

A Class of A -Stable Advanced Multistep Methods

By Jack Williams and Frank de Hoog

Abstract. A class of A -stable advanced multistep methods is derived for the numerical solution of initial value problems in ordinary differential equations. The methods, of all orders of accuracy up to ten, only require values of y' and are self starting. Results for the asymptotic behaviour of the discretization error and for estimating local truncation error are also obtained. The practical implementation of the fourth order method is described and the method applied to some stiff equations. Numerical comparisons are made with Gear's method.

1. **Introduction.** Recently, particular attention has been given to the study of A -stable methods for the solution of the m ordinary differential equations

$$(1.1) \quad y' = f(x, y), \quad y(a) = \eta, \quad x \in [a, b].$$

Let $\{y_n\}$ denote the solution of some difference method approximation of $y' = qy$, $y(0) = 1$, on the mesh $\{x_n\}$ with fixed step size h . Then the difference method is called A -stable (in the sense of Dahlquist [3]) if $\|y_n\|_\infty \rightarrow 0$ as $n \rightarrow \infty$ for any fixed $h > 0$ and any scalar constant q with $\operatorname{Re}(q) < 0$. This very strong stability requirement is particularly useful for the integration of stiff systems, that is, where the Jacobian matrix $(\partial f / \partial y)$ has some eigenvalues λ_i for which $\operatorname{Re}(\lambda_i)$ are negative and have greatly differing magnitudes.

Dahlquist [3] shows that within the class of linear multistep methods, A -stable methods are necessarily implicit and that the A -stable formula having the smallest truncation error is the trapezoidal rule (of order $p = 2$). Examples of more accurate A -stable methods are given in [7], [1], [2], [12], [18], [19] and [20]. Lapidus and Seinfeld [14] and Gear [10] also discuss and reference many other A -stable methods. In addition, the report of Dahlquist et al. [4] provides a valuable survey of methods for stiff systems.

Our main concern in this paper is with methods for the solution of stiff systems. Particularly relevant here are the implicit methods described by Rosser [16], Watts and Shampine [17], [18], and Watts [19]. From the viewpoint of advanced multistep methods, we discuss a subclass of the block implicit methods (in the terminology of Shampine and Watts) in [18] which are A -stable for all orders up to ten. An important feature of the methods is that they are used in practice like one-step methods and thus have a simple step-changing facility. Although essentially discrete variable methods, we show that they can easily be used to obtain convergent global approximations of $y(x)$ and its derivatives.

Finally, an algorithm is fully described for the practical implementation of the

Received January 11, 1971, revised May 29, 1973.

AMS (MOS) subject classifications (1970). Primary 65L05.

Key words and phrases. A -stable, advanced multistep methods, stiff systems.

Copyright © 1974, American Mathematical Society

fourth order method applied to stiff systems. Some numerical results are presented and compared with the results obtained by Gear's method.

2. Advanced Multistep Methods. The Results of Daniel. In general, methods using values of $f(x, y)$, which are advanced further in x than are the values of the dependent variable used to approximate y' , will be referred to as advanced multistep methods (AMM). A certain type of AMM has been considered by Daniel [5] (also see [6]), whose results we now describe.

Let $h = (b - a)/M$, $x_i = a + ih$, $0 \leq i \leq M$, where M is a positive integer. Given starting values $y_\mu = \eta_\mu$, $0 \leq \mu \leq k - 1$, a k -step AMM consists of finding a solution to

$$(2.1) \quad \sum_{i=0}^k \alpha_{k-i} y_{n-i} = h \sum_{i=n-l_n}^{n+u_n} \beta_{ni} f(x_i, y_i), \quad k \leq n \leq M.$$

We may summarize Daniel's main results in

THEOREM 2.1 (DANIEL). *Let the AMM (2.1) satisfy*

(a)
$$\rho(z) = \alpha_k z^k + \alpha_{k-1} z^{k-1} + \dots + \alpha_0$$

has zeros z_i , $1 \leq i \leq k$, which satisfy $|z_i| \leq 1$, $|z_i| = 1 \Rightarrow \rho'(z_i) \neq 0$ (stability condition).

(b)
$$|\beta_{ni}| \leq \beta, \quad 0 \leq u_n \leq u, \quad 0 \leq l_n \leq l,$$

$\|f(x, v) - f(x, w)\| \leq L\|v - w\|$, with β, u, l and L all independent of h .

(c)
$$\|\eta_\mu - y(x_\mu)\| = O(h^p), \quad 0 \leq \mu \leq k - 1,$$

$\|\sum_{i=0}^k \alpha_{k-i} y(x_{n-i}) - h \sum_{i=n-l_n}^{n+u_n} \beta_{ni} f(x_i, y(x_i))\| = O(h^{p+1})$, $k \leq n \leq M$ (consistency condition).

Then, for sufficiently small h , (2.1) has a unique solution with

$$\max_{0 \leq i \leq M} \|y_i - y(x_i)\| = O(h^p) \quad \text{as } h \rightarrow 0 \text{ where } \|\cdot\| = \|\cdot\|_\infty.$$

3. A Class of AMM's. Let $x_n = a + nh$, $n \geq 0$, and consider the subintervals $[x_{nk}, x_{nk+k}]$, $n = 0, 1, 2, \dots, N - 1$, where $k \geq 1$ and N are given fixed integers with $b - a = Nkh$. From (1.1),

$$y(x_{nk+r}) - y(x_{nk+r-1}) = \int_{x_{nk+r-1}}^{x_{nk+r}} f(x, y(x)) dx, \quad 1 \leq r \leq k.$$

We now approximate the components of $f(x, y(x))$ over $[x_{nk}, x_{nk+k}]$ by an interpolating polynomial of degree k , yielding the k -step AMM

$$(3.1) \quad y_{nk+r} - y_{nk+r-1} = h \sum_{s=0}^k \beta_{rs} f_{nk+s}, \quad 1 \leq r \leq k, \quad 0 \leq n \leq N - 1,$$

where $f_j = f(x_j, y_j)$.

