JOURNAL OF RESEARCH of the National Bureau of Standards—B. Mathematics and Mathematical Physics
Vol. 71B, Nos. 2 and 3, April-September 1967

Numerical Solution of Second-Order Linear Difference Equations

F. W. J. Olver

Institute for Basic Standards, National Bureau of Standards, Washington, D.C. 20234

(May 4, 1967)

A new algorithm is given for computing the solution of any second-order linear difference equation which is applicable when simple recurrence procedures cannot be used because of instability. Compared with the well-known Miller algorithm the new method has the advantages of (i) automatically determining the correct number of recurrence steps, (ii) applying to inhomogeneous difference equations, (iii) enabling more powerful error analyses to be constructed.

The method is illustrated by numerical computations, including error analyses, of Anger-Weber, Struve, and Bessel functions, and the solution of a differential equation in Chebyshev series.

Key Words: Chebyshev series, difference equations, error analysis, Miller algorithm, recurrence methods, special functions.

1. Introduction

A powerful computational algorithm for evaluating the most rapidly decreasing solution of a second-order homogeneous linear difference equation was published in 1952 by J. C. P. Miller ([1],[1] page xvii) in connection with the tabulation of modified Bessel functions. Since then, various writers have applied the algorithm to other special functions, and similar computational processes have been used by Clenshaw [2] for the numerical solution of ordinary differential equations in series of Chebyshev polynomials. Error analyses of the algorithm have been supplied by the present writer [3] and Oliver [12] and quite recently Gautschi [4] has examined the relation of the algorithm to classical results in the theory of continued fractions.

The present investigation stems from the observation that Miller's algorithm can be regarded as a procedure for solving a tridiagonal set of simultaneous linear algebraic equations. Adopting this more general standpoint, we shall show how to recast the algorithm into a new form which enables the correct number of recurrence steps to be determined automatically without appeal to an asymptotic or other analytical formula. In this respect it resembles an algorithm proposed recently by Shintani [5].

The new formulation has the further advantages of (i) being applicable to inhomogeneous difference equations, (ii) lending itself readily to powerful error analyses. There seems to be no alternative method of comparable power available at present for computing solutions of inhomogeneous equations in the case when forward recurrence and backward recurrence are both unstable.

2. Statement of the Problem

Let the given difference equation be denoted by

$$a_r y_{r-1} - b_r y_r + c_r y_{r+1} = d_r, \tag{2.01}$$

[1] Figures in brackets indicate the literature references at the end of this paper.

where a_r, b_r, c_r, and d_r are given functions of the nonnegative integer variable r. We assume that the general solution of (2.01) has the form

$$y_r = Af_r + Bg_r + h_r, \tag{2.02}$$

in which A and B are arbitrary constants, and the complementary functions f_r, g_r, and the particular solution h_r have the properties $f_0 \neq 0$, $g_r \neq 0$ for all sufficiently large r, and

$$f_r/g_r \to 0, \qquad h_r/g_r \to 0, \qquad (r \to \infty). \tag{2.03}$$

(It may be noted that we do not require either f_r or h_r to tend to zero as $r \to \infty$.)

The first problem we investigate is the computation of the solution of (2.01) which has the property

$$y_r/g_r \to 0 \qquad (r \to \infty), \tag{2.04}$$

and satisfies the *normalizing condition*

$$y_0 = k \tag{2.05}$$

for an arbitrarily assigned value of the constant k. Later (secs. 9–11) we allow for a more general form of normalizing condition and also drop the restriction $f_0 \neq 0$.

The given conditions ensure that y_r exists and is unique. For, from (2.03) and (2.04) the B of (2.02) is seen to be zero, and from (2.05) we derive $A = (k - h_0)/f_0$. Therefore

$$y_r = \frac{k - h_0}{f_0} f_r + h_r. \tag{2.06}$$

It is well known that direct use of (2.01) as a recurrence relation for generating y_2, y_3, . . . from given values of y_0 and y_1 (if available) is an unstable procedure. Essentially, each computational rounding error introduces into the numerical solution a small multiple of f_r and a small multiple of g_r, and in consequence of (2.04) the latter ultimately grows faster than the wanted solution.

It may also happen [2] in the inhomogeneous case that f_r grows more rapidly than y_r in the direction of decreasing r. In this event recurrence by use of (2.01) is unstable in this direction too.

3. Approach

Analogous work in the numerical solution of linear differential equations [3] suggests that a stable way of solving the present problem is to treat it directly as a boundary-value problem rather than use initial-value techniques. We are already given the value of y_0. Suppose that for some large integer N, the value of y_N can be obtained from an asymptotic formula or by other means. Then eqs (2.01) with $r = 1, 2, \ldots, N-1$ comprise a set of simultaneous linear algebraic equations for the unknowns $y_1, y_2, \ldots, y_{N-1}$, which are solvable by standard matrix computational processes.

This possibility has already been noted by Gautschi ([4], Introduction) following a suggestion by M. E. Rose, but Gautschi did not pursue the idea because of the difficulty of obtaining the value of y_N, in general. Following Miller's approach in the homogeneous case [1], we solve the algebraic equations with the value of y_N arbitrarily set equal to zero. It transpires that for large N the great majority of the y_r produced in this way are generally excellent approximations to the true values; only in the neighborhood of $r = N$ can substantial errors occur.

We first establish the convergence of the process.

[2] See Example 1 of section 6.
[3] See, for example, [6].

THEOREM 1. *With the conditions of section 2, suppose that for all sufficiently large* N *the system of equations*

$$a_r y_{r-1}^{(N)} - b_r y_r^{(N)} + c_r y_{r+1}^{(N)} = d_r, \quad (r = 1, 2, \ldots, N-1), \tag{3.01}$$

and

$$y_0^{(N)} = k, \qquad y_N^{(N)} = 0, \tag{3.02}$$

has a solution $y_0^{(N)}, y_1^{(N)}, \ldots, y_N^{(N)}$. *Then if* r *is fixed and* $N \to \infty$, $y_r^{(N)} \to y_r$.

To establish this result, we observe that since y_r is a particular solution of (2.01), we can express

$$y_r^{(N)} = A_N f_r + B_N g_r + y_r,$$

where A_N and B_N are independent of r. Setting $r = 0$, N in turn and using (2.05) and (3.02), we derive

$$A_N = \frac{g_0 y_N}{f_0 g_N - g_0 f_N}, \qquad B_N = -\frac{f_0 y_N}{f_0 g_N - g_0 f_N},$$

the denominator $f_0 g_N - g_0 f_N$ being nonvanishing for all sufficiently large N in consequence of the assumed conditions. In fact, from (2.03) we have

$$f_0 g_N - g_0 f_N \sim f_0 g_N \qquad (N \to \infty).$$

Hence from (2.04) it follows that $A_N \to 0$ and $B_N \to 0$. This completes the proof.

Thus for any given value of r, or for any finite range of values, y_r can be calculated to prescribed accuracy by solving the system of equations (3.01) and (3.02) with a sufficiently high value of N. Naturally, we now enquire what exactly constitutes a "sufficiently high" value? Or, to put the question another way, given N, to what accuracy does $y_r^{(N)}$ approximate y_r for $r < N$?

A simple practical way of providing an answer is to solve eqs (3.01) and (3.02) for increasing values of N until the results are in satisfactory numerical agreement. This procedure has two drawbacks. First, it is wasteful of computing time if the originally guessed values of N are either too low or much too high. Second, there is no absolute guarantee that values of y_r computed with two (or more) different values of N must be correct when they agree.

The optimum value of N, that is, the minimum value necessary to achieve specified accuracy in y_r for a given range of values of r, can be determined automatically when a suitable method is used to solve the algebraic equations. Accordingly, we consider this process next.

