

# A nonlinear optimization approach to the construction of general linear methods of high order

J.C. Butcher\*, Z. Jackiewicz<sup>†</sup> and H.D. Mittelmann<sup>‡</sup>

March 4, 1997

**Abstract.** We describe the construction of diagonally implicit multistage integration methods of order and stage order  $p = q = 7$  and  $p = q = 8$  for ordinary differential equations. These methods were obtained using state-of-the-art optimization methods, particularly variable-model trust-region least-squares algorithms.

**Key Words.** General linear method, ordinary differential equation, A-stability, L-stability, least-squares minimization.

**AMS Subject Classification.** 65L05, 65L20.

---

\*Department of Mathematics and Statistics, The University of Auckland, Private Bag 92019, Auckland, New Zealand. The work of this author was assisted by the Marsden Fund of New Zealand.

<sup>†</sup>Department of Mathematics, Arizona State University, Tempe, Arizona 85287. The work of this author was partially supported by the National Science Foundation under grant NSF DMS-9208048.

<sup>‡</sup>Department of Mathematics, Arizona State University, Tempe, Arizona 85287. The work of this author was partially supported by the National Science Foundation under grant NSF DMS-9403716

## 1 Introduction

In the recent papers [6], [7], and [9] we described the construction of a new class of diagonally implicit multistage integration methods (DIMSIMs) for ordinary differential equations (ODEs). These methods are special cases of general linear methods and have considerable potential for efficient implementation [8]. To construct such methods we impose the appropriate order and stage order conditions and then try to choose the remaining free coefficients to obtain some desirable stability properties. We are aiming at large regions of absolute stability in the explicit cases and at A-stability and L-stability in the implicit cases. These stability requirements lead, in principle, to large systems of nonlinear equations which we attempted to generate and solve using many different approaches. For low orders ( $p \leq 3$  and in some cases for  $p = 4$ ) this was successfully accomplished with the aid of symbolic manipulation packages such as MATHEMATICA or MAPLE [6], [7]. For moderate orders ( $p = 4$ ) the resulting systems of nonlinear equations were generated by symbolic manipulation software and then solved numerically with the aid of subroutines based on continuation methods from PITCON, ALCON, and HOMPACT [7]. For higher orders ( $p = 5$  and  $p = 6$ ) these nonlinear systems were generated by the algorithm based on least squares minimization. The preliminary version of this algorithm was described in [8] and resulted in an overdetermined system of nonlinear equations for the coefficients of the method. This algorithm was further refined in [9] where we were able to reduce the number of nonlinear equations to match exactly the number of unknown coefficients. These equations were obtained by a variant of the Fourier series method. These systems were then solved with the aid of subroutines `lmdif.f` and `lmdcr.f` from MINPACK. These subroutines minimize the sum of squares of nonlinear functions by a modification of the Levenberg-Marquardt algorithm [12]. Examples of explicit and implicit DIMSIMs of order  $p = 5$  and  $p = 6$  constructed by the above algorithm are presented in [9].

For still higher orders ( $p = 7$  and  $p = 8$ ) the subroutines based on the Levenberg-Marquardt algorithm are not powerful enough to solve the corresponding systems of nonlinear equations to a high accuracy in a reasonable time. To derive such methods we had to use more efficient optimization algorithms and the algorithm based on an improved version of NL2SOL [11] was able to do the job.

The organization of this paper is as follows. In the next section we give a brief introduction to DIMSIMs with  $p = q = r = s$ , where  $p$  is the order,  $q$  is the stage order,  $r$  is the number of external stages, and  $s$  is the num-

ber of internal stages. In Section 3 we review the construction of explicit and implicit DIMSIMs whose stability regions correspond to given stability functions chosen in advance to obtain favorable stability properties. In Section 4 we describe the construction of A-stable and L-stable generalized approximations to the exponential function. In Section 5 the optimization methods utilized are sketched and some details on their use are given. In Sections 6 the examples of DIMSIMs of type 1 and type 2 are presented. Finally in Section 8 some concluding remarks are made.

## 2 A short introduction to DIMSIMs

Given the vector  $c = [c_1, \dots, c_s]^T$  and the coefficient matrices  $A = [a_{ij}]$ ,  $U = [u_{ij}]$ ,  $B = [b_{ij}]$ , and  $V = [v_{ij}]$ , the DIMSIMs for the numerical solution of ODEs

$$\begin{cases} y'(x) = f(y(x)), & x \in [x_0, X], \\ y(x_0) = y_0, \end{cases} \quad (2.1)$$

are defined by

$$\begin{cases} Y_i = h \sum_{j=1}^s a_{ij} f(Y_j) + \sum_{j=1}^r u_{ij} y_j^{[n-1]}, & i = 1, 2, \dots, s, \\ y_i^{[n]} = h \sum_{j=1}^s b_{ij} f(Y_j) + \sum_{j=1}^r v_{ij} y_j^{[n-1]}, & i = 1, 2, \dots, r, \end{cases} \quad (2.2)$$

$n = 0, 1, \dots, N$ ,  $Nh = X - x_0$ . Here,  $Y_i$ ,  $i = 1, 2, \dots, s$ , are internal approximations to  $y(x_{n-1} + hc_i)$ ,  $x_{n-1} = x_0 + (n-1)h$ , and  $y_i^{[n]}$ ,  $i = 1, 2, \dots, r$  are external stages.

These methods were introduced by J.C. Butcher in a recent paper [3]. There is some theoretical and practical evidence [6], [7], [8], and [9] that DIMSIMs with  $p = q = r = s$  and  $U = I$ , where  $I$  is the identity matrix of appropriate dimension, have the biggest potential for practical use. For this reason in what follows we will restrict our attention to only such methods.