The results of Daniel may now be applied where we restrict the class of initial value problems (1.1) to those with sufficiently smooth solutions $y(x)$. Here, $\rho(z) = z - 1$ and, by construction, (3.1) has order of accuracy $p = k + 1$. There follows from Theorem 2.1,

THEOREM 3.1. *For starting values of order of accuracy $O(h^{k+1})$, the AMM (3.1) is convergent with rate of convergence at least $O(h^{k+1})$.*

For a system of m differential equations, define

$$Y_{n+1} = (y_{nk+1}^T, y_{nk+2}^T, \dots, y_{nk+k}^T)^T, \quad F_{n+1} = (f_{nk+1}^T, f_{nk+2}^T, \dots, f_{nk+k}^T)^T.$$

$0 \leq n \leq N - 1$, then, in terms of the appropriately defined $mk \times mk$ block matrices A, B, C and D , the k -step AMM (3.1) may be written

$$(3.2) \quad DY_{n+1} - AY_n = h(BF_{n+1} + CF_n),$$

$0 \leq n \leq N - 1$, where we define $Y_0 = (y_0^T, y_0^T, \dots, y_0^T)^T$ and $F_0 = (f_0^T, f_0^T, \dots, f_0^T)^T$.

Although (3.2) is referred to as a k -step method, it is computationally a one-step method with respect to the vector of values Y_{n+1} , since only y_{nk} and f_{nk} , the values from one point, are required for its calculation. Consequently, after computing a vector of values Y_{n+1} , a change in step length can easily be made.

Having applied Daniel's results, (3.2) may be written in a form more suitable for computation. With $G = D^{-1}A$, $Q = D^{-1}B$ and $R = D^{-1}C$, (3.2) becomes

$$(3.3) \quad Y_{n+1} - GY_n = h(QF_{n+1} + RF_n), \quad 0 \leq n \leq N - 1.$$

Due to the type of construction, these formulas are the same as those discussed in [18]. For $k = 2, 3$, the methods have the following explicit forms.

$k = 2.$

$$y_{2n+1} - y_{2n} = \frac{h}{12} (5f_{2n} + 8f_{2n+1} - f_{2n+2}),$$

$$y_{2n+2} - y_{2n} = \frac{h}{3} (f_{2n} + 4f_{2n+1} + f_{2n+2}).$$

$k = 3.$

$$y_{3n+1} - y_{3n} = \frac{h}{24} (9f_{3n} + 19f_{3n+1} - 5f_{3n+2} + f_{3n+3}),$$

$$y_{3n+2} - y_{3n} = \frac{h}{3} (f_{3n} + 4f_{3n+1} + f_{3n+2}),$$

$$y_{3n+3} - y_{3n} = \frac{h}{8} (3f_{3n} + 9f_{3n+1} + 9f_{3n+2} + 3f_{3n+3}).$$

Additional explicit formulas, for $4 \leq k \leq 8$, can be found in Rosser [16] or Watts [19].

4. A-Stability.

THEOREM 4.1. *The k -step AMM (3.3) is A-stable for each $1 \leq k \leq 8$. For $9 \leq k \leq 20$, the method is not A-stable.*

The results in Theorem 4.1 have been proved by Watts and Shampine [18] and Watts [19]. Also see Wright [20]. In view of this and the fact that our matrix method of proof is rather long, we shall not include it here. Unfortunately the proof in [18] and our proof both suffer from the fact that numerical methods are used in the final stages. In [18], the Routh-Hurwitz theorem is applied numerically to check the location of polynomial zeros; however, for $1 \leq k \leq 10$, the authors of [18]

were able to use exact arithmetic to establish the results of Theorem 4.1 conclusively. Our approach led to the use of the QR algorithm for checking that certain $k \times k$ matrices have eigenvalues with negative real parts. Hence, by using a different approach, we are able to confirm the results of Watts and Shampine.

5. Asymptotic Behaviour of the Discretization Error. In this section, we consider the asymptotic form of the discretization error for the k -step AMM (3.3). We treat the scalar differential equation and make the following assumptions:

- (i) The solution $y(x)$ has sufficiently many continuous derivatives.
 - (ii) $g(x) = (\partial f/\partial y)(x, y(x))$ is continuous and continuously differentiable for $x \in [a, b]$.
 - (iii) The starting value y_0 satisfies $\delta(h) = y_0 - y(a) = O(h^r)$, $r \geq k + 2$.
- We associate with the AMM (3.3) the difference operator

$$L[\Phi(x), h] = \Phi(x) - G\Phi(x - kh) - h(Q\Phi'(x) + R\Phi'(x - kh)),$$

with

$$\begin{aligned} \Phi(x) &= (\Phi(x + h), \Phi(x + 2h), \dots, \Phi(x + kh))^T, \\ \Phi'(x) &= (\Phi'(x + h), \Phi'(x + 2h), \dots, \Phi'(x + kh))^T, \end{aligned}$$

where $\Phi(x)$ is any sufficiently differentiable function. We find that

$$(5.1) \quad L[\Phi(x), h] = h^{k+2}\Phi^{(k+2)}(x)\alpha + h^{k+3}\Phi^{(k+3)}(x)\beta + O(h^{k+4}),$$

where α and β are k component vectors whose elements are constants. In particular, it can be shown that, for k even, $\alpha_k = 0$.

Now by using Theorem 3.1 and similar techniques presented in Henrici [11], results for the asymptotic behaviour of the discretization error $e_n = y_n - y(x_n)$ can be obtained. Since, however, similar results are developed in Watts [19], we shall not present proofs, but, for the sake of completeness and to aid the development of Sections 6 and 7, state the results as

THEOREM 5.1. *Under the conditions (i), (ii), (iii) above, the asymptotic behaviour of the discretization error in the solution of $y' = f(x, y)$, $y(a) = \eta$ by AMM (3.3) as $h \rightarrow 0$, $x = a + (kn + j)h$, $1 \leq j \leq k$, $0 \leq n \leq N - 1$, is given by:*

$$\begin{aligned} \text{for } k \text{ odd, } e_{kn+j} &= h^{k+1}e_1(x) + \delta(h)e_3(x) + O(h^{k+2}), \\ \text{for } k \text{ even, } e_{kn+j} &= h^{k+2}[e_2(x) - y^{(k+2)}(x)\alpha_j] + \delta(h)e_3(x) + O(h^{k+3}). \end{aligned}$$

The functions $e_1(x)$, $e_2(x)$ and $e_3(x)$ are defined as the solutions of certain initial value problems involving $g(x)$, $y^{(k+2)}(x)$ and $y^{(k+3)}(x)$.

