Comparison of efficiency among different techniques to avoid order reduction with Strang splitting

I. Alonso-Mallo[a], B. Cano[a], N. Reguera[b,*]

[a]*IMUVA, Departamento de Matemática Aplicada, Universidad de Valladolid, Facultad de Ciencias, Paseo de Belén 7, 47011 Valladolid, Spain.*
[b]*IMUVA, Departamento de Matemáticas y Computación, Escuela Politécnica Superior, Universidad de Burgos, Avda. Cantabria, 09006 Burgos, Spain*

Abstract

In this paper, we offer a comparison in terms of computational efficiency between two techniques to avoid order reduction when using Strang method to integrate nonlinear initial boundary value problems with time-dependent boundary conditions. We see that it is important to consider an exponential method for the integration of the linear nonhomogeneous and stiff part in the technique by Einkemmer et al. so that the latter is comparable in efficiency with that suggested by Alonso et al. Some other advantages of the technique suggested by Alonso et al. are stated in the conclusions.

Keywords: Strang splitting, avoiding order reduction, computational comparison

1. Introduction

The order reduction phenomenon which appears when time evolution problems are approximated by means of the method of lines is well known since many years. We can mention the seminal papers [8, 25], where the cases of implicit rational and explicit Runge-Kutta methods are studied. Sharp bounds for the order observed when implicit Runge-Kutta methods are used for the time integration are obtained in [7, 23].

There are different techniques to avoid the order reduction when Runge-Kutta methods (or other methods with internal stages) are used for the time integration of linear initial boundary value problems. The idea in [10] is to subtract to the solution a function in order to obtain a problem with no order reduction. On the other hand, it is also possible to avoid the order reduction of such problems by acting on the boundary values of the internal stages [1, 2, 24]. The use of appropriate boundary conditions for the split subproblems of non

*Corresponding author
Email addresses: isaias@mac.uva.es (I. Alonso-Mallo), bego@mac.uva.es (B. Cano), nreguera@ubu.es (N. Reguera)

exponential additive splitting methods for linear and even nonlinear hyperbolic problems has also been previously considered in the literature [13, 20, 21], where stability restrictions on the time stepsize with respect to the space grid were necessary.

Obviously, the performance of different methods to avoid the order reduction phenomenon can only be compared for the same underlying time integrator and the same kind of evolution problem. Our goal in this paper is to compare techniques to avoid the order reduction in time which turns up when integrating with Strang method nonlinear problems of the form

$$\begin{aligned} u'(t) &= Au(t) + f(t, u(t)), \quad 0 \le t \le T, \\ \partial u(t) &= g(t), \\ u(0) &= u_0, \end{aligned} \tag{1}$$

where A is an elliptic differential operator, f is a smooth real function which acts as a reaction term, ∂ is a boundary operator, g is the boundary condition which is time-dependent and u_0 is a smooth initial condition. We assume that the solution of (1) is regular enough. As it is well known, Strang is a splitting method with no stability restrictions if the linear and stiff subproblem is assumed to be integrated 'exactly', for which order 1 is generally observed when integrating (1) in the standard way, even when $f = f(t)$ [18].

There are several papers in the literature concerning how to conserve the second-order of this method. More particularly, in [15, 16] a technique is suggested to do it, in which each part of the splitting is assumed to be solved in an exact way for the analysis and, in the numerical experiments, standard subroutines are used to integrate each part in space and time. In [15], in a similar way to [10], the idea is to consider $\widetilde{u} = u - z$, where z is the solution of this elliptic problem at each time $t \in [0, T]$,

$$Az(t) = 0, \quad \partial z(t) = g(t). \tag{2}$$

Then, $\widetilde{u}$ satisfies an evolution problem with homogeneous boundary conditions. More precisely,

$$\begin{aligned} \widetilde{u}'(t) &= A\widetilde{u}(t) + f(t, \widetilde{u}(t) + z(t)) - z'(t), \quad 0 \le t \le T, \\ \partial\widetilde{u}(t) &= 0, \\ \widetilde{u}(0) &= u_0 - z(0), \end{aligned}$$

As it is explained in [15], splitting this problem in a proper way and applying Strang method to it leads to avoid order reduction. However, this technique has the same drawbacks than that in [10], which is the reason why [10] and [1, 24] for Runge-Kutta methods have never been explicitly compared in the literature. Although for one-dimensional problems, it is obvious to find analytically the solution of (2) for every value $t \in [0, T]$, in more dimensions numerical techniques should be applied, in principle for continuous time and having to calculate also a good approximation for $z'(t)$. This makes the technique complicated and

expensive. Nevertheless, that is successfully avoided in [16] by considering just a function q which coincides at the boundary with $f(g)$ but which does not need to satisfy any elliptic problem and for which no q' is needed. Then, splitting the problem as described in the preliminaries of this paper, the order reduction can be avoided, as it is justified in [16] with a similar analysis to that in [15]. The calculation of q can also be done analytically in one dimension and simple domains in two dimensions (as it is explained in Section 3 of this paper); and even numerically in a cheap way in more complicated domains, according to [14, 16], although it is not in fact applied to such problems there. In this paper, we will concentrate on the technique in [16] for 1-dimensional and 2-dimensional simple domains. Therefore, the cost of calculation of q is not included.

On the other hand, in [5], another technique is suggested in which appropriate boundary conditions are suggested for each part of the splitting. The analysis there considers both the space and time discretization. The linear and stiff part is integrated 'exactly' in time through exponential-type functions while the nonlinear but smooth part is assumed to be numerically integrated by a classical integrator just with the order of accuracy that the user wants to achieve with the whole method. Although the latter seems to be the most natural, in order to ensure that the errors coming from the approximate integration of the split subproblems are not bigger than the error of the splitting itself, we will first use standard subroutines which use variable stepsizes with given small tolerances for the nonlinear and smooth subproblems of both techniques and the linear nonhomogeneous differential problem in [16]. However, as using a standard subroutine with variable stepsizes is not the most efficient way for solving the linear nonhomogeneous partial differential subproblems in [16], we do not make an efficiency comparison between both techniques with this implementation, but we will make the comparison by just integrating the resulting nonlinear subproblems with one or few steps of an explicit Runge-Kutta integrator, so that the error is negligible when compared with that of the splitting error itself and, in the case of the linear problems in [16], by considering also one or few steps of exponential quadrature rules of the required accuracy.

We will concentrate on the extensively used second-order Strang splitting and the aim of the paper is to compare both techniques ([5] and [16]) in terms of computational efficiency and propose an efficient implementation of the technique in [16] so that it is comparable, in terms of computational efficiency, to the techniques in [5]. This efficient implementation of the technique in [16] had not been proposed before in the literature and in fact, in [15, 16] general subroutines were suggested (look at [6] for an efficiency comparison with general subroutines). For the space discretization, we center on finite differences for the sake of brevity and because, in such a case, sparse matrices turn up to which FFT techniques or Krylov-type methods can be applied for the calculation of the exponential-type matrices. (For a comparison with collocation spectral methods for which the matrices are dense, look also at [6].) For the numerical experiments, both one-dimensional and bidimensional problems will be considered. There is already another comparison in the literature between both techniques [17] but there they just compare in terms of error against the time stepsize

without entering into the details of implementation and its computational cost, which we believe that is the interesting comparison. Moreover, they just consider time-independent boundary conditions and 1-dimensional problems, for which many simplifications can be made.

The paper is structured as follows. Section 2 gives some preliminaries on the description of the different techniques and suggest different implementations for each of them. In Section 3 we use standard subroutines in time to solve accurately each of the split problems. We do not expect this to be the best implementation neither of the technique in [5], nor of the one in [16], and probably Einkemmer et al. did not either, but we consider this first implementation since no other explicit technique is detailed in [15, 16] to do it efficiently and because, in this manner, we can see just the error coming from the splitting itself in the proper way. Section 4 offers the efficiency comparison when using in principle just one step of a second-order explicit Runge-Kutta method for the integration of the nonlinear problems and the exponential midpoint rule without order reduction [11], i.e. with order 2, for the linear non-homogeneous time-dependent boundary value problems appearing in [16]. In such a way, the techniques which are used for the approximation of the splitting subproblems in both [16] and [5] are similar. In both Sections 3 and 4, numerical differentiation is also considered in order to achieve local order near 3 instead of just 2, and again the computational comparison is performed. The conclusions of this work are sketched in Section 5. Finally, in an appendix, a thorough error analysis (including numerical differentiation) is given for one of the implementations of the base technique which was suggested in [5], but which modifications were not included in the analysis.

