APPLICATIONES MATHEMATICAE
28,3 (2001), pp. 353–365

Zbigniew Bartoszewski (Gdańsk)
Zdzisław Jackiewicz (Tempe, AZ)

TOWARD A TWO-STEP RUNGE–KUTTA CODE FOR NONSTIFF DIFFERENTIAL SYSTEMS

Abstract. Various issues related to the development of a new code for nonstiff differential equations are discussed. This code is based on two-step Runge–Kutta methods of order five and stage order five. Numerical experiments are presented which demonstrate that the new code is competitive with the Matlab `ode45` program for all tolerances.

1. Introduction. A new code for nonstiff ordinary differential equations (ODEs)

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

$f : \mathbb{R}^m \to \mathbb{R}^m$, is described. This code is based on variable step size two-step Runge–Kutta (TSRK) method of order $p = 5$ and stage order $q = 5$ constructed in [1]. A general class of TSRK methods was introduced recently by Jackiewicz and Tracogna [12] and further investigated in [5], [9], [11], [13], [14], [17], and [18]. On a nonuniform grid

$$x_0 < x_1 < \ldots < x_N, \quad x_N \geq X,$$

these methods with coefficients $\eta \in \mathbb{R}$, $u, v, w \in \mathbb{R}^s$, and $A, B \in \mathbb{R}^{s\times s}$ take the form

$$\begin{cases} Y^{[n]} = (u \otimes I_m)\widetilde{y}_{n-1} + ((e-u) \otimes I_m)y_n \\ \qquad\qquad + h_{n+1}((A \otimes I_m)F(\widetilde{Y}^{[n-1]}) + (B \otimes I_m)F(Y^{[n]})), \\ y_{n+1} = \eta\widetilde{y}_{n-1} + (1-\eta)y_n \\ \qquad\qquad + h_{n+1}((v^T \otimes I_m)F(\widetilde{Y}^{[n-1]}) + (w^T \otimes I_m)F(Y^{[n]})), \end{cases} \tag{1.2}$$

2000 *Mathematics Subject Classification*: 65L05, 65L06.

Key words and phrases: two-step Runge–Kutta methods, starting procedure, local error estimation, step changing strategy.

$n = 1, \ldots, N-1$, where $h_{n+1} = x_{n+1} - x_n$, $\widetilde{y}_0 = y_0$, and

$$(1.3)\qquad \begin{cases} F(\widetilde{Y}^{[n-1]}) = (\widetilde{V} \otimes I_m) F(\widetilde{Y}^{[n-2]}) + (\widetilde{W} \otimes I_m) F(Y^{[n-1]}), \\ \qquad \widetilde{y}_{n-1} = y_{n-1} + (\widetilde{v} \otimes I_m) h_{n+1} F(\widetilde{Y}^{[n-2]}) \\ \qquad\qquad + (\widetilde{w} \otimes I_m) h_{n+1} F(Y^{[n-1]}), \end{cases}$$

$n = 2, \ldots, N$ (compare [17]). Denote by y the exact solution to (1.1). Then y_n denotes an approximation to $y(x_n)$, $Y^{[n]} \in \mathbb{R}^{m\cdot s}$, $Y^{[n]} = [Y_1^{[n]}, \ldots, Y_s^{[n]}]^T$ is the vector of stage values and its coordinates $Y_j^{[n]}$ are approximations to $y(x_n + c_j h_{n+1})$, $F(Y^{[n]}) \in \mathbb{R}^{m\cdot s}$, $F(Y^{[n]}) = [f(Y_1^{[n]}), \ldots, f(Y_s^{[n]})]^T$, $\widetilde{y}_{n-1}$ is an approximation to $y(x_n - h_{n+1})$, $F(\widetilde{Y}^{[n-1]}) = [f(\widetilde{Y}_1^{[n-1]}), \ldots, f(\widetilde{Y}_s^{[n-1]})]^T$, where stage values $\widetilde{Y}_j^{[n-1]}$ are approximations to $y(x_n - h_{n+1} + c_j h_{n+1})$, $\widetilde{V}, \widetilde{W} \in \mathbb{R}^{(p+1)\times s}$, $\widetilde{V} = \widetilde{G}\widetilde{D}TV$, $\widetilde{W} = \widetilde{G}\widetilde{D}TW$, $\widetilde{v}^T, \widetilde{w}^T \in \mathbb{R}^{p+1}$, $\widetilde{v} = \widetilde{\varDelta}V$, $\widetilde{w} = \widetilde{\varDelta}W$, $\widetilde{\varDelta} = \varDelta/\delta$, $\widetilde{D} = D/\delta$ and G and $\widetilde{G}$ are matrices defined by

$$G = \begin{bmatrix} e & c & \ldots & \dfrac{c^p}{p!} \end{bmatrix} \quad\text{and}\quad \widetilde{G} = \begin{bmatrix} e & c-e & \ldots & \dfrac{(c-e)^p}{p!} \end{bmatrix}.$$

Here, $e = [1, 1, \ldots, 1]^T \in \mathbb{R}^s$, $c = [c_1, \ldots, c_s]^T$ is the vector of stage abscissas, c^p means componentwise multiplication,

