Experiments with a variable-order type 1 DIMSIM code

J.C. Butcher [a,*], P. Chartier [b] and Z. Jackiewicz [c,**]

[a] *Department of Mathematics, The University of Auckland, Private Bag 92019, Auckland, New Zealand*
[b] *INRIA, Campus Universitaire de Beaulieu, F-35042 Rennes Cedex, France*
[c] *Department of Mathematics, Arizona State University, Tempe, AZ 85287, USA*

Received 5 November 1999
Communicated by C. Brezinski

The issues related to the development of a new code for nonstiff ordinary differential equations are discussed. This code is based on the Nordsieck representation of type 1 DIMSIMs, implemented in a variable-step size variable-order mode. Numerical results demonstrate that the error estimation employed in the code is very reliable and that the step and order changing strategies are very robust. This code outperforms the Matlab `ode45` code for moderate and stringent tolerances.

Keywords: DIMSIM methods, Nordsieck representation, local error estimation, step size and order changing strategy

AMS subject classification: 65L05

1. Introduction

It is the purpose of this paper to describe a new code for nonstiff ordinary differential equations (ODEs)

$$\begin{cases} y'(x) = f\big(y(x)\big), & x \in [x_0, X], \\ y(x_0) = y_0, \end{cases} \tag{1.1}$$

$f : \mathbb{R}^m \to \mathbb{R}^m$, which is based on diagonally implicit multistage integration methods (DIMSIMs) introduced by Butcher [2] and further investigated in [3–6,8–11]. This code utilizes the Nordsieck representation of DIMSIMs introduced recently by Butcher et al. [5]. On a nonuniform grid

$$x_0 < x_1 < \cdots < x_N, \quad x_N \geqslant X,$$

the Nordsieck representation of these methods takes the form

$$\begin{cases} Y^{[n]} = h_n(A \otimes I_m)F\big(Y^{[n]}\big) + \big(PD(\delta_n) \otimes I_m\big)z^{[n-1]}, \\ z^{[n]} = h_n(G \otimes I_m)F\big(Y^{[n]}\big) + \big(QD(\delta_n) \otimes I_m\big)z^{[n-1]}, \end{cases} \tag{1.2}$$

[*] The work of this author was assisted by the Marsden Fund of New Zealand.
[**] The work of this author was partially supported by the National Science Foundation under grant NSF DMS-9971164.

$n = 1, 2, \ldots, N,\ h_n = x_n - x_{n-1}$. Here, $z^{[n]}$ is an approximation of order p to the correct value function $z(x_n)$ defined by

$$z(x_n) = \begin{bmatrix} y(x_n) \\ h_n y'(x_n) \\ \vdots \\ h_n^p y^{(p)}(x_n) \end{bmatrix}$$

(these approximations propagate to the next step); $Y^{[n]}$ is an approximation of (stage) order q to the vector $Y(x_{n-1})$ with components $y(x_{n-1} + c_i h_n) \in \mathbb{R}^m$, $i = 1, 2, \ldots, s$, and

$$F\bigl(Y^{[n]}\bigr) = \begin{bmatrix} f(Y_1^{[n]}) \\ f(Y_2^{[n]}) \\ \vdots \\ f(Y_s^{[n]}) \end{bmatrix} \in \mathbb{R}^{ms}.$$

We will refer to the integer s as the number of internal stages and to the vector $c = [c_1, \ldots, c_s]^{\mathrm{T}}$ as the vector of stage abscissas. The coefficient matrices A, P, G, and Q are matrices of dimensions $p \times p$, $p \times (p+1)$, $(p+1) \times p$, and $(p+1) \times (p+1)$, respectively, and the rescaling matrix $D(\delta_n)$ has the form

$$D(\delta_n) = \operatorname{diag}\bigl(1, \delta_n, \ldots, \delta_n^p\bigr),$$

with $\delta_n = h_n / h_{n-1}$. It will always be assumed that A is lower triangular and Q is a rank one matrix of the form $Q = e_1 q^{\mathrm{T}}$, where $e_1 = [1, 0, \ldots, 0]^{\mathrm{T}} \in \mathbb{R}^{s+1}$ and $q = [1, q_1, \ldots, q_s]^{\mathrm{T}}$. The last condition guarantees that Q is power bounded and, as a consequence, the method (1.2) is zero-stable.

There is some theoretical and numerical evidence (see [5,8]) that the methods with $p = q = s$ and $c_i \neq c_j$ for $i \neq j$ have the greatest potential for practical use and in the remainder of this paper we will restrict our attention to such methods.

It was proved in [5] (compare also [8]) that the local discretization error $\Gamma(x_n)$ of the method (1.2) at the point x_n is given by

$$\Gamma(x_n) = h_n^{p+1} (\varphi_p \otimes I_m) y^{(p+1)}(x_{n-1}) + \mathrm{O}\bigl(h_n^{p+2}\bigr),$$

where

$$\varphi_p = a - \frac{Gc^p}{p!} \quad \text{and} \quad a = \begin{bmatrix} \frac{1}{(p+1)!} & \frac{1}{p!} & \cdots & 1 \end{bmatrix}^{\mathrm{T}}.$$

It was also demonstrated in [5] that the reliable control of the global discretization error of (1.2) can be achieved by bounding the local contributions

$$h_n^{p+1} \bigl(q^{\mathrm{T}} \varphi_p\bigr) y^{(p+1)}(x_{n-1})$$

at each step, where q is now used to denote the vector which defines the matrix Q. These contributions can be estimated by the formula

$$\begin{aligned} & h_n^{p+1}\big(q^{\mathrm{T}}\varphi_p\big)y^{(p+1)}(x_{n-1}) \\ & \quad = h_n\big(\beta^{\mathrm{T}}(\delta_n)\otimes I_m\big)F\big(Y^{[n]}\big) + h_{n-1}\big(\gamma^{\mathrm{T}}(\delta_n)\otimes I_m\big)F\big(Y^{[n-1]}\big) + \mathrm{O}\big(h^{p+2}\big), \end{aligned} \tag{1.3}$$

where $h = \max\{h_i\colon\ i = 0,\dots,n\}$ and where the vectors $\beta(\delta_n)$ and $\gamma(\delta_n)$ satisfy the system of equations

$$\begin{cases} \beta^{\mathrm{T}}(\delta_n)C_pKD(\delta_n)T + \gamma^{\mathrm{T}}(\delta_n)C_pK = 0, \\ \beta^{\mathrm{T}}(\delta_n)\left(C_pKD(\delta_n)a + \delta_n^{p+1}\dfrac{c^p}{p!}\right) + \gamma^{\mathrm{T}}(\delta_n)\dfrac{c^p}{p!} = \delta_n^{p+1}q^{\mathrm{T}}\varphi_p. \end{cases}$$

Here,

$$C_p = \left[e\ c\ \dots\ \frac{c^p}{p!}\right], \quad e = [1\ 1\ \dots\ 1]^{\mathrm{T}} \in \mathbb{R}^p,$$

$$K = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \qquad T = \begin{bmatrix} 1 & 1 & \dfrac{1}{2!} & \dots & \dfrac{1}{p!} \\ 0 & 1 & 1 & \dots & \dfrac{1}{(p-1)!} \\ 0 & 0 & 1 & \dots & \dfrac{1}{(p-2)!} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}.$$

Observe that the formula (1.3) depends on stage values $Y^{[n-1]}$ and $Y^{[n]}$ at two consecutive steps which, in a variable-order implementation, may be computed by methods of different orders. As a consequence, the formula is not very convenient in this case and we will use instead the error estimator based on $Y^{[n]}$ and $z^{[n-1]}$.

This error estimator is developed in section 2. The derivation of DIMSIMs in Nordsieck form is briefly reviewed in section 3 and the resulting methods are listed for $1 \leqslant p \leqslant 4$ along with the vectors $\beta(\delta_n)$ and $\gamma(\delta_n)$ which appear in the error estimating formula (2.1). The methods of order $5 \leqslant p \leqslant 8$ and the corresponding vectors $\beta(\delta_n)$ and $\gamma(\delta_n)$ are listed in the appendix. In section 4 we describe the issues related to the implementation of DIMSIMs such as the choice of initial order, step size, and initial vector $z^{[0]}$, step size and order changing strategies, and how to update the vector of external approximations $z^{[n]}$ so that it corresponds to the new order. In section 5 we present a selection of numerical experiments. These results demonstrate that our code is more efficient for moderate and stringent tolerances than the Matlab `ode45` code written by Shampine and Reichelt [16].

2. Local error estimation

In this section we will derive error estimates based on $Y^{[n]}$ and $z^{[n-1]}$. These estimates depend only on the quantities computed in the current step and, as a consequence, are more convenient in variable order implementation than the estimates previously derived in [5]. The approach to the derivation of these estimates is similar to that used in [7].

Theorem 2.1. Assume that the method (1.2) has order p and stage order $q = p$. Then the principal part of the local discretization error $\Gamma(x_n)$ can be estimated by the formula

$$\begin{aligned} &h_n^{p+1}\big(q^{\mathrm{T}}\varphi_p\big)y^{(p+1)}(x_{n-1}) \\ &\quad = h_n\big(\beta^{\mathrm{T}}(\delta_n)\otimes I_m\big)F\big(Y^{[n]}\big) + \big(\gamma^{\mathrm{T}}(\delta_n)\otimes I_m\big)z^{[n-1]} + \mathrm{O}\big(h^{p+2}\big), \end{aligned} \tag{2.1}$$

where the vectors $\beta(\delta_n)$ and $\gamma(\delta_n)$ satisfy the system of equations

$$\begin{cases} \beta^{\mathrm{T}}(\delta_n)C_pKD(\delta_n)T + \gamma^{\mathrm{T}}(\delta_n)(GC_pK + Q) = 0, \\ \beta^{\mathrm{T}}(\delta_n)\left(C_pKD(\delta_n)a + \delta_n^{p+1}\dfrac{c^p}{p!}\right) + \gamma^{\mathrm{T}}(\delta_n)G\dfrac{c^p}{p!} = \delta_n^{p+1}q^{\mathrm{T}}\varphi_p, \\ \big(\delta_n\beta^{\mathrm{T}}(\delta_n) + \gamma^{\mathrm{T}}(\delta_n)G\big)e = 0. \end{cases} \tag{2.2}$$

