

Error Analysis of Miller's Recurrence Algorithm

By F. W. J. Olver

Abstract. Miller's algorithm is a device for computing the most rapidly decreasing solution of a second-order linear difference equation. In this paper strict upper bounds are given for the errors in the values yielded by the algorithm, and general conclusions are drawn concerning the accuracy of the process.

1. Introduction. The recurrence algorithm we are concerned with here was originally devised by J. C. P. Miller for the computation of tables of the modified Bessel function $I_r(x)$ ([1], page xvii). The process was as follows. "Dummy" or "trial" values 1 and 0 are assigned to $I_n(x)$ and $I_{n+1}(x)$, respectively, where n is an arbitrarily chosen large positive integer. The recurrence relation

$$(1.01) \quad I_{r-1}(x) = (2r/x)I_r(x) + I_{r+1}(x)$$

is then applied for $r = n, n - 1, \dots, 1$ in turn to generate a corresponding sequence of trial values of $I_r(x)$. In consequence of the initial conditions the r th trial value is equal to

$$(1.02) \quad \lambda \left\{ I_r(x) + (-)^{n-r} \frac{I_{n+1}(x)}{K_{n+1}(x)} K_r(x) \right\}$$

where λ is independent of r (in fact $\lambda = xK_{n+1}(x)$, but this expression is not used computationally). Now as $n \rightarrow \infty$, $I_{n+1}(x)$ decreases rapidly and $K_{n+1}(x)$ increases rapidly. Hence by taking n to be sufficiently large we can ensure that for any given value of r , indeed for any given bounded range of values of r , the second term in the braces in (1.02) is negligible, so that the trial values effectively reduce to $\lambda I_r(x)$. The "normalizing factor" λ can be determined, for example, by dividing the trial value of $I_0(x)$ by the actual value of this function obtained from tables. Alternatively, the normalization can be effected by application of either of the relations

$$(1.03) \quad I_0(x) - 2I_2(x) + 2I_4(x) - \dots = 1,$$

$$(1.04) \quad I_0(x) + 2I_1(x) + 2I_2(x) + \dots = e^x.$$

The use of tables of Bessel functions is then avoided altogether, a striking feature which is of especial importance in automatic computation.

Following Miller's original use, similar processes have been used or described for other computations of Bessel functions ([2]–[9]), and also for the computation of repeated integrals of the error function ([10], [11]) and associated Legendre functions [12]. In some of these references, beginning with [2], the process is used in a form involving the ratios of the successive wanted functions; for this reason it is sometimes called the Ratio Method [8].

Recently, a method for solving ordinary differential equations in series of Chebyshev polynomials has been developed by C. W. Clenshaw [13] which has

features in common with the Miller algorithm. Indeed, for the differential system

$$(1.05) \quad y''(x) + k^2 y(x) = 0, \quad y(0) = \frac{1}{2}, \quad y'(0) = 0,$$

Clenshaw's method of solution applied to the range $-1 \leq x \leq 1$ becomes exactly the algorithm for the calculation of the Bessel functions $J_r(k)$, with a normalizing condition of the form (1.03).

For the general homogeneous linear difference equation of the second order

$$(1.06) \quad f_{r-1} + p_r f_r + q_r f_{r+1} = 0,$$

together with a normalizing relation of the form

$$(1.07) \quad m_0 f_0 + m_1 f_1 + m_2 f_2 + \cdots = 1,$$

in which the m_r are given numbers, the Miller algorithm may be formalized as follows. We construct a sequence of trial values a_r according to the equations

$$(1.08) \quad a_{n+1} = 0, \quad a_n = 1, \quad a_{r-1} = -p_r a_r - q_r a_{r+1} \quad (r = n, n-1, \cdots, 1),$$

and thence a further sequence

$$(1.09) \quad \varphi_r = a_r / \lambda,$$

in which λ is determined by

$$(1.10) \quad \lambda = m_0 a_0 + m_1 a_1 + \cdots + m_n a_n.$$

In the case $m_1 = m_2 = \cdots = 0$, that is, when f_0 is prescribed, W. Gautschi [11] has shown, by an extension of the argument given in [1], that if (1.06) possesses a second solution g_r which is non-vanishing and such that

$$(1.11) \quad f_r / g_r \rightarrow 0 \quad \text{as } r \rightarrow \infty,$$

then for any fixed r

$$(1.12) \quad \varphi_r \rightarrow f_r \quad \text{as } n \rightarrow \infty.$$

Gautschi has also shown, in unpublished work, that with suitable conditions the proof can be further extended to include the general normalizing condition (1.07).

