

The construction of order 4 DIMSIMs for ordinary differential equations.

W. M. Wright

*Department of Mathematics, The University of Auckland,
Private Bag 92019, Auckland, New Zealand.*

Diagonally Implicit Multistage Integration methods (DIMSIMs) of type 1 and 2 have considerable potential as numerical algorithms for ordinary differential equations. The aim of this paper is to construct such methods of order 4 of type 1 and 2, which completes the set for orders 1 through 8.

Keywords: Ordinary differential equations, Runge-Kutta stability, variable stepsize.

AMS Subject classification: 65L05.

1. Introduction

Even though general linear methods were proposed more than 35 years ago, it has proved very difficult to find essentially new practical methods in this large class. In [1], a class of practical general linear methods known as DIMSIMs was introduced. In order to construct these methods, both stage order and order conditions must be satisfied. This leaves free parameters to obtain desirable properties such as Runge-Kutta stability. These requirements result in having to solve complicated nonlinear systems. Methods with a particular special choice of the defining parameters have been derived for all orders up to 8 excluding order 4, see [1], [5], [6] and [7]. This paper shows how order 4 methods can be derived.

The organization of this paper is as follows. In the next section a short introduction to DIMSIMs will be covered motivating the special choice of methods. In section 3 a transformation is discussed which leads to a derivation of the order 4 methods. In section 4 a brief introduction to representing DIMSIMs in a Nordsieck formulation is included. This is required when implementing these methods in variable stepsize code. The order 4 Nordsieck methods are also included. The last section contains some conclusions.

2. An introduction to DIMSIMs

An autonomous ordinary differential equation has the form

$$\begin{aligned} y'(x) &= f(y(x)), & x \in [x_0, x_N] \subset \mathcal{R}, \\ y(x_0) &= y_0, \end{aligned} \tag{1}$$

where $f : \mathcal{R}^m \rightarrow \mathcal{R}^m$ and m is the dimensionality of the system. A DIMSIM used for the numerical solution of (1) is given by

$$\begin{aligned} Y^{[n]} &= h(A \otimes I_m)f(Y^{[n]}) + (U \otimes I_m)y^{[n-1]}, \\ y^{[n]} &= h(B \otimes I_m)f(Y^{[n]}) + (V \otimes I_m)y^{[n-1]}, \end{aligned} \tag{2}$$

where $n = 1, 2, \dots, N$, $Nh = x_N - x_0$, and h is the stepsize. These methods are characterized by a string of numbers $pqrst$, where p is the order, q is the stage order, r is the number of input and output approximations, s is the number of the internal stages and t is the type of method. The type of method determines the intended application, whether the problem is non-stiff or stiff and whether the computer architecture is sequential or parallel.

The A matrix determines the type of method and is strictly lower triangular for type 1 methods and is lower triangular with constant diagonal terms for type 2 methods. The most convenient choice for U is $U = I$. This choice is valid since a transformation can be found to force $U = I$ while keeping the structure of the other matrices unchanged. The V matrix determines stability, good zero stability and power-boundedness requires a spectrum from $\{0, 1\}$. An appropriate condition for this is $V = ev^T$, where $e, v \in \mathcal{R}^r$, such that $v^T e = 1$. This condition also guarantees that V is of rank 1.

The vector $y^{[n]}$ denotes the computed quantities at step n , which also represent the incoming values for step $n + 1$. Then, $y^{[n]}$ is an approximation of order p to a linear combination of scaled derivatives $h^k y^{(k)}(x_n)$. The vector $Y^{[n]}$ denotes the internal stage values, where $Y^{[n]} \approx y(x_{n-1} + ch) + O(h^q)$. Also, the vector $f(Y^{[n]})$ denotes the stage derivatives.

In [1], it is shown that if the order p and the stage order q are equal then the B matrix satisfies both order conditions, and is given by

$$B = B_0 - AB_1 - VB_2 + VA, \tag{3}$$

where the matrices B_0 , B_1 and B_2 depend only on the abscissae vector c .

