NORTH-HOLLAND

Varying the Componentwise Order of the Multistep Methods in Solving ODEs and Its Absolute Stability

Mohamed Bin Suleiman and Saiman Baok

Department of Mathematics
Universiti Pertanian Malaysia
43400 UPM SERDANG, Selangor, Malaysia

and

George Hall

Department of Mathematics
University of Manchester
Manchester, England

Transmitted by John Casti

ABSTRACT

Most variable order, variable stepsize codes for solving ODEs vary the order along the interval of integration but keep the order constant across the system. However, some codes that do partitioning of the ODEs into stiff and nonstiff subsystems vary the order for each equation in the system. In this paper, the reasons for varying the order componentwise are given and some of its salient features are illustrated using a numerical example. Also given is the direct proof of the existence of a region of absolute stability when different order Adams explicit methods are used to solve a system of ODEs.

APPLIED MATHEMATICS AND COMPUTATION 74:209–222 (1996)

655 Avenue of the Americas, New York, NY 10010
0096-3003/96/$15.00
SSDI 0096-3003(95)00093-W

1. INTRODUCTION

Variable order variable stepsize codes, see for instance [1–3], vary the order along the subintervals of integration but keep the order constant for the system as a whole. However, in [4, 5] the codes that partition a system of ODEs into stiff and nonstiff subsystems and solve them with the Adams and BDF methods, respectively, permit the order to vary separately for each equation in the system. This additional complexity requires justification in terms of efficiency and raises the question of whether such codes have a nonempty absolute stability region. Gear [6] expressed his apprehension over solving a system of ODEs using the multistep and varying the order componentwise, because it may lead to the absence of an absolute stability region. However, Söderlind [7], proved that for multistep methods that are stable, varying the order preserves their stability.

In this paper, we discuss the reasons for varying the order for each equation and illustrate, by way of a numerical example, some of its salient features. Next, a direct proof of the existence of the absolute stability region is given for the Adams explicit methods when the order is varied. For the implicit cause it may similarly be proved.

2. WHY USE DIFFERENT ORDER ADAMS METHODS?

For ease and uniformity of notation, the following are defined as k-step methods:

i) The explicit k-step Adams method.

ii) The PECE Adams method using a k-step explicit predictor and a k-step implicit corrector.

iii) The $(k-1)$-step BDF method implemented using a k-step explicit Adams formula as predictor and correcting twice the BDF method using a Newton-type correction.

The methods have local truncation errors (l.t.e) of different order in h. For (i) the l.t.e. is of $O(h^{k+1})$, while (ii) is of $O(h^{k+2})$ and (iii) is of $O(h^k)$. The methods given in (ii) and (iii) are used in [4] and [5] to solve nonstiff and stiff ODEs, respectively, which we shall be using to solve our numerical example.

Now consider the first order system of ODEs

$$y' = f(x, y), \qquad y(a) = \eta, \qquad y \in \mathbb{R}^s. \tag{2.1}$$

Codes for solving (2.1) may have different strategies for changing the order. We mention here the order-changing strategies of some nonstiff ODE codes, which are also applicable to stiff codes. Gear [1] has the following

order strategy. If $e_k = [e_{1k}, e_{2k}, \ldots, e_{sk}]^T$ is the local error of solving (2.1) with the k-step predictor-corrector method for the interval $[x_{n-1}, x_n]$ with stepsize h_n, then the next stepsize h_{n+1} for the interval $[x_n, x_{n+1}]$ using order r is given by

$$h_{n+1} = R_r^{1/(r+1)} h_n \quad \text{where } R_r = \frac{TOL}{\|e_r\|}, \quad r = k-1, k, k+1. \quad (2.2)$$

The choice of the order r for the said interval is such that h_{n+1} is the maximum. Krogh's choice in [6] is based on the ratio $10/R_k$ with other additional rules related to the terms in the local error function.

