

## An Extension of Olver's Method for the Numerical Solution of Linear Recurrence Relations

By J. R. Cash

**Abstract.** An algorithm is developed for computing the solution of a class of linear recurrence relations of order greater than two when unstable error propagation prevents the required solution being found by direct forward recurrence. By abandoning an appropriate number of initial conditions the original problem may be replaced by an inexact but well-conditioned boundary value problem, and in certain circumstances the solution of this new problem is a good approximation to the required solution of the original problem. The required solution of this reposed problem is generated using an algorithm based on Gaussian elimination, and a technique developed by Olver is extended to estimate automatically the truncation error of the proposed algorithm.

**1. Introduction.** Consider the numerical solution of the  $m$ th order linear recurrence relation

$$(1.1) \quad \sum_{j=0}^m \alpha_j(r) y_{r+j} = f(r),$$

where the  $\alpha_j(r)$ 's and  $f(r)$  are given sequences of the nonnegative integer variable  $r$ . If  $m$  initial conditions of the form  $y_j = k_j, j = 1(1)m$ , are prescribed, the required solution of (1.1) is in theory completely specified; but it may not be possible to compute the desired solution using these initial conditions, owing to an unstable build up of rounding error (see [2], [7] for example). In circumstances such as these it is sometimes possible to replace the original problem by a well-conditioned boundary value problem which has a solution closely approximating the required solution of (1.1). The ability to recognize potential instability and to repose our problem so that the required solution may be stably generated requires us to have some additional knowledge regarding the behavior of the required solution. In view of this, two of the main applications of our algorithm will be in the numerical solution of ordinary differential equations and in the solution of equations of the form (1.1) where one boundary condition is given at infinity. Assuming that we are in a position to recognize instability when it occurs, the first step of our algorithm is to replace the original initial values by boundary values of the form

$$y_j = k_j, \quad j = 1(1)q,$$
$$y_{N+j} = 0, \quad j = 0(1)m - 1 - q.$$

The conditions under which the required solution of (1.1) with the original initial conditions is closely approximated by a solution of (1.1) with these new boundary

---

Received September 16, 1976.

AMS (MOS) subject classifications (1970). Primary 65Q05; Secondary 65F05, 65L05.

Copyright © 1978, American Mathematical Society

conditions has been discussed in [8] and will not concern us here. This problem has also been considered recently for the special case  $q = 1$ ,  $f(r) \equiv 0$  by Zahar [13]. We shall assume that the required value of  $q$  is known and the problem that remains is to compute  $N$  and the required solution of the reposed problem. This type of approach, whereby the original problem is replaced by a suitable boundary value problem, is now well accepted and forms a basis for most of the more competitive algorithms for generating solutions of linear recurrence relations [5], [9], [10]. In particular, Olver [10] has developed an extremely powerful algorithm for the solution of second-order inhomogeneous recurrence relations with the 'optimal' value of  $N$  being automatically determined in all cases. One of the more competitive algorithms for computing solutions of higher-order recurrence relations is one proposed by Oliver [9]. This algorithm requires the value of  $N$  to be guessed, and then a sequence of approximations to the optimal value of  $N$  is built up at a relatively cheap computational cost. Although this algorithm is often quite efficient, it would seem to have some drawbacks especially if one of the boundary conditions is at infinity so that we have little idea in general of the required value of  $N$ , or if a more general normalizing condition of the form  $\sum_{r=1}^{\infty} \gamma_r y_r = k$  is imposed. Also, Oliver's algorithm requires the solution to be accepted if two successive approximations differ by less than a prescribed amount and, although this is a common practical procedure, it does not guarantee that the solution obtained is the required solution.

The algorithm developed in this paper stemmed from a desire to compute solutions of high-order recurrence relations when either one boundary condition is at 'infinity' or when the more general normalizing condition mentioned previously is imposed. Our aim is to develop an algorithm which yields an estimate of the optimal value of  $N$  before any back substitution has been performed. Although the algorithm which we propose in this paper does not always achieve this aim, it does so in many cases and a nonoptimal value of  $N$  can soon be recognized. Examples can no doubt be constructed for which our proposed algorithm is not as efficient as that of Oliver, and it may be that a combination of the two algorithms would be more efficient as a general purpose algorithm than either one by itself. Finally, we mention that this new approach allows us to identify positively the solution obtained as the required solution of the original problem since we are able to obtain an exact expression for the truncation error.

