Segregated Runge-Kutta methods for the incompressible Navier-Stokes equations

Oriol Colomés[1][*] and Santiago Badia[1,2]

[1]*Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE), Parc Mediterrani de la Tecnologia, UPC, Esteve Terradas 5, 08860 Castelldefels, Spain.*

[2]*Universitat Politècnica de Catalunya (UPC), Jordi Girona 1-3, Campus Nord UPC, Edifici C1, 08034 Barcelona, Spain.*

SUMMARY

In this work, we propose Runge-Kutta time integration schemes for the incompressible Navier-Stokes equations with two salient properties. First, velocity and pressure computations are segregated at the time integration level, without the need to perform additional fractional step techniques that spoil high orders of accuracy. Second, the proposed methods keep the same order of accuracy for both velocities and pressures. The *segregated Runge-Kutta* methods are motivated as an implicit-explicit Runge-Kutta time integration of the projected Navier-Stokes system onto the discrete divergence-free space, and its re-statement in a velocity-pressure setting using a discrete pressure Poisson equation. We have analyzed the preservation of the discrete divergence constraint for segregated Runge-Kutta methods, and their relation (in their fully explicit version) with existing half-explicit methods. We have performed a detailed numerical experimentation for a wide set of schemes (from first to third order), including implicit and IMEX integration of viscous and convective terms, for incompressible laminar and turbulent flows. Further, segregated Runge-Kutta schemes with adaptive time stepping are proposed. Copyright © 0000 John Wiley & Sons, Ltd.

1. INTRODUCTION

Many problems in science and engineering can be simulated by incompressible flow solvers, e.g., flows around aircrafts during take-off and landing, cars, bridges, wind turbines, etc. Incompressible flows are also encountered when simulating liquid metal blankets in fusion reactors or blood flow in bio-mechanics. The increasing computer power of super-computers has motivated the interest in high-fidelity massively parallel predictive tools on unstructured meshes for this type of applications. Higher levels of accuracy in space can be based on refined meshes or higher-order approximations, being the use of hp-adaptive simulations the most refined approach so far.

The transient incompressible Navier-Stokes system of partial differential equations is nonlinear (due to convection) and indefinite (due to the divergence-free constraint), which complicates its discretization and the linear solver step. A fully implicit time-integration involves nonlinear iterations at every time step, increasing computational cost. On the other hand, it is hard to define scalable parallel solvers for non-symmetric and indefinite problems. The definition of

---

[*]Correspondence to: Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE), Parc Mediterrani de la Tecnologia, UPC, Esteve Terradas 5, 08860 Castelldefels, Spain. E-mail: ocolomes@cimne.upc.edu

*Prepared using **nmeauth.cls** [Version: 2010/05/13 v3.00]*

scalable preconditioners for this nonlinear system of equations is an open problem both for domain decomposition and multigrid techniques. Further, the nonlinear nature of the problem requires frequent preconditioner set-up steps, that make this approach computationally intensive.

The velocity-pressure block-segregation can be understood as a solver (introducing an additional splitting error) instead of a preconditioner, leading to the popular pressure-correction or fractional-step methods [1, 2, 3, 4]. This approach involves to solve decoupled a momentum equation for the velocity and a pressure Poisson equation. This is the most popular approach for the simulation of turbulent incompressible flows. The time integration of the momentum equation can be carried out using explicit, semi-implicit, or fully implicit methods. The fully implicit method has some of the drawbacks considered above, whereas an explicit integration of the viscous terms is not suitable for wall-bounded flows. In order to capture the viscous effects around solids, very refined anisotropic meshes are required, leading to too stringent viscous CFL conditions. The use of hp-adaptivity makes hard to use explicit methods, since intensive local refinement in some parts, e.g., the tip of an airfoil, leads to global time steps that must go to zero as $\frac{h^2}{p^4}$ for stability purposes, being h the characteristic element size and p the degree of interpolation, and local time-stepping cannot be efficiently exploited on parallel platforms. The use of semi-implicit methods, which treat implicitly the diffusive term and explicitly the convective one, seems to be the perfect compromise for turbulent flows around objects and viscous flows. The time step restriction is given by the convective CFL condition, which is much weaker than the diffusive one and also avoids nonlinear iterations. In fact, the majority of direct and large eddy simulations of incompressible flows involve semi-implicit methods, in which only the viscous term is treated implicitly.

When high-order time integration aims to be achieved, a popular approach is to use Runge-Kutta (RK) schemes, due to the good stability properties, high-order accuracy, and the easy computation of time error estimates for adaptive time stepping (see [5, 6]). RK schemes involve several systems of equations at each time step. Further, when we use an implicit scheme all stages can be coupled, resulting in a large system of equations to be solved, which is quite impractical in terms of CPU cost. This drawback can be bypassed using an explicit scheme, with the problems related by the time step restriction to ensure stability. Diagonally Implicit RK methods (DIRK) can be used to avoid stability problems and solving implicitly each RK stage uncoupled (see, e..g, [7]).

Due to the differential-algebraic nature of the ordinary differential system that arises from the spatial discretization, the application of RK methods to the Navier-Stokes equations is not straightforward. The typical approach is to compute the velocity at the next time step by integrating the momentum equation using some RK method (freezing the pressure gradient term), and next recover the pressure using a pressure Poisson equation (see, e.g., [8]). However, it is unclear how this approach affects the convergence error of both velocities and pressures. Alternatively, other methods perform a RK time-integration in which the velocity at every stage is enforced to be divergence-free, and next a pressure segregation is applied at every stage [2, 9]. As a result, the error due to time RK discretization is spoiled by a second-order pressure splitting error. It is common in the literature not to report pressure error in time [8, 10] or to report at most second-order of accuracy [9]. An exception to this situation is the recent work [11], where the half-explicit RK (HERK) methods for index-2 algebraic-differential equation (DAE) systems (see [12]) have been applied to the incompressible Navier-Stokes equations. These methods provide error estimates for both velocities and pressures, but require an explicit treatment of both convection and diffusion terms. Other approaches include the energy-conserving implicit RK methods in [13] (coupling all stages at the linear system).

