A Note on Olver's Algorithm for the Solution of Second-Order Linear Difference Equations

By J. R. Cash

Abstract. An extension of Olver's algorithm is described. This allows the solution of certain linear difference equations which are in a sense ill-conditioned for solution by Olver's method. The new algorithm is illustrated by considering a particular problem.

1. Introduction. In a classic paper [2], F. W. J. Olver introduced a technique for the stable computation of recessive solutions of the linear, inhomogeneous, second-order difference equation

$$(1.1) \qquad a_r y_{r-1} - b_r y_r + c_r y_{r+1} = d_r, \quad y_0 = k, \quad y_1 = m.$$

Basically the approach proposed by Olver is to replace the original initial value problem (1.1) by an equivalent boundary value problem. This is done by abandoning the initial condition $y_1 = m$ and replacing it by a condition of the form $y_{N+1} = p$ for some sufficiently large value of N. Normally we set $p = 0$ but in cases where it is possible to obtain a better value for y_{N+1} (for example, via an asymptotic expansion) it may be better to use this more accurate value in setting up the boundary conditions. A similar idea had been discussed earlier by Gautschi [1] following a suggestion by M. E. Rose. However, Gautschi only suggests taking y_{N+1} to be the true value rather than some arbitrary value such as zero. The important aspect of Olver's algorithm is that it allows the optimal value of N to be estimated automatically before any solution values have been computed. Having set up this linear boundary value problem, the required solution can usually be found efficiently using either Gaussian elimination *without pivoting* or LU decomposition. However, for some problems it has been found to be advisable to use partial pivoting and, furthermore, the general theory for transmission of errors for Gaussian elimination [3, Chapter 4] usually assumes that the absolute values of the multipliers are bounded by unity. It is well known, however, that if partial pivoting is used with Olver's algorithm, the facility for estimating automatically the optimal value of N is lost and, as a result, the effectiveness of the algorithm is reduced. The purpose of the present paper is to introduce a modified algorithm which is still effective for cases where partial pivoting needs to be used, but which reduces identically to Olver's algorithm when partial pivoting is not required.

2. The Basic Algorithm. The first step of Olver's algorithm is to replace the initial condition $y_1 = m$ in (1.1) by the boundary condition $y_{N+1} = 0$ and to rewrite

the resulting system of equations in the matrix form

$$
(2.1) \quad
\begin{bmatrix}
-b_1 & c_1 & & & & \\
a_2 & -b_2 & c_2 & & & \\
 & a_3 & -b_3 & c_3 & & \\
 & & & & & \\
 & & & & a_{N-1} & -b_{N-1} & c_{N-1} \\
 & & & & & a_N & -b_N
\end{bmatrix}
\begin{bmatrix}
y_1 \\ y_2 \\ y_3 \\ \\ \\ y_N
\end{bmatrix}
=
\begin{bmatrix}
d_1 - a_1 k \\ d_2 \\ d_3 \\ \\ \\ d_N
\end{bmatrix}.
$$

Denoting the solution of this system by $y^{[N]}$ we may rewrite (2.1) in the symbolic form $A_N y^{[N]} = d^N$. This system is then transformed into one of the form $\bar{A}_N y^{[N]} = \bar{d}^N$, where $\bar{A}_N$ is upper triangular, using Gaussian elimination with partial pivoting. The coefficient matrix $\bar{A}_N$ may be written in the form

$$
(2.1a) \quad \bar{A}_N =
\begin{bmatrix}
-\bar{b}_1 & \bar{c}_1 & \bar{s}_1 & & & \\
 & -\bar{b}_2 & \bar{c}_2 & \bar{s}_2 & & \\
 & & & & & \\
 & & -\bar{b}_{N-2} & \bar{c}_{N-2} & \bar{s}_{N-2} & \\
 & & & -\bar{b}_{N-1} & \bar{c}_{N-1} & \\
 & & & & -\bar{b}_N &
\end{bmatrix},
$$

where

$$
\bar{s}_r =
\begin{cases}
0 & \text{if partial pivoting was not required for the } r\text{th row,} \\
c_{r+1} & \text{otherwise.}
\end{cases}
$$