Realistically, $\delta(h)$ may be identically zero in practice. For the application of extrapolation to the limit with k even, we must take account of the fact that the dominant term does not vary smoothly with j . For a given initial step length h , one possible method is to proceed with the normal halving of step lengths but to compute extrapolated values only for $x = a + knh$, $1 \leq n \leq N$. This set of x values is then always associated with the same value of j ($j = k$) for subsequent step lengths.

6. Global Convergence Properties. Consider $Y(x) \in C^1[a, b]$ defined by

$$(6.1) \quad Y(x) = y_{nk} + \int_{x_{nk}}^x F_n(t) dt, \quad x \in [x_{nk}, x_{nk+k}],$$

for $0 \leq n \leq N - 1$, where $F_n(t) = \sum_{s=0}^k l_s(t)f(x_{nk+s}, y_{nk+s})$ the Lagrangian form of the interpolating polynomial appearing in (3.1). Clearly by the definition of the quantities y_n , $0 \leq n \leq Nk$, $Y(x)$ and $Y'(x)$ interpolate the values of y_n and f_n , respectively. Hence in practice, for $x \in (x_{nk}, x_{nk+k})$, $Y(x) \equiv P(x)$, the polynomial of degree $(k + 1)$ uniquely determined by $P(x_{nk+r}) = y_{nk+r}$, $0 \leq r \leq k$, $P'(x_{nk}) = f_{nk}$.

We have the following theorem which states that $Y(x)$ and its appropriately defined $k + 1$ derivatives provide convergent global approximations to $y^{(r)}(x)$, $0 \leq r \leq k + 1$. The theorem is presented for the important case of k even.

THEOREM 6.1. *Let $f(x, y)$ be sufficiently smooth in $D_H := \{(x, y): a \leq x \leq b, |y - y(x)| \leq H\}$, a neighbourhood of the solution $y(x)$. Then for h sufficiently small and for finite k , k even, $x \in [a, b]$,*

$$(6.2) \quad Y(x) = y(x) + O(h^{k+2}),$$

$$(6.3) \quad Y^{(r)}(x) = y^{(r)}(x) + O(h^{k+2-r}), \quad 1 \leq r \leq k + 1,$$

and at the mesh points

$$Y'(x_{nk+j}) = y'(x_{nk+j}) + O(h^{k+2}), \quad 1 \leq j \leq k, 0 \leq n \leq N - 1.$$

We define for $2 \leq r \leq k + 1$, $1 \leq n \leq N - 1$,

$$(6.4) \quad Y^{(r)}(x_{nk}) = \frac{1}{2}(Y_+^{(r)}(x_{nk}) + Y_-^{(r)}(x_{nk})),$$

the mean of the right-hand and left-hand derivatives of $Y(x)$ at $x = x_{nk}$.

Proof. It has been shown in Theorem 5.1 that, for h sufficiently small and for $O(h^{k+2})$ starting values,

$$(6.5) \quad Y(x_n) = y(x_n) + O(h^{k+2}), \quad 0 \leq n \leq Nk.$$

Hence, since k is finite,

$$F_n(t) = \sum_{s=0}^k l_s(t)f(x_{nk+s}, y(x_{nk+s})) + O(h^{k+2})$$

for $0 \leq n \leq N - 1$, $t \in [x_{nk}, x_{nk+k}]$. The error formula for Lagrangian interpolation, as applied to the function $y'(t)$, now gives

$$F_n(t) = y'(t) + O(h^{k+1}).$$

Inserting this result and (6.5) into (6.1) implies (6.2). For $x \in (x_{nk}, x_{nk+k})$, $0 \leq n \leq N - 1$, differentiation of (6.1) yields

$$Y^{(r)}(x) = F_n^{(r-1)}(x), \quad 1 \leq r \leq k + 1.$$

Now similarly, using the error formula for the differentiated form of the interpolating polynomial [15, p. 78] gives

$$Y^{(r)}(x) = y^{(r)}(x) + O(h^{k+2-r}).$$

Hence with the definition (6.4), there follows (6.3) for $2 \leq r \leq k + 1$ and the cases $r = 1$ follow with aid of the interpolating property of $Y'(x)$.

7. Estimation of Local Truncation Error. The results of Section 5 allow the development of rigorous results for estimating the size of the local truncation error. For practical computation, we need only consider the case of k even. We now regard the AMM (3.3) as a corrector formula which is used by iterating to convergence. By introducing a suitable predictor formula, the local truncation error in (3.3) can be estimated by a generalization of Milne's method [11, p. 257]. The analysis applies to the system (1.1).

For the corrector

$$(7.1) \quad Y_{n+1} - GY_n = h(QF_{n+1} + RF_n),$$

(5.1) shows that the true solution $y(x)$ satisfies

$$(7.2) \quad Y(n + 1) - GY(n) = h(QF(n + 1) + RF(n)) + h^{k+2}\tau_n + O(h^{k+3}),$$

(with $Y(n + 1) = (y(x_{kn+1}), y(x_{kn+2}), \dots, y(x_{kn+k}))^T$) where since $\alpha_k = 0$ (k even),

$$\tau_n = (\alpha_1 y^{(k+2)}(x_{kn}), \alpha_2 y^{(k+2)}(x_{kn}), \dots, \alpha_{k-1} y^{(k+2)}(x_{kn}), 0)^T.$$

We consider the predictor formula

$$(7.3) \quad Y_{n+1}^* - GY_n = h(Q^*F_n + R^*F_{n-1}),$$

where $R^*F_{n-1} = (r_1 f_{(n-1)k}, r_2 f_{(n-1)k}, \dots, r_k f_{(n-1)k})^T$, and

$$(7.4) \quad Y(n + 1) - GY(n) = h(Q^*F(n) + R^*F(n - 1)) + h^{k+2}\sigma_n + O(h^{k+3})$$

with $\sigma_n = (d_1 y^{(k+2)}(x_{kn}), d_2 y^{(k+2)}(x_{kn}), \dots, d_k y^{(k+2)}(x_{kn}))^T$. The formula (7.3) yields the predicted values $Y_{n+1}^* = (y_{nk+1}^*, y_{nk+2}^*, \dots, y_{nk+k}^*)^T$. Our objective is to now estimate with respect to the maximum norm, $\|\tau_n\|$. We shall assume that $d_i \neq \alpha_i$ for $1 \leq i \leq k - 1$.