The stability behaviour of these methods is considered using the standard linear test problem

$$y' = \xi y,$$

where ξ is a complex parameter. If method (2) is applied to this test problem, then the stability matrix is given by

$$M(z) = V + zB(I - zA)^{-1}U,$$

where $z = h\xi$. Because the method has order p , it follows that $\exp(z) + O(z^{p+1})$ is an eigenvalue of $M(z)$. Therefore a general linear method has Runge-Kutta stability if the stability matrix has all except one of its eigenvalues identically equal to zero.

If the characteristic polynomial of $M(z)$, known as the stability function, has the special form

$$p(w, z) = \det(wI - M(z)) = w^{r-1}(w - R(z)),$$

where $R(z)$ is an order p approximation to $\exp(z)$ see [5], then the method is said to possess ‘‘Runge-Kutta stability’’. As a means of direct comparison with traditional Runge-Kutta methods, Runge-Kutta stability is required.

Due to the structure of the V matrix the characteristic polynomial has the form

$$p(w, z) = p_0(z)w^r - p_1(z)w^{r-1} + \cdots + (-1)^{r-1}p_{r-1}(z)w + (-1)^r p_r(z),$$

where

$$\begin{aligned} p_0(z) &= 1 + p_{01}z + \cdots + p_{0\ s-1}z^{s-1} + p_{0\ s}z^s, \\ p_1(z) &= 1 + p_{11}z + \cdots + p_{1\ s-1}z^{s-1} + p_{1\ s}z^s, \\ p_2(z) &= p_{21}z + \cdots + p_{2\ s-1}z^{s-1} + p_{2\ s}z^s, \\ &\vdots \\ p_r(z) &= p_{r\ s-1}z^{r-1} + \cdots + p_{r\ s}z^s. \end{aligned}$$

The coefficients p_{ij} are complicated expressions in terms of the free parameters of the A and V matrices. For the remainder of this paper, it is assumed that $p = q = r = s$. Then Runge-Kutta stability is equivalent to

$$p_{ij} = 0, \quad i = 2, 3, \dots, r, \quad j = i - 1, i, \dots, s.$$

Since $r = s$, this system of $(s-1)(s+2)/2$ nonlinear equations with $(s-1)(s+2)/2$ unknowns from the matrices A and V completely characterize the methods. The components of the vector c are usually chosen to be uniformly distributed on the interval $[0, 1]$. The coefficients of these methods for orders 2 and 3 can be generated and solved symbolically using, for example, Mathematica or Maple see [1] and [5]. For orders greater than 4 it is not possible to represent the equations symbolically and an alternative technique is required. A variant of the Fourier series approach was used in [6]. The coefficients were obtained numerically using a modification to the Levenberg-Marquardt algorithm implemented in `lmdif.f` and `lmdef.f` from Minpack. The subroutines used to find methods of order 5 and 6 were not adequate to find methods of order 7 and 8 to high enough accuracy in a reasonable time. In [7], an optimization method based on a particular variable-model trust-region least squares algorithm was effective. The minimization algorithms used were DN2G and DN2GB and are available in the NETLIB/PORT library.

3. Transformations

To obtain a method of order 4, symbolic packages such as Maple and Mathematica can represent the stability function in symbolic form, but are unable to directly give approximations to their solutions. A homotopy approach was used in [5], to find a method where the abscissae vector was $c = [\frac{1}{8}, \frac{3}{8}, \frac{5}{8}, \frac{7}{8}]^T$. To obtain the method with abscissae vector $c = [0, \frac{1}{3}, \frac{2}{3}, 1]^T$, a transformation of the matrices A , V , B_0 , B_1 and B_2 , allows Maple to represent the system and solve it numerically. The required transformation was found in [2] and results in a much simpler representation of the stability function. This transformation is included for completeness.

