

Quadratic SDIRK pair for treating chemical reaction problems.

Ch. Tsitouras*

TEI of Chalkis, Dept. of Applied Sciences, GR 34400 Psahna, Greece.

I. Th. Famelis†

TEI of Athens, Dept. of Mathematics, GR 12210 Egaleo, Greece.

(Received January 25, 2007)

Abstract

Most of the chemical reactions are modelled as initial value problems (IVP) where the derivative can be expressed as a quadratic function of the dependent variable. Some of the variables in these IVPs change rapidly as time progresses, whereas others vary very slowly, indicating the presence of stiffness. Singly Diagonal Implicit Runge–Kutta (SDIRK) pairs possess an interesting alternative for dealing with such problem. Software SDIRK4 applies a five stage pair SDIRK method of accuracy orders 4(3). In this paper by exploiting the presence of quadratic property, we derive a 5(3) order pair SDIRK methods at the same cost. Numerical results over well known chemical kinetics problems justify our effort.

1 Introduction.

Chemical reaction problems are modelled through systems of ordinary differential problems of the form

$$y' = f(x, y), \quad y(x_0) = y_0, \quad (1)$$

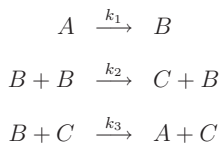
with $y \in \mathbb{R}^m$, $y_0 \in \mathbb{R}^m$ and $x \in [x_0, x_f]$. Function f is defined in $f : \mathbb{R}^{m+1} \rightarrow \mathbb{R}^m$, while the vector $y' \in \mathbb{R}^m$ contains the corresponding derivatives of y .

Expression (1) is called Initial Value Problem.

*E-Mail: tsitoura@teiha1.gr, URL address: <http://users.ntua.gr/tsitoura/>

†E-Mail: ifamelis@teiath.gr

An interesting example is the classical Robertson problem [14], that models the kinetics of an autocatalytic reaction . Let A , B and C be the chemical species involved with corresponding concentrations ${}_1y, {}_2y$ and ${}_3y$ and if k_1, k_2 and k_3 are the rates constants, then the structure of the reactions is sa follows:



Under the assumption of the mass action law for the rate functions and some idealized conditions, the mathematical modelling of the reaction is an IVP , of the form:

$$\begin{pmatrix} {}_1y' \\ {}_2y' \\ {}_3y' \end{pmatrix} = \begin{pmatrix} -0.04 {}_1y + 10^4 {}_2y {}_3y \\ 0.04 {}_1y - 10^4 {}_2y {}_3y - 3 \times 10^7 {}_2y^2 \\ 3 \times 10^7 {}_2y^2 \end{pmatrix}. \quad (2)$$

This problem has initial conditions

$${}_1y(0) = 1, {}_2y(0) = 0, {}_3y(0) = 0. \quad (3)$$

There are two things to take into consideration when dealing with such problems. First, the problem of stiffness is due to the large difference between the reaction rate constants, and its small but very quick initial transient. This phase is followed by a very smooth variation of the components where a large step-size would be appropriate for a numerical method.

Second, as in most chemistry models only second-order (bimolecular) reactions are taken into account, one mostly encounters functions that are quadratic in y [11, p. 5].

The former fact obliges us to deal with implicit methods. The latter drives us to derive methods with higher accuracy since we have to satisfy a smaller set of conditions.

2 Singly Diagonal Implicit Runge–Kutta methods.

The general s –stage embedded Runge–Kutta pair of orders $p(p-1)$, for the approximate solution of the problem (1) can be defined by the following Butcher scheme [2, 3]

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

where $A \in \mathfrak{R}^{s \times s}$ is strictly lower triangular, $b^T, \hat{b}^T, c \in \mathfrak{R}^s$ with $c = A \cdot e$ and $e = [1, 1, \dots, 1]^T \in \mathfrak{R}^s$. The vectors \hat{b}, b define the coefficients of the $(p-1)$ –th and p –th order approximations respectively.