A very accurate pressure is required in many applications, especially those involving fluid-structure interaction for high Reynolds number flows, and when evaluating drag and lift coefficients on objects. With the aim to develop high-order semi-implicit methods, we propose new time-integration schemes, that we will denote as *segregated Runge-Kutta* (SRK) methods, that do not involve any additional splitting error, since the pressure-velocity decoupled computation is already obtained at the time-integration level. These methods are motivated from the projected momentum equation onto the space of divergence-free functions, which allows us to eliminate the pressure and consider general RK schemes for the time integration. This way, we can easily prove the order of the pressure time error and attain higher than two schemes for the pressure too. The benefit of this

approach with respect to HERK methods is the flexibility to consider implicit and implicit-explicit (IMEX) versions of these methods.

In order to ease an effective preconditioning, we will favour SRK schemes that treat implicitly the viscous term, explicitly the convective terms, and segregated the pressure. The use of IMEX-SRK methods is very appealing for large-scale computations. At every stage, it only involves a vector-Laplacian (or elasticity-type) plus a mass matrix solver for the velocity and a discrete pressure Poisson solver. For this type of coercive and symmetric problems, we can make use of efficient and highly scalable domain decomposition or multigrid algorithms (see, e.g., [14]). Further, the set-up of these preconditioners can be kept on fixed meshes, reducing computational cost.

The statement of the incompressible Navier-Stokes equations is developed in Section 2. In Section 3 the time integration through RK schemes is introduced, giving an overview of the HERK methods and developing the proposed SRK schemes. Four different tests are exposed in Section 4, where the application of SRK schemes is assessed for two different manufactured analytical solutions and laminar and turbulent flow tests. Finally, some conclusions are stated in Section 5.

2. NAVIER-STOKES EQUATIONS

Let Ω be a bounded domain of $\mathbb{R}^d$, where $d = 2, 3$ is the number of space dimensions, $\Gamma = \partial \Omega$ its boundary and $(0, T]$ the time interval. The strong form of the incompressible Navier-Stokes problem consists of finding a velocity field $\mathbf{u}$ and a pressure p such that

$$\partial_t \mathbf{u} - \nabla \cdot (\nu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)) + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = \mathbf{f} \qquad \text{in } \Omega \times (0, T], \qquad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega \times (0, T], \qquad (2)$$

with $\mathbf{f}$ the force vector and ν the kinematic viscosity. Hereafter, bold characters will denote vectors and tensors. Equations (1) and (2) need to be supplied with appropriate boundary and initial conditions. The boundary Γ is divided into the Dirichlet (Γ_D) and the Neumann (Γ_N) parts such that $\Gamma_D \cup \Gamma_N = \Gamma$ and $\Gamma_D \cap \Gamma_N = \emptyset$. Then, the boundary and initial conditions can be written as

$$\mathbf{u} = \mathbf{u}_g \qquad \text{on } \Gamma_D \times (0, T], \qquad (3)$$

$$(-p \cdot \mathbf{I} + \nu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)) \cdot \mathbf{n} = \mathbf{t}_N \qquad \text{on } \Gamma_N \times (0, T], \qquad (4)$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \qquad \text{in } \Omega \times \{0\}, \qquad (5)$$

$\mathbf{n}$ being the unit outward vector normal to Γ.

We denote by $L^2(\Omega)$ the space of square integrable functions, and by $H^1(\Omega)$ the space of functions in $L^2(\Omega)$ with spatial derivatives in $L^2(\Omega)$. $L_0^2(\Omega)$ stands for the set of functions in $L^2(\Omega)$ with zero mean value on Ω. The space of functions in $H^1(\Omega)$ that vanish on the boundary are denoted by $H_0^1(\Omega)$. We consider homogeneous Dirichlet boundary conditions, since the non-homogeneous case is straightforward by a modification of the forcing term $\mathbf{f}$ and the introduction of g (see below). The weak form of the transient incompressible Navier-Stokes problem (1)-(5) reads as follows: find $[\mathbf{u}(t), p(t)] \in H_0^1(\Omega) \times L_0^2(\Omega)$ such that

$$(\partial_t \mathbf{u}, \mathbf{v}) + B(\mathbf{u}, (\mathbf{u}, p), (\mathbf{v}, q)) = \langle \mathbf{f}, \mathbf{v} \rangle + (g, q) \qquad \text{for any } [\mathbf{v}, q] \in H_0^1(\Omega) \times L_0^2(\Omega), \qquad (6)$$

almost everywhere in time, satisfying the initial condition (5) in a weak sense, where the form $B(\mathbf{a}, (\mathbf{u}, p), (\mathbf{v}, q))$ is defined as

$$B(\mathbf{a}, (\mathbf{u}, p), (\mathbf{v}, q)) := \nu(\nabla \mathbf{u}, \nabla \mathbf{v}) + b(\mathbf{a}, \mathbf{u}, \mathbf{v}) - (p, \nabla \cdot \mathbf{v}) + (q, \nabla \cdot \mathbf{u}), \qquad (7)$$

and $b(\mathbf{a}, \mathbf{u}, \mathbf{v})$ is the trilinear weak form of the convective term.

Let us now consider a quasi-uniform finite element partition $\mathcal{T}_h$ of the domain Ω from which we can construct the finite dimensional spaces for the velocity and pressure. After the discretization in

space of (6), we end up with an index-2 DAE system of equations:

$$M\dot{\mathbf{U}} + (K + C(\mathbf{U}))\mathbf{U} + G\mathbf{P} = \mathbf{F}, \tag{8}$$

$$D\mathbf{U} = \mathbf{H}, \tag{9}$$

where M is the mass matrix, K the contribution of the diffusion term, $C(\mathbf{U})$ the nonlinear convective term (related to the trilinear form b), G the pressure gradient operator and D the divergence matrix (note that $D = -G^T$). $\mathbf{U}$ and $\mathbf{P}$ are the nodal values of the discrete velocity and pressure, while $\mathbf{F}$ and $\mathbf{H}$ are the force terms of the momentum and incompressibility constraint equations, respectively.