Let T be the $p \times p$ matrix with (i, j) elements given by

$$t_{ij} = \begin{cases} \prod_{k < j} (c_i - c_k), & \text{if } j \leq i, \\ 0, & \text{if } j > i, \end{cases}$$

and the elements of the inverted matrix T^{-1} are given by

$$t_{ij}^{(-1)} = \begin{cases} \prod_{\substack{1 \leq k \leq i \\ k \neq j}} (c_j - c_k)^{-1}, & \text{if } j \leq i, \\ 0, & \text{if } j > i. \end{cases}$$

Let the polynomials $g(c_1 c_2), \dots, g(c_1 c_2 \dots c_p)$ denote divided differences. Therefore,

$$\begin{bmatrix} g(c_1) \\ g(c_2) \\ \vdots \\ g(c_p) \end{bmatrix} = T \begin{bmatrix} g(c_1) \\ g(c_1 c_2) \\ \vdots \\ g(c_1 c_2 \dots c_p) \end{bmatrix},$$

for any polynomial g in the vector space of polynomials of degree less than p . For the case when $p = 4$

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & \frac{1}{3} & 0 & 0 \\ 1 & \frac{2}{3} & \frac{2}{9} & 0 \\ 1 & 1 & \frac{2}{3} & \frac{2}{9} \end{bmatrix}, \quad T^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ \frac{9}{2} & -9 & \frac{9}{2} & 0 \\ -\frac{9}{2} & \frac{27}{2} & -\frac{27}{2} & \frac{9}{2} \end{bmatrix}.$$

The transformed matrices are given by $\hat{A} = T^{-1}AT$, $\hat{V} = T^{-1}VT$, $\hat{B}_0 = T^{-1}B_0T$, $\hat{B}_1 = T^{-1}B_1T$ and $\hat{B}_2 = T^{-1}B_2T$. Thus the relation (3) which satisfies the order conditions transforms to $\hat{B} = \hat{B}_0 - \hat{A}\hat{B}_1 - \hat{V}\hat{B}_2 + \hat{V}\hat{A}$.

The matrix \hat{A} is still lower triangular and the matrix \hat{V} is of the form

$$\hat{V} = \begin{bmatrix} 1 & \hat{v}_2 & \hat{v}_3 & \dots & \hat{v}_p \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}.$$

However, the most important simplification occurs in the matrices \hat{B}_0 , \hat{B}_1 and \hat{B}_2 . It is proved in [2], that \hat{B}_0 and \hat{B}_2 are upper Hessenberg and \hat{B}_1 is upper triangular.

In order to find the coefficient matrices, first transform the A and V matrices. After appropriately relabelling the non-zero elements, the other matrices are then transformed. The resulting stability function has the same number of nonlinear equations but in a much simpler form. This system of nonlinear equations is then solved numerically using the `fsolve.ms` command from Maple.

This command takes the equations to be solved and a region of convergence for each unknown. Suitably chosen convergence regions lead to the convergence of the solution. The required method is then found by calculating the original non-zero elements of the A and V matrices from the relabelled elements.

The type 1 method found in this way, has coefficient matrices

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.3739348246 & 0 & 0 & 0 \\ 0.2949848977 & 0.4816828233 & 0 & 0 \\ -0.6903740089 & 2.2712602977 & -0.2257249932 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 2.9444524372 & -6.1678663794 & 4.9955647402 & -1.0248304881 \\ 2.8194524372 & -5.6539774905 & 4.1761202957 & -0.6348764238 \\ 2.5372463716 & -4.6501532283 & 2.9967727914 & -0.2465466791 \\ 1.1878127621 & -0.2420326723 & -1.2663733375 & 0.7127522620 \end{bmatrix},$$

$$v^T = \begin{bmatrix} -18.3637007103 & 47.9911902596 & -32.4937789808 & 3.8662894315 \end{bmatrix}.$$

The type 2 method found in this way, has coefficient matrices

$$A = \begin{bmatrix} 0.5728160600 & 0 & 0 & 0 \\ 0.1502207502 & 0.5728160600 & 0 & 0 \\ 0.5951580814 & -0.2663280726 & 0.5728160600 & 0 \\ 1.7717286221 & -1.6423444439 & 0.3914732019 & 0.5728160600 \end{bmatrix},$$

$$B = \begin{bmatrix} 13.1531193538 & -25.4515984981 & 16.0299843903 & -3.7980057648 \\ 13.6009354138 & -27.2289738493 & 18.6474363059 & -4.9027858061 \\ 14.2891666323 & -29.5340826633 & 20.9994482658 & -5.4831960958 \\ 15.6798283176 & -33.7506969112 & 24.4071044651 & -5.9235937704 \end{bmatrix},$$

$$v^T = \begin{bmatrix} 15.6150365914 & -46.9672685076 & 41.2900821542 & -8.9378502380 \end{bmatrix}.$$

These methods have an equally spaced abscissae vector and Runge-Kutta stability function. The error constants are $1/120$ for the type 1 method and $\mathcal{L}_4(1/\lambda)\lambda^4$ for the type 2 method, see [8], where \mathcal{L}_4 is the fourth degree Laguerre polynomial. The value of λ on the diagonal of A is chosen as the second zero of \mathcal{L}_4 which guarantees L-stability.

