Applied Numerical Mathematics 11 (1993) 347–363
North-Holland

APNUM 380

Diagonally-implicit multi-stage integration methods

J.C. Butcher

Department of Mathematics and Statistics, University of Auckland, Private Bag 92019, Auckland, New Zealand

Abstract

Butcher, J.C., Diagonally-implicit multi-stage integration methods, Applied Numerical Mathematics 11 (1993) 347–363.

Even though general linear methods were proposed as long as 25 years ago, they have never been widely adopted as practical methods for incorporation into numerical software. The aim of the presented paper is to select from the large family of possible general linear methods, just a single class which has considerable potential for efficient implementation. The class, which will be known by the acronym DIMSIM, divides into four special subclasses. These have possible applications depending on the stiff or nonstiff nature of a problem to be solved and the parallel or sequential nature of the computing environment in which the solution is to be performed.

"Dimsim: A savoury Cantonese-style snack; a meal consisting of these."
The Oxford English Dictionary, 2nd ed., 1989.

1. Introduction

The methods with which we are concerned are motivated in a number of ways. The first is a desire to identify, from the enormous class of general linear methods, some special cases which have potential for practical use. Ideally, these new methods will have advantages not possessed by either Runge–Kutta or by linear multistep or other successful known methods. The additional motivations are aimed at overcoming some of the major disadvantages in known methods.

These disadvantages include the lack of high-order A-stable linear multistep methods and the high implementation costs for implicit Runge–Kutta methods. They also include the degradation of performance of methods with low stage order, such as diagonally-implicit Runge–Kutta methods and the additional costs required for many methods to provide reliable local error estimates and dense output. With the current interest in parallel computation and the difficulty of adapting existing methods to parallel operation, a further motivation is evident: to find methods for which parallelism is available and achievable.

Correspondence to: J.C. Butcher, Department of Mathematics and Statistics, University of Auckland, Private Bag 92019, Auckland, New Zealand.

General linear methods for the solution of a differential equation system have been discussed in a number of publications such as [2–4]. We will use the notation of [8] in this paper so that such a method is represented by a partitioned $(s+r)\times(s+r)$ matrix

$$C=\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}. \tag{1.1}$$

Denote by $y_1^{[n-1]}, y_2^{[n-1]},\ldots,y_r^{[n-1]}$, the r quantities computed in step number $n-1$ and available for use in computing $y_1^{[n]}, y_2^{[n]},\ldots,y_r^{[n]}$, in step number n. Also denote by $Y_1, Y_2,\ldots,Y_s$, the s stage values evaluated in this step. Thus, if the differential equation for which a numerical approximation is required, is

$$y'(x)=f(y(x)), \tag{1.2}$$

then the stage derivatives are given by $f(Y_1), f(Y_2),\ldots,f(Y_s)$. Write h for the stepsize and N for the dimension of the problem.

Let the (i,j) element of a submatrix C_{IJ}, $I,J=1,2$, occurring in (1.1), be written as c_{ij}^{IJ}. Then the stage values and the output values from step number n are given by

$$Y_i=h\sum_{j=1}^{s}c_{ij}^{11}f(Y_j)+\sum_{j=1}^{r}c_{ij}^{12}y_j^{[n-1]},\qquad i=1,2,\ldots,s, \tag{1.3}$$

$$y_i^{[n]}=h\sum_{j=1}^{s}c_{ij}^{21}f(Y_j)+\sum_{j=1}^{r}c_{ij}^{22}y_j^{[n-1]},\qquad i=1,2,\ldots,r.$$

We note that the C_{11} matrix has a similar role to the A matrix in a Runge–Kutta method. The structure of this matrix will determine the implementation costs. These will be low in the explicit case in which C_{11} is strictly lower triangular. Furthermore, in this explicit case, the height of the digraph determined by its pattern of nonzeros, will be a measure of its cost in an appropriate parallel environment [9].

For implicit methods, in which C_{11} is nonsingular, singly-implicit and preferably diagonally-implicit, structures are necessary to lower implementation costs. To see why this is the case, consider the system of nonlinear algebraic equations (1.3) to be solved in the evaluation of the stage values. Usually, systems of equations of this type are solved using a modified Newton scheme. If this approach is used, the linear equation system to be solved in each iteration has sN dimensions. For diagonally-implicit methods, this linear system is replaced by a sequence of s linear systems, each of N dimensions. It was shown in [7] that, for large N, a similar saving can be realized if C_{11} is "singly-implicit"; that is, the spectrum of C_{11} contains only a single eigenvalue.

We also note that the matrix C_{22} determines stability (in the classical zero-stability or Dahlquist sense). Thus, power-boundedness of C_{22} is necessary for convergence. Even better seems to be a spectrum consisting of a single eigenvalue at 1 with the remainder at 0. This strong version of the stability property is possessed by Runge–Kutta as well as by Adams methods and it is hoped that the good stability of these traditional methods will carry over to the methods proposed in this paper.

In addition to stability and order, a number of other characteristics of a method play a part in determining its accuracy. For example, it is now recognized that high stage order is an important property of methods intended for stiff problems.

Because of these various considerations, we will deal with methods characterized by the following requirements:

(a) The matrix C_{11} should have lower triangular form with a constant value on the diagonal.
(b) The stage order and the overall order should be identical.
(c) Convenient algorithms should exist for generating starting values.
(d) If possible, C_{22} should satisfy the strong stability property discussed above.
(e) Convenient means should be available for estimating accuracy, providing interpolatory output, and changing stepsize.