The quantities $\bar{b}_r$, $\bar{c}_r$ and $\bar{d}_r$ are the result of applying Gaussian elimination with partial pivoting to the finite tridiagonal system (2.1) and are thus defined for $1 \leqslant r \leqslant N$. Their explicit forms would seem to be too complicated to write down, but it is clear that they are defined uniquely in terms of the original coefficients a_r, b_r, c_r and d_r.

If we now consider the solution of (1.1) with the initial condition $y_1 = m$ replaced by the boundary condition $y_{N+2} = 0$, the resulting system of equations defines a solution $y^{[N+1]}$ as $A_{N+1} y^{[N+1]} = d^{N+1}$. Using Gaussian elimination with partial pivoting this can be transformed into the system $\hat{A}_{N+1} y^{[N+1]} = \hat{d}^{N+1}$, where the matrix $\hat{A}_{N+1}$ takes the form

$$(2.2) \quad \hat{A}_{N+1} = \begin{bmatrix} -\overline{b}_1 & \overline{c}_1 & \overline{s}_1 & & & & & \\ & -\overline{b}_2 & \overline{c}_2 & \overline{s}_2 & & & & \\ & & & & & & & \\ & & & & -\overline{b}_{N-2} & \overline{c}_{N-2} & \overline{s}_{N-2} & \\ & & & & & -\overline{b}_{N-1} & \overline{c}_{N-1} & \overline{s}_{N-1} \\ & & & & & & -\hat{b}_N & \hat{c}_N \\ & & & & & & & -\hat{b}_{N+1} \end{bmatrix}.$$

Note that the Nth row of $\hat{A}_{N+1}$ may not be the same as the Nth row of $\overline{A}_N$, since in forming $\hat{A}_{N+1}$ we may have interchanged the Nth and $(N+1)$th rows of A_{N+1} (i.e. performed partial pivoting on the Nth row of A_{N+1}). In this section we shall describe two algorithms, one being applicable when it is necessary to partially pivot on the Nth row of A_{N+1}, and the other being applicable when this is not necessary.

We start with the case where it is not necessary to partially pivot on the Nth row of A_{N+1}. From (2.1a) it follows that

$$(2.3a) \qquad \overline{b}_j y_j^{[N]} = \overline{c}_j y_{j+1}^{[N]} + \overline{s}_j y_{j+2}^{[N]} - \overline{d}_j, \quad j \in (1, N).$$

Similarly from (2.2) we have

$$(2.3b) \qquad \overline{b}_j y_j^{[N+1]} = \overline{c}_j y_{j+1}^{[N+1]} + \overline{s}_j y_{j+2}^{[N+1]} - \overline{d}_j, \quad j \in (1, N+1).$$

Suppose now, that our problem is to compute y_L to D decimal places for given values of the integers L and D. Since we have not allowed the possibility of pivoting on the Nth row of A_{N+1}, it follows that the $\overline{b}_j, \overline{c}_j, \overline{s}_j$ and $\overline{d}_j$ appearing in (2.3a) are identical with those in (2.3b) for all $j = 1, 2, \ldots, N$, and this allows us to develop a satisfactory procedure for estimating the optimal value of N for the solution of our problem in advance.