2. Preliminaries and suggestion of different implementations

The technique which is suggested in [16] consists of the following: A function $q(t)$ is constructed which satisfies $\partial q(t) = \partial f(t, u(t))$. Then, given the numerical approximation at the previous step u_n, the numerical approximation at the next step u_{n+1} is given by the following procedure:

$$
\begin{aligned}
&\left\{\begin{array}{rcl} v'_{n,1}(t) &=& Av_{n,1}(t) + q(t), \\ v_{n,1}(t_n) &=& u_n, \\ \partial v_{n,1}(t) &=& g(t), \end{array}\right. \\
&\left\{\begin{array}{rcl} w'_n(t) &=& f(t, w_n(t)) - q(t), \\ w_n(t_n) &=& v_{n,1}(t_n + \frac{k}{2}), \end{array}\right. \\
&\left\{\begin{array}{rcl} v'_{n,2}(t) &=& Av_{n,2}(t) + q(t), \\ v_{n,2}(t_n + \frac{k}{2}) &=& w_n(t_n + k), \\ \partial v_{n,2}(t) &=& g(t), \end{array}\right. \\
&u_{n+1} = v_{n,2}(t_n + k).
\end{aligned} \tag{3}
$$

However, we notice that two of three problems which turn up here are stiff and therefore solving them will be more expensive than solving the unique nonlinear

but smooth problem. In order to reverse that, the decomposition of the splitting method can be done in another order and then the following procedure would turn up for which, with similar arguments, no order reduction would either turn up:

$$\begin{aligned}
&\left\{\begin{array}{rcl} w'_{n,1}(t) &=& f(t, w_{n,1}(t)) - q(t), \\ w_{n,1}(t_n) &=& u_n, \end{array}\right. \\
&\left\{\begin{array}{rcl} v'_n(t) &=& Av_n(t) + q(t), \\ v_n(t_n) &=& w_{n,1}(t_n + \frac{k}{2}), \\ \partial v_n(t) &=& g(t) \end{array}\right. \\
&\left\{\begin{array}{rcl} w'_{n,2}(t) &=& f(t, w_{n,2}(t)) - q(t), \\ w_{n,2}(t_n + \frac{k}{2}) &=& v_n(k), \end{array}\right. \\
&u_{n+1} = w_{n,2}(t_n + k).
\end{aligned} \tag{4}$$

Then, two of the problems are cheap and just one is more expensive.

On the other hand, in [5], the main idea is to first consider the problem

$$\left\{\begin{array}{rcl} v'_n(s) &=& f(t_n + s, v_n(s)), \\ v_n(0) &=& u_n, \end{array}\right.$$

which is numerically integrated with order 2 from $s = 0$ to $s = k/2$ to obtain $\Psi_{\frac{k}{2}}^{f,t_n}(u_n) \approx v_n(k/2)$. The second problem considered is

$$\left\{\begin{array}{rcl} w'_n(s) &=& Aw_n(s), \\ w_n(0) &=& \Psi_{\frac{k}{2}}^{f,t_n}(u_n), \\ \partial w_n(s) &=& \partial[u(t_n) + \frac{k}{2} f(t_n, u(t_n)) + sAu(t_n)], \end{array}\right. \tag{5}$$

and the third problem is

$$\left\{\begin{array}{rcl} z'_n(s) &=& f(t_n + \frac{k}{2} + s, z_n(s)), \\ z_n(0) &=& w_n(k), \end{array}\right.$$

which is also numerically integrated with order 2 to obtain $\Psi_{\frac{k}{2}}^{f,t_n+\frac{k}{2}}(w_n(k)) \approx z_n(k/2)$.

Then, the final equation is

$$u_{n+1} = \Psi_{\frac{k}{2}}^{f,t_n+\frac{k}{2}}(w_n(k)).$$

Moreover, the procedure to integrate (5) is more explicitly stated. Firstly, in [5] (see also [4, 3, 12]), a general space discretization is introduced which discretizes the elliptic problem

$$Au = F, \quad \partial u = g,$$

through the 'elliptic projection' $R_h u$ which satisfies

$$A_{h,0} R_h u + C_h g = P_h F,$$

for a certain matrix $A_{h,0}$, an associated boundary operator C_h and a projection operator P_h. Then, given the numerical approximation at the previous step U_h^n, the procedure in [5] to obtain U_h^{n+1} reads as follows:

$$\begin{aligned} V_h^n &= \Psi_{\frac{k}{2}}^{f,t_n}(U_h^n), \\ W_{h,n}(k) &= e^{kA_{h,0}}V_h^n + k\varphi_1(kA_{h,0})C_h[g(t_n) + \frac{k}{2}\partial f(t_n, u(t_n))] \\ &\quad + k^2\varphi_2(kA_{h,0})C_h[g'(t_n) - \partial f(t_n, u(t_n)] \\ U_h^{n+1} &= \Psi_{\frac{k}{2}}^{f,t_n+\frac{k}{2}}(W_{h,n}(k)), \end{aligned} \tag{6}$$

where φ_1 and φ_2 are the standard functions which are used in exponential methods [5] and which are defined by

$$\varphi_j(tA_{h,0}) = \frac{1}{t^j}\int_0^t e^{(t-\tau)A_{h,0}}\frac{\tau^{j-1}}{(j-1)!}d\tau, \quad j \geq 1. \tag{7}$$

It is well-known that they can be calculated in a recursive way through the formulas

$$\varphi_{j+1}(z) = \frac{\varphi_j(z) - 1/j!}{z}, \quad z \neq 0, \qquad \varphi_{j+1}(0) = \frac{1}{(j+1)!}, \qquad \varphi_0(z) = e^z. \tag{8}$$

The original suggestion used this order for the decomposition thinking that Ψ_k is just an explicit method which is applied with a single step of size k, and therefore it would be cheaper than the equation in $W_{h,n}(k)$. We still believe that would be the best. However, in the first part of this paper, in order to ensure that errors arise from the splitting itself and not from the approximate integration of the split subproblems, we will solve that part with a standard variable stepsize subroutine for non-stiff problems until a given small tolerance. In such a way, the first and last problem may be more expensive than the middle one. Therefore, we will also consider this other implementation which comes from reversing the order of the problems in the decomposition (see the appendix):

$$\begin{aligned} W_{h,n}(\frac{k}{2}) &= e^{\frac{k}{2}A_{h,0}}U_h^n + \frac{k}{2}\varphi_1(\frac{k}{2}A_{h,0})C_h g(t_n) + \frac{k^2}{4}\varphi_2(\frac{k}{2}A_{h,0})C_h\partial Au(t_n) \\ V_h^n &= \Psi_k^{f,t_n}(W_{h,n}(\frac{k}{2})), \\ U_h^{n+1} &= e^{\frac{k}{2}A_{h,0}}V_h^n + \frac{k}{2}\varphi_1(\frac{k}{2}A_{h,0})C_h\partial[u(t_n) + \frac{k}{2}Au(t_n) + kf(t_n, u(t_n)] \\ &\quad + \frac{k^2}{4}\varphi_2(\frac{k}{2}A_{h,0})C_h\partial Au(t_n). \end{aligned} \tag{9}$$

In the following, we will denote by EO1 to (4), by EO2 to (3), by ACR1 to (6) and by ACR2 to (9).