The next section will expand further on requirement (a) and we will there specify what we mean by a DIMSIM method. In Section 3, we will explain how the order of the new methods is specified. This will be done on the basis of always requiring (b) to be satisfied. In Section 4 we will show by a detailed case study how a DIMSIM can be derived and this will lead in Section 5 to the identification of a family of DIMSIMs for which $r = s = p$. Members of this family of various types are considered in Sections 6, 7, and 8. An outline of further developments is presented in Section 9.

2. DIMSIMs

Diagonally-implicit Runge–Kutta methods were introduced [1] as a means of obtaining good accuracy and stability without the excessive implementation costs associated with a general implicit Runge–Kutta method. We will maintain this design feature in our general linear setting. Thus, we will always assume that C_{11} takes the form

$$C_{11} = \begin{bmatrix} \lambda & 0 & \cdots & 0 \\ a_{21} & \lambda & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ a_{s1} & a_{s2} & \cdots & \lambda \end{bmatrix}. \tag{2.1}$$

Because we will wish to be able to solve both nonstiff and stiff problems, we will need to consider cases when λ is and is not zero. These will be referred to respectively as explicit DIMSIMs and implicit DIMSIMs.

We consider also two further cases. In the explicit case, if $C_{11} = 0$, then all stages can be evaluated simultaneously or in any order. Thus a method of this type could be useful in a parallel environment. Similarly, for the implicit case, if $C_{11} = \lambda I$, for I the unit matrix, then each of the stages can be computed simultaneously although each of them separately requires an iterative method of computation.

To summarize this situation, we will list in Table 1 the four cases which we will name DIMSIMs of types 1, 2, 3, and 4.

Note that other possibilities exist and should also be considered. For example, between type 1 (where no parallelism is allowed for) and type 3 (with complete parallelism) there exist choices with intermediate levels of parallelism. Similar remarks apply to types 2 and 4. Finally, an obvious extension to the idea of type-4 methods is to allow the matrix A to be

$$A = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_s).$$

3. Order conditions

For convenience we will write DIMSIM methods in the form of general linear methods with

$$C_{11}=A, \qquad C_{12}=U, \qquad C_{21}=B, \qquad C_{22}=V.$$

We use the definition of order introduced by the author in a number of papers [4–6] with the additional requirement that the stage values have a specified order. This additional requirement actually allows a considerable simplification of the order conditions and it is no longer necessary to use the elaborate and difficult theory in [8]. We write p for the order being sought.

The key idea is to use starting values of the form

$$y_i^{[0]} = \sum_{k=0}^{p} \alpha_{ik} y^{(k)}(x_0)h^k + O(h^{p+1}), \tag{3.1}$$

where $y_i^{[n]}$ denotes approximation number i at integration point number n. The numbers α_{ik} must be chosen so that, within the first step, with stepsize h, the stage values, which are supposed to be approximations to the solution at points x_0+hc_i, $i=1, 2, \ldots, s$, are given by

$$Y_i = \sum_{k=0}^{p} \frac{c_i^k}{k!} y^{(k)}(x_0)h^k + O(h^{p+1}), \tag{3.2}$$

Table 1
The four DIMSIM types

	C_{11} structure	stiff vs. nonstiff	parallel vs. serial
Type 1	$\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ a_{21} & 0 & 0 & \cdots & 0 \\ a_{31} & a_{32} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ a_{s1} & a_{s2} & a_{s3} & \cdots & 0 \end{bmatrix}$	nonstiff	serial
Type 2	$\begin{bmatrix} \lambda & 0 & 0 & \cdots & 0 \\ a_{21} & \lambda & 0 & \cdots & 0 \\ a_{31} & a_{32} & \lambda & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ a_{s1} & a_{s2} & a_{s3} & \cdots & \lambda \end{bmatrix}$	stiff	serial
Type 3	$\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$	nonstiff	parallel
Type 4	$\begin{bmatrix} \lambda & 0 & 0 & \cdots & 0 \\ 0 & \lambda & 0 & \cdots & 0 \\ 0 & 0 & \lambda & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \lambda \end{bmatrix}$	stiff	parallel

and the output values computed at the end of the step are given by

$$y_i^{[1]} = \sum_{k=0}^{p} \alpha_{ik} y^{(k)}(x_1) h^k + O(h^{p+1}), \tag{3.3}$$

with $x_1 = x_0 + h$.

Note that, even though the starting values are given in (3.1) in terms of weighted Taylor series, there will not be any need in practice to evaluate the various derivatives $y^{(k)}(x_0)$, $k = 1, 2, \ldots$, which occur in this formula. If approximations have been obtained for sufficiently many solution values at equally spaced arguments—as is always needed by linear multistep methods for example—then approximations to the quantities appearing in (3.1) can always be found from these. However, it is hoped that families of DIMSIMs of various orders can be found for which it would be an easy matter to move orders upwards one unit at a time. In the very first step of an actual integration, order 1 would be used and this would not require anything more than the initial value associated with the problem being solved.

We now show how the requirements expressed by (3.2) and (3.3) can be represented in a simple way using functions of a complex variable.

