Journal of Mathematical Chemistry, Vol. 40, No. 1, July 2006 (© 2006)
DOI: 10.1007/s10910-006-9121-x

A new eighth-order A-stable method for solving differential systems arising in chemical reactions

Jesús Vigo-Aguiar* and Higinio Ramos

Scientific Computing Group. Universidad de Salamanca. Spain
E-mail: jvigo@usal.es

Implicit Runge–Kutta methods are successful algorithms for the numerical solution of stiff differential equations, as they usually appear in chemical reactions. This article describes the construction of a particular implicit method based on internal stages obtained from certain Chebyshev collocation points. The resulting method has algebraic order 8 and A-stability characteristic. An embedding technique using the Runge–Kutta method and a linear multistep one is provided in order to change the step size. Numerical experiments illustrate the behaviour of the new method, showing that it may reach great accuracy and be competitive with other well-known codes.

KEY WORDS: stiff problems, implicit Runge–Kutta method, A-stability, Chebyshev points

1. Introduction

The aim of this contribution is to present a new method with A-stability characteristic suitable for numerically solving stiff problems. Stiff problems are characterized by the fact that the numerical solutions of slow movements are considerably perturbed by nearby rapidly changing solutions, as it usually occurs in problems arising from chemical reactions. In fact, the first appearance of the term "stiff" was in the paper by Curtiss and Hirschfelder [1] on problems in chemical kinetics.

The following section is devoted to the construction of an stiffly accurate implicit Runge–Kutta method, which is proved to be A-stable and have eighth-order. A key to high efficiency in numerical methods is the capacity to vary the steplength according to the characteristics of the solution of the problem. For this purpose, a pair of embedding Runge–Kutta methods are commonly used [2]. But this technique results in a highly extra cost for the number of function evaluations. In section 4, a different strategy for the estimate of the local

*Corresponding author.

error is outlined. This new strategy is based in the use of a pair of methods: the implicit Runge–Kutta method, and a linear multistep one. In section 5, several numerical experiments are presented which demonstrate the effectiveness of the method proposed in this paper, which may be competitive with other well-known codes. Finally, in the last section some concluding remarks are given.

2. Construction of the new method

In this paper, we are interested in the numerical integration of initial-value problems of ordinary differential equations of the form

$$\begin{aligned} y'(t) &= f(t, y(t)), \quad t \in [t_0, T], \\ y(t_0) &= y_0, \end{aligned} \tag{1}$$

where the function $f: \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$ it is assumed to satisfy the necessary conditions in order to ensure that the problem has a unique solution.

We consider an implicit Runge–Kutta method of seven stages given by the Butcher tableau

$$\begin{array}{c|c} c & A \\ \hline & b \end{array}, \tag{2}$$

where $A = \left(a_{ij}\right)_{i,j=1}^{7}$, $c = (c_1, \ldots, c_7)$, $b = (b_1, \ldots, b_7)$, and the values c_i are chosen as

$$c_i = \xi_{i-1} = \frac{1}{2}\left(1 + \alpha_{i-1}\right), \quad i = 1, \ldots, 7$$

with the α_{i-1} the Chebyshev collocation points [3, p. 32] given by

$$\alpha_{i-1} = \cos(\theta_{i-1}), \quad \theta_{i-1} = \frac{(7-i)\pi}{6}, \quad i = 1, \ldots, 7. \tag{3}$$

Explicitly, we have

$$\begin{aligned} &c_1 = 0, \quad && c_2 = \tfrac{1}{4}\left(2 - \sqrt{3}\right), \quad && c_3 = \tfrac{1}{4}, \quad && c_4 = \tfrac{1}{2}, \\ &c_5 = \tfrac{3}{4}, \quad && c_6 = \tfrac{1}{4}\left(2 + \sqrt{3}\right), \quad && c_7 = 1. \end{aligned}$$

A desirable condition, often called *stiff accuracy*, for Runge–Kutta methods applied to very stiff problems is that the last stage is also the numerical solution for the step (see [4]). In our context, this means that we set

$$b_i = a_{7\,i}\,, \quad i = 1, \ldots, 7\,.$$

Now, in order to determine values for the $a_{i\,j}$ we write the Runge–Kutta method in (2) in the alternative form (see [5]):

$$y_{n+c_i} = y_n + h \sum_{j=1}^{7} a_{i\,j}\, f(t_n + c_j\, h, y_{n+c_j})\,, \quad i = 1, \ldots, 7\,,$$

$$y_{n+1} = y_n + h \sum_{i=1}^{7} b_i\, f(t_n + c_i\, h, y_{n+c_i})\,, \tag{4}$$

where, as it is usual, y_{n+c_i} stands for the approximation of the true values $y(t_n + c_i\, h)$.

