Numer. Math. 38, 299–307 (1982)

Numerische Mathematik

An Algorithm for a Special Case of a Generalization of the Richardson Extrapolation Process

Avram Sidi

Mathematical Support Section, NASA-Lewis Research Center, Cleveland, Ohio, USA

Summary. A special case of a generalization of the Richardson extrapolation process is considered, and its complete solution is given in closed form. Using this, an algorithm for implementing the extrapolation is devised. It is shown that this algorithm needs a very small amount of arithmetic operations and very little storage. Convergence and stability properties for some cases are also considered.

Subject Classifications: AMS(MOS): 63B05; CR: 5.13.

1. Introduction

In a recent paper [6] the author has presented a generalization of the well known Richardson extrapolation process – called GREP for short – which is suitable for accelerating the convergence of a large class of infinite sequences, and has also analyzed some of its convergence properties. By its definition, GREP is implemented through the solution of systems of linear equations. As has been mentioned in [6], the matrices of these linear systems become large in dimension if one wishes to improve the accuracy in extrapolation. Although the solution of these systems can be achieved, for example, by Gaussian elimination, it is clear that as the dimension of the matrices increases this becomes costly. Therefore, it is desirable to have efficient algorithms for solving the linear systems above. Obviously an algorithm, in order to be efficient, has to take advantage of the special character of the equations defining GREP. Unfortunately, so far no such algorithm has been developed for the general form of GREP. For a special case of GREP, however, such an algorithm can be given and this is done in the present work, whose contents can be summarized as follows:

a) In Sect. 2 we obtain the complete solution to a special case of GREP, which is defined by linear systems of equations of the form

$$A_n^{(j)} = A(y_l) + \phi(y_l) \sum_{i=0}^{n} \bar{\beta}_i y_l^{ir}, \qquad l = j, j+1, \ldots, j+n+1, \tag{1.1}$$

0029-599X/82/0038/0299/$01.80

where y_l, $A(y_l)$, $\phi(y_l)$ are given, r is a fixed constant, and $A_n^{(j)}$ and $\bar{\beta}_i$ are unknowns.

b) Using this solution, in Sect. 3 we devise an efficient algorithm, which we call the *W*-algorithm, for the recursive computation of $A_n^{(j)}$.

c) In Sect. 4 we make some statements about the convergence properties of this extrapolation method. Furthermore, for a special case, which comes up in the computation of oscillatory infinite integrals (see [7, 8]), we prove that the *W*-algorithm is stable.

Note that for some special cases of the extrapolation method dealt with in this work algorithms exist, and these cases are:

1) $\phi(y) \equiv$ constant, any y_l. This is the Richardson extrapolation process, and there is a recursive algorithm for it. For a detailed review of the subject and a list of references see [4].

2) $r=1$, $y_l=(l+1)^{-1}$, $l=0,1,\ldots$. This extrapolation method, which is known as the *T*-transformation, has been given by Levin [5], and a non-recursive algorithm for it exists.

3) $r=1$, $y_l=(x_0+l\tau)^{-1}$, $l=0,1,\ldots$, for some fixed x_0 and τ such that $\tau>0$. This has been given by in [7] and comes up in the computation of some oscillatory infinite integrals by methods that the author has denoted as $\bar{D}$ and $\tilde{D}$-transformations. The non-recursive algorithm given in [7] for this case is almost the same as that for the *T*-transformation.

The extrapolation problem defined by Eqs. (1.1) for general y_l and with $r=1$, has come up in the computation of what the author has called very oscillatory infinite integrals, see [8].

Before closing this section we note that recently Håvie [2] and Brezinski [1] have given an algorithm for the general extrapolation problem. Due to its generality, however, this algorithm, when applied to (1.1), is uneconomical compared with the *W*-algorithm. More details on this will be given in Sect. 3.

2. Theory

The following definition characterizes the class of sequences with which we shall be concerned.