**2. Statement of the Algorithm.** In this section we consider the numerical solution of the third-order linear recurrence relation

$$(2.1) \quad a_r y_{r-1} + b_r y_r + c_r y_{r+1} + d_r y_{r+2} = e_r,$$

where  $a_r$ ,  $b_r$ ,  $c_r$ ,  $d_r$  and  $e_r$  are given sequences depending on the nonnegative integer variable  $r$ . Although we are particularly interested in developing algorithms for the solution of  $m$ th-order recurrence relations, we shall confine our analysis in this section to the case  $m = 3$  since the extension to higher-order equations is relatively straightforward and will be given explicitly at a later stage. The general solution of (2.1) may be written in the form

$$(2.2) \quad y_r = Af_r + Bg_r + Ch_r + p_r,$$

where  $f_r$ ,  $g_r$  and  $h_r$  are complementary functions of (2.1) and  $p_r$  is a particular solution. We shall assume that it is possible to choose these solutions such that

$$(2.3) \quad \lim_{r \rightarrow \infty} (g_r/f_r) = \lim_{r \rightarrow \infty} (h_r/g_r) = \lim_{r \rightarrow \infty} (p_r/g_r) = 0.$$

The problem which we consider is the generation of a solution  $y_r$  which has the property that  $\lim_{r \rightarrow \infty} (y_r/f_r) = \lim_{r \rightarrow \infty} (y_r/g_r) = 0$  and which satisfies the condition  $y_1 = k_1$ . Provided that  $h_1 \neq 0$ , the solution of (2.1) with these boundary conditions is immediately seen to be

$$(2.4) \quad y_r = (k_1 - p_1)(h_r/h_1) + p_r.$$

If  $h_1 = 0$ , then the solution does not exist, unless  $k_1 = p_1$ , in which case there is an infinity of solutions. It is well known that if an attempt is made to generate the required solution of (2.1) using direct forward recurrence starting from three initial conditions of the form  $y_j = k_j$ , the resulting procedure will be completely ineffective due to an unstable propagation of rounding errors. If, however, we set  $y_N = y_{N+1} = 0$  for some large integer  $N$ , which need not be specified at present, we may rewrite (2.1) as the boundary value problem

$$(2.5) \quad \begin{aligned} b_2 y_2^{[N]} + c_2 y_3^{[N]} + d_2 y_4^{[N]} &= e_2 - a_2 k_1, \\ a_3 y_2^{[N]} + b_3 y_3^{[N]} + c_3 y_4^{[N]} + d_3 y_5^{[N]} &= e_3, \\ &\dots\dots\dots \\ a_{N-1} y_{N-2}^{[N]} + b_{N-1} y_{N-1}^{[N]} &= e_{N-1}. \end{aligned}$$

The conditions under which  $y_r^{[N]} \rightarrow y_r$  as  $N \rightarrow \infty$  are rather complicated and will not concern us here. Instead, we shall assume as a basic hypothesis that  $y_r^{[N]} \rightarrow y_r$  since this is usually the case in practice. (This general convergence problem has recently been considered by Lozier [4], and the reader is referred to his thesis for the details.) As an example of this convergence, we consider the constant coefficient equation

$$ay_{r-1} + by_r + cy_{r+1} + dy_{r+2} = 0.$$

The general solution of this recurrence relation is of the form (2.2) where

$$f_r = w_1^r, \quad g_r = w_2^r \quad \text{and} \quad h_r = w_3^r \quad (p_r = 0)$$

with the  $w_i$  being the three roots, assumed distinct in modulus, of the polynomial

$$dx^3 + cx^2 + bx + a = 0.$$

For the time being we shall assume that  $|w_1| > |w_2| > |w_3|$  so as to be consistent with our earlier notation. Now

$$y_r^{[N]} = A_N f_r + B_N g_r + C_N h_r \quad \text{and} \quad y_1^{[N]} = k_1, \quad y_N^{[N]} = 0, \quad y_{N+1}^{[N]} = 0.$$

These relations give rise to the three equations

$$\begin{aligned} k_1 &= A_N f_1 + B_N g_1 + C_N h_1, \\ 0 &= A_N f_N + B_N g_N + C_N h_N, \\ 0 &= A_N f_{N+1} + B_N g_{N+1} + C_N h_{N+1}. \end{aligned}$$

Solving these three equations by Cramer’s rule, for example, and then letting  $N \rightarrow \infty$ , we find that  $A_N \rightarrow 0$  and  $B_N \rightarrow 0$ ; but  $C_N \not\rightarrow 0$  showing that in this case  $y_r^{[N]} \rightarrow y_r$ . A slightly different argument is required in the case  $|w_1| = |w_2|$  but our final result is still the same.