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

D and $\varDelta$ are the rescaling matrix and vector given by

$$D = \operatorname{diag}(\delta, \delta^2, \ldots, \delta^{p+1}), \quad \delta = h_{n+1}/h_n,$$
$$\varDelta = \begin{bmatrix} 1-\delta & \dfrac{(1-\delta)^2}{2!} & \ldots & \dfrac{(1-\delta)^{p+1}}{(p+1)!} \end{bmatrix},$$

and the matrices $V, W \in \mathbb{R}^{(p+1)\times s}$ are defined by the relation

$$(1.4)\qquad V\widetilde{G} + WG = I_{p+1},$$

the so-called compatibility conditions

$$(1.5)\qquad \widetilde{G}TV = 0 \quad\text{and}\quad \widetilde{G}TW = I_s$$

and some additional conditions which will be given in the next section. It was explained in [17] that (1.4) and (1.5) imply that the rescaled quantities

$\widetilde{y}_n$ and $F(\widetilde{Y}^{[n]})$ are identical to $F(Y^{[n]})$ and y_n respectively if $\delta = 1$, i.e., if no step size change is performed. We can take advantage of this fact and put $h_2 = h_1$ to obtain $F(\widetilde{Y}^{[0]}) = F(Y^{[0]})$, $\widetilde{y}_0 = y_0$ and start the process with formulas (1.2) for $n = 2$. However, we should first apply a one-step method of order consistent with the order of the TSRK method to find y_1 and $Y^{[0]}$. But if the step h_2 is too large we should find $\widetilde{y}_0$ and $F(\widetilde{Y}^{[0]})$ by other methods, for example by the one-step method used to find y_1 and $Y^{[0]}$.

We will assume that the coefficient matrix A in (1.2) is strictly lower triangular, i.e., triangular with zero diagonal. This choice corresponds to the explicit methods (1.2)–(1.3) which are appropriate for nonstiff differential systems (1.1). In [12] and [1] we have also considered the diagonally implicit TSRK methods (with a constant element on the diagonal of A) which are appropriate for stiff differential systems.

Since TSRK methods depend on numerical values at two consecutive steps they are more difficult to implement than Runge–Kutta methods. Moreover, they also require a special starting procedure of sufficiently high order (see Section 4). However, these difficulties are offset by increased efficiency. For example, the formula used in this paper achieves order five with only four function evaluations per step. In contrast, to attain the same order with explicit Runge–Kutta methods would require at least six function evaluations in a current step. Additional advantages of TSRK formulas are the availability of asymptotically correct estimators of the local discretization error (see Section 3) and continuous interpolant of the same order (see [4], [13]) without any significant additional cost, which is not possible for Runge–Kutta methods.

The error constants $\widehat{C}_\nu$ and C_ν can be written in the form

$$\widehat{C}_\nu = \frac{1}{\nu!} - \frac{(-1)^\nu \eta}{\nu!} - \frac{v^T(c-e)^{\nu-1}}{(\nu-1)!} - \frac{w^T c^{\nu-1}}{(\nu-1)!},$$

$$C_\nu = \frac{c^\nu}{\nu!} - \frac{(-1)^\nu u}{\nu!} - \frac{A(c-e)^{\nu-1}}{(\nu-1)!} - \frac{Bc^{\nu-1}}{(\nu-1)!}.$$

It follows from the results of [12] and [13] that if

$$\widehat{C}_\nu = 0, \quad \nu = 1, \ldots, p, \tag{1.6}$$

$$C_\nu = 0, \quad \nu = 1, \ldots, p-1, \tag{1.7}$$

then the method (1.2)–(1.3) is convergent with order p and stage order $q = p$, i.e.,

$$\sup\{\|y(x_n) - y_n\| : 0 \le n \le N\} = O(h^p),$$

and

$$\sup\{\|y(x_n + ch_{n+1}) - Y^{[n]}\| : 0 \le n \le N\} = O(h^p),$$

$h = \max\{h_n : 1 \leq n \leq N\}$. We will refer to (1.6) and (1.7) as the *consistency conditions* and *stage consistency conditions*, respectively, of order p.

The organization of this paper is as follows. In Section 2 we review the construction of a TSRK method of order $p = q = 5$ with desirable stability properties [1] and the computation of the matrices V and W which appear in the definitions of $\widetilde{v}$ and $\widetilde{w}$, which in turn appear in formulas (1.3). The estimation of the principal part of the local discretization error of (1.2)–(1.3) is discussed in Section 3. In Section 4 we discuss the computation of the quantities $\widetilde{Y}_j^{[0]} \approx y(x_0 + c_j h_1)$ and $y_1 \approx y(x_1)$, which are required to start the TSRK method (1.2)–(1.3). In Section 5 we discuss the choice of the starting step size $h_1 = x_1 - x_0$. In Section 6 we describe the step size changing strategy which is based on the estimate of the local discretization error derived in Section 3. In Section 7 a selection of results of numerical experiments is presented. These results demonstrate that the new code described in this paper is competitive with the code `ode45` from the Matlab ODE suite developed by Shampine and Reichelt [16].