Shampine and Gordon [3] use a simple but effective order choice strategy. Except for the initial phase, they use the following rules. The order is raised from k to $k+1$ only after $k+1$ successful steps at constant stepsize, provided that for $1 < k \leqslant 12$, $\|e_{k+1}\| < \|e_k\| < \max(\|e_{k-1}\|, \|e_{k-2}\|)$, and for $k = 1$, $\|e_{k+1}\| < \|0.5e_k\|$, the order is lowered by one if for $k > 2$, $\max(\|e_{k-1}\|, \|e_{k-2}\|) \leqslant \|e_k\|$ and for $k = 2$, $\|e_{k-1}\| \leqslant \max(\|e_k\|, \|e_{k+1}\|)$.

Once the choice of k is made, then h_{n+1} is given as in (2.2). The componentwise version of this order strategy is used in [4] and [5], which are codes for both stiff and nonstiff ODEs. The advantage of a constant order for the system is that one has to calculate the ratio R_k for each step only once. When the order is varied componentwise and k_i is the order of the Adams method in the ith equation for system (2.1), then one has to calculate

$$R_{k_i} = \left(\frac{TOL}{\|e_{k_i}\|} \right)^{1/(k_i+1)},$$

for all $i = 1, \ldots, s$. The next step size then is given by $h_{n+1} = R^* h_n$ where $R^* = \min_i R_{k_i}$.

Why then vary the order componentwise? In the order strategy of Shampine and Gordon (henceforth in the discussion, this is the order-changing strategy that will be referred to for the constant-order case in the system and will be simply called the constant order k case), the order k for this system is determined by the component contributing to $\|e_k\|$. This value of k may be too conservative for some of the components, and can result in some loss of accuracy to the computed solutions of these components. It could also be too high for some others, which implies extra overhead due to more terms for the predictor and divided difference. These points will be further illustrated with the numerical example given later.

Another reason is the consequence of the remark of Shampine and Gordon that their code chooses lower order $k \leqslant 4$ following frequent step

failures whenever instability due to stiffness occurs in solving stiff equations. This is because the lower order Adams PECE methods have better stability. It seems natural that if different order methods are used componentwise, then those that are unaffected by instability can choose higher order methods. This would then be very useful in partitioning ODEs into nonstiff and stiff subsystems, especially when it has been notedthat in many practical problems the stiff subsystem is smaller compared with the nonstiff one. It implies that solving the nonstiff equations with Adams methods and the stiff ones with BDF would then reduce the number of equations in the Jacobian evaluation of the stiff subsystem and the subsequent solution of the linear equations.

What is more pleasing is that an equation is changed from nonstiff to stiff at the most appropriate moment, at the point where stability restricts the Adams method. To be successful it is necessary to identify the equations that are the cause of stiffness. They may or may not be the equations that have frequent step failures but certainly are among those with low order methods. Two variations of identifying stiff equations were given by the author in [5] and Hall and the author in [4].

The codes in [4] and [5] are adequate to solve most stiff systems, despite the fact that the partitioning does not take into account changes in the state of the equations from stiff to nonstiff. This is because most equations considered stiff do not revert to the nonstiff state and also the fact that nonstiff equations can still be solved by BDF methods, although they are computationally expensive. However, there are first order systems, for instance the Van Der Pols equations, see [8], where the state changes frequently from nonstiff to stiff and vice-versa. A code that is sensitive to these changes of states is called a type-insensitive code. Currently type-insensitive codes change the methods of solution to the system as a whole (see [9–12]). In other words no componentwise partitioning from nonstiff to stiff and vice-versa is done. In such a case the code should be able to handle changes from stiff to nonstiff. An early indicator of the need for such a change is to look at the behavior of variable order BDF methods when the transients are reintroduced. In the presence of transients, a reverse effect of the Adams methods in the presence of instabilities occurs. Here the BDF methods choose the highest permissible order, viz. the sixth order method with step failures and nonconvergences. These failures occur because the iteration matrix $I - h\beta J$, has not been refactored when h is reduced and when changes have occurred in the Jacobian matrix J during the change in the state of the system from stiff to nonstiff. The other equations in the stiff subsystem not affected by the transients choose low-order BDF methods. At this point the stiff equations that choose the sixth order BDF methods with frequent step failures are tested for nonstiffness. Hence this discussion

suggests that varying the order componentwise while using the multistep methods can be helpful in the development of type-insensitive codes, because the order variations of the multistep methods for each equation in the presence of instabilities or transients with step failures is a first indication of the need to switch methods.