Proof. As in [5] we can show that

$$\begin{aligned} h_nf\big(Y^{[n]}\big) &= (C_pK\otimes I_m)\widetilde{z}(x_{n-1}) + h_n^{p+1}\left(\frac{c^p}{p!}\otimes y^{(p+1)}(x_{n-1})\right) \\ &\quad + h_n\left(e\otimes\frac{\partial f}{\partial y}\big(y(x_{n-2})\big)\nu_{n-1}\right) + \mathrm{O}\big(h^{p+2}\big), \end{aligned}$$

and

$$\widetilde{z}(x_{n-1}) = \big(D(\delta_n)T\otimes I_m\big)\widetilde{z}(x_{n-2}) + h_{n-1}^{p+1}\big(D(\delta_n)a\otimes y^{(p+1)}(x_{n-2})\big) + \mathrm{O}\big(h^{p+2}\big),$$

where $\widetilde{z}(x_{n-1}) = D(\delta_n)z(x_{n-1})$ and $\nu_{n-1}\in\mathbb{R}^m$ are certain vectors defined in [5, theorem 10]. We also have

$$z^{[n-1]} = h_{n-1}(G\otimes I_m)f\big(Y^{[n-1]}\big) + (Q\otimes I_m)\widetilde{z}(x_{n-2}) + \mathrm{O}\big(h^{p+2}\big).$$

Substituting the above formulas into (2.1) and using the relations

$$y^{(p+1)}(x_{n-1}) = y^{(p+1)}(x_{n-2}) + \mathrm{O}(h), \qquad \frac{\partial f}{\partial y}\big(y(x_{n-1})\big) = \frac{\partial f}{\partial y}\big(y(x_{n-2})\big) + \mathrm{O}(h),$$

and $\nu_{n-1} = \nu_{n-2} + O(h^{p+1})$ (compare [5, theorem 10]) we obtain

$$\begin{aligned} &\delta_n^{p+1}h_{n-1}^{p+1}\big(q^{\mathrm{T}}\varphi_p\big)y^{(p+1}(x_{n-2}) \\ &\quad = \big(\big(\beta^{\mathrm{T}}(\delta_n)C_pKD(\delta_n)T + \gamma^{\mathrm{T}}(\delta_n)(GC_pK+Q)\big)\otimes I_m\big)\widetilde{z}(x_{n-2}) \end{aligned}$$

$$+\left(\beta^{\mathrm{T}}(\delta_n)\left(C_pKD(\delta_n)a+\delta_n^{p+1}\frac{c^p}{p!}\right)+\gamma^{\mathrm{T}}(\delta_n)\frac{c^p}{p!}\right)h_{n-1}^{p+1}y^{(p+1)}(x_{n-2})$$
$$+\left(\delta_n\beta^{\mathrm{T}}(\delta_n)+\gamma^{\mathrm{T}}(\delta_n)\right)eh_{n-1}\frac{\partial f}{\partial y}\big(y(x_{n-2})\big)\nu_{n-1}+\mathrm{O}\big(h^{p+2}\big).$$

Comparing corresponding terms in the above relation leads to the system (2.2). This completes the proof. □

In the implementation of these methods and similar methods of higher order, specific choices of the parameters $\gamma_2,\gamma_3,\dots,\gamma_{p+1}$ need to be made. The simplest option, and the one actually used in the experimental code we will describe, is to choose each γ_i equal to zero except for γ_{p+1}. This means that the error estimator takes the form

$$\sum_{i=0}^{p}\beta_ih_nf\big(Y_i^{[n]}\big)-\gamma_{p+1}z_{p+1}^{[n-1]}.$$

Because $z_{p+1}^{[n-1]}=\mathrm{O}(h_n^p)$, the same must be true for the term $\sum_{i=1}^{p}\beta_ih_nf(Y_i^{[n]})$ and accordingly, the only choice for this is a scalar multiple of the same expression also used as the value of $z_{p+1}^{[n]}$. Hence, the approximation to $\Gamma(x_n)$ that we need to use has the form $\alpha z_{p+1}^{[n]}-\alpha' z_{p+1}^{[n-1]}$ for some constants α and α'. Because $z_{p+1}^{[n]}=\delta_n^pz_{p+1}^{[n-1]}+\mathrm{O}(h_n^{p+1})$, α' must be chosen as $\delta_n^p\alpha$. To find the correct choice of α we note that for $p>1$

$$\begin{aligned}z_{p+1}^{[n]}&=h_n^py^{(p)}\left(x_{n-1}+\frac{h_n}{2}\right)+\mathrm{O}\big(h_n^{p+2}\big)\\&=h_n^py^{(p)}(x_{n-1})+\frac{h_n^{p+1}}{2}y^{(p+1)}(x_{n-1})+\mathrm{O}\big(h_n^{p+2}\big)\end{aligned}$$

and, similarly, that

$$z_{p+1}^{[n-1]}=h_{n-1}^py^{(p)}(x_{n-1})-\frac{h_{n-1}^{p+1}}{2}y^{(p+1)}(x_{n-1})+\mathrm{O}\big(h_{n-1}^{p+2}\big).$$

Hence,

$$z_{p+1}^{[n]}-\delta_n^pz_{p+1}^{[n-1]}=\left(\frac{1}{2}+\frac{1}{2\delta_n}\right)h_n^{p+1}y^{(p+1)}(x_{n-1})+\mathrm{O}\big(h_n^{p+2}\big),$$

and it follows that α should be chosen as $2\delta_n/(1+\delta_n)$. In summary, we may take as an approximation to the local discretization error for $p>1$ the following:

$$h_n^{p+1}\big(q^{\mathrm{T}}\varphi_p\big)y^{(p+1)}(x_{n-1})=q^{\mathrm{T}}\varphi_p\frac{2\delta_n}{1+\delta_n}\big(z_{p+1}^{[n]}-\delta_n^pz_{p+1}^{[n-1]}\big)+\mathrm{O}\big(h_n^{p+1}\big).$$

3. The derivation of DIMSIMs in Nordsieck form

To derive DIMSIMs in Nordsieck form (1.2) we will follow the approach described in [5] and use as starting point the representation introduced in [2]. This representation on uniform grid with step size h takes the form

$$\begin{cases} Y^{[n]} = h(A \otimes I_m)F\bigl(Y^{[n]}\bigr) + (U \otimes I_m)y^{[n-1]}, \\ y^{[n]} = h(B \otimes I_m)F\bigl(Y^{[n]}\bigr) + (V \otimes I_m)y^{[n-1]}, \end{cases} \tag{3.1}$$

$n = 1, 2, \ldots, N,\ A, U, B, V \in \mathbb{R}^{p\times p}$, where A is the same as in (1.2), $U = I_p$, and V is a rank one matrix of the form $V = ev^{\mathrm{T}}$, $v = [v_1, \ldots, v_p]^{\mathrm{T}}$ with $v^{\mathrm{T}}e = 1$. The vector $y^{[n]}$ is an approximation of order p to the linear combination

$$\sum_{k=0}^{p} h^k(\alpha_k \otimes I_m)y^{(k)}(x_n)$$

with

$$\alpha_0 = e, \qquad \alpha_k = \frac{c^k}{k!} - \frac{Ac^{k-1}}{(k-1)!}, \quad k = 1, 2, \ldots, p.$$

These vectors are collected in the matrix W defined by

$$W = [\alpha_0\ \alpha_1\ \ldots\ \alpha_p].$$

Denote by C the Vandermonde matrix

$$C = \bigl[e\ c\ \ldots\ c^{p-1}\bigr]$$

and by L the matrix with columns L_k defined by

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

Here, $e_i,\ i = 1, 2, \ldots, p+1$, is the canonical basis in $\mathbb{R}^{p+1}$. Then it was proved in [5] that the Nordsieck representation of DIMSIMs (1.2) can be computed from the relations

$$P = W, \qquad Q = e_1\bigl[1\ v^{\mathrm{T}}\alpha_1\ \ldots\ v^{\mathrm{T}}\alpha_p\bigr], \qquad G = LC^{-1}.$$

These formulas as well as the solution to the system (2.2) are very easy to implement in MATHEMATICA. We list below the resulting examples of DIMSIMs (1.2) of order $1 \leqslant p \leqslant 4$ corresponding to methods of the form (3.1) derived in [2,7,9,10] along with the vectors $\beta(\delta_n)$ and $\gamma(\delta_n)$ defined by (2.2). These vectors are used to estimate the principal part of the local discretization error of the resulting methods. The methods of order $5 \leqslant p \leqslant 8$ and the corresponding vectors $\beta(\delta_n)$ and $\gamma(\delta_n)$ appearing in the error estimating formulas are listed in the appendix.

1. $p = q = s = 1$:

$$\left[\begin{array}{c|c} A & P \\ \hline G & Q \end{array}\right] = \left[\begin{array}{c|cc} 0 & 1 & 0 \\ \hline 1 & 1 & 0 \\ 1 & 0 & 0 \end{array}\right], \qquad \beta(\delta_n) = \frac{\delta_n}{2}, \qquad \gamma(\delta_n) = \left[0 \ -\frac{\delta_n^2}{2}\right]^{\mathrm{T}}.$$

This method corresponds to the Euler formula

$$\left[\begin{array}{c|c} A & U \\ \hline B & V \end{array}\right] = \left[\begin{array}{c|c} 0 & 1 \\ \hline 1 & 1 \end{array}\right]$$

with $c = c_1 = 0$.

2. $p = q = s = 2$:

$$\left[\begin{array}{c|c} A & P \\ \hline G & Q \end{array}\right] = \left[\begin{array}{cc|ccc} 0 & 0 & 1 & 0 & 0 \\ 2 & 0 & 1 & -1 & \frac{1}{2} \\ \hline \frac{5}{4} & \frac{1}{4} & 1 & -\frac{1}{2} & \frac{1}{4} \\ 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \end{array}\right],$$

$$\beta(\delta_n) = \left[-\frac{\delta_n^2 + 3\gamma_2(1+\delta_n)}{3\delta_n(1+\delta_n)} \ \ \frac{\delta_n}{3(1+\delta_n)}\right]^{\mathrm{T}},$$

$$\gamma(\delta_n) = \left[0 \ \gamma_2 \ -\frac{\delta_n^3}{3(1+\delta_n)}\right]^{\mathrm{T}}.$$

Here, γ_2 is a free parameter. This method corresponds to the DIMSIM of type 1 of order $p = 2$ with $c = [0, 1]^{\mathrm{T}}$ constructed in [2].