2. Construction of variable step size TSRK methods. It was explained in [1] that solving the system (1.6) with respect to v^T and w_s and then the system (1.7) with respect to a_{ij} leads to a family of methods of order $p = q$ with free parameters η, c_i, u_i, $i = 1, \ldots, s$, w_i, $i = 1, \ldots, s-1$, and b_{ij}, $i = 2, \ldots, s$, $j = 1, \ldots, i-1$. These free parameters are then chosen in such a way that the stability polynomial of the method is equal to the prescribed polynomial with desired stability properties. This leads to a large system of polynomial equations which can be solved by techniques based on least squares minimization. The solution process is carefully explained in [1] and in the case of $s = 4$ and $p = q = 5$ leads to the TSRK method whose coefficients up to six decimal places of accuracy are listed below.

$$c = [\,0.0426809 \quad 0.179134 \quad 0.514122 \quad 0.864807\,]^T,$$

$$\eta = 0, \quad u = [\,3.37416 \quad 2.77718 \quad 1.53983 \quad 0.337209\,]^T,$$

$$A = \begin{bmatrix} 0.149087 & 1.06305 & 1.06295 & 1.14175 \\ 0.148093 & 0.817564 & 0.959052 & 0.774195 \\ -0.504349 & 1.47770 & -0.0344121 & 0.446085 \\ -2.52101 & 4.54789 & -2.56605 & 1.11104 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.257408 & 0 & 0 & 0 \\ -0.118572 & 0.787496 & 0 & 0 \\ -1.23797 & 1.43006 & 0.438059 & 0 \end{bmatrix},$$

$$v = [\,0.359241 \quad -0.671283 \quad 0.456387 \quad -0.150115\,]^T,$$

$$w = [\,0.754482 \quad -0.763885 \quad 0.795484 \quad 0.219689\,]^T.$$

We compute next the matrices $V, W \in \mathbb{R}^{(p+1)\times s}$ by solving the system (1.4) and the compatibility conditions (1.5). In the case of $s = 4$ this leads to a four-parameter family of solutions depending on w_{53}, w_{54}, w_{63}, and w_{64}. These free parameters are then chosen to satisfy the additional conditions

$$V\,e = 0, \tag{2.1}$$

and

$$V\,C_p = 0, \tag{2.2}$$

which are necessary for a reliable control of the local discretization error. This is discussed in detail in [17] and will be briefly addressed in Section 3.

It is easy to program the solution process to (1.4), (1.5), (2.1), and (2.2) in MATHEMATICA, and the resulting unique solution V and W corresponding to the TSRK method listed at the beginning of this section is given below.

$$V = \begin{bmatrix} -0.0125838 & 0.0252922 & -0.0158426 & 0.00313423 \\ 0.391021 & -0.78851 & 0.495509 & -0.0980198 \\ -4.77228 & 9.75851 & -6.21523 & 1.22900 \\ 18.0402 & -39.7319 & 27.0261 & -5.33442 \\ 59.0730 & -89.4004 & 37.9363 & -7.60893 \\ -462.024 & 837.008 & -467.869 & 92.8851 \end{bmatrix},$$

$$W = \begin{bmatrix} 1.41174 & -0.463082 & 0.0577709 & -0.00643001 \\ -10.0916 & 11.5495 & -1.64588 & 0.187973 \\ 19.4568 & -30.9511 & 13.2262 & -1.73193 \\ 99.3719 & -132.842 & 35.1429 & -1.67266 \\ -214.337 & 346.433 & -176.504 & 44.4082 \\ -1408.30 & 2057.80 & -807.490 & 157.989 \end{bmatrix}.$$

Full precision versions of the coefficients of these matrices and of the TSRK method can be obtained from the authors.

3. Local error estimation. It was proved in [17] that the local discretization error $\varphi_p(x_{n+1})$ of the TSRK method (1.2)–(1.3) at the point x_{n+1} is given by

$$\varphi_p(x_{n+1}) \\ = \left(\widehat{C}_{p+1}y^{(p+1)}(x_n) + ((v+w)^T C_p)\frac{\partial f}{\partial y}(y(x_n))y^{(p)}(x_n)\right)h_{n+1}^{p+1} + O(h_{n+1}^{p+1}).$$

Moreover, if V and W satisfy (1.4), (1.5), (2.1), (2.2), and the condition

$$\widetilde{G}DTWC_p = \delta^{p+1}C_p, \tag{3.1}$$

then the principal part of this error can be estimated by the formula

$$\operatorname{est}(x_{n+1}) = h_{n+1}(\beta_1^T \otimes I_m)F(Y^{[n]}) + h_{n+1}(\beta_2^T \otimes I_m)F(\widetilde{Y}^{[n-1]}), \tag{3.2}$$

where the vectors $\beta_1, \beta_2 \in \mathbb{R}^s$ satisfy the system of equations

$$\begin{cases} (\beta_1+\beta_2)^T e = 0, \\ \beta_1^T c^{\nu-1} + \beta_2^T (c-e)^{\nu-1} = 0, \quad \nu = 2,3,\dots,p, \\ \beta_1^T \dfrac{c^p}{p!} + \beta_2^T \dfrac{(c-e)^p}{p!} = \widehat{C}_{p+1}, \\ (\beta_1+\beta_2)^T C_p = (v+w)^T C_p, \end{cases}$$

with the constants $\widetilde{C}_{p+1}$ and C_p given in Section 1.