4. Nordsieck representation

Stepsize change is achieved by forcing the vector of incoming and outgoing stages to directly approximate to order p , a modified Nordsieck vector. This vector requires $p + 1$ derivative approximations including the zero derivative.

This representation was introduced in [3], which leads to a process which is zero stable for any stepsize pattern. Changing stepsize will be accomplished by a simple rescaling of the vector of external approximations. Consider a step from x_{n-1} to x_n with a stepsize h_n . The modified method takes the form

$$\begin{aligned} Y^{[n]} &= h(A \otimes I_m)f(Y^{[n]}) + (\tilde{U}D(\delta_n) \otimes I_m)\tilde{y}^{[n-1]}, \\ \tilde{y}^{[n]} &= h(\tilde{B} \otimes I_m)f(Y^{[n]}) + (\tilde{V}D(\delta_n) \otimes I_m)\tilde{y}^{[n-1]}, \end{aligned} \quad (4)$$

where $\delta_n = h_n/h_{n-1}$ and

$$D(\delta_n) = \text{diag} \left(1 \ \delta_n \ \delta_n^2 \ \cdots \ \delta_n^p \right).$$

The components $\tilde{y}_k^{[n]}$ are order p approximations to the scaled derivatives $h_n^{k-1}y^{(k-1)}(x_n)$. The zero stability property of the method (4) is determined by the eigenvalues of the matrix $\tilde{V}D(\delta_n)$. Due to the structure of the matrices \tilde{V} and $D(\delta_n)$, $\tilde{V}D(\delta_n)$ has one eigenvalue with magnitude one and a zero eigenvalue with multiplicity p , for any value of δ_n . Thus the modified method is zero stable for any choice of variable mesh.

The structure of the A matrix is consistent with method (2). However, since the Nordsieck vector has one more component than the original formulation, the matrices \tilde{U} , \tilde{B} and \tilde{V} must have size $s \times (p+1)$, $(p+1) \times s$ and $(p+1) \times (p+1)$ respectively. These matrices can be derived uniquely from the A , B , U and V matrices and the vector c . Let C denote the Vandermonde matrix

$$C = \begin{bmatrix} e & c & \cdots & c^{p-1} \end{bmatrix},$$

and define the columns of L by

$$L_k = (k-1)! \left(\sum_{j=0}^k \frac{e_{j+1}}{(k-j)!} - \tilde{V}e_{k+1} \right), \quad k = 1, 2, \dots, p,$$

where e_i , $i = 1, 2, \dots, p+1$, is the canonical basis in \mathcal{R}^{p+1} . In [3], it is proved that the Nordsieck representation of DIMSIMs (2) can be computed by the relations

$$\tilde{U} = W, \quad \tilde{B} = LC^{-1}, \quad \tilde{V} = e_1 \begin{bmatrix} 1 & v^T \alpha_1 & \cdots & v^T \alpha_p \end{bmatrix},$$

where

$$\alpha_k = \frac{c^k}{k!} - A \frac{c^{k-1}}{(k-1)!}, \quad k = 0, 1, \dots, p,$$

and the matrix W is given by

$$W = \begin{bmatrix} \alpha_0 & \alpha_1 & \cdots & \alpha_p \end{bmatrix}.$$

The Nordsieck representation for the type 1 method is

$$\tilde{B} = \begin{bmatrix} 2.9444524372 & -6.1678663794 & 4.9955647402 & -1.0248304881 \\ 0 & 0 & 0 & 1 \\ -1 & \frac{9}{2} & -9 & \frac{11}{2} \\ -9 & 36 & -45 & 18 \\ -27 & 81 & -81 & 27 \end{bmatrix},$$

$$\tilde{U} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 - 0.0406014913 & 0.0555555555 & 0.0061728395 & 0.0005144033 \\ 1 - 0.1100010543 & 0.0616612811 & 0.0226225592 & 0.0052571019 \\ 1 - 0.3551612955 & -0.1066034371 & 0.0906466486 & 0.0387934546 \end{bmatrix},$$

$$\tilde{v}^T = \begin{bmatrix} 1 & 0.2526796902 & 0.2504094544 & -0.0883843438 & 0.0038504422 \end{bmatrix}.$$