4. Solution of the Algebraic Equations

We shall solve the tridiagonal system of equations

$$\left.\begin{aligned}
-b_1 y_1^{(N)} + c_1 y_2^{(N)} &= d_1 - a_1 k, \\
a_2 y_1^{(N)} - b_2 y_2^{(N)} + c_2 y_3^{(N)} &= d_2, \\
a_3 y_2^{(N)} - b_3 y_3^{(N)} + c_3 y_4^{(N)} &= d_3, \\
&\cdots\cdots\cdots, \\
a_{N-2} y_{N-3}^{(N)} - b_{N-2} y_{N-2}^{(N)} + c_{N-2} y_{N-1}^{(N)} &= d_{N-2}, \\
a_{N-1} y_{N-2}^{(N)} - b_{N-1} y_{N-1}^{(N)} &= d_{N-1},
\end{aligned}\right\} \tag{4.01}$$

by simple elimination followed by back-substitution. To begin with we suppose that none of the c_r vanish.

Let the first of (4.01) be rewritten in the form

$$p_2 y_1^{(N)} - p_1 y_2^{(N)} = e_1, \tag{4.02}$$

where

$$p_1 = 1, \qquad p_2 = \frac{b_1}{c_1}, \qquad e_1 = \frac{a_1 k - d_1}{c_1}. \tag{4.03}$$

The result of eliminating $y_1^{(N)}$ from (4.02) and the second of (4.01) can be expressed as

$$p_3 y_2^{(N)} - p_2 y_3^{(N)} = e_2,$$

where

$$p_3 = \frac{b_2 p_2 - a_2 p_1}{c_2}, \qquad e_2 = \frac{a_2 e_1 - d_2 p_2}{c_2}.$$

Continuing the elimination, we obtain

$$p_{r+1} y_r^{(N)} - p_r y_{r+1}^{(N)} = e_r \qquad (r = 1, 2, \ldots, N-1), \tag{4.04}$$

where

$$p_{r+1} = \frac{b_r p_r - a_r p_{r-1}}{c_r}, \qquad e_r = \frac{a_r e_{r-1} - d_r p_r}{c_r}. \tag{4.05}$$

Thus p_r is the solution of the homogeneous form of the difference eq (2.01), with the initial conditions $p_0 = 0$ and $p_1 = 1$. We also observe that the second of (4.05) holds for $r = 1$ if we define

$$e_0 = k.$$

The final equation of the form (4.04) is used to begin the back-substitution. On substituting the second of (3.02), we derive

$$y_{N-1}^{(N)} = e_{N-1}/p_N; \tag{4.06}$$

thence $y_{N-2}^{(N)}, y_{N-3}^{(N)}, \ldots, y_1^{(N)}$ may be computed by use of (4.04) with descending values of r. The process fails if, and only if, one of the numbers $p_2, p_3, \ldots, p_N$ vanishes. In this event the set of eqs (3.01) and (3.02) has either no solution or an infinity of solutions, and the algorithm breaks down.

When one or more of the coefficients c_r vanishes the set of eqs (4.01) becomes uncoupled. A simple modification takes care of the situation. Suppose, for example, that $c_s = 0$ but all other c_r are nonzero. Then the first s equations of (4.01) determine $y_1^{(N)}, y_2^{(N)}, \ldots, y_s^{(N)}$ completely: they can be solved by application of the recurrence relations (4.05) for $r = 1, 2, \ldots, s-1$ and use of the back-substitution relation (4.04), beginning with

$$y_s^{(N)} = \frac{a_s e_{s-1} - d_s p_s}{b_s p_s - a_s p_{s-1}}. \tag{4.07}$$

The remaining $N-s-1$ equations are solvable for $y_{s+1}^{(N)}, y_{s+2}^{(N)}, \ldots, y_{N-1}^{(N)}$ by the method already described: eq (4.07) takes the place of the first of (3.02).

To ease the presentation we shall suppose in the remainder of the paper that none of the c_r vanish.

Applying ourselves to the problem of determining the optimum value of N, we observe that the effect of replacing N by $N+1$ is to prolong the elimination process by one step, beginning the back-substitution with $y_{N+1}^{(N+1)}=0$ instead of $y_N^{(N)}=0$. Thus we have

$$p_{r+1}y_r^{(N+1)}-p_r y_{r+1}^{(N+1)}=e_r \qquad (r=1, 2, \ldots, N). \tag{4.08}$$

Subtraction of (4.04) from (4.08) gives

$$y_r^{(N+1)}-y_r^{(N)}=\frac{p_r}{p_{r+1}}\,(y_{r+1}^{(N+1)}-y_{r+1}^{(N)}) \qquad (r\leqslant N-1), \tag{4.09}$$

and repeated application of this result leads to

$$y_r^{(N+1)}-y_r^{(N)}=\frac{p_r}{p_{r+1}}\frac{p_{r+1}}{p_{r+2}}\cdots\frac{p_{N-1}}{p_N}\,(y_N^{(N+1)}-y_N^{(N)}),$$

that is,

$$y_r^{(N+1)}-y_r^{(N)}=\frac{p_r e_N}{p_N p_{N+1}} \qquad (r=1, 2, \ldots, N). \tag{4.10}$$

By use of this formula we can predict the effect of changing N into $N+1$ *before* any back-substitution is carried out.

Suppose, for example, that we wish to compute y_L to D decimal places for given values of the integers L and D. Then the recurrence relations (4.05) are applied from $r=1$ past $r=L$ until a value of r is reached for which

$$\left|\frac{p_L e_r}{p_r p_{r+1}}\right|<\tfrac{1}{2}\times 10^{-D}. \tag{4.11}$$

If this value of r is taken as N, then we can be sure that the approximation $y_L^{(N)}$ yielded by the back-substitution agrees to D decimal places with the value $y_L^{(N+1)}$ that would be obtained from the next higher approximation. (Whether this value of N *is* adequate is considered in the next section.)

If, as is more usual, accurate values of y_r are required for a whole range of values of r, then the criterion (4.11) is used with $|p_L|$ denoting the greatest value of $|p_r|$ in the given range. We might, for example, desire the computation to D decimal places of *all* values of y_r that exceed $\frac{1}{2}\times 10^{-D}$ in absolute value—it being assumed, of course, in this case that $y_r\to 0$ as $r\to\infty$. Then N is determined by the condition that

$$|e_N/p_{N+1}|<\tfrac{1}{2}\times 10^{-D}, \tag{4.12}$$

provided that $|p_N|\geqslant|p_r|$ when $r<N$.

5. Expansions for the Solution and the Truncation Error

The method suggested in the last section for determining N is based on the criterion that the values of $y_r^{(N)}$ and $y_r^{(N+1)}$ must agree to within the prescribed tolerance for y_r. This does not guarantee, however, that their common value is y_r. To resolve this doubt we consider higher approximations.

Replacing N by $N+1$ in (4.10), and adding the result to (4.10) itself, we obtain

$$y_r^{(N+2)} - y_r^{(N)} = p_r \left(\frac{e_N}{p_N p_{N+1}} + \frac{e_{N+1}}{p_{N+1} p_{N+2}} \right).$$

Continuation of this process yields

$$y_r^{(N+s)} - y_r^{(N)} = p_r \left(\frac{e_N}{p_N p_{N+1}} + \frac{e_{N+1}}{p_{N+1} p_{N+2}} + \cdots + \frac{e_{N+s-1}}{p_{N+s-1} p_{N+s}} \right),$$

where s is an arbitrary positive integer. Letting $s \to \infty$ and using Theorem 1, we derive the following expression for the *truncation error*

$$\epsilon_r^{(N)} \equiv y_r - y_r^{(N)} = E_N p_r, \tag{5.01}$$

where E_N is the sum of the (necessarily convergent) series

$$E_N = \sum_{s=N}^{\infty} \frac{e_s}{p_s p_{s+1}}. \tag{5.02}$$

Thus the precise criterion for determining N is that $|E_N p_r|$ must not exceed the specified tolerance in y_r for each wanted value of r.

Once the value of N has been decided, the actual value of the truncation error can be found by continuing the computation of p_r and e_{r-1} beyond $r=N$ and using (5.01) and (5.02). Later [7], we shall show how to use these expansions to determine strict bounds for $\epsilon_r^{(N)}$ directly from the properties of the coefficients a_r, b_r, c_r, and d_r.