Defining $z_L^{[N]} = y_L^{[N+1]} - y_L^{[N]}$, and subtracting (2.3a) from (2.3b), we have for $j = L$,

$$(2.4) \qquad \overline{b}_L z_L^{[N]} = \overline{c}_L z_{L+1}^{[N]} + \overline{s}_L z_{L+2}^{[N]}.$$

The boundary conditions associated with (2.4) are

$$(2.5) \qquad \begin{cases} z_{N+1}^{[N]} \equiv y_{N+1}^{[N+1]} - y_{N+1}^{[N]} = -\overline{d}_{N+1}/\overline{b}_{N+1}, \\ z_{N+2}^{[N]} \equiv y_{N+2}^{[N+1]} - y_{N+2}^{[N]} = 0. \end{cases}$$

The problem now remains to find a compact expression for $z_L^{[N]}$ which can be used to estimate the optimal value of N. The required form is contained in the following lemma.

LEMMA 1. *The quantity $z_L^{[N]}$ satisfies the relation*

$$(2.6) \qquad z_L^{[N]} = u_{L+i} z_{L+i+1}^{[N]} + v_{L+i} z_{L+i+2}^{[N]}, \quad i = 0, 1, 2, \ldots, N-L,$$

where

$$(2.7) \quad \left.\begin{array}{ll} u_L = \overline{c}_L/\overline{b}_L, & u_{L+i+1} = \overline{c}_{L+i+1}u_{L+i}/\overline{b}_{L+i+1} + v_{L+i} \\ v_L = \overline{s}_L/\overline{b}_L, & v_{L+i+1} = \overline{s}_{L+i+1}u_{L+i}/\overline{b}_{L+i+1} \end{array}\right\},$$

$$i = 0, 1, 2, \ldots, N - L.$$

The proof of this lemma follows in a straightforward fashion by induction on i, so it will not be given here. If we now consider relation (2.6) for $i = N - L$, we have

$$z_L^{[N]} = u_N z_{N+1}^{[N]} + v_N z_{N+2}^{[N]}$$
$$= -u_N \overline{d}_{N+1}/\overline{b}_{N+1} \quad \text{from the boundary conditions (2.5).}$$

Thus we need only continue our computation of (2.7) past $r = L$ until a value of r is found satisfying

$$(2.8) \quad |u_r \overline{d}_{r+1}/\overline{b}_{r+1}| < (1/2) * 10^{-D},$$

and we then set $N = r$. If we now consider the case where partial pivoting is not used at all, we have $\overline{s}_r \equiv 0$. It now follows immediately from (2.7) that $v_r \equiv 0$ and so we have $u_{L+i+1} = \overline{c}_{L+i+1}u_{L+i}/\overline{b}_{L+i+1}$. It is now straightforward to show that in this case the resulting algorithm is exactly Olver's algorithm, which justifies our earlier claims.

We now mention very briefly the case where we have pivoted on the Nth row during the computation of $\hat{A}_{N+1}$ due to the multiplier being greater than one in modulus. The algorithm just described will no longer be applicable, since in this case $\overline{b}_N \neq \hat{b}_N$. The easiest way of getting around this problem is simply not to allow pivoting on the Nth row of $\hat{A}_{N+1}$! Since this row is the only one for which the multiplier may have modulus greater than 1, we would not normally expect any unmanageable build up of rounding errors and the algorithm just described should be satisfactory. However, if we find that the multiplier is extremely large at this stage (a pathological case) and we really do have to pivot, we can modify our algorithm in the following way. Equation (2.4) will now only be valid for $L \leqslant N - 1$. The boundary conditions associated with (2.4) can be found from (2.1a), (2.2) and are given by

$$(2.9) \quad z_{N+1}^{[N]} = -\hat{d}_{N+1}/\hat{b}_{N+1}, \quad z_N^{[N]} = -\hat{c}_N \hat{d}_{N+1}/(\hat{b}_N \hat{b}_{N+1}) - \hat{d}_N/\hat{b}_N + \overline{d}_N/\overline{b}_N.$$

Thus substituting into (2.6) for $i = N - L - 1$, we have

$$(2.10) \quad z_L^{[N]} = u_{N-1} z_N^{[N]} + v_{N-1} z_{N+1}^{[N]}.$$

Thus we continue our computation of (2.7) past $r = L$ and use relations (2.9) until a value of r is found satisfying

$$|z_L^{[r]}| < (1/2) * 10^{-D},$$

and then we set $N = r$. Note that this algorithm is more expensive than the first one, but for the problems where it has to be used, Olver's algorithm will be unsatisfactory.