THEOREM 7.1. *Let the predictor formula (7.3) satisfy (7.4) and let the conditions of Theorem 5.1 be satisfied. Then, for k even, the approximation τ_n for the local truncation error in the corrector formula (7.1) satisfies, for $x_{kn} = a + knh$, $0 \leq n \leq N - 1$ as $h \rightarrow 0$,*

$$\|\tau_n\| = h^{-(k+2)} K \max_{1 \leq i \leq k} \{ \|y_{nk+i} - y_{nk+i}^*\| / |d_i - \alpha_i| \} + O(h)$$

where $K = \max_{1 \leq i \leq k} |\alpha_i|$.

Proof. From the results of Theorem 5.1 (extended in a natural way to the system (1.1)), the corrector formula (7.1) satisfies

$$Y_n - Y(n) = O(h^{k+2}).$$

Hence, in the standard fashion, (7.1) and (7.2) yield

$$Y_{n+1} - Y(n + 1) - G(Y_n - Y(n)) = -h^{k+2}\tau_n + O(h^{k+3});$$

similarly, from (7.3) and (7.4),

$$Y_{n+1}^* - Y(n + 1) - G(Y_n - Y(n)) = -h^{k+2}\sigma_n + O(h^{k+3}).$$

Subtracting these two equations gives

$$Y_{n+1} - Y_{n+1}^* = h^{k+2}((d_1 - \alpha_1)y^{(k+2)}(x_{kn}), \dots, (d_k - \alpha_k)y^{(k+2)}(x_{kn}))^T + O(h^{k+3}),$$

from which with $\|\tau_n\| = K\|y^{(k+2)}\|$ the required result follows.

A considerable amount of practical experience with the cases $k = 2$ and 4 (for stiff and nonstiff systems) indicates that the above estimate for $h^{k+2}\|\tau_n\|$ is accurate and very reliable. For $k = 2$, the predictor formula (7.3), (7.4) is defined uniquely by

$$\begin{aligned} y(x_{2n+1}) - y(x_{2n}) &= \frac{h}{12} (23y'(x_{2n}) - 16y'(x_{2n-1}) + 5y'(x_{2n-2})) \\ &+ \frac{3}{8} h^4 y^{(4)}(x_{2n}) + O(h^5), \end{aligned} \tag{7.5}$$

$$\begin{aligned} y(x_{2n+2}) - y(x_{2n}) &= \frac{h}{3} (19y'(x_{2n}) - 20y'(x_{2n-1}) + 7y'(x_{2n-2})) \\ &+ \frac{8}{3} h^4 y^{(4)}(x_{2n}) + O(h^5). \end{aligned}$$

The corresponding estimate for the local truncation error reduces to

$$(7.6) \quad h^4 \|\tau_n\| \sim \frac{1}{8} \max\{\|y_{2n+1} - y_{2n+1}^*\|, \frac{1}{8} \|y_{2n+2} - y_{2n+2}^*\|\} = E_n.$$

Two numerical examples which illustrate the effectiveness of this estimate are described at the end of Section 8.

8. Practical Implementation of the Fourth Order Method. Numerical Examples.

For the satisfactory practical use of the AMM (3.3) for stiff systems, suitable requirements are as follows.

(i) An iteration scheme for obtaining Y_{n+1} , which (assuming the existence of a solution) is convergent for those values of h which are (essentially) restricted only by the requested accuracy in the numerical solution.

(ii) The control of accuracy based on estimates of the size of the local truncation error and corresponding step-size changing.

The requirement (i) implies the use of a Newton-type scheme for obtaining Y_{n+1} . The conventional halving and doubling of step size (performed blockwise) with the use of the estimate in Theorem 7.1 meet the needs of (ii).

For $k = 2$, the corrector formula follows Eq. (3.3) with corresponding predictor formula and local truncation error estimate given by (7.5) and (7.6). For the calculation of $(y_{2n+1}, y_{2n+2})^T$, the application of a NT (Newton-type) scheme requires an approximation to the matrix

$$J(n) = \left[\begin{array}{c|c} I - \frac{2}{3} h J_1 & \frac{h}{12} J_2 \\ \hline -\frac{4}{3} h J_1 & I - \frac{h}{3} J_2 \end{array} \right],$$

where J_r is the Jacobian matrix $((\partial f/\partial y)(x_{2n+r}, y_{2n+r}))$. For the numerical treatment of general stiff systems, the very simple scheme outlined below was used. $J(n)$ is approximated by

$$J(n) \approx \tilde{J}(n) = \left[\begin{array}{c|c} I - \frac{2}{3} h \tilde{J}_1 & \frac{h}{12} \tilde{J}_1 \\ \hline -\frac{4}{3} h \tilde{J}_1 & I - \frac{h}{3} \tilde{J}_1 \end{array} \right],$$

where \tilde{J}_1 represents an approximation to J_1 . \tilde{J}_1 is obtained by using first order forward difference approximations to $(\partial f^i / \partial y^j)(x_{2n+1}, y_{2n+1})$, $i, j = 1, 2, \dots, m$. The resulting $\tilde{J}(n)$ is held constant throughout the NT iteration. Only the essential details are presented on the understanding that the approximate Jacobian \tilde{J}_1 is reused wherever possible (or convenient).

Standard Scheme. Here the value of $Y_{n+1} = (y_{2n+1}, y_{2n+2})^T$, $n \geq 2$, with respect to the step size h is obtained. Let the previously computed block Y_n be defined with respect to the step size h_1 and let the LU factors of $\tilde{J}(n-1)$ be preserved.

Stage 1. If $h = h_1$ then set $\tilde{J}(n) = \tilde{J}(n-1)$ and compute the predictor Y_{n+1}^* ; also set test = 1. Otherwise, set test = -1 and $Y_{n+1}^* = (y_{2n}, y_{2n})^T$, then update \tilde{J}_1 and compute the LU factors of the resulting $\tilde{J}(n)$.