We consider the linear operators associated with the equations in (4) given by

$$\mathcal{L}_i(z(t), h) = z(t + c_i\, h) - z(t) - h \sum_{j=1}^{7} a_{i\,j}\, z'(t + c_j\, h) \tag{5}$$

for $i = 1, \ldots, 7$, where $z(t)$ is a sufficiently differentiable function. Expanding $z(t + c_i\, h)$ and $z'(t + c_j\, h)$ about t and imposing to the intermediate steps to have seventh order we get for $i = 1, \ldots, 7$ seven uncoupled algebraic systems of equations given by

$$\begin{aligned}
a_{i1} + a_{i2} + \cdots + a_{i\,7} &= c_i\,,\\
a_{i1}\, c_1 + a_{i2}\, c_2 + \cdots + a_{i\,7}\, c_7 &= \frac{c_i^2}{2}\,,\\
&\vdots\\
a_{i1}\, c_1^6 + a_{i2}\, c_2^6 + \cdots + a_{i\,7}\, c_7^6 &= \frac{c_i^7}{7}\,.
\end{aligned}$$

Solving these linear systems of Vandermonde type, the required values for the Runge–Kutta method are obtained and finally the matrix A in the Butcher tableau in (2) is given by

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{53+9\sqrt{3}}{2520} & \frac{256-47\sqrt{3}}{4032} & \frac{2269-1368\sqrt{3}}{20160} & \frac{164-93\sqrt{3}}{1260} & \frac{2339-1368\sqrt{3}}{20160} & \frac{256-145\sqrt{3}}{4032} & \frac{-17+9\sqrt{3}}{2520} \\ \frac{1}{280} & \frac{37}{504}+\frac{3\sqrt{3}}{64} & \frac{233}{2240} & \frac{-23}{1260} & \frac{23}{2240} & \frac{37}{504}-\frac{3\sqrt{3}}{64} & \frac{1}{280} \\ \frac{53}{2520} & \frac{16+7\sqrt{3}}{252} & \frac{319}{1260} & \frac{41}{315} & \frac{-31}{1260} & \frac{16-7\sqrt{3}}{252} & \frac{-17}{2520} \\ \frac{3}{280} & \frac{3\left(8+7\sqrt{3}\right)}{448} & \frac{489}{2240} & \frac{39}{140} & \frac{279}{2240} & \frac{3}{56}-\frac{3\sqrt{3}}{64} & \frac{3}{280} \\ \frac{53-9\sqrt{3}}{2520} & \frac{256+145\sqrt{3}}{4032} & \frac{2269+1368\sqrt{3}}{20160} & \frac{164+93\sqrt{3}}{1260} & \frac{2339+1368\sqrt{3}}{20160} & \frac{256+47\sqrt{3}}{4032} & \frac{-17-9\sqrt{3}}{2520} \\ \frac{1}{70} & \frac{8}{63} & \frac{8}{35} & \frac{82}{315} & \frac{8}{35} & \frac{8}{63} & \frac{1}{70} \end{bmatrix},$$

and for the other two vectors we have

$$c = \left[0, \frac{2-\sqrt{3}}{4}, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, \frac{2+\sqrt{3}}{4}, 1\right]^T,$$

$$b = \left[\frac{1}{70}, \frac{8}{63}, \frac{8}{35}, \frac{82}{315}, \frac{8}{35}, \frac{8}{63}, \frac{1}{70}\right].$$

3. Stability and order of the method

When an s-stage implicit Runge–Kutta method is applied to the test equation $y' = \lambda y$ yields

$$y_{n+1} = R(\lambda h)\, y_n$$

with the *stability function* $R(z)\colon \mathbb{C} \to \mathbb{C}$ given by

$$R(z) = 1 + z\, b^T (I - z\, A)^{-1}\, e,$$

where e is the vector of s components given by $e = (1, \ldots, 1)^T$. In this context, the method is called *A-stable* (see [4]) if the left-half complex plane is included in the stability domain, that is,

$$\mathbb{C}^- = \{z;\ Re\, z < 0\} \subset S = \{z;\ |R(z)| \leqslant 1\}. \tag{6}$$

