

# AN ALGORITHM FOR NONDOMINANT SOLUTIONS OF LINEAR SECOND-ORDER INHOMOGENEOUS DIFFERENCE EQUATIONS

TAKEMITSU HASEGAWA AND TATSUO TORII

**ABSTRACT.** An algorithm is given for computing a weighted sum of a nondominant solution of a linear second-order inhomogeneous difference equation to a prescribed accuracy by estimating the truncation error. The present method is an extension of both the stable numerical method due to Olver and Sookne and a summation technique due to Deuflhard for computing minimal solutions of a homogeneous difference equation. The method is illustrated by numerical examples.

## 1. INTRODUCTION

Let  $a_n$ ,  $b_n$ ,  $c_n$ ,  $d_n$  be given sequences of real or complex numbers and let

$$(1.1) \quad a_n y_{n-1} + b_n y_n + c_n y_{n+1} = d_n, \quad n = 1, 2, \dots,$$

be a given linear second-order inhomogeneous difference equation. In this paper, we shall extend the stable numerical method due to Olver and Sookne [20] for computing minimal solutions of the homogeneous equation associated with (1.1) (i.e.,  $d_n = 0$  for all  $n$ ):

$$(1.2) \quad a_n y_{n-1} + b_n y_n + c_n y_{n+1} = 0, \quad n = 1, 2, \dots,$$

satisfying a normalizing condition of the type

$$(1.3) \quad \sum_{m=0}^{\infty} \lambda_m y_m = s,$$

where  $\lambda_m$  is a given sequence of real or complex numbers and  $s$  is a constant, to the computation of a nondominant solution of the inhomogeneous equation (1.1).

We also extend and incorporate a summation technique due to Deuflhard [5] for a minimal solution of the homogeneous equation (1.2). Indeed, for a positive integer  $K$  and given constants  $\xi_m$ ,  $m = 0, 1, \dots, K$ , we compute

---

Received by the editor December 9, 1993 and, in revised form, June 2, 1994.

1991 *Mathematics Subject Classification*. Primary 65Q05; Secondary 65F05.

*Key words and phrases*. Three-term recurrence relation, nondominant solution, numerical instability, LU factorization, rank-one updating.

the weighted sum

$$(1.4) \quad S_K = \sum_{m=0}^K \xi_m y_m$$

of a nondominant solution of the inhomogeneous difference equation (1.1) to a prescribed accuracy by estimating the truncation error [9]. For applications that need a sum of the type (1.4), such as integration formulas of interpolatory type based on Chebyshev polynomials, we refer to [4, 10, 12]. See also §5.2.

Suppose that the homogeneous equation (1.2), where  $a_n$  and  $c_n$  do not vanish, has a pair of complementary solutions  $f_n$  and  $g_n$  such that  $\lim_{n \rightarrow \infty} f_n/g_n = 0$ . Then  $f_n$  and  $g_n$  are referred to as minimal and dominant, respectively [6]. Let the inhomogeneous equation (1.1) have a particular solution  $h_n$  for which  $\lim_{n \rightarrow \infty} h_n/g_n = 0$ , so that the general solution of (1.1) may be written as

$$(1.5) \quad y_n = \alpha f_n + \beta g_n + h_n,$$

where  $\alpha$  and  $\beta$  are arbitrary constants. Nondominant solutions of (1.1) in the form  $\alpha f_n + h_n$  cannot be found simply by forward and backward recurrence of the difference equation, owing to strong instability; see Gautschi [6], Jones and Thron [13, p.163, p.395] and Wimp [23] for surveys of nondominant solutions and the relationship between difference equations and continued fractions. See also Levrie and Bultheel [14].

Olver [18] introduced a stable method having a built-in error estimation technique [17] for the computation of nondominant solutions of (1.1). In Olver's method the original problem is replaced by an equivalent boundary value problem, which is solved by Gaussian elimination without pivoting; see §2.1.

Since then, extensions or reformulations of Olver's method have been made. Indeed, Van der Cruyssen [22] reformulates Olver's method by solving the boundary value problem by LU decomposition instead of Gaussian elimination to reduce both the amount of computational effort and the possible occurrence of overflow. An extension to high-order recurrence relations is made by Cash [2, 3], where the boundary value problem is solved either by using Gaussian elimination [2] or by using LU decomposition [3]. See Lozier [15] for a more detailed discussion. On the other hand, Wimp [23, p.98] formulates the case using a normalizing condition of the type (1.3) by using Gaussian elimination to solve a system of linear equations in which the coefficient matrix is a band matrix except for the first row representing (1.3).

In either method, whether the one based on Gaussian elimination or on LU decomposition, significant loss in accuracy may occasionally occur in the course of computation [22], [23, p.92]. Olver and Sookne [20] showed that in the case of a homogeneous difference equation it is possible to avoid the loss of accuracy in the elimination stage by combining Miller's algorithm and Olver's algorithm and by using a normalizing condition of the type (1.3). In §2.2, we briefly review their method [20].

In §3 we modify both the method due to Olver and Sookne and the summation technique due to Deuflhard to make them applicable in the more general case of inhomogeneous equations. Indeed, we develop a numerically stable al-

gorithm for computing the sum (1.4) of a nondominant solution of (1.1) by avoiding possible loss of accuracy.

Further, our method is an extension of the methods due to Olver [18] and Van der Cruyssen [22] for initial value problems to the case using the normalizing condition (1.3) as well as a generalization of the method given in Wimp [23, p.98]. In fact, our method also uses the LU decomposition scheme, along with a rank-one updating technique [8, p.592], to solve a system of linear equations which is obtained from the original difference equation (1.1) and the normalizing condition (1.3). The coefficient matrix is a band matrix except for having a row representing the normalizing condition (1.3) in an appropriate place in the matrix determined so that the loss of accuracy in the elimination stage may be avoided. The linear system is successively solved with the size of the system being increased until an estimated truncation error of the required solution is within the prescribed accuracy.

In §4, a convergence result for the present algorithm is discussed. Numerical examples are given in §5.

## 2. OLVER'S METHOD AND ITS MODIFICATION BY OLVER AND SOOKNE

**2.1. Olver's algorithm and loss of accuracy.** For simplicity, we assume that the initial value of the solution  $y_0$  of (1.1) is given, although in Olver's method no initial values of the desired solution are required and a case of a more general normalizing condition (1.3) is discussed. Beginning with  $p_0 = 0$  and  $p_1 = 1$ , one computes the solution  $p_n$  of (1.2) for  $n = 2, 3, \dots$  and a sequence  $\{q_n\}$  defined by

$$(2.1) \quad q_0 = y_0, \quad q_n = (a_n q_{n-1} - d_n p_n)/c_n, \quad n = 1, 2, \dots$$

The computation of  $p_n$  and  $q_n$  is terminated automatically at a certain value  $N$  of  $n$ , which is described below. Let  $\epsilon$  be a given tolerance in the approximation  $y_m^{(N)}$  ( $m < N$ ) to the wanted solution  $y_m$ . Then the value  $N$  is determined so that the following stopping criterion is satisfied

$$(2.2) \quad \left| y_m^{(N+1)} - y_m^{(N)} \right| = |p_m q_N / (p_N p_{N+1})| \leq \epsilon.$$

The approximations  $y_n^{(N)}$  can be generated by computing

$$y_n^{(N)} = (q_n + p_n y_{n+1}^{(N)}) / p_{n+1}, \quad n = N-1, \dots, 1,$$

with starting value  $y_N^{(N)} = 0$ .