The two main problems which will now concern us are how to choose the ‘optimal’ value of  $N$  to achieve a prescribed degree of precision and how to prove that the solution which we obtain is indeed the required solution. Following the approach adopted by Oliver, we may reduce (2.5) to the upper triangular form

$$\begin{aligned} (2.6) \quad & \hat{a}_2 y_2^{[N]} - \hat{b}_2 y_3^{[N]} - \hat{c}_2 y_4^{[N]} &&= \hat{e}_2, \\ & \hat{a}_3 y_3^{[N]} - \hat{b}_3 y_4^{[N]} - \hat{c}_3 y_5^{[N]} &&= \hat{e}_3, \\ & \dots\dots\dots && \\ & && \hat{a}_{N-1} y_{N-1}^{[N]} = \hat{e}_{N-1}, \end{aligned}$$

where

$$\begin{aligned} \hat{a}_2 &= b_2, \quad \hat{b}_2 = -c_2, \quad \hat{c}_2 = -d_2, \quad \hat{e}_2 = e_2 - a_2 y_1, \\ \hat{a}_r &= a_r \hat{b}_{r-1} + \hat{a}_{r-1} b_r, \quad \hat{b}_r = -a_r \hat{c}_{r-1} - \hat{a}_{r-1} c_r, \quad \hat{c}_r = -\hat{a}_{r-1} d_r, \quad r = 3(1)N - 1, \\ \hat{e}_r &= \hat{a}_{r-1} e_r - a_r \hat{e}_{r-1}. \end{aligned}$$

Up to now we have followed exactly the same approach as that adopted by Oliver. The procedure adopted by Oliver at this stage is to compute  $y_N$  for some trial value of  $N$  and then to calculate a better value of  $N$  essentially by recurring in the direction of decreasing  $r$ . We shall not follow this approach but shall instead attempt to calculate an optimal value for  $N$  before any solution values have been calculated. We first of all prove the existence of an infinite expansion for  $y_r^{[N]}$  by means of the following lemma:

LEMMA 1. *Suppose that a positive integer  $N$  is chosen so that  $y_N^{[N]} = y_{N+1}^{[N]} = 0$ . Define a sequence  $z_n^r$  as the solution of the recurrence relation  $\hat{a}_{n+1} z_{n+1}^r - \hat{b}_n z_n^r - \hat{c}_{n-1} z_{n-1}^r = 0$  with initial conditions  $z_{r-1}^r = 0, z_r^r = 1/\hat{a}_r$ . Then*

$$(2.7) \quad y_r^{[N]} = \sum_{s=r}^{N-1} \hat{e}_s z_s^r.$$

*Proof.* The relation satisfied by the  $z_n^r$  is immediately recognizable as the adjoint equation associated with the difference equation

$$\hat{a}_n y_n^{[N]} - \hat{b}_n y_{n+1}^{[N]} - \hat{c}_n y_{n+2}^{[N]} = 0,$$

which is represented by a typical row of the homogeneous form of (2.6). If we now multiply the  $(s - 1)$ th row of (2.6) by  $z_s^r$  and sum from  $s = r - 1$  to  $s = N - 1$ , we obtain

$$\sum_{s=r-1}^{N-1} (\hat{a}_s y_s^{[N]} - \hat{b}_s y_{s+1}^{[N]} - \hat{c}_s y_{s+2}^{[N]}) z_s^r = \sum_{s=r-1}^{N-1} \hat{e}_s z_s^r.$$

If we now use our given initial conditions  $z_{r-1}^r = 0$ ,  $z_r^r = 1/\hat{a}_r$  together with the definition of  $z_n^r$ , this reduces to

$$y_r^{[N]} = \sum_{s=r}^{N-1} \hat{e}_s z_s^r$$

as required, and this expression is in fact Green's formula [6, p. 48, Eq. (3.11)]. This completes the proof of the lemma.

It now follows immediately that

$$(\dagger) \quad y_r^{[N+1]} - y_r^{[N]} = \hat{e}_N z_N^r.$$

By using this expression we can estimate the effect of changing from  $N$  to  $N + 1$  before any back substitution has been carried out. We consider the following two particular cases which are common in practical applications:

(i) If we wish to compute  $y_L$  to  $D$  decimal places, we apply the recurrence relation for values of  $r$  past  $L$  until a value is found for which

$$(2.8) \quad |\hat{e}_n z_n^L| < \frac{1}{2} * 10^{-D}$$

and we then set  $N = n$ . As we shall show later, this case has its main practical application in the numerical solution of ordinary differential equations.

(ii) If we wish to compute  $y_r$  for a range of values of  $r$  between 1 and  $L$  say, we again continue using the recurrence relation past  $L$  until a value of  $n$  is found for which (2.8) holds, and we then set  $N = n$ . Putting  $r = N$  in Eq. ( $\dagger$ ) and using the relations  $y_N^{[N]} = 0$  and  $z_N^N = 1/\hat{a}_N$ , we obtain the expression  $y_N^{[N+1]} = \hat{e}_N/\hat{a}_N$ . In particular if we wish to compute all values of  $y_r$  that exceed  $\frac{1}{2} * 10^{-D}$  in absolute value (assuming, of course, that  $y_r \rightarrow 0$  as  $r \rightarrow \infty$ ) we compute values until  $|\hat{e}_n/\hat{a}_n| < \frac{1}{2} * 10^{-D}$ , and then set  $N = n$ .