Theorem 3.1. *The numbers* α_{ik}, $i = 1, 2, \ldots, r$, $k = 0, 1, 2, \ldots, p$, *in* (3.1) *and the matrices* A, B, U, *and* V *satisfy* (3.2) *and* (3.3) *iff*

$$e^{cz} = zAe^{cz} + Uw + O(z^{p+1}), \tag{3.4}$$

$$e^{z}w = zBe^{cz} + Vw + O(z^{p+1}), \tag{3.5}$$

where e^{cz} *denotes the vector with components given by* $e^{c_i z}$, $i = 1, 2, \ldots, s$, *and* w *denotes the vector with elements given by*

$$w_i = \sum_{k=0}^{p} \alpha_{ik} z^k, \quad i = 1, 2, \ldots, r. \tag{3.6}$$

Proof. Since Y_i, given by (3.2), is equal to

$$Y_i = y(x_0 + hc_i) + O(h^{p+1}), \tag{3.7}$$

it follows that

$$\begin{aligned} hf(Y_i) &= hy'(x_0 + hc_i) + O(h^{p+2}) \\ &= \sum_{k=1}^{p+1} \frac{c_i^{k-1}}{(k-1)!} y^{(k)}(x_0) h^k + O(h^{p+2}) \\ &= \sum_{k=1}^{p} \frac{c_i^{k-1}}{(k-1)!} y^{(k)}(x_0) h^k + O(h^{p+1}). \end{aligned}$$

Furthermore, by using Taylor expansions, we see that (3.3) can be written in the form

$$y_i^{[1]} = \sum_{k=0}^{p} \left(\sum_{l=0}^{k} \frac{1}{l!} \alpha_{i,k-l} \right) y^{(k)}(x_0) h^k + O(h^{p+1}).$$

Use the equations

$$Y_i = h\sum_{j=1}^{s} a_{ij} f(Y_j) + \sum_{j=1}^{r} u_{ij} y_j^{[0]}$$

and

$$y_i^{[1]} = h\sum_{j=1}^{s} b_{ij} f(Y_j) + \sum_{j=1}^{r} v_{ij} y_j^{[0]},$$

which define the computations performed in the first step, and it is found that

$$\sum_{k=0}^{p}\left(c_i^k - \sum_{j=1}^{s} k a_{ij} c_j^{k-1} - \sum_{j=1}^{r} u_{ij}\alpha_{jk} k!\right)\frac{h^k}{k!} y^{(k)}(x_0) = O(h^{p+1}),$$

$$\sum_{k=0}^{p}\left(\sum_{l=0}^{k}\frac{k!}{l!}\alpha_{i,k-l} - \sum_{j=1}^{s} k b_{ij} c_j^{k-1} - \sum_{j=1}^{r} v_{ij}\alpha_{jk} k!\right)\frac{h^k}{k!} y^{(k)}(x_0) = O(h^{p+1}).$$

Equating coefficients of $h^k y^{(k)}(x_0)/k!$, $k = 0, 1, \ldots, p$, to zero and then multiplying these coefficients by $z^k/k!$ and adding from $k = 0$ to $k = p$, we obtain

$$e^{c_i z} - \sum_{j=1}^{s} z a_{ij} e^{c_j z} - \sum_{j=1}^{r} u_{ij} w_j = O(z^{p+1}), \qquad i = 1, 2, \ldots, s,$$

$$e^{z} w_i - \sum_{j=1}^{s} z b_{ij} e^{c_j z} - \sum_{j=1}^{r} v_{ij} w_j = O(z^{p+1}), \qquad i = 1, 2, \ldots, r,$$

which are equivalent to (3.4) and (3.5) respectively. □

4. An example of a DIMSIM derivation

As an example of the derivation of a DIMSIM, we consider the construction of a type-1 method with $r = s = 3$ and $p = 5$. As a design choice, we select $U = I$ and suppose that

$$V = \begin{bmatrix} v_1 & v_2 & v_3 \\ v_1 & v_2 & v_3 \\ v_1 & v_2 & v_3 \end{bmatrix}.$$

The reason for this choice of V is to force there to be good stability at 0. Note that $Ve = e$ is implied by the results of Theorem 3.1. For convenience, we select $c_1 = 0$. Certain choices of c_2 and c_3 must be excluded so that the denominators in expressions found below do not vanish. For example, $c_3 \neq 0$, $c_3 \neq 1$, and $c_3 \neq c_2$ are necessary to avoid a zero denominator in (4.12).

Since $U = I$ and A is strictly lower triangular, the components of w are given by

$$w_1 = 1, \tag{4.1}$$

$$w_2 = e^{c_2 z} - z a_{21} + O(z^6), \tag{4.2}$$

$$w_3 = c^{c_3 z} - z(a_{31} + a_{32} e^{c_2 z}) + O(z^6), \tag{4.3}$$

and the order conditions (3.5) become

$$e^z w_1 = z(b_{11} + b_{12}e^{c_2 z} + b_{13}e^{c_3 z}) + v_1 w_1 + v_2 w_2 + v_3 w_+ O(z^6), \tag{4.4}$$

$$e^z w_2 = z(b_{21} + b_{22}e^{c_2 z} + b_{23}e^{c_3 z}) + v_1 w_1 + v_2 w_2 + v_3 w_3 + O(z^6), \tag{4.5}$$

$$e^z w_3 = z(b_{31} + b_{32}e^{c_2 z} + b_{33}e^{c_3 z}) + v_1 w_1 + v_2 w_2 + v_3 w_3 + O(z^6). \tag{4.6}$$

Subtract (4.4) from (4.5), substitute from (4.1) and (4.2) and divide by z. It is found that

$$\frac{e^{(1+c_2)z} - e^z}{z} = a_{21}e^z + (b_{21} - b_{11}) + (b_{22} - b_{12})e^{c_2 z} + (b_{23} - b_{13})e^{c_3 z} + O(z^5). \tag{4.7}$$