Significant loss in accuracy may occasionally occur in the course of the computation of  $p_n$ , especially when the value of  $|f_0|$  is very small [23, p.92]. This can be seen as follows. Since  $p_n$  is expressed as

$$(2.3) \quad p_n = (f_0 g_n - g_0 f_n) / (f_0 g_1 - g_0 f_1),$$

it increases ultimately in proportion to the dominant solution  $g_n$ . If the value of  $|f_0|$  is small compared with  $|g_0 f_1 / g_1|$ , however, initially  $p_n$  may behave more like a multiple of the minimal solution  $f_n$  but soon is contaminated with rounding error growing like  $g_n$  which causes a steady loss in significant figures.

Olver and Sookne [20, p.943] suggest that although there is a loss of significance if the condition prescribes the value of  $f_0$ , this loss in accuracy often does not occur if a more general condition of the form (1.3) is used. In the next subsection we outline their method.

**2.2. Modification by Olver and Sookne.** Olver and Sookne show that the above drawback in Olver's method, when applied to the homogeneous equation (1.2), can be overcome by incorporating Miller's algorithm along with the normalizing condition (1.3). Their modification applies when the dominant solution  $g_n$  tends to infinity with  $n$  in such a way that ultimately  $|g_n|$  is monotonic. In this connection, the following lemma due to Olver and Sookne [20] plays an important role.

**Lemma 2.1** (Olver and Sookne). *For a positive integer  $M$ , let the coefficients  $a_n$ ,  $b_n$  and  $c_n$  in (1.2) satisfy*

$$(2.4) \quad |b_n| \geq |a_n| + |c_n| \quad \text{when} \quad n \geq M + 1,$$

*and  $|b_n| < |a_n| + |c_n|$  otherwise. Let  $p_n$  be the solution of (1.2) satisfying  $p_M = 0$  and  $p_{M+1} = 1$ . Then  $|p_n| \geq |p_{n-1}|$  for all  $n$  greater than  $M$ .*

**Remark 2.1.** The computed solution  $p_n$  above, which is dominant, is free from the numerical difficulty mentioned in the previous subsection.

By using the solution  $p_n$  ( $n = M + 1, M + 2, \dots$ ) obtained in Lemma 2.1, with starting value  $q_M = 1$  instead of  $q_0 = y_0$  Olver and Sookne compute the sequence  $\{q_n\}$  defined by (2.1), where  $d_n$  are replaced by zero for all  $n$ , until the stopping criterion (2.2) is satisfied. Once the value  $N$  has been determined in this way, they apply Miller's method; the backward recursion of (1.2) is made with starting values  $y_N = 0$  and  $y_{N-1} = q_{N-1}/p_N$  to generate  $y_n$ ,  $n = N, N - 1, \dots, 0$ , which are finally renormalized by using the normalizing condition (1.3).

As is mentioned in Olver and Sookne [20], this scheme applies only to homogeneous difference equations; in the case of an inhomogeneous equation, both forward and backward recursion may be unstable. In the next section we consider the case of an inhomogeneous equation, for which we require some assumptions in addition to the condition (2.4).

### 3. STATEMENT OF THE ALGORITHM

Similarly to Olver and Sookne, here and henceforth we assume that the coefficients in (1.1) satisfy the condition (2.4). Further, we need some definition and assumptions on the relative behaviors of  $f_n$ ,  $g_n$ , and  $h_n$  in a finite range of  $n$  as well as their asymptotic behaviors mentioned previously. We begin with the definition of dominance provided by Oliver [16] with some modification. Let  $x_n$  and  $z_n$  be two sequences. Then we say that  $x_n$  dominates  $z_n$  (or  $z_n$  is dominated by  $x_n$ ) in the range  $r \leq n \leq s$  if there exist  $r \geq 0$  and  $s \geq r$  such that  $|x_{n+1}/x_n| > |z_{n+1}/z_n|$  for all  $n$  in the range  $r \leq n \leq s$ . Oliver's original definition is the case where  $s \rightarrow \infty$ . Now, we assume that in the range  $0 \leq n \leq M$  the wanted solution  $y_n = \alpha f_n + h_n$  does not dominate the complementary solutions  $f_n$  and  $g_n$ , whereas  $y_n$  dominates  $f_n$  and is dominated by  $g_n$  in the range  $n > M$ . Recall that since  $g_n$  is a dominant

solution,  $\lim_{n \rightarrow \infty} y_n / g_n = 0$ . Lozier [15, p.23] discusses the difference of the definition of dominance due to Oliver and the dominant solution.

Select an integer  $N$  ( $> K, M$ ), and then replace the problem of computing the sum  $S_K$  (1.4) of  $y_n$  satisfying (1.1) and (1.3) by the one of solving the linear system of equations,

$$(3.1) \quad \begin{bmatrix} A^{(N)} & \mathbf{0} \\ \boldsymbol{\xi}^T & -1 \end{bmatrix} \begin{bmatrix} \mathbf{y}^{(N)} \\ S_K^{(N)} \end{bmatrix} = \begin{bmatrix} \mathbf{d}^{(N)} \\ 0 \end{bmatrix},$$

where the  $(N+1) \times (N+1)$  matrix  $A^{(N)}$  and the  $(N+1)$ -vectors  $\mathbf{y}^{(N)}$ ,  $\mathbf{d}^{(N)}$  and  $\boldsymbol{\xi}$  are defined by

$$(3.2) \quad A^{(N)} = \begin{bmatrix} a_1 & b_1 & c_1 & & & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & & & \vdots \\ 0 & \dots & a_M & b_M & c_M & & \\ \lambda_0 & \dots & & \lambda_M & \lambda_{M+1} & \dots & \lambda_N \\ & & & a_{M+1} & b_{M+1} & c_{M+1} & \dots & 0 \\ \vdots & & & & \ddots & \ddots & \ddots & \vdots \\ & & & & & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & \dots & & & & & a_N & b_N \end{bmatrix},$$

$$(3.3) \quad \mathbf{y}^{(N)} = [y_0^{(N)}, y_1^{(N)}, \dots, y_N^{(N)}]^T,$$

$$(3.4) \quad \mathbf{d}^{(N)} = [d_1, \dots, d_M, s, d_{M+1}, \dots, d_N]^T,$$

$$(3.4) \quad \boldsymbol{\xi} = [\xi_0, \xi_1, \dots, \xi_K, 0, \dots, 0]^T,$$

respectively; we have omitted the dependence of  $\boldsymbol{\xi}$  on  $N$ . Then  $S_K^{(N)}$  ( $= \boldsymbol{\xi}^T \mathbf{y}^{(N)}$ ) and  $y_n^{(N)}$  are approximations to  $S_K$  and  $y_n$ , respectively.