3. $p = q = s = 3$:

$$\left[\begin{array}{c|c} A & P \\ \hline G & Q \end{array}\right] = \left[\begin{array}{ccc|cccc} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & -\frac{1}{2} & \frac{1}{8} & \frac{1}{48} \\ \frac{1}{4} & 1 & 0 & 1 & -\frac{1}{4} & 0 & \frac{1}{24} \\ \hline \frac{5}{4} & \frac{1}{3} & \frac{1}{6} & 1 & -\frac{3}{4} & \frac{1}{6} & \frac{1}{24} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -4 & 3 & 0 & 0 & 0 & 0 \\ 4 & -8 & 4 & 0 & 0 & 0 & 0 \end{array}\right],$$

$$\beta_1 = \frac{\delta_n^4 - 3\delta_n^2\gamma_2(1+\delta_n) + \gamma_3(2+9\delta_n+7\delta_n^2)}{3\delta_n^3(1+\delta_n)},$$

$$\beta_2 = -\frac{2(\delta_n^4 + 2\gamma_3(1+3\delta_n+2\delta_n^2))}{3\delta_n^3(1+\delta_n)},$$

$$\beta_3 = \frac{\delta_n^4 + \gamma_3(2+3\delta_n+\delta_n^2)}{3\delta_n^3(1+\delta_n)},$$

$$\gamma_1 = 0, \qquad \gamma_4 = -\frac{\delta_n^4 + 2\gamma_3(1-\delta_n^2)}{12(1+\delta_n)}.$$

Here, γ_2 and γ_3 are free parameters. This method corresponds to the DIMSIM of type 1 of order $p = 3$ with $c = [0, \frac{1}{2}, 1]^{\mathrm{T}}$ derived in [7].

4. $p = q = s = 4$:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.373935 & 0 & 0 & 0 \\ 0.294985 & 0.481683 & 0 & 0 \\ -0.690374 & 2.27126 & -0.225725 & 0 \end{bmatrix},$$

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & -0.0406015 & 0.0555556 & 0.00617284 & 0.000514403 \\ 1 & -0.110001 & 0.0616613 & 0.0226226 & 0.0052571 \\ 1 & -0.355161 & -0.106603 & 0.0906466 & 0.0387935 \end{bmatrix},$$

$$G = \begin{bmatrix} 2.94445 & -6.16787 & 4.99556 & -1.02483 \\ 0 & 0 & 0 & 1 \\ -1 & 4.5 & -9 & 5.5 \\ -9 & 36 & -45 & 18 \\ -27 & 81 & -81 & 27 \end{bmatrix},$$

$$q^{\mathrm{T}} = [1 \quad 0.25268 \quad 0.250409 \quad -0.0883843 \quad 0.00385044],$$

$$\beta_1 = \frac{-0.45\delta_n^5 - \gamma_2\delta_n^3(1+\delta_n) - \gamma_3(0.5 - 5.5\delta_n^2 - 5\delta_n^3) - \gamma_4(5.5 + 18\delta_n + 12.5\delta_n^2)}{\delta_n^4(1+\delta_n)},$$

$$\beta_2 = \frac{1.35\delta_n^5 + \gamma_3(1.5 - 9\delta_n^2 - 7.5\delta_n^3) + \gamma_4(16.5 + 45\delta_n + 28.5\delta_n^2)}{\delta_n^4(1+\delta_n)},$$

$$\beta_3 = \frac{-1.35\delta_n^5 - 1.5\gamma_3(1 - 3\delta_n^2 - 2\delta_n^3) - \gamma_4(16.5 + 36\delta_n + 19.5\delta_n^2)}{\delta_n^4(1+\delta_n)},$$

$$\beta_4 = \frac{0.45\delta_n^5 + 0.5\gamma_3(1 - 2\delta_n^2 - \delta_n^3) + \gamma_4(5.5 + 9\delta_n + 3.5\delta_n^2)}{\delta_n^4(1+\delta_n)},$$

$$\gamma_1 = 0,$$
$$\gamma_5 = -\frac{0.0166667\delta_n^5 + 0.0185185\gamma_3(1+\delta_n^3) + 0.203704\gamma_4(1-\delta_n^2)}{1+\delta_n}.$$

Here, γ_2, γ_3 and γ_4 are free parameters. This method corresponds to the DIMSIM of type 1 of order $p = 4$ with $c = [0, \frac{1}{3}, \frac{2}{3}, 1]^{\mathrm{T}}$ constructed by Wright [17,18].

4. Implementation of the algorithm

4.1. *Initial step size and order*

The initial order p_1 is always chosen to be equal to one. The initial step size h_1 is computed using a modification of the approach by Gladwell et al. [13] (compare also [14, p. 169; 15, p. 379]).

Put $sc_i = Atol_i + |y_i(x_0)| Rtol_i,\ i = 1, 2, \ldots, m$, where $Atol_i$ and $Rtol_i$ are absolute and relative error tolerances corresponding to the ith component of the solution and, as in [14], define the norm $\|\cdot\|_{sc}$ by

$$\|y\|_{sc} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\frac{y_i^2}{sc_i^2}}.$$

The initial local discretization error of the method of order $p_1 = 1$ is

$$\frac{1}{2}h_1^2 y_i''(x_0)$$

and this suggests the formula

$$h_1 = \sqrt{2/\|d_2\|_{sc}},$$

where $d_2 \approx y''(x_0)$. We compute d_2 by

$$d_2 = \frac{f(x_0 + h_0, y_0 + h_0 f(x_0, y_0)) - f(x_0, y_0)}{h_0},$$

where the step size h_0 corresponds to the method of order zero and is given by

$$h_0 = \frac{1}{\|f(x_0, y_0)\|_{sc}}.$$

4.2. *Starting vector $z^{[0]}$*

The method with $p = q = s = 1$ takes the form

$$\begin{cases} Y^{[n]} = z_1^{[n-1]}, \\ z_1^{[n]} = h_n f\bigl(Y^{[n]}\bigr) + z_1^{[n-1]}, \\ z_2^{[n]} = h_n f\bigl(Y^{[n]}\bigr). \end{cases}$$

Since $Y^{[n]} \approx y(x_{n-1})$, $z_1^{[n]} \approx y(x_n)$ and $z_2^{[n]} \approx h_n y'(x_{n-1})$, this suggests to define the initial vector $z^{[0]}$ as

$$z^{[0]} = \begin{bmatrix} y_0 \\ h_1 f(y_{-1}) \end{bmatrix},$$

where y_{-1} is an approximation to $y(x_0 - h_1)$. This approximation will be computed by the formula

$$y_{-1} = y_0 - h_1 f(y_0).$$

If the differential equation (1.1) is not defined to the left of the initial point x_0 the starting verctor $z^{[0]}$ is defined by

$$z^{[0]} = \begin{bmatrix} y_0 \\ h_1 f(y_0) \end{bmatrix}.$$

4.3. *Step size and order changing strategy*

Assume we have completed a step from x_{n-1} to x_n with a step size h_n and order p_n which resulted in the computation of quantities $Y^{[n]}$ and $z^{[n]}$. We then compute an estimate of the local discretization error at the point x_n by the formula

$$\mathit{est}(x_n, p_n) = h_n\big(\beta^{\mathrm{T}}(\delta_n) \otimes I_m\big) F\big(Y^{[n]}\big) + \big(\gamma^{\mathrm{T}}(\delta_n) \otimes I_m\big) z^{[n-1]},$$

where $\delta_n = h_n / h_{n-1}$. We also compute the measure *err* of the local discretization error by

$$\mathit{err} = \big\| \mathit{est}(x_n, p_n) \big\|_{sc}$$

with

$$sc_i = \mathit{Atol}_i + \max\big\{ \big|z^{[n-1]}(1:m)\big|, \big|z^{[n]}(1:m)\big| \big\} \mathit{Rtol}_i,$$

where $z(1:m)$ stands for the first m components of the vector z, and compare it to one to find an optimal step size for the next step. This optimal step size is

$$h_{\mathrm{opt}} = h_n r_{\mathrm{opt}}, \quad \text{where} \quad r_{\mathrm{opt}} = (1/\mathit{err})^{1/(p_n+1)}.$$

If $\mathit{err} \leqslant 1$ the step is accepted and the new step size h_{n+1} is computed from the formula

$$h_{n+1} = h_n \cdot \min\{\mathit{facmax}, \mathit{fac} \cdot r_{\mathrm{opt}}\},$$

compare [14]. Here, *facmax*, and *fac* are safety factors built into a code to prevent the step from increasing too rapidly and, thus, to avoid an excessive number of rejected steps. We have chosen $\mathit{facmax} = 2$, and $\mathit{fac} = 0.9$.

The choice of the new order p_{n+1} is based on monitoring $\mathit{est}(x_n, p_n)$ as well as the estimate $\mathit{est}(x_n, p_n - 1)$ (for $p_n > 1$) which corresponds to the method of order $p_n - 1$. For $p_n = 1$, we compare the error estimate with the estimated order in “order zero” result defined by $y_n = y_{n-1}$. Since the last m components $z^{[n]}(p_n m + 1 : (p_n + 1)m)$

of the vector $z^{[n]}$ approximate $h_n^{p_n} y^{(p_n)}(x_n)$ this estimate can be easily obtained from the expression

$$\mathit{est}(x_n, p_n - 1) = C(p_n - 1) z^{[n]}\big(p_n m + 1 : (p_n + 1)m\big),$$

where $C(p_n - 1) = q^{\mathrm{T}} \varphi_{p_n - 1}$ is the error constant of the method of order $p_n - 1$ for $p_n > 1$ and $C(p_n - 1) = 1$ for $p_n = 1$. The decision about the new order is based on the *ratio* defined by

$$\mathit{ratio} = \frac{\|\mathit{est}(x_n, p_n)\|}{\|\mathit{est}(x_n, p_n - 1)\|}.$$

If $\mathit{ratio} < r_{\min}$, $p_n < p_{\max}$, and the previous step was not rejected the new order is chosen as $p_{n+1} = p_n + 1$. If $\mathit{ratio} > r_{\max}$ and $p_n > 1$ the new order is $p_{n+1} = p_n - 1$. Otherwise, the order is not changed. We have chosen $r_{\min} = 0.9$, $r_{\max} = 1.1$ and the maximal order is $p_{\max} = 8$.