Eliminate a_{21}, $b_{21} - b_{11}$, $b_{22} - b_{12}$, and $b_{23} - b_{13}$ by applying to (4.7) the operator $D(D-1)\cdot(D-c_2)(D-c_3)$, where D denotes differentiation with respect to z. It is found that

$$D(D-1)(D-c_2)(D-c_3)\left(\frac{e^{(1+c_2)z} - e^z}{z}\right) = O(z),$$

which is equivalent to

$$\int_1^{1+c_2} x(x-1)(x-c_2)(x-c_3)\,dx = 0,$$

and, on the assumption that $c_2 \neq 0$, this implies that

$$c_3 = \frac{30 + 30c_2 + 5c_2^2 - 3c_2^3}{30 + 10c_2 - 5c_2^2}. \tag{4.8}$$

Having selected values of c_2 and c_3 which satisfy (4.8), a_{21}, $b_{21} - b_{11}$, $b_{22} - b_{12}$, and $b_{23} - b_{13}$ can be evaluated by operating on (4.7) with each of $D(D-c_2)(D-c_3)$, $(D-1)(D-c_2)\cdot(D-c_3)$, $D(D-1)(D-c_3)$, and $D(D-1)(D-c_2)$, respectively, and setting $z=0$ in each case. It is found in this way that

$$a_{21} = \frac{\int_1^{1+c_2} x(x-c_2)(x-c_3)\,dx}{(1-c_2)(1-c_3)},$$

$$b_{21} - b_{11} = \frac{-\int_1^{1+c_2} (x-1)(x-c_2)(x-c_3)\,dx}{c_2 c_3},$$

$$b_{22} - b_{12} = \frac{\int_1^{1+c_2} x(x-1)(x-c_3)\,dx}{c_2(c_2-1)(c_2-c_3)},$$

$$b_{23} - b_{13} = \frac{\int_1^{1+c_2} x(x-1)(x-c_2)\,dx}{c_3(c_3-1)(c_3-c_2)}.$$

Thus,

$$a_{21} = \frac{c_2(12 + 6c_2 - c_2^3) - c_2c_3(12 - 2c_2^2)}{12(1 - c_2)(1 - c_3)}, \tag{4.9}$$

$$b_{21} - b_{11} = \frac{c_2(-6 - 2c_2 + c_2^2) + c_2c_3(6 - 2c_2)}{12c_3}, \tag{4.10}$$

$$b_{22} - b_{12} = \frac{c_2(6 + 8c_2 + 3c_2^2) - c_2c_3(6 + 4c_2)}{12(c_2 - 1)(c_2 - c_3)}, \tag{4.11}$$

$$b_{23} - b_{13} = \frac{c_2^2(6 + 2c_2 - c_2^2)}{12c_3(c_3 - 1)(c_3 - c_2)}. \tag{4.12}$$

Similarly, to find the values of a_{31}, a_{32}, $b_{31} - b_{11}$, $b_{32} - b_{12}$, and $b_{33} - b_{13}$, we find from (4.4) and (4.6) that

$$\frac{e^{(1+c_3)z} - e^z}{z} = a_{31}e^z + a_{32}e^{(1+c_2)z} + (b_{31} - b_{11}) + (b_{32} - b_{12})e^{c_2z} + (b_{33} - b_{13})e^{c_3z} + O(z^5),$$

and this leads to the conclusions that

$$\begin{aligned} a_{31} &= \frac{-\int_1^{1+c_3} x(x - 1 - c_2)(x - c_2)(x - c_3)\,dx}{c_2(1 - c_2)(1 - c_3)} \\ &= \frac{c_3(60c_2 - 60c_2^2) + c_3^2(-30 + 60c_2)}{60c_2(1 - c_2)(1 - c_3)} \\ &\quad + \frac{c_3^3(-30 + 10c_2 + 10c_2^2) + c_3^4(-5 - 10c_2) + 3c_3^5}{60c_2(1 - c_2)(1 - c_3)}, \end{aligned} \tag{4.13}$$

$$\begin{aligned} a_{32} &= \frac{\int_1^{1+c_3} x(x - 1)(x - c_2)(x - c_3)\,dx}{c_2(1 + c_2)(1 + c_2 - c_3)} \\ &= \frac{c_3^2(30 - 30c_2) + c_3^3(30 - 10c_2) + c_3^4(5 + 5c_2) - 3c_3^5}{60c_2(1 + c_2)(1 + c_2 - c_3)}, \end{aligned} \tag{4.14}$$

$$\begin{aligned} b_{31} - b_{11} &= \frac{\int_1^{1+c_3} (x - 1)(x - 1 - c_2)(x - c_2)(x - c_3)\,dx}{c_2c_3(1 + c_2)} \\ &= \frac{c_3(-30c_2 + 30c_2^2) + c_3^2(20 - 30c_2 - 10c_2^2) + c_3^3(10 + 10c_2) - 3c_3^4}{60c_2(1 + c_2)}, \end{aligned} \tag{4.15}$$

$$b_{32}-b_{12}=\frac{-\int_1^{1+c_3}x(x-1)(x-1-c_2)(x-c_3)\,\mathrm{d}x}{c_2(c_2-1)(c_2-c_3)}$$
$$=\frac{30c_2c_3^2+c_3^3(-20+10c_2)+c_3^4(-10-5c_2)+3c_3^5}{60c_2(c_2-1)(c_2-c_3)}, \tag{4.16}$$

$$b_{33}-b_{13}$$
$$=\frac{\int_1^{1+c_3}x(x-1)(x-1-c_2)(x-c_2)\,\mathrm{d}x}{c_3(c_3-1)(c_3-1-c_2)(c_3-c_2)}$$
$$=\frac{c_3(-15c_2+15c_2^2)+c_3^2(10-30c_2+10c_2^2)+c_3^3(15-15c_2)+6c_3^4}{30(c_3-1)(c_3-c_2-1)(c_3-c_2)}. \tag{4.17}$$

