.666667	0.691666	0.693254	0.693170	0.693153
1	0.750000	0.666667	0.691666	0.693254	0.693170	0.693153	
2	0.708333	0.683333	0.692857	0.693187	0.693156		
3	0.700000	0.688095	0.693055	0.693166			
4	0.697024	0.690079	0.693111				
5	0.695636	0.691090					
6	0.694877						

Table 4.1b.

m	$n=0$	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
0	0.000000	0.683940	0.606531	0.631861	0.632155	0.632122	0.632115
1	0.683940	0.606531	0.631861	0.632155	0.632122	0.632115	
2	0.645235	0.623418	0.632081	0.632129	0.632116		
3	0.637963	0.627750	0.632110	0.632120			
4	0.635409	0.629494	0.632115				
5	0.634226	0.630367					
6	0.633583						

4.2. Application of Romberg Extrapolation. The Romberg extrapolation is applied to the initial values $\{\hat{S}_n\}$ of Tables 1.2a and 1.2b, which yield Tables 4.2a and 4.2b, respectively.

Table 4.2a.

m	$n=0$	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
0	0.000000	1.500000	0.388889	0.735374	0.690385	0.693239	0.693146
1	0.750000	0.666667	0.692064	0.693197	0.693150	0.693147	
2	0.708333	0.685714	0.693055	0.693153	0.693147		
3	0.697024	0.691220	0.693141	0.693148			
4	0.694122	0.692661	0.693147				
5	0.693391	0.693025					
6	0.693208						

Table 4.2b.

m	$n=0$	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
0	0.000000	1.367879	0.352748	0.671818	0.629488	0.692206	0.692119
1	0.683940	0.606531	0.631935	0.632134	0.632121	0.632121	
2	0.645235	0.625584	0.632109	0.632121	0.632121		
3	0.635409	0.630477	0.632120	0.632121			
4	0.632943	0.631709	0.632121				
5	0.632326	0.632018					
6	0.632172						

4.3. Application of the ρ-Algorithm. We apply the ρ-algorithm of §2.3 to the initial values $\{S_n\}$ of Tables 1.1a and 1.1b, and the results are given in Tables 4.3a and 4.3b, respectively.

Table 4.3a.

m	$n=0$	$n=2$	$n=4$	$n=6$
1	0.750000	0.671053		
2	0.708333	0.687500	0.693680	
3	0.700000	0.690741	0.693165	0.693153
4	0.697024	0.691815	0.693155	
5	0.695635	0.692302		
6	0.694877			

Table 4.3b.

m	$n=0$	$n=2$	$n=4$	$n=6$
1	0.683940	0.610677		
2	0.645235	0.627325	0.632377	
3	0.637963	0.630093	0.632122	0.632115
4	0.635409	0.631000	0.632116	
5	0.634226	0.631408		
6	0.633583			

4.4. Application of the $\hat{\rho}$-Algorithm. We apply the ρ-algorithm of §2.4 to the initial values $\{S_n\}$ of Tables 1.2a and 1.2b, and the results are given in Tables 4.4a and

4.4b, respectively.

Table 4.4a.

m	$n=0$	$n=2$	$n=4$	$n=6$	$n=8$
1	0.750000	0.689189			
2	0.708333	0.688705	0.689242		
3	0.697024	0.692030	0.692789	0.692921	
4	0.694122	0.692868	0.693057	0.693093	0.693106
5	0.693391	0.693077	0.693125	0.693134	
6	0.693208	0.693130	0.693142		
7	0.693162	0.693143			
8	0.693151				

Table 4.4b.

m	$n=0$	$n=2$	$n=4$	$n=6$	$n=8$
1	0.683940	0.627480			
2	0.645235	0.628354	0.625600		
3	0.635409	0.631180	0.631817	0.631927	
4	0.632943	0.631885	0.632045	0.632075	0.632086
5	0.632326	0.632062	0.632102	0.632109	
6	0.632172	0.632106	0.632116		
7	0.632133	0.632117			
8	0.632124				

4.5. Application of the ϵ-Algorithm. We apply the ρ-algorithm of §2.5 to the initial values $\{S_n\}$ of Tables 1.2a and 1.2b, and the results are given in Tables 4.5a and 4.5b, respectively.

Table 4.5a.

m	$n=0$	$n=2$	$n=4$	$n=6$
1	0.750000	0.710526		
2	0.708333	0.692810	0.693121	
3	0.697024	0.693120	0.693148	0.693147
4	0.694122	0.693145	0.693147	
5	0.693391	0.693147		
6	0.693208			

Table 4.5b.

m	$n=0$	$n=2$	$n=4$	$n=6$
1	0.693940	0.647308		
2	0.645235	0.632066	0.632117	
3	0.635409	0.632117	0.632121	0.632121
4	0.632943	0.632120	0.632121	
5	0.632326	0.632121		
6	0.632172			

4.6. Corresponding Repeated Application of the ϵ-Algorithm. We apply one cycle of the corresponding repeated application of the ϵ-algorithm, which is described in §2.5.1, to the initial values $\left\{\hat{S}_n\right\}$ of Tables 1.2a and 1.2b, and obtain the results which are shown in Tables 4.6a and 4.6b, respectively.