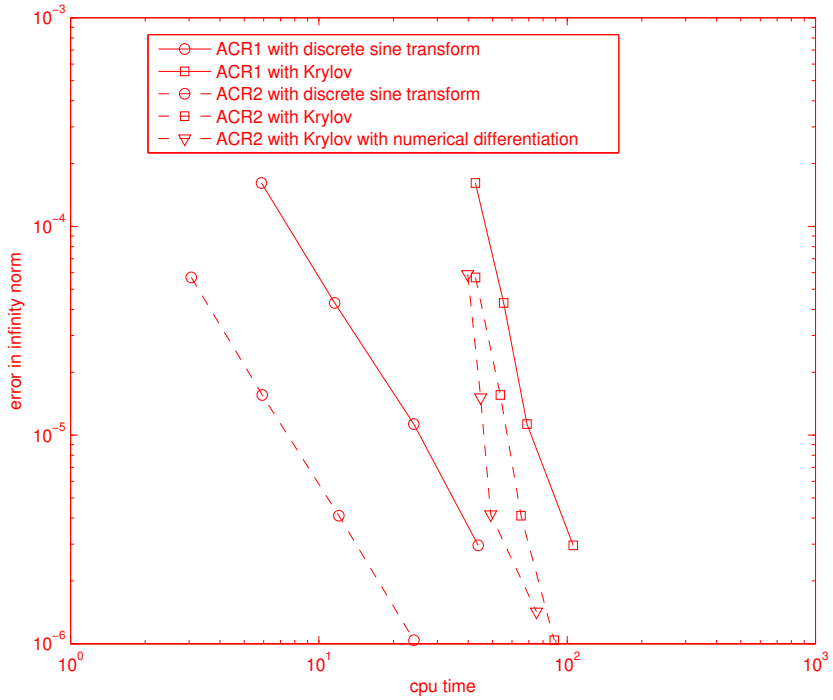

Figure 1: Numerical comparison with exact time integration of split subproblems for the 1-dimensional problem (10)

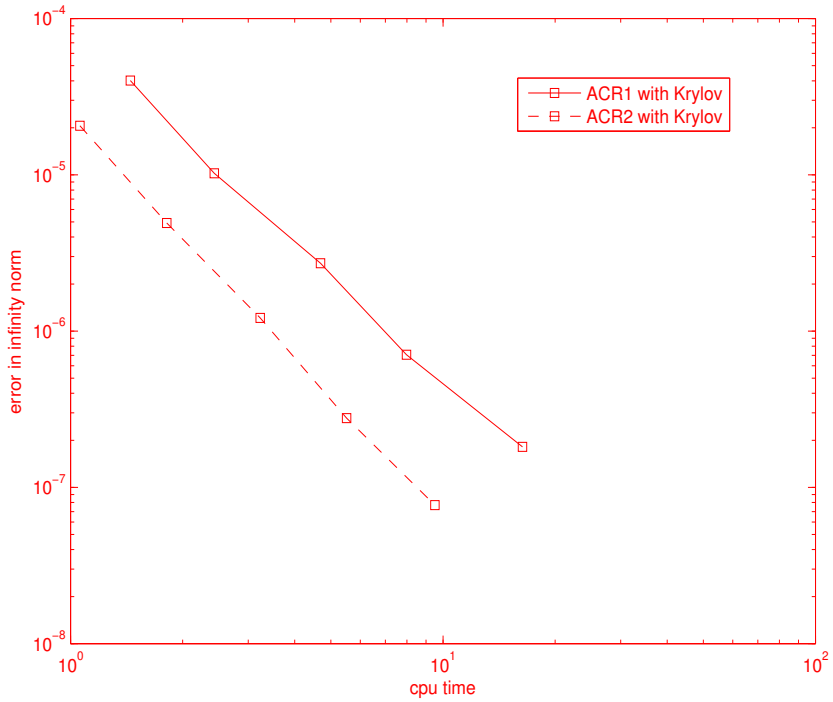

Figure 2: Numerical comparison with exact time integration of split subproblems for the 2-dimensional problem (13)

3. Numerical comparison with exact time integration of the split subproblems

As Strang method just has second-order accuracy, it is usually used for problems in which a very high precision is not required. Because of that, in space we have considered finite differences of just second order accuracy in the space grid. More particularly, as in our examples we take the operator A as the Laplacian, we use the standard symmetric second-order difference scheme in 1 dimension and the five-point formula in 2 dimensions [26]. We have considered as space grid $h = 5 \times 10^{-4}$ for the 1-dimensional case and $h = 2 \times 10^{-2}$ for the 2-dimensional case. With this type of implementation, the matrix $A_{h,0}$ is sparse and, in this particular case, their eigenvalues and eigenvectors are well-known [19]. Because of the sparsity, it is natural to use standard Krylov subroutines [22] in ACR1 and ACR2 to calculate the application of exponential-type functions over vectors. When the eigenvalues and eigenvectors are known, which is specific of this particular example and space discretization, in order to calculate the same terms, it seems advantageous to use the discrete sine transform in the same way that FFT is used in Poisson solvers [19]. In this section, for the one-dimensional case, we show the results with both Krylov and discrete sine transform. For the sake of brevity and because it is more general, on the bidimensional case, we have centered on Krylov subroutines. When using Krylov subroutines [22], we have considered the default tolerance 10^{-7}. This value is enough since, when using it, the dominant error is that of the splitting itself and the error of both Krylov subroutines and space discretization is negligible because the error diminishes with second order in the time-stepsize.

On the other hand, for the numerical integration of the nonlinear and smooth subproblems in ACR, we have used subroutine ode45 from MATLAB. Similarly to Krylov subroutines, we have considered 10^{-7} and 10^{-8} as relative and absolute tolerances respectively for this subroutine. These accuracies are suitable taking into account the errors of our numerical results.

In a first place, we have considered the following one-dimensional Dirichlet boundary value problem whose exact solution is $u(x,t) = e^{t+x^3}$:

$$\begin{aligned} &u_t(x,t) = u_{xx}(x,t) + u^2 - e^{t+x^3}(9x^4 + 6x + e^{t+x^3} - 1), \quad 0 \le x \le 1, \\ &u(x,0) = e^{x^3}, \\ &u(0,t) = e^t, \quad u(1,t) = e^{t+1}, \quad t \in [0, 0.2]. \end{aligned} \tag{10}$$

In Figure 1 we see the results for ACR1 and ACR2 with the different values of k which turn up in Table 1. We observe that ACR2 is more competitive since not only the computational time is smaller for a fixed value of k but also the error is smaller. In this particular case, considering discrete sine transforms is much cheaper than using Krylov techniques. However, for a general operator A, that may not be possible and that is why it is also interesting to see the comparison when using these techniques. Besides, in Table 1, the avoidance of order reduction with all ACR1, ACR2, can be clearly checked.

Notice that we do not compare the results in terms of computational efficiency shown in Figure 1 for ACR1 and ACR2 with those obtained with EO techniques using also general variable stepsize subroutines. This is due to the fact that using such subroutines to integrate the linear, nonhomogeneous partial differential subproblems of EO1 and EO2 would not be the best option [6], being a much more efficient implementation of EO techniques the one that we propose in Section 4. However, we show here the results in Table 2 for EO1 and EO2 (using subroutine ode45 for the nonlinear and smooth subproblems and subroutine ode15s for the linear, nonhomogeneous partial differential subproblems) which show also that order reduction is avoided. Notice that, as it is a one-dimensional problem, the function q in EO1 and EO2 is calculated directly for every value of t as the straight line which joins the corresponding values $f(t, 0, e^t)$ and $f(t, 1, e^{t+1})$ at $x = 0$ and $x = 1$ respectively. We notice that, for a fixed value of k, EO2 leads to smaller errors than EO1, in the same way that happened with ACR2 with respect to ACR1.

k	10^{-3}	5×10^{-4}	2.5×10^{-4}	1.25×10^{-4}
ACR1 Global error	1.61×10^{-4}	4.30×10^{-5}	1.13×10^{-5}	2.96×10^{-6}
ACR1 Order		1.9	1.9	1.9
ACR2 Global error	5.70×10^{-5}	1.56×10^{-5}	4.11×10^{-6}	1.04×10^{-6}
ACR2 Order		1.9	1.9	2.0

Table 1: Global error and order for ACR1 and ACR2 with discrete sine transform or with Krylov for the 1-dimensional problem, for the values of k which have been used in Figure 1

k	8×10^{-3}	4×10^{-3}	2×10^{-3}
EO1 Global error	5.96×10^{-3}	1.72×10^{-3}	4.80×10^{-4}
EO1 Order		1.8	1.8
EO2 Global error	2.5×10^{-3}	7.61×10^{-4}	2.20×10^{-4}
EO2 Order		1.7	1.8
EO2 with num. dif. Local error	5.8×10^{-4}	9.17×10^{-5}	1.43×10^{-5}
EO2 with num. dif. Local Order		2.7	2.7
EO2 with num. dif. Global error	5.56×10^{-3}	1.37×10^{-3}	3.15×10^{-4}
EO2 with num. dif. Global Order		2.0	2.1

Table 2: Global error and order for EO1 and EO2

Moreover, following [17], we have also considered numerical differentiation in order to try to get local order 3 with EO1 and EO2 in (10). More precisely, theoretically, a function q should be taken for which $\partial q(t) = \partial f(t, u(t))$ and $\partial Aq(t) = \partial Af(t, u(t))$. Although, even when that function can be constructed, the order for the global error does not improve, it is interesting to see whether the fact that the local errors maybe smaller implies a better overall behaviour.