If $\mathit{err} > 1$ the step is rejected and the computations are repeated with a new step size $\widetilde{h}_n$ equal to

$$\widetilde{h}_n = h_n \cdot \min\{\mathit{facmin}, \mathit{fac} \cdot r_{\mathrm{opt}}\},$$

where *facmin* is a safety factor built into a code to prevent the step from decreasing too rapidly. We have chosen $\mathit{facmin} = 0.5$.

A new order $\widetilde{p}_n$ is chosen according to the following rule. After the first rejection the order was not changed. After the second rejection the order was reduced by one (if $p_n > 1$). After the third rejection the order was dropped to one.

4.4. Updating the vector of external approximations $z^{[n]}$

After a succesful step from x_{n-1} to x_n is completed, we have to update the vector $z^{[n]} = z^{[n]}(p_n)$ of external approximations so it corresponds to the new order p_{n+1}. If $p_{n+1} = p_n$ then the updated vector $z^{[n]}(p_{n+1})$ is equal to $z^{[n]}(p_n)$. If $p_{n+1} = p_n + 1$ then

$$z^{[n]}(p_{n+1}) = \begin{bmatrix} z^{[n]} \\ \mathit{est}(x_n, p_n)/C(p_n) \end{bmatrix},$$

where $C(p_n) = q^{\mathrm{T}} \varphi_{p_n}$ is the error constant corresponding to the method of order p_n. In this case we also want to find constants $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_p$ so that it is appropriate to add corrections to $z^{[n]}$ when the order is increased, as follows:

$$\begin{bmatrix} z_2^{[n]} \\ z_3^{[n]} \\ \vdots \\ z_{p+1}^{[n]} \end{bmatrix} \to \begin{bmatrix} z_2^{[n]} + \varepsilon_1 z_{p+2}^{[n]} \\ z_3^{[n]} + \varepsilon_2 z_{p+2}^{[n]} \\ \vdots \\ z_{p+1}^{[n]} + \varepsilon_p z_{p+2}^{[n]} \end{bmatrix},$$

where $z_{p+2}^{[n]} = est(x_n, p_n)/C(p_n)$ is an approximation to $h_n^{p+1} y^{(p+1)}(x_n)$ computed from the error estimator. We choose ε_k, $k = 2, 3 \ldots, p$, so that

$$\sum_{j=1}^{p} g_{k+1,j} h_n y'(x_n - h_n + h_n c_j) + \varepsilon_k h_n^{p+1} y^{(p+1)}(x_n) = h_n y^{(k)}(x_n) + \mathrm{O}\big(h_n^{p+2}\big).$$

If y is a polynomial of degree $p+1$ the two sides will be equal with the $\mathrm{O}(h_n^{p+2})$ term missing. Hence, substituting $y'(x_n + h_n t) = \varphi(t)$, where φ is a polynomial of degree p, we find

$$\sum_{j=1}^{p} g_{k+1,j} \varphi(c_j - 1) + \varepsilon_k \varphi^{(p)}(0) = \varphi^{(k-1)}(0).$$

Choosing $\varphi(t) = \prod_{j=1}^{p}(t + 1 - c_j)$ so that the g terms vanish, it follows that $\varepsilon_k = \varphi^{(k-1)}(0)/p!$. It is convenient to calculate $\varepsilon_2, \varepsilon_3, \ldots, \varepsilon_p$ from the coefficients in the expansion of φ,

$$\prod_{j=1}^{p}(t + 1 - c_j) = t^p + \alpha_p t^{p-1} + \cdots + \alpha_2 t + \alpha_1,$$

and we find that $\varepsilon_k = \alpha_k (k-1)!/p!$.

If the order is reduced by one, i.e., $p_{n+1} = p_n - 1 \geqslant 1$ then the new vector $z^{[n]} = z^{[n]}(p_{n+1})$ is given by

$$z^{[n]}(p_{n+1}) = z^{[n]}(1 : p_n m).$$

If the step size from x_{n-1} to x_n is rejected the order is never increased and the step is repeated with a new step size $\widetilde{h}_n$ and a new order $\widetilde{p}_n$ equal to p_n, $p_n - 1$ or 1, and an updated vector $z^{[n-1]}(\widetilde{p}_n)$ equal to $z^{[n-1]}(p_n)$, $z^{[n-1]}(1 : p_n m)$, or $z^{[n-1]}(1 : 2m)$, respectively.

5. Numerical experiments

An experimental code was written in Matlab based on methods (1.2) of order $1 \leqslant p \leqslant 8$. The implementation details are discussed in section 4. This code was applied to many problems to test its accuracy, efficiency, the reliability of local error estimation, and the robustness of step size and order changing strategies. We present below a selection of numerical results on the following three problems.

Example 5.1. The scalar problem [14, p. 237].

$$\begin{cases} y' = -\mathrm{sign}(x)\big|1 - |x|\big|y^2, & x \in [-2, 2], \\ y(-2) = \dfrac{2}{3}. \end{cases} \tag{5.1}$$

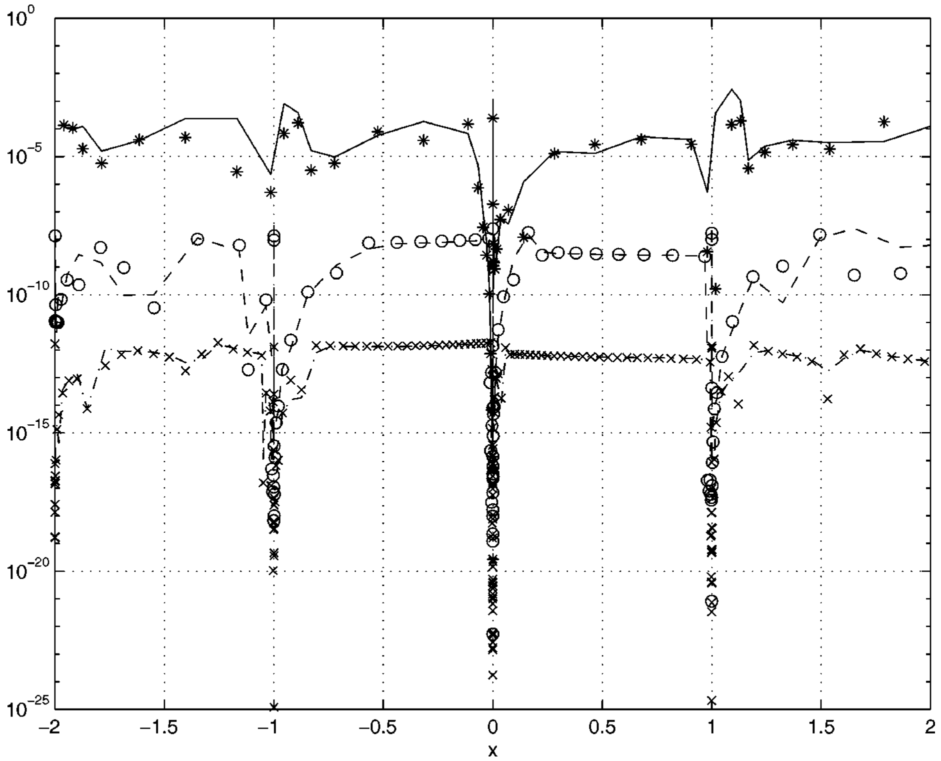

Figure 1. Local error versus local error estimate for the problem (5.1).

The solution y to this problem has a discontinuity in the first derivative y' at $x = 0$ and discontinuities in the second derivative y'' at $x = -1$ and $x = 1$ and the step size and order mechanism adjusts its step size and order accordingly in their neighborhoods. We have plotted in figure 1 the local errors and the local error estimates for *Atol* = *Rtol* = *tol* = 10^{-4} (solid line, symbol '$*$'), 10^{-8} (dashed line, symbol '$\circ$') and 10^{-12} (dash-dotted line, symbol '$\times$'). The corresponding step size and order patterns are plotted in figure 2, where we have used solid, dashed, and dash-dotted lines in figure 2(a) and solid line and symbols '$*$', '$\circ$', and '$\times$' in figure 2(b) for *tol* = 10^{-4}, 10^{-8}, and 10^{-12}, respectively. In figure 2(a) the rejected steps are indicated by '$\times$'.

In table 1 we have listed the number of steps *ns*, number of rejected steps *nr*, number of function evaluations *nfe* for our experimental code based on DIMSIMs with the maximum order $p_{\max}$ restricted to 4, 5, 6, 7, and 8, as well as for the Matlab `ode45` code for *Atol* = *Rtol* = *tol* = 10^{-4}, 10^{-8}, and 10^{-12}. This code was written by Shampine and Reichelt (see [16]) and it is based on explicit Runge–Kutta [4,5] pair DOPRI5 constructed by Dormand and Prince [12]. This code uses local extrapolation so it is effectively of order 5.

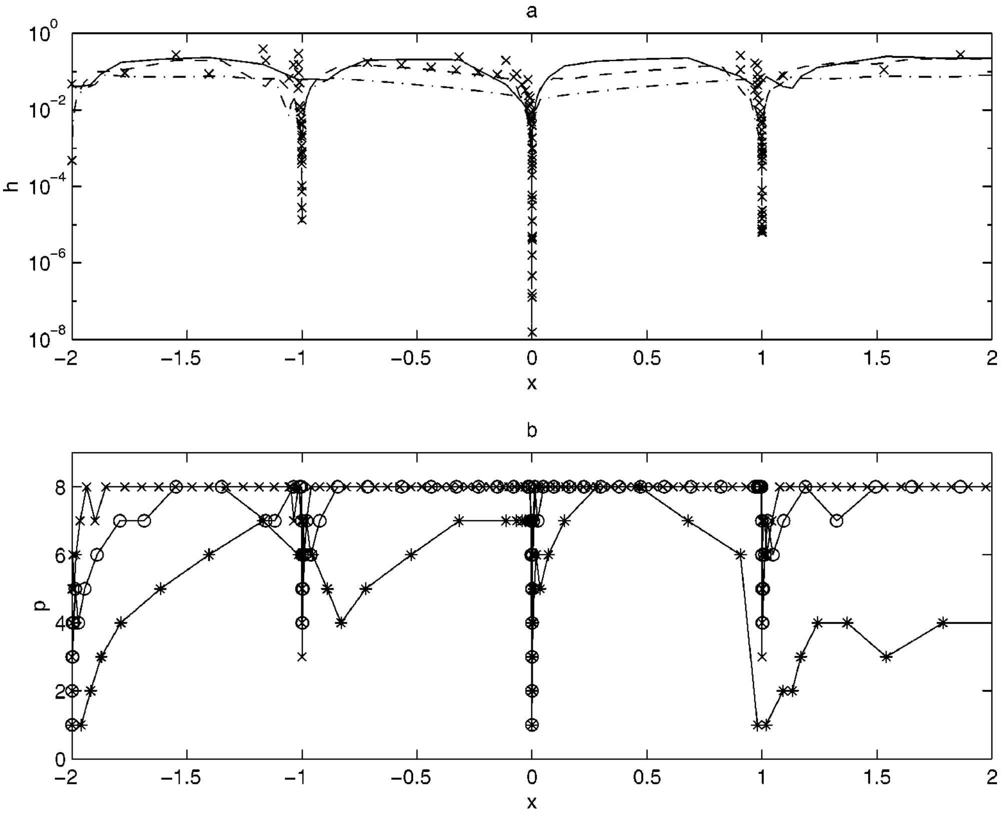

Figure 2. Step size control (a) and order control (b) for the problem (5.1).