As a special case of (5.01) we have the expansion

$$y_r = p_r \sum_{s=r}^{\infty} \frac{e_s}{p_s p_{s+1}}. \tag{5.03}$$

Subtraction of (5.01) from (5.03) yields

$$y_r^{(N)} = p_r \sum_{s=r}^{N-1} \frac{e_s}{p_s p_{s+1}} \qquad (r < N), \tag{5.04}$$

a result which is obtainable more directly by repeated use of the back-substitution relation (4.04). Thus *the whole of our computing scheme is equivalent to approximating the convergent infinite series (5.03) by the partial sum (5.04).*

6. Examples

EXAMPLE 1. *Anger-Weber functions.*

For integer values of r the function $\mathbf{E}_r(x)$ satisfies eq (2.01) with

$$a_r = c_r = 1, \qquad b_r = \frac{2r}{x}, \qquad d_r = -\frac{2\{1-(-1)^r\}}{\pi x}.$$

We restrict ourselves here to positive values of the argument x. The principal properties of $\mathbf{E}_r(x)$ are established in [8], chapter 10. In particular, we have

$$\mathbf{E}_{2r}(x) = \frac{2x}{(4r^2-1)\pi} \sum_{s=0}^{\infty} \alpha_s(r) x^{2s}, \qquad \mathbf{E}_{2r+1}(x) = \frac{2}{(2r+1)\pi} \sum_{s=0}^{\infty} \frac{2r-2s-1}{2r-1} \alpha_s(r) x^{2s}, \tag{6.01}$$

where $\alpha_0(r) = 1$, and

$$\alpha_s(r) = \frac{1}{(4r^2-3^2)(4r^2-5^2)\ \ldots\ \{4r^2-(2s+1)^2\}} \qquad (s>0). \tag{6.02}$$

Using the inequality

$$\left|\frac{\alpha_s(r)}{\alpha_{s-1}(r)}\right| \leqslant \frac{1}{|4r-1|},$$

we deduce that if x is fixed and $r \to \infty$, then

$$\mathbf{E}_{2r}(x) \sim \frac{2x}{(4r^2-1)\pi}, \qquad \mathbf{E}_{2r+1}(x) \sim \frac{2}{(2r+1)\pi}. \tag{6.03}$$

The corresponding homogeneous form of (2.01) has the Bessel functions $J_r(x)$ and $Y_r(x)$ as solutions. For fixed x and large r, we have

$$J_r(x) \sim \frac{1}{(2\pi r)^{1/2}}\left(\frac{ex}{2r}\right)^r, \qquad Y_r(x) \sim -\left(\frac{2}{\pi r}\right)^{1/2}\left(\frac{2r}{ex}\right)^r. \tag{6.04}$$

Thus ultimately $J_r(x)$ decays more rapidly than $\mathbf{E}_r(x)$, and $|Y_r(x)|$ grows rapidly. In consequence, both simple forward recurrence and simple backward recurrence are unstable methods for generating $\mathbf{E}_r(x)$ from (2.01) when $r > x$.

With

$$f_r = J_r(x), \qquad g_r = Y_r(x), \qquad h_r = \mathbf{E}_r(x),$$

the conditions of section 2 are satisfied, provided that $J_0(x) \neq 0$. Let us apply the method of section 4 to a specific example, say the computation of $\mathbf{E}_r(x)$ for $x = 1$, $r = 1(1)10$, correct to within 2 units of the eighth decimal place. We suppose that we are given $\mathbf{E}_0(1) = -0.56865\ 6627$, this value having been extracted from [9] and confirmed by evaluation of the first of (6.01).

Beginning with $p_0 = 0$, $p_1 = 1$, and $e_0 = \mathbf{E}_0(1)$, values of p_r and e_r were generated by use of (4.05). They are recorded in the upper part of table 1, correct to 9 significant figures. After passing the last of the given values of r, namely 10, the "test function" $p_{10}e_r/(p_r p_{r+1})$ was computed. This falls below the value 2×10^{-8} for the first time when $r = 14$. In accordance with the criterion of section 4 this is the value [4] to be assigned to N. The column of values $y_r^{(14)}$ was then generated by backward use of (4.04), beginning with $y_{14}^{(14)} = 0$. For $r = 1(1)10$ these are the wanted approximations to $\mathbf{E}_r(1)$.

To test the accuracy of the results, the computations were repeated for $N = 32$ and $N = 34$, using a time-sharing automatic computer and working to 36 floating binary figures, with an exponent of 12 binary figures. As further checks the values for $r = 1(1)5$ were compared with the 10-decimal values given in [9], and the values for $r = 10$ and 11 computed from the expansions (6.01). The full results of these computations are not included here, but the digits in $y_r^{(14)}$ which differ from those in the more accurate values of $\mathbf{E}_r(1)$ are printed in italic type, and the difference $\epsilon_r^{(14)}$ between the two values is recorded, in units of the 9th decimal place, in the penultimate column of the upper part of table 1. As expected, this error does not exceed 2×10^{-8} in absolute value within the wanted range $r = 1(1)10$.

[4] The more precise criterion of section 5 requires that the sum of $p_{10}e_r/(p_r p_{r+1})$ and all subsequent values in this column be less than 2×10^{-8}. Inspection of table 1 indicates that this also is satisfied for $N = 14$. This would not be the case, however, if we reduced our error tolerance in y_r from 2×10^{-8} to 1×10^{-8}.

263-048 O - 67 - 5

TABLE 1. *Anger-Weber function* $\mathbf{E}_r(1)$

r	p_r	e_r	$\frac{p_{10}e_r}{p_rp_{r+1}}$	$y_r^{(14)}$	$10^9\epsilon_r^{(14)}$	$10^9E_{14}p_r$
0	0	−0.56865 6627				
1	1	0.70458 2918		0.43816 2436		
2	2	0.70458 2918		.17174 1955		
3	7	9.61725 973		.24880 5382		
4	40	9.61725 973		.04785 0795		
5	313	$4.08141\ 237\times10^{2}$		.13400 0978		
6	3090	$4.08141\ 237\times10^{2}$		.01891 9443		
7	36767	$4.72213\ 396\times10^{4}$		.09303 2343		
8	5 11648	$4.72213\ 396\times10^{4}$		.01029 3811		
9	81 49601	$1.04236\ 156\times10^{7}$		.07166 8637	1	1
10	1461 81170	$1.04236\ 156\times10^{7}$	3.6×10^{-3}	.00650 2*117*	12	12
11	$2.91547\ 380\times10^{9}$	$3.72252\ 015\times10^{9}$	2.9×10^{-3}	.05837 *3706*	240	240
12	$6.39942\ 424\times10^{10}$	$3.72252\ 015\times10^{9}$	5.5×10^{-6}	.00447 *9865*	5279	5279
13	$1.53294\ 634\times10^{12}$	$1.95553\ 042\times10^{12}$	4.7×10^{-6}	.04*914 3054*	12 6445	12 6444
14	$3.97926\ 106\times10^{13}$	$1.95553\ 042\times10^{12}$	6.5×10^{-9}	.*00000 0000*		
15	$1.11266\ 015\times10^{15}$	$1.41863\ 843\times10^{15}$	5.6×10^{-9}			
16	$3.33400\ 119\times10^{16}$					

r	p_r	e_r	$\frac{e_r}{p_rp_{r+1}}$	
14	$3.97926\ 106\times10^{13}$	$1.95553\ 042\times10^{12}$	4.41672×10^{-17}	$E_{14}=8.24845\times10^{-17}$
15	$1.11266\ 015\times10^{15}$	$1.41863\ 843\times10^{15}$	3.82422×10^{-17}	
16	$3.33400\ 119\times10^{16}$	$1.41863\ 843\times10^{15}$	0.00399×10^{-17}	
17	$1.06576\ 772\times10^{18}$	$1.35839\ 625\times10^{18}$	$.00352\times10^{-17}$	
18	$3.62027\ 625\times10^{19}$	$1.35839\ 625\times10^{18}$	$.00000\times10^{-17}$	
19	$1.30223\ 368\times10^{21}$			

It is of interest to apply the expansions of section 5 to this example. The necessary computations for evaluating the expansion (5.02) are given in the lower part of table 1, and the values of $10^9E_{14}p_r$ appear in the final column of the upper part of the same table. They agree with $10^9\epsilon_r^{(14)}$ to within a unit.