It was observed by Tracogna [17] that the estimate (3.2) is more reliable if the condition $(\beta_1+\beta_2)^T e = 0$ is replaced by the two conditions $\beta_1^T e = 0$ and $\beta_2^T e = 0$. Solving the resulting system of equations corresponding to the TSRK method given in Section 2 leads to the following coefficients β_1 and β_2:

$$\beta_1 = [\,1.76797 \quad -2.32030 \quad 0.655654 \quad -0.103315\,]^T,$$

$$\beta_2 = [\,-0.158241 \quad 0.409025 \quad -0.692396 \quad 0.441612\,]^T.$$

We now have to discuss the role of (3.1). It follows from (1.5) that this condition is automatically satisfied if $\delta = 1$, i.e., if no step change is permitted, and it is approximately satisfied if δ is close to one. In the actual implementation of the code we permit the values of δ in the range $[\delta_{\min}, \delta_{\max}]$, where $\delta_{\min} = 0.1$ and $\delta_{\max} = 2$. The results of numerical experiments presented in Section 6 demonstrate that the estimate (3.2) is quite reliable for all values of δ in this range (compare Figs. 7.1a and 7.2a in Section 7).

4. Starting procedure. General TSRK methods of the form (1.2)–(1.3) require a starting procedure to compute $\widetilde{Y}^{[0]}$ and y_1 in addition to the given initial value y_0. This starting procedure must be compatible with the TSRK method, which means that the terms of order up to $p-1$ in the B-series corresponding to $\widetilde{Y}^{[0]}$ computed by the starting procedure coincide with the corresponding terms in the TSRK formula (compare [9], [18]; for the notion of B-series we refer the reader to [8]). However, as observed by Hairer and Wanner [9] and Tracogna and Welfert [18], the situation is much simpler for TSRK methods of stage order $q = p$ or $q = p-1$. In this case it is possible to choose as starting procedure any continuous RK method of order p or to use repeatedly a (discrete) RK method of order

p with step sizes $c_i(x_1 - x_0)$, $i = 1, \dots, s$, to compute $\widetilde{Y}_j^{[0]}$, where the c_i correspond to the TSRK formula. In our code we adopt the former approach and compute $\widetilde{Y}^{[0]}$ and y_1 by the continuous RK method constructed by Owren and Zennaro [15] by minimizing the continuous coefficients of the local discretization error. The resulting optimal method of order $p = 5$ with $s = 8$ stages and stage reuse has the form

$$(4.1) \qquad \begin{cases} K_i = f\Big(y_0 + h_1 \displaystyle\sum_{j=1}^{i-1} \widetilde{a}_{ij} K_j\Big), \\ \xi(x_0 + \theta h_1) = y_0 + h_1 \displaystyle\sum_{i=1}^{s} \widetilde{b}_i(\theta) K_i, \end{cases}$$

$i = 1, \dots, s$, $\theta \in [0,1]$, $h_1 = x_1 - x_0$, where the vector $\widetilde{c} = [\widetilde{c}_1, \dots, \widetilde{c}_s]^T$ of stage abscissas, the coefficient matrix $\widetilde{A} = [\widetilde{a}_{i,j}]_{i,j=1}^{s}$, and the vector $\widetilde{b}(\theta) = [\widetilde{b}_1(\theta), \dots, \widetilde{b}_s(\theta)]^T$ of continuous weights are given by

$$\begin{array}{c|c} \widetilde{c} & \widetilde{A} \\ \hline & \widetilde{b}^T(\theta) \end{array} = \begin{array}{c|cccccccc} 0 & & & & & & & & \\ \frac{1}{6} & \frac{1}{6} & & & & & & & \\ \frac{1}{4} & \frac{1}{16} & \frac{3}{16} & & & & & & \\ \frac{1}{2} & \frac{1}{4} & -\frac{3}{4} & 1 & & & & & \\ \frac{1}{2} & -\frac{3}{4} & \frac{15}{4} & -3 & \frac{1}{2} & & & & \\ \frac{9}{14} & \frac{369}{1372} & -\frac{243}{343} & \frac{297}{343} & \frac{1485}{9604} & \frac{297}{4802} & & & \\ \frac{7}{8} & -\frac{133}{4512} & \frac{1113}{6016} & \frac{7945}{16544} & -\frac{12845}{24064} & -\frac{315}{24064} & \frac{156065}{198528} & & \\ 1 & \frac{83}{945} & 0 & \frac{248}{825} & \frac{41}{180} & \frac{1}{36} & \frac{2401}{38610} & \frac{6016}{20475} & \\ \hline & \widetilde{b}_1(\theta) & \widetilde{b}_2(\theta) & \widetilde{b}_3(\theta) & \widetilde{b}_4(\theta) & \widetilde{b}_5(\theta) & \widetilde{b}_6(\theta) & \widetilde{b}_7(\theta) & \widetilde{b}_8(\theta) \end{array}$$