Starting with a given value $y(x_0) = y_0$, this method produces approximations at the mesh points $x_0 < x_1 < x_2 < \dots < x_f$. Throughout this paper, we assume that local extrapolation is applied, hence the integration is advanced using the p –th order approximation. For estimating the error, two approximations are evaluated at each step from x_n to $x_{n+1} = x_n + h_n$. These are:

$$\hat{y}_{n+1} = y_n + h_n \sum_{j=1}^s \hat{b}_j f_j \quad \text{and} \quad y_{n+1} = y_n + h_n \sum_{j=1}^s b_j f_j, \quad (4)$$

where

$$f_i = f(x_n + c_i h_n, y_n + h_n \sum_{j=1}^s a_{ij} f_j), \quad i = 1, 2, \dots, s. \quad (5)$$

The local error estimate $E_n = \|y_n - \hat{y}_n\|$ of the $(p-1)$ –th order Runge–Kutta pair is used for the automatic selection of the step size. Given a tolerance parameter TOL , if $TOL > E_n$ the algorithm:

$$h_{n+1} = 0.9 \cdot h_n \cdot \left(\frac{TOL}{E_n}\right)^{\frac{1}{p}}, \quad (6)$$

provides the next step length. In case $TOL \leq E_n$ we reject the current step and retry by estimating a new stepsize using the same formula (6) replacing in the left side h_{n+1} by h_n .

Let $y_n(x)$ be the solution of the local initial value problem,

$$y'_n(x) = f(x, y_n(x)), \quad x \geq x_n, \quad y_n(x_n) = y_n.$$

. Then E_{n+1} is an estimate of the error of the local solution $y_n(x)$ at $x = x_{n+1}$. The method (4-5) is implicit as f_i is involved in both the right and left part of nonlinear (generally) equation (5). Thus f_i 's can not be evaluated explicitly and nonlinear systems of equations have to be solved. This system shares $(m \times s)$ equations with $(m \times s)$ unknowns. In each step we have to apply a modified Newton iteration scheme inverting the matrix $I_{m \times s} - hA \otimes \frac{\partial f}{\partial y}$.

For a method to be called Singly Diagonal Implicit has to satisfy

$$a_{ii} = d, \quad i = 1, 2, \dots, s \text{ and } a_{ij} = 0, \text{ for } j > i. \quad (7)$$

What we gain from this fact is that we have to deal with s systems with m unknowns. Thus Newton method involves the inversion of the much easier $I_m - h \frac{\partial f}{\partial y}$. Here we treat the stages one after the other and the specific matrix is common for all stages when implementing the Newton method.

The reason of using implicit methods instead of the explicit ones is the concept of A-stability which is essential when dealing with stiff systems. In this concept we consider problem (1) having the form:

$$y' = \lambda y, \quad (8)$$

with $\text{Re}(\lambda) < 0$. When a Runge-Kutta method is applied to (8) it yields

$$y_{n+1} = (1 + h\lambda b(I_m - h\lambda A)^{-1} \cdot e) \cdot y_n,$$

with $e = [1, 1, \dots, 1] \in \mathfrak{R}^s$. We set $z = \lambda h$ and $R(z) = 1 + zb(I_m - zA)^{-1} \cdot e$. Then, a method is called A-stable if and only if $|R(z)| < 1$ for every z with $\text{Re}(z) < 0$. Only implicit methods can be A-stable therefore we choose to work with such methods only.

Another useful property is L-stability. A method is called L-stable if it is A-stable and in addition

$$\lim_{z \rightarrow \infty} R(z) = 0. \quad (9)$$

An L-stable method damps out the transient phase much faster.

3 Derivation of the new method.

Let us now consider the quadratic ordinary differential equation:

$${}_k y' = \alpha_k + \sum_{i=1}^m \beta_{ki} {}_i y + \sum_{i=1}^m \sum_{j=1}^m \gamma_{ki} {}_i y {}_j y \quad (10)$$

for $k = 1, 2, \dots, m$.