Stage 2. Allow ≤ 4 NT iterations and, if convergent, continue at Stage 3. Otherwise, update \tilde{J}_1 , then compute the LU factors of the resulting $\tilde{J}(n)$ and allow ≤ 3 NT iterations. If convergent, continue at Stage 3, otherwise, half h and return to Stage 1.

Stage 3. Y_{n+1} has now been obtained and is accepted if test = -1. Otherwise, perform appropriate tests for the size of the local truncation error using (7.6). If a step change is required, return to Stage 1; otherwise, the value of Y_{n+1} is accepted.

Starting Scheme. Here, by a suitable modification of the above standard scheme, we compute both Y_1 and Y_2 with respect to a step size h . The size of h is automatically chosen so that, with the resulting Y_2^* and Y_2 , the local truncation error test is satisfied. An initial value of h is first specified and the predictor $Y^* = (y_0, y_0)^T$ is used.

We note that for small enough h the above corrector iteration is convergent. The allowable number of iterations in Stage 2 are based on a large number of numerical tests. An alternative starting scheme is to use the generalized technique of step doubling for error estimation and thus test the accuracy of Y_1 directly.

From Stages 1 and 3 it is seen that the effect of a step change (halving or doubling) is to set test = -1; then Y_{n+1} is accepted without performing the local truncation error test. Therefore, there exists the possibility that the accepted Y_{n+1} would fail the local error test. As far as practical stiff problems are concerned, it is unlikely that the above scheme would yield values of Y_{n+1} which would fail the local error test over several consecutive steps. At the beginning of the range, Y_2 always satisfies the local error test.

Given a standard LU factorization routine, the above scheme is not difficult to program.

A large number of stiff problems have been successfully solved using the above scheme (including the ten test problems in [4]). We compared our results with those obtained by using Gear's variable order method [8] (up to order 6) with difference approximations for Jacobians. The Gear program used the double precision Fortran subroutine DIFSUB [9] and was run on a PDP 10 system whose double precision is equivalent to approximately sixteen decimal digits. The AMM was implemented in Algol and run on a CDC 7600 machine with a similar precision of approximately sixteen decimal digits.

In the AMM program, the following local error test was used:

$$E_n \leq \epsilon \max\{1, \|Y_{n+1}\|_\infty\}.$$

Here, E_n is the estimate (7.6) of the local truncation error with respect to Y_{n+1} and ϵ is the requested error tolerance. In DIFSUB, a very similar local error test was used by updating the YMAX array so that

$$YMAX(i) = \max(1, |y^i|), \text{ for } i = 1, 2, \dots, m.$$

We first summarize the results and then describe three illustrative problems.

1. For moderate accuracy, that is with ϵ ranging between 10^{-3} and 10^{-6} :

(a) the two methods were comparable in terms of the number of derivative evaluations,

(b) the number of LU factorizations (of matrices order $2m$) in the AMM was approximately equal to the number of factorizations (of matrices order m) in the Gear program.

TABLE I

Requested error	maximum absolute error		ND/LU		Range
	AMM	GEAR	AMM	GEAR	
10^{-3}	$1.4 \cdot 10^{-4}$	$9.8 \cdot 10^{-4}$	104/6	94/5	(0, .01)
	$1.4 \cdot 10^{-4}$	$9.8 \cdot 10^{-4}$	165/9	133/8	(0, .1)
	$3.4 \cdot 10^{-3}$	$2.3 \cdot 10^{-3}$	314/17	213/11	(0, 1)
	$3.4 \cdot 10^{-3}$	$2.3 \cdot 10^{-3}$	387/21	266/15	(0, 10)
	$3.4 \cdot 10^{-3}$	$2.3 \cdot 10^{-3}$	500/27	366/24	(0, 1000)
10^{-4}	$1.3 \cdot 10^{-5}$	$3.5 \cdot 10^{-5}$	121/5	111/6	(0, .01)
	$1.3 \cdot 10^{-5}$	$7.5 \cdot 10^{-5}$	192/9	195/10	(0, .1)
	$2.3 \cdot 10^{-4}$	$8.0 \cdot 10^{-5}$	335/14	283/13	(0, 1)
	$2.3 \cdot 10^{-4}$	$9.0 \cdot 10^{-5}$	418/18	367/16	(0, 10)
	$2.3 \cdot 10^{-4}$	$1.1 \cdot 10^{-4}$	545/24	477/24	(0, 1000)
10^{-5}	$1.9 \cdot 10^{-6}$	$2.2 \cdot 10^{-7}$	165/5	145/6	(0, .01)
	$2.2 \cdot 10^{-6}$	$1.4 \cdot 10^{-5}$	252/9	291/12	(0, .1)
	$1.6 \cdot 10^{-5}$	$1.4 \cdot 10^{-5}$	416/11	437/17	(0, 1)
	$1.6 \cdot 10^{-5}$	$2.1 \cdot 10^{-5}$	532/15	537/21	(0, 10)
	$1.6 \cdot 10^{-5}$	$2.1 \cdot 10^{-5}$	702/22	677/28	(0, 1000)
10^{-6}	$1.9 \cdot 10^{-7}$	$9.1 \cdot 10^{-8}$	270/6	207/7	(0, .01)
	$1.9 \cdot 10^{-7}$	$2.7 \cdot 10^{-6}$	379/10	310/12	(0, .1)
	$1.7 \cdot 10^{-6}$	$2.7 \cdot 10^{-6}$	660/14	465/15	(0, 1)
	$1.7 \cdot 10^{-6}$	$2.9 \cdot 10^{-6}$	836/19	603/20	(0, 10)
	$1.7 \cdot 10^{-6}$	$2.9 \cdot 10^{-6}$	1062/26	809/29	(0, 1000)

- 2. For higher accuracy, up to $\epsilon = 10^{-8}$:
 - (a) the AMM required up to twice as many derivative evaluations,
 - (b) the behaviour in 1(b) applied.

3. For some problems with eigenvalues close to the imaginary axis, Gear's method can be prohibitively inefficient as compared to the A -stable AMM.

Generally, for moderate accuracy, Gear's method tends to favour the use of the lower order methods, thus leading to the behaviour in 1 [14, p. 286]. On many machines with moderately sized m , the difference in the linear algebra would not be significant, but, in this respect, the AMM cannot be expected to be competitive with Gear's method for large systems.

