Monografías Matemáticas *García de Galdeano* **40**, 77–86 (2016)

On the performance of low storage Additive Runge-Kutta methods

Inmaculada Higueras and Teo Roldán

Abstract. Given a differential system that involves terms with different stiffness properties, a natural approach to obtain numerical approximations is the use of implicit-explicit time-discretizations. These systems, often with a large number of equations, arise from the semidiscretization of some time-dependent partial differential equations. In the construction of Runge-Kutta methods, properties like stability and accuracy are important items that must be taken into account. However, in some contexts, storage requirements of the schemes also play an important role. When the high dimension of the problem compromises the computer memory capacity, it is important to incorporate low memory usage to other properties of the scheme.

In a recent work the authors have studied and constructed implicit-explicit Runge-Kutta methods with low-storage requirements. In this paper we develop and test new low-storage Runge-Kutta methods that complete the study done in the mentioned work.

Keywords: Additive Runge-Kutta methods, low-storage, stiff problems.

AMS classification: 65L04, 65L06, 62L20.

§1. Introduction

Space discretization of some time-dependent partial differential equations (PDEs) gives rise to systems of ordinary differential equations in additive form. These systems, often with a large number of equations, arise, e.g., from semidiscretisations of convection-diffusion problems and hyperbolic systems with relaxation [2, 5, 11, 15, 16].

When the differential system involves terms with different stiffness properties, a natural approach to obtain numerical approximations is the use of implicit-explicit (IMEX) time-discretizations. IMEX Runge-Kutta methods have been deeply studied in the literature (see, e.g., [1, 4, 2, 11, 15, 19]).

Sometimes, the additive differential system is of the form

$$\begin{aligned} u' &= f_1(u,v)\,, \\ v' &= f_2(u,v) + \frac{1}{\varepsilon}\, g_2(u,v)\,, \end{aligned} \tag{1}$$

where ε is the stiffness parameter. These systems have been considered, e.g., in [1, 3, 4, 2, 10, 15, 16], where robust IMEX Runge-Kutta methods have been analyzed. In particular, in [4, 2], uniform convergence in the stiffness parameter ε has been studied.

For systems with a large number of equations, memory storage requirement is an important issue. When the high dimension of the problem compromises the computer memory capacity, it is important to incorporate low memory usage to some other properties of the scheme. These ideas have been developed in [6, 7, 8, 9, 10, 12, 13, 14, 17, 18], where different low-storage Runge-Kutta methods have been constructed. In particular, in [6, 12], explicit Runge-Kutta methods are implemented by using the van der Houwen format [17], while diagonally implicit Runge-Kutta (DIRK) schemes have been explored in [14]. Robust IMEX Runge-Kutta methods with low-storage requirements were recently constructed in [10]. In particular, new Additive Semi-Implicit Runge-Kutta methods (ASIRK), a special class of IMEX Runge-Kutta methods, have been developed. These new methods can be implemented by using just three memory registers. In this paper we complete that work constructing new low-storage methods.

The rest of the paper is organized as follows. In section 2 we briefly introduce ASIRK methods, a special class of IMEX Runge-Kutta schemes. In section 3 we review low-storage ASIRK implementations in three memory registers. Taking into account the results obtained in [10], we construct new low-storage ASIRK schemes in section 4. Some numerical experiments are displayed in section 5.

§2. Additive Semi-Implicit Runge-Kutta methods

We consider a special class of IMEX Runge-Kutta schemes that have been used in [19] looking for computational efficiency. These schemes are usually referred as ASIRK-sA methods. The numerical solution of a general additive differential problem

$$y' = f(y) + g(y)\,, \qquad y(t_0) = y_0\,, \tag{2}$$

with an ASIRK-sA method is given by

$$y_{n+1} = y_n + \sum_{i=1}^{s} \omega_i\, K_{i,n+1}\,, \tag{3}$$

where the internal derivatives $K_{i,n+1}$ are given by

$$K_{i,n+1} = h\Big(f(y_n + \sum_{j=1}^{i-1} b_{ij}\, K_{j,n+1}) + g(y_n + \sum_{j=1}^{i-1} c_{ij}\, K_{j,n+1} + c_{ii}\, K_{i,n+1})\Big), \quad i = 1, \dots, s\,, \tag{4}$$