The coefficients A , b and c of a conventional Runge–Kutta method have to satisfy certain conditions in order to attain an prescribed order of accuracy. These are given in Table 1 for a method of order five. The interested reader may see [9] for further details.

$$\begin{aligned} be = 1, & & bc = \frac{1}{2}, & & bc^2 - \frac{1}{3}, & & bAc = \frac{1}{6}, & & \frac{1}{4}bc^3 = \frac{1}{24}, \\ b(c * Ac) = \frac{1}{8}, & & bAc^2 = \frac{1}{12}, & & bA^2c = \frac{1}{24} & & bc^4 = \frac{1}{5}, & & bAc^3 = \frac{1}{20}, \\ b(c * Ac^2) = \frac{1}{15}, & & b(c^2 * Ac) = \frac{1}{10}, & & bA^2c^2 = \frac{1}{60}, & & b(c * A^2c) = \frac{1}{30}, & & bA^3c = \frac{1}{120}, \\ bA(c * Ac) = \frac{1}{40}, & & b(Ac)^2 = \frac{1}{20}. \end{aligned}$$

Table 1: Order conditions for a fifth order Runge–Kutta method.

When dealing with problems of the form (10) there is no need to satisfy all these seventeen equations but only the equations given in Table 2 [12].

$$\begin{aligned} be = 1, & & bc = \frac{1}{2}, & & bc^2 - \frac{1}{3}, & & bAc = \frac{1}{6}, & & b(c * Ac) = \frac{1}{8}, \\ bAc^2 = \frac{1}{12}, & & bA^2c = \frac{1}{24}, & & b(c * Ac^2) = \frac{1}{15}, & & bA^2c^2 = \frac{1}{60}, & & b(c * A^2c) = \frac{1}{30}, \\ bA^3c = \frac{1}{120}, & & bA(c * Ac) = \frac{1}{40}, & & b(Ac)^2 = \frac{1}{20}. \end{aligned}$$

Table 2: Order conditions for a fifth order quadratic Runge–Kutta method.

The thirteen equations of Table 2 and the four equations $\hat{b}e = 1$, $\hat{b}c = \frac{1}{2}$, $\hat{b}c^2 - \frac{1}{3}$, $\hat{b}Ac = \frac{1}{6}$, involving \hat{b} , form a set of conditions needed to derive a pair of orders five and three.

We consider a five stages pair as in [10, p. 100]. That pair had orders four and three, whereas we indent to construct a pair attaining orders five and three.

The twenty free parameters are $c_1 = d$, c_2, c_3, c_4, c_5 , a_{32}, a_{42}, a_{43} , a_{52}, a_{53}, a_{54} , b_1, b_2, b_3, b_4, b_5 and $\hat{b}_1, \hat{b}_2, \hat{b}_3, \hat{b}_5$. The rest coefficients are defined by (7) and the

$d = 0.2780538411364523$	$c_2 = -0.3676844045443509$	$c_3 = 0.4026030661794143$
$c_4 = 1.477424060656945$	$c_5 = 0.4789677054135209$	$a_{21} = -0.6457382456808033$
$a_{31} = -0.09776783840898377$	$a_{32} = 0.2223170634519457$	$a_{41} = -0.03971759296778165$
$a_{42} = 0.09093113685756394$	$a_{43} = 1.14815667563071$	$a_{51} = 0.4516391997886194$
$a_{52} = 0.0402931106382387$	$a_{53} = -0.01906448555386518$	$a_{54} = -0.02897550714589753$
$b_1 = 0.438321681756929$	$b_2 = 0.02688635109307992$	$b_3 = 0.03745399288026874$
$b_4 = 0.01837026885620139$	$b_5 = 0.4789677054135209$	$\hat{b}_1 = 0.3938856814975873$
$\hat{b}_2 = 0.04758554768869072$	$\hat{b}_3 = -0.01486594344074314$	$\hat{b}_5 = 0.5733947142544651$

Table 3: The new pair.

obligatory assumption $A \cdot e = c$. We set $\hat{b}_4 = 0$ to distinguish lower order weights from the higher order ones.

We solved the seventeen equations of Table- 2 and demanded $|R(z)| < 1$ for $x \in \mathcal{C}^- = \{z : Re(z) \leq 0\}$. We also required L-stability property (9). The coefficients of the resulting pair are listed in Table- 3.