Notice that, in (10),

$$\frac{d}{dx^2} f = f_{xx} + 2f_{x,u}u_x + f_{uu}u_x^2 + f_u u_{xx}. \tag{11}$$

As $\partial u_{xx} = g'(t) - \partial f(t,u)$, numerical differentiation is just required to calculate ∂u_x. For that, we have considered the second-order scheme

$$\begin{aligned} u_x(0,t) &\approx \frac{-\frac{3}{2}u(0,t) + 2u(h,t) - \frac{1}{2}u(2h,t)}{h}, \\ u_x(1,t) &\approx \frac{\frac{3}{2}u(1,t) - 2u(1-h,t) + \frac{1}{2}u(1-2h,t)}{h}. \end{aligned}$$

From a theoretical point of view, to achieve local order 3, at each step we would need these derivatives at any continuous time $t \in [t_n, t_{n+1})$. However, we just have approximations for the interior values $u(h,t)$, $u(2h,t)$, $u(1-h,t)$, $u(1-2h,t)$ at time t_n. Because of this, in formula (11), we have evaluated all terms at continuous t except for the term u_x, which is just approximated at $t = t_n$. In such a way, the local error shows order a bit less than 3 but higher than 2.5, as it can be observed in Table 2. Although for the values of k considered in Table 2 the global errors are not smaller when using numerical differentiation, for smaller values of k the results when using numerical differentiation are slightly better than when not using it. Therefore, the conclusion we reach is that using numerical differentiation is, at best, slightly worth doing. This conclusion will be corroborated when in Section 4 we propose an efficient implementation for EO technique. We also notice that, although numerical differentiation is unstable, its effect is still not visible with the considered value of h for the first derivative and the range of errors which we are considering.

As for ACR2, considering also terms of second order in s for the boundaries of the problems in which the operator A appears, the following full scheme turns

up (see the appendix):

$$\begin{aligned}
W_{h,n}(\frac{k}{2}) &= e^{\frac{k}{2}A_{h,0}}U_h^n + \frac{k}{2}\varphi_1(\frac{k}{2}A_{h,0})C_h g(t_n) + \frac{k^2}{4}\varphi_2(\frac{k}{2}A_{h,0})C_h\partial A u(t_n) \\
&\quad + \frac{k^3}{8}\varphi_3(\frac{k}{2}A_{h,0})C_h\partial A^2 u(t_n) \\
V_h^n &= \Psi_k^{f,t_n}(W_{h,n}(\frac{k}{2})), \\
U_h^{n+1} &= e^{\frac{k}{2}A_{h,0}}V_h^n \\
&\quad + \frac{k}{2}\varphi_1(\frac{k}{2}A_{h,0})C_h\partial[u(t_n) + k(\frac{1}{2}Au(t_n) + f(t_n,u(t_n)) \\
&\qquad + k^2(\frac{1}{8}A^2u(t_n) + \frac{1}{2}f_u(t_n,u(t_n))Au(t_n) \\
&\qquad + \frac{1}{2}(f_t(t_n,u(t_n)) + f_u(t_n,u(t_n))f(t_n,u(t_n)))] \\
&\quad + \frac{k^2}{4}\varphi_2(\frac{k}{2}A_{h,0})C_h\partial[Au(t_n) + \frac{k}{2}A^2u(t_n) + kAf(t_n,u(t_n))] \\
&\quad + \frac{k^3}{8}\varphi_3(\frac{k}{2}A_{h,0})C_h\partial A^2 u(t_n). \qquad (12)
\end{aligned}$$

As $\partial A^2 u = \partial A\dot{u} - \partial Af = \ddot{g} - \partial(f_t + f_u\dot{u}) - \partial Af$, what is again necessary is to approximate u_x with numerical differentiation, but just at t_n, and we have done it in the same way as before. Figure 1 also shows the results with numerical differentiation for $k = 10^{-3}, 5\times10^{-4}, 2.5\times10^{-4}, 1.25\times10^{-4}$ and it is observed that *there is a small improvement in efficiency when using numerical differentiation with ACR2 although it is not extremely significant.*

Notice that for ACR2, for a fixed value of k, the computational cost does not increase but is slightly smaller when using numerical differentiation. This must be due to the fact that the standard subroutines which are used converge more quickly when numerical differentiation is applied. A full explanation for that is out of the scope of this paper although it might be a subject of future research.

Let us now see what happens with a bidimensional problem. We have considered

$$\begin{aligned}
&u_t(x,y,t) = u_{xx}(x,y,t) + u_{yy}(x,y,t) + f(t,x,y,u(x,y,t)), \quad 0 \le x,y \le 1, \\
&u(x,y,0) = x^2 + y^2, \\
&u(0,y,t) = e^t y^2, \quad u(1,y,t) = e^t(1+y^2), \\
&u(x,0,t) = e^t x^2, \quad u(x,1,t) = e^t(1+x^2), \quad t \in [0,0.2], \qquad (13)
\end{aligned}$$

where $f(t,x,y,u) = u^2 - e^{2t}(x^2+y^2)^2 + e^t(x^2+y^2-4)$, so that the exact solution is $u(x,y,t) = e^t(x^2+y^2)$.

We have implemented ACR1 and ACR2 again with Krylov subroutines [22]. In Figure 2 we have displayed the results corresponding to ACR1, ACR2 for the values of k in Table 3. We can see that again, in this problem, the second

implementation is the most efficient for ACR. Besides, the approximate second order of all implementations is also clearly checked.

When implementing EO1 and EO2, the only remarkable difference with respect to the one-dimensional case is that the function $q(t,x,y)$ must be chosen in a different way. We consider a function of the form

$$q(t,x,y) = r(t,x)f(t,1,y,e^{t+1+y^3}) + s(t,x)f(t,0,y,e^{t+y^3})$$

which satisfies the corresponding conditions at the boundary. That is achieved if $r(t,x)$ and $s(t,x)$ satisfy

$$\begin{pmatrix} f(t,1,0,e^t) & f(t,0,0,0) \\ f(t,1,1,2e^{t+2}) & f(t,0,1,e^t) \end{pmatrix} \begin{pmatrix} r(t,x) \\ s(t,x) \end{pmatrix} = \begin{pmatrix} f(t,x,0,x^2e^t) \\ f(t,x,1,(x^2+1)e^t) \end{pmatrix}.$$

(Notice that this technique to calculate q analytically can be applied in a rectangular domain but not in more complicated domains in two dimensions.) The results in Table 4 show that the order reduction is avoided.

k	5×10^{-3}	2.5×10^{-3}	1.25×10^{-3}	6.25×10^{-4}	3.125×10^{-4}
ACR1 Global error	4.02×10^{-5}	1.02×10^{-5}	2.72×10^{-6}	7.06×10^{-7}	1.82×10^{-7}
ACR1 Order		2.0	1.9	2.0	2.0
ACR2 Global error	2.06×10^{-5}	4.92×10^{-6}	1.22×10^{-6}	2.78×10^{-7}	7.71×10^{-8}
ACR2 Order		2.1	2.0	2.1	1.9

Table 3: Global error and order for ACR1 and ACR2 with Krylov for the 2-dimensional problem, for the values of k which have been used in Figure 2

k	4×10^{-2}	2×10^{-2}	10^{-2}
EO1 Global error	2.31×10^{-3}	5.64×10^{-4}	1.39×10^{-4}
EO1 Order		2.0	2.0
EO2 Global error	2.37×10^{-5}	7.01×10^{-6}	2.34×10^{-6}
EO2 Order		1.8	1.6

Table 4: Global error and order for EO1 and EO2 with ode15s for the 2-dimensional problem