Now we solve the linear system (3.1) by using an LU decomposition similarly to Van der Cruyssen [22] and Cash [3] along with a rank-one updating technique [8, p.592]. Let  $B$  denote the same  $(N+1) \times (N+1)$ -matrix as  $A^{(N)}$ , but having the elements  $\lambda_k$  ( $k = M+2, \dots, N$ ) replaced with zeros. Further, let  $\mathbf{e}$  be a unit vector

$$(3.5) \quad \mathbf{e} = [e_0, \dots, e_N]^T, \quad \text{where } e_k = 0 \ (k = 0, \dots, N, k \neq M) \text{ and } e_M = 1,$$

and let  $\mathbf{r}$  be a vector defined by

$$(3.6) \quad \mathbf{r} = [0, \dots, 0, \lambda_{M+2}, \dots, \lambda_N]^T.$$

Suppose that we have the LU factorization  $LU = B$ ; then we have

$$(3.7) \quad A^{(N)} = B + \mathbf{e} \mathbf{r}^T = LU + \mathbf{e} \mathbf{r}^T,$$

where  $L$  and  $U$  are lower and upper triangular matrices, respectively, given by

$$(3.8) \quad L = \begin{bmatrix} 1 & & & & & & \\ \vdots & \ddots & & & & & \\ 0 & \dots & \ddots & & & & \\ \mu_0 & \dots & \mu_{M-1} & 1 & & & \\ & & & l_{M+1} & 1 & & \\ \vdots & & & & \ddots & \ddots & \\ 0 & \dots & & & & l_N & 1 \end{bmatrix},$$

$$(3.9) \quad U = \begin{bmatrix} a_1 & b_1 & c_1 & & & \dots & 0 \\ & \ddots & \ddots & \ddots & & & \vdots \\ & & a_M & b_M & c_M & & \\ & & & u_M & \zeta & & \\ & & & & u_{M+1} & c_{M+1} & \\ & & & & & \ddots & \ddots \\ & & & & & & u_N \end{bmatrix}.$$

The values of  $\mu_0, \dots, \mu_{M-1}, l_{M+1}, \dots, l_N, u_M, \dots, u_N$ , and  $\zeta$  in (3.8) and (3.9) will be given later.

**Theorem 3.1.** Let  $A^{(N)}$ ,  $L$  and  $U$  be the  $(N+1) \times (N+1)$ -matrices defined by (3.2), (3.8) and (3.9), respectively. For the vectors  $\mathbf{d}^{(N)}$ ,  $\mathbf{e}$ ,  $\mathbf{r}$ ,  $\boldsymbol{\xi}$  defined by (3.3), (3.5), (3.6), (3.4), respectively, let  $\mathbf{z}$ ,  $\mathbf{h}$ ,  $\mathbf{v}$  and  $\mathbf{w}$  be  $(N+1)$ -vectors obtained by computing  $L\mathbf{z} = \mathbf{d}^{(N)}$ ,  $L\mathbf{h} = \mathbf{e}$ ,  $U^T\mathbf{v} = \mathbf{r}$  and  $U^T\mathbf{w} = \boldsymbol{\xi}$  by the process of forward substitution, respectively. Further, for the solution  $\mathbf{y}^{(N)}$  of  $A^{(N)}\mathbf{y}^{(N)} = \mathbf{d}^{(N)}$ , let  $\mathbf{x}$  be a vector defined by  $\mathbf{x} = U\mathbf{y}^{(N)}$ . Then, for  $\mathbf{x}$  and  $S_K^{(N)}$  in (3.1) we have

$$(3.10) \quad \mathbf{x} = \mathbf{z} - \mathbf{h} \{ \mathbf{v}^T \mathbf{z} / (1 + \mathbf{v}^T \mathbf{h}) \} \quad \text{and} \quad S_K^{(N)} = \mathbf{w}^T \mathbf{x}.$$

*Proof.* We see from (3.7) that  $A^{(N)}\mathbf{y}^{(N)} = \mathbf{d}^{(N)}$  can be rewritten as

$$(LU + L\mathbf{h}\mathbf{v}^T U)U^{-1}\mathbf{x} = L\mathbf{z},$$

which yields

$$(3.11) \quad \mathbf{x} + \mathbf{h}\mathbf{v}^T \mathbf{x} = \mathbf{z}.$$

The first equation in (3.10) follows easily by solving (3.11) for  $\mathbf{x}$ . The second equation in (3.10) is easily verified from the fact that  $S_K^{(N)} = \boldsymbol{\xi}^T \mathbf{y}^{(N)} = \boldsymbol{\xi}^T U^{-1} \mathbf{x} = \mathbf{w}^T \mathbf{x}$ .  $\square$

**Remark 3.1.** The solution  $\mathbf{y}^{(N)}$  of  $A^{(N)}\mathbf{y}^{(N)} = \mathbf{d}^{(N)}$  can also be obtained if we solve  $U\mathbf{y}^{(N)} = \mathbf{x}$  by back substitution after the appropriate value of  $N$  has been determined in such a way as described below in this section.

We proceed to show how to compute the nontrivial elements in  $L$  and  $U$  given by (3.8) and (3.9), respectively, and the elements of vectors  $\mathbf{z}$ ,  $\mathbf{h}$ ,  $\mathbf{v}$  and  $\mathbf{w}$  defined in Theorem 3.1. Let  $\boldsymbol{\mu} = [\mu_0, \dots, \mu_{M-1}]^T$ , and  $\boldsymbol{\lambda} =$

$[\lambda_0, \dots, \lambda_{M-1}]^T$ , where  $\mu_k$  and  $\lambda_k$  are elements in the  $(M+1)$ th rows of  $L$  (3.8) and  $A^{(N)}$  (3.2), respectively. Let  $U_M$  be the leading principal submatrix of order  $M$  of  $U$  (3.9)

$$U_M = \begin{bmatrix} a_1 & b_1 & c_1 & \dots & 0 \\ & \ddots & \ddots & \ddots & \vdots \\ & & & \ddots & c_{M-2} \\ \vdots & & & \ddots & b_{M-1} \\ 0 & \dots & & & a_M \end{bmatrix}.$$