Definition 2.1. We shall say that a function $A(y)$, defined for $0<y\leqq b$, for some $b>0$, where y can be a discrete or continuous variable, belongs to the set $F^{(1)}$, if there exist functions $\phi(y)$ and $\beta(y)$ and a constant A, such that

$$A=A(y)+\phi(y)\,\beta(y), \tag{2.1}$$

where $A=\lim_{y\to 0+} A(y)$ whenever this limit exists, in which case $\lim_{y\to 0+}\phi(y)=0$, and $\beta(\xi)$, as a function of the continuous variable ξ, is continuous for $0\leqq\xi\leqq b$, and for some constant $r>0$, as $\xi\to 0+$, has a Poincaré-type asymptotic expansion of the form

$$\beta(\xi)\sim\sum_{i=0}^{\infty}\beta_i\,\xi^{ir}. \tag{2.2}$$

If, in addition, the function $B(t)\equiv\beta(t^{1/r})$ is infinitely differentiable for $0\leqq t\leqq b^r$, we shall say that $A(y)$ belongs to the set $F_\infty^{(1)}$.

For the definition of the sets $F^{(m)}$ and $F_\infty^{(m)}$ see ([6], Definition 1.1).

Our problem is to extrapolate $A(y)$ to $y=0$ and obtain A (or an approximation to it) whether $\lim_{y\to 0+} A(y)$ exists or not. Following Sidi ([6], Definition 1.2) this is done as follows.

Definition 2.2. Let $A(y)\in F^{(1)}$ with the notation of Definition 2.1. Then $A_n^{(j)}$, the approximation to A, and the parameters $\bar{\beta}_i$, $i=0,1,\ldots,n$, are defined to be the solution of the system of $n+2$ linear equations

$$A_n^{(j)}=A(y_l)+\phi(y_l)\sum_{i=0}^{n}\bar{\beta}_i y_l^{ir}, \qquad l=j,j+1,\ldots,j+n+1, \tag{2.3}$$

such that $y_l>0$ for all $l\geqq 0$, $b\geqq y_0>y_1>y_2>\ldots$, and $\lim_{l\to\infty} y_l=0$, provided, of course, that the matrix of the coefficients of Eqs. (2.3) is nonsingular.

Note that for determining $A_n^{(j)}$ one has to know $\phi(y)$ explicitly. The function $\beta(y)$, however, does *not* have to be known explicitly; mere knowledge of its existence and of r in (2.2) is enough. Under certain conditions it can be shown that the $\bar{\beta}_i$ are approximations to the corresponding β_i in (2.2), see [6].

Before going on we shall let $t=y^r$ and $t_l=y_l^r$, $l=0,1,\ldots$, and define $a(t)\equiv A(y)$ and $\varphi(t)\equiv\phi(y)$. Then Eqs. (2.3) become

$$A_n^{(j)}=a(t_l)+\varphi(t_l)\sum_{i=0}^{n}\bar{\beta}_i t_l^i, \qquad l=j,j+1,\ldots,j+n+1. \tag{2.3$'$}$$

From here on we shall use both $A(y)$, $\phi(y)$ and $a(t)$, $\varphi(t)$.

The solution of Eqs. (2.3)′, or equivalently of (2.3), can be computed as in the following theorem.

Theorem 2.1. *Let $D_k^{(s)}$ denote the divided difference operator over the set of points $t_s, t_{s+1},\ldots,t_{s+k+1}$, where for any function $g(t)$ defined at these points*

$$D_k^{(s)}\{g(t)\}\equiv g[t_s,t_{s+1},\ldots,t_{s+k+1}], \qquad s\geqq 0,\ k\geqq -1, \tag{2.4}$$