with

$$\begin{aligned} \widetilde{b}_1(\theta) &= \tfrac{596}{315}\theta^5 - \tfrac{4969}{819}\theta^4 + \tfrac{17893}{2457}\theta^3 - \tfrac{3292}{819}\theta^2 + \theta, \\ \widetilde{b}_2(\theta) &= 0, \\ \widetilde{b}_3(\theta) &= -\tfrac{1984}{275}\theta^5 + \tfrac{1344}{65}\theta^4 - \tfrac{43568}{2145}\theta^3 + \tfrac{5112}{715}\theta^2, \\ \widetilde{b}_4(\theta) &= \tfrac{118}{15}\theta^5 - \tfrac{1465}{78}\theta^4 + \tfrac{3161}{234}\theta^3 - \tfrac{123}{52}\theta^2, \\ \widetilde{b}_5(\theta) &= 2\theta^5 - \tfrac{413}{78}\theta^4 + \tfrac{1061}{234}\theta^3 - \tfrac{63}{52}\theta^2, \\ \widetilde{b}_6(\theta) &= -\tfrac{9604}{6435}\theta^5 + \tfrac{2401}{1521}\theta^4 + \tfrac{60025}{50193}\theta^3 - \tfrac{40817}{33462}\theta^2, \\ \widetilde{b}_7(\theta) &= -\tfrac{48128}{6825}\theta^5 + \tfrac{96256}{5915}\theta^4 - \tfrac{637696}{53235}\theta^3 + \tfrac{18048}{5915}\theta^2, \\ \widetilde{b}_8(\theta) &= 4\theta^5 - \tfrac{109}{13}\theta^4 + \tfrac{75}{13}\theta^3 - \tfrac{18}{13}\theta^2. \end{aligned}$$

The approximations y_1 and $\widetilde{Y}^{[0]}$ are now computed from the formulas

$$y_1 = \xi(x_1), \qquad \widetilde{Y}_j^{[0]} = f(\xi(x_0 + c_j h_1)), \qquad j = 1, 2, 3, 4.$$

Owren and Zennaro [15] constructed also an embedded discrete RK method of order four which can be used for error control. However, we did not find this estimate very reliable and decided instead to estimate the local discretization error of the first step by the Richardson extrapolation

$$\text{est}(x_1) = \frac{32(y_1 - y_1^*)}{31}. \tag{4.2}$$

Here, y_1^* is an approximation to $y(x_1)$ computed by a continuous RK method (4.1) over two steps of size $h_1/2$. For the problem of the form (1.1) this requires 14 additional function evaluations, so the cost of (4.2) is quite high. However, this estimate is used only in the first step so its contribution to the overall cost of the algorithm is not significant.

5. Selection of initial step size. We choose the initial step size $h_1 = x_1 - x_0$ following the approach given in [8], which is a modification of the approach proposed by Gladwell, Shampine and Brankin [7]. All the heuristic constants and safety factors below have also been adopted from [8].

Put $\text{sc}_i = \text{Atol}_i + |y_i(x_0)|\text{Rtol}_i$, $i = 1, \ldots, m$, where Atol_i and Rtol_i are absolute and relative error tolerances corresponding to the ith component of the solution and define the norm $\|\cdot\|_{\text{sc}}$ by

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

The algorithm proposed in [8] is the following. As first guess for the step size we let

$$h_0 = 0.01(d_0/d_1),$$

where $d_0 = \|y_0\|_{\text{sc}}$, $d_1 = \|f(y_0)\|_{\text{sc}}$. If either d_0 or d_1 is smaller than 10^{-5} we put $h_0 = 10^{-6}$. We next compute $\widetilde{h}_0$ from the formula

$$\widetilde{h}_0 = (0.01/\max\{d_1, d_2\})^{1/6},$$

where

$$d_2 = \frac{\|f(y_0 + h_0 f(y_0)) - f(y_0)\|_{\text{sc}}}{h_0}.$$

If $\max\{d_1, d_2\} \le 10^{-15}$ we put $\widetilde{h}_0 = \max\{10^{-6}, h_0 \cdot 10^{-3}\}$. Then the proposed starting step size h_1 is given by

$$h_1 = \min\{100 \cdot h_0, \widetilde{h}_0\}.$$

The initial step is accepted if $\text{err} \le 1$, where

$$\text{err} = \|\text{est}(x_1)\|_{\text{sc}},$$

with $\text{sc}_i = \text{Atol}_i + \max\{|y_{0i}|, |y_{1i}|\}\text{Rtol}_i$, $i = 1, \ldots, m$, and $\text{est}(x_1)$ computed by (4.2). The initial step is rejected if $\text{err} > 1$ and the computations are repeated with a new step size $\widetilde{h}_1$ adjusted according to the formula

$$\widetilde{h}_1 = h_1 \cdot \min\{\delta_{\max}, \max\{\delta_{\min}, \delta_{\text{sf}} \cdot (1/\text{err})^{1/6}\}\},$$

with safety factors $\delta_{\max} = 2$, $\delta_{\min} = 0.1$, $\delta_{\text{sf}} = 0.9$. After the first step computed by the continuous RK method (4.1) is accepted we define $h_2 = h_1$, where h_1 is the size of the accepted step from x_0 to x_1, and compute the quantity $F(\widetilde{Y}^{[0]})$, which is needed to continue integration with the TSRK method (1.2)–(1.3), and to compute the local error estimate (3.2).