Observe that the propagation function is

$$R(z) \approx \frac{1 - 0.390269z - 0.11713z^2 + 0.0296971z^3 + 0.0114377z^4}{1 - 1.39027z + 0.773139z^2 - 0.214974z^3 + 0.298872z^4 - 0.00166205z^5}.$$

Starting the process of deriving the coefficients we had to nullify the factor of z^5 in the numerator of $R(z)$. This leads to the polynomial equation $\frac{1}{120} - \frac{5}{24}c_1 + \frac{5}{3}c_1^2 - 5c_1^3 + 5c_1^4 - c_1^5 = 0$. The latter polynomial has five real roots. Only the chosen one lies in the interval that ensures L-stability [10, p. 98].

For this method we have constructed a third order continuous extension

$$y_{n+1} = y_n + h_n \sum_{j=1}^s b_j(\vartheta) f_j = y(x_n + \vartheta h_n) + O(h_n^4)$$

with

$$\begin{aligned} b_1(\vartheta) &= 1.43485027951414766 \vartheta - 1.19504225595235896 \vartheta^2 \\ &\quad - 0.183116142941936452 \vartheta^3 + 0.381629801137076787 \vartheta^4 \\ b_2(\vartheta) &= 0.215853035886902714 \vartheta - 0.579087229303158891 \vartheta^2 \\ &\quad + 0.567891501264597077 \vartheta^3 - 0.177770956755260981 \vartheta^4 \\ b_3(\vartheta) &= -0.382391279532112815 \vartheta + 2.04171664782253553 \vartheta^2 \\ &\quad - 2.07121080238737550 \vartheta^3 + 0.449339426977221524 \vartheta^4 \end{aligned}$$

$$\begin{aligned} b_4(\vartheta) &= 0.0371406079784377094 \vartheta - 0.0125127577943165203 \vartheta^2 \\ &\quad - 0.164027002731974498 \vartheta^3 + 0.157769421404054698 \vartheta^4 \\ b_5(\vartheta) &= -0.305452643847375271 \vartheta - 0.255074404772701160 \vartheta^2 \\ &\quad + 1.85046244679668937 \vartheta^3 - 0.810967692763092028 \vartheta^4. \end{aligned}$$

This method is useful for implementation purposes as well.

4 Implementation and Numerical Experiments.

In order to test the behavior of our New Method we compare it its similar to the 5-stage 4(3) order Diagonally Implicit Runge Kutta method of Hairer et. al. ([10, p. 100]). The numerical tests were done using the code SDIRK4, given in [8]. The implementation procedure in this code is thoroughly explained in [10, pp. 118-128]. The code was modified to apply our method too. This modifications take into account the fact that our method is of order five and that in our case the coefficients of the last stage are not equal to the b_i 's. The difference in the order of the methods force us to change slightly the stepsize control mechanism from the formula (6) used by the code to the formula:

$$h_{n+1} = 0.9 \cdot h_n \cdot \left(\frac{TOL}{\|E_n\|^{\frac{5}{4}}} \right)^{\frac{1}{5}}.$$

when our method is used to integrate the problems. Since $E_n = O(h^4)$, the above step-size change algorithm is asymptotically correct. Following the SDIRK4 code implementation we use the continuous solution of the new method too.

We have chosen four quadratic test problem originated from Chemical Engineering. We have solved the problems for various values of the tolerance parameter TOL ($10^{-6} - 10^{-10}$) and we have measured, in a reference solution point t_f , the absolute maximum error (maxer), the number of function evaluations (feval), the Jacobian evaluations (jeval), the total number of steps (nstep) and finally the number of rejected steps (nrej). For each problem

we present a table containing the results and we have plotted an Efficiency Curve graph where we plot $(\log_{10}(feval), \log_{10}(maxer))$ for each value of the tolerance parameter TOL .

The problems chosen for the tests are the following.

4.1 The Robertson Problem

The problem has been integrated on the interval $[0..10^{11}]$ with initial stepsize $h_0 = 10^{-6}$. As a reference solution at the end of the integration interval t_f has been taken from the source of the test set [13].

$$1y(t_f) = 0.208334015 \times 10^{-7}, \quad 2y(t_f) = 0.8333 \times 10^{-13}, \quad 3y(t_f) = 0.999999979166505$$

The results for this problem are presented in Table 4 and in Figure 1.