These results of Miller and Gautschi provide a satisfactory account of the *convergence* of the algorithm. They show that with appropriate conditions each f_r can be computed to any desired accuracy by taking n , the point at which the backwards recurrence begins, to be large enough, $n \geq n_r$, say. These results do not, however, supply a value for n_r in any given example. Sometimes a suitable value for n_r can be determined by use of known analytical properties of the solutions of the difference equation, as in [4], [8] and [11]. An alternative procedure is to compare the results of applying the algorithm for different values of n , described, for example, in [5] and [9]; this is generally an adequate check on the convergence in practice, though not entirely satisfactory from a mathematical standpoint.

In the present paper, we do not use the condition (1.11). Instead, we make other assumptions concerning the nature of the solutions of the difference equation, from which we determine, by quite elementary analysis, simple strict error bounds for the results yielded by the algorithm. From these bounds we are able to draw interesting conclusions concerning the general accuracy of the method.

We shall suppose at first that the algorithm is applied with the use of exact arithmetic. Later, in §4, we allow for the fact that in practice the trial values a_r are generally rounded at each step of the recurrence, thus introducing another possible source of error. Only in [8] does the effect of rounding the a_r appear to have been analysed before, but the treatment in this reference is of an approximate character, and is also less general than that presented below.

2. Assumptions. The basic assumption we make in this paper is that the difference equation (1.06) has independent solutions f_r and g_r such that $|f_r|$ has acquired a moderately rapidly decreasing character and $|g_r|$ a moderately rapidly increasing character at the stage ($r = n$) at which the algorithm is applied. More precisely, we suppose that f_n and g_{n+1} do not vanish, that

$$(2.01) \quad \rho = |f_{n+1}/f_n|$$

is small, and that the least quantity σ such that the inequalities

$$(2.02) \quad |g_r/g_{n+1}| \leq \sigma^{n+1-r}$$

hold for $r = 0, 1, \dots, n$, also is small. Here n is an arbitrary positive integer, and ρ, σ in general depend on n , but it is unnecessary to indicate this explicitly in the notation.

We further suppose that the application of the algorithm is delayed until the series on the left of (1.07) is converging reasonably rapidly after its $(n + 1)$ th term. To express this precisely, we introduce the notation

$$(2.03) \quad m = \max_{0 \leq r \leq n} |m_r|, \quad \tau = \left| \frac{1}{mf_n} \sum_{r=n+1}^{\infty} m_r f_r \right|.$$

Then the assumption of this paragraph is that τ is small. We may notice that this condition is automatically fulfilled by the simplest type of normalization, in which f_0 is prescribed (and non-zero), for in this case τ vanishes.

Perhaps it should be stressed that actual values for the quantities ρ, σ and τ do not need to be known in any given example, a mere knowledge that they are small suffices for the application of the error analysis of the following sections.

In the case of Miller's Bessel-function example, given in §1, the above conditions are fulfilled provided that n is taken to be somewhat larger than the argument x . This can be verified by inspecting tables of Bessel functions, or alternatively by examining the asymptotic behaviour of the Bessel functions for large orders. Similar verifications might be made in the case of other well-known special functions, for example, Legendre functions.

In any case, however, it is frequently quite straightforward to determine the main features of the asymptotic behaviour of f_r and g_r directly from the difference equation (1.06), as in the example given in §5 below. This will settle immediately the question whether the above conditions can be fulfilled; then inspection of the sequence of trial values will generally indicate whether n has been chosen large enough for f_r and g_r to have attained their asymptotic forms at $r = n$.

3. Error Analysis: Exact Arithmetic. The sequence a_r , constructed according to equations (1.08), can be expressed as a linear combination of the f_r and g_r .