6. Step size changing strategy. Assume we have completed a step by the TSRK method from x_n to x_{n+1} with a step size h_{n+1}, $n \geq 1$, which resulted in the computation of the quantities $Y^{[n]}$, $F(Y^{[n]})$, and y_{n+1}. We next compute the estimate $\text{est}(x_{n+1})$ of the local discretization error at x_{n+1} using formula (3.2) and form the quantity

$$\text{err} = \|\text{est}(x_{n+1})\|_{\text{sc}},$$

with $\text{sc}_i = \text{Atol}_i + \max\{|y_{ni}|, |y_{n+1,i}|\}\text{Rtol}_i$, $i = 1, \ldots, m$. This quantity is then compared to one to compute an optimal step size (compare [8])

$$h_{\text{opt}} = h_{n+1} \cdot (1/\text{err})^{1/6}.$$

Let, for a given computer, $m_\varepsilon > 0$ be the smallest number for which $1 + m_\varepsilon \neq 1$. If $m_\varepsilon < \text{err} \leq 1$ the step is accepted and a new step size h_{n+2} is computed from the formula

$$h_{n+2} = h_{n+1} \cdot \min\{\delta_{\max}, \max\{\delta_{\min}, \delta_{\text{sf}} \cdot h_{\text{opt}}\}\}$$

with $\delta_{\max}$, $\delta_{\min}$, and δ_{sf} defined in Section 4. If $\text{err} \leq m_\varepsilon$ we put $h_{n+2} = h_{n+1} \cdot \delta_{\min}$. The step is then completed by the computation of the quantities $F(\widetilde{Y}^{[n]})$ and $\widetilde{y}_n$ using formula (1.3).

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

$$\widetilde{h}_{n+1} = h_{n+1} \cdot \min\{\delta_{\max}, \max\{\delta_{\min}, \delta_{\text{sf}} \cdot h_{\text{opt}}\}\}$$

and the new quantities $F(\widetilde{Y}^{[n-1]})$ and $\widetilde{y}_{n-1}$ corresponding to $\widetilde{h}_{n+1}$. If $n = 1$ then $\widetilde{Y}_j^{[0]}$, $j = 1, \ldots, s$, and $\widetilde{y}_0$ are computed by the continuous RK method (4.1) at the points

$$x_1 + (c_j - 1)\widetilde{h}_2, \qquad x_1 - \widetilde{h}_2,$$

i.e., $\widetilde{Y}_j^{[0]} = \xi(x_1 + (c_j - 1)\widetilde{h}_2)$, $\widetilde{y}_0 = \xi(x_1 - \widetilde{h}_2)$. If $n > 1$ then $F(\widetilde{Y}^{[n-1]})$ and $\widetilde{y}_{n-1}$ are computed from

$$\begin{cases} F(\widetilde{Y}^{[n-1]}) = (\widetilde{V} \otimes I_m)F(\widetilde{Y}^{[n-2]}) + (\widetilde{W} \otimes I_m)F(Y^{n-1]}), \\ \widetilde{y}_{n-1} = y_{n-1} + (\widetilde{v} \otimes I_m)\widetilde{h}_{n+1}F(\widetilde{Y}^{[n-2]}) + (\widetilde{w} \otimes I_m)\widetilde{h}_{n+1}F(Y^{[n-1]}), \end{cases}$$

with $\widetilde{V}$, $\widetilde{W}$, $\widetilde{v}$ and $\widetilde{w}$ corresponding to the new ratio $\delta = \widetilde{h}_{n+1}/h_n$.

7. Numerical examples. An experimental code `tsrk5` based on TSRK methods (1.2)–(1.3) with coefficients defined in Section 2 was written in Matlab and applied to many problems to test its accuracy, efficiency, the reliability of local error estimation, and robustness of step changing strategy. The implementation issues of this code were described in Sections 3–6. The problems for the above tests were taken from the paper [10] one of whose objectives in presenting the test problems, methods and comparison criteria was to provide a rigorous conceptual basis for comparing numerical methods for ordinary differential equations. For illustration we present below a selection of results of numerical experiments on the following two problems: Van der Pol equation and orbit equation, i.e. problem E2 and D5 in [10].

Example 1. Van der Pol equation ([10]):

$$\begin{cases} y_1' = y_2, & y_1(0) = 2, \\ y_2' = (1 - y_1^2)y_2 - y_1, & y_2(0) = 0, \end{cases}$$

$x \in [0, 20]$.

Example 2. Orbit equation ([10]):

$$\begin{cases} y_1' = y_3, & y_1(0) \ = 1 - \varepsilon, \\ y_2' = y_4, & y_2(0) \ = 0, \\ y_3' = -y_1/(y_1^2 + y_2^2)^{3/2}, & y_3(0) \ = 0, \\ y_4' = -y_2/(y_1^2 + y_2^2)^{3/2}, & y_4(0) \ = \sqrt{(1+\varepsilon)/(1-\varepsilon)}, \end{cases}$$

$x \in [0, 20]$, $\varepsilon = 0.9$. Here, ε is the eccentricity of the orbit.