It was demonstrated in [3] that (2.2) has order  $p$  equal to the stage order  $q$  if and only if

$$B = B_0 - AB_1 - VB_2 + VA, \quad (2.3)$$

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

$$\int_0^{1+c_i} l_j(x) dx, \quad l_j(1+c_i), \quad \int_0^{c_i} l_j(x) dx,$$

respectively, with

$$l_j(x) = \prod_{k \neq j} \frac{x - c_k}{c_j - c_k},$$

compare [3], [7], [9]. These matrices are uniquely determined by the vector  $c$  and can be easily calculated with the aid of symbolic manipulation software. This is a very convenient representation of the order conditions and once the coefficient matrices  $A$  and  $V$  are determined from appropriate stability requirements, the coefficient matrix  $B$  will always be computed by the formula (2.3).

We will consider explicit and implicit DIMSIMs corresponding to the lower triangular matrix  $A$  of the form

$$A = \begin{bmatrix} \lambda & 0 & 0 & \dots & 0 \\ a_{21} & \lambda & 0 & \dots & 0 \\ a_{31} & a_{32} & \lambda & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{s1} & a_{s2} & \dots & a_{s,s-1} & \lambda \end{bmatrix},$$

where  $\lambda = 0$  (type 1 methods) or  $\lambda > 0$  (type 2 methods). These methods are appropriate for nonstiff or stiff differential systems, respectively, in a sequential computing environment. Moreover, we will always assume that  $V$  is a rank one matrix given by

$$V = \begin{bmatrix} v_1 & v_2 & \dots & v_s \\ v_1 & v_2 & \dots & v_s \\ \vdots & \vdots & \ddots & \vdots \\ v_1 & v_2 & \dots & v_s \end{bmatrix},$$

with  $\sum_{i=1}^s v_i = 1$ . This choice guarantees that  $V$  is power bounded which is a necessary condition for convergence.

As explained in [3], [6], and [7] the stability properties of the method (2.2) are determined by the stability matrix

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

and the corresponding stability function

$$p(w, z) = \det(wI - M(z)),$$

where  $w$  and  $z$  are complex numbers. In what follows we will try to construct methods whose stability regions correspond to the functions  $p(w, z)$  which are chosen to possess some desirable stability properties. This process is briefly described in the next section.

### 3 Construction of DIMSIMs of type 1 and 2 with given stability function

Consider first DIMSIMs of type 1. The stability function  $p(w, z)$  of such methods has the form

$$p(w, z) = \sum_{k=0}^s (-1)^k p_k(z) w^{s-k},$$

where  $p_0(z) \equiv 1$  and

$$p_k(z) = \sum_{l=k-1}^s p_{kl} z^l$$

are polynomials of degree less than or equal to  $s$  whose coefficients  $p_{kl}$  depend on  $a_{ij}$  and  $v_i$ . We will try to compute  $a_{ij}$  and  $v_i$  so that the stability function will be equal to

$$p^*(w, z) = w^{s-1}(w - R(z)), \quad (3.1)$$

where

$$R(z) = \sum_{j=0}^s \frac{z^j}{j!}$$

is the approximation of order  $s$  to the exponential function  $\exp(z)$ . To this end we will choose the points  $w_\mu$ ,  $\mu = 1, 2, \dots, N_1$  and  $z_\nu$ ,  $\nu = 1, 2, \dots, N_2$  in the complex plane and then construct the objective function given by

$$f(a_{ij}, v_i) = \sum_{\mu=1}^{N_1} \sum_{\nu=1}^{N_2} |p(w_\mu, z_\nu) - p^*(w_\mu, z_\nu)|^2. \quad (3.2)$$

This function is then minimized using standard optimization techniques. The coefficients  $a_{ij}$  and  $v_i$  which correspond to the zero minimum value of  $f$  yield the desired DIMSIM of type 1.

The refinement of the above technique was examined in [9] which is based on the computation of the coefficients  $p_{kl}$  by a variant of the Fourier series method. Assuming that  $w_\mu$  and  $z_\nu$  are uniformly distributed on the unit circle and that  $N_1, N_2 \geq s + 1$  this leads to the system of  $(s - 1)(s + 2)/2$  nonlinear equations

$$\sum_{\mu=1}^{N_1} \sum_{\nu=1}^{N_2} w_\mu^{k-s} z_\nu^{-l} p(w_\mu, z_\nu) = 0, \quad (3.3)$$

$k = 2, 3, \dots, s$ ,  $l = k - 1, k, \dots, s$  for the  $(s - 1)(s + 2)/2$  unknown coefficients  $a_{ij}$ ,  $i = 2, 3, \dots, s$ ,  $j = 1, 2, \dots, i - 1$ , and  $v_i$ ,  $i = 1, 2, \dots, s - 1$  of the

method of type 1. The solution to this system can then be attempted by techniques for the numerical solution of nonlinear equations or least squares minimization. These techniques are discussed in Section 5.

Consider next type 2 methods. It was demonstrated in [9] that by making the substitutions

$$\hat{z} = \frac{z}{1 - \lambda z} \quad \text{and} \quad \hat{A} = A - \lambda I$$

we obtain the stability polynomial

$$\hat{p}(w, \hat{z}) = \sum_{k=0}^s (-1)^k \hat{p}_k(\hat{z}) w^{s-k}, \quad (3.4)$$

where  $\hat{p}_0(\hat{z}) \equiv 1$  and

$$\hat{p}_k(\hat{z}) = \sum_{l=k-1}^s \hat{p}_{kl} \hat{z}^l,$$

which has the same form as polynomials  $p_k(z)$  corresponding to type 1 methods.

In [9] we constructed implicit DIMSIMs of order  $p = 5$  and  $p = 6$  with stability polynomial of the form

$$\hat{p}^*(w, \hat{z}) = w^{s-1}(w - \hat{R}(\hat{z})),$$

where  $\hat{R}(\hat{z})$  is the stability function of the SDIRK method of order  $s$ . In this paper we will follow a different approach and we will attempt to construct methods with stability polynomial  $\hat{p}(w, \hat{z})$  of the form

$$\hat{p}^*(w, \hat{z}) = w^{s-2}(w^2 - \hat{p}_1^*(\hat{z})w + \hat{p}_2^*(\hat{z})), \quad (3.5)$$

where  $\hat{p}_1^*(\hat{z})$  and  $\hat{p}_2^*(\hat{z})$  are polynomials of degree less than or equal to  $s$ . These polynomials will be chosen in such a way that the corresponding methods are A-stable and L-stable. This process is described in the next section.