To evaluate v_1, v_2, and v_3 and to find the values of b_{11}, b_{12}, and b_{13}, write (4.4) in the form

$$e^z=[v_1+(b_{11}-v_2a_{21}-v_3a_{31})z]$$
$$+[v_2+(b_{12}-v_3a_{32})z]e^{c_2z}+[v_3+b_{13}z]e^{c_3z}+\mathrm{O}(z^6)$$

and operate in turn by $D(D-c_2)^2(D-c_3)^2$, $(D-c_2)^2(D-c_3)^2$, $D^2(D-c_2)(D-c_3)^2$, $D^2(D-c_3)^2$, $D^2(D-c_2)^2(D-c_3)$, and $D^2(D-c_2)^2$, followed in each case by the substitution $z=0$. It is found, respectively, that

$$b_{11}-v_2a_{21}-v_3a_{31}=\frac{(1-c_2)^2(1-c_3)^2}{c_2^2c_3^2}, \tag{4.18}$$

$$v_1=\frac{(1-c_2)^2(1-c_3)^2(c_3(c_2+2)+2c_2)}{c_2^3c_3^3}, \tag{4.19}$$

$$b_{12}-v_3a_{32}=\frac{(1-c_2)(1-c_3)^2}{c_2^2(c_3-c_2)^2}, \tag{4.20}$$

$$v_2=\frac{(1-c_3)^2(c_3(3c_2-2)+4c_2-5c_2^2)}{c_2^3(c_3-c_2)^3}, \tag{4.21}$$

$$b_{13}=\frac{(1-c_2)^2(1-c_3)}{c_3^2(c_3-c_2)^2}, \tag{4.22}$$

$$v_3=\frac{(1-c_2)^2(5c_3^2-c_3(3c_2+4)+2c_2)}{c_3^3(c_3-c_2)^3}. \tag{4.23}$$

The method derived here is not claimed to be of practical use and, for this reason, further details will not be given. Experience gained in deriving a wide variety of DIMSIM methods has led to a preference for selecting r and s as each being equal to the order p which is to be imposed. Even though these values of r and s are undoubtedly greater than the minimum

required to obtain a given order, the additional freedom allows other desirable choices to be made. We will consider the systematic development of a special family of DIMSIMs in th next section.

5. A general class of DIMSIMs

In this section, we suppose that $U = I$, and that c is given as are A and V, with $Ve = e$, as required by the order conditions. Our aim is to find a formula for B so as to achieve an order $p = r = s$. We restrict ourselves to the non-confluent case, so that all components of c are distinct. Having found methods of this type, we can ask how the elements of A and V might be chosen to achieve some desirable goal such as a good stability region.

The key result of the section is a formula for the matrix B.

Theorem 5.1. *Let $r = s = p$. Then the DIMSIM*

$$\begin{bmatrix} A & I \\ B & V \end{bmatrix},$$

with $Ve = e$, is of order p and of stage order p if and only if

$$B = B_0 - AB_1 - VB_2 + VA,$$

where the (i, j) elements of B_0, B_1, and B_2 are given respectively by

$$\frac{\int_0^{1+c_i} \phi_j(x)\,\mathrm{d}x}{\phi_j(c_j)}, \qquad \frac{\phi_j(1+c_i)}{\phi_j(c_j)}, \quad \text{and} \quad \frac{\int_0^{c_i} \phi_j(x)\,\mathrm{d}x}{\phi_j(c_j)}$$

and, for $j = 1, 2, \ldots, s$, ϕ_j is given by

$$\phi_j(x) = \prod_{k \neq j} (x - c_k).$$

Proof. From Theorem 3.1, the vector-valued function w satisfies the relation

$$w(z) = (I - zA)e^{cz} + \mathrm{O}(z^{p+1})$$

(given by (3.4)) so that, substituting into (3.5), it is found that

$$e^z(I - zA)e^{cz} = zBe^{cz} + V(I - zA)e^{cz} + \mathrm{O}(z^{p+1}).$$

Hence,

$$(B - VA)ze^{cz} = e^z e^{cz} - A(ze^z e^{cz}) - Ve^{cz} + \mathrm{O}(z^{p+1}). \tag{5.1}$$

The condition $Ve = e$ is found by substituting $z = 0$. Given this condition, we operate on (5.1) by $\Phi_j(D)$, $j = 1, 2, \ldots, s$, where

$$\Phi_j(x) = \int_0^x \phi_j(t)\,\mathrm{d}t,$$

and set $z = 0$. Since the set of polynomials consisting of 1 together with Φ_j, for $j = 1, 2, \ldots, s$, forms a basis for the polynomials of degree not exceeding $s = p$, the result of carrying out these

operations and then substituting $z = 0$ is equivalent to equating the coefficients up to degree p in the Taylor series of (5.1) to zero.

The result of these computations is

$$(B - VA)\big(\Phi_j(D)ze^{cz}\big)\big|_{z=0}$$
$$= \big(\Phi_j(D)e^z e^{cz}\big)\big|_{z=0} - A\big(\Phi_j(D)ze^z e^{cz}\big)\big|_{z=0} - V\big(\Phi_j(D)e^{cz}\big)\big|_{z=0},$$

which is equivalent to

$$(B - VA - B_0 + AB_1 + VB_2)e_j = 0,$$

for e_j a member of the usual basis for $\mathbb{R}^s$. □

To assist in the use of the result of Theorem 5.1, we give the values of the matrices B_0, B_1, and B_2 for some simple special cases:

$$c = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad B_0 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 2 \end{bmatrix}, \qquad B_1 = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 0 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix},$$

and

$$c = \begin{bmatrix} 0 \\ \frac{1}{2} \\ 1 \end{bmatrix}, \qquad B_0 = \begin{bmatrix} \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \frac{3}{8} & 0 & \frac{9}{8} \\ \frac{4}{3} & -\frac{8}{3} & \frac{10}{3} \end{bmatrix},$$

$$B_1 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & -3 & 3 \\ 3 & -8 & 6 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 0 & 0 & 0 \\ \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{bmatrix}.$$

6. A type-1 DIMSIM

As an illustration of the result of the previous section, a type-1 DIMSIM will be derived with $r = s = p = 2$.