In Tables 7.1 and 7.2 we have listed the number of steps ns, number of rejected steps nr, number of function evaluations nfe for our experimental code `tsrk5` based on the TSRK method (1.2)–(1.3) as well as for the Matlab `ode45` code for $\text{Atol}_i = \text{Rtol}_i = \text{tol} = 10^{-4}$, 10^{-8}, and 10^{-12}. This code was written by Shampine and Reichelt (see [16]) and it is based on the explicit Runge–Kutta $(4, 5)$ pair DOPRI5 constructed by Dormand and Prince [6]. This code uses local extrapolation so it is effectively of order five. The results presented in Tables 7.1 and 7.2 demonstrate that the `tsrk5` code is competitive with the Matlab `ode45` code for all tolerances.

Table 7.1. Numerical results for Example 1

tol	10^{-4}			10^{-8}			10^{-12}		
method	ns	nr	nfe	ns	nr	nfe	ns	nr	nfe
`tsrk5`	112	14	530	513	28	2190	2368	33	9630
`ode45`	78	27	631	425	9	2605	2653	4	15943

Table 7.2. Numerical results for Example 2

tol	10^{-4}			10^{-8}			10^{-12}		
method	ns	nr	nfe	ns	nr	nfe	ns	nr	nfe
`tsrk5`	144	39	2378	579	3	2378	2674	2	10754
`ode45`	98	18	697	516	0	3097	3245	0	19471

We have plotted in Figs. 7.1a and 7.2a the local errors and local error estimates for $\mathrm{tol}_i = \mathrm{Rtol}_i = \mathrm{tol} = 10^{-4}$ (solid line, symbol '$*$'), 10^{-8} (dashed line, symbol '$\circ$') and 10^{-12} (dashdotted line, symbol '$\times$'). The corresponding step size patterns are plotted in Figs. 7.1b and 7.2b, where we have used solid, dashed, and dashdotted lines in Figs. 7.1a and 7.2a and solid line and symbols '$*$', '$\circ$', and '$\times$' in Figs. 7.1b and 7.2b for tol $= 10^{-4}$, 10^{-8}, and 10^{-12}, respectively. In Figs. 7.1b and 7.2b the rejected steps are indicated by '$\times$'. We can see that the error estimation employed in our code is very reliable and that the step size changing mechanism is very robust. In all cases the error at the endpoint was about $10 \cdot \mathrm{tol}$.

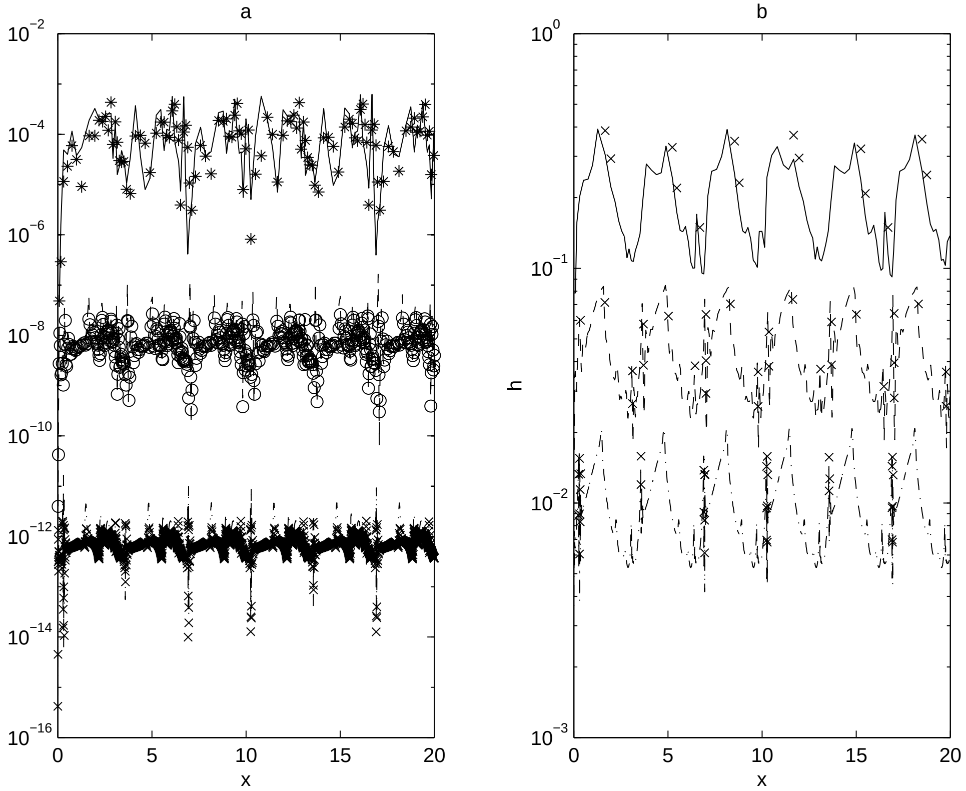

Fig. 7.1. Example 1: a. Local error versus local error estimate. b. Step size control