The following three problems were solved over ranges up to $[0, 1000]$ and results are given at the first step to pass 10^i for $i = -2, -1, 0, 1$ and 3. Tables 1-3 include the requested error ϵ , the maximum absolute/relative error to date in the least ac-

TABLE 2

Requested error	Maximum relative error		ND/LU		Range
	AMM	GEAR	AMM	GEAR	
10^{-3}	$5.8 \cdot 10^{-5}$	$4.1 \cdot 10^{-4}$	99/6	87/5	(0, .01)
	$5.8 \cdot 10^{-5}$	$4.1 \cdot 10^{-4}$	162/9	162/11	(0, .1)
	$3.9 \cdot 10^{-4}$	$5.4 \cdot 10^{-3}$	407/19	406/16	(0, 1)
	$3.9 \cdot 10^{-4}$	$5.4 \cdot 10^{-3}$	474/23	480/20	(0, 10)
	$3.9 \cdot 10^{-4}$	$5.4 \cdot 10^{-3}$	594/29	635/35	(0, 1000)
10^{-4}	$9.5 \cdot 10^{-6}$	$4.4 \cdot 10^{-5}$	111/5	116/5	(0, .01)
	$9.5 \cdot 10^{-6}$	$4.4 \cdot 10^{-5}$	188/9	167/8	(0, .1)
	$4.0 \cdot 10^{-5}$	$4.3 \cdot 10^{-4}$	473/16	454/14	(0, 1)
	$4.0 \cdot 10^{-5}$	$4.3 \cdot 10^{-4}$	563/21	803/24	(0, 10)
	$4.0 \cdot 10^{-5}$	$4.3 \cdot 10^{-4}$	752/29	943/32	(0, 1000)
10^{-5}	$1.5 \cdot 10^{-6}$	$4.5 \cdot 10^{-6}$	141/5	148/7	(0, .01)
	$1.5 \cdot 10^{-6}$	$4.5 \cdot 10^{-6}$	213/8	273/14	(0, .1)
	$2.7 \cdot 10^{-6}$	$3.2 \cdot 10^{-5}$	611/12	613/12	(0, 1)
	$2.7 \cdot 10^{-6}$	$3.3 \cdot 10^{-5}$	733/17	812/30	(0, 10)
	$2.7 \cdot 10^{-6}$	$3.3 \cdot 10^{-5}$	880/24	1002/38	(0, 1000)
10^{-6}	$1.9 \cdot 10^{-7}$	$6.4 \cdot 10^{-7}$	232/6	202/9	(0, .01)
	$1.9 \cdot 10^{-7}$	$9.3 \cdot 10^{-7}$	356/11	307/15	(0, .1)
	$2.8 \cdot 10^{-7}$	$5.0 \cdot 10^{-6}$	998/15	763/22	(0, 1)
	$2.8 \cdot 10^{-7}$	$5.0 \cdot 10^{-6}$	1176/21	1026/32	(0, 10)
	$2.8 \cdot 10^{-7}$	$5.0 \cdot 10^{-6}$	1370/28	1261/40	(0, 1000)

TABLE 3

β_2	Maximum absolute error		ND/LU		Range
	AMM	GEAR	AMM	GEAR	
1	$2.2 \cdot 10^{-8}$	$1.4 \cdot 10^{-7}$	400/8	224/6	(0, .01)
	$2.2 \cdot 10^{-8}$	$1.4 \cdot 10^{-7}$	508/13	341/13	(0, .1)
	$2.2 \cdot 10^{-8}$	$1.4 \cdot 10^{-7}$	638/14	421/16	(0, 1)
	$2.2 \cdot 10^{-8}$	$1.4 \cdot 10^{-7}$	968/17	581/23	(0, 10)
	$2.2 \cdot 10^{-8}$	$1.4 \cdot 10^{-7}$	1276/21	728/28	(0, 100)
10	As for $\beta_2 = 1$	$1.4 \cdot 10^{-7}$	As for $\beta_2 = 1$	224/6	(0, .01)
		$1.4 \cdot 10^{-7}$		341/13	(0, .1)
		$1.4 \cdot 10^{-7}$		421/16	(0, 1)
		$1.4 \cdot 10^{-7}$		581/23	(0, 10)
		$4.4 \cdot 10^{-7}$		3111/34	(0, 100)
100	As for $\beta_2 = 1$	$1.4 \cdot 10^{-7}$	As for $\beta_2 = 1$	224/6	(0, .01)
		$1.4 \cdot 10^{-7}$		341/13	(0, .1)
		$1.4 \cdot 10^{-7}$		421/16	(0, 1)
		$7.4 \cdot 10^{-7}$		3803/32	(0, 10)
		$7.4 \cdot 10^{-7}$		5001/32	(0, 12)
		Abandoned		Abandoned	

curate component, the number of derivative evaluations (ND) and the number of LU splits (or matrix inversions in Gear's method). All the problems were run with an initial step of size equal to 2^{-13} . For Problem 1, however, with $\epsilon = 10^{-4}$, 10^{-5} and 10^{-6} , we give Gear's published results [9] for this problem (these results are in very close agreement with those obtained from our implementation of Gear's method).

Each problem is critically stable in the sense that, if an error in excess of about 10^{-3} occurs, the solution of the perturbed problem may have an unbounded solution.

Krogh Problem 1 [13].

$$y' = -By + U^T(z_1^2, z_2^2, z_3^2, z_4^2)^T, \quad y(0) = (-1, -1, -1, -1)^T,$$

where $(z_1, z_2, z_3, z_4)^T = z = Uy$, $B = U \text{diag}(\beta_1, \beta_2, \beta_3, \beta_4)U$, U is the unitary matrix with diagonal elements equal to $-\frac{1}{2}$ and all other elements equal to $\frac{1}{2}$, and $\beta_1 = 1000$, $\beta_2 = 800$, $\beta_3 = -10$ and $\beta_4 = 0.001$. The eigenvalues of the Jacobian matrix are equal to $-(\beta_i + 2)$ at $x = 0$ and approach $-|\beta_i|$ as $x \rightarrow \infty$. The solution is

$$y = Uz \quad \text{with } z_i = \beta_i / (1 - (1 + \beta_i)e^{\beta_i x}), \quad i = 1, 2, 3, 4.$$

The results are given in Table 1.