EXAMPLE 2. *Struve functions.*

The function $\mathbf{H}_r(x)$ satisfies eq (2.01) with

$$a_r=c_r=1, \qquad b_r=\frac{2r}{x}, \qquad d_r=\frac{(\frac{1}{2}x)^r}{\sqrt{\pi}\,\Gamma\left(r+\frac{3}{2}\right)}.$$

For fixed x and large r we have ([8], sec. 10.4)

$$\mathbf{H}_r(x)\sim\frac{x}{\sqrt{2}\ \pi r}\left(\frac{ex}{2r}\right)^r. \tag{6.05}$$

Since the complementary functions of the difference equation are again the Bessel functions $J_r(x)$ and $Y_r(x)$, the conditions of section 2 are satisfied, provided that $J_0(x)\neq0$.

Let us evaluate $\mathbf{H}_r(0.1)$ to 8 significant figures for all positive integer values of r such that $|\mathbf{H}_r(0.1)|$ exceeds $\frac{1}{2}\times10^{-30}$.

The computations are shown in table 2. The value

$$e_0=\mathbf{H}_0(0.1)=0.06359\ 12700$$

TABLE 2. *Struve function* $\mathbf{H}_r(0.1)$

r	p_r	d_r	e_r	$\frac{e_r}{p_r p_{r+1}}$	$y_r^{(15)}$	$\frac{10^9 \epsilon_r^{(15)}}{y_r^{(15)}}$
0	0	0.63661 9772	0.06359 12700			
1	1	$2.12206\ 591 \times 10^{-2}$	.04237 06109		$2.12065\ 160 \times 10^{-3}$	0
2	20	$4.24413\ 182 \times 10^{-4}$	.03388 23473		$4.24211\ 125 \times 10^{-5}$	0
3	799	$6.06304\ 546 \times 10^{-6}$	.02903 79740		$6.06080\ 029 \times 10^{-7}$	−1
4	47920	$6.73671\ 718 \times 10^{-8}$	.02580 97391		$6.73467\ 605 \times 10^{-9}$	0
5	38 32801	$6.12428\ 835 \times 10^{-10}$	.02346 24212		$6.12271\ 820 \times 10^{-11}$	2
6	3832 32180	$4.71099\ 104 \times 10^{-12}$	.02165 70178		$4.70994\ 424 \times 10^{-13}$	4
7	$4.59840\ 288 \times 10^{10}$	$3.14066\ 069 \times 10^{-14}$	.02021 28155		$3.14004\ 492 \times 10^{-15}$	4
8	$6.43738\ 080 \times 10^{12}$	$1.84744\ 746 \times 10^{-16}$	.01902 35432		$1.84712\ 338 \times 10^{-17}$	−1
9	$1.02993\ 494 \times 10^{15}$	$9.72340\ 768 \times 10^{-19}$	.01802 20955		$9.72186\ 442 \times 10^{-20}$	0
10	$1.85381\ 852 \times 10^{17}$	$4.63019\ 413 \times 10^{-21}$	.01716 37415		$4.62952\ 313 \times 10^{-22}$	4
11	$3.70753\ 405 \times 10^{19}$	$2.01312\ 788 \times 10^{-23}$	.01641 73675		$2.01285\ 948 \times 10^{-24}$	4
12	$8.15638\ 953 \times 10^{21}$	$8.05251\ 152 \times 10^{-26}$	.01576 05733		$8.05151\ 746 \times 10^{-27}$	2
13	$1.95749\ 641 \times 10^{24}$	$2.98241\ 167 \times 10^{-28}$	.01517 67673	1.5×10^{-53}	$2.98206\ 890 \times 10^{-29}$	−1
14	$5.08940\ 910 \times 10^{26}$	$1.02841\ 782 \times 10^{-30}$	.01465 33634	2.0×10^{-58}	$1.02829\ 540 \times 10^{-31}$	1 1520
15	$1.42501\ 497 \times 10^{29}$	$3.31747\ 684 \times 10^{-33}$	.01418 06180	2.3×10^{-63}	0	
16	$4.27499\ 402 \times 10^{31}$					

was extracted from [9], and confirmed by evaluation of the expansion

$$\mathbf{H}_r(x) = (\tfrac{1}{2}x)^{r+1} \sum_{s=0}^{\infty} \frac{(-)^s (\tfrac{1}{2}x)^{2s}}{\Gamma\left(s+\frac{3}{2}\right) \Gamma\left(s+r+\frac{3}{2}\right)}. \tag{6.06}$$

The largest of the wanted values of r was determined by the criterion

$$|e_r/p_{r+1}| > \tfrac{1}{2} \times 10^{-30} \quad \text{and} \quad |e_{r+1}/p_{r+2}| \leqq \tfrac{1}{2} \times 10^{-30};$$

compare (5.03). This gave $r = 13$. Next, we have from (5.01), (5.02), and (5.03),

$$\frac{\epsilon_r^{(N)}}{y_r} \doteqdot \frac{p_r p_{r+1}}{e_r} \frac{e_N}{p_N p_{N+1}}.$$

From table 2 we see that the right of this relation is an increasing function of r, hence N is the least value for which

$$\frac{e_N}{p_N p_{N+1}} \leqq (\tfrac{1}{2} \times 10^{-8}) \frac{e_{13}}{p_{13} p_{14}}.$$

From the entries in the column headed $e_r/(p_r p_{r+1})$ we see immediately that this gives $N = 15$.

The values of $y_r^{(15)}$, computed from (4.04), appear in the penultimate column of the table. For $r \leqq 13$ they are the required approximations to $\mathbf{H}_r(0.1)$. Again, more accurate values were obtained by automatic computation with a higher value of $N(26)$, and also by evaluation of the expansion (6.06) for $r = 1(1)15$. In the final column the relative error $\epsilon_r^{(15)}/y_r^{(15)}$ is given in units of the 9th decimal place. As expected, it lies within the stipulated limit $\frac{1}{2} \times 10^{-8}$ in the required range.

7. Propagation of Rounding Errors

In addition to the truncation error $\epsilon_r^{(N)}$ which has been analyzed in sections 4 and 5, the other possible sources of error in the final solution are the rounding errors introduced during the calculations. Since the computing process is essentially the solution of a finite system of linear algebraic equations, the nature of the transmission of these errors is available from general theory [10]

chapter 4; [6], chapter 9. However, because of special features of the present problem, including the fact that in our form of elimination the absolute values of the multipliers are not bounded by unity, some comments on the effects of rounding errors may be helpful.

Consider first the computation of the sequence p_r. From the conditions $p_0=0$, $p_1=1$, we see that in terms of the fundamental solutions f_r and g_r of section 2

$$p_r=(f_0g_r-g_0f_r)/(f_0g_1-g_0f_1), \tag{7.01}$$

the denominator here necessarily being nonzero since f_r and g_r are independent solutions of the difference equation. By hypothesis, $f_0\neq 0$; therefore p_r always contains a multiple of g_r. And since $f_r/g_r\to 0$ as $r\to\infty$, p_r ultimately becomes proportional to g_r when a fixed number of significant figures is maintained in the computations.

Each rounding error in the formation of the p_r can be regarded as introducing unwanted small multiples of f_r and g_r. Ultimately, the former dies out in comparison with the latter; the error is then propagated at the same rate as p_r itself. Before this stage is attained, however, some loss of accuracy is possible. If the value of $|f_0|$ is unduly small compared with $|g_0f_1/g_1|$, then from (7.01) we see that initially p_r behaves like a multiple of f_r. But the rounding errors are still propagated in proportion to g_r, and this generally causes a steady loss of significant figures. The loss ceases when the term f_0g_r in (7.01) overtakes g_0f_r in magnitude, at which stage the computation becomes completely stable.