Focusing on the matrix system (8)-(9), if we derive with respect to the time equation (9), we have that $D\dot{\mathbf{U}} = \dot{\mathbf{H}}$, assuming that D is constant in time. Then, multiplying the first equation (8) by DM^{-1} and invoking this result, we obtain an alternative equation for $\mathbf{U}$ and $\mathbf{P}$.

$$\dot{\mathbf{H}} + DM^{-1}(K + C(\mathbf{U}))\mathbf{U} + DM^{-1}G\mathbf{P} = DM^{-1}\mathbf{F}. \tag{10}$$

Assuming that $D\mathbf{U}(0) = \mathbf{H}(0)$, systems (8)-(9) and (8)-(10) are equivalent. Further, matrix $DM^{-1}G$ is invertible, due to the inf-sup condition to be satisfied by the mixed finite element space (see [15]). As a result, the pressure can be expressed in terms of $\mathbf{U}$ using (10), getting

$$-DM^{-1}G\mathbf{P} = DM^{-1}(K + C(\mathbf{U})\mathbf{U} - \mathbf{F}) + \dot{\mathbf{H}}. \tag{11}$$

Replacing this expression in (8), we obtain an equation for the velocity field only:

$$M\dot{\mathbf{U}} + \Pi(K + C(\mathbf{U}))\mathbf{U} = \Pi\mathbf{F} + G(DM^{-1}G)^{-1}\dot{\mathbf{H}}, \qquad \text{with } \Pi := (I - G(DM^{-1}G)^{-1}DM^{-1}), \tag{12}$$

I being the identity matrix. This system matrix stands for the projected Navier-Stokes system onto the discrete divergence-free space; we can easily chech that $DM^{-1}\Pi = 0$, readily leading to $D\dot{\mathbf{U}} = \dot{\mathbf{H}}$. It is a system of ordinary differential equations (ODEs) and the use of RK methods is now straightforward.

3. RUNGE-KUTTA TIME INTEGRATION

Let us consider now the Navier-Stokes semi-discrete problem given by equations (8)-(9) or (8) and (10). We consider a space discretization using mixed finite element spaces satisfying the inf-sup condition. This way, we avoid the use of stabilized formulations that involve extra terms that may couple pressure and velocity fields or fill the diagonal block related to the pressure, and change the mathematical structure of the system.

Following the motivation in Section 1, we aim to develop RK schemes for the incompressible Navier-Stokes equations that will *segregate the velocity and pressure computation while keeping high order of accuracy*. This splitting leads to the use of optimal solvers for the velocity block and the pressure block, respectively, without the need to develop efficient and scalable algorithms for indefinite systems. Further, when considering explicitly the convective term, we can maintain the same preconditioner at all stages (while the mesh does not change) and avoid the need to deal with non-symmetric (and possibly convection-dominant) systems.

A RK scheme consists of a multistage integration in which each stage is computed as a combination of the unknowns evaluated in other stages. This combination can give an implicit scheme or an explicit scheme, depending on the definition of the Butcher tableau. Implicit and explicit schemes can be combined, leading to IMEX schemes, i.e., different Butcher tableaus are used for the implicit and explicit terms (see Appendix A). System (8)-(9) can be compactly written as

$$M\dot{\mathbf{U}} = \mathscr{F}(\mathbf{U}) + \mathscr{G}(\mathbf{U},\mathbf{P}), \qquad D\mathbf{U} = \mathbf{H}, \tag{13}$$

where $\mathscr{F}(\mathbf{U})$ and $\mathscr{G}(\mathbf{U},\mathbf{P})$ are the terms to be treated implicitly and explicitly, respectively. For the implicit integration of $\mathscr{F}$, we will use the so called DIRK method; for a given stage i, it only involves the stages j such that $1 \le j \le i$. For the explicit integration of $\mathscr{G}$, in a given stage i, the method only concerns about the contribution of the stages j such that $1 \le j \le i - 1$.

3.1. Half-explicit Runge-Kutta schemes

The first approach to get a RK time integration with the properties described above is the use of half-explicit RK (HERK) methods (see [16, 17, 18] for the general definition of these methods and [11] for its very recent application to the Navier-Stokes equations). So, we consider $\mathscr{F} = 0$. HERK methods combine an explicit RK method for the momentum equation with an implicit enforcement of the discrete divergence constraint. A half-explicit integration of the Navier-Stokes equations reads as:

$$\frac{1}{\delta t}M\mathbf{U}_i = \frac{1}{\delta t}M\mathbf{U}_n + \sum_{j=1}^{i-1}\hat{a}_{ij}\mathscr{G}(\mathbf{U}_j,\mathbf{P}_j), \qquad D\mathbf{U}_i = \mathbf{H}(t_i), \tag{14}$$

where $t_i := t_n + \hat{c}_i\delta t$. We observe that the computation of $\mathbf{U}_i$ at every stage does only depend on $\mathbf{P}_j$ with $j = 1,\ldots,i-1$. Applying DM^{-1} over the momentum equation and recalling the discrete divergence constraint, we obtain the equivalent method

$$\frac{1}{\delta t}M\mathbf{U}_i = \frac{1}{\delta t}M\mathbf{U}_n + \sum_{j=1}^{i-1}\hat{a}_{ij}\mathscr{G}(\mathbf{U}_j,\mathbf{P}_j), \qquad D\mathbf{U}_n + \delta t\sum_{j=1}^{i-1}\hat{a}_{ij}DM^{-1}\mathscr{G}(\mathbf{U}_j,\mathbf{P}_j) = \mathbf{H}(t_i). \tag{15}$$

We can easily check that the second equation is a linear system for $\mathbf{P}_{i-1}$ with the system matrix $DM^{-1}G$ (see [11] for different implementations). At the end of the multi-stage computation, we update the velocity field:

$$\frac{M}{\delta t}\mathbf{U}_{n+1} = \frac{M}{\delta t}\mathbf{U}_n + \sum_{i=1}^{s}\hat{b}_i\mathscr{G}(\mathbf{U}_i,\mathbf{P}_i), \qquad D\mathbf{U}_{n+1} = \mathbf{H}(t_{n+1}). \tag{16}$$

At the velocity update, we compute the last stage pressure $\mathbf{P}_s$ as above. $\mathbf{P}_{n+1}$ does not appear in the definition of the method, but it can easily be defined using a pressure Poisson equation (see [11] for different alternatives).