The stability function for the method in the above section is the rational function

$$R(z) = \frac{N(z)}{D(z)},$$

where

$$N(z) = 2580480 + 1290240\,z + 291840\,z^2 + 38400\,z^3 + 3108\,z^4 + 146\,z^5 + 3\,z^6\,,$$

$$D(z) = 2580480 - 1290240\,z + 291840\,z^2 - 38400\,z^3 + 3108\,z^4 - 146\,z^5 + 3\,z^6\,,$$

which has modulus less than one on the left-half complex plane, and thus, according to (6), the method is A-stable. Moreover, with the help of the Mathematica system it may be checked that the order conditions of the Butcher theory for order 8 are verified, but not for order 9, and so the method has algebraic order 8. In fact, for the equation that approximates the final value (which coincides with the equation for the last stage) we have

$$y(t_n + h) - \left(y(t_n) + h \sum_{i=1}^{7} b_i\, y'(t_n + c_i\, h) \right) = \frac{h^9\, y^{(9)}(t_n)}{26011238400} + \mathcal{O}(h^{10})\,.$$

4. Notes on the implementation and step size selection

After applying the method described in the above sections to a given problem of the form in (1) we obtain on each step an algebraic system of $6 \times m$ equations with $6 \times m$ unknowns (the values of the solution components at the intermediate stages). For nonlinear differential equations this system has to be solve iteratively and for this purpose different versions of the Newton method may be used (see [6,7]).

As it is usually done for the Runge–Kutta methods we could consider the use of an embedded pair of methods for the step size selection. Taking

$$\hat{b} = \left[0, \frac{7}{45}, \frac{1}{5}, \frac{13}{45}, \frac{1}{5}, \frac{7}{45}, 0 \right],$$

it is easy to check with the help of the Mathematica system that the implicit Runge–Kutta method

$$\begin{array}{c|c} c & A \\ \hline & \hat{b} \end{array}$$

verify the order conditions to have order 6, where A and c are the same as before.

Thus, we would have an approximation of $y(t_n + h)$ by a Runge–Kutta method of lower order, given by

$$\hat{y}_{n+1} = y_n + h \sum_{i=1}^{7} \hat{b}_i \, f(t_n + c_i \, h, \, y_{n+c_i}) \, . \tag{7}$$

The difference between both approximations could be taken as an estimation for the local truncation error, given by

$$err = y_{n+1} - \hat{y}_{n+1} = h \sum_{i=1}^{7} \left(b_i - \hat{b}_i \right) \, f(t_n + c_i \, h, \, y_{n+c_i}) \, , \tag{8}$$

which behaves like $\mathcal{O}(h^7)$ for $h \to 0$.

But the use of the embedded pair of methods described above would be very closely in terms of the number of functions evaluations. Instead of that, we propose a different approach to obtain an estimate for the local truncation error, using the linear multistep method of second-order (which is the two-step backward differentiation formula of step size $h/2$, [11]) given by

$$h \, f_{n+1} = 3 \, y_{n+1} + y_n - 4 \, y_{n+1/2} \, . \tag{9}$$

From this formula, if we assume that $z(t)$ is the true solution of the problem in (1), we have

$$h \, f(t_{n+1}, z(t_{n+1})) = 3 \, z(t_{n+1}) + z(t_n) - 4 \, z(t_{n+1/2}) + \mathcal{O}(h^3) \, , \tag{10}$$

where t_{n+1}, $t_{n+1/2}$ are abbreviations, respectively, for $t_n + h$ and $t_n + \frac{1}{2} \, h$.

Now, if $y_{n+1/2}$ is the approximate value at the intermediate stage obtained with the Runge–Kutta method described in section 2, assuming the localization hypothesis [5]

$$y_n = z(t_n) \, , \quad y_{n+1/2} = z(t_{n+1/2})$$

after expanding in Taylor series about (t_{n+1}, y_{n+1}) the function on the left-hand side in (10), we have the approximation

$$\begin{aligned} h \left[f(t_{n+1}, y_{n+1}) + \frac{\delta f}{\delta y}(t_{n+1}, y_{n+1}) \, \left(z(t_{n+1}) - y_{n+1} \right) \right] \\ \simeq 3 \, z(t_{n+1}) + y_n - 4 \, y_{n+1/2} \, , \end{aligned} \tag{11}$$

where $\delta f / \delta y$ refers to the Jacobian matrix.