It should be realized that this loss of accuracy is not attributable to the method of computation, but to the fact that, as a rule, the whole problem is ill-posed when $|f_0|$ is small compared with $|g_0f_r/g_r|$ for at least one value of r. For from (2.06) we see that

$$\left|\frac{\delta y_r}{y_r}\right|=\left|\frac{f_r\delta k}{f_0y_r}\right|\gg\left|\frac{g_r\delta k}{g_0y_r}\right|, \tag{7.02}$$

where δy_r is the change in y_r consequent upon an arbitrary change δk in the value of k. Since $|g_r/g_0|$ is generally large compared with $|y_r|$ (see (2.04)), the relative error in y_r is very sensitive to rounding errors in the given value of k.

Examples 1 and 2 of section 6 would be ill-posed in this way if the chosen value of x were close to a zero of $J_0(x)$, say $x=5.52$. This would become apparent at the beginning of the computations: the early p_r would diminish in size, in contrast to the behavior they exhibit in tables 1 and 2.

The difficulty could be overcome in these and other examples by carrying out the computation of k and p_r to higher precision, and making the necessary prolongation of the recurrences until the criteria of sections 4 and 5 for terminating them are met.

If the value of y_1 can be found, however, a preferable alternative is to apply the algorithm of sections 3 and 4 with the given y_1 as normalizing value, instead of $y_0=k$. In effect, this means that the recurrences (4.05) are begun with $p_1=0$, $p_2=1$, and $e_1=y_1$. Subsequently the value of y_0 can be computed from y_1 and y_2 by a single backward application of (2.01).

The other part of the elimination process is the computation of the right-hand sides e_r. From (4.05) we see that in the inhomogeneous case instability could arise from this source if there were persistent heavy cancellation between a_re_{r-1} and d_rp_r. No naturally occurring examples of this phenomenon have been encountered so far, however.

Lastly, we see from (4.04) that a rounding error introduced in $y_s^{(N)}$ during the back-substitution is multiplied by the factor p_r/p_s when it is transmitted to $y_r^{(N)}$ $(r<s)$. Except when the problem is ill-posed, this factor decays with diminishing r at a faster rate than $y_r^{(N)}$ itself, because p_r contains a substantial multiple of g_r, and y_r contains no multiple of this function.

Summarizing this section, we have shown that unstable transmission of rounding errors can occur only when the original problem is ill-posed or when heavy cancellation takes place during the calculation of e_r from the second of (4.05).

8. Comparison With the Algorithms of Miller and Shintani

In section 4 we solved the set of eqs (3.01) and (3.02) by eliminating the variables in the order $y_1^{(N)}, y_2^{(N)}, \ldots, y_{N-2}^{(N)}$: we may call this *forward elimination*. Suppose now that these variables are eliminated in the reverse order: *backward elimination*. The resulting set of pivotal equations can be expressed in the form

$$u_{r+1}^{(N)} y_r^{(N)} - u_r^{(N)} y_{r+1}^{(N)} = v_r^{(N)} \qquad (r = N-2, N-3, \ldots, 0), \tag{8.01}$$

where the quantities $u_r^{(N)}$ and $v_r^{(N)}$ are defined by

$$u_{N-1}^{(N)} = 1, \qquad u_{N-2}^{(N)} = b_{N-1}/a_{N-1}, \qquad v_{N-2}^{(N)} = d_{N-1}/a_{N-1}, \tag{8.02}$$

and

$$u_{r-1}^{(N)} = \frac{b_r u_r^{(N)} - c_r u_{r+1}^{(N)}}{a_r}, \qquad v_{r-1}^{(N)} = \frac{c_r v_r^{(N)} + d_r u_r^{(N)}}{a_r}, \qquad (r < N-1). \tag{8.03}$$

(It should be observed that $u_r^{(N)}$ and $v_r^{(N)}$ depend on N as well as r, unlike the p_r and e_r of section 4.)

The last of eqs (8.01) is used to begin the back-substitution. It yields

$$y_1^{(N)} = (u_1^{(N)} k - v_0^{(N)})/u_0^{(N)},$$

where k is again the given value of y_0. Then $y_2^{(N)}, y_3^{(N)}, \ldots, y_{N-1}^{(N)}$ may be computed by successive application of (8.01) with $r = 1, 2, \ldots, N-2$.

Thus the elimination process consists of constructing a sequence $u_r^{(N)}$ which satisfies the homogeneous form of the given difference equation (2.01) and the conditions $u_N^{(N)} = 0$, $u_{N-1}^{(N)} = 1$. This is exactly the first stage of Miller's algorithm [1], [3]: the $u_r^{(N)}$ are the so-called trial values. And in the homogeneous case, given by $d_r = 0$, all the quantities $v_r^{(N)}$ vanish, causing the formulas (8.01) for back-substitution to reduce to

$$y_r^{(N)} = \frac{u_r^{(N)}}{u_{r-1}^{(N)}} y_{r-1}^{(N)} = \frac{u_r^{(N)}}{u_{r-1}^{(N)}} \frac{u_{r-1}^{(N)}}{u_{r-2}^{(N)}} \cdots \frac{u_1^{(N)}}{u_0^{(N)}} y_0^{(N)} = \frac{k}{u_0^{(N)}} u_r^{(N)}.$$

This is the second stage of the Miller algorithm: $k/u_0^{(N)}$ is the normalizing factor.

Accordingly, *in the homogeneous case the Miller recurrence algorithm can be regarded as the solution of the set of equations (3.01) and (3.02), with* $d_r = 0$, *by backward elimination.* In the inhomogeneous case the solution by backward elimination, described above, can be regarded as a generalization of the Miller algorithm.

Compared with the forward elimination process of section 4, the Miller algorithm suffers from the disadvantages that it does not determine automatically the correct value of N, and if a second value of N is used as a check on the adequacy of the original value, then the computations must begin afresh. The advantage of the Miller algorithm is that the process of back-substitution is less laborious; this advantage is restricted to the homogeneous case, however, and is offset if more than one trial value of N has to be used.

The method Shintani [5][5] has developed for solving second-order linear difference equations in the homogeneous case consists of the use of the Miller algorithm preceded by two forward recurrence processes to determine the optimum value of N. In the present notation, Shintani takes $a_r = 1$ and $d_r = 0$. His formulas for forward recurrence are given by ([5], Theorem 1)

[5] See also [4], section 4.

$$P_{r+1}(\nu) = b_{r+1}P_r(\nu) - c_rP_{r-1}(\nu), \tag{8.04}$$

where $\nu = 0$ or 1, and

$$P_{-1}(0) = 0, \qquad P_0(0) = 1; \qquad P_0(1) = 0, \qquad P_1(1) = 1. \tag{8.05}$$

It is easily verified, for example, that the quantities $P_r(0)$ appear when the forward elimination procedure is applied to eqs (4.01) with $d_r = 0$ and the multipliers chosen in such a way that the constant value $-k$ is preserved on the right-hand sides. The resulting pivotal equations are in fact

$$-P_r(0)y_r^{(N)} + c_rP_{r-1}(0)y_{r+1}^{(N)} = -k. \qquad (r = 1, 2, \ldots, N-1). \tag{8.06}$$

In our notation

$$P_r(0) = c_1c_2 \ldots c_rp_{r+1}. \tag{8.07}$$

From the computational standpoint, the evaluation of Shintani's sequence $P_r(0)$ may be compared with the evaluation of our sequence p_r, the evaluation of his $P_r(1)$ with our e_r, and the application of the Miller algorithm with our process of back-substitution. In the first stage the computing effort is identical, but in the second and third stages our method requires considerably less effort.

9. More General Form of Normalizing Condition

Let us consider now the solution of the difference eq (2.01) when (2.05) is replaced by the more general normalizing condition

$$m_0y_0 + m_1y_1 + m_2y_2 + \ldots = k, \tag{9.01}$$

in which m_0, m_1, . . ., and k are given constants. We again suppose that the general solution of (2.01) has the form (2.02), but instead of the conditions imposed on f_r, g_r, and h_r in section 2, we assume that

$$\left| \sum_{r=0}^{N} m_rg_r \right| \to \infty \quad \text{as} \quad N \to \infty, \tag{9.02}$$

and

$$\sum_{r=0}^{\infty} m_rf_r = F, \qquad \sum_{r=0}^{\infty} m_rh_r = H, \tag{9.03}$$

where F and H are finite, and $F \neq 0$. Then (2.01) has a unique solution fulfilling (9.01). It is given by

$$y_r = \frac{k-H}{F} f_r + h_r; \tag{9.04}$$

compare (2.06).