Method	tol	maxer	feval	jeval	nstep	rej
SDIRK4	10^{-6}	0.3344E - 08	1987	145	171	0
	10^{-7}	0.7899E - 09	3322	229	277	1
	10^{-8}	0.5601E - 09	5793	402	475	2
	10^{-9}	0.9838E - 10	10729	682	840	11
	10^{-10}	0.1480E - 09	18930	1172	1448	18
New	10^{-6}	0.2640E - 08	1966	117	135	0
	10^{-7}	0.1288E - 07	2398	155	173	0
	10^{-8}	0.1825E - 09	3567	256	270	1
	10^{-9}	0.8130E - 11	5438	421	436	1
	10^{-10}	0.4879E - 11	9024	724	745	3

Table 4: Robertson results

4.2 The Hires problem

The 'High Irradiance Responses' (Hires) problem [7, 15] is a problem of eight nonlinear Ordinary Differential Equations which models the involvement of light in photomorphogenesis. The mathematical form of the problem is

$$\begin{pmatrix} 1y' \\ 2y' \\ 3y' \\ 4y' \\ 5y' \\ 6y' \\ 7y' \\ 8y' \end{pmatrix} = \begin{pmatrix} -1.71 \ 1y + 0.43 \ 2y + 8.32 \ 3y + 0.0007 \\ 1.71 \ 1y - 8.75 \ 2y \\ -10.03 \ 3y + 0.43 \ 4y + 0.035 \ 5y \\ 8.32 \ 2y + 1.71 \ 3y - 1.12 \ 4y \\ -1.745 \ 5y + 0.43 \ 6y + 0.43 \ 7y \\ -280 \ 6y \ 8y + 0.69 \ 4y + 1.75 \ 5y - 0.43 \ 6y + 0.69 \ 7y \\ 280 \ 6y \ 8y - 1.81 \ 7y \\ -280 \ 6y \ 8y + 1.81 \ 7y \end{pmatrix}$$

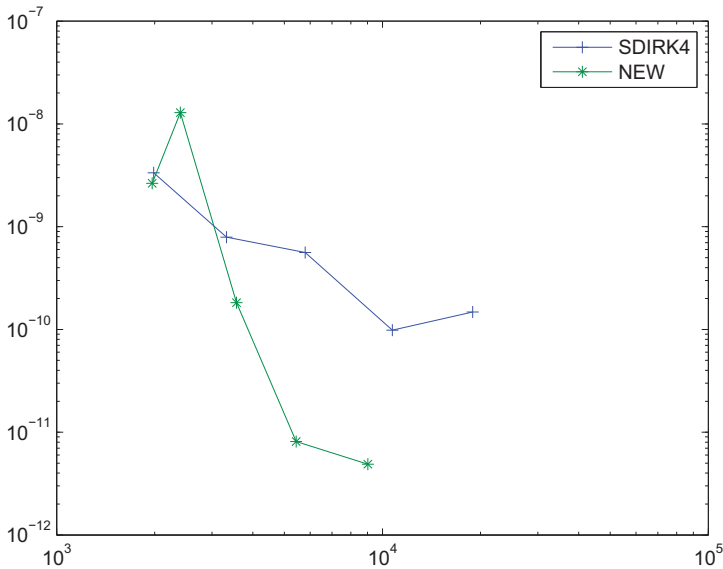


Figure 1: The Robertson Problem Efficiency Comparison curves

with initial conditions being

$${}_1y(0) = 1, {}_2y(0) = {}_3y(0) = {}_4y(0) = {}_5y(0) = 0 = {}_6y(0) = {}_7y(0) = 0, {}_8y(0) = 0.0057.$$

The above nonlinear system of odes has been integrated on the interval [0..321.8122]

with initial stepsize $h_0 = 10^{-6}$ and the reference solution ([13]) on t_f is

$$\begin{aligned} {}_1y(t_f) &= 0.7371312573325668 \times 10^{-3}, & {}_2y(t_f) &= 0.1442485726316185 \times 10^{-3}, \\ {}_3y(t_f) &= 0.5888729740967575 \times 10^{-4}, & {}_4y(t_f) &= 0.1175651343283149 \times 10^{-2}, \\ {}_5y(t_f) &= 0.2386356198831331 \times 10^{-2}, & {}_6y(t_f) &= 0.6238968252742796 \times 10^{-2}, \\ {}_7y(t_f) &= 0.2849998395185769 \times 10^{-2}, & {}_8y(t_f) &= 0.2850001604814231 \times 10^{-2}. \end{aligned}$$

The results for the HIRES problem are given in Table 5 and in Figure 2.