(see [3], Chap. 2). Then provided $\varphi(t_l)\neq 0$, $j\leqq l\leqq j+n+1$, $A_n^{(j)}$ and $\bar{\beta}_i$, $i=0,\ldots,n$, can be computed recursively from

$$D_{n-p-1}^{(j)}\{[A_n^{(j)}-a(t)]\,t^{-p-1}/\varphi(t)\}=D_{n-p-1}^{(j)}\left\{\sum_{i=0}^{n}\bar{\beta}_i t^{i-p-1}\right\}, \qquad p=-1,0,1,\ldots,n, \tag{2.5}$$

in this order. The explicit expression for $A_n^{(j)}$ is

$$A_n^{(j)}=\frac{D_n^{(j)}\{a(t)/\varphi(t)\}}{D_n^{(j)}\{1/\varphi(t)\}}. \tag{2.6}$$

$\bar{\beta}_0$ is given by

$$\bar{\beta}_0=(-1)^n\left(\prod_{i=j}^{j+n}t_i\right)D_{n-1}^{(j)}\{[A_n^{(j)}-a(t)]\,t^{-1}/\varphi(t)\}, \tag{2.7}$$

and the rest of the $\bar{\beta}_i$ can be computed from

$$\bar{\beta}_p=(-1)^{n-p}\left(\prod_{i=j}^{j+n-p} t_i\right) D^{(j)}_{n-p-1}\left\{[A^{(j)}_n-a(t)]\,t^{-p-1}/\varphi(t)-\sum_{i=0}^{p-1}\bar{\beta}_i t^{i-p-1}\right\},$$
$$p=1,2,\ldots,n. \tag{2.8}$$

Proof. Let

$$c^{(s)}_{k,l}=\prod_{\substack{i=s\\ i\neq l}}^{s+k+1}(t_l-t_i)^{-1}, \qquad l=s,s+1,\ldots,s+k+1, \qquad k\geqq 0. \tag{2.9}$$

Then, see ([3], Chap. 2),

$$D^{(s)}_k\{g(t)\}=\sum_{l=s}^{s+k+1} c^{(s)}_{k,l}\,g(t_l), \tag{2.10}$$

i.e., $D^{(s)}_k\{g(t)\}$ is a linear combination of $g(t_l)$, $l=s,s+1,\ldots,s+k+1$.
Let us now express equations (2.3)′ in the form

$$[A^{(j)}_n-a(t_l)]/\varphi(t_l)=\sum_{i=0}^{n}\bar{\beta}_i t_l^i, \qquad l=j,j+1,\ldots,j+n+1. \tag{2.11}$$

For $-1\leqq p\leqq n$, multiplying the first $n-p+1$ of equations (2.11) by $c^{(j)}_{n-p-1,l}\,t_l^{-p-1}$, $l=j,j+1,\ldots,j+n-p$, and adding all, and finally invoking (2.10), we obtain (2.5).

Let $p=-1$ in (2.5). Then $\sum_{i=0}^{n}\bar{\beta}_i t^{i-p-1}=\sum_{i=0}^{n}\bar{\beta}_i t^i$ is a polynomial of degree at most n. But

$$D^{(s)}_k\{g(t)\}=0, \quad \text{if} \quad g(t) \quad \text{is a polynomial of degree } \leqq k, \tag{2.12}$$

see ([3], Chap. 2). Therefore, the right hand side of (2.5) vanishes when $p=-1$. Using this with the fact that $D^{(j)}_n$ is a linear operator, we obtain (2.6).

Next let us put $p=0$ in (2.5). Then $\sum_{i=0}^{n}\bar{\beta}_i t^{i-p-1}=\bar{\beta}_0 t^{-1}+\sum_{i=1}^{n}\bar{\beta}_i t^{i-1}$, the summation on the right hand side of this equality being a polynomial of degree at most $n-1$. Therefore, with the help of (2.12), for $p=0$, (2.5) yields

$$\bar{\beta}_0=\frac{D^{(j)}_{n-1}\{[A^{(j)}_n-a(t)]\,t^{-1}/\varphi(t)\}}{D^{(j)}_{n-1}\{t^{-1}\}} \tag{2.13}$$

with $A^{(j)}_n$ already computed in (2.6). Using the result

$$D^{(s)}_k\{t^{-1}\}=(-1)^{k+1}\left(\prod_{i=s}^{s+k+1} t_i\right)^{-1}, \qquad s\geqq 0, k\geqq -1, \tag{2.14}$$

in (2.13), (2.7) follows. (2.14) can be proved by induction with the help of a recursion relation given in (3.7) in the next section. The validity of (2.8) can be established in a similar way. The details will not be given here. □