TABLE 4. *Problem 1*

Requested error ϵ	x	Estimate E_n	Exact T_n
10^{-4}	0.125	$1.50 \cdot 10^{-5}$	$1.58 \cdot 10^{-5}$
	0.5625	$4.28 \cdot 10^{-5}$	$3.20 \cdot 10^{-5}$
	2	$6.36 \cdot 10^{-7}$	$4.26 \cdot 10^{-7}$
	4	$1.60 \cdot 10^{-5}$	$7.00 \cdot 10^{-6}$
	6	$2.08 \cdot 10^{-5}$	$1.39 \cdot 10^{-5}$
	10	$3.34 \cdot 10^{-6}$	$2.15 \cdot 10^{-6}$
10^{-6}	0.1094	$8.83 \cdot 10^{-7}$	$9.43 \cdot 10^{-7}$
	0.6094	$1.18 \cdot 10^{-7}$	$1.07 \cdot 10^{-7}$
	2	$3.53 \cdot 10^{-8}$	$2.91 \cdot 10^{-8}$
	4	$4.28 \cdot 10^{-8}$	$3.51 \cdot 10^{-8}$
	6	$1.33 \cdot 10^{-7}$	$1.00 \cdot 10^{-7}$
	10	$2.31 \cdot 10^{-7}$	$1.61 \cdot 10^{-7}$
10^{-8}	0.1016	$3.31 \cdot 10^{-9}$	$3.39 \cdot 10^{-9}$
	0.5078	$8.62 \cdot 10^{-9}$	$8.81 \cdot 10^{-9}$
	2.0625	$1.88 \cdot 10^{-9}$	$1.71 \cdot 10^{-9}$
	4	$2.57 \cdot 10^{-9}$	$2.32 \cdot 10^{-9}$
	5.5	$6.84 \cdot 10^{-10}$	$6.32 \cdot 10^{-10}$
	10	$7.95 \cdot 10^{-10}$	$7.24 \cdot 10^{-10}$

Krogh Problem 2 [13].

$$y' = -By + U^T(\frac{1}{2}z_1^2 - \frac{1}{2}z_2^2, z_1z_2, z_3^2, z_4^2)^T, \quad y(0) = (0, -2, -1, -1)^T,$$

where $z = Uy$, U is defined in Problem 1,

$$B = U \begin{bmatrix} \beta_1 & -\beta_2 & 0 & 0 \\ \beta_2 & \beta_1 & 0 & 0 \\ 0 & 0 & \beta_3 & 0 \\ 0 & 0 & 0 & \beta_4 \end{bmatrix} U,$$

and

$$\beta_1 = -10, \quad \beta_2 = 10, \quad \beta_3 = 1000, \quad \beta_4 = 0.001.$$

The eigenvalues are equal to $-(\beta_1 + 2) \pm i\beta_2$, $-(\beta_3 + 2)$ and $-(\beta_4 + 2)$ at $x = 0$, and approach $-|\beta_1| \pm i\beta_2$, $-|\beta_3|$ and $-|\beta_4|$ as $x \rightarrow \infty$. The solution is $y = Uz$ with

$$z_1 = 2(\beta_1 w_1 - \beta_2 w_2)/(w_1^2 + w_2^2),$$

$$z_2 = 2(\beta_2 w_1 + \beta_1 w_2)/(w_1^2 + w_2^2),$$

$$z_i = \beta_i/(1 - (1 + \beta_i)e^{\beta_i x}), \quad i = 3, 4,$$

where

$$w_1 = 1 - e^{\beta_1 x}[(1 + \beta_1) \cos \beta_2 x - \beta_2 \sin \beta_2 x]$$

and

$$w_2 = e^{\beta_1 x}[\beta_2 \cos \beta_2 x + (1 + \beta_1) \sin \beta_2 x].$$

The results are given in Table 2.

Problem 3. This is a modified form of Problem 2.

$$y' = -By + U(0, 0, z_3^2, z_4^2)^T, \quad y(0) = (-1, -1, 0, 0)^T,$$

where $z = Uy$, U is defined in Problem 1, and B is defined in Problem 2 with $\beta_1 = 1$, $\beta_3 = 1000$, and $\beta_4 = 0.001$. Two of the eigenvalues are equal to $-\beta_1 \pm i\beta_2$ for all x ;

TABLE 5. *Problem 2*

Requested error ϵ	x	Estimate E_n	Exact T_n
10^{-4}	0.125	$1.77 \cdot 10^{-5}$	$2.59 \cdot 10^{-5}$
	0.5156	$1.72 \cdot 10^{-5}$	$1.19 \cdot 10^{-5}$
	1.125	$5.93 \cdot 10^{-6}$	$4.04 \cdot 10^{-6}$
	3	$2.34 \cdot 10^{-6}$	$1.47 \cdot 10^{-6}$
	6	$2.11 \cdot 10^{-6}$	$1.39 \cdot 10^{-6}$
	10	$3.36 \cdot 10^{-6}$	$2.15 \cdot 10^{-6}$
10^{-6}	0.1016	$4.85 \cdot 10^{-7}$	$5.31 \cdot 10^{-7}$
	0.5039	$8.26 \cdot 10^{-8}$	$7.49 \cdot 10^{-8}$
	1.0156	$6.96 \cdot 10^{-8}$	$6.11 \cdot 10^{-8}$
	3	$1.35 \cdot 10^{-7}$	$1.05 \cdot 10^{-7}$
	6	$1.33 \cdot 10^{-7}$	$1.00 \cdot 10^{-7}$
	10	$2.31 \cdot 10^{-7}$	$1.61 \cdot 10^{-7}$
10^{-8}	0.1016	$1.94 \cdot 10^{-9}$	$1.98 \cdot 10^{-9}$
	0.5020	$5.34 \cdot 10^{-9}$	$5.08 \cdot 10^{-9}$
	1.0078	$4.49 \cdot 10^{-9}$	$4.23 \cdot 10^{-9}$
	3	$7.97 \cdot 10^{-9}$	$6.97 \cdot 10^{-9}$
	5.75	$5.66 \cdot 10^{-10}$	$5.24 \cdot 10^{-10}$
	10	$7.95 \cdot 10^{-10}$	$7.25 \cdot 10^{-10}$

the remainder are $-(\beta_i + 2)$ when starting and approach $-|\beta_i|$ as $x \rightarrow \infty$, for $i = 3$ and 4. We consider the effect of eigenvalues close to the imaginary axis by running this problem with $\beta_2 = 1, 10$ and 100, and with a requested error $\epsilon = 10^{-7}$. In each case, the solution is $y = Uz$ with $z_1 = z_2 = 0$ and $z_i = \beta_i / (1 - (1 + \beta_i)e^{\beta_i x})$, $i = 3, 4$. The results are given in Table 3 and indicate clearly how the stability regions for Gear's method can restrict the increase in step size.