Then one can see from (3.2) and (3.7) that  $\boldsymbol{\mu}^T U_M = \boldsymbol{\lambda}^T$ , i.e.,  $U_M^T \boldsymbol{\mu} = \boldsymbol{\lambda}$ , which is easily solved for  $\boldsymbol{\mu}$  by forward substitution as follows:

$$(3.12) \quad \mu_i = \frac{\lambda_i - b_i \mu_{i-1} - c_{i-1} \mu_{i-2}}{a_{i+1}}, \quad i = 0, 1, \dots, M-1,$$

with starting values  $\mu_{-1} = \mu_{-2} = 0$ . By using the values of  $\boldsymbol{\mu} = [\mu_0, \dots, \mu_{M-1}]^T$  obtained above, the elements  $\zeta$ ,  $l_j$  ( $j = M+1, \dots, N$ ) and  $u_j$  ( $j = M, \dots, N$ ) in  $L$  (3.8) and  $U$  (3.9) are computed as follows:

$$(3.13) \quad \zeta = \lambda_{M+1} + c_M \mu_{M-1}, \quad u_M = \lambda_M - b_M \mu_{M-1} - c_{M-1} \mu_{M-2},$$

$$(3.14) \quad l_{M+1} = \frac{a_{M+1}}{u_M}, \quad u_{M+1} = b_{M+1} - \zeta l_{M+1},$$

$$(3.15) \quad l_j = \frac{a_j}{u_{j-1}}, \quad u_j = b_j - c_{j-1} l_j, \quad j = M+2, \dots, N.$$

Now, we evaluate each component of the vectors  $\mathbf{z}$ ,  $\mathbf{h}$ ,  $\mathbf{v}$  and  $\mathbf{w}$ , for which we set  $\mathbf{z} = [z_0, \dots, z_N]^T$ ,  $\mathbf{h} = [h_0, \dots, h_N]^T$ ,  $\mathbf{v} = [v_0, \dots, v_N]^T$  and  $\mathbf{w} = [w_0, \dots, w_N]^T$ , respectively. From the equations  $L\mathbf{z} = \mathbf{d}^{(N)}$  and  $L\mathbf{h} = \mathbf{e}$  we have

$$\begin{aligned} z_i &= d_{i+1}, & h_i &= 0, & i &= 0, \dots, M-1, \\ z_M &= s - \sum_{k=0}^{M-1} \mu_k z_k, & h_M &= 1, \\ z_j &= d_j - l_j z_{j-1}, & h_j &= -l_j h_{j-1}, & j &= M+1, \dots, N. \end{aligned}$$

Similarly, from  $U^T \mathbf{v} = \mathbf{r}$  and  $U^T \mathbf{w} = \boldsymbol{\xi}$  we have

$$w_{-1} = w_{-2} = 0, \quad w_i = \frac{\xi_i - b_i w_{i-1} - c_{i-1} w_{i-2}}{a_{i+1}}, \quad i = 0, 1, \dots, M-1,$$

$$w_M = \frac{\xi_M - b_M w_{M-1} - c_{M-1} w_{M-2}}{u_M}, \quad w_{M+1} = \frac{\xi_{M+1} - \zeta w_M - c_M w_{M-1}}{u_{M+1}},$$

$$v_0 = v_1 = \dots = v_{M+1} = 0,$$

$$w_j = \frac{\xi_j - c_{j-1} w_{j-1}}{u_j}, \quad v_j = \frac{\lambda_j - c_{j-1} v_{j-1}}{u_j}, \quad j = M+2, \dots, N,$$

where we have set  $\xi_i \equiv 0$  ( $i > K$ ).

From (3.10) we can compute the values of  $\mathbf{x}$  and  $S_K^{(N)}$  as follows:

$$(3.16) \quad \begin{aligned} x_i &= z_i, \quad i = 0, \dots, M-1, & x_j &= z_j - h_j \phi(N), \quad j = M, \dots, N, \\ S_K^{(N)} &= \sum_{k=0}^N w_k x_k = \sum_{k=0}^N w_k z_k - \phi(N) \sum_{k=M}^N w_k h_k, \end{aligned}$$

where  $\phi(N)$  is defined by

$$\phi(N) = \frac{\sum_{k=M+2}^N v_k z_k}{1 + \sum_{k=M+2}^N v_k h_k}.$$

It follows immediately from (3.16) that

$$(3.17) \quad S_K^{(N+1)} - S_K^{(N)} = w_{N+1} z_{N+1} - \phi(N+1) \sum_{k=M}^{N+1} w_k h_k + \phi(N) \sum_{k=M}^N w_k h_k,$$

and the truncation error in  $S_K^{(N)}$  can be written in the form

$$(3.18) \quad \begin{aligned} S_K - S_K^{(N)} &= \sum_{n=N}^{\infty} \{S_K^{(n+1)} - S_K^{(n)}\} \\ &= \sum_{k=N+1}^{\infty} w_k z_k - \phi(\infty) \sum_{k=M}^{\infty} w_k h_k + \phi(N) \sum_{k=M}^N w_k h_k. \end{aligned}$$

Generally, the right-hand side of (3.17) converges to zero rapidly as  $N$  increases and therefore the truncation error (3.18) is of the same order of magnitude as the first term  $S_K^{(N+1)} - S_K^{(N)}$ . Here, we consider the following two cases: if we wish to compute  $S_K^{(N)}$  to the required absolute accuracy  $\varepsilon_a$ , then we apply the recurrence relation for values of  $n$  past both  $K$  and  $M$  until

$$(3.19) \quad |S_K^{(n+1)} - S_K^{(n)}| \leq \varepsilon_a,$$

is satisfied, and we then set  $N = n$ . On the other hand, if we wish to compute  $S_K^{(N)}$  to the relative accuracy  $\varepsilon_r$ , then we examine the condition

$$(3.20) \quad |(S_K^{(n+1)} - S_K^{(n)}) / S_K^{(n)}| \leq \varepsilon_r,$$

instead of (3.19). If the series (3.18) does not converge rapidly, the stopping criteria above could fail. To enhance the reliability of the criteria, it might be advisable to continue the computation until the condition (3.19) (or (3.20)) is satisfied several times, at least twice consecutively. In the numerical examples below we checked the criterion (3.19) twice.

Now we note that the appropriate choice of  $M$  in the present algorithm plays an important role in reducing the number of computations required as well as in



reducing the possible occurrence of loss of accuracy mentioned in §2.1. Indeed, the present algorithm requires less computational effort if  $M$  can be set as large as possible so as to satisfy the condition (2.4). To see this, we now count the number of computations required in the present algorithm. Factorization into an  $LU$  product requires  $N + M$  multiplications and  $N$  divisions. To solve for  $\mathbf{v}$  (and  $\mathbf{w}$ ),  $N - M - 1$  (and  $N + M$ ) multiplications and  $N - M - 1$  (and  $N + 1$ ) divisions are required. Similarly,  $N$  and  $N - M$  multiplications are required in solving for  $\mathbf{z}$  and  $\mathbf{h}$ , respectively. Further,  $4N - 3M + 1$  multiplications and a single division are required to compute  $S_K^{(N)}$ . To sum up, we need  $9N - 3M - 1$  multiplications and  $3N - M + 1$  divisions.