Theorem 2.1 provides us with a method of evaluating $A_n^{(j)}$, $\bar{\beta}_0, \bar{\beta}_1, \ldots, \bar{\beta}_n$, in this order. The following variation of Theorem 2.1 provides us with another method that enables us to evaluate $A_n^{(j)}$, $\bar{\beta}_n, \bar{\beta}_{n-1}, \ldots, \bar{\beta}_0$, in this order.

Theorem 2.2. *$A_n^{(j)}$ and the $\bar{\beta}_i$ in (2.3)′ can be computed recursively from*

$$D_{n-p-1}^{(j)}\{[A_n^{(j)}-a(t)]/\varphi(t)\}=D_{n-p-1}^{(j)}\left\{\sum_{i=0}^{n}\bar{\beta}_i t^i\right\}, \quad p=-1,0,1,\ldots,n, \tag{2.15}$$

in this order. The exact expression for $A_n^{(j)}$ is as given in (2.6) *and the $\bar{\beta}_i$ can be computed from the formulas*

$$\bar{\beta}_n=D_{n-1}^{(j)}\{[A_n^{(j)}-a(t)]/\varphi(t)\} \tag{2.16}$$

and

$$\bar{\beta}_{n-p}=D_{n-p-1}^{(j)}\left\{[A_n^{(j)}-a(t)]/\varphi(t)-\sum_{i=n-p+1}^{n}\bar{\beta}_i t^i\right\}, \quad p=1,2,\ldots,n, \tag{2.17}$$

in this order.

Proof. The proof of (2.15) is achieved exactly as that of (2.5). The proof of (2.16) is achieved by letting $p=0$ in (2.15) and using the fact that

$$D_k^{(s)}\{t^{k+1}\}=1, \quad s\geqq 0, k\geqq -1. \tag{2.18}$$

The proof of (2.17) is achieved similarly. □

In most problems it is mainly $A_n^{(j)}$ that we are interested in. In some problems though we may wish to know the first few β_i in the asymptotic expansion of $\beta(x)$ as given in (2.2). As mentioned previously, $\bar{\beta}_i$ is an approximation to $\beta_i, i=0,1,\ldots$, hence Theorem 2.1 can be used to obtain $A_n^{(j)}$ and the first few of the $\bar{\beta}_i$ without having to compute the rest.

3. The *W*-Algorithm

We now give the *W*-algorithm that enables us to compute the $A_n^{(j)}$ recursively in an efficient manner. Algorithms for each of the $\bar{\beta}_i$ can be devised similarly.

Theorem 3.1. *(The W-algorithm.) Define*

$$\left.\begin{aligned} M_{-1}^{(s)}&=a(t_s)/\varphi(t_s)\\ N_{-1}^{(s)}&=1/\varphi(t_s)\end{aligned}\right\}, \quad s=0,1,2,\ldots, \tag{3.1}$$

and define recursively

$$\left.\begin{aligned} M_k^{(s)}&=\frac{M_{k-1}^{(s+1)}-M_{k-1}^{(s)}}{t_{s+k+1}-t_s}\\ N_k^{(s)}&=\frac{N_{k-1}^{(s+1)}-N_{k-1}^{(s)}}{t_{s+k+1}-t_s}\end{aligned}\right\}, \quad s=0,1,\ldots, \quad k=0,1,\ldots. \tag{3.2}$$