The obvious extension of the approach of section 3 is to solve the system of linear algebraic equations given by

$$a_ry_{r-1}^{(N)} - b_ry_r^{(N)} + c_ry_{r+1}^{(N)} = d_r \qquad (r = 1, 2, \ldots, N-1), \tag{9.05}$$

$$\sum_{r=0}^{N} m_ry_r^{(N)} = k, \tag{9.06}$$

and

$$y_N^{(N)} = 0. \tag{9.07}$$

THEOREM 2. *In addition to the other conditions of this section, assume that for all sufficiently large* N *the system of equations* (9.05), (9.06), *and* (9.07) *has a solution, that* $g_N \neq 0$, *and that*

$$\frac{f_N}{g_N}\sum_{r=0}^{N} m_r g_r \to 0, \qquad \frac{h_N}{g_N}\sum_{r=0}^{N} m_r g_r \to 0, \qquad (N \to \infty). \tag{9.08}$$

Then if r *is fixed and* $N \to \infty$, $y_r^{(N)} \to y_r$.

This result may be established by expressing $y_r^{(N)}$ in the form

$$y_r^{(N)} = A_N f_r + B_N g_r + h_r. \tag{9.09}$$

Using (9.06) and (9.07), we find that

$$A_N = \frac{h_N \sum_{r=0}^{N} m_r g_r - g_N\left(\sum_{r=0}^{N} m_r h_r - k\right)}{g_N \sum_{r=0}^{N} m_r f_r - f_N \sum_{r=0}^{N} m_r g_r}, \qquad B_N = \frac{f_N\left(\sum_{r=0}^{N} m_r h_r - k\right) - h_N \sum_{r=0}^{N} m_r f_r}{g_N \sum_{r=0}^{N} m_r f_r - f_N \sum_{r=0}^{N} m_r g_r}.$$

In consequence of the assumed conditions, the denominators are asymptotic to Fg_N as $N \to \infty$. Hence $A_N \to (k-H)/F$. Next, the assumed conditions imply that f_N/g_N and h_N/g_N both tend to zero. Hence $B_N \to 0$. Comparison of (9.04) and (9.09) completes the proof.

When the forward elimination process of section 4 is applied to eqs (9.05), (9.06), and (9.07), the following pivotal equations are obtained:

$$p_{r+1} y_r^{(N)} - p_r y_{r+1}^{(N)} + q_r\left(\sum_{s=r+1}^{N-1} m_s y_s^{(N)}\right) = e_r \qquad (r = 0, 1, \ldots, N-1), \tag{9.10}$$

(compare (4.04)), where

$$q_0 = 1, \qquad q_r = \frac{a_1 a_2 \ldots a_r}{c_1 c_2 \ldots c_r} \qquad (r \geq 1), \tag{9.11}$$

$$p_0 = 0, \qquad p_1 = m_0, \qquad e_0 = k, \tag{9.12}$$

and, if $r \geq 1$,

$$p_{r+1} = \frac{b_r p_r - a_r p_{r-1}}{c_r} + q_r m_r, \qquad e_r = \frac{a_r e_{r-1} - d_r p_r}{c_r}. \tag{9.13}$$

In consequence of (9.07), the final equation of the form (9.10) reduces to

$$p_N y_{N-1}^{(N)} = e_{N-1}. \tag{9.14}$$

This yields the value of $y_{N-1}^{(N)}$; thence $y_{N-2}^{(N)}, y_{N-3}^{(N)}, \ldots, y_0^{(N)}$ may be computed from (9.10) by back-substitution.

The value of N may be determined in a similar way to that suggested in section 4. Suppose, for example, that all nonvanishing values of y_r are needed to a fixed number of decimal places,

D, say—a common form of requirement with the present type of normalizing condition. Then N is determined by the condition

$$|e_N/p_{N+1}| < \tfrac{1}{2} \times 10^{-D}, \tag{9.15}$$

provided that $|p_r| \leq |p_N|$ when $r \leq N$, and also

$$|p_r/q_{r-1}| > |m_r|, |m_{r+1}|, \ldots, |m_N|. \tag{9.16}$$

EXAMPLE 3. *Bessel functions.*

Let us evaluate $J_0(x), J_1(x), \ldots$, for $x=5$ to 5 decimal places, by use of the relations

$$J_{r-1}(x) - (2r/x)\ J_r(x) + J_{r+1}(x) = 0, \tag{9.17}$$

and

$$J_0(x) + 2J_2(x) + 2J_4(x) + \ldots = 1. \tag{9.18}$$

In the present notation, we have

$$a_r = c_r = 1, \qquad b_r = 2r/x, \qquad d_r = 0,$$

$$m_0 = 1, \qquad m_1 = m_3 = \ldots = 0, \qquad m_2 = m_4 = \ldots = 2, \qquad k = 1.$$

Accordingly, eqs (9.11) through (9.13) yield

$$q_r = 1, \qquad e_r = 1, \qquad p_0 = 0, \qquad p_1 = 1,$$

and

$$p_{r+1} = b_r p_r - p_{r-1} + m_r. \tag{9.19}$$

Table 3 gives the values of p_r correct to 6 significant figures. The criterion (9.15) suggests that N be taken as the least value of r for which $|p_{r+1}| > 2 \times 10^5$. This gives [6] $N = 14$. The column of values of $y_r^{(14)}$ is then generated upwards by use of (9.10), starting with $y_{14}^{(14)} = 0$. These are the required approximations to $J_n(5)$: their differences, $\epsilon_r^{(14)}$, from the true values are recorded in the final column in units of the 5th decimal place. The agreement is satisfactory.

TABLE 3. *Bessel function* $J_r(5)$

r	b_r	m_r	p_r	$y_r^{(14)}$	$\sum^{13} m_s y_s^{(14)}$	$10^5 \epsilon_r^{(14)}$
0	0.0	1	0	−0.17758		−2
1	0.4	0	1	−.32758	1.17758	0
2	0.8	2	0.4	.04655	1.17758	2
3	1.2	0	1.32	.36482	1.08448	1
4	1.6	2	1.184	.39123	1.08448	0
5	2.0	0	2.5744	.26114	0.30202	0
6	2.4	2	3.9648	.13105	.30202	0
7	2.8	0	8.94112	.05338	.03992	0
8	3.2	2	21.0703	.01841	.03992	0
9	3.6	0	60.4838	.00552	.00310	0
10	4.0	2	196.671	.00147	.00310	0
11	4.4	0	728.200	.00035	.00016	0
12	4.8	2	2007.41	.00008	.00016	0
13	5.2	0	13709.4	.00001	.00000	1
14	5.6	2	68281.5	.00000	.00000	0
15			368669.			

[6] When the condition (9.16) is violated—as it is in this example near the beginning of the range—it would be safer in practice to take N slightly higher than the value predicted by (9.15).

It may be noted that estimates of the optimum value of N for generating Bessel functions from (9.17) and (9.18) by Miller's algorithm have been computed by Makinouchi [11] for $x=0.01(.01)0.1(.1)1(1)10(10)100$ and precisions of 9, 10, 18, 20, and 30 significant figures. These values were obtained by use of the asymptotic approximations (6.04) above. In constructing a program for generating the $J_r(x)$ for arbitrary x and arbitrary precision, however, it would be simpler to determine the optimum N by use of (9.15) (or (4.11)). The resulting gain would tend to offset the extra effort needed in applying the back-substitution relation (9.10) compared with the normalizing of the trial values in the Miller algorithm.