On subtracting $3 \, y_{n+1}$ in the two sides of (11) and rearranging, we get

$$\begin{aligned} \left(3 \, I_m - h \, \frac{\delta f}{\delta y}(t_{n+1}, y_{n+1}) \right) (z(t_{n+1}) - y_{n+1}) \\ \simeq h \, f(t_{n+1}, y_{n+1}) - y_n + 4 \, y_{n+1/2} - 3 \, y_{n+1} \, , \end{aligned} \tag{12}$$

where I_m stands for the identity matrix of order m.

Finally, the estimation for the local truncation error may be given by

$$\begin{aligned} \text{err} &= \left\| z(t_{n+1}) - y_{n+1} \right\| \\ &\simeq \left\| M^{-1} \left(h\, f(t_{n+1}, y_{n+1}) - y_n + 4\, y_{n+1/2} - 3\, y_{n+1} \right) \right\| , \end{aligned} \tag{13}$$

where M is the matrix

$$M = 3\, I_m - h\, \frac{\delta f}{\delta y}(t_{n+1}, y_{n+1}) .$$

Note that with this strategy, the extra cost for changing the step size consists just in one more function evaluation per step. The numerical experiments confirm that using this procedure we can obtain great accuracy.

Once we have derived an estimation for the local error, the standard step size prediction (see [8,9]) leads to

$$h_{\text{new}} = \tau\, h_{\text{old}} \left(\frac{\text{atol}}{\|\text{err}\|} \right)^{1/7}$$

for a given tolerance, atol, where τ is a safety factor. In order to avoid excessive computations, we also consider the common strategy that if the new step size satisfies

$$k_1\, h_{\text{old}} \leqslant h_{\text{new}} \leqslant k_2\, h_{\text{old}}$$

with, say $k_1 = 1.0$ and $k_2 = 1.25$, then we retain h_{old} for the following step.

5. Numerical results

To check the numerical behaviour when used to solve initial-value problems, we have applied the above method to a variety of well-known problems, which have appeared different times in the literature. These problems model different chemical reactions where usually certain variables change rapidly whereas others vary very slowly, indicating the presence of stiffness. The strategy for changing the step size was based on the estimation of the local truncation error using the "embedding technique" based on the Runge–Kutta method and the linear multistep method in the form describe in the above section. The code thus obtained will be named $RKC6$.

5.1. The Robertson problem

This classical problem that models the kinetics of a chemical reaction (see [4]) consists of a system of three equations given by

$$\begin{aligned} y_1'(t) &= -0.04\, y_1(t) + 10^4\, y_2(t) y_3(t)\,, \\ y_2'(t) &= 0.04\, y_1(t) - 10^4\, y_2(t) y_3(t) - 3 \times 10^7\, y_2(t)^2\,, \\ y_3'(t) &= 3 \times 10^7\, y_2(t)^2 \end{aligned}$$

with initial conditions $y_1(0) = 1$, $y_2(0) = y_3(0) = 0$. This special system, as it is typical for problems arising in chemical kinetics, has a small very quick initial transients. It has been integrated on the interval $[0, 10^{11}]$. The reference solution at the end of the integration interval has been taken from the test set in [10],

$$\begin{aligned} y_1(t_f) &= 0.2083340149701255 \times 10^{-7}\,, \\ y_2(t_f) &= 0.8333360770334713 \times 10^{-13}\,, \\ y_3(t_f) &= 0.9999999791665050\,. \end{aligned}$$

The results obtained with the new method are presented in table 1 with some of the data for different codes that appear in [10]. The parameters listed in the table are the prescribed tolerance, atol, the total number of steps, *n*step, the number of function evaluations, *f*eval, and a measure of the error given by the scd factor,

$$\mathrm{scd} = -\log_{10} \max_i \left\{ \frac{|y_i\,\mathrm{exact}(t_f) - y_i\,\mathrm{computed}(t_f)|}{|y_i\,\mathrm{exact}(t_f)|} \right\},$$

where t_f is the final point at the integration interval. In figure 1, we show the efficiency curves for these codes, where we have plotted the polygonals joining the points $(\log_{10}(f\mathrm{eval}), \mathrm{scd})$, that is, the relative error (scd factor) versus

Table 1
Results for problem 5.1.