For problems of type (i) this procedure is relatively satisfactory since, as we shall show in the next section, it does yield an acceptable value for  $N$ , automatically. In case (ii) the algorithm is not as good as Olver's in that it does not always guarantee that we have found the optimal value of  $N$ . The criterion which would enable us to determine the required value of  $N$  is

$$(2.9) \quad \left( \text{Max}_{1 \leq r \leq L} |z_n^r| \right) |\hat{e}_n| < \frac{1}{2} * 10^{-D};$$

but because of the excessive amount of computation involved in calculating all of the  $z_n^r$ 's, we shall use criterion (2.8). This criterion is often useful when one of the

boundary conditions is at infinity since it provides us with a practical approximation to infinity before any solution values have been calculated. Algorithms which require a value of  $N$  to be guessed as a practical approximation to infinity would seem to be of rather uncertain value in general. Problems may occur when we do not know anything about the behavior of  $z_n^r$  and so criterion (2.8) may not yield the optimal value of  $N$ . In cases such as this it is important that we should be able to test the optimality of  $N$  before too much computation has been carried out. This may easily be done by setting  $y_N^{[N]} = y_{N+1}^{[N]} = 0$  and then solving the equation

$$(*) \quad \hat{a}_n y_n = \hat{b}_n y_{n+1} + \hat{c}_n y_{n+2}, \quad y_N = 1/\hat{a}_N, \quad y_{N+1} = 0$$

in the direction of decreasing  $n$ . It may be shown that  $y_n = z_n^n$  and so at each step the value of  $|\hat{e}_n z_n^n|$  may be computed and a check made to ensure that this quantity is less than  $\frac{1}{2} * 10^{-D}$ . If this is not the case and the above quantity takes its maximum value at a point  $t$  our procedure for estimating  $N$  needs to be carried out again using the criterion

$$|\hat{e}_n z_n^t| < \frac{1}{2} * 10^{-D},$$

and then we set  $N = n$ . In extreme cases a new value of  $N$  may have to be computed several times, and exactly how competitive our algorithm is with that of Oliver in these cases is not clear. It may be that in these cases it is best to use a combination of the algorithms proposed in this paper and Oliver's algorithm with the former providing the first trial value of  $N$  for use with the latter. This combination should be particularly effective when one of the boundary conditions is at infinity and so it is difficult to guess a reasonable value for  $N$ . If it was known, for example, that  $|z_n^r|$  was an increasing function of  $r$  for fixed  $n \geq r$  then we could apply criterion (2.8) with safety since in this case

$$\text{Max}_{1 \leq r \leq L} |z_n^r \hat{e}_n| = |z_n^L \hat{e}_n|.$$

The present author has so far been unable to derive a useful sufficient criterion for this to be the case and this is an area where more results, perhaps along the lines of those developed in [11], would be valuable. Finally, in this section we mention very briefly the more general form of the linear normalizing condition  $\sum_{r=1}^{\infty} \gamma_r y_r = k$ . If we are able to solve the problem with the initial condition  $y_1 = c$  (where  $c$  is arbitrary), then we need only to compute  $\sum_{r=1}^{\infty} \gamma_r y_r$  and renormalize the solution by a factor  $k/(\sum_{r=1}^{\infty} \gamma_r y_r)$ . If this procedure fails owing to the value of  $y_1$  being zero or very small, then we could try for example  $y_2$  arbitrary and carry out a similar procedure. The point is that if we can solve the initial value problem described earlier in this section, then we can also solve the problem with the general linear normalizing condition.

**3. Examination of the Truncation Error.** In this section we examine in more detail the validity of the solution which we have obtained using the algorithms described in the previous section. In particular, we wish to show that the value of  $N$

chosen using a criterion of the form

$$|y_r^{[N+1]} - y_r^{[N]}| < \frac{1}{2} * 10^{-D}$$

guarantees that  $y_r^{[N+1]}$  agrees with the true solution  $y_r$  to  $D$  decimal places. We have already shown that

$$y_r^{[N+1]} - y_r^{[N]} = \hat{e}_N z_N^r.$$

Replacing  $N$  by  $N + 1, N + 2, \dots, N + s - 1$  in turn and adding the resulting equations, we obtain

$$y_r^{[s+N]} - y_r^{[N]} = \sum_{k=0}^{s-1} \hat{e}_{N+k} z_{N+k}^r.$$

Putting  $E_N^r = y_r - y_r^{[N]}$  and letting  $s \rightarrow \infty$  we have, by virtue of the assumption that  $\lim_{N \rightarrow \infty} y_r^{[N]} = y_r$  for all  $r$ ,

$$E_N^r = \sum_{s=N}^{\infty} \hat{e}_s z_s^r.$$

In order that  $y_r^{[N]}$  and  $y_r$  should agree to any prescribed tolerance  $\epsilon$  it is necessary that  $|E_N^r|$  should be less than  $\epsilon$  for each value of  $r$ . Once a value of  $N$  has been calculated we may compute the truncation error by continuing the recurrence past  $r = N$ . Putting  $N = r$  in our expression for the truncation error, we have