We shall illustrate some of our points with a numerical example and in the process show some of the salient features of varying the order in a multistep method. The example used is the system

$$y_1' = -10\,y_1 + 3\,y_2$$

$$y_2' = -3\,y_1 - 10\,y_2$$

$$y_3' = -4\,y_3$$

$$y_4' = -y_4$$

$$y_5' = -0.5\,y_5$$

$$y_6' = -0.1\,y_6$$

$$y_1(0) = y_2(0) = y_3(0) = y_4(0) = y_5(0) = y_6(0) = 1,$$

$$0 \leqslant x \leqslant 20,$$

eigenvalues: $-0.1, -0.5, -1, -4, -10 \pm i3$, (given in Enright et al. [8]).

We proceed to solve the above system by using the Adams PECE methods varying the order componentwise and then switching to a BDF method for any equation found to be stiff. The order-changing strategy is that of Shampine and Gordon applied componentwise. Because the higher order explicit BDF methods are not zero stable, we restrict the order of the PECE BDF methods (as defined earlier) to six, i.e., the 6-step explicit BDF predictor and the 5-step BDF implicit corrector. This is despite the 6-step implicit BDF method being stiffly stable. Table 1 below gives the pair (k_i, e_i), $i = 1, 2, \ldots, 6$, which are the order and the local error for the ith equation in the system at seven tabulated points denoted by (x, h) where x is the said point and h the step size used. The tolerance used for this example is 10^{-6}. A number of the form $a(-b)$ in the table will mean $a \times 10^{-b}$.

The first tabulated point is the first step, therefore the order is 1 for all equations. The second tabulated point is just a few steps after the initial phase of the code, during which the order is raised at every step, hence the

TABLE 1

THE ORDER AND LOCAL ERROR FOR EACH EQUATION ALONG THE POINT OF INTEGRATION

Points	(x, h)	(k_i, e_i), for i equals					
		1	2	3	4	5	6
1	(6.9(−5), 6.9(−5))	(1, 7.5(−8))	(1, 3.6(−7))	(1, 3.8(−8))	(1, 2.4(−9))	(1, 6.0(−10))	(1, 2.4(11))
2	(1.1(−1), 8.9(−3))	(5, 4.9(−9))	(5, 5.6(−9))	(5, 2.7(−11))	(5, 8.4(−15))	(5, 1.4(−16))	(5, 9.1(−21))
3	(7.8(−1), 3.6(−2))	(8, 1.9(−8))	(8, 1.8(−8))	(10, 4.2(−12))	(10, 3.0(−18))	(9, 3.2(−20))	(7, 6.1(−22))
4	(2.21, 1.4(−1))	(3, 6.6(−7))	(4, 1.1(−7))	(10, 1.6(−9))	(11, 5.4(−15))	(11, 3.9(−17))	(8, 2.0(−19))
5	(5.68, 1.4(−1))	(4, 5.8(−8))	(4, 2.9(−7))	(4, 4.4(−13))	(11, 1.1(−14))	(12, 4.9(−17))	(9, 2.4(−20))
6	(18.60, 5.7(−1))	(3, 6.9(−11))	(3, 2.8(−10))	(7, 3.5(−6))	(6, 1.1(−8))	(9, 7.4(−12))	(11, 1.0(−17))
7	(19.60, 1.4(−1))	(2, 3.2(−13))	(2, 2.9(−12))	(6, 1.2(−7))	(4, 7.2(−13))	(8, 1.9(−15))	(11, 3.7(−22))

constant order for all equations. The third point shows the early variation of the order for all the equations where the transients are still dominant. The fourth tabulated point shows that the transients in the first two equations have almost died out and the step size is approaching instability and hence the first two equations choose lower order methods, viz., 3 and 4. However, the transients on the rest of the equation are still strong and therefore the high order. The $\|e_k\|$ here is due to the first equation, hence in a constant order k code the order for the system is 3. This will definitely affect the accuracy of the third equation onward.