Table 1
Numerical results for the example 5.1.

tol	10^{-4}			10^{-8}			10^{-12}		
$p_{\max}$	*ns*	*nr*	*nfe*	*ns*	*nr*	*nfe*	*ns*	*nr*	*nfe*
4	49	28	250	208	49	945	938	47	3877
5	46	23	283	153	41	832	466	45	2380
6	46	21	283	126	42	753	300	47	1817
7	52	26	391	116	52	956	249	61	1716
8	49	26	384	99	45	927	188	43	1431
ode45	19	7	157	89	32	727	387	44	2587

Example 5.2. The Brusselator problem [14, p. 170]).

$$\begin{cases} y_1' = 1 + y_1^2 y_2 - 4y_1, & y_1(0) = 1.5, \\ y_2' = 3y_1 - y_1^2 y_2, & y_2(0) = 3, \end{cases} \tag{5.2}$$

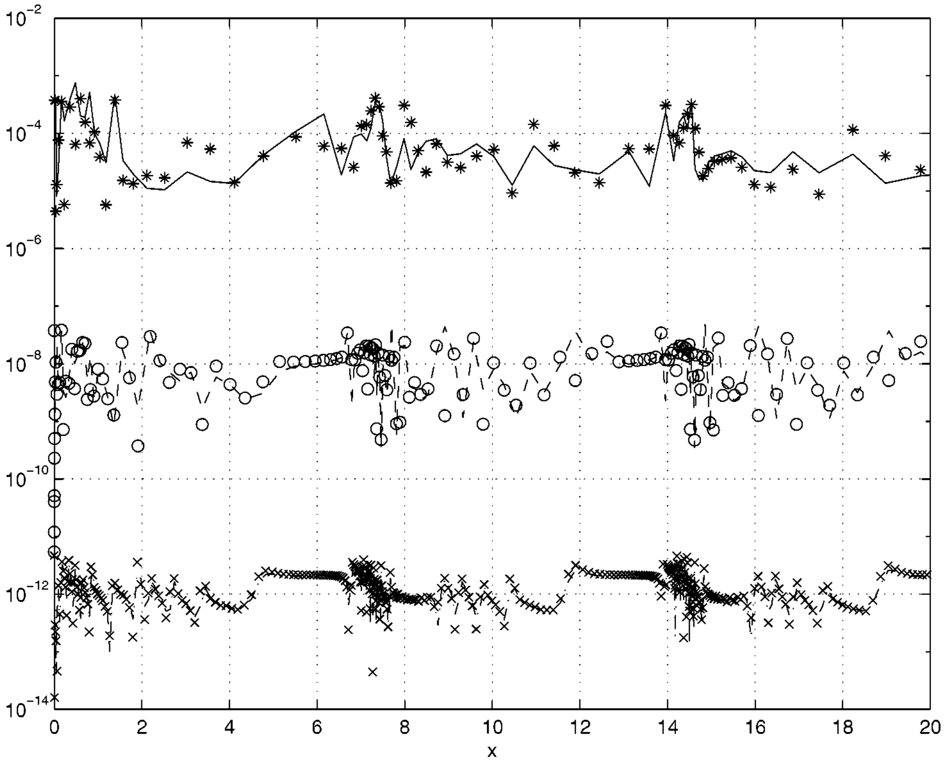

Figure 3. Local error versus local error estimate for the problem (5.2).

Table 2
Numerical results for the example 5.2.

tol	10^{-4}			10^{-8}			10^{-12}		
$p_{\max}$	*ns*	*nr*	*nfe*	*ns*	*nr*	*nfe*	*ns*	*nr*	*nfe*
4	101	37	538	570	34	2402	3518	16	14130
5	95	35	599	340	29	1820	1515	19	7655
6	89	21	619	239	48	1676	826	29	5103
7	78	20	636	186	50	1575	535	35	3938
8	75	17	674	153	45	1498	377	27	3173
`ode45`	64	17	487	336	7	2059	2089	1	12541

$x \in [0, 20]$. The local errors and local error estimates are plotted in figure 3 and step size and order patterns in figure 4 for *tol* $= 10^{-4}$, 10^{-8}, and 10^{-12}. The cost statistics (*ns*, *nr*, *nfe*) for this problem are listed in table 2.

Example 5.3. A celestial mechanics problem "the Pleiades" [14, p. 245]. This problem describes the movement of seven stars in the plane with coordinates x_i, y_i, and masses

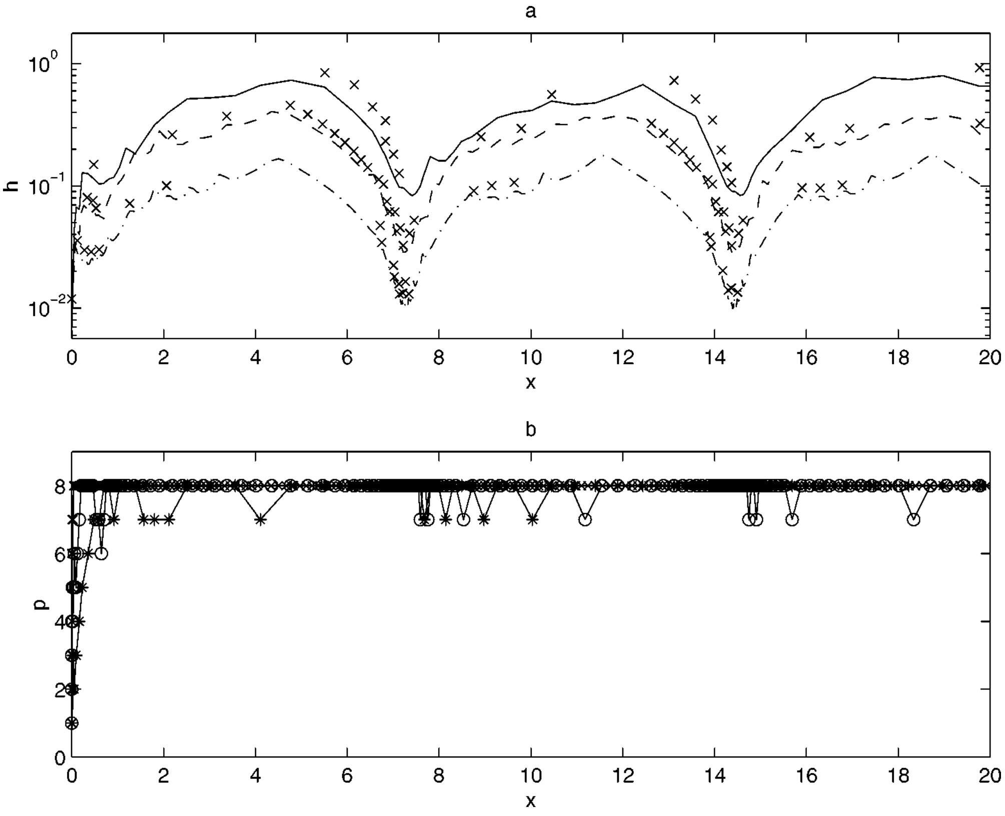

Figure 4. Step size control (a) and order control (b) for the problem (5.2).

$m_i = i,\ i = 1, 2, \ldots, 7$. The equations of motion are

$$\begin{cases} x_i'' = \sum_{j \neq i} m_j(x_j - x_i)/r_{ij}, \\ y_i'' = \sum_{j \neq i} m_j(y_j - y_i)/r_{ij}, \end{cases} \tag{5.3}$$

$t \in [0, 3]$, where

$$r_{ij} = \left((x_i - x_j)^2 + (y_i - y_j)^2\right)^{3/2}.$$

The initial values are

$$\begin{array}{llll} x_1(0) = 3, & x_2(0) = 3, & x_3(0) = -1, & x_4(0) = -3, \\ x_5(0) = 2, & x_6(0) = -2, & x_7(0) = 2, & \\ y_1(0) = 3, & y_2(0) = -3, & y_3(0) = 2, & y_4(0) = 0, \\ y_5(0) = 0, & y_6(0) = -4, & y_7(0) = 4, & \end{array}$$