However, when considering some implicit terms, the implicit treatment of the constraint in the spirit of HERK methods is not affordable. For instance, treating the diffusive term implicitly, it would involve the system matrix $D(M + \delta tK)^{-1}G$ for the pressure. So, the extension of this approach to implicit and IMEX integration schemes for the momentum equation is not feasible. It has motivated the schemes introduced below.

3.2. Segregated Runge-Kutta schemes

In order to get implicit or IMEX RK schemes for the incompressible Navier-Stokes equations, we consider the velocity-only projected system (12), which is just an ODE system that can straightforwardly be integrated using RK schemes. Let us write this problem in compact form as:

$$M\dot{\mathbf{U}} = \mathscr{F}(\mathbf{U}) + \mathscr{G}(\mathbf{U}). \tag{17}$$

In particular, we can define a method in which the viscous term is treated implicitly and the convective and pressure-related term are treated explicitly. One choice is to define the operators $\mathscr{F}$ and $\mathscr{G}$ as

$$\mathscr{F}(\mathbf{U}) := -K\mathbf{U} \qquad \text{and} \tag{18}$$

$$\mathscr{G}(\mathbf{U}) := \mathbf{F} - C(\mathbf{U})\mathbf{U} + G(DM^{-1}G)^{-1}\left(DM^{-1}(K + C(\mathbf{U})\mathbf{U} - \mathbf{F}) + \dot{\mathbf{H}}\right), \tag{19}$$

i.e., using an implicit treatment of the viscous term and an explicit one for the convective and forcing term. We note that the evaluation of the pressure-related term involves a discrete pressure Poisson equation $DM^{-1}G$. However, in order to have a segregated RK method, *the term involving $DM^{-1}G$ is treated always explicitly*. Alternatively, we could choose other definitions of $\mathscr{F}$ and $\mathscr{G}$, e.g., the

convective term and the force term could be considered implicitly, leading to

$$\mathscr{F}(\mathbf{U}) := -K\mathbf{U} + \mathbf{F} - C(\mathbf{U})\mathbf{U} \qquad \text{and} \tag{20}$$

$$\mathscr{G}(\mathbf{U}) := G(DM^{-1}G)^{-1}\left(DM^{-1}(K + C(\mathbf{U})\mathbf{U} - \mathbf{F}) + \dot{\mathbf{H}}\right). \tag{21}$$

IMEX-SRK methods could also be of especial interest in turbulent flows in which the time step restriction due to the convective CFL is in most situations smaller than the one needed to capture the smallest time scales in the flow (see for instance [19, 20]).

Considering a RK method with s stages, the velocity at the stage i, $\mathbf{U}_i$, for $1 \le i \le s$ is computed as

$$\frac{1}{\delta t}M\mathbf{U}_i = \frac{1}{\delta t}M\mathbf{U}_n + \sum_{j=1}^{i} a_{ij}\mathscr{F}(\mathbf{U}_j) + \sum_{j=1}^{i-1} \hat{a}_{ij}\mathscr{G}(\mathbf{U}_j), \tag{22}$$

where a_{ij} and $\hat{a}_{ij}$ are the coefficients of the implicit and explicit Butcher tableau, respectively. After some manipulation, we can rewrite (22) as

$$\frac{1}{\delta t}M\mathbf{U}_i = \frac{1}{\delta t}M\mathbf{U}_n + \sum_{j=1}^{i} a_{ij}\mathscr{F}(\mathbf{U}_j) + \sum_{j=1}^{i-1} \hat{a}_{ij}\mathscr{G}(\mathbf{U}_j, \mathbf{P}_j), \tag{23a}$$

$$-DM^{-1}G(\mathbf{P}_i) = DM^{-1}((K + C(\mathbf{U}_i))\mathbf{U}_i - \mathbf{F}(t_i)) + \dot{\mathbf{H}}(t_i). \tag{23b}$$

For the choice of the operator in (18), we would define $\mathscr{G}(\mathbf{U}, \mathbf{P}) := \mathbf{F} - C(\mathbf{U})\mathbf{U} - G\mathbf{P}$, leading to an IMEX-SRK scheme, whereas the choice in (20) is obtained with $\mathscr{G}(\mathbf{U}, \mathbf{P}) := -G\mathbf{P}$ and corresponds to the fully implicit SRK scheme. At the end of the multi-stage computation, we update the velocity and pressure fields as it can be shown in the following equations:

$$\frac{M}{\delta t}\mathbf{U}_{n+1} = \frac{M}{\delta t}\mathbf{U}_n + \sum_{i=1}^{s} b_i\mathscr{F}(\mathbf{U}_i) + \sum_{i=1}^{s} \hat{b}_i\mathscr{G}(\mathbf{U}_i, \mathbf{P}_i), \tag{24a}$$

$$-DM^{-1}G(\mathbf{P}_{n+1}) = DM^{-1}((K + C(\mathbf{U}_{n+1}))\mathbf{U}_{n+1} - \mathbf{F}(t_{n+1})) + \dot{\mathbf{H}}(t_{n+1}). \tag{24b}$$

Due to the fact that the resulting method involves a segregated computation of velocity and pressure, it is coined as *Segregated Runge-Kutta* (SRK) method. In this approach, we can naturally consider the viscous and/or convective term implicitly, while keeping a simple pressure Poisson equation. On the other hand, $\mathbf{P}_{n+1}$ is already defined, which is a difference compared to HERK methods.