3. Numerical Results. The desirability of having an algorithm which allows partial pivoting to be used when it is called for is self-evident. There is always the possibility that, if partial pivoting is not used, one or more of the multipliers in the Gaussian elimination algorithm will become extremely large in modulus and a significant loss in precision will occur. Nevertheless, we will in this section consider a numerical example to illustrate our approach. Two important points which we wish to emphasize are the following. Firstly, for the majority of linear second-order recurrence relations of practical interest, no partial pivoting is required at all. Furthermore, when partial pivoting is required, the range of r for which it is needed is small. However, since a breakdown of the pivoting scheme is one of the few weak points of Olver's basic algorithm, it does seem desirable to make Olver's algorithm more robust by incorporating partial pivoting, even though it is not often absolutely necessary, since the extension requires so little extra computation. A second point which we wish to make is that our approach may be extended in a straightforward fashion to higher-order linear recurrence relations.

TABLE 1

r	Olver's algorithm	Multipliers	Modified algorithm	Multipliers	True Solution
2	5.999988	-2799981866.	6.000000	$.357*10^{-9}$	6.000000
3	12.000000	$.357*10^{-8}$	12.000000	.100000	12.000000
4	20.000000	-1.314285	20.000000	.076087	20.000000
5	30.000000	-1.554394	30.000000	.048949	30.000000
6	42.000000	-1.542322	42.000000	.031737	42.000000
7	56.000000	-1.477117	56.000000	.021486	56.000000
8	72.000000	-1.414682	72.000000	.015188	72.000000
9	90.000000	-1.364408	90.000000	.011131	90.000000
10	109.999999	-1.324657	109.999999	.008403	110.000000
11	131.999998	-1.292715	131.999998	.006501	132.000000

The particular example which we consider is the linear, homogeneous recurrence relation

(3.1) $$a_r y_{r-1} - b_r y_r + y_{r+1} = 0, \quad y_1 = 2,$$

where the coefficients a_r and b_r are chosen so that the basis solutions of (3.1) are u_r and v_r given by

$$u_r = r^2 + r, \quad r = 1, 2, \ldots,$$

$$v_1 = 1 + \epsilon, \quad v_r = (r + \epsilon)v_{r-1}, \quad r = 2, 3, \ldots.$$

Here ϵ is a small parameter. Clearly $\lim_{r \to \infty}(u_r/v_r) = 0$ and the problem which we consider is the computation of $y_r = u_r$ correct to 5 decimal places in the range $2 \leqslant r \leqslant 11$. Obviously this example is a contrived one but it is chosen because it brings out clearly the points which we wish to make. The particular value $\epsilon = 10^{-8}$ was chosen and the results obtained are given in Table 1. The estimated value of N was 18. As can be seen from Table 1, the results for the range $3 \leqslant r \leqslant 11$, where the multipliers for Olver's algorithm are all bounded by 2 in absolute value, and so partial pivoting is not really required, are identical. However, the value $r = 2$, which is the place where partial pivoting should have been used, is unsatisfactory for Olver's algorithm but remains satisfactory for the modified algorithm. Finally, we mention that for the more realistic case where $\epsilon = 0$, Olver's algorithm breaks down completely, whereas the modified algorithm is still satisfactory. Admittedly, Olver's algorithm could be changed to deal with particular problems of this form as they arise, but the algorithm proposed in this paper does this automatically without need for human intervention.

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

1. W. GAUTSCHI, "Computational aspects of three-term recurrence relations," *SIAM Rev.*, v. 9, 1967, pp. 24–82.
2. 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.
3. J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford Univ. Press, Oxford, 1965.