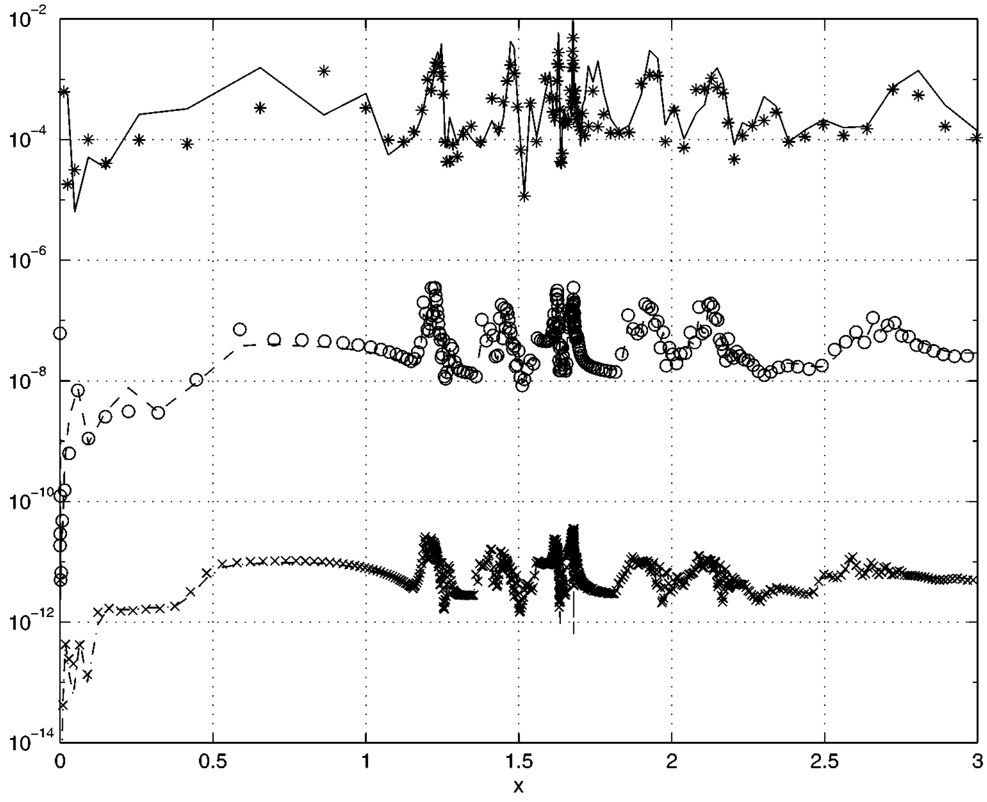

Figure 5. Local error versus local error estimate for the problem (5.3).

Table 3
Numerical results for the example 5.3.

tol	10^{-4}			10^{-8}			10^{-12}		
$p_{\max}$	*ns*	*nr*	*nfe*	*ns*	*nr*	*nfe*	*ns*	*nr*	*nfe*
4	176	62	947	1025	1	4099	6457	1	25821
5	138	48	919	595	8	3003	2749	1	13729
6	122	45	984	411	56	2781	1501	1	8977
7	111	37	1001	315	75	2698	955	1	6640
8	109	30	1027	254	69	2540	670	6	5337
ode45	98	23	727	576	0	3457	3651	0	21907

$x_i'(0) = y_i'(0) = 0$, for all i with the exception of

$$x_6'(0) = 1.75, \qquad x_7'(0) = -1.5, \qquad y_4'(0) = -1.25, \qquad y_5'(0) = 1.$$

The local errors and local error estimates are plotted in figure 5 and step size and order patterns in figure 6 for *tol* $= 10^{-4}$, 10^{-8}, and 10^{-12}. The cost statistics (*ns*, *nr*, *nfe*) for this problem are listed in table 3.

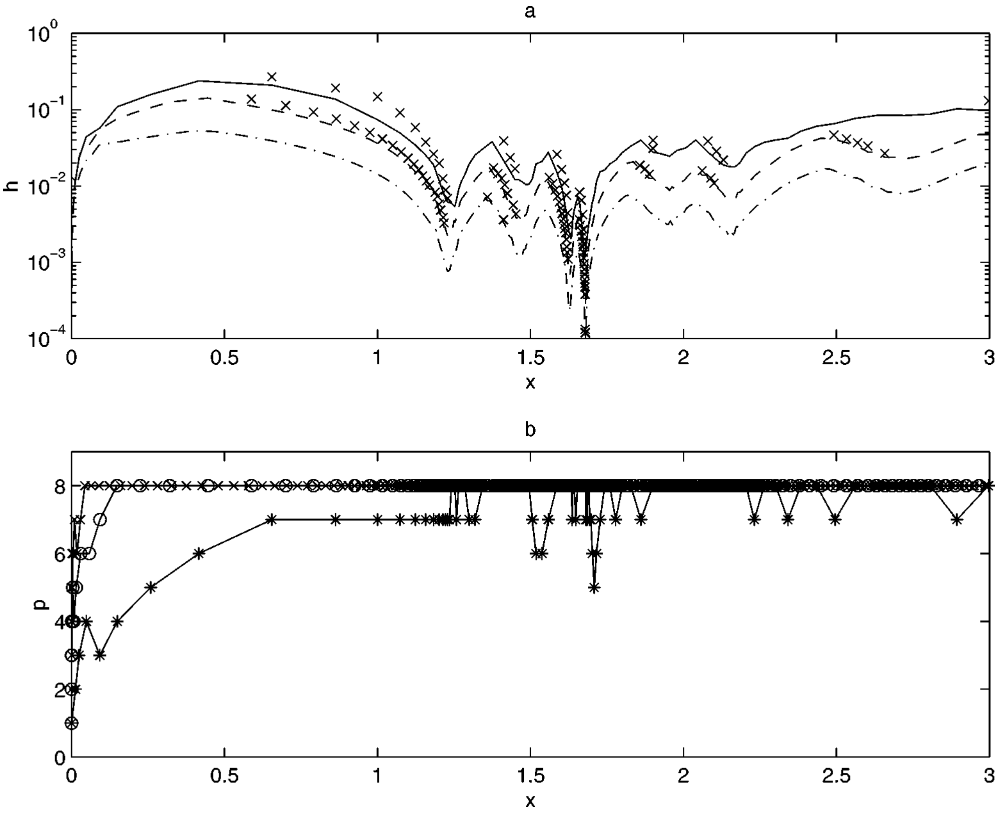

Figure 6. Step size control (a) and order control (b) for the problem (5.3).

Analyzing the plots in figures 1–3 we can see that the local error estimators employed in the code are very reliable, especially for higher tolerances (10^{-8}, 10^{-12}). Figures 2, 4, and 6 demonstrate that step size and order changing strategy described in section 4.3 are very robust and are able to detect automatically the discontinuities in the derivatives of the solution.

The numerical results presented in tables 1–3 demonstrate that the code based on DIMSIMs with the maximum order $p_{\max}$ restricted to 5 is less efficient than the `ode45` code for low tolerance (10^{-4}) but it is more efficient for moderate (10^{-8}) and high (10^{-12}) tolerances. It is more efficient on problems (5.2) and (5.3) by a factor of about 1.14 for *tol* $= 10^{-8}$ and by a factor of about 1.6 for *tol* $= 10^{-12}$. On the problem (5.1) our code is less efficient for *tol* $= 10^{-8}$ by a factor of 1.14 and more efficient for *tol* $= 10^{-12}$ by a factor of 1.1. The reason our code is not as competitive on this problem as it is on problems (5.2) and (5.3) is that it drops its order drastically in the vicinity of $x = -1$, $x = 0$, and $x = 1$ (down to 1 in the neighborhood of $x = 0$) and, as a consequence, it has to take many small steps with methods of low orders.

The code based on DIMSIMs with $p_{\text{max}} = 8$ is still less efficient than `ode45` for $tol = 10^{-4}$ but it is more efficient by a factor of 1.37 for $tol = 10^{-8}$ and by a factor of about 4 for $tol = 10^{-12}$ on problems (5.1) and (5.2). On problem (5.1) our code with $p_{\text{max}} = 8$ is less efficient for $tol = 10^{-4}$ and $tol = 10^{-8}$ but more efficient by a factor of 1.8 for $tol = 10^{-12}$.

Appendix

We list below DIMSIMs of order $5 \leqslant p \leqslant 8$ and the vectors $\beta(\delta_n)$ and $\gamma(\delta_n)$ appearing in the error estimating formulas. The coefficients are listed with six significant digits. The coefficients of these methods to the full precision can be obtained from the authors.

1. $p = q = s = 5$:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1.17653 & 0 & 0 & 0 & 0 \\ 1.98058 & 0.401718 & 0 & 0 & 0 \\ 3.05328 & 0.734996 & 0.267236 & 0 & 0 \\ 1.41933 & 2.65349 & -2.27785 & 1.19789 & 0 \end{bmatrix},$$

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -0.926528 & 0.03125 & 0.00260417 & 0.00016276 & 8.13802\text{E}{-6} \\ 1 & -1.8823 & 0.0245705 & 0.00827964 & 0.00155803 & 0.000195033 \\ 1 & -3.30552 & -0.0361169 & 0.0139394 & 0.00570213 & 0.00116198 \\ 1 & -1.99286 & 0.0771363 & 0.0315701 & -0.00201486 & -0.00195914 \end{bmatrix},$$

$$G = \begin{bmatrix} 3.16302 & 1.97434 & -0.810425 & 0.540922 & 0.0550787 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & -\dfrac{16}{3} & 12 & -16 & \dfrac{25}{3} \\ \dfrac{44}{3} & -\dfrac{224}{3} & 152 & -\dfrac{416}{3} & \dfrac{140}{3} \\ 96 & -448 & 768 & -576 & 160 \\ 256 & -1024 & 1536 & -1024 & 256 \end{bmatrix},$$

$$q^{\mathrm{T}} = [1 \ \ -3.92294 \ \ -0.0491424 \ \ 0.0265981 \ \ 0.00619566 \ \ 0.000696231].$$

There is a four parameter family of error estimates depending on the parameters γ_2, γ_3, γ_4 and γ_5. We list below the error estimate corresponding to $\gamma_2 = \gamma_3 =$

$\gamma_4 = \gamma_5 = 0$:

$$\begin{aligned}
&\beta_1 = 0.711111\,\delta_n/(1+\delta_n), && \beta_2 = -2.84444\,\delta_n/(1+\delta_n),\\
&\beta_3 = 4.26667\,\delta_n/(1+\delta_n), && \beta_4 = -2.84444\,\delta_n/(1+\delta_n),\\
&\beta_5 = 0.711111\,\delta_n/(1+\delta_n), && \gamma_1 = 0,\\
&\gamma_6 = -0.00277778\,\delta_n^6/(1+\delta_n).
\end{aligned}$$

This method corresponds to DIMSIM of type 1 of order $p = 5$ with $c = [0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1]^{\mathrm{T}}$ derived in [9].