We have used a different approach than in [9] for the following reasons. It is known that the stability function of the SDIRK method of order  $p = 7$  cannot be A-stable for any  $\lambda$  (compare [16]) so that the approach of [9] would not lead to type 2 DIMSIMs which are A-stable. The situation is different for  $p = 8$  where  $\lambda = 0.23437316$  corresponds to the SDIRK method which is A-stable and L-stable (compare again [16]). However such a parameter  $\lambda$  is unique while the approach used with the function of the form (3.5) leads to an entire interval for the suitable parameter  $\lambda$ . Since we can only compute approximations to DIMSIMs with desired stability properties we

believe that the new approach to the construction of type 2 methods is more robust than the approach presented in [9].

The objective function  $\hat{f}(\hat{a}_{ij}, v_i)$  corresponding to type 2 methods takes the form

$$\hat{f}(\hat{a}_{ij}, v_i) = \sum_{\mu=1}^{N_1} \sum_{\nu=1}^{N_2} |\hat{p}(w_\mu, \hat{z}_\nu) - \hat{p}^*(w_\mu, \hat{z}_\nu)|^2, \quad (3.6)$$

where  $w_\mu, \mu = 1, 2, \dots, N_1$ , and  $\hat{z}_\nu, \nu = 1, 2, \dots, N_2$  are appropriately chosen points in the complex plane. Assuming again that  $w_\mu$  and  $\hat{z}_\nu$  are uniformly distributed on the unit circle and that  $N_1, N_2 \geq s + 1$  we obtain a system of  $(s - 1)(s + 2)/2$  equations

$$\begin{cases} \frac{1}{N_1 N_2} \sum_{\mu=1}^{N_1} \sum_{\nu=1}^{N_2} w_\mu^{2-s} \hat{z}_\nu^{-l} \hat{p}(w_\mu, \hat{z}_\nu) = \hat{p}_{2,l}^*, & l = 1, 2, \dots, s, \\ \sum_{\mu=1}^{N_1} \sum_{\nu=1}^{N_2} w_\mu^{k-s} \hat{z}_\nu^{-l} \hat{p}(w_\mu, \hat{z}_\nu) = 0, & l = k - 1, k, \dots, s, \end{cases} \quad (3.7)$$

$k = 3, 4, \dots, s$ , where  $\hat{p}_{2,l}^*$  is the coefficient of  $z^l$  in  $\hat{p}_2^*(z)$ .

## 4 Construction of highly stable generalized approximations to the exponential function

In this section we will describe the construction of A-stable and L-stable generalized approximations to  $\exp(z)$  [5]. We will look for approximations of the form

$$p^*(w, z) = (1 - \lambda z)^s w^s - p_1^*(z) w^{s-1} + p_2^*(z) w^{s-2}, \quad (4.1)$$

where  $p_1^*(z)$  and  $p_2^*(z)$  are polynomials of degree less than or equal to  $s$ . Let

$$p_k^*(z) = \sum_{l=k-1}^s p_{kl}^* z^l, \quad k = 1, 2.$$

Since  $p^*(w, z)$  corresponds to the method of order  $s$  it follows that [5]

$$p^*(\exp(z), z) = O(z^{s+1}), \quad (4.2)$$

which leads to the system of  $s + 1$  polynomial equations for the parameter  $\lambda$  and the coefficients  $p_{kl}^*$  of  $p_k^*(z)$ ,  $k = 1, 2$ . Assuming that

$$p_{1,l}^* = 0, \quad l = 5, 6, \dots, s,$$

$$p_{2,l}^* = 0, \quad l = s - 3, s - 2, \dots, s,$$

$s = 7$  or  $s = 8$  and solving the system corresponding to (4.2) we can express the remaining coefficients  $p_{1,l}^*$ ,  $l = 0, 1, 2, 3, 4$ , and  $p_{2,l}^*$ ,  $l = 1, 2, \dots, s - 4$ , in terms of  $\lambda$ . These expressions are not reproduced here. The parameter  $\lambda$  can then be chosen in such a way that the polynomial  $p^*(w, z)$  is A-stable. Once this is done the corresponding function  $p^*(w, z)$  will also be L-stable since  $p_{1,s}^* = p_{2,s}^* = 0$ .

The polynomial  $p^*(w, z)$  has a root  $w = 0$  of multiplicity  $s - 2$  and to assure A-stability the remaining two roots  $R_1^*(z)$  and  $R_2^*(z)$  should satisfy

$$|R_k^*(z)| \leq 1, \quad k = 1, 2, \quad (4.3)$$

for  $\text{Re}(z) \leq 0$ . It can be verified using the Schur theorem [18] that the condition (4.3) is satisfied for

$$0.25864444 \leq \lambda \leq 0.27688498$$

if  $p = 7$  and for

$$0.19799408 \leq \lambda \leq 0.20136462$$

if  $p = 8$ . We will choose  $\lambda = 13/50$  for  $p = 7$  and  $\lambda = 1/5$  for  $p = 8$ . The corresponding functions  $p^*(w, z)$  take the form (4.1) with

$$\begin{aligned} p_1^*(z) &= 1 - \frac{1022815846}{1708984375} z - \frac{7864050101}{68359375000} z^2 + \\ &\quad \frac{21301028013}{136718750000} z^3 - \frac{4289969757}{136718750000} z^4, \\ p_2^*(z) &= \frac{757102683}{3417968750} z + \frac{469409809}{68359375000} z^2 - \frac{3769899337}{410156250000} z^3 \end{aligned}$$

and

$$\begin{aligned} p_1^*(z) &= 1 - \frac{33929}{31250} z + \frac{74267}{437500} z^2 + \frac{102897}{2187500} z^3 - \frac{301687}{26250000} z^4, \\ p_2^*(z) &= -\frac{15179}{31250} z - \frac{146989}{437500} z^2 - \frac{556099}{6562500} z^3 - \frac{71741}{8750000} z^4, \end{aligned}$$