Then

$$A_k^{(s)} = M_k^{(s)}/N_k^{(s)}, \quad s=0,1,\ldots, \quad k=0,1,\ldots, \tag{3.3}$$

or equivalently

$$A_k^{(s)} = \frac{N_{k-1}^{(s+1)} A_{k-1}^{(s+1)} - N_{k-1}^{(s)} A_{k-1}^{(s)}}{N_{k-1}^{(s+1)} - N_{k-1}^{(s)}}, \quad s=0,1,\ldots, \quad k=0,1,\ldots, \tag{3.4}$$

where we have defined

$$A_{-1}^{(s)} = a(t_s), \quad s=0,1,\ldots. \tag{3.5}$$

Proof. (3.3) follows by showing that

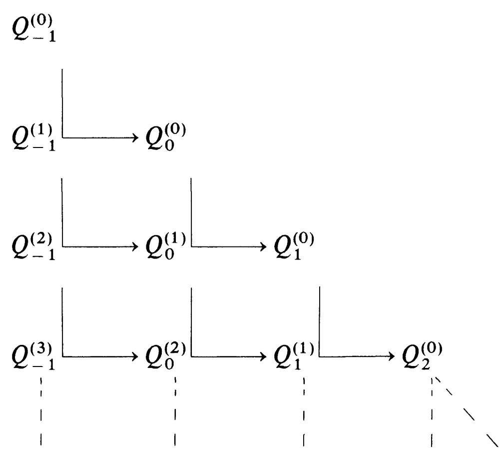
$$\begin{aligned} M_k^{(s)} &= D_k^{(s)}\{a(t)/\varphi(t)\} \\ N_k^{(s)} &= D_k^{(s)}\{1/\varphi(t)\} \end{aligned}, \quad s=0,1,\ldots, \quad k=-1,0,1,\ldots, \tag{3.6}$$

which is a consequence of the relation

$$D_k^{(s)}\{g(t)\} = \frac{D_{k-1}^{(s+1)}\{g(t)\} - D_{k-1}^{(s)}\{g(t)\}}{t_{s+k+1} - t_s}, \quad s=0,1,\ldots, \quad k=0,1,\ldots. \tag{3.7}$$

(3.4) follows by substituting (3.2) in (3.3). This completes the proof of the theorem. □

From Theorem 3.1 we can form two tables of the form



where $Q_k^{(s)}$ stands for $M_k^{(s)}$ in one table and for $N_k^{(s)}$ in the other, and the arrows show the order of computation.

Now the elements of the first column, namely $Q_{-1}^{(s)}$, $s=0,1,\ldots$, are known from (3.1). This enables us to compute the elements of the second column from those of the first, etc.

It is not difficult to see that given the sequence $a(t_l)$, $l=0,1,\ldots,n+1$, the *W*-algorithm, defined through (3.1)-(3.3), enables one to obtain all the approximations $A_k^{(s)}$, $0 \leqq s \leqq k \leqq n$, that obtain from this sequence, the operation count being $3n^2/2 + O(n)$ divisions and $3n^2/2 + O(n)$ subtractions. If we compute only the sequence $A_k^{(o)}$, $0 \leqq k \leqq n$, then the number of divisions drops to $n^2 + O(n)$, the number of subtractions staying the same. (As is mentioned in [6], the sequences $A_k^{(j)}$, $k = 0, 1, \ldots, j$ being fixed, have the best convergence properties, and in practice one takes $j = 0$.) If we apply the algorithm of Håvie [2] or Brezinski [1] to compute $A_k^{(o)}$, $0 \leqq k \leqq n$, then the operation count is basically $O(n^3)$ multiplications, $O(n^3)$ divisions, and $O(n^3)$ additions, see [1]. This, of course, is due to the fact that this algorithm, unlike the W-algorithm, is designed to treat a very general extrapolation problem hence does not take advantage of the special character of the problem dealt with in this work.