Just prior to the fifth tabulated point, $x = 5.68$, stiffness tests as given in [11] are done on the first two equations. The rest of the equations are omitted from the tests because the high order indicates they are still in the nonstiff state. At $x = 5.68$ the first two equations are treated as stiff and solved using the BDF methods, while the rest still use the Adams methods. The sixth point at $x = 18.60$ shows that now the third equation has the largest magnitude for the local error and determines the order in a constant order system, which is 7. This is not possible with the BDF methods used. Notice for the sixth and seventh points the orders of the BDF methods for the first two equations are 2 and 3. Had the whole system solved at these points with a constant order BDF method, then the order attained, determined by the third equation, would be 6 (because the Adams method with smaller local error chooses order at least 6). This would unnecessarily push the order of the first two equations too high, implying higher overhead.

Let us now consider the effect of the reintroduction of the transients on equations that are already treated as stiff. For the sake of argument, let us assume the above problem is treated with the BDF methods at $x = 5.68$ onward. Then for the first two equations the orders would be less or equal to four, but the rest of the equations would choose the BDF method of order six, indicating the transients are still strong in these equations. A constant order method would choose order $\leqslant 4$ for all equations, since the most dominant local error belongs to the first equation. This clearly nullifies the effect of the transients. It is not until $x > 13$, that the local error of the third equation dominates and the order picks up. Variation of order componentwise identifies the transients from the onstart at $x = 5.68$.

The above reasons, especially the last two, justify the need to vary the order for each equation in codes that do partitioning or those that are componentwise type-insensitive.

3. THE ADAMS EXPLICIT CASE

Before proving the existence of absolute stability region when the order of the Adams explicit method is varied componentwise, we give the following lemma.

LEMMA. *Let* ${}^{r}A = ({}^{r}a_1, {}^{r}a_2, \ldots, {}^{r}a_s)$ *an* $s \times s$ *matrix where* ${}^{r}a_i$ *is an* R^s *column vector. Then*

$$\det\left(\sum_{r=1}^{m} {}^{r}A\right) = \sum_{n^s} \det({}^{r_1}a_1, {}^{r_2}a_2, \ldots, {}^{r_s}a_s)$$

where each $r_j \in \{1, \ldots, m\}$, $j = 1, \ldots, s$, *that is the sum of all possible determinants of the matrices formed by having the* j*th column of elements* ${}^{r}a_j$ *of* ${}^{r}A$, $r = 1, \ldots, m$ *which is a permutation of* n^s.

PROOF. Let $G = \sum_{r=1}^{m} {}^{r}A$

$$\det(G) = \det\left(\sum_{r=1}^{m} {}^{r}a_1, \sum_{r=1}^{m} {}^{r}a_2, \ldots, \sum_{r=1}^{m} {}^{r}a_m\right).$$

Then from the property of determinant, we have

$$\det(G) = \sum_{i=1}^{n^s} \det(P_i)$$

where $P_i = ({}^{r_1}a_1, {}^{r_2}a_2, \ldots, {}^{r_s}a_s)$ and ${}^{r_j}a_j$, $r_j \in \{1, \ldots, m\}$, $j = 1, \ldots, s$ any vector from the jth column of ${}^{r}A$ and has m possibilities from the m matrices. Because P_i had s elements of the vectors, therefore there are n^s different matrices P_i. Hence the result.