respectively. Making the substitution  $z = \hat{z}/(1 + \lambda\hat{z})$  the polynomial

$$\hat{p}^*(w, \hat{z}) = \frac{p^*(w, z)}{(1 - \lambda z)^s}$$

can be rewritten as

$$\hat{p}^*(w, \hat{z}) = w^s - \hat{p}_1^*(\hat{z})w^{s-1} + \hat{p}_2^*(\hat{z})w^{s-2} \quad (4.4)$$

with

$$\begin{aligned} \hat{p}_1^*(\hat{z}) &= 1 + \frac{4175071433}{3417968750}z + \frac{126776049293}{341796875000}z^2 + \frac{24844782161}{1708984375000}z^3 + \\ &\quad \frac{209188231309}{85449218750000}z^4 + \frac{5180883892327}{2136230468750000}z^5 - \\ &\quad \frac{304728451611597}{2136230468750000000}z^6 - \frac{1078437059539437}{13351440429687500000}z^7, \\ \hat{p}_2^*(\hat{z}) &= \frac{757102683}{3417968750}z + \frac{17207866799}{48828125000}z^2 + \frac{287549223877}{1281738281250}z^3 + \\ &\quad \frac{18699774431377}{256347656250000}z^4 + \frac{54099471240529}{4272460937500000}z^5 + \\ &\quad \frac{997732278309637}{915527343750000000}z^6 + \frac{173150891219683}{5006790161132812500}z^7 \end{aligned}$$

if  $p = 7$  and

$$\begin{aligned} \hat{p}_1^*(\hat{z}) &= 1 + \frac{16071}{31250}z - \frac{503707}{2187500}z^2 - \frac{333233}{1562500}z^3 - \\ &\quad \frac{7167059}{131250000}z^4 - \frac{1000631}{164062500}z^5 - \frac{1162677}{2734375000}z^6 - \\ &\quad \frac{253999}{5126953125}z^7 - \frac{223201}{58593750000}z^8, \\ \hat{p}_2^*(\hat{z}) &= -\frac{15179}{31250}z - \frac{2222487}{2187500}z^2 - \frac{29397383}{32812500}z^3 - \\ &\quad \frac{56506619}{131250000}z^4 - \frac{19919071}{164062500}z^5 - \frac{23524633}{1171875000}z^6 - \\ &\quad \frac{2641049}{1464843750}z^7 - \frac{27871967}{410156250000}z^8 \end{aligned}$$

if  $p = 8$ .

## 5 Least squares minimization

As was noted above, the determination of the coefficients of higher-order DIMSIMs is of increasing difficulty. When we attacked this problem for  $p > 6$ , first the approaches employed for  $p \leq 6$  were used again. Only the least squares approach based on (3.2) and (3.6) was producing coefficient vectors that had a somewhat though insufficiently small residual  $f$ . However, an excessive amount of function evaluations was necessary even to produce these unsatisfactory results. Next, a number of methods not utilized before were tried to minimize (3.2) and (3.6) or to solve the systems (3.3) and (3.7). Based on the experience gathered a choice was made to use the rather sophisticated algorithm available in DN2G and DN2GB. These are nonlinear least squares routines available in NETLIB/PORT, the public part of the PORT library [21]. They are new versions of the original code NL2SOL [10, 11]. A description of the improvements can be found in [1]. In particular, the least-squares problem is considered with additional bound-constraints on its variables as proposed in [14].

The least squares solution in NL2SOL was developed as a generalization and improvement of standard approaches such as the Levenberg-Marquardt in `lmdif.f` and `lmdcr.f` from MINPACK which we had used for  $p \leq 6$ . Specifically, section 8 in [10] contains a comparison with the latter. DN2G (without bound constraints) and DN2GB utilize adaptive quadratic modeling. Special problem structure is exploited by maintaining a secant approximation to the second-order part of the Hessian of the objective function  $f$ . The program switches adaptively between a Gauss-Newton and an augmented Hessian approximation where the Gauss-Newton steps are computed from a corrected seminormal equation approach. If we write any of the nonlinear least-squares functionals in the generic form

$$f(x) = \sum_{i=1}^N r_i(x)^2 = \frac{1}{2} R(x)^T R(x)$$

and denote the Jacobian of  $R$  by  $J$ , then the Hessian of  $f$  is

$$\nabla^2 f(x) = J(x)^T J(x) + \sum_{i=1}^N r_i(x) \nabla^2 r_i(x)$$

and the Gauss-Newton model at the iterate  $x = x_k$  is

$$\begin{aligned} q_k^G(x) &= \frac{1}{2} R(x_k)^T R(x_k) + (x - x_k)^T J(x_k)^T R(x_k) \\ &+ \frac{1}{2} (x - x_k)^T J(x_k)^T J(x_k) (x - x_k). \end{aligned}$$

In NL2SOL one adds an approximation to the difference between this and the standard quadratic Taylor model of Newton's method to obtain another model

$$\begin{aligned}
 q_k^S(x) &= \frac{1}{2}R(x_k)^T R(x_k) + (x - x_k)J(x_k)^T R(x_k) \\
 &+ \frac{1}{2}(x - x_k)^T [J(x_k)^T J(x_k) + S_k](x - x_k).
 \end{aligned}$$

To update  $S_k$  a straightforward modification of the Oren-Luenberger self-scaling technique [20] is used. Finally, the choice which of the above models to use is intimately related to the trust region approach utilized to pick  $\Delta x_k$  which has the form

$$\Delta x_k = (H_k + \lambda_k D_k^2)^{-1} \nabla f(x_k),$$

where  $H_k$  is the current approximation of the Hessian,  $D_k$  a diagonal scaling matrix, and  $\lambda_k \geq 0$  is chosen by the same procedure as in [19]. More details can be found in [11, 1]. The code is much larger and more complex than standard least squares programs but has proven to be very robust and at the same time reasonably efficient.