and b_{ij}, c_{ij}, ω_j are the coefficients of the method. Observe that the resulting scheme is explicit in f and diagonally implicit in g. On the following, we denote the matrices and vector containing the coefficients of the ASIRK-sA method by $\mathcal{B} = (b_{ij})$, $C = (c_{ij})$, and ω, respectively. Below we display these matrices for $s = 3$,

$$\mathcal{B} = \begin{pmatrix} 0 & 0 & 0 \\ b_{21} & 0 & 0 \\ b_{31} & b_{32} & 0 \end{pmatrix}, \qquad C = \begin{pmatrix} c_{11} & 0 & 0 \\ c_{21} & c_{22} & 0 \\ c_{31} & c_{32} & c_{33} \end{pmatrix}, \qquad \omega = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}. \tag{5}$$

Even though method (3)-(4) can be considered as an additive Runge-Kutta method, it is important to notice that the coefficients in (5) are not the standard coefficients of an additive Runge-Kutta scheme. However, as we show below, equations (3)-(4) can be rewritten as an additive Runge-Kutta method (9). With this approach, some issues for ASIRK-sA methods, e.g., the set of order conditions, can be obtained in a simple way from the theory of additive Runge-Kutta methods.

Equations (3)-(4) are given in terms of $K_{n+1} = (K^t_{1,n+1}, \dots, K^t_{s,n+1})^t$, the internal derivative vector. By using the Kronecker product $\otimes$ and denoting $K_{n+1} = h\,(F(Y_{n+1}) + G(\hat{Y}_{n+1}))$, where $F(Y_{n+1}) = (f(Y_{1,n+1})^t, \dots, f(Y_{s,n+1})^t)^t$ and $G(\hat{Y}_{n+1}) = (g(\hat{Y}_{1,n+1})^t, \dots, g(\hat{Y}_{s,n+1})^t)^t$, it is possible to rewrite (3)-(4) as

$$y_{n+1} = y_n + h\,(\omega^t \otimes I_k)\,(F(Y_{n+1}) + G(\hat{Y}_{n+1}))\,, \tag{6}$$

where the internal stages $Y_{n+1} = (Y^t_{1,n+1}, \dots, Y^t_{s,n+1})^t$ and $\hat{Y}_{n+1} = (\hat{Y}^t_{1,n+1}, \dots, \hat{Y}^t_{s,n+1})^t$ are obtained from

$$Y_{n+1} = e \otimes y_n + (\mathcal{B} \otimes I_k)\,(h\,F(Y_{n+1}) + h\,G(\hat{Y}_{n+1}))\,, \tag{7}$$

$$\hat{Y}_{n+1} = e \otimes y_n + (C \otimes I_k)\,(h\,F(Y_{n+1}) + h\,G(\hat{Y}_{n+1}))\,. \tag{8}$$

This formulation corresponds to a $2s$-stage additive Runge-Kutta method, where the $2s$ internal stages are Y_{n+1} and $\hat{Y}_{n+1}$. Then, the double Butcher tableau of scheme (6)-(8) is

$$\begin{array}{c|cc|cc} \mathcal{B}e & \mathcal{B} & 0 & 0 & \mathcal{B} \\ Ce & C & 0 & 0 & C \\ \hline & \omega^t & 0 & 0 & \omega^t \end{array} \tag{9}$$

Now, from the set of order conditions for IMEX methods, it is easy to derive the set of order conditions for ASIRK-sA scheme (3)-(4) in terms of matrices $\mathcal{B}$, C, and vector ω (see [10] for details).

In the same way, it is posible to derive stability properties of ASIRK-sA methods (3)-(4) applying some known results on linear stability for additive Runge-Kutta methods [11, 15, 19]; our approach is similar to the one made in [15]. Thus, we consider the simplified linear model equation

$$y' = \xi_1\, y + \xi_2\, y\,, \qquad y(0) = 1\,, \tag{10}$$

where $\xi_1, \xi_2 \in \mathbb{C}$, with $\mathrm{Re}\,(\xi_2) < 0$, and we solve the model equation (10) with method (3)-(4). In this way, we can write the numerical solution as

$$y_1 = R(z_1, z_2)\,,$$

where the function of absolute stability $R(z_1, z_2)$ is obtained in terms of the coefficients of the method and $z_1 = \xi_1\, h$, $z_2 = \xi_2\, h$. We are interested in obtaining the largest set

$$S_1 = \{z_1 \in \mathbb{C} \;:\; \sup_{z_2 \in \mathbb{C}^-} |R(z_1, z_2)| \le 1\}\,. \tag{11}$$

Besides, in order to get a correct asymptotic decay for the stiff terms, the L-stability condition for the implicit step, that is,

$$\lim_{z_2 \to -\infty} R(0, z_2) = 0\,, \tag{12}$$

must be imposed. For more details, see [15].