On the other hand, for the matrix  $A^{(N)}$  (3.2) in the special case where  $M = 0$  and  $\lambda_1 = 0$ , the computation of (3.15) might occasionally suffer from loss of accuracy similar to §2.1. The reason is simply seen as follows. Now, we assume that  $\lambda_0 \neq 0$  in addition to  $M = 0$  and  $\lambda_1 = 0$ . Then we see from (3.13) and (3.14) that  $\zeta = 0$ ,  $u_0 = \lambda_0$ ,  $l_1 = a_1/\lambda_0$  and  $u_1 = b_1$ . Further, if we denote  $u_n = -c_n \eta_{n+1}/\eta_n$ ,  $n \geq 1$ , using a sequence  $\{\eta_n\}$ , then it is seen from above and (3.15) that  $\eta_n$ ,  $n \geq 1$ , satisfies the recurrence relation (1.2) with starting value  $\eta_0 = 0$  and any  $\eta_1 \neq 0$ . This means that  $\eta_n$  is expressed as a multiple of the right-hand side of (2.3), and therefore the computation in (3.15) becomes very susceptible to loss of significance when  $f_0$  happens to be very small in modulus compared to  $g_0$ ; see Table 1 below.

Finally, the solution  $\mathbf{y}^{(N)}$  can also be obtained by solving  $U\mathbf{y}^{(N)} = \mathbf{x}$  by backward substitution as follows:

$$(3.21) \quad y_N^{(N)} = \frac{x_N}{u_N}, \quad y_j^{(N)} = \frac{x_j - c_{j+1} y_{j+1}^{(N)}}{u_j}, \quad j = N-1, \dots, M+1,$$

$$y_M^{(N)} = \frac{x_M - \zeta y_{M+1}^{(N)}}{u_M}, \quad y_i^{(N)} = \frac{x_i - b_{i+1} y_{i+1}^{(N)} - c_{i+1} y_{i+2}^{(N)}}{a_{i+1}}, \quad i = M-1, \dots, 0.$$

We conclude this section with some remarks on the relationship between the present algorithm and other existing methods. The algorithm given in this section includes the method of Deuffhard, the method of Van der Cruyssen, and that given in Wimp as particular cases. Indeed, we can see that the Deuffhard scheme [5] coincides with the present one in the special case where  $d_n \equiv 0$ ,  $n = 0, \dots, N$  and  $s = 0$  in (3.1) and (3.3) and  $M = N$  in (3.2), provided the last diagonal elements in  $L$  (3.8) and  $U$  (3.9) are replaced with  $\mu_N$  and  $a_{N+1}$ , respectively. Here  $\mu_N$  is determined from the relation (3.12) with  $i$  set to  $N$ .

Another case of our problem, where  $M = 0$  and  $\xi_i = 0$  ( $i = 0, \dots, K-1$ )  $\xi_K = 1$ , agrees with that given in Wimp [23, p.98], for which the Gaussian elimination technique is applied instead of LU decomposition with rank-one updating technique. Further, the present scheme reduces to the method due to Van der Cruyssen [22] for the initial value problem, where we set  $M = 0$  and  $\lambda_i = 0$  ( $i = 1, \dots, N$ ) in (3.2) as well as  $\xi_i = 0$  for all  $i \neq K$  but  $\xi_K = 1$  in (3.4), while Cash's algorithm [3] is an extension of the Van der Cruyssen method to high-order difference equations.

## 4. CONVERGENCE RESULTS

We discuss convergence as  $N \rightarrow \infty$  of the algorithm given in the previous section, in particular, of  $y_n^{(N)}$  and therefore of  $S_K^{(N)}$  (3.10). We seek nondominant solutions  $y_n$  of (1.1) in the form  $y_n = \alpha f_n + h_n$  with a constant  $\alpha$ , which is determined so as to satisfy the normalizing condition (1.3), i.e.,

$$(4.1) \quad \alpha = \left( s - \sum_{m=0}^{\infty} \lambda_m h_m \right) / \sum_{m=0}^{\infty} \lambda_m f_m.$$

We now examine the convergence of  $y_n^{(N)}$  to  $y_n$  as  $N \rightarrow \infty$ .

It is seen from (1.5) that the solution  $y_n^{(N)}$  of (3.1) can be written as

$$(4.2) \quad y_n^{(N)} = \zeta_1^{(N)} f_n + \zeta_2^{(N)} g_n + h_n,$$

for some constants  $\zeta_1^{(N)}$ ,  $\zeta_2^{(N)}$ . On the other hand, the solution  $y_n^{(N)}$  must satisfy the conditions

$$y_{N+1}^{(N)} = 0, \quad \sum_{m=0}^N \lambda_m y_m^{(N)} = s.$$

The above relations along with (4.2) yield

$$(4.3) \quad \zeta_1^{(N)} F_N + \zeta_2^{(N)} G_N + H_N = s,$$

$$(4.4) \quad \zeta_1^{(N)} f_{N+1} + \zeta_2^{(N)} g_{N+1} + h_{N+1} = 0,$$

where  $F_N$ ,  $G_N$  and  $H_N$  are defined, respectively, by

$$(4.5) \quad F_N = \sum_{m=0}^N \lambda_m f_m, \quad G_N = \sum_{m=0}^N \lambda_m g_m, \quad H_N = \sum_{m=0}^N \lambda_m h_m.$$

Solving (4.3) and (4.4) for  $\zeta_1^{(N)}$  and  $\zeta_2^{(N)}$  gives

$$(4.6) \quad \zeta_1^{(N)} = -\{h_{N+1} G_N + g_{N+1}(s - H_N)\} / (f_{N+1} G_N - g_{N+1} F_N),$$

$$(4.7) \quad \zeta_2^{(N)} = \{f_{N+1}(s - H_N) + h_{N+1} F_N\} / (f_{N+1} G_N - g_{N+1} F_N).$$

**Theorem 4.1.** *Let  $f_n$ ,  $g_n$  and  $h_n$  be solutions of (1.1) having the properties given in §1, that is,  $\lim_{n \rightarrow \infty} f_n/g_n = 0$  and  $\lim_{n \rightarrow \infty} h_n/g_n = 0$ . Further, let  $F_N$ ,  $G_N$  and  $H_N$  be defined by (4.5), respectively, with the properties*

$$\lim_{N \rightarrow \infty} f_{N+1} G_N / g_{N+1} = 0, \quad \lim_{N \rightarrow \infty} h_{N+1} G_N / g_{N+1} = 0,$$

*and  $F_N$  and  $H_N$  do not tend to 0. Then the algorithm given in the previous section converges and*

$$(4.8) \quad \lim_{N \rightarrow \infty} y_n^{(N)} = y_n.$$

*Proof.* To show (4.8), it is sufficient to verify in (4.2) that  $\lim_{N \rightarrow \infty} \zeta_1^{(N)} = \alpha$  and  $\lim_{N \rightarrow \infty} \zeta_2^{(N)} = 0$ , which follow simply by using the assumption of Theorem 4.1 in (4.6) and (4.7), respectively, and by noting (4.1).  $\square$

## 5. NUMERICAL EXAMPLES

Examples in this section were computed on the SONY NEWS 3860 in double precision; the machine precision is  $\epsilon = 2.22 \dots \times 10^{-16}$ .