In a first phase of the solution process a search was performed executing a suitable maximal number of iterations, typically 1000–5000 for each of a small number (10-15) of starting points with coefficients uniformly distributed in  $[-1, 1]$ . Those points that had a small  $f$ -value and not too large components were subsequently improved. For the initial phase the problems (3.2) or (3.6) led in general to a more effective reduction of the residual. In the second phase, however, it occasionally happened that DN2G(B) terminated prematurely when applied to (3.2) or (3.6). Then, a perturbation that led to a continued decrease of the residual, was accomplished by switching between the functionals (3.2) or (3.6) and (3.3) or (3.7). This way, in all cases solutions with sufficiently small residuals could be found. The bound-constraints in DN2GB were mainly used to exclude solutions with unduly large components. Never was a final solution computed which had one of the bound constraints active.

First, the PORT routines DN2F(B) were used which employ a simple finite difference approximation of the Jacobian of  $f$ . They were found not to yield satisfactory results and instead a high-order numerical differentiation provided in DONLP2 [13] was used in the routines DN2G(B). Alternatives would have been to code the exact derivatives as in [14] or to use automatic differentiation, for example, ADIFOR [2]. The numerical differentiation in [13] uses sixth-order extrapolation, specifically Richardson extrapolation of

three values of the symmetric difference quotient with a relatively large stepsize. Since the maximum of seventh partial derivatives on which the optimal stepsize  $\delta$  depends is unknown, it is replaced by 1 in the formula for this stepsize, namely

$$\delta = 0.25 * \text{macheps}^{1/7}.$$

The gradient vector  $\nabla f = [\nabla f_1, \dots, \nabla f_N]^T$  of a function  $f(x)$  is then approximated through the following algorithm.

```

for  $i := 1(1)N$  do
   $\delta x = \delta(1 + |x_i|)$ ;
   $f_1 = f(x - \delta x e_i)$ ;  $f_2 = f(x + \delta x e_i)$ ;
   $f_3 = f(x - 2\delta x e_i)$ ;  $f_4 = f(x + 2\delta x e_i)$ ;
   $f_5 = f(x - 4\delta x e_i)$ ;  $f_6 = f(x + 4\delta x e_i)$ ;
   $s_1 = (f_2 - f_1)/(2\delta x)$ ;
   $s_2 = (f_4 - f_3)/(4\delta x)$ ;
   $s_3 = (f_6 - f_5)/(8\delta x)$ ;
   $\nabla f_i := s_1 + 0.4(s_1 - s_2) + (s_1 - 2s_2 + s_3)/45$ 
enddo

```

Here,  $e_i$  denotes the  $i$ -th unit vector.

This numerical differentiation procedure produced errors in the last significant solution components when compared with exact differentiation on a large number of optimization test problems solved by DONLP2 [13].

With the techniques described in this section solution to the least squares problems were obtained which satisfied quality criteria defined for the specific problem at hand and outlined in Section 6. While these solutions did not have zero residuals their quality as coefficients of high-order DIMSIM methods was deemed quite satisfactory and while attempts, for example, through use of extended precision computation to further decrease their residuals were made, they will not be reported here. The additional effort spent does not seem to be justified for our purposes.

## 6 Examples of DIMSIMs of type 1 and 2

We present below the examples of type 1 DIMSIMs of order  $p = 7$  and  $p = 8$ . As explained in Section 5 the coefficients  $a_{ij}$  and  $v_i$  were computed by the PORT library routines DN2G and DN2GB applied to (3.2) and (3.6) or (3.3) and (3.7) and then represented in the rational form by using the

MATHEMATICA function `Rationalize[x, dx]` with  $x = a_{ij}$  or  $x = v_i$  and  $dx = 10^{-16}$ . The consistency condition  $\sum_{i=1}^s v_i = 1$  has been lost because of the rational approximations and hence, in any practical use of the methods, one of the  $v_i$  should be found numerically in terms of the others to ensure that consistency is exactly maintained. The coefficient matrix  $B$  was then computed using the formula (2.3), where the matrices  $B_0$ ,  $B_1$ , and  $B_2$  correspond to the vectors  $c$  with components uniformly distributed on the unit interval  $[0, 1]$ , that is

$$c = \left[ 0 \quad \frac{1}{s-1} \quad \frac{2}{s-1} \quad \cdots \quad \frac{s-2}{s-1} \quad 1 \right]^T.$$

The matrix  $B$  is not displayed here.

To measure how well the derived methods approximate DIMSIMs with given stability properties we compared stability polynomials  $p(w, z)$  and  $\hat{p}(w, \hat{z})$  of the methods listed below with the stability polynomials  $p^*(w, z)$  given by (3.1) for type 1 methods and with  $\hat{p}^*(w, \hat{z})$  given by (3.5) for type 2 DIMSIMs. To be more specific, we monitored for type 1 methods the size of the coefficients of the polynomials  $p_k(z)$ ,  $k = 2, 3, \dots, s$ , and the relative errors of the coefficients  $p_{1,l}$  of the polynomial  $p_1(z)$ . These polynomials are defined in Section 3. This information is reflected in the following vectors

$$Abserr = \left[ \left( \sum_{l=1}^s p_{2,l}^2 \right)^{1/2} \quad \cdots \quad \left( \sum_{l=s-1}^s p_{s,l}^2 \right)^{1/2} \right]$$

and

$$Relerr = \left[ \frac{|p_{1,0} - p_{1,0}^*|}{p_{1,0}^*} \quad \cdots \quad \frac{|p_{1,s} - p_{1,s}^*|}{p_{1,s}^*} \right],$$

where  $p_{1,j}^* = 1/j!$ ,  $j = 0, 1, \dots, s$ . Similarly, for type 2 methods we monitored the size of the coefficients of the polynomials  $\hat{p}_k(\hat{z})$ ,  $k = 3, 4, \dots, s$ , and the relative errors of the coefficients  $\hat{p}_{1,l}$  and  $\hat{p}_{2,l}$  of the polynomials  $\hat{p}_1(\hat{z})$  and  $\hat{p}_2(\hat{z})$ . These polynomials are defined in Section 3. This information is reflected in the following vectors