METHOD	atol	h_0	*n*step	*f*eval	scd
RKC6	10^{-8}	10^{-6}	46	312	7.16
	10^{-9}	10^{-6}	81	567	10.28
	10^{-10}	10^{-6}	152	1064	12.04
DASSL	10^{-11}		1278	1549	3.47
MEBDF	10^{-14}	10^{-12}	1624	5252	7.85
VODE	10^{-14}		3306	3873	5.91
RADAU	10^{-14}	10^{-12}	108	3420	7.53

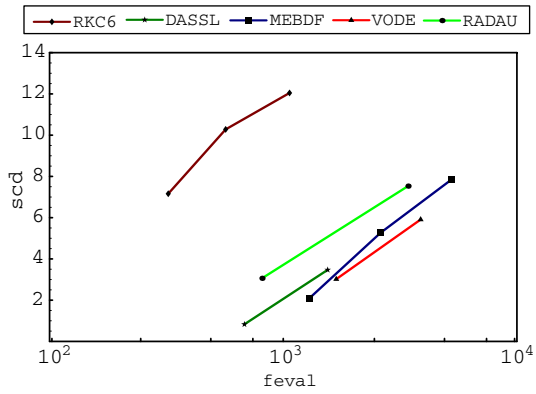

Figure 1. Efficiency comparison curves for problem 5.1.

the computational cost measured by the logarithm of the number of function evaluations.

5.2. *The HIRES problem*

This I.V.P. originated from plant physiology describes the "High Irradiance Response" of photomorphogenesis on the basis of phytochrome, and consists of a system of eight nonlinear ordinary differential equations given by

$$
\begin{aligned}
y_1'(t) &= -1.71\, y_1(t) + 0.43\, y_2(t) + 8.32\, y_3(t) + 0.0007\,,\\
y_2'(t) &= 1.71\, y_1(t) - 8.75\, y_2(t)\,,\\
y_3'(t) &= -10.03\, y_3(t) + 0.43\, y_4(t) + 0.035\, y_5(t)\,,\\
y_4'(t) &= 8.32\, y_2(t) + 1.71\, y_3(t) - 1.12\, y_4(t)\,,\\
y_5'(t) &= -1.745\, y_5(t) + 0.43\, y_6(t) + 0.43\, y_7(t)\,,\\
y_6'(t) &= -280\, y_6(t)\, y_8(t) + 0.69\, y_4(t) + 1.71\, y_5(t) - 0.43\, y_6(t) + 0.69\, y_7(t)\,,\\
y_7'(t) &= 280\, y_6(t)\, y_8(t) - 1.81\, y_7(t)\,,\\
y_8'(t) &= -280\, y_6(t)\, y_8(t) + 1.81\, y_7(t)
\end{aligned}
$$

with initial vector $\mathbf{y}_0$ given by $\mathbf{y}(0) = (1, 0, 0, 0, 0, 0, 0, 0.0057)^T$. The problem has been numerically solved on the interval $[0, 321.8122]$ as it was done in the test set [10] and the reference values were taken from there. The results are presented in table 2 where it appears the same parameters and comparisons with the same integrators as in the above problem.

In figure 2 we show the efficiency curves for these codes, where it appears the relative error (scd factor) versus the number of function evaluations in logarithmic scale.

Table 2
Results for problem 5.2.

METHOD	atol	h_0	$n$step	$f$eval	scd
RKC6	10^{-9}	10^{-3}	189	1323	10.30
	10^{-10}	10^{-3}	400	2800	12.53
DASSL	10^{-7}		380	591	3.81
	10^{-10}		1160	1557	6.78
MEBDF	10^{-7}	10^{-9}	218	767	4.24
	10^{-10}	10^{-12}	420	1492	7.30
VODE	10^{-7}		415	608	3.98
	10^{-10}		933	1224	6.20
RADAU	10^{-7}	10^{-9}	51	985	4.91
	10^{-10}	10^{-12}	69	1511	8.03

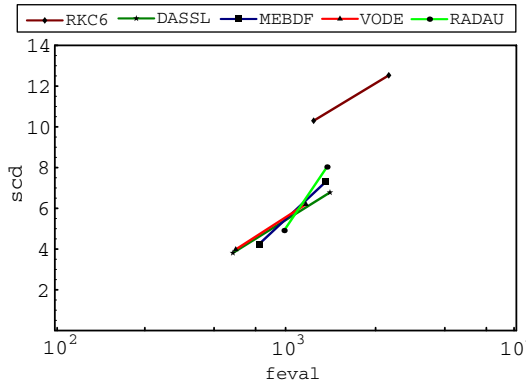

Figure 2. Efficiency comparison curves for problem 5.2.