Using the given values at $r = n$ and $n + 1$, we find that

$$(3.01) \quad a_r = \frac{g_{n+1}f_r - f_{n+1}g_r}{f_n g_{n+1} - f_{n+1}g_n},$$

the non-vanishing of the denominator being a consequence of the assumed conditions of §2 (compare (3.09) below). From (1.10) and (1.07), we find that

$$(3.02) \quad \lambda = \left\{ g_{n+1} \left(1 - \sum_{j=n+1}^{\infty} m_j f_j \right) - f_{n+1} \sum_{j=0}^n m_j g_j \right\} / (f_n g_{n+1} - f_{n+1} g_n).$$

The "error" in which we are interested is the difference between φ_r , that is, a_r/λ , and f_r . Rather than consider this directly, however, we shall seek bounds for the quantity e_r , defined by

$$(3.03) \quad e_r = \lambda(f_r - \varphi_r) = \lambda f_r - a_r = a_r(f_r - \varphi_r)/\varphi_r.$$

An interpretation of e_r is as follows. We have

$$\log_{10} |e_r| = \log_{10} |a_r| - \log_{10} |\varphi_r/(f_r - \varphi_r)|.$$

Hence $\log_{10} |e_r|$ is the excess of the number of digits in the integer part of a_r over the number of "correct" significant figures in φ_r . In particular, *if $|e_r| \leq 1$ then the number of correct significant figures in φ_r is not less than the number of digits in the integer part of the corresponding trial value a_r .*

Using (3.01) and (3.02), we derive

$$(3.04) \quad e_r = \left\{ f_{n+1}g_r - \left(\sum_{j=n+1}^{\infty} m_j f_j \right) g_{n+1}f_r - \left(\sum_{j=0}^n m_j g_j \right) f_{n+1}f_r \right\} / (f_n g_{n+1} - f_{n+1} g_n).$$

Now when $0 \leq r \leq n$, we find from (2.01) and (2.02) that

$$(3.05) \quad |f_{n+1}g_r| \leq \rho\sigma^{n+1-r} |f_n g_{n+1}|.$$

With the aid of (2.03), we obtain

$$(3.06) \quad \left| \left(\sum_{j=n+1}^{\infty} m_j f_j \right) g_{n+1}f_r \right| \leq a\tau |f_n g_{n+1}|,$$

where

$$(3.07) \quad a = m \max_{0 \leq r \leq n} |f_r|.$$

Similarly,

$$(3.08) \quad \left| \left(\sum_{j=0}^n m_j g_j \right) f_{n+1}f_r \right| \leq \left(\sum_{j=0}^n m |g_{n+1}| \sigma^{n+1-j} \right) \rho |f_n| \frac{a}{m} \leq \frac{a\rho\sigma}{1-\sigma} |f_n g_{n+1}|.$$

Next, we have $|f_{n+1}g_n| \leq \rho\sigma |f_n g_{n+1}|$, and so

$$(3.09) \quad |f_n g_{n+1} - f_{n+1} g_n| \geq (1 - \rho\sigma) |f_n g_{n+1}|.$$

Substituting (3.05), (3.06), (3.08) and (3.09) in (3.04), we obtain our desired error bound:

$$(3.10) \quad |e_r| \leq (1 - \rho\sigma)^{-1} \{ \sigma^{n-r} + (1 - \sigma)^{-1} a \} \rho\sigma + a\tau \quad (0 \leq r \leq n).$$

With the assumed conditions of §2, the quantities ρ , σ and τ are all small. Clearly if a is not large compared with unity then $|e_r|$ will not exceed unity.

The interpretation of this extra condition depends on the relative magnitudes of the coefficients m_r . We may judge from examples, however, that it is a reasonable condition to impose on the use of the algorithm. Suppose, for example, we normalize by use of a prescribed value of f_0 . Then we have $m = 1/|f_0|$, and

$$a = \max_{0 \leq r \leq n} |f_r/f_0|.$$

Thus a could be large only if f_0 is small compared with other members of the sequence f_r , and we could not expect this method of normalization to be accurate in these circumstances. Alternatively, consider the case in which

$$m_0 = \frac{1}{2}m, \quad m_j = \pm m \quad (j \geq 1);$$

this occurs with (1.04), and also frequently with the boundary conditions in the Clenshaw method [13]. Then by requiring a to be of order unity, we are stipulating that there must not be excessive cancellation in summing the series (1.07), and it is known [1] that if such cancellation occurs, then inaccuracies may result.

Summarizing this section so far, we have shown that with the conditions of §2 and the restriction that the quantity a defined by (3.07) must not be large compared with unity, $|e_r|$ cannot exceed unity. Accordingly, when the algorithm is carried out with exact arithmetic each φ_r has at least as many correct significant figures as there are digits in the integer part of the corresponding a_r .