$$Abserr = \left[ \left( \sum_{l=2}^s \hat{p}_{3,l}^2 \right)^{1/2} \quad \cdots \quad \left( \sum_{l=s-1}^s \hat{p}_{s,l}^2 \right)^{1/2} \right],$$

$$Relerr_1 = \left[ \frac{|\hat{p}_{1,0} - \hat{p}_{1,0}^*|}{|\hat{p}_{1,0}^*|} \quad \cdots \quad \frac{|\hat{p}_{1,s} - \hat{p}_{1,s}^*|}{|\hat{p}_{1,s}^*|} \right],$$

$$Relerr_2 = \left[ \frac{|\hat{p}_{2,1} - \hat{p}_{2,1}^*|}{|\hat{p}_{2,1}^*|} \quad \dots \quad \frac{|\hat{p}_{2,s} - \hat{p}_{2,s}^*|}{|\hat{p}_{2,s}^*|} \right],$$

where  $\hat{p}_{kl}^*$  are the coefficients of the polynomials  $\hat{p}_k^*$ ,  $k = 1, 2$ , defined in Section 4.

**Example 1.** DIMSIM of type 1 with  $p = q = r = s = 7$ :

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{26225209}{210944121} & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{5872079}{29242011} & \frac{72498603}{165085843} & 0 & 0 & 0 & 0 & 0 \\ \frac{10269405}{119181617} & -\frac{1241524}{30758345} & \frac{57678976}{136838653} & 0 & 0 & 0 & 0 \\ 46724395 & -202076298 & 33989436 & 36154700 & 0 & 0 & 0 \\ 93207344 & -422268925 & 89167643 & 110456583 & 0 & 0 & 0 \\ 197176599 & -226625878 & 26645008 & -12647561 & \frac{17039825}{43988836} & 0 & 0 \\ 247564823 & -252937909 & 35221845 & -140225087 & 0 & 0 & 0 \\ 46683873 & 33428207 & 14947905 & 4240409 & \frac{22361824}{143643055} & \frac{250799284}{740963981} & 0 \\ 78558233 & 212785478 & 183866696 & 37806653 & 0 & 0 & 0 \end{bmatrix},$$

$$v = \begin{bmatrix} -28395433 & 174764707 & 61068807 & 78763283 & 108651209 & -172181322 & 203608151 \\ -37421856 & 42695076 & -7786489 & 18719361 & 18468937 & -18569233 & 43410586 \end{bmatrix}^T.$$

$$Abserr = \left[ 3.17E-9 \quad 2.54E-9 \quad 4.63E-9 \quad 5.66E-9 \quad 1.80E-8 \quad 2.97E-8 \right],$$

$$Relerr = \left[ 1.78E-15 \quad 7.77E-11 \quad 5.12E-10 \quad 5.88E-9 \quad 2.35E-9 \quad 3.60E-7 \quad 3.02E-6 \quad 1.01E-5 \right].$$

**Example 2.** DIMSIM of type 2 with  $p = q = r = s = 7$ :

$$A = \begin{bmatrix} \frac{13}{50} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{8399358}{117950125} & \frac{13}{50} & 0 & 0 & 0 & 0 & 0 \\ \frac{137677936}{85302451} & -\frac{158251283}{168989756} & \frac{13}{50} & 0 & 0 & 0 & 0 \\ \frac{459984888}{139340321} & -\frac{172907567}{219474067} & -\frac{80139292}{152818937} & \frac{13}{50} & 0 & 0 & 0 \\ \frac{117318730}{25139701} & \frac{55751383}{64457867} & -\frac{40708070}{31456679} & -\frac{7529384}{55624863} & \frac{13}{50} & 0 & 0 \\ \frac{54651586}{12102437} & \frac{248094804}{53042371} & -\frac{288405184}{66758097} & \frac{74141340}{51360623} & -\frac{41799922}{92623973} & \frac{13}{50} & 0 \\ \frac{40862117}{18940814} & \frac{364213522}{37737729} & -\frac{116889055}{10983246} & \frac{345753185}{53804472} & -\frac{444619571}{239868105} & \frac{3474421}{119096105} & \frac{13}{50} \end{bmatrix},$$

$$v = \begin{bmatrix} -\frac{70954508}{110162699} & \frac{372657244}{73956795} & -\frac{455524031}{27231397} & \frac{917894351}{29737292} & -\frac{241476488}{7074439} & \frac{329952632}{14897231} & -\frac{180142057}{32466393} \end{bmatrix}^T.$$

$$Abserr = \left[ 1.79E - 8 \quad 1.21E - 8 \quad 5.53E - 8 \quad 9.84E - 8 \quad 7.00E - 8 \right],$$

$$Relerr_1 = \left[ 1.07E - 14 \quad 4.35E - 10 \quad 4.23E - 9 \quad 2.50E - 7 \quad 2.38E - 7 \quad 4.08E - 6 \quad 1.37E - 4 \quad 1.22E - 4 \right],$$

$$Relerr_2 = \left[ 2.40E - 9 \quad 8.33E - 10 \quad 1.06E - 8 \quad 6.68E - 8 \quad 2.73E - 7 \quad 1.07E - 5 \quad 1.66E - 4 \right].$$