4. Some Convergence and Stability Properties

Using (2.10) in the numerator of (2.6), and recalling the second of equations (3.6), we obtain

$$A_n^{(j)} = \sum_{l=j}^{j+n+1} \gamma_{n,l}^{(j)} a(t_l), \tag{4.1}$$

where

$$\gamma_{n,l}^{(j)} = \frac{c_{n,l}^{(j)}/\varphi(t_l)}{N_n^{(j)}}, \qquad l = j, j+1, \ldots, j+n+1. \tag{4.2}$$

Some of the results and observations in [6], which have been given for GREP, for the special case treated in this paper, read as follows:

1. $\sum_{l=j}^{j+n+1} \gamma_{n,l}^{(j)} = 1$ hence $\Gamma_n^{(j)} = \sum_{l=j}^{j+n+1} |\gamma_{n,l}^{(j)}| \geqq 1$.

2. a) If $A(y) \in F^{(1)}$, $\sup_j \Gamma_n^{(j)} < \infty$ with n fixed, and $\lim_{y \to 0+} A(y)$ exists, then $|A - A_n^{(j)}| = o(y_j^{r(n+1)})$ as $j \to \infty$. (See Corollary 3 to Theorem 3.1 in [6].)

b) If $A(y) \in F_\infty^{(1)}$, $\sup_n \Gamma_n^{(j)} < \infty$ with j fixed, and $\lim_{y \to 0+} A(y)$ exists, then $|A - A_n^{(j)}| = o(n^{-\lambda})$ as $n \to \infty$ for any $\lambda > 0$. (See Corollary 3 to Theorem 3.2 in [6].)

3. No matter which method is used for computing $A_n^{(j)}$, it has been observed that the size of $\Gamma_n^{(j)}$ controls the round-off error propagation in the computed value of $A_n^{(j)}$; the larger $\Gamma_n^{(j)}$ is, the larger is the round-off error in $A_n^{(j)}$.

It follows from 1), 2), 3) above that the most ideal situation for both convergence and stability is one in which $\gamma_{n,l}^{(j)} > 0$, $j \leqq l \leqq j+n+1$, for then $\Gamma_n^{(j)} = \sum_{l=j}^{j+n+1} \gamma_{n,l}^{(j)} = 1$, and this is the minimum value that $\Gamma_n^{(j)}$ can have. This situation occurs in some cases in which $A(y)$ oscillates about A as $y \to 0+$. Since $A(y) = A - \phi(y)\beta(y)$, and $\beta(y)$ has a fixed sign as $y \to 0+$, $\phi(y)$ should be oscillatory about zero. Examples of this case would be some alternating series, infinite oscillatory integrals of the kinds treated in [7, 8].

We now wish to address ourselves to the problem of stability of the W-algorithm as described in Theorem 3.1, in situations in which $\gamma_{n,l}^{(j)} > 0$, $j \leqq l \leqq j+n+1$.

Lemma 4.1. *A necessary and sufficient condition for* $\gamma_{n,l}^{(j)} > 0$, $j \leqq l \leqq j+n+1$, *is*

$$\phi(y_l)\,\phi(y_{l+1}) < 0, \quad j \leqq l \leqq j+n. \tag{4.3}$$

Proof. Assume first that (4.3) holds. Recalling $y_l > y_{l+1}$, hence $t_l > t_{l+1}$, $l = 0, 1, \ldots$, (see Definition 2.2), it is easy to see from (2.9) that

$$c_{k,l}^{(s)}\, c_{k,l+1}^{(s)} < 0, \quad s \leqq l \leqq s+k, \quad s \geqq 0, \quad k \geqq 0. \tag{4.4}$$