The general multistep method of solving (2.1) is given by

$$\rho(E)\, x_n = h\sigma(E)\, f_n \tag{3.1}$$

where E is such that $Ex_n = x_{n+1}$, the forward operator and

$$\rho(t) = \sum_{i=0}^{k} \alpha_i t^{k-1} \quad \text{and} \quad \sigma(t) = \sum_{i=0}^{k} \beta_i t^{k-1}.$$

An important characterization of (3.1) is the existence of the region of absolute stability. If λ_i is the eigenvalue of $\partial f/\partial y$, then the stability polynomial of (3.1) is given by (see [13], for a detailed discussion),

$$L(t) = \rho(t) - \bar{h}\sigma(t), \qquad \bar{h} = h\lambda_i. \tag{3.2}$$

The absolute stability region of (3.1) is then defined to be the region in the $\bar{h}$-plane for which the roots t_j of (3.2) is such that $|t_j| < 1$, $j = 1, \ldots, k$; $i = 1, \ldots, s$.

The standard approach for stability investigations is to consider the constant coefficients linear system

$$y' = Ay, \qquad y(a) = \eta, \qquad y \in R^s. \tag{3.3}$$

If (3.3) is to be solved for all equations in the systems by the explicit case of the Adams method in (3.1), then the backward difference formulation of the k-step method is given by

$$y_{n+1} - y_n - h \sum_{i=0}^{k-1} \gamma_i \nabla^i f_n = 0, \quad \text{where } \gamma_0 = 1 \tag{3.4}$$

and the associated stability polynomial given in (3.2) can be rearranged as

$$L(t) = t^k - t^{k-1} - h\lambda\Big(t^{k-1} + \gamma_1 t^{k-2}(t-1) + \gamma_2 t^{k-3}(t-1)^2 + \cdots + \gamma_{k-1}(t-1)^{k-1}\Big),$$

λ an eigenvalue of A.

Now let (3.3) be solved with different Adams methods for each equation in the system. Without loss of generality let the ith equation be solved by the k_i-step Adams explicit method such that $k_i \leqslant k_m$ for $i < m$. Hence $k_1 = \min_i(k_i)$. Then the backward difference formulation for solving (3.3) with the stated variable order multistep method is given by

$$\begin{aligned} y_{n+1} = y_n &+ hA\Big[y_n + \gamma_1 \nabla y_n + \gamma_2 \nabla^2 y_n + \cdots + \gamma_{K_1 - 1} \nabla^{K_1 - 1} y_n\Big] \\ &+ hA_1\Big[\gamma_{K_1} \nabla^{K_1} y_n + \cdots + \gamma_{K_2 - 1} \nabla^{K_2 - 1} y_n\Big] + \cdots \\ &+ hA_{s-1}\Big[\gamma_{K_{s-1}} \nabla^{k_s - 1} y_n + \cdots + \gamma_{k_n - 1} \nabla^{k_s - 1} y_n\Big], \end{aligned} \tag{3.5}$$

where

$$A_r = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \\ a_{r+1,1} & a_{r+1,2} & \cdots & a_{r+1,s} \\ \vdots & \vdots & & \vdots \\ a_{s,1} & a_{s,2} & \cdots & a_{s,s} \end{bmatrix}$$

i.e., the matrix with the first r rows having zero elements and the last $(s - r)$ rows with elements of A. The stability polynomial associated with (3.5) is then given by $L_k(t) = \det(B)$ where $k = k_s$ and

$$B = (t^k - t^{k-1})I - hA\Big[t^{k-1} + \gamma_1 t^{k-2}(t-1) + \cdots + \gamma_{k_1-1}t^{k-k_1}(t-1)^{k_1-1}\Big] - hA_1\Big[\gamma_{k_1}t^{k-k_1-1}(t-1)^{k_1} + \cdots + \gamma_{k_2-1}t^{k-k_2}(t-1)^{k_2-1}\Big]\cdots - hA_{s-1}\Big[\gamma_{k_{(s-1)}}t^{k-k_{(s-1)}-1}(t-1)^{k_{(s-1)}} + \cdots + \gamma_{k-1}(t-1)^{k-1}\Big]. \tag{3.6}$$

Consider

$$B_0(t) = (t^k - t^{k-1})I, \tag{3.7}$$

then the roots of $\det(B_0(t))$ are the spurious root 0 of multiplicity $(k-1)s$ and the nonspurious root 1 of multiplicity s.