**Example 3.** DIMSIM of type 1 with  $p = q = r = s = 8$ :

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{41686181}{32414671} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{64258923}{7925396} & \frac{97865666}{185134823} & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{456302423}{32742546} & -\frac{14182864}{117877009} & \frac{22834578}{57844559} & 0 & 0 & 0 & 0 & 0 \\ -\frac{567357943}{38684087} & -\frac{50235523}{37656308} & \frac{59902158}{149212159} & \frac{13847132}{46776463} & 0 & 0 & 0 & 0 \\ -\frac{172785083}{11244470} & -\frac{644919221}{188639011} & \frac{55003604}{83971505} & \frac{11321982}{58316113} & \frac{34847813}{129216592} & 0 & 0 & 0 \\ -\frac{547956953}{25421975} & -\frac{102353633}{17944926} & \frac{37989857}{71735872} & \frac{60118622}{79790849} & -\frac{19638527}{132550897} & \frac{45822500}{140763599} & 0 & 0 \\ -\frac{465883421}{16020725} & -\frac{231177059}{46568133} & \frac{106717669}{46748560} & \frac{275086438}{77857801} & -\frac{135771059}{97419189} & \frac{46828849}{134197529} & \frac{50169104}{169521953} & 0 \end{bmatrix},$$

$$v = \begin{bmatrix} \frac{10373}{162375432} & -\frac{1418039}{14617506} & \frac{52835174}{89629025} & -\frac{113716474}{76228883} & \frac{88764879}{43132139} & -\frac{250493771}{147034220} & \frac{214334244}{244179187} & \frac{76608154}{99863579} \end{bmatrix}.$$

$$Abserr = \begin{bmatrix} 4.25E-8 & 6.98E-8 & 4.52E-8 & 5.29E-8 & 1.12E-7 & 1.55E-7 & 1.73E-9 \end{bmatrix},$$

$$Relerr = \begin{bmatrix} 0 & 4.82E-10 & 4.79E-9 & 5.69E-9 & 1.75E-7 & 2.55E-7 & 1.64E-5 & 2.65E-5 & 2.70E-3 \end{bmatrix}.$$

**Example 4.** DIMSIM of type 2 with  $p = q = r = s = 8$ :

$$A = \begin{bmatrix} \frac{1}{5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{29660777}{96054083} & \frac{1}{5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{81577825}{30562841} & -\frac{32950648}{124641221} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{50671085}{14723711} & \frac{128721381}{213530981} & \frac{1}{5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{135894950}{99550547} & -\frac{1583791}{127479829} & -\frac{8852243}{71744777} & \frac{1}{5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{113532128}{81033671} & \frac{63892011}{81294886} & \frac{162948701}{124794139} & -\frac{51753098}{126004593} & \frac{1}{5} & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{540060619}{66152888} & \frac{124601993}{62713455} & \frac{73715217}{308886970} & \frac{67425553}{68492811} & -\frac{65237504}{157412471} & \frac{1}{5} & 0 & 0 & 0 & 0 & 0 \\ -\frac{625208861}{33912725} & -\frac{190997406}{17260115} & \frac{260676039}{36675763} & -\frac{337300447}{100694107} & \frac{540483031}{184731048} & -\frac{9881646}{49220819} & \frac{1}{5} & 0 & 0 & 0 & 0 \\ \frac{8685405}{178211477} & -\frac{20626923}{53900582} & \frac{183577433}{141021902} & -\frac{172451360}{69672767} & \frac{226709081}{84276081} & -\frac{177354843}{128426080} & -\frac{8162528}{72728809} & \frac{44809749}{34193171} & \frac{1}{5} & 0 & 0 \end{bmatrix},$$

$$v = \begin{bmatrix} \frac{8685405}{178211477} & -\frac{20626923}{53900582} & \frac{183577433}{141021902} & -\frac{172451360}{69672767} & \frac{226709081}{84276081} & -\frac{177354843}{128426080} & -\frac{8162528}{72728809} & \frac{44809749}{34193171} \end{bmatrix}^T.$$

$$Abserr = \left[ 2.15E - 6 \quad 1.43E - 6 \quad 1.33E - 5 \quad 8.31E - 5 \quad 1.89E - 5 \quad 2.98E - 6 \right],$$

$$Relerr_1 = \left[ 0 \quad 5.23E - 8 \quad 4.76E - 7 \quad 3.14E - 7 \quad 1.08E - 5 \quad 1.75E - 5 \quad 6.87E - 3 \quad 1.20E - 1 \quad 7.23E - 1 \right],$$

$$Relerr_2 = \left[ 5.54E - 8 \quad 8.10E - 8 \quad 1.85E - 7 \quad 3.52E - 7 \quad 6.11E - 6 \quad 6.10E - 7 \quad 1.29E - 3 \quad 3.85E - 2 \right].$$

## 7 Assessment of the methods

Even though the methods derived here have not yet been subjected to numerical testing, it is possible to say something about their likely performance in comparison with standard methods. We will consider, in particular, the type 1 method of order 8 derived here and attempt to assess it in comparison with the 12 stage explicit Runge-Kutta method of the same order derived by Dormand and Prince (DPRK8) [15]. Making an appropriate rescaling to compensate for the differing numbers of stages, it is found that the stability regions are almost identical. It is not possible to compare error constants for the two methods since, in the case of DPRK8, this is problem dependent. However, using the canonical test problem  $y' = \lambda y$ , we can compare the work to produce equivalent errors. It is found that from this point of view the two methods are again very similar; the scaled work for the DIMSIM method is approximately 6% greater than for DPRK8.

These comparisons indicate that the order 8 type 1 DIMSIM is likely to have at least comparable performance to DPRK8. However, the fact that the stage order of the DIMSIM method is  $q = p = 8$ , suggests that this method will actually have several advantages. Without any additional computation in a step, order 8 interpolation is possible as is an asymptotically correct error estimate. These two properties are not achieved for DPRK8. It is also believed that the higher stage order will make the DIMSIM method more successful for mildly stiff problems.

As we have remarked, from the point of view of truncation error, at least for constant-coefficient linear problems, the DPRK8 and the DIMSIM methods are more or less equivalent. However, the fact that there are 12 rather than 8 stages in PRRK8 means that failed steps will cost much more for the Runge-Kutta method. Hence, for problems where such failures can arise, the DIMSIM is to be preferred.