Finally, we include two numerical examples which illustrate the effectiveness of the estimate for the local truncation error (7.6). Test Problems 1 and 2 were solved using the AMM program with the local error test $E_n \leq \epsilon$, for the values $\epsilon = 10^{-4}$, 10^{-6} and 10^{-8} . The results are given in Tables 4 and 5. They include the estimate E_n and the corresponding exact value of the normed local truncation error

$$T_n = \|Y(n+1) - GY(n) - h(QF(n+1) + RF(n))\|_{\infty}.$$

The x values are chosen from those which are automatically selected by the program.

It is of interest to note that for these two problems the estimate E_n has the desirable property of tending to overestimate the size of T_n . For the respective problems, the proportion of overestimates was approximately 90% and 70%.

9. Conclusions. A class of A -stable advanced multistep methods for the numerical solution of initial value problems has been described. The fourth order method for the solution of stiff equations described in Section 8 is easy to implement, and the numerical results indicate that in terms of satisfying the user's error tolerance, it is extremely reliable. For systems which are not large and for moderate accuracy, the method compares favourably with Gear's method. Also, for some problems with eigenvalues close to the imaginary axis, the AMM can be considerably more efficient than Gear's method. This may be of importance in some problem areas, for example, problems arising in circuit analysis.

10. Acknowledgements. Frank de Hoog wishes to acknowledge a Commonwealth Postgraduate Research Award from the Australian Department of Education and Science.

The authors wish to thank the referee for his valuable suggestions concerning the structure of this paper. Also we wish to thank Mr. D. Sayers and Mr. R. Coleman for their programming assistance.

Department of Mathematics
The University
Manchester M13 9PL, England

Computer Centre
Australian National University
Canberra 2600, Australia

1. O. AXELSSON, "A class of A -stable methods." *Nordisk Tidskr. Informationsbehandling (BIT)*, v. 9, 1969, pp. 185-199. MR 40 #8266.

2. F. H. CHIPMAN, " A -stable Runge-Kutta processes," *Nordisk Tidskr. Informationsbehandling (BIT)*, v. 11, 1971, pp. 384-388.

3. G. DAHLQUIST, "A special stability problem for linear multistep methods," *Nordisk Tidskr. Informationsbehandling*, v. 3, 1963, pp. 27-43. MR 30 #715.

4. G. DAHLQUIST et al., *Survey of Stiff Ordinary Differential Equations*, The Royal Institute of Technology, Stockholm, Report NA 70.11, 1970.
5. J. W. DANIEL, *Non-Linear Equations Arising in Deferred Correction of Initial Value Problems*, MRC Technical Report No. 818, 1967; Also in *Acta Ci. Venezolana*, v. 19, 1968, pp. 123–128. (Spanish) MR 40 #8269.
6. J. W. DANIEL, V. PEREYRA & L. L. SCHUMAKER, *Iterated Deferred Corrections for Initial Value Problems*, MRC Technical Report #808, Madison, Wis., 1967; Also in *Acta Ci. Venezolana*, v. 19, 1968, pp. 128–135. (Spanish) MR 40 #8270.
7. B. L. EHLE, "High order A -stable methods for the numerical solution of systems of D.E.'s," *Nordisk Tidskr. Informationsbehandling (BIT)*, v. 8, 1968, pp. 276–278. MR 39 #1119.
8. C. W. GEAR, *The Automatic Integration of Stiff Ordinary Differential Equations*, Proc. IFIP Congress, Supplement, Booklet A: 81–85, 1968.
9. C. W. GEAR, "DIFSUB for solution of ordinary differential equations," *Comm. ACM*, v. 14, 1971, pp. 185–190.
10. C. W. GEAR, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, N. J., 1971.
11. P. HENRICI, *Discrete Variable Methods in Ordinary Differential Equations*, Wiley, New York, 1962. MR 24 #B1772.
12. B. L. HULME, "Discrete Galerkin and related one-step methods for ordinary differential equations," *Math. Comp.*, v. 26, 1972, pp. 881–891.
13. F. T. KROGH, *On Testing a Subroutine for the Numerical Integration of Ordinary Differential Equations*, Jet Propulsion Laboratory, Pasadena, Calif., Tech. Mem. No. 217, 1970.
14. L. LAPIDUS & J. H. SEINFELD, *Numerical Solution of Ordinary Differential Equations*, Math. in Sci. and Engineering, vol. 74, Academic Press, New York and London, 1971. MR 43 #7073.
15. A. RALSTON, *A First Course in Numerical Analysis*, McGraw-Hill, New York, 1965. MR 32 #8479.
16. J. B. ROSSER, "A Runge-Kutta for all seasons," *SIAM Rev.*, v. 9, 1967, pp. 417–452. MR 36 #2325.
17. L. F. SHAMPINE & H. A. WATTS, "Block implicit one-step methods," *Math. Comp.*, v. 23, 1969, pp. 731–740. MR 41 #9445.
18. H. A. WATTS & L. F. SHAMPINE, " A -stable block implicit one-step methods," *Nordisk Tidskr. Informationsbehandling (BIT)*, v. 12, 1972, pp. 252–266.
19. H. A. WATTS, *A-Stable Block Implicit One-Step Methods*, Sandia Laboratories Report SC-RR-71 0296.
20. K. WRIGHT, "Some relationships between implicit Runge-Kutta, collocation and Lanczos τ methods, and their stability properties," *Nordisk Tidskr. Informationsbehandling (BIT)*, v. 10, 1970, pp. 217–227. MR 42 #1345.