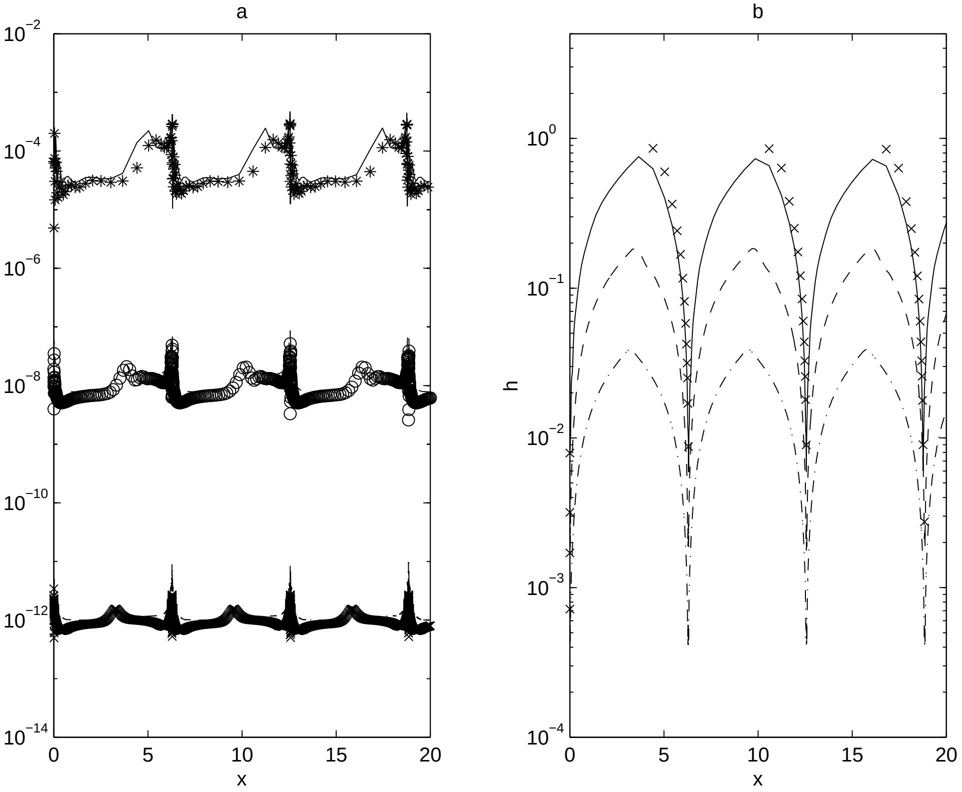

Fig. 7.2. Example 2: a. Local error versus local error estimate. b. Step size control

Acknowledgements. The authors wish to express their gratitude to the anonymous referee for his helpful comments.

References

[1] Z. Bartoszewski and Z. Jackiewicz, *Construction of two-step Runge–Kutta methods of high order for ordinary differential equations*, Numer. Algorithms 18 (1998), 51–70.

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

[3] —, *The Numerical Analysis of Ordinary Differential Equations. Runge–Kutta and General Linear Methods*, Wiley, Chichester, 1987.

[4] 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.

[5] J. C. Butcher and S. Tracogna, *Order conditions for two-step Runge–Kutta methods*, Appl. Numer. Math. 24 (1997), 351–364.

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

[7] 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.

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

[9] E. Hairer and G. Wanner, *Order conditions for general two-step Runge–Kutta methods*, SIAM J. Numer. Anal. 34 (1997), 2087–2089.

[10] T. E. Hull, W. H. Enright, B. M. Fellen and A. E. Sedgwick, *Comparing numerical methods for ordinary differential equations*, SIAM J. Numer. Anal. 9 (1972), 603–637.

[11] Z. Jackiewicz and S. Tracogna, *A representation formula for two-step Runge–Kutta methods*, in: *Hellenic European Research on Mathematics and Informatics'94* (Athens, 1994), E. A. Lipitakis (ed.), Hellenic Math. Soc., Athens, 1994, 111–120.

[12] —, —, *A general class of two-step Runge–Kutta methods for ordinary differential equations*, SIAM J. Numer. Anal. 32 (1995), 1390–1427.

[13] —, —, *Variable stepsize continuous two-step Runge–Kutta methods for ordinary differential equations*, Numer. Algorithms 12 (1996), 347–368.

[14] Z. Jackiewicz and R. Vermiglio, *General linear methods with external stages of different orders*, BIT 36 (1996), 688–712.

[15] B. Owren and M. Zennaro, *Derivation of efficient*, *continuous*, *explicit Runge–Kutta methods*, SIAM J. Sci. Statist. Comput. 13 (1992), 1488–1501.

[16] L. F. Shampine and M. W. Reichelt, *The Matlab ODE suite*, SIAM J. Sci. Comput. 18 (1997), 1–22.

[17] S. Tracogna, *Implementation of two-step Runge–Kutta methods for ordinary differential equations*, J. Comput. Appl. Math. 76 (1997), 113–136.

[18] S. Tracogna and B. Welfert, *Two-step Runge–Kutta*: *Theory and practice*, BIT 40 (2000), 775–799.

Faculty of Technical Physics and
Applied Mathematics
Technical University of Gdańsk
G. Narutowicza 11/12
80-952 Gdańsk, Poland
E-mail: zbart@mif.pg.gda.pl

Department of Mathematics
Arizona State University
Tempe, AZ 85287, U.S.A.
E-mail: jackiewi@math.la.asu.edu

Received on 29.9.2000;
revised version on 14.5.2001 (1553)