Now consider,

$$B_1(t) = (t^k - t^{k-1})I - hAt^{k-1}. \tag{3.8}$$

Since the root of a polynomial is a continuous function of its coefficients and that the additional matrix term hAt^{k-1} in (3.8) is $O(h)$; then for small h the nonspurious root of $\det(B_1(t)) = 0$ is of the form $t = 1 + b_1h + b_2h^2 + \cdots$. However, we note that the determinant $\det((B_1(t))t^{-(k-1)})$ represents the Euler's stability polynomial. Hence, the roots of $\det(B_1(t)) = 0$ are given by $(t^{k-1})^s(t - (1 + \lambda_1 h))(t - (1 + \lambda_2 h))\cdots(t - (1 + \lambda_s h)) = 0$, where λ_1 are the eigenvalues of A. Therefore, the region of stability is given by $|t| = |1 + \lambda_1 h| < 1$ for $i = 1, \ldots, s$. (In our discussion we ignore the spurious roots near $t = 0$ because they cannot violate the stability condition for small h). Next consider,

$$B_2(t) = B_1(t) - hA\gamma_1 t^{k-2}(t-1). \tag{3.9}$$

Again here the coefficients of the polynomial $\det(B_2(t))$ differ from $\det(B_1(t))$ by the contribution of the matrix of $O(h)$, namely $-hA\gamma_1 t^{k-2}(t-1)$.

Again by the argument of the continuity of the roots of the polynomial, the principal root t^* of $\det(B(t)) = 0$ is of the form.

$$t^* = 1 + c_1 h + c_2 h^2 + c_3 h^3 + \cdots, \tag{3.10}$$

where the c_i are the constants.

Substituting t^* in $B_1(t)$ gives

$$\begin{aligned} B_1(t^*) &= \left(1 + c_1 h + c_2 h^2 + \cdots\right)^{k-1}\left(\left(c_1 h + c_2 h^2 + \cdots\right) I - hA\right) \\ &= (c_1 I - A)h + \left(\left(c_2 + c_1^2\binom{k-1}{1}\right) I - c_1\binom{k-1}{1} A\right) h^2 \\ &\quad + \left(\left(c_3 + 2c_1 c_2\binom{k-1}{1}\right) I + c_1^3\binom{k-1}{2}\right) I \\ &\quad - \left(\left(c_2\binom{k-1}{1} + c_1^2\binom{k-1}{2}\right) A\right) h^3 + \cdots. \end{aligned}$$

Also $hA\gamma_1 t^{*k-2}(t^* - 1) = hA\gamma_1(1 + c_1 h + c_2 h^2 + \cdots)^{k-2}(c_1 h + c_2 h^2 + \cdots) = A\gamma_1(c_1 h^2 + (c_2 + c_1^2\binom{k-2}{1})h^3 + \cdots)$. Note that the contribution of $hA\gamma_1 t^{*k-2}(t^* - 1)$ is a matrix of $O(h^2)$. Hence

$$\begin{aligned} B_2(t^*) &= (c_1 I - A)h + \left(\left(c_2 + c_1^2\binom{k-1}{2}\right) I - c_1\binom{k-1}{1} + \gamma_1((c_1) A\right) h^2 \\ &\quad + \left(\left(c_3 + 2c_1 c_2\binom{k-1}{1}\right) + c_1^3\binom{k-1}{2} I - c_2\binom{k-1}{1} + c_1^2\binom{k-1}{2}\right. \\ &\quad \left. + \gamma_1 c_1^2\binom{k-2}{1}\right) A\Big) h^3 + \cdots. \end{aligned} \tag{3.11}$$