We are also interested in stiff accuracy, that is, accuracy of the numerical solution of problem (1) for small values of ε. This issue has been studied in [2, 15] for IMEX Runge-Kutta methods. The analysis done leads to the following additional order conditions for an IMEX Runge-Kutta method with coefficients (A, b^t, c), $(\tilde{A}, \tilde{b}^t, \tilde{c})$,

$$b^t A^{-1} \tilde{c} = 1\,, \quad b^t A^{-1} \tilde{c}^2 = 1\,, \quad b^t A^{-1} \tilde{A} \tilde{c} = 1/2\,. \tag{13}$$

For ASIRK-sA methods, see [10] for details, we get these additional orden conditions

$$\omega^t\, e = 1\,, \quad \omega^t\, C^{-1}\, (Ce)^2 = 1\,, \quad \omega^t\, \mathcal{B}\, e = 1/2\,. \tag{14}$$

The linear system

$$\begin{cases} u' = \delta_1\, u + \sigma_1\, v\,, \\ v' = \delta_2\, u + \sigma_2\, v + \dfrac{1}{\varepsilon}(c\, u - v)\,, \end{cases} \tag{15}$$

is a particular case of (1). If we consider consistent initial values, $v_0 = c\, u_0$, then, after one time step, the exact solution of (15) satisfies

$$\begin{aligned} u(\varepsilon, h) &= u_0 \Big(1 + \hat{a}\, h + \tfrac{1}{2}\, \hat{a}^2\, h^2 + \sigma_1\, \hat{b}\, h\, \varepsilon\Big) + O(h^3, h\, \varepsilon^2)\,, \\ v(\varepsilon, h) &= u_0 \Big((1 + \hat{a}\, h + \tfrac{1}{2}\, \hat{a}^2\, h^2)\, c + \hat{b}\, \varepsilon + (\hat{a} + \sigma_1\, c)\, \hat{b}\, h\, \varepsilon\Big) + O(h^3, h\, \varepsilon^2)\,, \end{aligned}$$

where, as in [15], we have denoted $\hat{a} = \delta_1 + \sigma_1\, c$, and $\hat{b} = \delta_2 + (\sigma_2 - \delta_1)\, c - \sigma_1\, c^2$. We will construct the numerical solution $(u_1(\varepsilon, h), v_1(\varepsilon, h))$ of system (15) with the ASIRK-sA scheme (3)-(4), and from the difference between the numerical and the exact solution

$$u_1(\varepsilon, h) - u(\varepsilon, h)\,, \qquad v_1(\varepsilon, h) - v(\varepsilon, h)\,, \tag{16}$$

we will get conditions on the coefficients of the method.

§3. ASIRK methods implemented in three memory registers

In this section we consider a kind of ASIRK-sA methods that can be implemented in an efficient way by using three memory registers for any number of stages. The way this schemes have been developed can be seen with details in [10]. The coefficients of these methods have a particular structure that is mainly enforced by the possibility of implementing the scheme in three memory registers. The ASIRK-sA methods we propose are of the form

$$\mathcal{B} = \begin{pmatrix} 0 & & & & \\ \omega_1 + \gamma_1 & 0 & & & \\ \omega_1 & \omega_2 + \gamma_2 & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \\ \omega_1 & \omega_2 & \dots & \omega_{s-1} + \gamma_{s-1} & 0 \end{pmatrix}, \quad C = \begin{pmatrix} \lambda_1 & & & & \\ \omega_1 & \lambda_2 & & & \\ \omega_1 & \omega_2 & \lambda_3 & & \\ \vdots & \vdots & \ddots & \ddots & \\ \omega_1 & \omega_2 & \dots & \omega_{s-1} & \lambda_s \end{pmatrix}, \quad \omega = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_s \end{pmatrix}. \tag{17}$$