2. $p = q = s = 6$:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0\\ -0.378732 & 0 & 0 & 0 & 0 & 0\\ 3.41547 & 1.07585 & 0 & 0 & 0 & 0\\ 3.51745 & 0.49391 & 0.483207 & 0 & 0 & 0\\ -0.398097 & -1.98829 & 1.04384 & 0.253273 & 0 & 0\\ 1.34887 & -2.74161 & 1.38831 & 0.032758 & 0.364587 & 0 \end{bmatrix},$$

$$P = \begin{bmatrix} e & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & \alpha_6 \end{bmatrix},$$

$$\alpha_1 = \begin{bmatrix} 0 & 0.578732 & -4.09133 & -3.89457 & 1.88927 & 0.607086 \end{bmatrix}^{\mathrm{T}},$$

$$\alpha_2 = \begin{bmatrix} 0 & 0.578732 & -4.09133 & -3.89457 & 1.88927 & 0.607086 \end{bmatrix}^{\mathrm{T}},$$

$$\alpha_3 = \begin{bmatrix} 0 & 0.578732 & -4.09133 & -3.89457 & 1.88927 & 0.607086 \end{bmatrix}^{\mathrm{T}},$$

$$\alpha_4 = \begin{bmatrix} 0 & 0.578732 & -4.09133 & -3.89457 & 1.88927 & 0.607086 \end{bmatrix}^{\mathrm{T}},$$

$$\alpha_5 = \begin{bmatrix} 0 & 0.578732 & -4.09133 & -3.89457 & 1.88927 & 0.607086 \end{bmatrix}^{\mathrm{T}},$$

$$\alpha_6 = \begin{bmatrix} 0 & 0.578732 & -4.09133 & -3.89457 & 1.88927 & 0.607086 \end{bmatrix}^{\mathrm{T}},$$

$$G = \begin{bmatrix} 5.26063 & -1.61659 & 1.67527 & -0.607677 & 0.73622 & -0.0763826\\ 0 & 0 & 0 & 0 & 0 & 1\\ -1 & \frac{25}{4} & -\frac{50}{3} & 25 & -25 & \frac{137}{12}\\ -\frac{125}{6} & \frac{1525}{12} & -325 & \frac{2675}{6} & -\frac{1925}{6} & \frac{375}{4}\\ -\frac{875}{4} & \frac{5125}{4} & -\frac{6125}{2} & \frac{7375}{2} & -\frac{8875}{4} & \frac{2125}{4}\\ -1250 & 6875 & -15000 & 16250 & -8750 & 1875\\ -3125 & 15625 & -31250 & 31250 & -15625 & 3125 \end{bmatrix},$$

$q_1 = 1,\qquad q_2 = -4.37146,\qquad q_3 = 0.00522456,\qquad q_4 = -0.0230401,$

$q_5 = -0.0022647,\qquad q_6 = 0.000553397,\qquad q_7 = 0.000270169.$

There is a five parameter family of error estimates depending on the parameters γ_2, γ_3, γ_4, γ_5 and γ_6. We list below the error estimate corresponding to $\gamma_2 = \gamma_3 = \gamma_4 = \gamma_5 = \gamma_6 = 0$:

$$\begin{array}{ll}
\beta_1 = -1.2655\,\delta_n/(1+\delta_n), & \beta_2 = 6.32794\,\delta_n/(1+\delta_n), \\
\beta_3 = -12.6559\,\delta_n/(1+\delta_n), & \beta_4 = 12.6559\,\delta_n/(1+\delta_n), \\
\beta_5 = -6.32794\,\delta_n/(1+\delta_n), & \beta_6 = 1.26559\,\delta_n/(1+\delta_n), \\
\gamma_1 = 0, & \gamma_6 = -0.000404988\,\delta_n^7/(1+\delta_n).
\end{array}$$

This method corresponds to DIMSIM of type 1 of order $p = 6$ with $c = [0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1]^{\mathrm{T}}$ derived in [9].

3. $p = q = s = 7$:

$$\begin{array}{llll}
a_{21} = -0.124323, & a_{31} = -0.20081, & a_{32} = 0.439157, & a_{41} = 0.086166, \\
a_{42} = -0.0403638, & a_{43} = 0.421511, & a_{51} = 0.501295, & a_{52} = -0.478549, \\
a_{53} = 0.381186, & a_{54} = 0.32732, & a_{61} = 0.796465, & a_{62} = -0.895974, \\
a_{63} = 0.756491, & a_{64} = -0.0901947, & a_{65} = 0.387279, & a_{71} = 0.594258, \\
a_{72} = 0.157098, & a_{73} = 0.0812975, & a_{74} = 0.11216, & a_{75} = 0.155676, \\
a_{76} = 0.338477, & & &
\end{array}$$

$$\begin{array}{lll}
p_{11} = 1, & p_{12} = 0, & p_{13} = 0, \\
p_{14} = 0, & p_{15} = 0, & p_{16} = 0, \\
p_{17} = 0, & p_{18} = 0, & p_{21} = 1, \\
p_{22} = 0.29099, & p_{23} = 0.0138889, & p_{24} = 0.000771605, \\
p_{25} = 0.0000321502, & p_{26} = 1.07167\mathrm{E}{-6}, & p_{27} = 2.97687\mathrm{E}{-8}, \\
p_{28} = 7.08779\mathrm{E}{-10}, & p_{31} = 1, & p_{32} = 0.094986, \\
p_{33} = -0.0176373, & p_{34} = 0.0000734368, & p_{35} = 0.000175548, \\
p_{36} = 0.0000201746, & p_{37} = 1.43456\mathrm{E}{-6}, & p_{38} = 7.76505\mathrm{E}{-8}, \\
p_{41} = 1, & p_{42} = 0.032687, & p_{43} = -0.00877631, \\
p_{44} = -0.00202333, & p_{45} = 0.0000333928, & p_{46} = 0.0000448878, \\
p_{47} = 7.28954\mathrm{E}{-6}, & p_{48} = 7.48239\mathrm{E}{-7}, & p_{51} = 1, \\
p_{52} = -0.0645859, & p_{53} = 0.0112582, & p_{54} = -0.00606282, \\
p_{55} = -0.000572472, & p_{56} = 0.0000642989, & p_{57} = 0.0000241336, \\
p_{58} = 3.79733\mathrm{E}{-6}, & p_{61} = 1, & p_{62} = -0.120732, \\
p_{63} = 0.031299, & p_{64} = -0.00792023, & p_{65} = -0.00113031, \\
p_{66} = 0.0000360447, & p_{67} = 0.0000386442, & p_{68} = 8.69415\mathrm{E}{-6}, \\
p_{71} = 1, & p_{72} = -0.438968, & p_{73} = 0.0047892, \\
p_{74} = -0.00617332, & p_{75} = -0.0016271, & p_{76} = -0.0000882244, \\
p_{77} = 0.0000523332, & p_{78} = 0.0000193992, &
\end{array}$$

$g_{11} = -0.258331,\quad g_{12} = 2.6467,\quad g_{13} = -2.14768,\quad g_{14} = 1.93169,$
$g_{15} = -1.49763,\quad g_{16} = 1.26187,\quad g_{17} = -0.200936,\quad g_{21} = 0,$
$g_{22} = 0,\quad g_{23} = 0,\quad g_{24} = 0,\quad g_{25} = 0,$
$g_{26} = 0,\quad g_{27} = 1,\quad g_{31} = 1,\quad g_{32} = -\frac{36}{5},$
$g_{33} = \frac{45}{2},\quad g_{34} = -40,\quad g_{35} = 45,\quad g_{36} = -36,$
$g_{37} = \frac{147}{10},\quad g_{41} = \frac{137}{5},\quad g_{42} = -\frac{972}{5},\quad g_{43} = 594,$
$g_{44} = -1016,\quad g_{45} = 1053,\quad g_{46} = -\frac{3132}{5},\quad g_{47} = \frac{812}{5},$
$g_{51} = 405,\quad g_{52} = -2808,\quad g_{53} = 8289,\quad g_{54} = -13392,$
$g_{55} = 12447,\quad g_{56} = -6264,\quad g_{57} = 1323,\quad g_{61} = 3672,$
$g_{62} = -24624,\quad g_{63} = 69336,\quad g_{64} = -104544,\quad g_{65} = 88776,$
$g_{66} = -40176,\quad g_{67} = 7560,\quad g_{71} = 19440,\quad g_{72} = -124416,$
$g_{73} = 330480,\quad g_{74} = -466560,\quad g_{75} = 369360,\quad g_{76} = -155520,$
$g_{77} = 27216,\quad g_{81} = 46656,\quad g_{82} = -279936,\quad g_{83} = 699840,$
$g_{84} = -933120,\quad g_{85} = 699840,\quad g_{86} = -279936,\quad g_{87} = 46656,$

$q_1 = 1,\quad q_2 = -0.735682,\quad q_3 = -0.0432704,\quad q_4 = 0.00288702,$
$q_5 = -0.00162336,\quad q_6 = -0.000334725,\quad q_7 = 0.0000486512,$
$q_8 = 0.0000352537.$

There is a six parameter family of error estimates depending on the parameters γ_2, γ_3, γ_4, γ_5, γ_6 and γ_7. We list below the error estimate corresponding to $\gamma_2 = \gamma_3 = \gamma_4 = \gamma_5 = \gamma_6 = \gamma_7 = 0$:

$$\begin{aligned}
&\beta_1 = 2.32589\,\delta_n/(1+\delta_n), && \beta_2 = -13.9553\,\delta_n/(1+\delta_n),\\
&\beta_3 = 34.8884\,\delta_n/(1+\delta_n), && \beta_4 = -46.5178\,\delta_n/(1+\delta_n),\\
&\beta_5 = 34.8884\,\delta_n/(1+\delta_n), && \beta_6 = -13.9553\,\delta_n/(1+\delta_n),\\
&\beta_7 = 2.32589\,\delta_n/(1+\delta_n), && \gamma_1 = 0,\\
&\gamma_8 = -0.0000498519\,\delta_n^8/(1+\delta_n).
\end{aligned}$$

This method corresponds to DIMSIM of type 1 of order $p = 7$ with $c = [0, \frac{1}{6}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{5}{6}, 1]^{\mathrm{T}}$ derived in [10].