5.3. *The OREGONATOR system*

This problem originates from the celebrated Belousov–Zhabotinskii reaction and consists of a stiff system of three nonlinear differential equations:

$$
\begin{aligned}
y_1'(t) &= s\ (\, y_2(t) + y_1(t)(1 - q\, y_1(t) - y_2(t))\,)\ , \\
y_2'(t) &= (\, y_3(t) - (1 + y_1(t))y_2(t))\,/s\ , \\
y_3'(t) &= w\,(y_1(t) - y_3(t))
\end{aligned}
$$

with $s = 77.27$, $w = 0.161$, $q = 8.375\times10^{-6}$, and initial values $\mathbf{y}(0) = (1, 2, 3)^T$. The integration interval has been taken [0, 360] as in [10], and the reference values at the final point t_f have been considered from there. In table 3 the data for the new method $RKC6$ and the other codes are presented, with the same considerations as in Problem 5.1.

Table 3
Results for problem 5.3.

METHOD	atol	h_0	nstep	feval	scd
RKC6	10^{-6}	10^{-3}	929	6503	8.49
	10^{-8}	10^{-3}	4466	31262	13.09
DASSL	10^{-7}		2725	4210	5.57
	10^{-10}		8192	11119	8.66
MEBDF	10^{-7}	10^{-9}	1586	5399	6.39
	10^{-10}	10^{-12}	3248	10754	9.65
VODE	10^{-7}		3083	4348	4.73
	10^{-10}		7890	9903	7.51
RADAU	10^{-7}	10^{-9}	267	6859	7.48
	10^{-10}	10^{-12}	261	12917	9.82

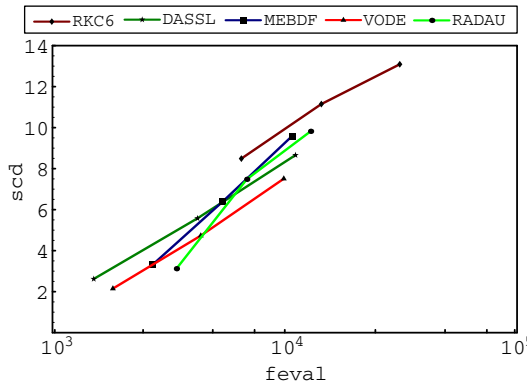

Figure 3. Efficiency comparison curves for problem 5.3.

In figure 3 we have plotted the efficiency curves for these codes showing the scd factor versus the number of function evaluations in logarithmic scale.

5.4. *The BRUSSELATOR system*

Consider the diffusion-free "Brusselator" system [11] given by the equations

$$y_1'(t) = B + y_1^2(t)\, y_2(t) - (A+1)\, y_1(t)\,, \qquad y_1(0) = y_1^0\,,$$
$$y_2'(t) = A\, y_1(t) - y_1^2(t)\, y_2(t)\,, \qquad y_2(0) = y_2^0\,,$$

where A and B are positive real constants [12]. It can be shown that the only critical point of the system is $(y_1^*, y_2^*) = (B, A/B)$. For our numerical experiment we take $A = 3$, $B = 1$, and initial values $y_1(0) = 1.5$, $y_2(0) = 3$, and consider the integration interval $[0, 20]$ as it was done in a recent article by Butcher and

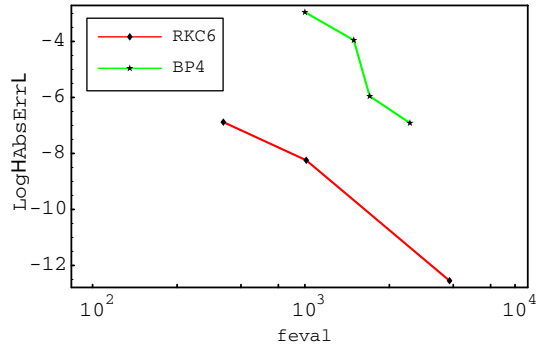

Figure 4. Efficiency comparison curves for problem 5.4.