For example, a low-storage ASIRK-3A method can be displayed in this way

$$\begin{aligned}
K_1 &= h\,f(y_n) + h\,g(y_n + \lambda_1\,K_1)\,,\\
K_2 &= h\,f(y_n + \omega_1\,K_1 + \gamma_1\,K_1) + h\,g(y_n + \omega_1\,K_1 + \lambda_2\,K_2)\,,\\
K_3 &= h\,f(y_n + \omega_1\,K_1 + \omega_2\,K_2 + \gamma_2\,K_2) + h\,g(y_n + \omega_1\,K_1 + \omega_2\,K_2 + \lambda_3\,K_3)\,,\\
y_{n+1} &= y_n + \omega_1\,K_1 + \omega_2\,K_2 + \omega_3\,K_3\,.
\end{aligned}$$

For each stage, a memory register (Register 1) is used for the storage of $y_n + \sum_{j=1}^{i-1}\omega_j\,K_j$,

$$\text{Register 1:}\quad Y_i = Y_{i-1} + \omega_{i-1}\,K_{i-1}\,,\quad i = 1,\ldots,s+1,$$

where we consider $b_0 = 0$ and $Y_0 = y_n$. After the last stage, we obtain the numerical solution y_{n+1} in the Register 1 as $Y_{s+1} = Y_s + \omega_s\,K_s$. The second memory register (Register 2) is used for the storage of the evaluation of the function f,

$$\text{Register 2:}\quad L_i = h\,f(Y_i + \gamma_{i-1}\,K_{i-1})\,,\quad i = 1,\ldots,s,$$

where $\gamma_0 = 0$. Finally, the third memory register (Register 3) is used for the internal derivative K_i

$$\text{Register 3:}\quad K_i = L_i + h\,g(Y_i + \lambda_i\,K_i)\,,\quad i = 1,\ldots,s. \tag{18}$$

§4. A new method implemented in three memory registers

Given a low-storage ASIRK-sA method of the form (17), we determine its coefficients by imposing accuracy, stiff accuracy and stability properties. In [10] the cases of two stages, $s = 2$, and three stages, $s = 3$, were studied in detail. In this paper we consider $s = 3$ and we make a different choice of one of the free parameters.

Observe that if $\lambda_s = \omega_s$ in (17), then the implicit scheme is stiffly accurate. Besides, in this case, the implicit part of the method satisfies condition (12). Accordingly, we set $\lambda_3 = \omega_3$ and we impose the first order condition $\omega^t e = 1$, that is, $\omega_3 = 1-\omega_1-\omega_2$. With these choices, from schemes (17) we get a family of ASIRK-3A methods with 6 parameters,

$$\mathcal{B} = \begin{pmatrix} 0 & 0 & 0\\ \omega_1+\gamma_1 & 0 & 0\\ \omega_1 & \omega_2+\gamma_2 & 0\end{pmatrix},\quad C = \begin{pmatrix} \lambda_1 & 0 & 0\\ \omega_1 & \lambda_2 & 0\\ \omega_1 & \omega_2 & 1-\omega_1-\omega_2\end{pmatrix},\quad \omega = \begin{pmatrix} \omega_1\\ \omega_2\\ 1-\omega_1-\omega_2\end{pmatrix}. \tag{19}$$

For methods (19), second order is obtained if the parameters satisfy

$$\omega_2(\omega_1+\gamma_1) + (-\omega_1-\omega_2+1)(\omega_1+\omega_2+\gamma_2) = \frac{1}{2}\,, \tag{20a}$$

$$\omega_1\,\omega_2 + \omega_1\,\lambda_1 - \omega_1 + \omega_2\,\lambda_2 - \omega_2 + 1 = \frac{1}{2}\,. \tag{20b}$$

It can be checked that, with the 6 parameters in (19), it is not possible to achieve order three.

For second order ASIRK-3A methods of the form (19), the additional order conditions (14) are satisfied; observe that in (19), the last row of C is equal to ω.

Furthermore, in (19) we set $\lambda = \lambda_1 = \lambda_2$. In this way, if function g is linear, or if we use Newton-like methods for solving the nonlinear systems, the same LU-factorization can be used when the first two internal derivatives are computed (see equation (18)).