$$E_r^r \equiv y_r - y_r^{[r]} = y_r - \sum_{s=r}^{\infty} \hat{e}_s z_s^r.$$

Thus, our scheme is equivalent to approximating this convergent series for  $y_r$  by a partial sum and in this respect it has obvious similarities with Olver's algorithm. Finally, in this section we mention briefly the possible effect of rounding errors. Since we are assuming that the boundary value problem to be solved is well conditioned, any standard method for solving linear systems should not experience difficulties with unstable error propagation. Information regarding the nature of error build up when using most standard methods is available from the general theory [12]. The procedure which we have adopted in practice to reduce our original system to upper triangular form is Gaussian elimination with row changes if a pivot is nearly zero (partial pivoting) and practical experience has shown this procedure to be satisfactory in general.

**4. Numerical Results.** In this section we present some numerical results to illustrate some of the algorithms proposed in the previous section for third-order recurrence relations. The first problem which we consider is:

*Problem 1.* Integrate the scalar differential equation  $y' = -y, y(0) = 1$ , in the range  $0 \leq x \leq 0.2$  using the discretization algorithm

$$(4.1) \quad y_{n+2} - 9y_n + 8y_{n-1} = h(3y'_{n+1} - 6y'_n - 3y'_{n-1}).$$

Scheme (4.1), which is of the type considered for nonstiff equations in [1], is not even zero-stable, owing to the fact that there are two roots of the polynomial equation  $x^3 - 9x + 8 = 0$  which lie outside the unit circle. In view of this it is clear that if we were to generate a solution of (4.1) using direct forward recurrence, assuming that we can find the two extra initial conditions required, the solution obtained would soon bear no relation to the required solution. If instead we use the algorithm derived in Section 2 to generate the required solution, it can be shown that the resulting procedure is at least  $A(0)$ -stable in the sense of Widlund (see [3, p. 233] for example) and has order 4. Procedures for finding stability regions of schemes similar to (4.1) have been discussed in more detail elsewhere [1] and so will not be considered here. In general, the procedure that we shall use to solve a system of first-order O.D.E.'s, especially if it is stiff, is to advance one integration step at a time and, on the basis of an estimate obtained for the local truncation error, decide whether or not the stepsize of integration should be changed. Thus, in general it is case (i) discussed in Section 2 which is important in the solution of O.D.E.'s. We note that changes in the steplength are relatively easy to perform since scheme (4.1) coupled with the algorithm described in Section 2 is essentially a one step method in that it requires only one initial condition. In order to simplify our numerical example we shall not be concerned with procedures for changing the stepsize but shall assume instead that a constant stepsize of  $h = 0.02$  is used so that the problem is to find the first ten values of  $y_j$ . The results obtained for this problem are given in Table 1 and, as can be seen, satisfactory agreement is obtained.

*Problem 2.* The second problem which we consider is the solution of the difference equation

$$(4.2) \quad y_{r+2} - \frac{(r^3 + 3r^2 + r/4 + 3/4)}{r^2 - r/2 + 1/2} y_{r+1} + \frac{(3r^3/2 + 11r^2/4 + 5r/4 + 1/4)}{r^2 - r/2 + 1/2} y_r - \frac{(r^3/2 + 3r^2/4 + r/2)}{r^2 - r/2 + 1/2} y_{r-1} = 0$$

that is recessive as  $r \rightarrow \infty$  and has the initial value  $y_0 = 1$ . It can be shown that the complementary functions of this problem can be chosen so as to satisfy the conditions imposed in Section 2 and, in particular, the required solution is  $y_r = (1/2)^r$ . All values of the solution which have magnitude greater than  $10^{-6}$  are sought and so this is an example of a problem where one boundary condition may be thought of as being at infinity and one of the problems is to obtain a practical approximation to infinity. As can be seen from Table 2, this approximation is obtained automatically and satisfactory agreement with the true solution is obtained. It was found that for this problem, by actually computing the quantities  $z_n^r$  for increasing values of  $r$ , that  $z_n^r$  is a nondecreasing function of  $r$  for fixed  $n$ . It now follows immediately that the value of  $N$  obtained is optimal (see Section 2) and numerical experiments confirm that this is indeed the case.