Combining (4.3) and (4.4), we see that all the terms $c_{n,l}^{(j)}/\varphi(t_l)$, $j \leqq l \leqq j+n+1$, are nonzero and have the *same* sign, i.e.,

$$\begin{aligned} c_{n,l}^{(j)}/\varphi(t_l) &= |c_{n,l}^{(j)}/\varphi(t_l)| \operatorname{sgn}[c_{n,j}^{(j)}/\varphi(t_j)] \\ &= |c_{n,l}^{(j)}/\varphi(t_l)| \operatorname{sgn} \varphi(t_j), \quad j \leqq l \leqq j+n+1, \end{aligned} \tag{4.5}$$

since $\operatorname{sgn} c_{k,s}^{(s)} = +1$. Now by (3.6) and (2.10) we have

$$N_k^{(s)} = \sum_{l=s}^{s+k+1} c_{k,l}^{(s)}/\varphi(t_l), \quad s \geqq 0, \quad k \geqq 0. \tag{4.6}$$

Therefore, from (4.5) it follows that

$$N_n^{(j)} = \left(\sum_{l=j}^{j+n+1} |c_{n,l}^{(j)}/\varphi(t_l)| \right) \operatorname{sgn} \varphi(t_j) \neq 0, \tag{4.7}$$

hence we can divide by $N_n^{(j)}$. This also proves that the solution to equations (2.3) exists. Combining (4.5) and (4.7), and recalling (4.2), $\gamma_{n,l}^{(j)} > 0$, $j \leqq l \leqq j+n+1$, follows. The necessity of (4.3) can be proved similarly and we shall omit the details. □

Lemma 4.2. *Suppose that* (4.3) *is satisfied. Then*

$$N_k^{(s+1)} N_k^{(s)} < 0, \quad j \leqq s \leqq j+n-k-1, \quad -1 \leqq k \leqq n-1. \tag{4.8}$$

Proof. For $k = -1$ we have $N_{-1}^{(s)} = 1/\varphi(t_s)$ from (3.1). Therefore, by (4.3) it is seen that (4.8) holds for $k = -1$. For $j \leqq s \leqq j+n-k-1$ and $0 \leqq k \leqq n-1$, (4.7) applies to $N_k^{(s)}$, hence we obtain

$$\operatorname{sgn}[N_k^{(s+1)} N_k^{(s)}] = \operatorname{sgn}[\varphi(t_{s+1})\,\varphi(t_s)] = -1 \tag{4.9}$$

by (4.3). This proves the lemma. □

Theorem 4.1. *Suppose that* (4.3) *is satisfied. Let* $\bar{a}(t_l) = a(t_l) + \varepsilon_l$, $\bar{\varphi}(t_l) = \varphi(t_l)\,(1+\eta_l)^{-1}$, *such that* $|\varepsilon_l| \leqq \varepsilon$, $|\eta_l| \leqq \eta < 1$, *i.e., assume that errors have been introduced in* $a(t_l)$, $\varphi(t_l)$, $j \leqq l \leqq j+n+1$. *Let us now apply the W-algorithm to* $\bar{a}(t_l)$, $\bar{\varphi}(t_l)$, $j \leqq l \leqq j+n+1$, *and compute* $\bar{M}_k^{(s)}$, $\bar{N}_k^{(s)}$, *and* $\bar{A}_k^{(s)}$ *by*

$$\begin{gathered} \bar{M}_{-1}^{(s)} = \bar{a}(t_s)/\bar{\varphi}(t_s), \quad \bar{N}_{-1}^{(s)} = 1/\bar{\varphi}(t_s), \quad j \leqq s \leqq j+n+1, \\ \bar{M}_k^{(s)} = \frac{\bar{M}_{k-1}^{(s+1)} - \bar{M}_{k-1}^{(s)}}{t_{s+k+1} - t_s}, \quad \bar{N}_k^{(s)} = \frac{\bar{N}_{k-1}^{(s+1)} - \bar{N}_{k-1}^{(s)}}{t_{s+k+1} - t_s}, \quad j \leqq s \leqq j+n-k, \quad 0 \leqq k \leqq n, \\ \bar{A}_k^{(s)} = \bar{M}_k^{(s)}/\bar{N}_k^{(s)}, \quad j \leqq s \leqq j+n-k, \quad -1 \leqq k \leqq n. \end{gathered} \tag{4.10}$$