10. Bounds for the Truncation Error

In order to obtain strict bounds for the truncation error associated with the algorithm of section 9, we proceed as in section 5. Write, temporarily,

$$\eta_r = y_r^{(N+1)} - y_r^{(N)}. \tag{10.01}$$

Then from (9.10) we obtain

$$p_{r+1}\eta_r = p_r\eta_{r+1} - q_r(m_{r+1}\eta_{r+1} + \ldots + m_N\eta_N) \qquad (r<N). \tag{10.02}$$

Therefore

$$|\eta_r| \leq \rho_r(|\eta_{r+1}| + |\eta_{r+2}| + \ldots + |\eta_N|) \qquad (r<N), \tag{10.03}$$

where ρ_r is the greater of

$$\left|\frac{p_r - q_r m_{r+1}}{p_{r+1}}\right| \text{ and } \left|\frac{q_r}{p_{r+1}}\right| \sup_{2\leq s\leq\infty} |m_{r+s}|. \tag{10.04}$$

Equations (9.07), (9.14), and (10.01) yield $\eta_N = e_N/p_{N+1}$. From this result and (10.03) we may verify that

$$|\eta_{N-1}| \leq \rho_{N-1}|e_N/p_{N+1}|, \tag{10.05}$$

and thence by induction that

$$|\eta_r| \leq \rho_r(1+\rho_{r+1})(1+\rho_{r+2}) \ldots (1+\rho_{N-1})\,|e_N/p_{N+1}| \qquad (r\leq N-2). \tag{10.06}$$

The left-hand side of the last relation is $|y_r^{(N+1)} - y_r^{(N)}|$. Replacing N by $N+1, N+2, \ldots$, in turn and summing, and applying Theorem 2, we find that

$$|\epsilon_r^{(N)}| \leq \rho_r(1+\rho_{r+1})(1+\rho_{r+2}) \ldots (1+\rho_{N-1})E_N \qquad (r\leq N-2), \tag{10.07}$$

where

$$\epsilon_r^{(N)} = y_r - y_r^{(N)}, \tag{10.08}$$

and

$$E_N = \left|\frac{e_N}{p_{N+1}}\right| + (1+\rho_N)\left|\frac{e_{N+1}}{p_{N+2}}\right| + (1+\rho_N)(1+\rho_{N+1})\left|\frac{e_{N+2}}{p_{N+3}}\right| + \ldots, \tag{10.09}$$

provided that the last series converges. Similarly

$$|\epsilon_{N-1}^{(N)}| \leq \rho_{N-1}E_N, \qquad |y_N| \leq E_N. \tag{10.10}$$

The results (10.07) and (10.10) are strict bounds for the truncation error, in contrast to the expansion of section 5 which is exact (for the algorithm of sections 3 and 4). Often the bound (10.07) is a considerable overestimate.[7] Thus in Example 3, the right-hand side of (10.07) or (10.10) has the following values for $N = 14$, in units of the 5th decimal place:

$$568,\ 237,\ 29,\ 12,\ 3,\ 1,\ 1,\ \text{then zero for } r = 7,\ 8,\ \ldots,\ 14.$$

In consequence, if N is determined by the criterion that for each required value of r the right-hand sides of (10.07) and (10.10) must not exceed the specified tolerance in y_r, then the resulting value is perfectly safe but often unnecessarily high. Applied to Example 3, this criterion yields $N = 18$, compared with the value 14 which we used and found to be quite adequate.

In the next section we give an alternative formulation of the algorithm of section 9. Although perhaps less elegant, it generally yields a sharper assessment of the truncation error than that of this section.

11. Alternative Method for the General Normalizing Condition

The algorithm of sections 3 and 4 can be applied to the problem of section 9 in the following way. First, we construct a solution f_r of the homogeneous form of the given equation (2.01). The choice of this solution is arbitrary, provided that the first of (2.03) is satisfied. Then by means of an additional back-substitution we construct an arbitrary solution h_r of (2.01) itself. The required solution y_r may then be computed from (9.04), in which k is defined by (9.01), and F, H by (9.03). In the case when the given difference eq (2.01) is itself homogeneous, only the solution f_r need be computed, and (9.04) reduces to

$$y_r = (k/F) f_r. \tag{11.01}$$

The simplest choice of the normalizing conditions needed for constructing f_r and h_r is given by

$$f_0 = 1, \qquad h_0 = 0. \tag{11.02}$$

The first of these may be an inconvenient or even impossible condition, however; in this event we may follow the suggestion given in section 7 and use instead

$$f_1 = 1, \qquad h_1 = 0. \tag{11.03}$$

To assess the truncation error in the final solution y_r, let $\varphi_r^{(N)}$ and $\theta_r^{(N)}$ be the truncation errors in the approximations $f_r^{(N)}$ and $h_r^{(N)}$ to f_r and h_r; thus

$$f_r = f_r^{(N)} + \varphi_r^{(N)}, \qquad h_r = h_r^{(N)} + \theta_r^{(N)}. \tag{11.04}$$

Bounds for $\varphi_r^{(N)}$ and $\theta_r^{(N)}$ are computable from the expansions of section 5. From (9.04) we have

$$y_r = \frac{k - H_N - \tau_N}{F_N + \sigma_N}(f_r^{(N)} + \varphi_r^{(N)}) + h_r^{(N)} + \theta_r^{(N)} \qquad (r = 0,\ 1,\ \ldots,\ N), \tag{11.05}$$

where F_N, H_N are the computed quantities

$$F_N = \sum_{r=0}^{N-1} m_r f_r^{(N)}, \qquad H_N = \sum_{r=0}^{N-1} m_r h_r^{(N)}, \tag{11.06}$$

[7] This can be traced to the fact that over most of the range the second of the two quantities (10.04) is usually very much smaller than the first.

and σ_N, τ_N assessable errors [8]

$$\sigma_N = \sum_{r=0}^{N} m_r \varphi_r^{(N)} + \sum_{r=N+1}^{\infty} m_r f_r, \tag{11.07}$$

$$\tau_N = \sum_{r=0}^{N} m_r \theta_r^{(N)} + \sum_{r=N+1}^{\infty} m_r h_r. \tag{11.08}$$

If $F_N \neq 0$, then to the first order of small quantities the truncation error in the formula

$$y_r \doteq \frac{k - H_N}{F_N} f_r^{(N)} + h_r^{(N)} \tag{11.09}$$

is composed of three parts:

$$\frac{k - H_N}{F_N} \varphi_r^{(N)}, \qquad -\left\{\tau_N + \frac{\sigma_N (k - H_N)}{F_N}\right\} \frac{f_r^{(N)}}{F_N}, \qquad \theta_r^{(N)}, \qquad (r \leq N). \tag{11.10}$$

In the homogeneous case they reduce to two:

$$\frac{k}{F_N} \varphi_r^{(N)}, \qquad -\frac{k\sigma_N}{F_N^2} f_r^{(N)}, \qquad (r \leq N). \tag{11.11}$$

The first of (11.11) is the normalized multiple of the truncation error in the formula $f_r \doteq f_r^{(N)}$; the second of (11.11) is a fixed relative error arising from the approximate representation of the normalizing factor k/F by k/F_N.

The equivalence of the method of this section to the algorithm of section 9 can be seen from the fact that the function on the right of (11.09) is exactly the solution of the set of eqs (9.05), (9.06), and (9.07).

EXAMPLE 4.[9]

Let us compute to 5 decimal places the solution of the homogeneous equation

$$(2r-1)y_{r-1} - 12ry_r + (2r+1)y_{r+1} = 0, \tag{11.12}$$

satisfying the condition

$$\tfrac{1}{2} y_0 + y_1 + y_2 + y_3 + \ldots = 1. \tag{11.13}$$

In the notation of sections 2 and 9 we have

$$a_r = 2r-1, \qquad b_r = 12r, \qquad c_r = 2r+1, \qquad d_r = 0, \qquad m_0 = \tfrac{1}{2}, \qquad m_r = 1 \quad (r > 0), \qquad k = 1.$$

The computations are shown in table 4. Values of p_r were generated from $p_0 = 0$, $p_1 = 1$, and (11.12) when $r > 1$, correct to 6 significant figures. With $e_0 = 1$ (compare the first of (11.02)), we find from the second of (4.05) that $e_r = 1/(2r+1)$. The least value of r for which $e_r/p_{r+1} < \frac{1}{2} \times 10^{-5}$ is 7; in

[8] See also [7].

[9] [3], section 5.