From the stiff accuracy analysis (16), cancellation of the term εh in the u variable, and the term ε^2/h in the v variable, leads to the equations

$$\frac{\omega_2\lambda(\omega_1+\gamma_1)+(\omega_1+\omega_2-1)\,(\omega_1(\omega_2+\gamma_2)-\lambda(\omega_1+\omega_2+\gamma_2))}{\lambda^2} = 1\,, \tag{21}$$

$$\frac{(\omega_1-\lambda)(\omega_2-\lambda)}{\lambda^2(\omega_1+\omega_2-1)} = 0\,. \tag{22}$$

If (21) and (22) hold, the difference between the numerical and the exact solution of problem (15) is of the form

$$u_1(\varepsilon,h)-u(\varepsilon,h) = O(h^3, h^2\,\varepsilon)\,, \qquad v_1(\varepsilon,h)-v(\varepsilon,h) = O(h^3, h\,\varepsilon)\,. \tag{23}$$

In order to impose condition (22), there exists two posibilities: $\lambda = \omega_1$ or $\lambda = \omega_2$. In [10] only the case $\lambda = \omega_1$ was studied. If we substitute this value in the other three equations, namely, (20a), (20b) and (21), and we solve the corresponding system, we get a ω_1-family of second order methods such that the implicit method is L-stable, provided that it is A-stable. Furthermore, the methods satisfy the additional conditions (14), they can be implemented by using just three memory registers and, for the model problem (15), the errors are given by (23).

In order to choose the parameter ω_1 in this family of methods, different approaches can be followed. If we minimized the local truncation error, then we obtain the scheme named *ASIRK-LSe(3,2)*. For more details see [10].

In this paper we study the other posibility, that is $\lambda = \omega_2$ in (22). If we substitute this value in the other three equations (20a), (20b) and (21), then, from the corresponding system we get

$$\begin{aligned}
\gamma_2 &= \frac{-4\omega_2^5-4\omega_2^4+12\omega_2^3-6\omega_2^2+\omega_2}{4\omega_2^4-12\omega_2^3+12\omega_2^2-6\omega_2+1}\,,\\
\omega_1 &= \frac{-2\omega_2^2+2\omega_2-1}{2(2\omega_2-1)}\,,\\
\gamma_1 &= \frac{8\omega_2^6-72\omega_2^5+140\omega_2^4-112\omega_2^3+46\omega_2^2-10\omega_2+1}{4\omega_2(2\omega_2-1)^2\left(2\omega_2^2-2\omega_2+1\right)}\,.
\end{aligned} \tag{24}$$

This is another ω_2-family of second order methods with the same properties as the family obtained for the case $\lambda = \omega_1$ in [10]. We can follow again different approaches in order to choose the parameter ω_2 in (24). We can minimize the local truncation error, but also we can optimize the stability region (11). In both cases the derived methods have a similar behavior, so in this work we will only show the method obtained by minimizing the local truncation error.

ASIRK-LSe2(3,2) method

We have denoted by ASIRK-LSe2(3,2) the method in (24) that minimizes the sum of the absolute values of the coefficients in the leading truncation error term. The optimum value for this sum is 0.563, and it is obtained for $\omega_2 = 0.144288$. We remark that the optimum value is significantly higher than the one obtained for the method ASIRK-LSe(3,2) in [10], namely 0.218. A rational approximation of $\omega_2 = 0.144288$ is $\omega_2 = 1/7$. In that case we obtain the method.

$$\mathcal{B} = \begin{pmatrix} 0 & 0 & 0 \\ \frac{41663}{25900} & 0 & 0 \\ \frac{37}{70} & \frac{250}{851} & 0 \end{pmatrix}, \quad C = \begin{pmatrix} \frac{1}{7} & 0 & 0 \\ \frac{37}{70} & \frac{1}{7} & 0 \\ \frac{37}{70} & \frac{1}{7} & \frac{23}{70} \end{pmatrix}, \quad w = \begin{pmatrix} \frac{37}{70} \\ \frac{1}{7} \\ \frac{23}{70} \end{pmatrix}. \tag{25}$$

§5. Numerical Experiments

In this section we study the performance of the numerical scheme constructed in this paper, named ASIRK-LSe2(3,2), and we compare it with the method ASIRK-LSe(3,2) given in [10], and the ASIRK-3A method considered by Zhong in [19].