Remark. It might perhaps be thought that the condition that the second solution g_r is increasing at $r = n$ is overly strong, and that g_r could be permitted to decrease, provided that its rate of decrease were less than that of f_r . Certainly in these circumstances the algorithm may converge; see, for example, (1.11) and (1.12). In general, however, it would be untrue that $|e_r| \leq 1$; this is readily seen by considering the contribution of the term

$$\frac{f_{n+1} g_r}{f_n g_{n+1} - f_{n+1} g_n} \approx \frac{f_{n+1}}{f_n} \frac{g_r}{g_{n+1}}$$

on the right of (3.04). Thus in these circumstances conclusions concerning the accuracy of the sequence φ_r would necessarily be weaker.

4. Effect of Rounding the Sequence of Trial Values. In applications, after computing each a_r it is generally the practice to round its value to the nearest integer. To examine the effect of this upon the final sequence φ_r , it suffices to consider the propagation of a single rounding error, say ϵ_s in a_s , where $0 \leq s \leq n - 1$, and $|\epsilon_s| \leq \frac{1}{2}$.

In this analysis, we shall use the well-known identity

$$(4.01) \quad f_s g_{s+1} - f_{s+1} g_s = q_{s+1} (f_{s+1} g_{s+2} - f_{s+2} g_{s+1}),$$

obtained directly from the difference equation (1.06). Repeated application of this result gives

$$(4.02) \quad f_s g_{s+1} - f_{s+1} g_s = q_{s+1} q_{s+2} \cdots q_n (f_n g_{n+1} - f_{n+1} g_n).$$

As a consequence of (3.09), we see that the cross-product on the left of this equation

cannot vanish, provided that no coefficient q_r vanishes; we shall assume this to be the case.

Equation (3.01) now holds only for $r > s$. For other values of r , we find that

$$(4.03) \quad a_r = \frac{g_{n+1}f_r - f_{n+1}g_r}{f_n g_{n+1} - f_{n+1}g_n} + \frac{g_{s+1}f_r - f_{s+1}g_r}{f_s g_{s+1} - f_{s+1}g_s} \epsilon_s \quad (r \leq s).$$

Accordingly, we obtain from (1.10) and (1.07)

$$(4.04) \quad \lambda = \frac{g_{n+1} \left(1 - \sum_{j=n+1}^{\infty} m_j f_j \right) - f_{n+1} \sum_{j=0}^n m_j g_j}{f_n g_{n+1} - f_{n+1} g_n} + \frac{g_{s+1} \left(1 - \sum_{j=s+1}^{\infty} m_j f_j \right) - f_{s+1} \sum_{j=0}^s m_j g_j}{f_s g_{s+1} - f_{s+1} g_s} \epsilon_s.$$

Again writing $e_r = \lambda f_r - a_r$, we find that

$$(4.05) \quad e_r = e_r^{(0)} + \gamma_{r,s} \epsilon_s,$$

where $e_r^{(0)}$ denotes the right side of (3.04) (that is, the e_r of §3), and

$$(4.06) \quad (f_s g_{s+1} - f_{s+1} g_s) \gamma_{r,s} = \begin{cases} f_{s+1} g_r - \left(\sum_{j=s+1}^{\infty} m_j f_j \right) g_{s+1} f_r \\ \quad - \left(\sum_{j=0}^s m_j g_j \right) f_{s+1} f_r & (r \leq s), \\ \left(1 - \sum_{j=s+1}^{\infty} m_j f_j \right) g_{s+1} f_r \\ \quad - \left(\sum_{j=0}^s m_j g_j \right) f_{s+1} f_r & (r \geq s+1). \end{cases}$$

The bound (3.10) applies to $e_r^{(0)}$. To derive corresponding bounds for $\gamma_{r,s}$, we denote by v the largest quantity for which the inequalities

$$(4.07) \quad \left| \frac{f_r}{f_n} \right| \leq \frac{1}{v^{n-r}} \quad (r = 0, 1, \dots, n)$$

hold. Since we have assumed in §2 that $|f_r|$ has acquired a moderately rapidly decreasing character by the time the point $r = n$ has been reached, it follows that v is small.