Considering numerical differentiation in two dimensions is also possible but we would like to remark that, with EO techniques, that it is not as plausible as in one dimension since, apart from approximating numerically $\partial A f(t,u(t))$ at each step and calculating a function $\tilde{q}(t)$ which coincides with it at the boundary, a function $q(t)$ must be calculated such that

$$\begin{aligned} Aq(t) &= \tilde{q}(t), \\ \partial q(t) &= \partial f(t,u(t)). \end{aligned} \tag{14}$$

In one dimension, this was achieved just by integrating twice the linear function $\tilde{q}(t)$ and that was done analytically for every value of $t \in [t_n, t_{n+1})$. However,

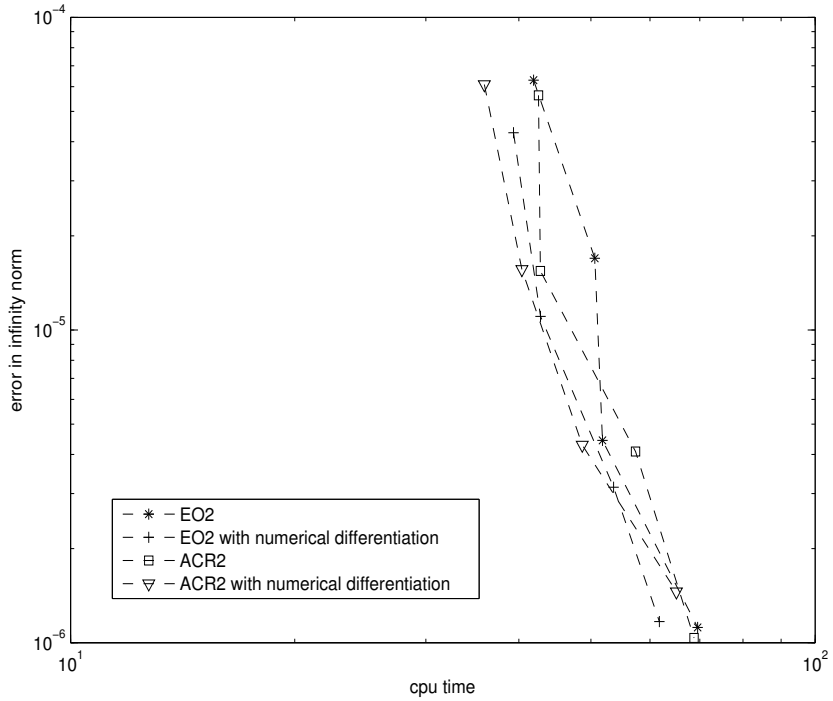

Figure 3: Numerical comparison with approximate time integration of split subproblems for the 1-dimensional problem (10)

in two dimensions, that cannot be done any more and the elliptic problems (14) should be numerically solved, not only for every value t_n, but even theoretically for every $t \in [t_n, t_{n+1})$. In contrast, notice that numerical differentiation with ACR (12) just requires approximating $\partial A f(t_n, u(t_n))$ at each step and no elliptic problem must be numerically solved. Besides, with respect to the same method but without numerical differentation, the additional cost mainly consists of just two more terms per step which contain $\varphi_3(\frac{k}{2}A_{h,0})$. In any case, we do not either include numerical differentiation with ACR here for the sake of clarity and brevity.

4. Numerical comparison with approximate time integration of the split subproblems

In this section we consider again both techniques, EO and ACR, using finite differences in space but, for the time integration of each of the subproblems of the decomposition, we take in principle a single step of a method of the required

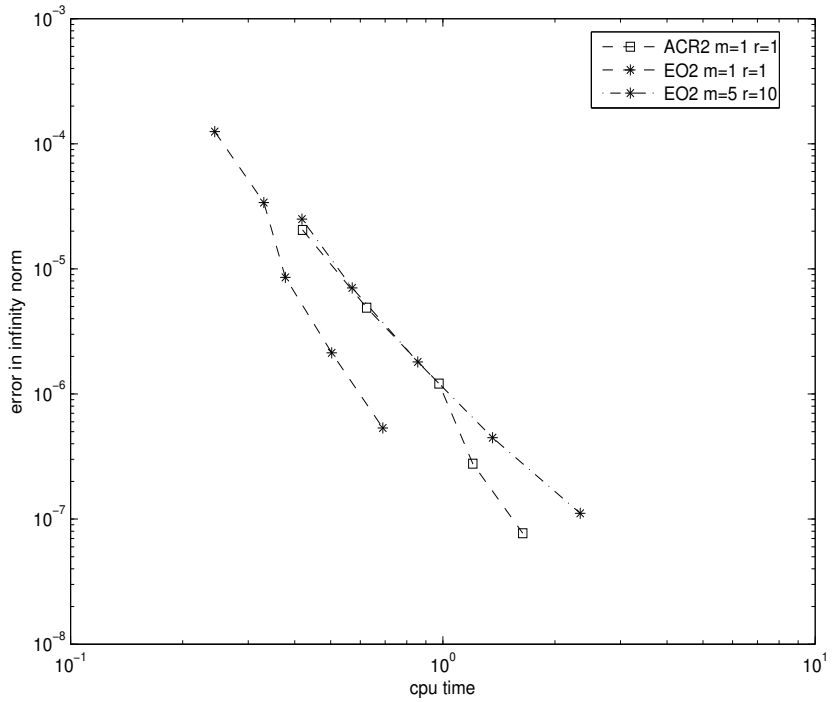

Figure 4: Numerical comparison with approximate time integration of split subproblems for the 2-dimensional problem (13)

final accuracy.

More precisely, for ACR, we do it similarly to what was done in [5]; for the time integration of the nonlinear part of the problem, we advance a single step with the explicit Runge-Kutta method of local order 3 which is given by the Butcher tableau

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ \hline & 0 & 1 \end{array}.$$

In such a way, the local error which is added to that of the splitting itself is an order higher if numerical differentiation is not applied and of the same order if the latter is used.

On the other hand, for EO, we also consider the same technique to integrate the nonlinear subproblem and, for the linear part, we advance a single step with an exponential quadrature rule of local order 3. In such a way, there is more similarity with the technique suggested in [5] for the linear part in the sense that exponential-type functions are finally used. The difference is that, with ACR, after performing a spatial discretization of problem (5), an inhomogeneous problem turns up with an inhomogeneity which consists of a linear expression in s due to the boundary. Because of that, the exact solution of problem (5) after spatial discretization can be calculated exactly in terms of functions φ_1 and φ_2. This is not the case with EO, since the boundary in the middle part of (4) is not even a polynomial and, therefore, the system which turns up after spatial discretization of that part can just be approximately integrated. The exponential quadrature rule which we have taken is the exponential midpoint rule without order reduction for time-dependent boundary conditions, which was proved to be very efficient and of local order 3 in [11]. It reads like this for the first and third subproblems in (3) when advancing a single step of size $\frac{k}{2}$ from U_h^n and W_h^n respectively, where W_h^n is the result of applying a single step

of size k of the previous Runge-Kutta method to the second subproblem in (3):

$$\begin{aligned}
V_{h,n,1} &= e^{\frac{k}{2}A_{h,0}}U_h^n + \frac{k}{2}\varphi_1(\frac{k}{2}A_{h,0})C_h g(t_n) + (\frac{k}{2})^2\varphi_2(\frac{k}{2}A_{h,0})C_h[\dot{g}(t_n) - \partial q(t_n)] \\
&\quad +(\frac{k}{2})^3\varphi_3(\frac{k}{2}A_{h,0})C_h[\ddot{g}(t_n) - \partial\dot{q}(t_n) - \partial Aq(t_n)] \\
&\quad +\frac{k}{2}\big[\varphi_1(\frac{k}{2}A_{h,0})q(t_n + \frac{k}{4}) + \frac{k}{2}\varphi_2(\frac{k}{2}A_{h,0})C_h\partial q(t_n + \frac{k}{4}) \\
&\qquad +(\frac{k}{2})^2\varphi_3(\frac{k}{2}A_{h,0})C_h\partial Aq(t_n + \frac{k}{4})\big]. \\
V_{h,n,2} &= e^{\frac{k}{2}A_{h,0}}W_h^n + \frac{k}{2}\varphi_1(\frac{k}{2}A_{h,0})C_h g(t_n + \frac{k}{2}) \\
&\quad +(\frac{k}{2})^2\varphi_2(\frac{k}{2}A_{h,0})C_h[\dot{g}(t_n + \frac{k}{2}) - \partial q(t_n + \frac{k}{2})] \\
&\quad +(\frac{k}{2})^3\varphi_3(\frac{k}{2}A_{h,0})C_h[\ddot{g}(t_n + \frac{k}{2}) - \partial\dot{q}(t_n + \frac{k}{2}) - \partial Aq(t_n + \frac{k}{2})] \\
&\quad +\frac{k}{2}\big[\varphi_1(\frac{k}{2}A_{h,0})q(t_n + \frac{3k}{4}) + \frac{k}{2}\varphi_2(\frac{k}{2}A_{h,0})C_h\partial q(t_n + \frac{3k}{4}) \\
&\qquad +(\frac{k}{2})^2\varphi_3(\frac{k}{2}A_{h,0})C_h\partial Aq(t_n + \frac{3k}{4})\big].
\end{aligned}$$

For the sake of brevity and clarity and because they have been seen to lead to smaller errors for a given stepsize in the previous sections for the problems considered, we have centered on EO2 and ACR2.