In [10] we tested the methods on different problems but here we will just see a simple prototype of stiff system of the form

$$\begin{cases} u' = -v\,, \\ v' = u + \dfrac{1}{\varepsilon}(e(u) - v)\,, \end{cases} \tag{26}$$

with $e(u) = \sin u$. We can choose arbitrarily the initial value $u(0)$, but there is no freedom in the choice of $v(0)$. If we consider non-consistent initial values, then the solution presents an initial layer in the v component when $\varepsilon \to 0$. We have chosen $u(0) = \pi/2$ and we have integrated the problem in $[0, 1]$, with time step $h = 0.05$, assuming different initial data $v(0)$. Consistent initial values are obtained if we take $v(0) = \sin(u(0)) = 1$. By adding a small perturbation δ, we get non-consistent initial values; in the numerical experiments we have taken $\delta = 0.05$. Finally we have also considered *well prepared* initial data to obtain a smooth solution.

$$\begin{aligned} &\mathrm{C}_{\mathrm{InVal}}: && v(0) = 1\,, \\ &\mathrm{NC}_{\mathrm{InVal}}: && v(0) = 1 + \delta\,, \\ &\mathrm{WP}_{\mathrm{InVal}}: && v(0) = 1 + \frac{\pi}{2}\varepsilon - \frac{\pi}{2}\varepsilon^3\,. \end{aligned} \tag{27}$$

For each method we compute the convergence rates for a wide range of values of the parameter ε; in our tests we consider $\varepsilon = 10^{-j}$, $j = 0, 1, 2, \ldots, 6$. We try to check whether the convergence is uniform in ε, particularly in the intermediate regime. In order to show the convergence rates, for each value of ε we compute E_h and $E_{h/2}$, an estimation of the relative L_∞-global error with stepsizes h and $h/2$, respectively. With these values we compute the convergence rate in the standard way.

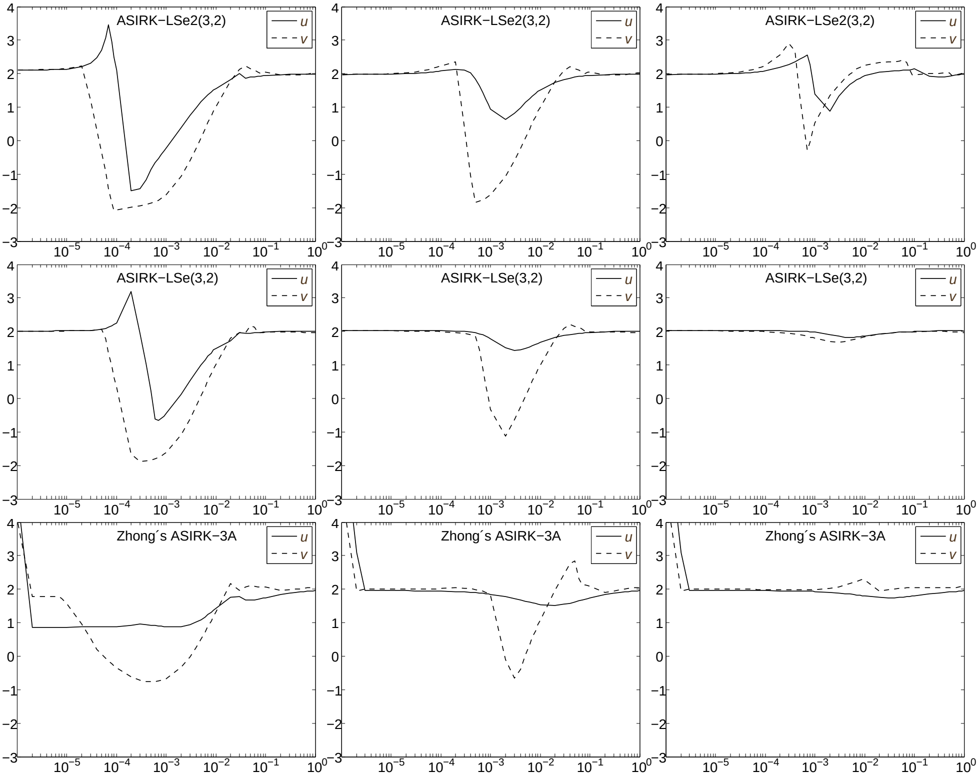

Figure 1: Convergence rates of u (continuous line) and v (dashed line) vs. stiffness parameter ε for problem (26) for stepsize $h = 0.05$.
From top to bottom: ASIRK-LSe2(3,2) (25), ASIRK-LSe(3,2) and Zhong's method. From left to right: $\mathrm{NC}_{\mathrm{InVal}}$, $\mathrm{C}_{\mathrm{InVal}}$ and $\mathrm{WP}_{\mathrm{InVal}}$ initial values (27).

In figure 1 we show the convergence rates versus the stiff parameter ε for the system (26), when methods ASIRK-LSe2(3,2), ASIRK-LSe(3,2) and Zhong's method are used.