**5.1. Example 1.** We compute a finite sum  $S_K = \sum_{n=0}^K y_n$  of the nondominant solution  $y_n = 2^{-n}$  ( $n = 0, 1, \dots$ ) of the following linear second-order difference equation

$$(5.1) \quad y_{n-1} - \frac{2n}{x} y_n + y_{n+1} = \left(2.5 - \frac{2n}{x}\right) 2^{-n}, \quad n \geq 1.$$

Here, the minimal and dominant solutions of (5.1) are Bessel functions of the first and second kind  $J_n(x)$  and  $Y_n(x)$ , respectively, both of which are of oscillatory type [19] for  $0 \leq n \leq [x]$ . For  $n \geq [x]$ , on the other hand,  $|J_n|$  and  $|1/Y_n|$  decrease monotonically to zero as  $n$  increases. The condition (2.4) is satisfied if we choose  $M = [x]$ .

As mentioned in §3, when  $x$  in (5.1) is a zero of the minimal solution  $J_0(x)$ , numerical instability may occur in our algorithm in the special case where  $\lambda_1 = 0$  and  $M = 0$  in (3.2), although this numerical difficulty could be eliminated by setting  $M$  to be any positive integer  $\leq [x]$ . To see this, we choose  $x = 8.653727912911012$ , in Table 1 (next page). Namely we choose the third positive zero of  $J_0(x)$ , and a normalizing condition of the type  $y_0 + 2(y_2 + y_3 + \dots) = 2$ .

In Table 1 we list the computed solutions  $y_n^{(N)}$  and the approximate sum  $S_K^{(N)} = \sum_{n=0}^K y_n^{(N)}$  with  $K = 14$  satisfying the condition (3.19) for the required tolerance  $\epsilon_a = 10^{-10}$ . As is shown in the second and fourth columns in Table 1, the estimated  $N$  for which  $|S_{14}^{(N+1)} - S_{14}^{(N)}| \leq 10^{-10}$  is  $N = 40$  when we incorrectly set  $M = 0$  in (3.2), but  $N = 35$  when  $M = [x] = 8$ . The computed value of  $S_{14}^{(N)}$  in the last row of Table 1 is seen to satisfy the tolerance  $\epsilon_a = 10^{-10}$  for  $M = 8$ , while the value of  $S_{14}^{(40)}$  in the column  $M = 0$  fails, owing to numerical instability, as expected.

It should be noted that the particular solution  $y_n$  ( $n \geq 0$ ) wanted is dominated by  $J_n$  only for  $0 \leq n \leq [x]$  and dominates  $J_n$  otherwise, while  $Y_n$  dominates  $y_n$  for all  $n \geq 0$ . This relative dominance was crucial for the present algorithm to work. If this relative dominance is violated, the present method fails. To see this, consider the recurrence relation of the type (5.1) having the right-hand side so that the particular solution  $y_n = a^n$  if  $0 \leq n \leq [x]$  and  $y_n = a^{2[x]-n}$  if  $n > [x]$ , with some constant  $a > 1$ . Choose the normalizing condition  $y_0 + 2(y_2 + y_3 + \dots) = 1 + 2((a+1)a^{[x]} - a^2)/(a-1)$  and let  $x = 20.5$  and  $a = 5$ . Then the computed value of the weighted sum  $S_3 = \sum_{n=0}^3 y_n$  is  $S_3^{(N)} = 155.99948883$  for the required accuracy  $\epsilon_a = 10^{-10}$  and the estimated  $N$  is  $N = 47$ . The actual error of  $S_3^{(N)}$  is  $5 \times 10^{-4}$ , which is much greater than  $\epsilon_a$  ( $= 10^{-10}$ ).

TABLE 1. Computed values  $y_n^{(N)}$  of nondominant solutions  $y_n = 2^{-n}$  of the recurrence relation (5.1) and approximate sum  $S_K^{(N)} = \sum_{n=0}^K y_n^{(N)}$  with  $K = 14$  for the required absolute accuracy  $\varepsilon_a = 10^{-10}$ . Italic digits are incorrect digits in the computed  $y_n^{(N)}$  and  $S_{14}^{(N)}$ . Complementary solutions of (5.1) are Bessel functions  $J_n(x)$  and  $Y_n(x)$ . Setting  $x = 8.653727912911012$ , the third positive zero of  $J_0(x)$ , causes numerical instability, seen in the second column, when we choose  $M = 0$  in (3.2), while with an appropriate choice of  $M = [x] = 8$  no numerical difficulty is detected, as shown in the fourth column

n	$M = 0$ ♣		$M = [x] = 8$ ◇	
	$y_n^{(N)} (= y_n^{(40)})$	error	$y_n^{(N)} (= y_n^{(35)})$	error
♣ 0	1.00000000E 00	2E-16	1.00000000000000E 00	0E 00
1	5.40487415E-01	4E-02	5.0000000000982E-01	1E-11
2	2.59357219E-01	9E-03	2.5000000000227E-01	2E-12
3	8.88377584E-02	4E-02	1.2499999999123E-01	9E-12
4	2.80699505E-02	4E-02	6.2499999991649E-02	8E-12
5	3.55831387E-02	4E-03	3.1250000001051E-02	1E-12
6	5.50623004E-02	4E-02	1.5625000009565E-02	1E-11
7	5.81664984E-02	5E-02	7.8125000122127E-03	1E-11
◇ 8	4.59316423E-02	4E-02	3.9062500101927E-03	1E-11
9	2.93004691E-02	3E-02	1.9531250066327E-03	7E-12
10	1.58344232E-02	2E-02	9.7656250360358E-04	4E-12
11	7.47957323E-03	7E-03	4.8828125169565E-04	2E-12
12	3.15994063E-03	3E-03	2.4414062570719E-04	7E-13
13	1.21737417E-03	1E-03	1.2207031276565E-04	3E-13
14	4.36059716E-04	4E-04	6.1035156340957E-05	9E-14
15	1.48643095E-04	1E-04	3.0517578153650E-05	3E-14
16	4.97416161E-05	4E-05	1.5258789070863E-05	8E-15
17	1.70154514E-05	9E-06	7.6293945335265E-06	2E-15
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
32	2.32830597E-10	5E-17	2.3282628627229E-10	4E-15
33	1.16415310E-10	1E-17	1.1638371709852E-10	3E-14
34	5.82076593E-11	2E-18	5.7970976272870E-11	2E-13
35	2.91038293E-11	1E-18	2.7275594464178E-11	2E-12
36	1.45519059E-11	9E-18		
37	7.27588098E-12	8E-17		
38	3.63733286E-12	7E-16		
39	1.81339310E-12	6E-15		
40	8.59698592E-13	5E-14		
$S_{14}^{(N)} =$	2.00000000E 00	6E-05	1.9999389648847E 00	4E-11

**5.2. Example 2.** The present method has practical applications in constructing interpolatory integration rules for an integral of highly oscillatory functions, which is difficult to evaluate with ordinary quadrature rules. In particular, our scheme could be effectively used to obtain a set of approximations to an indefinite integral, such as  $\int_{-1}^x e^{i\omega t} f(t) dt$  with a set of values of  $x$  and a set of values of  $\omega$ , where  $f(t)$  is a given smooth function and  $|x| \leq 1$ .