TABLE 1

*Absolute accuracy required = 4 decimal places*

*Estimated value of N = 20*

$r$	$y_r^{[20]}$	$\hat{e}_r z_r^{10}$	True solution
1	0.98019866	—	0.98019867
2	0.96078941	—	0.96078944
3	0.94176445	—	0.94176453
4	0.92311614	—	0.92311635
5	0.90483693	—	0.90483742
6	0.88691927	—	0.88692044
7	0.86935546	—	0.86935823
8	0.85213722	—	0.85214379
9	0.83525451	—	0.83527021
10	0.81869387	0.11177507	0.81873075
-----			
11	0.80243001	0.12182301	0.80251880
12	0.78642142	$0.29441481 \cdot 10^{-1}$	0.78662786
13	0.77054711	$0.18819592 \cdot 10^{-1}$	0.77105156
14	0.75463621	$0.60512259 \cdot 10^{-2}$	0.75578374
15	0.73792540	$0.31003434 \cdot 10^{-2}$	0.74081822
16	0.71986262	$0.11451524 \cdot 10^{-2}$	0.72614904
17	0.69487642	$0.52979218 \cdot 10^{-3}$	0.71177032
18	0.66432684	$0.20940220 \cdot 10^{-3}$	0.69767632
19	0.58170411	$0.92293239 \cdot 10^{-4}$	0.68386141
20	0.00000000	$0.37702526 \cdot 10^{-4}$	0.67032005

TABLE 2

*Estimated value of N = 21*

$r$	$y_r^{[21]}$	error	$r$	$y_r^{[21]}$	error
1	0.49999976	$0.244 \cdot 10^{-6}$	11	$0.48779381 \cdot 10^{-3}$	$0.487 \cdot 10^{-6}$
2	0.24999963	$0.366 \cdot 10^{-6}$	12	$0.24365307 \cdot 10^{-3}$	$0.488 \cdot 10^{-6}$
3	0.12499957	$0.427 \cdot 10^{-6}$	13	$0.12158270 \cdot 10^{-3}$	$0.488 \cdot 10^{-6}$
4	$0.62499957 \cdot 10^{-1}$	$0.457 \cdot 10^{-6}$	14	$0.60547511 \cdot 10^{-4}$	$0.488 \cdot 10^{-6}$
5	$0.31249528 \cdot 10^{-1}$	$0.472 \cdot 10^{-6}$	15	$0.30029918 \cdot 10^{-4}$	$0.488 \cdot 10^{-6}$
6	$0.15624520 \cdot 10^{-1}$	$0.480 \cdot 10^{-6}$	16	$0.14771122 \cdot 10^{-4}$	$0.488 \cdot 10^{-6}$
7	$0.78120161 \cdot 10^{-2}$	$0.484 \cdot 10^{-6}$	17	$0.71417237 \cdot 10^{-5}$	$0.488 \cdot 10^{-6}$
8	$0.39057642 \cdot 10^{-2}$	$0.486 \cdot 10^{-6}$	18	$0.33270257 \cdot 10^{-5}$	$0.488 \cdot 10^{-6}$
9	$0.19526383 \cdot 10^{-2}$	$0.487 \cdot 10^{-6}$	19	$0.14196984 \cdot 10^{-5}$	$0.488 \cdot 10^{-6}$
10	$0.97607530 \cdot 10^{-3}$	$0.487 \cdot 10^{-6}$	20	$0.46649278 \cdot 10^{-6}$	$0.487 \cdot 10^{-6}$



Then if we multiply the  $j$ th equation of the system (5.1) by  $z_{q+j}^r$  and sum from  $j = r - q$  to  $j = N - 1 - q$  (this is the generalization of the procedure used to prove Lemma 1), we obtain

$$(5.3) \quad \hat{a}_{r,r}y_r^{[N]} = \hat{e}_rz_r^r + \hat{e}_{r+1}z_{r+1}^r + \cdots + \hat{e}_{N-1}z_{N-1}^r, \quad q + 1 \leq r \leq N - 1.$$

It now follows that

$$(5.4) \quad y_r^{[N+1]} - y_r^{[N]} = \hat{e}_N z_N^r / \hat{a}_{r,r}.$$

This is very similar to the expression derived in Section 2 for the third-order case, and a very similar procedure to that described in Section 2 may be used to estimate the optimal value of  $N$  in advance. As before, we shall assume as a basic hypothesis that  $y_r^{[N]} \rightarrow y_r$  as  $N \rightarrow \infty$ . On this assumption it is clear that

$$(5.5) \quad \hat{a}_{r,r}y_r = \sum_{j=r}^{\infty} \hat{e}_j z_j^r,$$

which is the generalization of the Green's formula derived in Section 2. Subtracting (5.3) from (5.5), we have

$$(5.6) \quad \hat{a}_{r,r}(y_r - y_r^{[N]}) = \sum_{j=N}^{\infty} \hat{e}_j z_j^r.$$

It can easily be seen that in the case  $m = 2, q = 1$  the Eqs. (5.3), (5.5) and (5.6) are effectively the same as Eqs. (5.04), (5.03) and (5.01), respectively, of Olver's original paper [10]. In this sense our results may be regarded as generalizations of those of Olver. The basic reason why Olver's algorithm is so effective for the second-order case is that in his case the adjoint equation is of order 1 and so all solutions may be expressed as a multiple of the single solution of the adjoint. This property will also hold for the higher-order case when  $q = m - 1$ . In general, however, it will be necessary to construct the sequence  $z_n^r$  for the largest value of  $r$  required and test convergence in the manner described in Section 2. Since the extension of our algorithm is now immediate we shall not consider it any further but instead go on to give some numerical results for the case  $m = 4$  where the required solution is an intermediate one, i.e. neither the most dominant nor the most recessive.