Introduce the matrices $B_r(c_i)$, which are functions of the coefficients c_i, where

$$B_1^*(c_i) = c_1 I - A = (b_{11}^*, \ldots, b_{1s}^*)$$

$$B_2^*(c_i) = \left(c_2 + c_1^2\binom{k-1}{1}\right) I - \left(c_1\binom{k-1}{1} + \gamma_1 c_1\right) A = (b_{21}^*, \ldots, b_{2s}^*)$$

$$B_3^*(c_i) = \left(\left(c_3 + 2c_1c_2\binom{k-1}{1} + c_1^3\binom{k-1}{2}\right)\right) I$$

$$-\left(c_2\binom{k-1}{1} + c_1^2\binom{k-1}{2} + \gamma_1 c_1\binom{k-2}{1}\right) A$$

$$= (b_{31}^*, \ldots, b_{3s}^*).$$

In general, let $B_r^*(c_i)$ the coefficient of h^i in (3.11) be $B_r^*(c_i) = (b_{r1}^*, \ldots, b_{rs}^*)$ where the b_{rm}^* are $s \times 1$ vectors. Then $\det(B_2(t^*)) = \det(hB_1^* + h^2B_2^* + h^3B_3^* + \cdots)$. Using the lemma given earlier,

$$\det(B_2(t^*)) = h^s(\beta_1) + h\beta_2 + h^2\beta_3 + \cdots + h^1\beta_1 + \cdots \qquad (3.11)$$

where $h^s\beta_1 = \det(hB_1^*)$, $h^{s+1}\beta_2$ is the sum of determinants of each matrix formed from combinations of $(s-1)$ column vectors hb_{rm}^* of B_1^* and one from $h^2B_2^*$, $h^{s+2}\det(\beta_3)$ is the sum of determinants of each matrix formed using combinations of vectors hb_{1m}^*, $h^2b_{2m}^*$, *and* $h^3b_{3m}^*$ such that the determinant will factor out the h^{s+2} term, and so on for the others.

For $\det(B_2(t)) = 0$, then necessarily the coefficients of the powers of h in (3.12) must vanish. This implies $\beta_1 = 0$ or $\det(c_1 1 - A)$ must vanish, i.e., $c_1 = \lambda$. We note that $\beta_2 = 0$ involves coefficients c_1 and c_2, hence it will determine c_2. Similarly $\beta_3 = 0$ will determine c_3, and so on. Hence the contribution of the terms $hA\gamma_1 t^{k-2}(t-1)$ in (3.9) is at most of $O(h^2)$. Similarly consider

$$B_3(t) = B_2(t) - Ah\gamma_2 t^{k-3}(t-1)^2. \qquad (3.13)$$

Then again letting the root of (3.13) be

$$t^* = 1 + d_1h + d_2h^2 + d_3h^3 + \cdots \qquad (3.14)$$

we have

$$t^{*k-3}(t^* - 1)^2 = d_1^2 h^2 + \left(2 d_1 d_2 - 2 d_1^3 + 6 d_1^3 \binom{k-2}{1}\right) h^3 + \cdots .$$

Hence it is seen that the addition of the term $Ah\gamma_2 t^{*k-3}(t^* - 1)^2$ in (3.13) affects only the h^3 coefficients of $B_2(t^*)$. We have, $B_3(t^*) = h\hat{B}_1 + h^2\hat{B}_2 + h^3\hat{B}_3 + \cdots$ where $\hat{B}_r = \hat{B}_r(d_i)$ and $\hat{B}_1(d_i) = B_1^*(d_i)$, $\hat{B}_2(d_i) = B_2^*(d_i)$ and $\hat{B}_3(d_i) = B_3^*(d_i) - d_1^2\gamma_2 A$.