Suppose that  $f(t)$  is expanded in terms of Chebyshev polynomials  $T_n(t) = \cos n\theta$  ( $t = \cos \theta$ ) as follows:

$$(5.2) \quad f(t) = \sum_{n=0}^{\infty}{}' a_n T_n(t), \quad |t| \leq 1,$$

where the prime denotes a summation whose first term is halved. In practice,  $f(t)$  can be efficiently approximated by a finite sum of Chebyshev polynomials  $p_m(t) = \sum_{n=0}^m a_n^{(m)} T_n(t)$  by using the Fast Fourier Transform (FFT) [1, 7, 11]. If  $f(t)$  is a smooth function, the approximation  $p_m$  converges to  $f$  rapidly as  $m$  increases. For the function  $f(t)$  given in (5.2) (or the approximation  $p_m(t)$ ) we prove the following lemma in Appendix A.

**Lemma 5.1.** *For the function  $f(t)$  given by (5.2), write the indefinite integral  $\int_{-1}^x e^{i\omega t} f(t) dt$  in the form [10, 12]*

$$(5.3) \quad I(x, \omega; f) \equiv \int_{-1}^x e^{i\omega t} f(t) dt = \frac{1}{i\omega} e^{i\omega x} g(x), \quad |x| \leq 1, \quad \omega > 0,$$

and expand  $g(x)$  in terms of Chebyshev polynomials

$$(5.4) \quad g(x) = \sum_{n=0}^{\infty}{}' y_n T_n(x).$$

Then, the coefficients  $y_n$  in (5.4) satisfy the inhomogeneous difference equation

$$(5.5) \quad y_{n-1} - i \frac{2n}{\omega} y_n - y_{n+1} = a_{n-1} - a_{n+1}, \quad n \geq 1,$$

and a normalizing condition of the type  $\sum_{n=0}^{\infty}{}' y_n (-1)^n = 0$ .

**Remark 5.1.** For the approximation  $p_m(t)$  instead of  $f(t)$ , it suffices to replace  $a_{n-1}$  and  $a_{n+1}$  in the right-hand side of (5.5) by  $a_{n-1}^{(m)}$  and  $a_{n+1}^{(m)}$ , respectively, where we assume that  $a_n^{(m)} = 0$  for all  $n > m$ .

The coefficient  $y_n$  in (5.4) is a particular solution of (5.5). The complementary solutions of (5.5) are modified Bessel functions  $I_n(-i\omega)$  and  $(-1)^n K_n(-i\omega)$  ( $= K_n(i\omega)$ ). In the range  $n \leq [\omega]$ , both of  $|I_n|$  and  $|K_n|$  are of oscillatory type, whereas for  $n > [\omega]$ ,  $|I_n|$  and  $|1/K_n|$  decrease rapidly to zero as  $n$  increases.

Here we show that with the choice of  $M = [\omega]$  in (2.4) the present scheme could effectively evaluate the particular solutions  $y_n$  and the weighted sum  $g(x)$  (5.4) to a prescribed accuracy  $\varepsilon_a$ . In reality, instead of  $g(x)$  given in (5.4) it is sufficient to obtain a truncated Chebyshev series

$$(5.6) \quad g(x) \approx g_K(x) = \sum_{n=0}^K{}' y_n T_n(x),$$

where  $K$  is determined so that the truncation error of  $g_K(x)$  is at the levels of the rounding error of the computer, which is usually less than the required

accuracy  $\varepsilon_a$  for  $y_n$  or the weighted sum  $g(x)$  of  $y_n$ . In practice, it might be allowed to set  $K$  to the smallest  $n$  such that  $|y_{n+1}| \leq \varepsilon$ , provided  $|y_n| = O(r^n)$  for a positive constant  $r < 1$  and for sufficiently large  $n$ , because then the error  $|\sum_{n=K+1}^{\infty} y_n T_n(x)| \leq \sum_{n=K+1}^{\infty} |y_n| \approx |y_{K+1}|/(1-r)$ . If the required accuracy  $\varepsilon_a$  is much larger than  $\varepsilon$ , then we might determine the smallest  $K$  so that  $|y_{K+1}| \leq c \varepsilon_a$  for some small constant  $c > 0$ , say,  $c = 0.1$  at most. In incorporating this scheme into the present algorithm, we approximate  $\{y_n\}$  by  $\{y_n^{(n)}\} = \{x_n/u_n\}$ , the first relation in (3.21).

Now, we compute the weighted sum  $S_K \equiv g_K(x=1) = \sum_{n=0}^K y_n$  (5.6) to the prescribed accuracy  $\varepsilon_a = 10^{-10}$  for the indefinite integral (5.3) of  $f(t)$ , where in particular we choose

$$f(t) = \frac{1 - \alpha^2}{1 - 2\alpha t + \alpha^2} = 2 \sum_{n=0}^{\infty} \alpha^n T_n(t), \quad \alpha = 0.9,$$

and  $\omega = 150$ . In this case the right-hand side of (5.5) is  $2(\alpha^{-1} - \alpha)\alpha^n$  for  $n \geq 1$ . We choose  $M = [\omega] = 150$  in (3.2) to satisfy the condition (2.4). The estimated value of the smallest  $K$  such that  $|y_{K+1}| \leq 0.1 \times \varepsilon_a = 0.1 \times 10^{-10}$  is  $K = 223$  and the estimated  $N$  for which  $|S_{223}^{(N+1)} - S_{223}^{(N)}| \leq 10^{-10}$  is  $N = 225$ . The computed  $S_{223}^{(N)} \equiv g_{223}(x=1) = 10.83928930523 + i 6.550524798738$  and the error is estimated to be  $E \equiv g(1) - g_{223}(1) = (0.8 + i 7) \times 10^{-11}$  by using the value of  $g(x=1)$  evaluated as follows. From (5.3) we have  $g(x) = i\omega e^{-i\omega x}(\Re I + i\Im I)$ , where  $\Re I$  and  $\Im I$  are  $\Re I \equiv \Re I(x, \omega; f) = \int_{-1}^x \cos(\omega t) f(t) dt$  and  $\Im I \equiv \Im I(x, \omega; f) = \int_{-1}^x \sin(\omega t) f(t) dt$ , respectively. To compute  $\Re I(x=1, \omega=150; f)$  and  $\Im I(x=1, \omega=150; f)$  we used the QUADPACK [21] routine DQAWO with the required absolute accuracy  $= 10^{-14}$ .