Remark 3.1
Note that Eq. (23b) (respectively, (24b)) is equivalent to solve a Darcy-type problem, with the following expression:

$$\begin{bmatrix} M & G \\ D & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U}^* \\ \mathbf{P}_k \end{bmatrix} = \begin{bmatrix} \mathbf{F}(t_k) - (K + C(\mathbf{U}_k))\mathbf{U}_k \\ \dot{\mathbf{H}}(t_k) \end{bmatrix}, \tag{25}$$

with k being i (respectively, $n+1$), and $\mathbf{U}^*$ an auxiliar velocity field, which satisfies the discrete incompressibility constraint. System (25) can be easily preconditioned by the spectrally equivalent matrix $\mathrm{diag}(M, \tilde{L})$, where $\tilde{L}$ is in turn an optimal (and scalable) preconditioner of the Laplacian matrix [4, 15]. For large scale simulations, $\tilde{L}$ can be, e.g., an extremely scalable balancing domain decomposition preconditioner for the Poisson problem [21, 14, 22].

In the SRK methods the discrete divergence constraint $D\mathbf{U} = \mathbf{H}$ is not explicitly enforced, a difference with respect to HERK methods. However, it is implicitly enforced by the pressure Poisson equation. Let us remind that both equations lead to equivalent systems at the continuous level. In the next proposition we analyze the equivalence between the HERK method and the fully explicit version of the scheme (23)-(24), i.e., taking $\mathscr{F} = 0$.

Proposition 3.1
Let us assume that $\mathbf{H}$ is independent of time, the initial condition satisfies $D\mathbf{U}_0 = \mathbf{H}$, and the SRK scheme (23)-(24) is fully explicit, i.e., $\mathscr{F} = 0$. Then, the HERK scheme (14)-(15) and the SRK scheme (23)-(24) are equivalent.

Proof
Let us assume that $D\mathbf{U}_n = \mathbf{H}$. Both methods start with $\mathbf{U}_1 = \mathbf{U}_n$ at stage 1, see Appendix A where this condition is exposed. The SRK method also computes the pressure $\mathbf{P}_1$ as the solution of $DM^{-1}\mathscr{G}(\mathbf{U}_1,\mathbf{P}_1) = 0$. At the second stage, the HERK method computes

$$M\mathbf{U}_2 = M\mathbf{U}_n + \hat{a}_{21}\delta t\mathscr{G}(\mathbf{U}_1,\mathbf{P}_1), \qquad D\mathbf{U}_2 = \mathbf{H}.$$

The constraint leads to the pressure equation $DM^{-1}\mathscr{G}(\mathbf{U}_1,\mathbf{P}_1) = 0$. As a result, $\mathbf{P}_1$ and $\mathbf{U}_2$ are identical for both methods, the difference being the fact that $\mathbf{P}_1$ is computed at stage 1 in SRK and at stage 2 in HERK. Next, we proceed by induction. Let us assume that both methods are equivalent till stage $i-1$, i.e., we obtain the same $\mathbf{U}_j$ for $j = 1,\ldots,i-1$ and $\mathbf{P}_j$ for j for $j = 1,\ldots,i-2$. At stage i, the HERK method computes first $\mathbf{P}_{i-1}$. Since $D\mathbf{U}_n = \mathbf{H}$, and using the fact that $DM^{-1}\mathscr{G}(\mathbf{U}_j,\mathbf{P}_j) = 0$ for $j = 1,\ldots,i-2$ (due to the equivalence with the SRK method and Eq. (23)), we finally get $DM^{-1}\mathscr{G}(\mathbf{U}_{i-1},\mathbf{P}_{i-1}) = 0$. As a result, $\mathbf{P}_{i-1}$ is the same as the one obtained with SRK. Since the velocity steps at (14) and (22) are identical for both methods, we also get the same $\mathbf{U}_i$. Within SRK, the pressure $\mathbf{P}_s$ at the last stage is computed from $DM^{-1}\mathscr{G}(\mathbf{U}_s,\mathbf{P}_s) = 0$. The velocity update in both cases is also identical. Further, using the velocity update at (15) and proceeding as above, we can also check that the pressure at the last stage of HERK also satisfies this equation. As a result of the equivalence, we note that $D\mathbf{U}_{n+1} = \mathbf{H}$ also holds for SRK. The initial assumption holds for the first time step, since $D\mathbf{U}_0 = \mathbf{H}$. As a result, $D\mathbf{U}_1 = \ldots = D\mathbf{U}_n = \mathbf{H}$ holds, proving the proposition. □

   This result has another implication. Using the SRK method, we also preserve the discrete divergence constraint exactly in many situations of interest. Now, the question that arises is whether we have this property too when $\mathscr{F} \neq 0$. We analyze the fulfillment of the discrete divergence constraint for SRK.

Proposition 3.2
Let us assume that every component of $\mathbf{H}(t)$ is a p-th order polynomial in time, the initial condition satisfies $D\mathbf{U}_0 = \mathbf{H}(0)$, the RK integrator integrates exactly polynomials of order $p-1$, and $b_i = \hat{b}_i$ for $i = 1,\ldots,s$, s being the number of stages of the scheme. Then, the SRK method preserves the exact discrete divergence constraint at all time steps.

Proof
We assume that $D\mathbf{U}_n = \mathbf{H}(t_n)$. The equation $DM^{-1}(\mathscr{F}(\mathbf{U}_i) + \mathscr{G}(\mathbf{U}_i,\mathbf{P}_i)) = \dot{\mathbf{H}}(t_i)$ holds at every stage of the SRK method. Applying DM^{-1} over the velocity update (24a), we get $D\mathbf{U}_{n+1} = \sum_{j=1}^{s} DM^{-1}(b_i\mathscr{F}(\mathbf{U}_i) + \hat{b}_i\mathscr{G}(\mathbf{U}_i,\mathbf{P}_i))$. Clearly, $D\mathbf{U}_{n+1} = D\mathbf{U}_n + \sum_{j=1}^{s} b_i\dot{\mathbf{H}}(t_i)$ if $b_i = \hat{b}_i$. Since the components of $\dot{\mathbf{H}}$ are $p-1$ polynomials in time, their time integration is exact by assumption, i.e., $\sum_{j=1}^{s} b_i\dot{\mathbf{H}}(t_i) = \mathbf{H}(t_{n+1}) - \mathbf{H}(t_n)$, and $D\mathbf{U}_{n+1} = \mathbf{H}(t_{n+1})$. Since $D\mathbf{U}_0 = \mathbf{H}(0)$, it proves the proposition. □

   This result is certainly strong. Even though we are not explicitly enforcing the discrete divergence constraint at every time step, the solution does keep this desired property in many cases. (We note that the intermediate stage corrections do not hold the discrete divergence constraint unless we consider $\mathscr{F} = 0$, see Proposition 3.1.)