Then for $j \leqq s \leqq j+n-k,\ -1 \leqq k \leqq n,$

$$\bar{N}_k^{(s)} = N_k^{(s)}(1+\eta_k^{(s)}), \qquad |\eta_k^{(s)}| \leqq \eta, \tag{4.11}$$

$$\bar{M}_k^{(s)} = M_k^{(s)} + E_k^{(s)}, \qquad |E_k^{(s)}| \leqq |N_k^{(s)}|(\eta a + \varepsilon(1+\eta)), \tag{4.12}$$

where $a = \max_{j \leqq l \leqq j+n+1} |a(t_l)|$. *and*

$$|\bar{A}_k^{(s)} - A_k^{(s)}| \leqq \frac{2\eta a + \varepsilon(1+\eta)}{1-\eta}. \tag{4.13}$$

Proof: We shall prove (4.11) by induction on k. (4.11) is true for $k=-1$, with $\eta_{-1}^{(s)} = \eta_s$. Assume that (4.11) holds for $k \leqq q-1 < n$. Then from (4.10) we have

$$\bar{N}_q^{(s)} = \frac{N_{q-1}^{(s+1)}(1+\eta_{q-1}^{(s+1)}) - N_{q-1}^{(s)}(1+\eta_{q-1}^{(s)})}{t_{s+q+1} - t_s}. \tag{4.14}$$

Applying Lemma 4.2 in (4.14), we obtain

$$|\bar{N}_q^{(s)}| = \frac{|N_{q-1}^{(s+1)}|(1+\eta_{q-1}^{(s+1)}) + |N_{q-1}^{(s)}|(1+\eta_{q-1}^{(s)})}{|t_{s+q+1} - t_s|}, \tag{4.15}$$

hence

$$|\bar{N}_q^{(s)}| = \left(\frac{|N_{q-1}^{(s+1)}| + |N_{q-1}^{(s)}|}{|t_{s+q+1} - t_s|}\right)(1+\eta_q^{(s)}), \tag{4.16}$$

where $\min(\eta_{q-1}^{(s)}, \eta_{q-1}^{(s+1)}) \leqq \eta_q^{(s)} \leqq \max(\eta_{q-1}^{(s)}, \eta_{q-1}^{(s+1)})$. By Lemma 4.2 again, the term in parentheses on the right hand side of (4.16) is just $|N_q^{(s)}|$. (4.11) now follows easily. (4.12) can be proved similarly by induction. (4.13) follows from (4.11) and (4.12) together with

$$|M_k^{(s)}| \leqq |N_k^{(s)}| a \tag{4.17}$$

which can be proved easily also by induction. □

Acknowledgement. The author wishes to thank the anonymous referee for his criticism of the original version of this paper, which helped to improve Theorem 4.1.

References

1. Brezinski, C.: A general extrapolation algorithm. Numer. Math. **35**, 175–187 (1980)
2. Håvie, T.: Generalized Neville type extrapolation schemes. BIT **19**, 204–213 (1979)
3. Hildebrand, F.B.: Introduction to numerical analysis. New York: McGraw Hill, 1956
4. Joyce, D.C.: Survey of extrapolation processes in numerical analysis. SIAM Review **13**, 435–490 (1971)
5. Levin, D.: Development of non-linear transformations for improving convergence of sequences. Internat. J. Computer Math. B**3**, 371–388 (1973)
6. Sidi, A.: Some properties of a generalization of the Richardson extrapolation process. J. Inst. Math. Appl. **24**, 327–346 (1979)
7. Sidi, A.: Extrapolation methods for oscillatory infinite integrals. J. Inst. Math. Appl. **26**, 1–20 (1980)
8. Sidi, A.: Methods for the accurate evaluation of very oscillatory infinite integrals. TR # 152, Computer Science Dept., Technion, Haifa, Israel. (in press, 1979b)

Received January 4, 1981/Revised May 5, 1981