Hence by considering $\det(B_3(t^*)) = 0$, we will get $d_1 = c_1 = \lambda$, $d_2 = c_2$, but d_3 will in general differ from c_3. In general, if

$$B_{k_r+1} = B_{k_r} - hA_r\gamma_{k_r} t^{k-k_{r-1}}(t - 1)^{k_r} \tag{3.15}$$

where B_{k_r} are the terms in (3.6) up to

$$hA_{r-1}\gamma_{k_r-1} t^{k-k_r}(t - 1)^{k_r-1}$$

and further if the root of $\det(B_{k_r+1}) = 0$ is of the form in (3.14), and writing

$$\det(B_{k_r+1}) = \det(hD_1 + h^2 D_2 + \cdots), \tag{3.16}$$

then the D_m will consists of combinations of $d_i (i = 1, \ldots, m)$, $\gamma_1 (i = 1, \ldots, m - 1)$, and the matrices A_i associated with γ_i. Expanding the right-hand side of (3.16) by using the previous lemma and equating to zero the coefficients of h^i of the resulting expansion will determine the d_i.

If for the stability polynomial $\det(B_{k_r})$, the principal zero is of the form $t^* = 1 + d_1^* h + d_2^* h^2 + \cdots$, then it can be seen that $d_1 = d_1^*$, $d_2 = d_2^*, \ldots, d_{k_r} = d_{k_r}^*$ but in general $d_{k_r+1} \neq d_{k_r+1}^*$. In other words the additional terms in (3.14), a matrix of $O(h^{k_r+1})$, affects the principal root at $O(h^{k_r+1})$.

In this way it is seen that for t of the form in (3.10) to be a principal root of $\det(B) = 0$, then $c_1 = \lambda$ and the succeeding terms in (3.6) do not affect this conclusion.

Hence for small h, the roots of $L_k(t) = 0$ approximate to $O(h^2)$ the roots of the Euler stability polynomial (3.8), thereby verifying that a region of absolute stability exists for the method of (3.1). PECE with different orders for different equations in the system will also have a region of stability. If one examines the stability polynomial, one root (principal) will still be $t = 1 + \lambda h + \cdots$ and the rest tend to zero as $h \to 0$.

In the case of a reliable code using different orders for each equation in a system, the code then chooses the orders to maximize step lengths either for optimum region or to satisfy the accuracy requirement, whichever is governing the behavior of the integration process. This may lead to computational efficiency.

REFERENCES

1 C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, New Jersey 1971.

2 F. T. Krogh, A variable step, variable order multistep method for the numerical solution of ordinary differential equation, in *Proceedings of the IFIP Congress in Information Processing* 68, 1968, pp. 194–199.

3 L. F. Shampine and M. K. Gordon, *Computer Solution of Ordinary Differential Equations*, W. H. Freeman, San Francisco, 1975.

4 G. Hall and M. B. Suleiman, A single code for the solution of stiff and nonstiff ODEs, *SIAM J. Stat. Comput.* 6:684–697 (1985).

5 M. B. Suleiman and S. Baok, Using nonconvergence of iteration to partition ODEs, *Appl. Math. Comput.* 49:111–140 (1992).

6 C. W. Gear, The stability of numerical methods for second order differential equations, *J. SIAM. Numer. Anal.*, 15:188–197, (1977).

7 G. Söderlind, Some stability properties of linear multi-step compound discretizations of partitioned differential systems, TRITA-NA-7910, Department of Numerical Analysis Technical Report, The Royal Institute of Technology, Stockholm, Sweden, 1979.

8 W. H. Enright, T. E. Hull, and B. Lindberg, Comparing numerical methods for stiff system of ODEs, *BIT*, 15:10–48 (1975).

9 J. Bruder, K. Strehmel, and R. Weiner, Partitioned adaptive Runge-Kutta methods for the solution of nonstiff and stiff systems, *Numer. Math.*, 52:621–638 (1988).

10 S. P. Norsett and P. G. Thomsen, Switching between modified Newton and fix-point iteration for implicit ODE-solvers, *BIT* 26:339–348 (1986).

11 L. Petzold, Automatic selection of methods for solving stiff and nonstiff system of ordinary differential equations, *SIAM J. Sci. Stat. Comput.* 4:136–148 (1983).

12 P. Rentrop, Partitioned Runge-Kutta methods with stiffness detection and stepsize control, *Numer. Math.* 47:545–564 (1985).

13 J. D. Lambert, *Computational Methods in Ordinary Differential Equations*, John Wiley, New York, 1973.