From (2.02), (2.03), (3.07) and (4.07), we find that

$$(4.08) \quad |f_{s+1} g_r| \leq \sigma^{n+1-r} v^{s+1-n} |f_n g_{n+1}|, \quad |g_{s+1} f_r| \leq \sigma^{n-s} v^{r-n} |f_n g_{n+1}|,$$

$$(4.09) \quad \left| \left(\sum_{j=s+1}^{\infty} m_j f_j \right) g_{s+1} f_r \right| \leq \left(\sum_{j=s+1}^n m v^{j-n} + m \tau \right) |f_n g_{n+1}| \sigma^{n-s} \frac{a}{m} \\ \leq \left(\frac{1}{1-v} + \tau v^{n-s-1} \right) a \sigma^{n-s} v^{s+1-n} |f_n g_{n+1}|,$$

and

$$(4.10) \quad \left| \left(\sum_{j=0}^s m_j g_j \right) f_{s+1} f_r \right| \leq \left(\sum_{j=0}^s m \sigma^{n+1-j} |g_{n+1}| \right) v^{s+1-n} |f_n| \frac{a}{m} \\ \leq \frac{a \sigma^{n+1-s} v^{s+1-n}}{1 - \sigma} |f_n g_{n+1}|.$$

Also, from (3.09) and (4.02) we have

$$(4.11) \quad |f_s g_{s+1} - f_{s+1} g_s| \geq |g_{s+1} g_{s+2} \cdots g_n| (1 - \rho \sigma) |f_n g_{n+1}|.$$

Substituting these inequalities in (4.06) and using the notation

$$(4.12) \quad b = \max_{0 \leq s \leq n-1} \left\{ \frac{1}{|g_{s+1} g_{s+2} \cdots g_n|} \left(\frac{\sigma}{v} \right)^{n-s} \right\},$$

we find that

$$(4.13) \quad |\gamma_{r,s}| \leq b v (1 - \rho \sigma)^{-1} [\sigma^{s-r+1} + a \{ \tau v^{n-s-1} + (1 - v)^{-1} + \sigma (1 - \sigma)^{-1} \}] \\ (r \leq s),$$

and

$$(4.14) \quad |\gamma_{r,s}| \leq b v (1 - \rho \sigma)^{-1} [v^{r-s-1} + a \{ \tau v^{n-s-1} + (1 - v)^{-1} + \sigma (1 - \sigma)^{-1} \}] \\ (r \geq s + 1).$$

In §3 we observed that if ρ, σ, τ are small and a is of order unity, then $|e_r^{(0)}| < 1$. Inspection of (4.13) and (4.14) shows that since v is small we shall also have $|\gamma_{r,s}| < 1$, provided that b is not large compared with unity. In order to evaluate this additional condition, consider the case in which p_r and q_r are both independent of r . The difference equation (1.06) is then solvable explicitly, and we find that $\sigma/v = q_r$ and $b = 1$. In the general case we have to assume that p_r and q_r are relatively slowly-varying functions of r . Then f_r and g_r behave locally as solutions of a difference equation with constant coefficients and hence we may expect b to be of order unity.

Thus we conclude that if the sequence of trial values a_r is rounded to the nearest integer at each step of the algorithm, then each φ_r still has at least as many correct significant figures as there are digits in the a_r .† In other words, the rounding procedure is justified, in the sense that it does not adversely affect our “guaranteed” accuracy of the algorithm. This point is discussed further in the next section.

5. Example. We consider the solution of the difference equation

$$(5.01) \quad (2r - 1)f_{r-1} - 12rf_r + (2r + 1)f_{r+1} = 0,$$

with the condition

$$(5.02) \quad \frac{1}{2}f_0 + f_1 + f_2 + f_3 + \cdots = 1.$$

This problem arose in the solution of a first-order differential equation by Clenshaw's method ([14], §15).

† In the case of the modified Bessel functions, this result is stated, without proof, in [3] as a consequence of practical observation.

We apply formulas (1.08) to (1.10), taking

$$p_r = -12r/(2r - 1), \quad q_r = (2r + 1)/(2r - 1)$$

and $n = 4$. The following sequences of trial values are then obtained.

r	a_r (exact)	a_r (rounded)	ϵ_r
0	$4324\frac{1}{5}$	4413	0
1	$372\frac{1}{3}\frac{2}{5}$	380	$-\frac{1}{3}$
2	$47\frac{2}{3}\frac{3}{5}$	49	0
3	$6\frac{6}{7}$	7	$\frac{1}{7}$
4	1	1	0
5	0	0	0

Exact arithmetic has been used to compute the column headed a_r (exact). In forming the next column, however, the a_r have been rounded to the nearest integer at each step, the corresponding rounding errors ϵ_r being given in the fourth column.