**6. Numerical Results.** In this section we present some additional numerical results which serve to illustrate the extension of our algorithms to recurrence relations of order greater than three. The most important application of our algorithm in this case is in the generation of intermediate solutions since the most dominant (recessive) solution (assuming of course that such solutions exist) can usually be found by forward (backward) recursion. As a first example we consider:

*Problem 3.*

$$(6.1) \quad y_{r+3} - 111.1y_{r+2} + 1121.1y_{r+1} - 1111y_r + 100y_{r-1} = 0$$

with the conditions  $y_1 = 1, y_2 = 1$ , and  $y_r$  bounded as  $r \rightarrow \infty$ . The general solution of this recurrence relation, which corresponds to a problem with  $m = 4, q = 2$  in our

TABLE 3

error tolerance = $(1/2) * 10^{-4}$			error tolerance = $(1/2) * 10^{-6}$	
$N = 15$			$N = 17$	
$r$	Solution obtained	$\hat{e}_r z_r^{10} / \hat{a}_{r,r}$	Solution obtained	
3	1.00000000	—	1.00000000	
4	1.00000000	—	1.00000000	
5	1.00000000	—	1.00000000	
6	0.99999999	—	1.00000000	
7	0.99999998	—	1.00000000	
8	0.99999989	—	1.00000000	
9	0.99999890	—	0.99999999	
10	0.99998900	—	0.99999989	
-----				
11	0.99989000	$0.980100 * 10^{-1}$	0.99999890	
12	0.99890010	$0.989010 * 10^{-2}$	0.99998900	
13	0.98901000	$0.989901 * 10^{-3}$	0.99989000	
14	0.89100000	$0.989990 * 10^{-4}$	0.99890010	
15		$0.989999 * 10^{-5}$	0.98901000	
16		$0.990000 * 10^{-6}$	0.89100000	
17		$0.990000 * 10^{-7}$		

notation, is

$$y_r = A(1/10)^r + B + C(10)^r + D(100)^r,$$

and the given initial conditions are clearly such that  $A = C = D = 0$ ,  $B = 1$ . For our particular example we shall consider the generation of the required solution of this problem in the range  $3 \leq r \leq 10$  for the two different absolute error tolerances  $(1/2) * 10^{-4}$  and  $(1/2) * 10^{-6}$ . As can be seen from the results presented in Table 3, the optimal value of  $N$  is obtained automatically in both cases and agreement with the exact solution is satisfactory throughout the given range. As our second problem in this section, we consider one arising from the solution of an ordinary differential equation:

*Problem 4.* Integrate the scalar differential equation  $y' = -xy$ ,  $y(0) = 1$  using the discretization algorithm

$$(6.2) \quad y_{r+3} - y_{r+2} = (h/720) \{-19y'_{r+4} + 346y'_{r+3} + 456y'_{r+2} - 74y'_{r+1} + 11y'_r\}.$$

Some remarks regarding the source of (6.2) would seem to be relevant at this stage. The first point that we note is that (6.2) is rather like an implicit Adams method (see [3, p. 41] for example) except that the value  $f_{r+4}$  as well as the usual  $f_r, \dots, f_{r+3}$  is used in the expression for  $y_{r+3}$ . By using this additional forward value it is possible to derive integration procedures which have higher orders of accuracy than the more conventional Adams methods, and it may be shown that scheme (6.2) has a local truncation error of order  $O(h^6)$ . In order to solve for  $y_{r+3}$ , however, we need to have an

TABLE 4

error tolerance = $(1/2) * 10^{-4}$			error tolerance = $(1/2) * 10^{-6}$	
	$N = 11$		$N = 12$	
$r$	Solution obtained	$\hat{e}_r z_r^{10} / \hat{a}_{r,r}$	True Solution	Solution obtained
3	0.99955010	—	0.99955010	0.99955010
4	0.99920032	—	0.99920032	0.99920032
5	0.99875078	—	0.99875078	0.99875078
6	0.99820162	—	0.99820162	0.99820162
7	0.99755299	—	0.99755299	0.99755299
8	0.99680511	—	0.99680511	0.99680511
9	0.99595819	—	0.99595819	0.99595819
10	0.99498364	—	0.99501248	0.99501248
-----				
11	—	$0.2886 * 10^{-4}$	0.99396826	0.99393684
12	—	$0.9127 * 10^{-9}$	—	—
13	—	$0.3126 * 10^{-13}$	—	—