For $\mathrm{C}_{\mathrm{InVal}}$ and $\mathrm{NC}_{\mathrm{InVal}}$ the new method ASIRK-LSe2(3,2) gives results similar to the ones obtained with ASIRK-LSe(3,2) scheme. However, for $\mathrm{WP}_{\mathrm{InVal}}$ initial conditions, the new method in this paper has a worse behavior in the middle stiff regime, that is, when h is close to ε. These poor results may be due to the higher value of the leading truncation error terms. Consequently, the choice of $\lambda = \omega_2$ does not lead to a numerical scheme better than the ones obtained in [10].

Acknowledgements

Supported by Ministerio de Economía y Competitividad, project MTM2011-23203.

References

[1] Ascher, U., Ruuth, S., , and Spiteri, R. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Appl. Numer. Math. 25* (1997), 151–167.

[2] Boscarino, S., , and Russo, G. On a class of uniformly accurate IMEX Runge-Kutta schemes and applications to hyperbolic systems with relaxation. *SIAM Journal on Scientific Computing 31* (2009), 1926–1945.

[3] Boscarino, S. Error analysis of IMEX Runge-Kutta methods derived from differential-algebraic systems. *SIAM Journal on Numerical Analysis 15* (2007), 1600–1621.

[4] Boscarino, S. On an accurate third order implicit-explicit Runge-Kutta method for stiff problems. *Applied Numerical Mathematics 59* (2009), 1515–1528.

[5] Caflisch, R. E., Jin, S., and Russo, G. Uniformly accurate schemes for hyperbolic systems with relaxation. *SIAM Journal on Numerical Analysis 34* (1997), 246–281.

[6] Calvo, M., Franco, J., , and Rández, L. Minimum storage Runge-Kutta schemes for computational acoustics. *Computers & Mathematics with Applications 45* (2003), 535–545.

[7] Carpenter, M. H., and Kennedy, C. A. Fourth-order 2n-storage Runge-Kutta schemes. *NASA Technical Memorandum 109112* (1994).

[8] Gottlieb, S., and Shu, C. Total variation diminishing Runge-Kutta schemes. *Math. Comp. 67* (1998), 73–85.

[9] Happenhofer, N., Koch, O., and Kupka, F. IMEX Methods for the ANTARES Code. *ASC Report 27* (2011).

[10] Higueras, I., and Roldán, T. Construction of robust implicit-explicit Runge-Kutta methods with low-storage requirements. *Submitted to Journal of Scientific Computing* (2014).

[11] Kennedy, C. A., and Carpenter, M. H. Additive runge-kutta schemes for convection-diffusion-reaction equations. *Applied numerical mathematics 44* (2003), 139–181.

[12] Kennedy, C. A., Carpenter, M. H., and Lewis, R. M. Low-storage, explicit Runge-Kutta schemes for the compressible Navier-Stokes equations. *Applied numerical mathematics 35* (2000), 177–219.

[13] Ketcheson, D. I. Highly efficient strong stability-preserving Runge-Kutta methods with low-storage implementations. *SIAM Journal on Scientific Computing 30* (2008), 2113–2136.

[14] Koch, O., Kupka, F., Löw-Baselli, B., Mayrhofer, A., and Zaussinger, F. SDIRK Methods for the ANTARES Code. *ASC Report 32* (2010).

[15] Pareschi, L., and Russo, G. Implicit-explicit Runge-Kutta schemes for stiff systems of differential equations. *Recent Trends in Numerical Analysis 3* (2000), 269–289.

[16] Pareschi, L., and Russo, G. High order asymptotically strong-stability-preserving methods for hyperbolic systems with stiff relaxation. *Hyperbolic Problems: Theory, Numerics, Applications* (2003), 241–251.

[17] Van Der Houwen, P. J. *Construction of integration formulas for initial value problems*. North Holland, 1977.

[18] Williamson, J. Low-storage Runge-Kutta Schemes. *Journal of Computational Physics 35* (1980), 48–56.

[19] Zhong, X. Additive semi-implicit Runge-Kutta Methods for computing high-speed nonequilibrium reactive flows. *Journal of Computational Physics 128* (1996), 19–31.

Inmaculada Higueras and Teo Roldán
Dpto. Ing. Matemática e Informática
Universidad Pública de Navarra
higueras@unavarra.es and teo@unavarra.es