4.3 The Orego Problem

The third problem, maned Oregonator, which was named after The University of Oregon where in 1972 was proposed to model the Belousov-Zhabotinskii

Method	tol	maxer	feval	jeval	nstep	rej
SDIRK4	10^{-6}	$0.1066E - 05$	1005	40	100	1
	10^{-7}	$0.1519E - 05$	1628	71	176	2
	10^{-8}	$0.9175E - 07$	3096	136	341	1
	10^{-9}	$0.1035E - 06$	6461	274	726	2
	10^{-10}	$0.1014E - 07$	13612	510	1539	0
New	10^{-6}	$0.4356E - 05$	978	43	79	0
	10^{-7}	$0.1904E - 06$	1625	71	134	3
	10^{-8}	$0.1509E - 06$	2941	109	263	12
	10^{-9}	$0.2357E - 08$	5498	196	522	0
	10^{-10}	$0.3636E - 09$	11850	381	1154	1

Table 5: Hires results

(BZ) reaction [6, 5]. It consist of a stiff system of three non-linear ODEs.

$$\begin{pmatrix} 1y' \\ 2y' \\ 3y' \end{pmatrix} = \begin{pmatrix} 77.27(2y - 1y 2y + 1y - 8.375 \times 10^{-6} 1y^2) \\ \frac{1}{77.27}(-2y - 1y 2y + 3y) \\ 0.161(1y - 3y) \end{pmatrix}$$

with initial conditions $1y(0) = 1$, $2y(0) = 2$, $3y(0) = 3$.

When certain reactants, like bromous acid, bromide ion and cerium ion, are combined, they exhibit a chemical reaction which after an induction period of inactivity, oscillates with change in structure and in color, from red to blue and viceversa. The problem has been integrated in the interval $[0..360]$ with initial stepsize $h_0 = 10^{-6}$. Again a reference solution at the end of the integration interval t_f has been taken from the source of the test set [13],

$$1y(t_f) = 1.00081487031852, 2y(t_f) = 1228.17852154988, 3y(t_f) = 132.055494284651.$$

Table 6 and Figure 3 present the numerical results for the Orego problem.

Method	tol	maxer	feval	jeval	nstep	rej
SDIRK4	10^{-6}	$0.1943E - 03$	15871	921	1644	15
	10^{-7}	$0.2343E - 04$	34350	1582	3743	4
	10^{-8}	$0.1859E - 05$	75667	2716	8418	3
	10^{-9}	$0.1507E - 06$	168965	3801	18668	3
	10^{-10}	$0.1433E - 07$	374773	5506	41041	10
New	10^{-6}	$0.5638E - 04$	15083	840	1240	7
	10^{-7}	$0.1773E - 05$	31348	1424	2786	4
	10^{-8}	$0.1364E - 06$	69532	2080	6474	2
	10^{-9}	$0.1943E - 07$	160876	2932	14907	3
	10^{-10}	$0.7103E - 08$	359600	4242	33473	1