Remark 3.2
The assumption $b_i = \hat{b}_i$ for $i = 1,\ldots,s$ is satisfied by many RK time integrators; in particular, schemes *(1-1)*, *(1-2)*, *(2-2/1)*, *(2-3)*, *(3-3)* and *(5-3)* defined in Appendix A satisfy this condition. The assumption that a p-th order RK scheme integrates exactly $p-1$ polynomials is one of the standard so-called simplifying conditions of RK methods [12], stated as $\sum_{j=1}^{s} b_j c_j^{q-1} = \frac{1}{q}$, for $q = 1,\ldots,p$.

Remark 3.3
In this section, we have worked with the most general case in which $\mathbf{H}(t)$ is time-dependent. It only happens when the Dirichlet boundary conditions are enforced strongly and the velocity trace $\mathbf{u}_g(t)$ to be enforced on Γ_D is time-dependent. For exactly fulfilling the discrete divergence constraint in all cases, we strongly favour the weak imposition of the boundary conditions, i.e., using Nitsche's

method, having $D\mathbf{U} = 0$ in all cases. When $\mathbf{H}(t)$ is not a p-th polynomial, the method still shows the right convergence order, but the discrete divergence constraint is only computed approximately due to the time integration error.

Remark 3.4

The previous results are obtained assuming that the linear systems are being solved with exact arithmetic. Since the divergence-free condition is enforced in an incremental way, there is a potential loss of accuracy in the constraint equation due to the accumulated error of the linear system for long-term simulations (see Section 4). In any case, it is easy to correct it by projecting the velocity field when the divergence residual is larger than some threshold. Given a velocity field $\mathbf{U}$, the projected discrete divergence-free velocity field $\tilde{\mathbf{U}}$ is computed as follows:

$$F\tilde{\mathbf{U}} + G\phi = F\mathbf{U}, \qquad D\tilde{\mathbf{U}} = 0,$$

with $F = M$ (L^2-projection) or $F = M + K$ (H^1-projection). This kind of techniques are heavily used in MHD simulations in order to clean the induced magnetic field of its non-divergence free component.

Finally, let us analyze the error introduced by the SRK methods, which is straightforward from the general RK schemes.

Proposition 3.3

Let us consider the SRK method with a p-th order scheme. The error for the velocity and the pressure is reduced as $\mathcal{O}(\delta t^p)$.

Proof

Let us denote the time-continuous solution as $(\mathbf{U}^{\mathrm{ex}}(t), \mathbf{P}^{\mathrm{ex}}(t))$. The convergence order for the velocity is $\|\mathbf{U}_n - \mathbf{U}^{ex}(t_n)\|_M \leq c\delta t^p$ (where $\|\cdot\|_M$ denotes the norm endowed by matrix M) since the SRK method for the velocity amounts to a standard RK scheme for the ODE system (17) (see [23]). The pressure error is straightforward from the velocity estimate, Eq. (24b), and the fact that $DM^{-1}G$ is invertible. In order to obtain the pressure error, let us subtract Eq. (11) from Eq. (24b) at time value t_n, getting:

$$-DM^{-1}G(\mathbf{P}_n - \mathbf{P}^{\mathrm{ex}}(t_n)) = DM^{-1}((K + C(\mathbf{U}_n))\mathbf{U}_n - DM^{-1}(K + C(\mathbf{U}^{\mathrm{ex}}(t_n))\mathbf{U}^{\mathrm{ex}}(t_n)$$
$$= DM^{-1}((K + C(\mathbf{U}_n))(\mathbf{U}_n - \mathbf{U}^{\mathrm{ex}}(t_n))) + DM^{-1}C(\mathbf{U}_n - \mathbf{U}^{\mathrm{ex}}(t_n))\mathbf{U}^{\mathrm{ex}}(t_n).$$

Since $DM^{-1}G$ is a positive-definite matrix (due to the discrete inf-sup condition) we easily get

$$\|(\mathbf{P}_n - \mathbf{P}^{\mathrm{ex}}(t_n))\|_M \leq c\|\mathbf{U}_n - \mathbf{U}^{\mathrm{ex}}(t_n)\|_M \leq c\delta t^p,$$

where the constant c certainly depends on the spatial mesh and the modulus of the computed and exact velocity, but not on δt. It proves the proposition.

$\square$

3.3. Order reduction phenomena

It is known that RK approximation for PDEs suffer from order reduction phenomena for schemes of order greater than two. Many works have been devoted to this issue, e.g., [24, 25], where they prove that only under certain conditions the order in time of the fully discrete scheme equals the conventional order of the RK formula. However, the authors of [24, 25] say that these conditions are not natural and, in general, the order in time for schemes of conventional order greater than two will be strictly smaller. In particular, [25] stated that for DIRK schemes of third and fourth-order, in case of inhomogeneous and time dependent boundary conditions, the actual order of these methods is 2.

Boscarino in [26] derives a third-order IMEX RK method for stiff problems that *does not suffer from the order reduction phenomena*. It is stated in this work that most of IMEX RK schemes of order greater than two suffer from order reduction in the stiff regime. The important point here is

that for convection-diffusion problems the semi-discrete system becomes stiffer when the spatial mesh is refined. Then, the order reduction phenomena becomes relevant when we use high-order time integrators in fine spatial meshes. This scheme is included in Appendix A, denoted by *(5-3)*, and tested in section 4.