estimate for  $f_{r+4}$  and hence, in our case,  $y_{r+4}$ . Thus, our problem is of an ideal form for solution using the algorithm developed in the previous section with  $m = 4, q = 3$ . In our numerical experiment scheme (6.2) was used with a constant step of  $h = 0.01$  and the required solution was sought in the range  $3 \leq r \leq 10$  for the two different tolerances  $(1/2) * 10^{-4}$  and  $(1/2) * 10^{-6}$ . In order to use scheme (6.2) we need to know the values of  $y_1$  and  $y_2$  and for our purposes we set  $y_1 = \exp(-\frac{1}{2}h^2), y_2 = \exp(-2h^2)$  which are in fact the exact solutions. In a practical example we would of course need to generate these extra initial conditions using a one step method, such as a fourth-order Runge-Kutta method for example, but since this does not present us with any additional problems there is no loss in generality in using the exact values. As can be seen from the results presented in Table 4, the optimal value of  $N$  is again found automatically in both cases and agreement with the exact solution is again satisfactory throughout. We note that the value of  $\hat{e}_r z_r^{10} / \hat{a}_{r,r}$  decreases very rapidly resulting in our algorithm converging very quickly for any prescribed error tolerance.

**7. Conclusion.** The purpose of the present paper has been to develop an algorithm for the solution of high-order linear recurrence relations in cases where the required solution is not stably generated by direct forward recurrence due to unstable propagation of rounding errors. Our main aim was to extend Olver's algorithm so that, if the original problem is replaced by a well-posed boundary value problem, the size,  $N$ , of the boundary value problem to be solved to give the required degree of precision is estimated automatically in all cases. We were particularly interested in the case where one of the boundary conditions was at infinity so that Oliver's algorithm, which requires a practical approximation to infinity to be guessed, is not so easy to use. We showed in Section 2 that in some cases we are able to estimate the value of  $N$  automatically while in other cases we had to perform a check to test that

the estimated value of  $N$  was acceptable. In cases where one of the boundary conditions was at infinity the algorithm developed in Section 2 provided a practical approximation to infinity before any back substitutions were performed. In Section 3 we were able to derive an exact expression for the required solution and this allowed us to positively identify the solution obtained as the required solution of our original problem. Finally in Section 5 the extension of our analysis to higher-order equations was considered and we were able to show that all of the important results developed in Sections 2 and 3 for third-order recurrences could be immediately extended to equations of arbitrary order. In conclusion, we may say that in some cases, where the optimal value of  $N$  is obtained automatically, the proposed algorithm is at a reasonably satisfactory level. In other cases, where we are not sure that the estimated value of  $N$  is optimal, it may be that a combination of the algorithm proposed in this paper and that proposed by Oliver would be better than either one by itself, especially if one of the boundary conditions is at infinity.

**Acknowledgement.** The author is very grateful to F. W. J. Olver, whose many suggestions and criticisms greatly improved the presentation of this paper.

Department of Mathematics  
Imperial College  
South Kensington  
London S. W. 7, England

1. J. R. CASH, "High order methods for the numerical integration of ordinary differential equations," *Numer. Math.* (To appear.)
2. W. GAUTSCHI, "Computational aspects of three-term recurrence relations," *SIAM Rev.*, v. 9, 1967, pp. 24–82.
3. J. D. LAMBERT, *Computational Methods in Ordinary Differential Equations*, Wiley, London and New York, 1973.
4. D. W. LOZIER, *A Stable Algorithm for Computing any Solution of an Arbitrary Linear Difference Equation*, Ph.D. Thesis, University of Maryland.
5. J. C. P. MILLER, *British Association for the Advancement of Science: Bessel Functions, Part II. Mathematical Tables*, Vol. 10, Cambridge Univ. Press, Cambridge, 1952.
6. K. S. MILLER, *Linear Difference Equations*, Benjamin, New York, 1968.
7. J. OLIVER, "Relative error propagation in the recursive solution of linear recurrence relations," *Numer. Math.*, v. 9, 1967, pp. 323–340.
8. J. OLIVER, "The numerical solution of linear recurrence relations," *Numer. Math.*, v. 11, 1968, pp. 349–360.
9. J. OLIVER, "An extension of Olver's error estimation technique for linear recurrence relations," *Numer. Math.*, v. 12, 1968, pp. 459–467.
10. F. W. J. OLVER, "Numerical solution of second order linear difference equations," *J. Res. Nat. Bur. Standards Sect. B*, v. 71, 1967, pp. 111–129.
11. F. W. J. OLVER, "Bounds for the solutions of second-order linear difference equations," *J. Res. Nat. Bur. Standards Sect. B*, v. 71, 1967, pp. 161–166.
12. J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford Univ. Press, Oxford, 1965.
13. R. V. M. ZAHAR, "A mathematical analysis of Miller's algorithm," *Numer. Math.*, v. 27, 1977, pp. 427–447.