We will choose $c_1 = 0$ and $c_2 = 1$ so that the matrices B_0, B_1, and B_2 given for this choice in Section 5 can be used. So that good stability at 0 can be achieved, we choose V of the form

$$V = \begin{bmatrix} 1 - v & v \\ 1 - v & v \end{bmatrix}$$

which, with the type-1 choice of A of the form

$$A = \begin{bmatrix} 0 & 0 \\ a & 0 \end{bmatrix},$$

gives B as

$$B = \begin{bmatrix} \frac{1}{2}(1 - v) + av & \frac{1}{2}(1 - v) \\ -\frac{1}{2}v + av & 2 - \frac{1}{2}v - a \end{bmatrix}.$$

From the choices available to us, we select a and v in such a way that the linear stability properties for the method are exactly as for two-stage second-order explicit Runge–Kutta methods. This is, of course, a somewhat arbitrary choice, but it does have the advantage of providing a direct comparison with a standard method.

Since the stability is determined by the matrix

$$\begin{aligned} M(z) &= V + zB(I - zA)^{-1} \\ &= V + zB + z^2BA, \end{aligned}$$

and since

$$\det(M(z)) = z\left(\tfrac{3}{2} - v - a(1-v)\right) + z^2(1-v)\left(1 - \tfrac{1}{2}a\right),$$

this is achieved if and only if

$$\tfrac{3}{2} - v - a(1-v) = 0,$$

and

$$(1-v)\left(1 - \tfrac{1}{2}a\right) = 0,$$

that is, if and only if $a = 2$ and $v = \frac{1}{2}$.

With this choice the method takes the form

$$\left[\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \\ \hline \frac{5}{4} & \frac{1}{4} & \frac{1}{2} & \frac{1}{2} \\ \frac{3}{4} & -\frac{1}{4} & \frac{1}{2} & \frac{1}{2} \end{array}\right].$$

This DIMSIM has similar computational costs to that of a two-stage second-order Runge–Kutta method and, as we have remarked, has identical linear stability behaviour. The advantage it has over a Runge–Kutta method is that it has stage order 2. This makes it possible to achieve dense output by interpolation and to obtain low cost error estimates.

The key to these features is the derivation of approximations for $y(x_1)$, $hy'(x_1)$, and $h^2y''(x_1)$ of the form

$$\begin{aligned} y(x_1) &\approx \beta_{01}hf(Y_1) + \beta_{02}hf(Y_2) + \gamma_{01}y_1^{[0]} + \gamma_{02}y_2^{[0]}, \\ hy'(x_1) &\approx \beta_{11}hf(Y_1) + \beta_{12}hf(Y_2) + \gamma_{11}y_1^{[0]} + \gamma_{12}y_2^{[0]}, \\ h^2y''(x_1) &\approx \beta_{21}hf(Y_1) + \beta_{22}hf(Y_2) + \gamma_{21}y_1^{[0]} + \gamma_{22}y_2^{[0]}. \end{aligned}$$

By extending the ideas of Theorem 3.1, it can be shown that these approximations are correct to within $O(h^3)$ iff

$$\begin{aligned} e^z\begin{bmatrix} 1 \\ z \\ z^2 \end{bmatrix} &= z\begin{bmatrix} \beta_{01} & \beta_{02} \\ \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix}e^{cz} + \begin{bmatrix} \gamma_{01} & \gamma_{02} \\ \gamma_{11} & \gamma_{12} \\ \gamma_{21} & \gamma_{22} \end{bmatrix}w + O(z^3) \\ &= z\tilde{B}e^{cz} + \tilde{V}w + O(z^3), \end{aligned}$$

say, and this implies

$$\tilde{B} = \begin{bmatrix} 2 - \frac{3}{2}g_0 & \frac{1}{2}g_0 \\ -\frac{3}{2}g_1 & 1 + \frac{1}{2}g_1 \\ -1 - \frac{3}{2}g_2 & 1 + \frac{1}{2}g_2 \end{bmatrix}, \qquad \tilde{V} = \begin{bmatrix} g_0 & 1 - g_0 \\ g_1 & -g_1 \\ g_2 & -g_2 \end{bmatrix},$$

for g_0, g_1, and g_2 arbitrary.

We wish to restrict the choice of g_0, g_1, and g_2 so that if the approximate $y(x_1)$, $hy'(x_1)$, and $h^2y''(x_1)$ are substituted into (3.1) (with x_0 replaced by x_1 and with the $O(h^3)$ term omitted), then the values found for $y_1^{[1]}$ and $y_2^{[1]}$ will be identical to those computed using one step of the method. If W is given by

$$W = \begin{bmatrix} \alpha_{10} & \alpha_{11} & \alpha_{12} \\ \alpha_{20} & \alpha_{21} & \alpha_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & \frac{1}{2} \end{bmatrix},$$

then this means that

$$W\tilde{B} = B, \qquad W\tilde{V} = V,$$

implying $g_0 = \frac{1}{2}$ and $g_2 = 2g_1$. Write $g_1 = g$ so that

$$\tilde{B} = \begin{bmatrix} \frac{5}{4} & \frac{1}{4} \\ -\frac{3}{2}g & 1 + \frac{1}{2}g \\ -1 - 3g & 1 + g \end{bmatrix}, \qquad \tilde{V} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ g & -g \\ 2g & -2g \end{bmatrix}.$$

To determine a suitable choice of g, consider the procedure for changing stepsize using the Nordsieck technique. At the end of step n, the three quantities $y(x_n)$, $hy'(x_n)$, and $h^2y''(x_n)$ are rescaled by factors 1, δ, and δ^2 respectively, where δ denotes the ratio of new to old stepsize. Once this has been done, the method can be restarted using the elements in the matrix W applied to the rescaled weighted derivatives.