4. NUMERICAL EXPERIMENTS

In this section we aim to see the performance of the SRK methods proposed previously with different test cases. We start with an analytical manufactured solution that belongs to the finite element space, so we do not have any spatial error. The next test case is the laminar flow around a cylinder, a widely used benchmark for laminar flows. Finally, we test a turbulent case, the also widely used Taylor-Green vortex flow.

All the different schemes have been implemented in the FEMPAR (Finite Element Multiphysics and massively PARallel) numerical software. FEMPAR is an open source in-house developed, parallel hybrid OpenMP/MPI, object-oriented (OO) framework which, among other features, provides the basic tools for the efficient parallel distributed-memory implementation of substructuring domain decomposition solvers [21, 14].

4.1. Manufactured analytical solution

With this test, we want to check that the methods proposed in this work achieve the desired order of convergence in time for the Butcher tableaus defined in Appendix A. In order to analyze the convergence order in time, we want to make sure that the solution is not polluted by the spatial error. Then, to eliminate the error of the spatial component, we define an analytical solution in the finite element space. This means that we can capture exactly the solution in space but not in time. The analytical solution in a 2D domain is chosen to be:

$$\mathbf{u}(x,y,t) = \begin{pmatrix} x \\ -y \end{pmatrix} \sin\left(\frac{\pi}{10}t\right) \exp\left(\frac{t}{25}\right), \tag{26}$$

$$p(x,y) = x+y. \tag{27}$$

This solution belongs to the finite element space, even for linear elements. We use inf-sup stable elements of the type $Q2 - Q1$ in order to avoid the use of a stabilization which would introduce non-desired extra terms. This test is solved in the unit square $\Omega = (0,1)^2$, discretized with a very coarse mesh with 10 elements per direction. With this coarse discretization we do not expect significant impact of the order reduction phenomena exposed in Subsection 3.3 for small viscosity since it occurs in the stiff regime, i.e., small element sizes and high viscosity. We define three different viscosities, $v = \{1.0, 0.1, 0.01\}$. We run this test in the time domain $(t_0, T) = (0, 0.1)$, using several time step sizes. In particular, we start with a large time step $\delta t = 0.1$ and we reduce it by a half recursively four times until we reach $\delta t = 1.25 \cdot 10^{-2}$. The errors in the velocity and pressure fields, e_u and e_p, are computed as the ℓ^∞-norm (with respect to the nodal values) of the difference between the computed solution and the analytical one at $t = T$.

As we use either explicit or IMEX time integration schemes, we have to be careful with the CFL number. Here we will have a diffusive CFL number (CFL_v), which will limit the method when we use a fully explicit scheme, and a convective CFL number (CFL_u), which will limit the method when we use an IMEX scheme with only the diffusive term integrated implicitly, i.e.,

$$\text{CFL}_v = \frac{v\delta t}{(h/p^2)^2}, \qquad \text{CFL}_u = \frac{u\delta t}{(h/p^2)},$$

being h the characteristic element size, p the degree of interpolation, and u the characteristic velocity. In this test, we have $h = 0.1$, $p = 2$ and $u \sim 0.0315$. Table I shows the CFL values for each time step and viscosity.

Table I. CFL values.

δt	CFL$_\nu$			CFL$_u$
	$\nu = 1.0$	$\nu = 0.1$	$\nu = 0.01$	
$1.0 \cdot 10^{-1}$	160	16	1.6	0.126
$5.0 \cdot 10^{-2}$	80	8	0.8	0.063
$2.5 \cdot 10^{-2}$	40	4	0.4	0.032
$1.25 \cdot 10^{-2}$	20	2	0.2	0.016

Note that the characteristic velocity u is very small, so the CFL$_u$ number will also be very small (much lower than 1.0), since it only depends linearly on the mesh size. Then we do not expect instabilities due to the explicit treatment of the convective term in this test. We cannot say the same for the CFL$_\nu$ number, which is larger than 1.0 in most of the cases because of the quadratic dependency on the mesh size.

In this test we consider three different situations: 1) Fully implicit SRK scheme; 2) SRK scheme with diffusive term integrated implicitly and convective term explicitly; 3) Fully explicit SRK scheme.

Fully implicit SRK. Here we consider a SRK scheme with the convective and diffusive terms integrated implicitly. That is to set the operators $\mathscr{F}$ and $\mathscr{G}$ that appear in Eqs. (23) and (24) equal to

$$\mathscr{F}(\mathbf{U}) := \mathbf{F} - (K + C(\mathbf{U}))\mathbf{U}, \qquad \text{and} \qquad \mathscr{G}(\mathbf{U}, \mathbf{P}) := -G\mathbf{P}.$$

It is important to highlight here that this scheme is nonlinear, since we have the convective velocity on the left-hand side of the equation.

Fig. 1 shows the convergence rate of the velocity and the pressure fields for the different viscosity choices. Looking at the stiffest case, Figs. 1(a) and 1(b), where $\nu = 1.0$, we see that schemes *(2-2/2)* and *(4-3)* seem to converge in a rate lower than two, which could be justified by the order reduction phenomena. Note that precisely these two methods do not satisfy the incompressibility constrain when time dependent boundary conditions are applied, see Proposition 3.2. Almost the same behaviour is observed for the scheme *(2-3)* which also has an order reduction in its convergence. On the other hand, scheme *(3-3) has a reduction on the order of convergence but it is still greater than two. Finally, scheme (5-3) is not affected by the order reduction phenomena.*

When we reduce the viscosity to $\nu = 0.1$, the schemes seem to start to recover their prescribed convergence rate. In Figs. 1(c) and 1(d) the convergence rate in time are plotted for the velocity and pressure fields, respectively. It is seen that for small time steps all schemes converge with the correct rate, but for larger time steps the *(2-2/1)* and *(4-3)* schemes have order reduction.