Table 4.6a.

m	$n=0$	$n=1$	$n=2$
0	0.0000000000	0.0000000000	0.0000000000
1	0.7500000000	0.7500000000	0.7500000000
2	0.7083333433	0.7105263247	0.7125000080
3	0.6970238239	0.6928104744	0.6783857166
4	0.6941218525	0.6931209674	0.6935168160
5	0.6933912188	0.6931477872	0.6931468452
6	0.6932082027	0.6931471604	0.6931471192
7	0.6931623966	0.6931471007	0.6931471035
8	0.6931509525	0.6931471291	0.6931471127

Table 4.6b.

m	$n=0$	$n=1$	$n=2$
0	0.0000000000	0.0000000000	0.0000000000
1	0.6839396954	0.6839396954	0.6839396954
2	0.6452351809	0.6473081781	0.6491704063
3	0.6354094148	0.6320662755	0.6212051356
4	0.6329434216	0.6321171834	0.6324294655
5	0.6323263198	0.6321205681	0.6321205462
6	0.6321720108	0.6321205743	0.6321205734
7	0.6321334057	0.6321205687	0.6321205694
8	0.6321237292	0.6321204544	0.6321205736

4.7. Application of the Generalized Euler Transformation. Since the numbers E_n in Tables 1.2a and 1.2b are almost of the form constant$\times(0.25)^n$, we will apply the generalized Euler transformation of §3 with $t = 0.25$ to the initial values $\{\hat{S}_n\}$ and present the results in Tables 4.8a and 4.8b, respectively.

Table 4.7a.

m	$n=0$	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$
1	0.75000					
2	0.70833	0.69444				
3	0.69702	0.69325	0.69286			
4	0.68412	0.69315	0.69312	0.69321		
5	0.69339	0.69315	0.69315	0.69315	0.69313	
6	0.69321	0.69315	0.69315	0.69315	0.69315	0.69315

Table 4.7b.

m	$n=0$	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$
1	0.68394					
2	0.64524	0.63233				
3	0.63541	0.63213	0.63207			
4	0.63294	0.63212	0.63212	0.63213		
5	0.63233	0.63212	0.63212	0.63212	0.63212	
6	0.63217	0.63212	0.63212	0.63212	0.63212	0.63212

5. Fortran Codes. The Fortran codes are as follows.

Subroutine	Purpose
eps.f90	Displays ϵ-algorithm defined by (2.11)
epsr.f90	Displays corresponding repeated application of ϵ-algorithm defined in §2.5.1.
eul.f90	Displays generalized Euler transformation defined in §3.1.
f1.f90	Defines integrand in I_1
f2.f90	Defines integrand in I_2
fn1.f90	Produces S_m defined by (2.13)
fn2.f90	Computes terms of the series $\sum_{m=0}^{\infty} \frac{t^m}{m+1}$
inteps.f90	Displays ϵ-algorithm defined by (2.11) over the interval (a,b)
intepsr.f90	Displays corresponding repeated application of ϵ-algorithm over (a,b)
inteul.f90	Displays generalized Euler transformation defined in §3.1 over (a,b)
intphi.f90	Displays ϕ-algorithm over (a,b)
intrho.f90	Displays ρ-algorithm over (a,b)
intrhohat.f90	Displays $\hat{\rho}$-algorithm over (a,b)
intromx.f90	Displays Romberg extrapolation over (a,b)
phi.f90	Displays ϕ-algorithm
rho.f90	Displays ρ-algorithm
romb.f90	Computes $\tilde{T}_{n+1}$ and $\tilde{E}_n$ defined in (1.3)
ser1.f90	Computes terms of the series S_m defined by (2.14)
trap.f90	Computes T_{n+1} and E_n defined in (1.2)
sub1.f90	To produce $\hat{S}_{m+1}$ defined by (2.6)

The f90 programs to produce most of the tables can be found in the appropriate subdirectory of Algorithms on the CDRom.

For example, to produce Table 2.7, one types 'make' in the tbl27 subdirectory of the Algorithms directory. A terminal session is shown below.

```
make

f95 -c ./ser1.f90

f95 -c ./epsr.f90

f95 -c tbl27.f90

f95 -o tbl27.x tbl27.o epsr.o ser1.o
```

After typing the executable name 'tbl27.x', the user is prompted to provide `mmax, nor, kol`, with suggested values display with the prompt:

```
input mmax, nor, kol (5,1,3)
```

Table 2.7 is then produced. Other tables are produced is the same way.

References

L. C. Breaux, *A Numerical Study of the Application of Acceleration Techniques and Prediction Algorithms to Numerical Integration*, M. S. Thesis, University of New Orleans, 1971.

D. R. Hartree, *Numerical Analysis*, Oxford Univ. Press, Oxford, 1952.

N. E. Norlund, *Vorlesungen über Differenzenrechnun*, Springer-Verlag, Berlin, 1937.

P. Wynn, *A note on Salzer's method for summing certain slowly convergent series*, J. Math. Phys., **35** (1956a), 318–320.

———, *On a procrustean technique for the numerical transformation of slowly convergent sequences and series*, Proc. Camb. Phil. Soc., **52** (1956b), 91–96.

———, *On a device for computing the $e_m\ (S_n)$ transformation*, Math. Comp., **10** (1956c), 91–62.

———, *A note on programming repeated application of the epsilon algorithm*, Rev. Franc. de Trait. de l'Inf. (Chiffres), **8** (1956d), 23–62.

———, *On the convergence and stability of the epsilon algorithm*, J. SIAM Numer. Anal., **3** (1966), 91–122.