We now form λ by means of (1.10), taking $m_0 = \frac{1}{2}$ and $m_j = 1$ ($j \geq 1$). This gives

$$\lambda = 2590\frac{1}{7}\frac{9}{6} \quad (\text{for exact } a_r), \quad \lambda = 2643\frac{1}{2} \quad (\text{for rounded } a_r).$$

The algorithm is completed by computing $\varphi_r = a_r/\lambda$, and the results are given in the third and fifth columns in the next table.

r	f_r	exact a_r		rounded a_r	
		φ_r	e_r	φ_r	e_r
0	1.66925 37	1.66940 03	-0.380	1.66937 77	-0.328
1	0.14373 42	0.14374 67	-0.032	0.14374 88	-0.039
2	0.01851 87	0.01851 98	-0.003	0.01853 60	-0.046
3	0.00264 94	0.00264 73	0.005	0.00264 80	0.004
4	0.00039 79	0.00038 61	0.031	0.00037 83	0.052

For comparison, we also give in this table accurate seven-decimal values of the f_r , obtained by use of the algorithm with a higher value of n . Thence we are able to evaluate our measure e_r , defined by (3.03), for the discrepancy between f_r and φ_r .

Our principal observation concerning these numerical results is that $|e_r|$ nowhere exceeds unity. This result is predicted by the analysis of the preceding sections, because the assumed conditions are readily seen to be satisfied in this example. For large r the difference equation (5.01) becomes asymptotically

$$f_{r-1} - 6f_r + f_{r+1} = 0,$$

with solutions

$$f_r = (\text{constant}) \times \alpha^{-r}, \quad g_r = (\text{constant}) \times \alpha^r, \quad \text{where } \alpha = 3 + 2\sqrt{2} = 5.83 \dots$$

It is clear from the computed numerical values that this asymptotic character is attained by the decreasing solution f_r by the stage $r = 4$. It is also quite easy to see, without carrying out detailed arithmetic, that the solution g_r will have attained its rapidly increasing character by this stage. Therefore the quantities

ρ, σ, τ and v defined in §§2 and 4 are all small. The other conditions, that a and b (defined in §§3 and 4) be of order unity, are seen to be fulfilled, first because there was no cancellation in summing the series (1.10), and secondly because the coefficients p_r and q_r are relatively slowly-varying functions of r .

As a further illustration, let us normalize by means of the given value $f_0 = 1.66925\ 37$, instead of by use of (5.02). We then obtain the following results.

r	exact a_r		rounded a_r	
	φ_r	e_r	φ_r	e_r
0	1.66925 37	0	1.66925 37	0
1	0.14373 40	0.001	0.14373 81	-0.010
2	0.01851 82	0.001	0.01853 47	-0.042
3	0.00264 70	0.006	0.00264 78	0.004
4	0.00038 60	0.031	0.00037 83	0.052

Again $|e_r| < 1$, as expected. It may also be observed that this method of normalization yields somewhat smaller errors $|e_r|$ than the previous method in the case when the a_r are computed exactly. The bound (3.10) indicates that this is generally true, unless the quantity a happens to be unduly large, because of the vanishing of the term in τ . On the other hand, we may expect the contributions of the rounding errors ϵ_s in the two methods to be comparable, because the dominant terms $a(1 - v)^{-1}$ and v^{r-s-1} (when $r = s + 1$) in the square brackets in (4.13) and (4.14) are of comparable size in the two cases. By forming the differences between the values of e_r obtained from the exact and rounded a_r in the two methods, we see that this conclusion, also, is borne out by the example.

As a final observation, we remark that the column of values given in this section headed a_r (exact) differs from the corresponding numerical values given by Fox in the column $a_r^{(0)}$ in Table 2 of [14]. This is because Fox, in effect, uses different initial conditions for his sequence. By superimposing an artificial "rounding error" $\epsilon_2 = -9/5$ on our values of a_r (exact), however, we reproduce Fox's sequence (apart from differences in sign); in this way his results are brought within the scope of the present error analysis.