For the one-dimensional problem (10), without resorting to numerical differentiation and resorting to it, the numerical comparison in terms of efficiency is given in Figure 3. There, as in Figure 1, $h = 5 \times 10^{-4}$ has also been considered for the finite differences in space and $k = 10^{-3}, 5\times10^{-4}, 2.5\times10^{-4}, 1.25\times10^{-4}$, for the time integration. Moreover, with both techniques, Krylov subroutines have been used to calculate the multiplication by φ_j-functions with tolerance 10^{-7}. *The conclusion is that, with this implementation, both techniques are very similar in the comparison of efficiency. Besides, with numerical differentiation both techniques are slightly better than without resorting to it.* (We also remark here that, in a similar way to what happened in the previous section, with numerical differentiation, when $\partial u_x(t_n + sk)$ is needed for $s \neq 0$, we just approximate it at $s = 0$ because it is there where we have approximations of the interior values.) We also notice that it can be checked that, for given stepsizes, the errors which are committed with the implementations of the previous section are very similar. Therefore, we have managed that the errors which come from the approximation of the split problems are negligible compared to the errors of the splitting itself.

As for the bidimensional problem (13), the comparison is given in Figure 4 for $h = 2 \times 10^{-2}$ and $k = 4 \times 10^{-2}, 2 \times 10^{-2}, 10^{-2}, 5 \times 10^{-3}, 2.5 \times 10^{-3}$ for EO2 and $k = 5 \times 10^{-3}, 2.5 \times 10^{-3}, 1.25 \times 10^{-3}, 6.25 \times 10^{-4}, 3.125 \times 10^{-4}$ for ACR2. Now, numerical differentiation has not been considered for the same reasons which we stated at the end of Section 3. Although, for a fixed value of

k, the computational cost of EO2 is as bigger than ACR2, for this particular problem the error is quite smaller (even much smaller than with EO1) and, because of this, in this particular problem EO2 can be cheaper than ACR2 in a relation 5/3. However, comparing the size of the errors with those obtained with the implementation of the previous section, we see that, although for given stepsizes, the errors for ACR2 are more or less the same, the same does not happen with EO2. The error coming from the approximate integration of the split subproblems is more important than that of the splitting itself. Following something similar to what was already suggested in [15] for the nonlinear subproblems, we have integrated both the linear and nonlinear subproblems in (3) with the same integrators but, instead of using just one step, with r uniform steps for the nonlinear subproblems and m uniform steps for the linear ones. The minimum values of r and m that we have had to take so that the error is negligible compared with that of the splitting itself has been $r = 10$ and $m = 5$. In such a case, *EO2 is equally comparable in efficiency with ACR2*, as it happened with the 1-dimensional problem with the implementation that we propose in this section. In any case, we would like to remark here that the cost of calculating the function q with EO2 is not considered in this paper and that ACR does never need to calculate such a function q. Besides, although some techniques have been proposed to calculate q efficiently [14], they have not either been tested in complicated domains.

5. Conclusions

If suitable time integrators of IVP are chosen, then ACR and EO behave in a similar way from the computational point of view. In section 4, we consider a very recent technique [11] to approximate the solution of one of the split subproblems for EO, which improves the computational performance of previous EO proposals.

On the other hand, applying numerical differentiation to get local order 3 causes less problems with ACR than with EO. Moreover, ACR has the advantage that you do not have to worry about the calculation of q which cost, although small according to [14], has not even been included for the comparisons here.

Acknowledgements

This work has been supported by Ministerio de Ciencia e Innovación and Regional Development European Funds through project MTM 2015-66837-P and by Junta de Castilla y León and Feder through projects VA024P17, VA041P17 and VA105G18.

References

[1] I. Alonso-Mallo, *Runge–Kutta methods without order reduction for linear initial boundary value problems*, Numer Math **91** (2002), 577–603.

[2] I. Alonso-Mallo, B. Cano and J.C. Jorge *Spectral-fractional step Runge-Kutta discretizations for initial boundary value problems with time dependent boundary conditions*, Math Comp **73** (2004), 1801–1825.

[3] I. Alonso–Mallo, B. Cano and N. Reguera, *Avoiding order reduction when integrating linear initial boundary value problems with Lawson methods*, IMA J Numer Anal **37** (2017), 2091–2119.

[4] I. Alonso–Mallo, B. Cano and N. Reguera, *Avoiding order reduction when integrating linear initial boundary value problems with exponential splitting methods*, IMA J Num Anal **38** (2018), 1294–1323.

[5] I. Alonso-Mallo, B. Cano and N. Reguera, *Avoiding order reduction when integrating reaction-diffusion boundary value problems with exponential splitting methods*, J. Comp. Appl. Math. **357** (2019), pp. 228–250.

[6] I. Alonso-Mallo, B. Cano and N. Reguera, *Looking for efficiency when avoiding order reduction in nonlinear problems with Strang splitting*, arXiv:1709.09849.

[7] I. Alonso–Mallo and C. Palencia, *Optimal orders of convergence for Runge-Kutta methods and linear, initial boundary value problems*, Appl Numer Math **44** (2003), 1–19.

[8] P. Brenner, M. Crouzeix and V. Thomée, *Single step methods for inhomogeneous linear differential equations in Banach space*, R.A.I.R.O. Anal Numer, **16** (1982), 5–26.

[9] C. Bernardy and Y. Maday, *Approximations spectrales de problemes aux limites elliptiques*, Springer-Verlag France, Paris, 1992. MR 94f:65112.

[10] M. P. Calvo and C. Palencia, *Avoiding the order reduction of Runge–Kutta methods for linear initial boundary value problems*, Math Comp **71** (2002), 1529–1543.

[11] B. Cano and M. J. Moreta, *Exponential quadrature rules without order reduction*, SIAM J. Num. Anal. **56-3**(2018), 1187–1209.

[12] B. Cano and N. Reguera, *Avoiding order reduction when integrating reaction-diffusion boundary value problems with exponential splitting methods*, J Comp Appl Math **316** (2017), 86–99.

[13] J. Connors, J.W. Banks, J.A. Hittinger, and C.S. Woodward, *Quantification of errors for operator-split advection-diffusion calculations*, Comput Methods Appl Mech Engrg. **272** (2014), 181–197.

[14] L. Einkemmer, M. Moccaldi and A. Ostermann, *Efficient boundary corrected Strang splitting*, Appl Math Comput **332** (2018), 76–89.

[15] L. Einkemmer and A. Ostermann, *Overcoming order reduction in diffusion-reaction splitting. Part 1: Dirichlet boundary conditions*, SIAM J Sci Comput **37 (3)** (2015), A1577–A1592.

[16] L. Einkemmer and A. Ostermann, *Overcoming order reduction in diffusion-reaction splitting. Part 2: Oblique boundary conditions*, SIAM J Sci Comput **38** (2016), A3741–A3757.

[17] L. Einkemmer and A. Ostermann, *A comparison of boundary correction methods for Strang splitting*, Discrete and Continuous Dynamical Systems Series B, **23** (7) (2018), 2641–2660.

[18] E. Faou, A. Ostermann and K. Schratz, *Analysis of exponential splitting methods for inhomogeneous parabolic equations*, IMA J. Numer. Anal. **35** (1) (2015), 161–178.