Table 6: Orego results

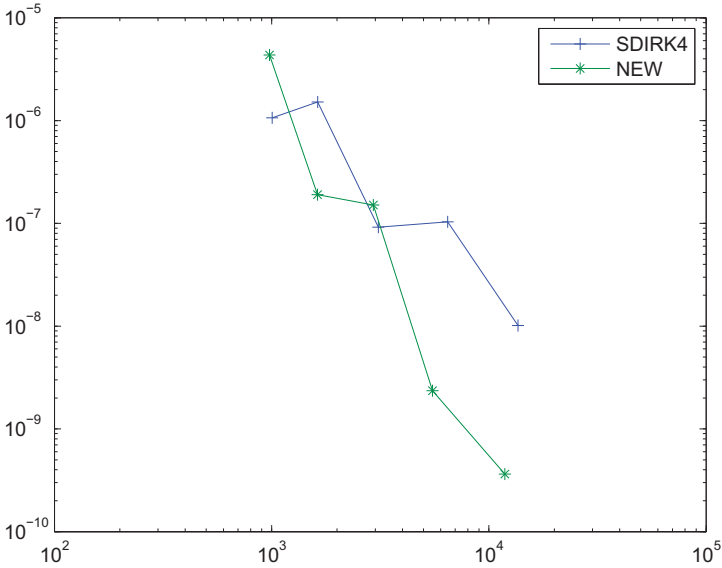


Figure 2: The HIRES Problem Efficiency Comparison curves

4.4 The F5 Problem

Our final choice is a model for chemical Kinetics known as F5 problem taken from the Detest Test Set [4].

$$\begin{pmatrix} 1y' \\ 2y' \\ 3y' \\ 4y' \end{pmatrix} = \begin{pmatrix} 10^{11}(-3 \ 1y \ 2y + 0.0012 \ 4y - 9 \ 1y \ 3y) \\ -3 \times 10^{11} \ 1y \ 2y + 2 \times 10^7 \ 4y \\ 10^{11}(-9 \ 1y \ 3y + 0.001 \ 4y) \\ 10^{11}(3 \ 1y \ 2y - 0.0012 \ 4y + 9 \ 1y \ 3y) \end{pmatrix} \quad (11)$$

Its initial conditions are

$$1y(0) = 3.365 \times 10^{-7}, \quad 2y(0) = 8.261 \times 10^{-3}, \quad 3y(0) = 8.261 \times 10^{-3}, \quad 4y(0) = 9.38 \times 10^{-6}.$$

The above system of ODEs has been integrated in the interval [0..100] with initial stepsize $h_0 = 10^{-7}$ while the reference solution ([4]) on t_f is:

$$\begin{aligned} 1y(t_f) &= 1.713564284690712 \times 10^{-7}, & 2y(t_f) &= 3.713563071160676 \times 10^{-3}, \\ 3y(t_f) &= 6.189271785267793 \times 10^{-3}, & 4y(t_f) &= 9.545143571530929 \times 10^{-6}. \end{aligned}$$

Finally, the F5 problem numerical testing results are included in Table 7 and in Figure 4.

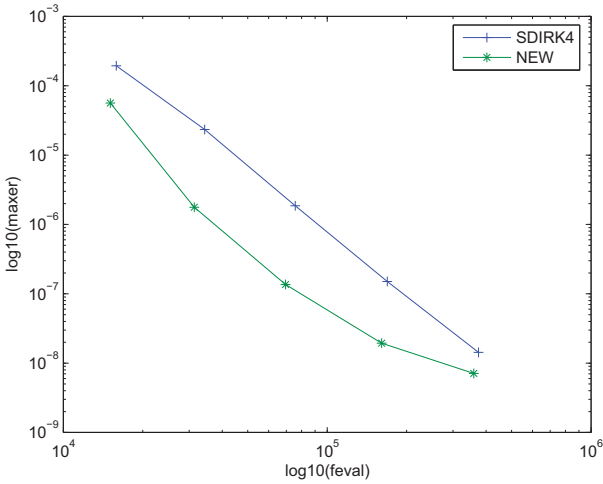


Figure 3: The Orejo Problem Efficiency Comparison curves

4.5 Conclusions

Interpreting the results it is obvious that at least one digit of accuracy was gained by using the new pair. This is justified since we manage to increase the algebraic order of our method without any extra cost.

Method	tol	maxer	feval	jeval	nstep	rej
SDIRK4	10^{-6}	$0.2965E - 09$	261	10	29	0
	10^{-7}	$0.7597E - 11$	392	16	41	0
	10^{-8}	$0.3220E - 10$	596	26	59	0
	10^{-9}	$0.1908E - 10$	1158	56	107	0
	10^{-10}	$0.3069E - 10$	2133	96	189	4
New	10^{-6}	$0.1868E - 11$	293	11	27	1
	10^{-7}	$0.1837E - 11$	377	15	33	0
	10^{-8}	$0.2080E - 11$	550	24	47	1
	10^{-9}	$0.3369E - 11$	827	34	69	1
	10^{-10}	$0.3176E - 11$	1344	62	117	3