6. Conclusion. The Miller algorithm for the computation of the solution of the difference equation (1.06) which satisfies the condition (1.07) is given by (1.08), (1.09) and (1.10). We have shown that if the wanted solution f_r and an independent solution g_r satisfy the relations (2.01) and (2.02), in which ρ and σ are positive numbers less than unity, then the errors in the values φ_r yielded by the algorithm are bounded by the strict inequality (3.10), in which $e_r = \lambda(f_r - \varphi_r)$ and τ and a are defined by (2.03) and (3.07).

We have also investigated the effects of rounding the sequence of trial values a_r , and have found that an error ϵ_s introduced in a_s increases e_r by $\gamma_{r,s}\epsilon_s$, where $\gamma_{r,s}$ is subject to the strict inequalities (4.13) and (4.14) in which $v (< 1)$ and b are defined by (4.07) and (4.12), respectively.

We have seen that in circumstances in which the algorithm is commonly applied, ρ, σ, τ and v are small, and a and b are not large compared with unity. In consequence $|e_r| < 1$ and $|\gamma_{r,s}| < 1$, which mean that the number of correct significant figures

in each φ_r yielded by the algorithm is not less than the number of digits in the integer part of the corresponding trial value a_r , whether or not the a_r are rounded during the computation. Thus the practice of repeating the algorithm with different starting positions n merely to ascertain the accuracy of the φ_r is often quite unnecessary.

This theory has application to the automatic computation of many transcendental functions, and also to Clenshaw's method of solving ordinary linear differential equations in series of Chebyshev polynomials in the case when the coefficients in this series satisfy a three-term recurrence relation. A comprehensive error analysis of the Clenshaw method has not been attempted, but by making assumptions of the type used in this paper it is quite possible that similar analysis will yield useful bounds in more complicated cases.

Acknowledgement. The author thanks Mr. C. W. Clenshaw and Mr. G. F. Miller of the National Physical Laboratory, Teddington, for reading the manuscript of this paper and making valuable criticisms.

National Bureau of Standards
Washington, D. C.

1. BRITISH ASSOCIATION FOR THE ADVANCEMENT OF SCIENCE, "Bessel functions, Part II," *Mathematical Tables*, v. X, Cambridge University Press, 1952.
2. C. W. JONES, "A short table for the Bessel functions $I_{n+\frac{1}{2}}(x)$, $(2/\pi)K_{n+\frac{1}{2}}(x)$," *Royal Society Shorter Mathematical Tables*, No. 1, Cambridge University Press, 1952.
3. L. FOX, "A short table for Bessel functions of integer orders and large arguments," *Royal Society Shorter Mathematical Tables*, No. 3, Cambridge University Press, 1954.
4. F. J. CORBATÓ, "On the computation of auxiliary functions for two-center integrals by means of a high-speed computer," *J. Chem. Phys.*, v. 24, 1956, p. 452-453.
5. I. A. STEGUN & M. ABRAMOWITZ, "Generation of Bessel functions on high speed computers," *MTAC*, v. 11, 1957, p. 255-257.
6. J. B. RANDELS & R. F. REEVES, "Note on empirical bounds for generating Bessel functions," *Comm. Assoc. Comput. Mach.*, v. 1, May 1958, p. 3-5.
7. M. GOLDSTEIN & R. M. THALER, "Recurrence techniques for the calculation of Bessel functions," *MTAC*, v. 13, 1959, p. 102-108.
8. F. J. CORBATÓ & J. L. URETSKY, "Generation of spherical Bessel functions in digital computers," *J. Assoc. Comput. Mach.*, v. 6, 1959, p. 366-375.
9. NATIONAL BUREAU OF STANDARDS, "Handbook of mathematical functions," *Appl. Math. Ser.* 55, Government Printing Office, Washington, D. C. (In press.) (Especially Section 9.12, Examples 1 and 7.)
10. M. ABRAMOWITZ, Review of a paper by J. Kaye, *MTAC*, v. 10, 1956, p. 176-177.
11. W. GAUTSCHI, "Recursive computation of the repeated integrals of the error function," *Math. Comp.*, v. 15, 1961, p. 227-232.
12. A. ROTENBERG, "The calculation of toroidal harmonics," *Math. Comp.* v. 14, 1960, p. 274-276.
13. C. W. CLENSHAW, "The numerical solution of linear differential equations in Chebyshev series," *Proc. Cambridge Phil. Soc.*, v. 53, 1957, p. 134-149.
14. L. FOX, "Chebyshev methods for ordinary differential equations," *Computer J.*, v. 4, 1962, p. 318-331.