[19] A. Iserles, *A First Course in the Numerical Analysis of Differential Equations*, Cambridge University Press, Cambridge, 2008.

[20] R. J. LeVeque *Intermediate boundary conditions for time-split methods applied to hyperbolic partial differential equations.* Math Comp **47** (1986) , 37–54.

[21] R. J. LeVeque and J. Oliger *Numerical methods based on additive splittings for hyperbolic partial differential equations.* Math Comp **40** (1983), 469–497.

[22] J. Niesen, and W. M. Wright, *Algorithm 919: a Krylov subspace algorithm for evaluating the φ-functions appearing in exponential integrators*, ACM Trans Math Software **38**, no. 3, Art. 22 (2012).

[23] A. Ostermann and M. Roche, *Runge–Kutta methods for partial differential equations and fractional orders of convergence*, Math Comp **59** (1992), 403–420.

[24] D. Pathria, *The correct formulation of intermediate boundary conditions for Runge-Kutta time integration of initial-boundary value problems*, *SIAM J Sci Comput* **18** (1997), 1255–1266.

[25] J. M. Sanz-Serna, J. Verwer and W. Hunsdorfer, *Convergence and order reduction of Runge–Kutta methods schemes applied to evolutionary partial differential equations*, Numer Math **50** (1987), 405–418.

[26] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth & Brooks, United States of America, 1989.

Appendix: Analysis of ACR2 with and without numerical differentiation

In this appendix, we state where formula (9) comes from for ACR2 implementation and justify that the local and global error behaves with second order of accuracy. Moreover, we also state where formula (12) comes from for ACR2 implementation with numerical differentation and justify that the local error behaves with third order of accuracy in such a case. We concentrate here on the results for the local errors after time semidiscretization since the results for the errors after full discretization would follow in the same way than in [5].

As in [5], the problem to be solved may be given in an abstract setting as follows.

Let X and Y be Banach spaces and let $A : D(A) \to X$ and $\partial : X \to Y$ be linear operators. Then, we consider the nonlinear abstract non homogeneous initial boundary value problem

$$\begin{array}{rcl} u'(t) & = & Au(t) + f(t, u(t)), \quad 0 \le t \le T, \\ u(0) & = & u_0 \in X, \\ \partial u(t) & = & g(t) \in Y, \quad 0 \le t \le T, \end{array} \tag{15}$$

where the functions $f : [0, T] \times X \to X$ (in general nonlinear) and $g : [0, T] \to Y$ are regular enough.

The abstract setting (15) permits to cover a wide range of nonlinear evolutionary problems governed by partial differential equations. We use the following hypotheses

(A1) The boundary operator $\partial : D(A) \subset X \to Y$ is onto.

(A2) $\mathrm{Ker}(\partial)$ is dense in X and $A_0 : D(A_0) = \ker(\partial) \subset X \to X$, the restriction of A to $\mathrm{Ker}(\partial)$, is the infinitesimal generator of a C_0- semigroup $\{e^{tA_0}\}_{t\ge 0}$ in X, which type ω is assumed to be negative.

(A3) If $z \in \mathbb{C}$ satisfies $\Re(z) > \omega$ and $v \in Y$, then the steady state problem

$$Ax = zx, \tag{16}$$

$$\partial x = v, \tag{17}$$

possesses a unique solution denoted by $x = K(z)v$. Moreover, the linear operator $K(z) : Y \to D(A)$ satisfies

$$\|K(z)v\| \le C\|v\|, \tag{18}$$

where the constant C holds for any z such that $Re(z) \ge \omega_0 > \omega$.

(A4) The nonlinear source f belongs to $C^1([0, T] \times X, X)$.

(A5) The solution u of (15) satisfies $u \in C^2([0, T], X)$, $u(t) \in D(A^2)$ for all $t \in [0, T]$ and $Au, A^2u \in C^1([0, T], X)$.

(A6) $f(t, u(t)) \in D(A)$ for all $t \in [0, T]$, and $Af(\cdot, u(\cdot)) \in C([0, T], X)$.

The problems to be solved after time semidiscretization are

$$\left\{\begin{array}{rcl} w'_{n,1}(s) & = & Aw_{n,1}(s), \\ w_{n,1}(0) & = & u_n, \\ \partial w_{n,1}(s) & = & \partial \hat{w}_{n,1}(s), \end{array}\right. \tag{19}$$

$$\left\{\begin{array}{rcl} w'_{n,2}(s) & = & Aw_{n,2}(s), \\ w_{n,2}(0) & = & \Psi_k^{f,t_n}(w_{n,1}(\frac{k}{2})), \\ \partial w_{n,2}(s) & = & \partial \hat{w}_{n,2}(s), \end{array}\right. \tag{20}$$

with

$$\hat{w}_{n,1}(s) = u(t_n) + sAu(t_n), \tag{21}$$

$$\hat{w}_{n,2}(s) = u(t_n) + \frac{k}{2}Au(t_n) + kf(t_n, u(t_n)) + sAu(t_n), \tag{22}$$

and $\Psi_k^{f,t_n}(w_{n,1}(\frac{k}{2})) \approx v_n(k)$ is a numerical time integrator approximating the solution of

$$\left\{\begin{array}{rcl} v'_n(s) & = & f(t_n + s, v_n(s)) \\ v_n(0) & = & w_{n,1}(\frac{k}{2}) \end{array}\right.$$

Then,

$$u^{n+1} = w_{n,2}(\frac{k}{2}),$$

and the following result follows.

Theorem 1. *Let assume that hypotheses (A1)-(A6) are satisfied, and that the numerical integrator Ψ_k integrates (23) with order $p \geq 1$. Then, when integrating in time (15) with Strang method using the technique (19)-(20) with $\hat{w}_{n,1}$ in (21) and $\hat{w}_{n,2}$ in (22), the local error ρ_{n+1} satisfies $\rho_{n+1} = O(k^2)$.*

Proof. By definition, $\rho_{n+1} = \bar{u}_{n+1} - u(t_{n+1})$, where $\bar{u}_{n+1}$ is calculated through $\bar{w}_{n,1}$ and $\bar{w}_{n,2}$ as in (19) and (20) but substituting u_n by $u(t_n)$. Then,

$$\begin{array}{rcl} \bar{w}'_{n,1}(s) - \hat{w}'_{n,1}(s) & = & A(\bar{w}_{n,1}(s) - \hat{w}_{n,1}(s)) + sA^2u(t_n), \\ \bar{w}_{n,1}(0) - \hat{w}_{n,1}(0) & = & 0, \\ \partial[\bar{w}_{n,1}(s) - \hat{w}_{n,1}(s)] & = & 0, \end{array}$$

from what, using the variation of constants,

$$\bar{w}_{n,1}(\frac{k}{2}) - \hat{w}_{n,1}(\frac{k}{2}) = \int_0^{\frac{k}{2}} e^{(\frac{k}{2}-\tau)A_0}\tau A^2u(t_n)d\tau = \frac{k^2}{4}\varphi_2(\frac{k}{2}A_0)A^2u(t_n).$$

On the other hand,

$$\begin{array}{rcl} \bar{w}'_{n,2}(s) - \hat{w}'_{n,2}(s) & = & A(\bar{w}_{n,2}(s) - \hat{w}_{n,2}(s)) \\ & & +\frac{k}{2}A^2u(t_n) + kAf(t_n, u(t_n)) + sA^2u(t_n), \\ \bar{w}_{n,2}(0) - \hat{w}_{n,2}(0) & = & \Psi_k^{f,t_n}(\bar{w}_{n,1}(\frac{k}{2})) - u(t_n) - \frac{k}{2}Au(t_n) - kf(t_n, u(t_n)), \\ \partial[\bar{w}_{n,2}(s) - \hat{w}_{n,2}(s)] & = & 0. \end{array}$$

Therefore, also by the variation of constants formula,

$$\bar{w}_{n,2}(\frac{k}{2}) - \hat{w}_{n,2}(\frac{k}{2}) = e^{\frac{k}{2}A_0}[\Psi_k^{f,t_n}(\bar{w}_{n,1}(\frac{k}{2})) - u(t_n) - \frac{k}{2}Au(t_n) - kf(t_n,u(t_n))]$$
$$+\frac{k^2}{4}\varphi_1(\frac{k}{2}A_0)A^2u(t_n) + \frac{k^2}{2}\varphi_1(\frac{k}{2}A_0)Af(t_n,u(t_n)) + \frac{k^2}{4}\varphi_2(\frac{k}{2}A_0)A^2u(t_n).$$

From this, using (21) and (22) and Taylor expansions,

$$\rho_{n+1} = \bar{w}_{n,2}(\frac{k}{2}) - u(t_{n+1}) = O(k^2).$$

□

In order to be able to apply a summation-by-parts argument, so that order 2 is also proved for the global error, the following result is necessary, which assumes a bit more regularity on the solution of the problem and a bit more of accuracy on the integrator Ψ_k (see [5]).