At first sight, the multivalued nature of the DIMSIM method creates implementation difficulties that do not arise for Runge-Kutta methods. This especially applies to the need for starting and stepsize changing algorithms. In a paper now under preparation, [4], it is shown how stepsize changing can be carried out in an inexpensive manner without loss of stability. As for a starting method, this can be avoided by constructing a variable stepsize algorithm making use of type 1 methods that are now known from orders 1 up to the orders 7 and 8 methods announced in the present paper. The crucial elements of such an algorithm are all present because it is not only possible to estimate local truncation errors for each method in the sequence, but it is possible, without extra stages, to estimate the truncation errors for

the next higher, and of course next lower, members of the sequence.

Similar comments apply to DIMSIMs of type 2. Without any additional computation in a step interpolation of order  $p = q$  is possible as is an asymptotically correct error estimate. The problem of starting the integration can again be avoided by construction of a variable-step variable-order algorithm starting with a method of order one. Moreover, since these methods have stage order  $q$  equal to the order  $p$  they will not suffer from order reduction phenomenon while integrating stiff systems of ODEs as do formulas of low stage order such as, for example, SDIRK methods.

## 8 Concluding remarks

We described the approach to the construction of DIMSIMs for the numerical solution of ODEs. These methods form a subclass of general linear methods and have a considerable potential for efficient implementation. We constructed both type 1 (explicit) and type 2 (implicit) methods which are appropriate for nonstiff or stiff differential systems, respectively, in a sequential computing environment. These methods were obtained using the approach based on least squares minimization with the aid of state-of-the-art PORT library routines DN2G and DN2GB. The examples of explicit and implicit methods are presented of order  $p$  and stage order  $q$  equal to  $p = q = 7$  and  $p = q = 8$ . The explicit methods have the same stability region as polynomial approximation to the exponential function of the same order. The stability function of the implicit methods is a generalized approximation to the exponential function which is A-stable and L-stable.

This paper deals only with the construction of high order DIMSIMs of type 1 and 2 with prescribed stability properties. We are, however, well aware that various implementation issues related to these methods are equally, or perhaps even more important than construction, and may influence the choice of appropriate formulas. These implementation issues such as local error estimation, strategies for changing the stepsize and order of the methods, and construction of continuous interpolants are treated in recent papers [4], [8], [17], and [22]. The choice of starting procedures is considered in [22]. Note that we envisage that DIMSIMs would be used in a variable order implementation, and in such a case, the need for a specific starting method for each order is eliminated.

**Acknowledgment.** We would like to express our gratitude to the anonymous referee for constructive criticism which helped us to improve the presentation of some parts of the paper.

## References

- [1] D.S. Bunch, D.M. Gay, and R.E. Welsch, Algorithm 717, Subroutines for maximum likelihood and quasi-likelihood estimation of parameters in nonlinear regression models, *ACM Trans. Math. Software* 19(1993), 109–130.
- [2] C. Bischof, A. Carle, P. Khademi and A. Mauer, The ADIFOR2.0 system for the automatic differentiation of Fortran 77 programs, Argonne Preprint ANL-MCS-P481-1194, Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439-4844, 1994.
- [3] J.C. Butcher, Diagonally-implicit multi-stage integration methods, *Appl. Numer. Math.* 11(1993), 347-363.
- [4] J.C. Butcher, P. Chartier and Z. Jackiewicz, Nordsieck representation of DIMSIMs, manuscript.
- [5] J.C. Butcher and F.H. Chipman, Generalized Padé approximations to the exponential functions, *BIT* 32(1992), 118–130.
- [6] J.C. Butcher and Z. Jackiewicz, Diagonally implicit general linear methods for ordinary differential equations, *BIT* 33(1993), 452–472.
- [7] 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.
- [8] J.C. Butcher and Z. Jackiewicz, Implementation of diagonally implicit multistage integration methods for ordinary differential equations, to appear in *SIAM J. Numer. Anal.*
- [9] J.C. Butcher and Z. Jackiewicz, Construction of high order DIMSIMs for ordinary differential equations, submitted.
- [10] J.E. Dennis, D.M. Gay, and R.E. Welsch, An adaptive nonlinear least-squares algorithm, *ACM Trans. Math. Software* 7(1981), 348–368.
- [11] J.E. Dennis, D.M. Gay, and R.E. Welsch, Algorithm 573, NL2SOL - An adaptive nonlinear least-squares algorithm, *ACM Trans. Math. Software* 7(1981), 369–383.
- [12] J.E. Dennis and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, (Prentice-Hall, Englewood Cliffs, 1983).

- [13] DONLP2, A SQP method for nonlinear constrained minimization by P. Spellucci, source code (f77) of latest version (1996) available at ftp: [//plato.la.asu.edu/pub](ftp://plato.la.asu.edu/pub)
- [14] D.M. Gay, A trust-region approach to linearly constrained optimization, *Numerical Analysis Proceedings* (Dundee, 1983), D.F. Griffiths (ed.), Springer-Verlag, 72–105.
- [15] E. Hairer, S.P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations I. Nonstiff Problems*, (Springer-Verlag, Berlin, Heidelberg, New York, 1993).
- [16] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, (Springer-Verlag, Berlin, Heidelberg, New York, 1996).
- [17] Z. Jackiewicz, R. Vermiglio and M. Zennaro, Variable stepsize diagonally implicit multistage integration methods for ordinary differential equations, *Appl. Numer. Math.* 16(1995), 343–367.
- [18] J.D. Lambert, *Computational Methods in Ordinary Differential Equations*, (John Wiley & Sons, Chichester, New York, 1973).
- [19] J.J. Moré, The Levenberg-Marquardt algorithm: Implementation and theory, in *Lecture Notes in Mathematics* 630, G. Watson, ed., Springer-Verlag, New York, 1978, 105–116.
- [20] S.S. Oren, Self-scaling variable metric algorithms without line search for unconstrained minimization, *Math. Comp.* 27(1973), 873–885.
- [21] The PORT Mathematical Subroutine Library, 3rd ed., AT&T Laboratories, Murray Hill, NJ, 1984.
- [22] J. Van Wieren, Implementation of DIMSIMs of type 1, Report, Naval Air Warfare Center, Weapons Division, China Lake, California.