Table 7: F5 results

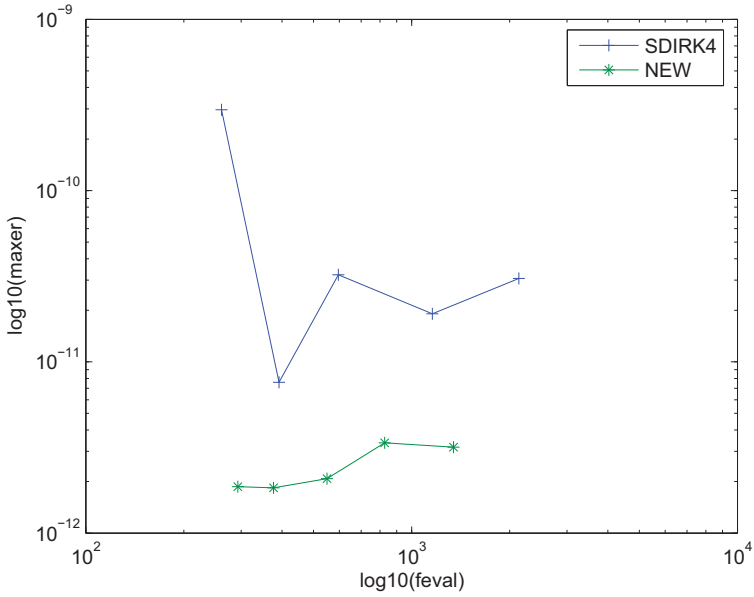


Figure 4: The F5 Problem Efficiency Comparison curves

References

- [1] J. C. Butcher, Coefficients for the study of Runge-Kutta integration processes, *J. Austr. Math. Soc.*, 3 (1963) 185–201.
- [2] J. C. Butcher, Implicit Runge-Kutta processes, *Math. Comput.*, 18 (1964) 50–64.
- [3] J. C. Butcher, On Runge-Kutta processes of high order, *J. Austral. Math. Soc.*, 4 (1964) 179–194.
- [4] W.H. Enright, J.D. Pryce, Two FORTRAN packages for assessing initial value methods, *ACM Transactions on Mathematical Software*, 13 (1987) 1–27.
- [5] R.J. Field, E. Kö r ö s, and R.M. Noyes, Oscillation in chemical systems II Thorough analysis of temporal oscillations in the bramate-cerium-malonic acid systems, *J. Amer. Chem. Soc.*, 94 (1972) 8649–8664.

- [6] C. Gray, An analysis of the Belousov-Zhaboyinskii reaction, *Rose-Hulman Undergraduate Mathematics Journal*, 3(1), 2002. <http://www.rose-hulman.edu/mathjournal>
- [7] B. A. Gottwald, MISS- ein einfaches Simulations-System für biologische und chemische Prozesse, *EDV in Medizin und Biologie*, 3 (1977) 85–90.
- [8] E. Hairer, Fortran and Matlab Codes, Software from the Internet, <http://www.unige.ch/hairer/software.html>
- [9] E. Hairer, S. Nørsett and G. Wanner, *Solving Ordinary differential equations I: Nonstiff problems*, Springer-Verlag, Berlin, 1993.
- [10] E. Hairer, G. Wanner, *Solving Ordinary differential equations II: Stiff and Differential-Algebraic problems*, Springer-Verlag, Berlin, 1996.
- [11] W. Hundsdorfer and J. G. Verwer, *Numerical solution of time-dependent advection-diffusion reaction equations*, Springer-Verlag, Berlin, 2003.
- [12] A. Iserles, G. Ramaswami and m. Sofroniou, Runge-Kutta methods for quadratic ordinary differential equations, *BIT* 38, (1998) 315–346.
- [13] F. Mazzia and F. Iavernaro, *Test set for initial value problems solvers*, Department of Mathematics, University of Bari, 2003.
- [14] H. H. Robertson, *The solution of a set of reaction rate equations*, pp. 178-182, Academic Press, 1966.
- [15] E. Schäfer, A new approach to explain the 'high irradiance responses' of photomorphogenesis on the basis of phytochrome., *J. Math. Biol.*, 2 (1975) 41–56.