4. $p = q = s = 8$:

$a_{21} = 1.28603,\quad a_{31} = -8.10798,\quad a_{32} = 0.528618,\quad a_{41} = -13.9361,$
$a_{42} = -0.120319,\quad a_{43} = 0.394758,\quad a_{51} = -14.6664,\quad a_{52} = -1.33405,$
$a_{53} = 0.401456,\quad a_{54} = 0.296028,\quad a_{61} = -15.3662,\quad a_{62} = -3.4188,$
$a_{63} = 0.655027,\quad a_{64} = 0.194148,\quad a_{65} = 0.269685,\quad a_{71} = -21.5545,$
$a_{72} = -5.70376,\quad a_{73} = 0.52958,\quad a_{74} = 0.753453,\quad a_{75} = -0.148158,$
$a_{76} = 0.325528,\quad a_{81} = -29.08,\quad a_{82} = -4.96428,\quad a_{83} = -2.2828,$
$a_{84} = 3.53319,\quad a_{85} = -1.39368,\quad a_{86} = 0.348955,\quad a_{87} = 0.295945,$

$$\begin{array}{lll}
p_{11} = 1, & p_{12} = 0, & p_{13} = 0, \\
p_{14} = 0, & p_{15} = 0, & p_{16} = 0, \\
p_{17} = 0, & p_{18} = 0, & p_{19} = 0, \\
p_{21} = 1, & p_{22} = -1.14317, & p_{23} = 0.0102041, \\
p_{24} = 0.000485909, & p_{25} = 0.0000173539, & p_{26} = 4.95825\text{E}{-7}, \\
p_{27} = 1.18054\text{E}{-8}, & p_{28} = 2.40926\text{E}{-10}, & p_{29} = 4.30225\text{E}{-12}, \\
p_{31} = 1, & p_{32} = 7.86507, & p_{33} = -0.0347006, \\
p_{34} = -0.0015068, & p_{35} = 0.0000208019, & p_{36} = 6.69283\text{E}{-6}, \\
p_{37} = 4.93441\text{E}{-7}, & p_{38} = 2.4598\text{E}{-8}, & p_{39} = 9.74017\text{E}{-10}, \\
p_{41} = 1, & p_{42} = 14.0902, & p_{43} = -0.00376269, \\
p_{44} = -0.00176527, & p_{45} = -0.0000704005, & p_{46} = 0.0000129643, \\
p_{47} = 2.40238\text{E}{-6}, & p_{48} = 2.30069\text{E}{-7}, & p_{49} = 1.60823\text{E}{-10}, \\
p_{51} = 1, & p_{52} = 15.8744, & p_{53} = 0.112274, \\
p_{54} = 0.00113875, & p_{55} = -0.000353493, & p_{56} = 3.2911\text{E}{-6}, \\
p_{57} = 6.97949\text{E}{-6}, & p_{58} = 1.11211\text{E}{-6}, & p_{59} = 1.13915\text{E}{-7}, \\
p_{61} = 1, & p_{62} = 18.3805, & p_{63} = 0.319039, \\
p_{64} = 0.0070283, & p_{65} = -0.000972717, & p_{66} = -0.0000441027, \\
p_{67} = 0.0000154429, & p_{68} = 3.65635\text{E}{-6}, & p_{69} = 4.94354\text{E}{-7}, \\
p_{71} = 1, & p_{72} = 26.655, & p_{73} = 0.560096, \\
p_{74} = 0.0134941, & p_{75} = -0.00184609, & p_{76} = -0.000124154, \\
p_{77} = 0.0000252694, & p_{78} = 7.74437\text{E}{-6}, & p_{79} = 1.2718\text{E}{-6}, \\
p_{81} = 1, & p_{82} = 34.5427, & p_{83} = 0.640655, \\
p_{84} = 0.0158275, & p_{85} = -0.00231651, & p_{86} = -0.000162404, \\
p_{87} = 0.0000277633, & p_{88} = 9.80871\text{E}{-6}, & p_{89} = 1.98507\text{E}{-6},
\end{array}$$

$$\begin{array}{llll}
g_{11} = -29.346, & g_{12} = -5.26245, & g_{13} = -2.10258, & g_{14} = 3.55214, \\
g_{15} = -1.58397, & g_{16} = 0.531577, & g_{17} = 0.220292, & g_{18} = 0.0119837, \\
g_{21} = 0, & g_{22} = 0, & g_{23} = 0, & g_{24} = 0, \\
g_{25} = 0, & g_{26} = 0, & g_{27} = 0, & g_{28} = 1, \\
g_{31} = -1, & g_{32} = \frac{49}{6}, & g_{33} = -\frac{147}{5}, & g_{34} = \frac{245}{4}, \\
g_{35} = -\frac{245}{3}, & g_{36} = \frac{147}{2}, & g_{37} = -49, & g_{38} = \frac{363}{20}, \\
g_{41} = -\frac{343}{10}, & g_{42} = \frac{49931}{180}, & g_{43} = -\frac{9849}{10}, & g_{44} = 2009, \\
g_{45} = -\frac{46501}{18}, & g_{46} = \frac{43071}{20}, & g_{47} = -\frac{10927}{10}, & g_{48} = \frac{22981}{90}, \\
g_{51} = -\frac{9947}{15}, & g_{52} = \frac{634207}{120}, & g_{53} = -\frac{91924}{5}, & g_{54} = \frac{872935}{24}, \\
g_{55} = -\frac{133427}{3}, & g_{56} = \frac{1347647}{40}, & g_{57} = -\frac{218834}{15}, & g_{58} = \frac{331681}{120}, \\
g_{61} = -\frac{16807}{2}, & g_{62} = \frac{196882}{3}, & g_{63} = -\frac{444185}{2}, & g_{64} = 422576, \\
g_{65} = -\frac{2926819}{6}, & g_{66} = 340942, & g_{67} = -\frac{266511}{2}, & g_{68} = \frac{67228}{3}, \\
g_{71} = -\frac{420175}{6}, & g_{72} = \frac{1596665}{3}, & g_{73} = -\frac{3479049}{2}, & g_{74} = \frac{9495955}{3}, \\
g_{75} = -\frac{20756645}{6}, & g_{76} = 2268945, & g_{77} = -\frac{4958065}{6}, & g_{78} = \frac{386561}{3}, \\
g_{81} = -352947, & g_{82} = 2588278, & g_{83} = -8117781, & g_{84} = 14117880,
\end{array}$$

$$\begin{array}{llll}
g_{85} = -14706125, & g_{86} = 9176622, & g_{87} = -3176523, & g_{88} = 470596,\\
g_{91} = -823543, & g_{92} = 5764801, & g_{93} = -17294403, & g_{94} = 28824005,\\
g_{95} = -28824005, & g_{96} = 17294403, & g_{97} = -5764801, & g_{98} = 823543,
\end{array}$$

$$\begin{array}{llll}
q_1 = 1, & q_2 = 34.979, & q_3 = 0.654797, & q_4 = 0.0160543,\\
q_5 = -0.00235222, & q_6 = -0.000167098, & q_7 = 0.0000282391, & \\
q_8 = 0.0000100532, & q_9 = 2.00797\mathrm{E}{-6}. & &
\end{array}$$

There is a seven parameter family of error estimates depending on the parameters γ_2, γ_3, γ_4, γ_5, γ_6, γ_7 and γ_8. We list below the error estimate corresponding to $\gamma_2 = \gamma_3 = \gamma_4 = \gamma_5 = \gamma_6 = \gamma_7 = \gamma_8 = 0$:

$$\begin{array}{ll}
\beta_1 = -4.31517\,\delta_n/(1+\delta_n), & \beta_2 = 30.2062\,\delta_n/(1+\delta_n),\\
\beta_3 = -90.6185\,\delta_n/(1+\delta_n), & \beta_4 = 151.031\,\delta_n/(1+\delta_n),\\
\beta_5 = -151.031\,\delta_n/(1+\delta_n), & \beta_6 = 90.6185\,\delta_n/(1+\delta_n),\\
\beta_7 = -30.2062\,\delta_n/(1+\delta_n), & \beta_8 = 4.31517\,\delta_n/(1+\delta_n),\\
\gamma_1 = 0, & \gamma_8 = -5.23976 \cdot 10^{-6}\,\delta_n^9/(1+\delta_n).
\end{array}$$

This method corresponds to DIMSIM of type 1 of order $p = 8$ with $c = [0, \frac{1}{7}, \frac{2}{7}, \frac{3}{7}, \frac{4}{7}, \frac{5}{7}, \frac{6}{7}, 1]^{\mathrm{T}}$ derived in [10].

References

[1] J.C. Butcher, *The Numerical Analysis of Ordinary Differential Equations* (Wiley, Chichester/New York, 1987).

[2] J.C. Butcher, Diagonally-implicit multi-stage integration methods, Appl. Numer. Math. 11 (1993) 347–363.

[3] J.C. Butcher, A transformation for the analysis of DIMSIMs, BIT 34 (1994) 25–32.

[4] J.C. Butcher and P. Chartier, The construction of DIMSIMs for stiff ODEs and DAEs, Report Series No. 308, The University of Auckland, New Zealand (July 1994).

[5] J.C. Butcher, P. Chartier and Z. Jackiewicz, Nordsieck representation of DIMSIMs, Numer. Algorithms 16 (1997) 209–230.

[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, SIAM J. Numer. Anal. 34 (1997) 2119–2141.

[9] J.C. Butcher and Z. Jackiewicz, Construction of high order diagonally implicit multistage integration methods for ordinary differential equations, Appl. Numer. Math. 27 (1998) 1–12.

[10] J.C. Butcher, Z. Jackiewicz and H.D. Mittelmann, Nonlinear optimization approach to construction of general linear methods of high order, J. Comput. Appl. Math. 81 (1997) 181–196.

[11] P. Chartier, The potential of parallel multi-value methods for the simulation of large real-life problems, CWI Quarterly 11 (1998) 7–32.

[12] J.R. Dormand and P.J. Prince, A family of embedded Runge–Kutta formulae, J. Comput. Appl. Math. 6 (1980) 19–26.

[13] I. Gladwell, L.F. Shampine and R.W. Brankin, Automatic selection of the initial step size for an ODE solver, J. Comput. Appl. Math. 18 (1987) 175–192.

[14] E. Hairer, S.P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations I. Nonstiff Problems* (Springer, Berlin, 1993).

[15] L.F. Shampine, *Numerical Solution of Ordinary Differential Equations* (Chapman Hall, New York/London, 1994).

[16] L.F. Shampine and M.W. Reichelt, The Matlab ODE suite, SIAM J. Sci. Comput. 18 (1977) 1–22.

[17] W.M. Wright, The construction of order 4 DIMSIM for ordinary differential equations, submitted.

[18] W.M. Wright, General linear methods for ordinary differential equations. Theoretical and practical investigation, M.Sc. thesis, The University of Auckland, New Zealand (1999).