[10] An alternative way of estimating N in this particular example is to observe that, except for small r, the wanted solution of (11.12) behaves roughly like $A\lambda^{-r}$, where A is a constant and $\lambda^2 - 6\lambda + 1 = 0$, giving $\lambda \doteq 3 + \sqrt{8} = 5.8$. Assuming that A is of order unity, as is reasonable in view of the condition (11.13), we find that $N \doteq 5/\log_{10} \lambda \doteq 7$. This method of estimation is somewhat less certain than the one we have used, and it is not universally applicable.

accordance with (4.12) this is the value [10] to ascribe to N. The back-substitution process for the determination of $f_r^{(7)}$ is given by $f_7^{(7)}=0$, and

$$p_{r+1}\, f_r^{(7)} = p_r\, f_{r+1}^{(7)} + e_r \qquad (r=6,\ 5,\ \ldots,\ 0);$$

compare (4.04). Division of $f_r^{(7)}$ by $F_7=0.599069$, computed from the first of (11.06), yields the wanted approximations $y_r^{(7)}$ to y_r.

TABLE 4

r	p_r	e_r	$f_r^{(7)}$	$y_r^{(7)}$
0	0	1.000000	1.000000	1.66926
1	1	0.333333	0.086107	0.14373
2	4	.200000	.011094	.01852
3	18.6	.142857	.001587	.00265
4	92.8	.111111	.000238	.00040
5	480.467	.090909	.000037	.00006
6	2544.80	.076923	.000006	.00001
7	13687.7	.066667	.000000	.00000
8	74445.6			

The example is now complete, but it is of interest to illustrate the error analysis of this section. Accordingly, the whole calculation was repeated twice, keeping four extra significant figures throughout. In the first repetition the same value $N=7$ was used. In the second repetition a new N was determined by the condition $|e_N/p_{N+1}| < \frac{1}{2}\times 10^{-9}$; this gave $N=12$.

The results appear in table 5. The column headed $10^9\epsilon_r^{(7)}$ gives the difference of $10^9 y_r^{(7)}$ from the more accurate values $10^9 y_r^{(12)}$. The next columns give $10^9\varphi_r^{(7)}/F_7$ and $-10^9\sigma_7 f_r^{(7)}/F_7^2$; the value of $\varphi_r^{(7)}$ was obtained by subtracting $f_r^{(7)}$ from $f_r^{(12)}$, and σ_7 computed from (11.07), using the values of $f_r^{(12)}$ for f_r when $r \geq 8$. As expected, the values of $10^9\epsilon_r^{(7)}$ are in good agreement with the sum of the entries on the same row in the following two columns.

TABLE 5

r	$f_r^{(12)}$	$y_r^{(12)}$	$f_r^{(7)}$	$y_r^{(7)}$	$10^9\epsilon_r^{(7)}$	$10^9\frac{\phi_r^{(7)}}{F_7}$	$-10^9\frac{\sigma_7 f_r^{(7)}}{F_7^2}$
0	1.00000 00000	1.66925 3684	1.00000 00000	1.66925 7339	−3655	0	−3655
1	0.08610 68379	0.14373 4156	0.08610 68378	0.14373 4471	−315	0	−315
2	.01109 40183	.01851 8731	.01109 40180	.01851 8771	−40	1	−41
3	.00158 71852	.00264 9415	.00158 71839	.00264 9418	−3	2	−6
4	.00023 83677	.00039 7896	.00023 83614	.00039 7887	9	11	−1
5	.00003 68169	.00006 1457	.00003 67845	.00006 1403	54	54	0
6	.00000 57914	.00000 9667	.00000 56199	.00000 9381	286	286	0
7	.00000 09228	.00000 1540	.00000 00000	.00000 0000	1540	1540	0
8	.00000 01485	.00000 0248					
9	.00000 00241	.00000 0040					
10	.00000 00039	.00000 0007	$F_7=0.59906\ 88055$		$\sigma_7=0.00000\ 13118$		
11	.00000 00006	.00000 0001	$F_{12}=0.59907\ 01173$				
12	.00000 00000	.00000 0000					

12. Summary

In this paper we have described a new algorithm for computing the solution y_r of any second-order linear difference equation, homogeneous or inhomogeneous, which is applicable when simple forward recurrence (and possibly also backward recurrence) cannot be used because of instability.

In the first part (secs. 2–8) we considered the case in which the wanted solution y_r has a specified value at the beginning of the range $r=0$, and an appropriate convergence condition as $r\to\infty$. In this case the algorithm is based on the solution of a finite number, N, of simultaneous linear algebraic equations of tridiagonal form by forward elimination. As $N\to\infty$ the solution $y_r^{(N)}$ of these equations converges to y_r (sec. 3). In sections 4 and 5 it was shown that during the process of computing $y_r^{(N)}$ the minimum value of N necessary to achieve specified tolerance in $|y_r-y_r^{(N)}|$ emerges automatically. Analyses of the truncation error and of the propagation of rounding errors were made in sections 5 and 7. The former leads to a convergent series expansion for y_r; the latter shows that the method of computation is quite stable, unless the problem itself is ill-posed. Numerical examples (sec. 6) illustrated the algorithm and confirmed the error analyses.

In section 8 it was shown that the well-known algorithm of J. C. P. Miller for the homogeneous case can be regarded as the computation of $y_r^{(N)}$ by backward elimination, taking a guessed value of N. It was also shown that the recent extension of Miller's algorithm by Shintani is related to the process of forward elimination.

In the second part of the paper (secs. 9–11) a more general form of normalizing condition for y_r was considered. An extended form of the algorithm was developed in section 9 and applied to a numerical example in the same section. In section 10 bounds for the truncation error were given and discussed. In the concluding section (sec. 11) it was shown that the more general problem can also be solved by application of the original algorithm of sections 3 and 4.

It is hoped that the results of this paper will prove to be of considerable usefulness in the computation of special functions from recurrence relations, in the solution of ordinary differential equations in Chebyshev series by Clenshaw's method, and in the solution of the discretized form of boundary-value problems in ordinary differential equations when one boundary is at infinity. In the last two connections, it may be possible to extend the present approach to difference equations of order higher than the second.

The writer thanks C. W. Clenshaw, G. F. Miller, and D. L. Yarmush for valuable comments on this work, and also his wife, Mrs. G. E. Olver, for carrying out the desk and automatic computations of the numerical examples, only a few of which are included in the paper. The automatic computation was supported by National Institutes of Health Grant No. NB05613.01.

13. References

[1] British Association for the Advancement of Science, Bessel functions—Part II, Mathematical Tables, v. **10**. (Cambridge University Press, 1952.)

[2] Clenshaw, C. W., The numerical solution of linear differential equations in Chebyshev series, Proc. Camb. Philos. Soc. **53**, 134–149 (1957).

[3] Olver, F. W. J., Error analysis of Miller's recurrence algorithm, Math. Comp. **18**, 65–74 (1964).

[4] Gautschi, W., Computational aspects of three-term recurrence relations, S.I.A.M. Rev. **9**, 24–82 (1967).

[5] Shintani, H., Note on Miller's recurrence algorithm, J. Sci. Hiroshima Univ. Ser. A–I **29**, 121–133 (1965).

[6] Fox, L., The numerical solution of two-point boundary problems in ordinary differential equations. (Oxford University Press, 1957.)

[7] Olver, F. W. J., Bounds for the solutions of second-order linear difference equations, J. Res. NBS.

[8] Watson, G. N., Theory of Bessel functions, Second edition. (Cambridge University Press, 1944.)

[9] Robinson, C., A numerical and analytical investigation of Struve's function. Thesis, London Univ. (1948).

[10] Wilkinson, J. H., The algebraic eigenvalue problem. (Oxford University Press, 1965.)

[11] Makinouchi, S., Note on the recurrence techniques for the calculation of Bessel functions $J_\nu(x)$, Tech. rep. Osaka Univ. **15**, 185–201 (1965).

[12] Oliver, J., Relative error propagation in the recursive solution of linear recurrence relations, Num. Math. **9**, 323–340 (1967).

(Paper 71B2&3–206)