For this single step, the zero stability properties are determined by the matrix $WD\tilde{V}$, where $D = \text{diag}(1, \delta, \delta^2)$. It is found that

$$WD\tilde{V} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} - \delta g + \delta^2 g & \frac{1}{2} + \delta g - \delta^2 g \end{bmatrix},$$

with eigenvalues 1 and $\delta g - \delta^2 g$. Since we do not wish to be restricted by stability any more than necessary, it seems best to choose $g = 0$.

Finally, we consider the estimation of local truncation errors. The components of the local truncation error itself can be found from the coefficient of z^{p+1} in the vector

$$\exp(z)w(z) - zB\exp(cz) - Vw(z).$$

This is found to be $[\frac{1}{24}, \frac{7}{24}]^{\mathrm{T}}$ implying an error in the two components of

$$\tfrac{1}{24}h^3y^{(3)}(x_n) + O(h^4) \quad \text{and} \quad \tfrac{7}{24}h^3y^{(3)}(x_n) + O(h^4),$$

respectively. Because $\frac{1}{2}(y_1^{[n]} + y_2^{[n]})$ is propagated to succeeding steps, the local truncation error is effectively

$$\tfrac{1}{2}(\tfrac{1}{24} + \tfrac{7}{24})h^3y^{(3)}(x_n) + O(h^4) = \tfrac{1}{6}h^3y^{(3)}(x_n) + O(h^4).$$

It is easy to see that, assuming constant stepsize, an asymptotically correct estimate of this local truncation error may be found as

$$\tfrac{1}{6}hf(Y_2) - \tfrac{1}{2}hf(Y_1) + \tfrac{1}{3}\left(y_1^{[n-1]} - y_2^{[n-1]}\right).$$

In the case of variable stepsize, this estimate must be slightly modified if it is to approximate the local truncation error correctly. In fact, it turns out that, if δ is the ratio of the current stepsize to the stepsize in the previous step, then the appropriate error estimator is

$$\frac{2\delta}{1+\delta}\left(\tfrac{1}{6}hf(Y_2) - \tfrac{1}{2}hf(Y_1) + \tfrac{1}{3}\left(y_1^{[n-1]} - y_2^{[n-1]}\right)\right).$$

7. A type-2 DIMSIM

In this section we consider the design of a method suitable for stiff problems in a conventional environment. Specifically, we will search for a second-order method of the form

$$\begin{bmatrix} A & U \\ B & V \end{bmatrix},$$

with

$$A = \begin{bmatrix} \lambda & 0 \\ a_{21} & \lambda \end{bmatrix}, \qquad U = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}, \qquad V = \begin{bmatrix} v_1 & v_2 \\ v_1 & v_2 \end{bmatrix}.$$

As it happens, it is possible to choose $c = [0, 1]^T$ and then determine the other free parameters so that this method has exactly the same linear stability as the diagonally-implicit Runge–Kutta method

$$\begin{array}{c|cc} \frac{2-\sqrt{2}}{2} & \frac{2-\sqrt{2}}{2} & 0 \\ 1 & \frac{\sqrt{2}}{2} & \frac{2-\sqrt{2}}{2} \\ \hline & \frac{\sqrt{2}}{2} & \frac{2-\sqrt{2}}{2} \end{array}$$

In fact the method is given by the matrix

$$\left[\begin{array}{cc|cc} \frac{2-\sqrt{2}}{2} & 0 & 1 & 0 \\ \frac{6+2\sqrt{2}}{7} & \frac{2-\sqrt{2}}{2} & 0 & 1 \\ \hline \frac{73-34\sqrt{2}}{28} & \frac{-5+4\sqrt{2}}{4} & \frac{3-\sqrt{2}}{2} & \frac{-1+\sqrt{2}}{2} \\ \frac{87-48\sqrt{2}}{28} & \frac{-45+34\sqrt{2}}{28} & \frac{3-\sqrt{2}}{2} & \frac{-1+\sqrt{2}}{2} \end{array}\right].$$

To obtain dense output, we proceed as for the type-2 method discussed in Section 6. It is now found that $\tilde{B}$ and $\tilde{V}$ take the forms

$$\tilde{B}=\begin{bmatrix} \frac{73-34\sqrt{2}}{28}+\frac{40-31\sqrt{2}}{28}g & \frac{-1+2\sqrt{2}}{4}+\frac{-4+3\sqrt{2}}{4}g \\ \frac{9-11\sqrt{2}}{14}g & 1+\frac{-1+\sqrt{2}}{2}g \\ -1+\frac{5-24\sqrt{2}}{49}g & 1+\frac{-1+2\sqrt{2}}{7}g \end{bmatrix},$$

$$\tilde{V}=\begin{bmatrix} \frac{3-\sqrt{2}}{2}+\frac{2-\sqrt{2}}{2}g & \frac{-1+\sqrt{2}}{2}+\frac{-2+\sqrt{2}}{2}g \\ g & -g \\ \frac{6+2\sqrt{2}}{7}g & -\frac{6+2\sqrt{2}}{7}g \end{bmatrix}.$$

An analysis of variable stepsize stability both for zero and infinity, points to the choice $g=-\frac{1}{14}(30-3\sqrt{2})$. With this value, for which

$$\tilde{B}=\begin{bmatrix} \frac{-26+41\sqrt{2}}{28} & \frac{62-37\sqrt{2}}{28} \\ \frac{-48+51\sqrt{2}}{28} & \frac{64-33\sqrt{2}}{28} \\ \frac{-20+15\sqrt{2}}{14} & \frac{20-9\sqrt{2}}{14} \end{bmatrix}, \quad \tilde{V}=\begin{bmatrix} \frac{-12+11\sqrt{2}}{14} & \frac{26-11\sqrt{2}}{14} \\ \frac{-30+3\sqrt{2}}{14} & \frac{30-3\sqrt{2}}{14} \\ -\frac{12+3\sqrt{2}}{7} & \frac{12+3\sqrt{2}}{7} \end{bmatrix},$$

stability is assured for any stepsize ratio up to doubling of stepsize.