TABLE 2. Weighted sums  $g_{223}(x) = y_0 T_0(x)/2 + y_1 T_1(x) + \dots + y_{223} T_{223}(x)$  ( $x = 0.1, 0.3, \dots, 0.9$ ) of the computed nondominant solutions  $y_n$  of the recurrence relation (5.5) for the required absolute accuracy  $\varepsilon_a = 10^{-10}$  and their actual errors. Clenshaw's algorithm is used to evaluate the summation. Italic digits are incorrect digits

$x$	$g_{223}(x) = \Re g + i\Im g$		Error = $\Re E + i\Im E$	
	$\Re g$	$\Im g$	$\Re E$	$\Im E$
0.1	0.1198685750311	0.05338395512740	-3E-12	-3E-12
0.3	0.09819379394844	0.01279216097213	-2E-12	-3E-12
0.5	0.1895492858980	-0.04626511161379	9E-13	-1E-11
0.7	0.3905994887744	-0.01898488674630	2E-12	-1E-11
0.9	1.025563752769	0.1025908390260	-9E-14	3E-12

Once all the values of  $y_n^{(N)}$  ( $n = 0, 1, \dots, 223$ ) are computed from (3.21) after the weighted sum  $S_{223}^{(N)} = g_{223}(x=1)$  of  $y_n^{(N)}$  is obtained, any value of  $g_{223}(x)$  for many  $x$ , where  $|x| \leq 1$ , can be easily evaluated by using Clenshaw's algorithm. Table 2 lists the computed values of  $g_{223}(x_\nu)$ , where  $x_\nu = 0.1, 0.3, \dots, 0.9$  and their errors, which are estimated by using the

QUADPACK in a similar way mentioned above. The computed  $g_{223}(x)$ 's listed in the second and third columns are seen to satisfy the required accuracy  $\varepsilon_a = 10^{-10}$ .

#### APPENDIX A

Here we prove Lemma 5.1. Differentiating both sides of (5.3) with respect to  $x$  and changing the variable  $x$  into  $t$  yield

$$f(t) = g(t) + g'(t)/(i\omega).$$

Integrating this relation on  $[-1, x]$  gives

$$(5.7) \quad \int_{-1}^x f(t) dt = \int_{-1}^x g(t) dt + \{g(x) - g(-1)\}/(i\omega).$$

Substituting the Chebyshev series expansions (5.2) and (5.4) into (5.7) and noting

$$2 \int_{-1}^x T_n(t) dt = \frac{T_{n+1}(x)}{n+1} - \frac{T_{n-1}(x)}{n-1} - (-1)^n \frac{2}{n^2-1}, \quad n \geq 2,$$

we obtain the relation (5.5) by comparing the coefficients of  $T_n(x)$ . For (5.3) to hold with  $x = -1$  requires the condition  $g(-1) = 0$ , which yields along with (5.4) the normalizing condition  $\sum_{n=0}^{\infty} y_n (-1)^n = 0$ .

#### ACKNOWLEDGMENT

We are grateful to the referees for their valuable comments.

#### BIBLIOGRAPHY

1. M. Branders and R. Piessens, *An extension of Clenshaw-Curtis quadrature*, J. Comput. Appl. Math. **1** (1975), 55–65.
2. J. R. Cash, *An extension of Olver's method for the numerical solution of linear recurrence relations*, Math. Comp. **32** (1978), 497–510.
3. ———, *A note on the numerical solution of linear recurrence relations*, Numer. Math. **34** (1980), 371–386.
4. C. W. Clenshaw and A. R. Curtis, *A method for numerical integration on an automatic computer*, Numer. Math. **2** (1960), 197–205.
5. P. Deufhard, *A summation technique for minimal solutions of linear homogeneous difference equations*, Computing **18** (1977), 1–13.
6. W. Gautschi, *Computational aspects of three-term recurrence relations*, SIAM Rev. **9** (1967), 24–82.
7. W. M. Gentleman, *Implementing Clenshaw-Curtis quadrature II. Computing the cosine transformation*, Comm. ACM **15** (1972), 343–346.
8. G. H. Golub and C. F. Van Loan, *Matrix computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, 1989.
9. T. Hasegawa and T. Torii, *A stable algorithm for numerical solutions of second-order linear difference equations*, J. Inform. Process. **23** (1982), 583–590 (in Japanese).
10. ———, *Indefinite integration of oscillatory functions by the Chebyshev series expansion*, J. Comput. Appl. Math. **17** (1987), 21–29.

11. T. Hasegawa, T. Torii, and H. Sugiura, *An algorithm based on the FFT for a generalized Chebyshev interpolation*, Math. Comp. **54** (1990), 195–210.
12. T. Hasegawa and A. Sidi, *An automatic integration procedure for infinite range integrals involving oscillatory kernels*, submitted.
13. W. B. Jones and W. J. Thron, *Continued fractions: analytic theory and applications*, Addison-Wesley, Reading, MA, 1980.
14. P. Levrie and A. Bultheel, *Convergence acceleration for the numerical solution of second-order linear recurrence relations*, SIAM J. Numer. Anal. **27** (1990), 166–177.
15. D. W. Lozier, *Numerical solution of linear difference equations*, report NBSIR 80–1976, NBS, Washington, 1980.
16. J. Olver, *Numerical solution of linear recurrence relations*, Numer. Math. **11** (1968), 349–360.
17. ———, *An extension of Olver's error estimation technique for linear recurrence relations*, Numer. Math. **12** (1968), 459–467.
18. F. W. J. Olver, *Numerical solution of second-order linear difference equations*, J. Res. NBS **71** (B) (1967), 111–129.
19. ———, *Error bounds for linear recurrence relations*, Math. Comp. **50** (1988), 481–499.
20. F. W. J. Olver and D. J. Sookne, *Note on backward recurrence algorithms*, Math. Comp. **26** (1972), 941–947.
21. R. Piessens, E. deDoncker-Kapenga, C. W. Überhuber, and D. K. Kahaner, *QUADPACK, A subroutine package for automatic integration*, Springer-Verlag, Berlin-Heidelberg-New York-Tokyo, 1983.
22. P. Van der Cruyssen, *A reformulation of Olver's algorithm for the numerical solution of second-order linear difference equations*, Numer. Math. **32** (1979), 159–166.
23. J. Wimp, *Computation with recurrence relations*, Pitman, Boston, 1984.

DEPARTMENT OF INFORMATION SCIENCE, FACULTY OF ENGINEERING, FUKUI UNIVERSITY,  
FUKUI, 910, JAPAN

E-mail address: [hasegawa@fuis.fuis.fukui-u.ac.jp](mailto:hasegawa@fuis.fuis.fukui-u.ac.jp)

DEPARTMENT OF INFORMATION ENGINEERING, FACULTY OF ENGINEERING, NAGOYA UNIVERSITY,  
NAGOYA, 464-01, JAPAN