The Nordsieck representation for the type 2 method is

$$\tilde{B} = \begin{bmatrix} 13.1531193538 & -25.4515984981 & 16.0299843903 & -3.7980057648 \\ 0 & 0 & 0 & 1 \\ -1 & \frac{9}{2} & -9 & \frac{11}{2} \\ -9 & 36 & -45 & 18 \\ -27 & 81 & -81 & 27 \end{bmatrix},$$

$$\tilde{U} = \begin{bmatrix} 1 - 0.5728160600 & 0 & 0 & 0 \\ 1 - 0.3897034769 & -0.1353831311 & 0.0256502749 & -0.0030214983 \\ 1 - 0.2349794022 & -0.0708791269 & -0.0631137377 & -0.0184127597 \\ 1 - 0.0936734402 & 0.2136499534 & -0.1154940502 & -0.0629967580 \end{bmatrix},$$

$$\tilde{v}^T = \begin{bmatrix} 1 & 0.4936844588 & 1.5223996106 & -0.3689795400 & -0.0552972506 \end{bmatrix},$$

where \tilde{v}^T is the first and only non-zero row of \tilde{V} . It is shown in [3], that the eigenvalues and the linear stability properties of the method remain unchanged for constant stepsizes through the augmentation process. Consequently, the Nordsieck methods have the same Runge-Kutta stability functions and error constants as the original methods. The type 2 method is still L-stable.

5. Concluding remarks

This paper has shown the derivation of order 4 DIMSIMs for the application of non-stiff and stiff problems in a sequential computing environment. The methods were chosen so that $p = q = r = s$ and the abscissae vector was equally spaced in the interval $[0, 1]$. There is evidence to suggest these methods have potential for practical use, see [3], [4] and [5]. These methods were found using Maple after the application of a suitably chosen transformation. The methods were given in both original and Nordsieck representations. The type 1 and 2 methods have the same stability regions as an explicit Runge-Kutta and a DIRK method respectively. The type 2 method was chosen to have both A-stability and L-stability properties. This completes the set of methods for orders 1 through 8. In [4], a variable order code is implemented using the suite of methods for orders 1 through 8 which includes the method of type 1 constructed in this paper. This code performs well against `ode45.m` if the order is restricted to 4 or 5 and the reader is referred to this paper for numerical evidence.

References

- [1] J.C. Butcher, Diagonally implicit multistage integration methods, *Appl. Numer. Math.* 11, (1993), 347-363.
- [2] J.C. Butcher, A transformation for the analysis of DIMSIMs, *BIT* 11, (1993), 347-363
- [3] J.C. Butcher, P. Chartier and Z. Jackiewicz, Nordsieck representation of DIMSIMs, *Numer. Alg.* 16, (1997), 209-230.
- [4] J.C. Butcher, P. Chartier and Z. Jackiewicz, Experiments with a variable-order type 1 DIMSIM code, *Numer. Alg.* 22, (1999), 237-261.
- [5] J.C. Butcher and Z. Jackiewicz, Construction of diagonally implicit general linear methods of type 1 and 2 for ordinary differential equations, *Appl. Numer. Math.* 21, (1996), 385-415.
- [6] J.C. Butcher and Z. Jackiewicz, Construction of high order DIMSIMs for ordinary differential equations, *Appl. Numer. Math.* 27, (1998), 1-12.
- [7] J.C. Butcher, Z. Jackiewicz and H.D. Mittelmann, A nonlinear optimization approach to the construction of general linear methods of high order, *Comp. Appl. Math.* 81, (1997), 181-196.
- [8] E. Hairer, and G. Wanner, *Solving Ordinary Differential Equations, II. Stiff and Differential Algebraic Problems*, (Springer, Berlin, 1996).