8. DIMSIMs of types 3 and 4

We consider two examples of methods suitable for a parallel environment. In each case $c=[0, 1]^{\mathrm{T}}$ and the order is 2. The methods are

$$\left[\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline -\frac{3}{8} & -\frac{3}{8} & -\frac{3}{4} & \frac{7}{4} \\ -\frac{7}{8} & \frac{9}{8} & -\frac{3}{4} & \frac{7}{4} \end{array}\right]$$

and

$$\left[\begin{array}{cc|cc} \frac{3-\sqrt{3}}{2} & 0 & 1 & 0 \\ 0 & \frac{3-\sqrt{3}}{2} & 0 & 1 \\ \hline \frac{18-11\sqrt{3}}{4} & \frac{-12+7\sqrt{3}}{4} & \frac{3-2\sqrt{3}}{2} & \frac{-1+2\sqrt{3}}{2} \\ \frac{22-13\sqrt{3}}{4} & \frac{-12+9\sqrt{3}}{4} & \frac{3-2\sqrt{3}}{2} & \frac{-1+2\sqrt{3}}{2} \end{array}\right].$$

While it is not possible to achieve the Runge–Kutta type of linear stability as for the methods in Sections 6 and 7, it is at least possible to obtain reasonably good stability regions. In particular, the type-3 method given above has the stability polynomial

$$p(w, z) = \det(wI - M(z)) = \det\bigl(wI - V - zB(I - zA)^{-1}\bigr),$$

given by

$$p(w, z) = w^2 - \bigl(1 + \tfrac{3}{4}z\bigr)w - \bigl(\tfrac{1}{4}z + \tfrac{3}{4}z^2\bigr)$$

and this yields a satisfactory stability region containing, for example, the real interval $[-\frac{4}{3}, 0]$.

For the given type-4 method, the stability polynomial is given by

$$p(w, z) = \left(1 - \frac{3-\sqrt{3}}{2}z\right)^2 w^2 - \left(1 - \frac{3-\sqrt{3}}{2}z\right)w + \frac{1-\sqrt{3}}{2}z.$$

This gives an A-stable stability region with the additional property that the spectral radius of the stability matrix is 0 for $z = \infty$.

9. Further developments

In this introduction to DIMSIMs, it has not been possible to convey a full appreciation of the potential of these methods. In particular, further examples of type-4 methods have been derived and these show that full parallelism along with A-stability is possible for quite high orders. It is intended that a discussion of type-3 and type-4 methods will be presented in a later paper.

For these methods, as for type-1 and type-2 methods, it seems to be a straightforward matter to provide dense output, stepsize variation, and local error estimation at no significant cost and later papers will give more details on these matters. The feature used in Section 6 and 7 of selecting from the family of possible type-1 and type-2 methods, special cases for which linear stability is identical with that for corresponding Runge–Kutta methods seems a reasonable design choice. It is not yet known for what orders this is possible but it is certainly true that methods generalizing the method of Section 6 exist for orders 3 and 4.

Even if increasingly high orders along with a Runge–Kutta-type stability behaviour are not possible, good methods may still exist for which some more general stability polynomials occurs.

Finally, although detailed numerical evidence is not included in this paper, some encouraging computational results have been found. These are for a simple index-1 differential algebraic equation where loss of stage order would be a serious difficulty if it occurred. The method used had the values $r = s = 2$ and is A-stable of order $p = 3$. Very clear-cut order-3 behaviour was observed whereas for a third-order DIRK method with exactly the same stability properties, the observed order was only 2.

Acknowledgement

A number of people have patiently listened to me talking about DIMSIM methods, have asked searching questions and made helpful comments. I am very grateful. In particular, I wish to thank Zdzisław Jackiewicz who has taken a close interest in this work and made a number of suggestions leading to improvements in the text of this paper. The comments by the referees have been of great value to me and I express my gratitude for their assistance.

References

[1] R. Alexander, Diagonally implicit Runge–Kutta methods for stiff ODEs, *SIAM J. Numer. Anal.* 14 (1977) 1006–1021.

[2] K. Burrage and J.C. Butcher, Non-linear stability of a general class of differential equation methods, *BIT* 20 (1980) 185–203.

[3] J.C. Butcher, On the convergence of numerical solutions to ordinary differential equations, *Math. Comp.* 20 (1966) 1–10.

[4] J.C. Butcher, The order of numerical methods for ordinary differential equations, *Math. Comp.* 27 (1973) 793–806.

[5] J.C. Butcher, Order conditions for a general class of numerical methods for ordinary differential equations, in: *Topics in Numerical Analysis* (Academic Press, New York, 1973).

[6] J.C. Butcher, The order of differential equation methods, in: Lecture Notes in Mathematics 362 (Springer, New York, 1973).

[7] J.C. Butcher, On the implementation of implicit Runge–Kutta methods, *BIT* 16 (1976) 237–240.

[8] J.C. Butcher, *The Numerical Analysis of Ordinary Differential Equations: Runge–Kutta and General Linear Methods* (Wiley, Chichester, England, 1987).

[9] A. Iserles and S.P. Nørsett, On the theory of parallel Runge–Kutta methods, *IMA J. Numer. Anal.* 10 (1990) 463–488.