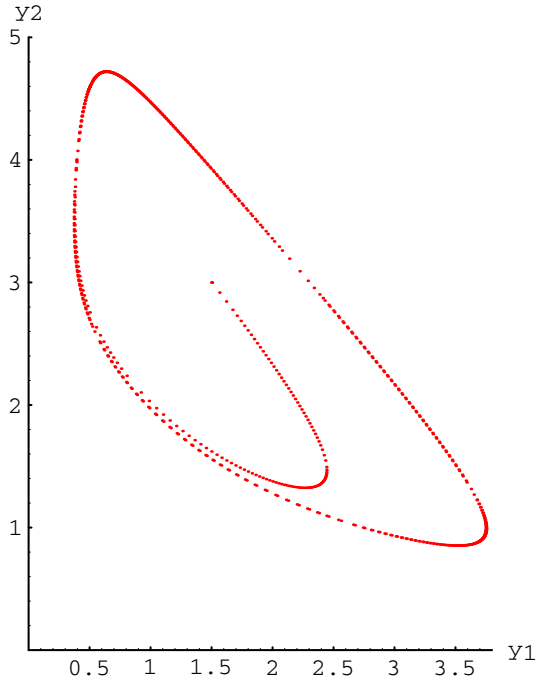

Figure 5. Phase portrait for problem 5.4 with $A = 3$ and $B = 1$.

Podhaisky [13]. In figure 4 we present in a double logarithmic scale the results obtained with the method $RKC6$ and the best of the results on [13] with a variable-order method that will be named $BP4$. Now, we consider the number of function evaluations versus the absolute error measured in the L_2 norm.

In figure 5, the phase portrait of the system using the $RKC6$ method is depicted.

6. Conclusions

In this paper, the construction of an implicit Runge–Kutta method appropriate for solving stiff initial-value problems is described. In the context of stability properties, the resulting method is A-stable and, interpreted as a general linear method, it is inherently Runge–Kutta stable [14]. We have derived the coefficients of the method which has algebraic order 8. A new strategy using an appropriate backward differentiation formula, instead of an embedded pair of Runge–Kutta methods, deserves a special attention in the task of changing the step size. This procedure may be applied in the general context of implicit Runge–Kutta methods, avoiding the extra cost of the embedding technique. It was shown by numerical examples on chemical kinetics that these methods are promising and indicate that may be competitive with other codes commonly used for stiff differential equations.

Acknowledgments

The authors wish to thank JCYL project SA 024/04 and MICYT project BMF 2004-295 for financial support.

References

[1] C.F. Curtiss and J.O. Hirschfelder 'Integration of stiff equation', Proc. Nat. Acd. Sci. 38 (1952) 235–243.
[2] J.C. Butcher, *Numerical Methods for Ordinary Differential Equations* (Wiley, Chichester, England, 2003).
[3] L. Fox and I.B. Parker, *Chebyshev Polynomials in Numerical Analysis* (Oxford U. Press, London, 1968) p. 32.
[4] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems* (Springer, Berlin, 1996).
[5] J.D. Lambert, *Numerical Methods for Ordinary Differential Systems* (John Wiley, England, 1991).
[6] J.C. Butcher, On the implementation of implicit Runge-Kutta methods, BIT 6 (1976) 237–240.
[7] E. Hairer and G. Wanner, Stiff differential equations solved by Radau methods, J. Comp. Appl. Math. 111 (1999) 93–111.
[8] L.F. Shampine and M.K. Gordon, *Computer solution of Ordinary Differential Equations. The initial Value Problem* (Freeman, San Francisco, CA, 1975).
[9] P. Henrici, Numerical Methods for Ordinary Differential Equations.
[10] F. Mazzia and F. Iavernaro, Test set for initial value problems solvers, Department of Mathematics, University of Bari, 2003.
[11] E. Hairer, S.P. Norsett and G. Wanner, *Solving Ordinary Differential Equations I* (Springer, Berlin, 1993).
[12] E.H. Twizell, A.B. Gumel and Q. Cao, A second-order scheme for the Brusselator reaction-diffusion system, J. Math. Chem. 26 (1999) 333–344.
[13] J.C. Butcher and H. Podhaisky, On error estimation in general linear methods for stiff ODEs, Appl. Numer. Math. to appear. Avalilable on internet (*www.elsevier.com*)
[14] J.C. Butcher, General Linear Methods, Comput. Math. Appl. 31 (4/5) (1996) 105–112.