Theorem 2. *Let us assume that hypotheses (A1)-(A6) are satisfied, $u \in C^3([0,T],X)$, $f \in C^2([0,T]\times X, X)$ and $f_u(\cdot,u(\cdot))f(\cdot,u(\cdot)), f_u(\cdot,u(\cdot))Au(\cdot) \in C([0,T],X)$. Then, when integrating (15) with Strang method using the technique (19)-(20) with $\hat{w}_{n,1}$ in (21) and $\hat{w}_{n,2}$ in (22), and assuming that the numerical integrator Ψ_k is of order $p \geq 2$, the local error ρ_{n+1} satisfies $A_0^{-1}\rho_{n+1} = O(k^3)$.*

Proof. It suffices to notice that the terms in k^2 in the previous expression of ρ_{n+1} can also be written as

$$e^{\frac{k}{2}A_0}\left[\frac{k^2}{4}\varphi_2(\frac{k}{2}A_0)A^2u(t_n) + \frac{k^2}{2}f_u(t_n,u(t_n))Au(t_n)\right.$$
$$\left.+\frac{k^2}{2}[f_t(t_n,u(t_n)) + f_u(t_n,u(t_n))f(t_n,u(t_n))]\right]$$
$$+\frac{k^2}{4}\varphi_1(\frac{k}{2}A_0)A^2u(t_n) + \frac{k^2}{2}\varphi_1(\frac{k}{2}A_0)Af(t_n,u(t_n))$$
$$+\frac{k^2}{4}\varphi_2(\frac{k}{2}A_0)A^2u(t_n) - \frac{k^2}{2}u''(t_n).$$

Then, using that, because of the definition of φ_j [5],

$$A_0^{-1}e^{\frac{k}{2}A_0} = A_0^{-1} + \frac{k}{2}\varphi_1(\frac{k}{2}A_0), \quad A_0^{-1}\varphi_1(\frac{k}{2}A_0) = A_0^{-1} + \frac{k}{2}\varphi_2(\frac{k}{2}A_0),$$
$$A_0^{-1}\varphi_2(\frac{k}{2}A_0) = \frac{1}{2}A_0^{-1} + \frac{k}{2}\varphi_3(\frac{k}{2}A_0),$$

the following is deduced simplifying the notation,

$$A_0^{-1}\rho_{n+1} = \frac{k^2}{2}A_0^{-1}[A^2u + f_uAu + Af + f_t + f_uf - u''] + O(k^3) = O(k^3).$$

□

With numerical differentiation, the problems to be solved after time semidiscretization are those in (19) and (20), but with

$$\begin{aligned}
\hat{w}_{n,1}(s) &= u(t_n) + sAu(t_n) + \frac{s^2}{2}A^2u(t_n), \qquad (23)\\
\hat{w}_{n,2}(s) &= u(t_n) + \frac{k}{2}Au(t_n) + \frac{k^2}{8}A^2u(t_n) + kf(t_n,u(t_n))\\
&\quad + \frac{k^2}{2}[f_u(t_n,u(t_n))Au(t_n) + f_t(t_n,u(t_n)) + f_u(t_n,u(t_n))f(t_n,u(t_n))]\\
&\quad + sAu(t_n) + \frac{sk}{2}A^2u(t_n) + skAf(t_n,u(t_n)) + \frac{s^2}{2}A^2u(t_n)]. \qquad (24)
\end{aligned}$$

Then, we have the following result for the local error which implies, through the standard argument of convergence which was used in [5] for Lie-Trotter, that the global error for the full discretization behaves with order 2 in the timestepsize.

Theorem 3. *Let us assume that hypotheses (A1)-(A6) are satisfied, $u \in C^3([0,T],X)$, $f \in C^2([0,T]\times X, X)$ and $f_u(\cdot,u(\cdot))f(\cdot,u(\cdot)), f_u(\cdot,u(\cdot))Au(\cdot) \in C([0,T],X)$, $u(t) \in D(A^3)$ for $t \in [0,T]$ and $A^3u \in C([0,T],X)$, when integrating (15) with Strang method using the technique (19)-(20) with $\hat{w}_{n,1}$ in (23) and $\hat{w}_{n,2}$ in (24), and assuming that the numerical integrator Ψ_k is of order $p \geq 2$, the local error ρ_{n+1} satisfies $\rho_{n+1} = O(k^3)$.*

Proof. We notice that now

$$\begin{aligned}
\bar{w}'_{n,1}(s) - \hat{w}'_{n,1}(s) &= A(\bar{w}_{n,1}(s) - \hat{w}_{n,1}(s)) + \frac{s^2}{2}A^3u(t_n),\\
\bar{w}_{n,1}(0) - \hat{w}_{n,1}(0) &= 0,\\
\partial[\bar{w}_{n,1}(s) - \hat{w}_{n,1}(s)] &= 0,
\end{aligned}$$

Therefore, by the variation of constants formula,

$$\bar{w}_{n,1}(\frac{k}{2}) - \hat{w}_{n,1}(\frac{k}{2}) = \int_0^{\frac{k}{2}} e^{(\frac{k}{2}-\tau)A_0}\frac{\tau^2}{2}A^3u(t_n)d\tau = \frac{k^3}{8}\varphi_3(\frac{k}{2}A_0)A^3u(t_n).$$

On the other hand, simplifying the notation,

$$\begin{aligned}
\bar{w}'_{n,2}(s) - \hat{w}'_{n,2}(s) &= A(\bar{w}_{n,2}(s) - \hat{w}_{n,2}(s)) + \frac{k^2}{8}A^3u + \frac{k^2}{2}Af_uAu\\
&\quad + \frac{k^2}{2}A(f_t + f_uf) + \frac{sk}{2}A^3u + skAf + \frac{s^2}{2}A^3u,\\
\bar{w}_{n,2}(0) - \hat{w}_{n,2}(0) &= \Psi_k^{f,t_n}(\bar{w}_{n,1}(\frac{k}{2}))\\
&\quad - [u + \frac{k}{2}Au(t_n) + \frac{k^2}{8}A^2u + kf + \frac{k^2}{2}f_uAu + \frac{k^2}{2}(f_t + f_uf)],\\
\partial[\bar{w}_{n,2}(s) - \hat{w}_{n,2}(s)] &= 0,
\end{aligned}$$

from what, also by the variation of constants formula,

$$\begin{aligned}
\bar{w}_{n,2}(\frac{k}{2}) - \hat{w}_{n,2}(\frac{k}{2}) &= e^{\frac{k}{2}A_0}\Big[\Psi_k^{f,t_n}(\bar{w}_{n,1}(\frac{k}{2})) \\
&\qquad -[u + \frac{k}{2}Au(t_n) + \frac{k^2}{8}A^2u + kf + \frac{k^2}{2}f_uAu + \frac{k^2}{2}(f_t + f_uf)]\Big] \\
&\quad + \int_0^{\frac{k}{2}} e^{(\frac{k}{2}-\tau)A_0}[\frac{k^2}{8}A^3u + \frac{k^2}{2}Af_uAu + \frac{k^2}{2}A(f_t + f_uf) + \frac{\tau k}{2}A^3u + \tau kAf + \frac{\tau^2}{2}A^3u]d\tau \\
&\quad = O(k^3),
\end{aligned}$$

and therefore

$$\begin{aligned}
\bar{w}_{n,2}(\frac{k}{2}) &= u + k(Au + f) + \frac{k^2}{2}(A^2u + Af + f_uAu + f_t + f_uf) + O(k^3) \\
&= u + k\dot{u} + \frac{k^2}{2}\ddot{u} = u(t_{n+1}) + O(k^3).
\end{aligned}$$

□