Finally, in Figs. 1(e) and 1(f), where we show the convergence rate for the velocity and pressure fields with $\nu = 0.01$, all schemes considered in this work perform with the prescribed convergence rate.

IMEX-SRK. In this case the SRK scheme is defined only with the diffusive term integrated implicitly, while the convective one is treated explicitly. Then, the operators $\mathscr{F}$ and $\mathscr{G}$ will be

$$\mathscr{F}(\mathbf{U}) := -K\mathbf{U}, \qquad \text{and} \qquad \mathscr{G}(\mathbf{U}, \mathbf{P}) := \mathbf{F} - C(\mathbf{U})\mathbf{U} - G\mathbf{P}.$$

Note that for this case, there will be the limitation on the hyperbolic CFL$_u$ number, which is less restrictive than the parabolic one. As it is seen in Table I, it is always less than 1.0 for the chosen time step sizes.

Fig. 2 depicts the velocity and pressure convergence rate using different viscosities for this second case. For the highest viscosity $\nu = 1.0$ (Figs. 2(a) and 2(b)), we note that almost all methods perform in a similar way as the fully implicit SRK case, with the difference that scheme *(3-3)* is also showing an order reduction in its convergence rate, being of 2nd order. In this case, we can state that the *(5-3)*

(a) Velocity convergence, $\nu = 1.0$ (b) Pressure convergence, $\nu = 1.0$

(c) Velocity convergence, $\nu = 0.1$ (d) Pressure convergence, $\nu = 0.1$

(e) Velocity convergence, $\nu = 0.01$ (f) Pressure convergence, $\nu = 0.01$
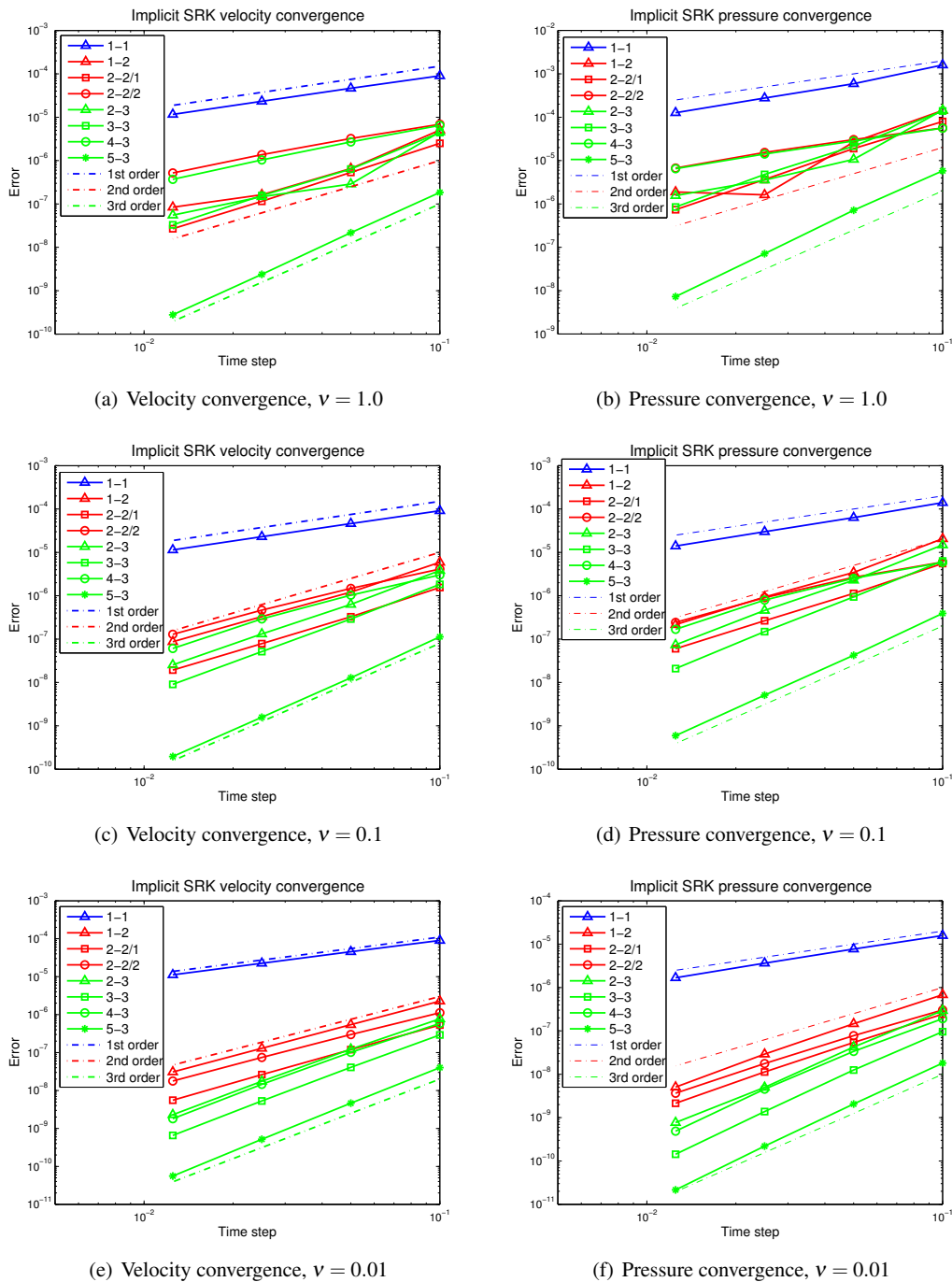
Figure 1. Fully implicit SRK.

scheme is the only third-order scheme that does not show an order reduction in the time convergence, although for the smallest time steps, the velocity convergence rate is a little bit lower than 3.

When we reduce the viscosity to $\nu = 0.1$ or $\nu = 0.01$, all schemes show the same behaviour as the fully implicit SRK case, see Figs. 2(c), 2(d), 2(e), and 2(f). Let us note that the *(2-2/1)* scheme has much lower error in the IMEX-SRK case than in the fully implicit SRK case.
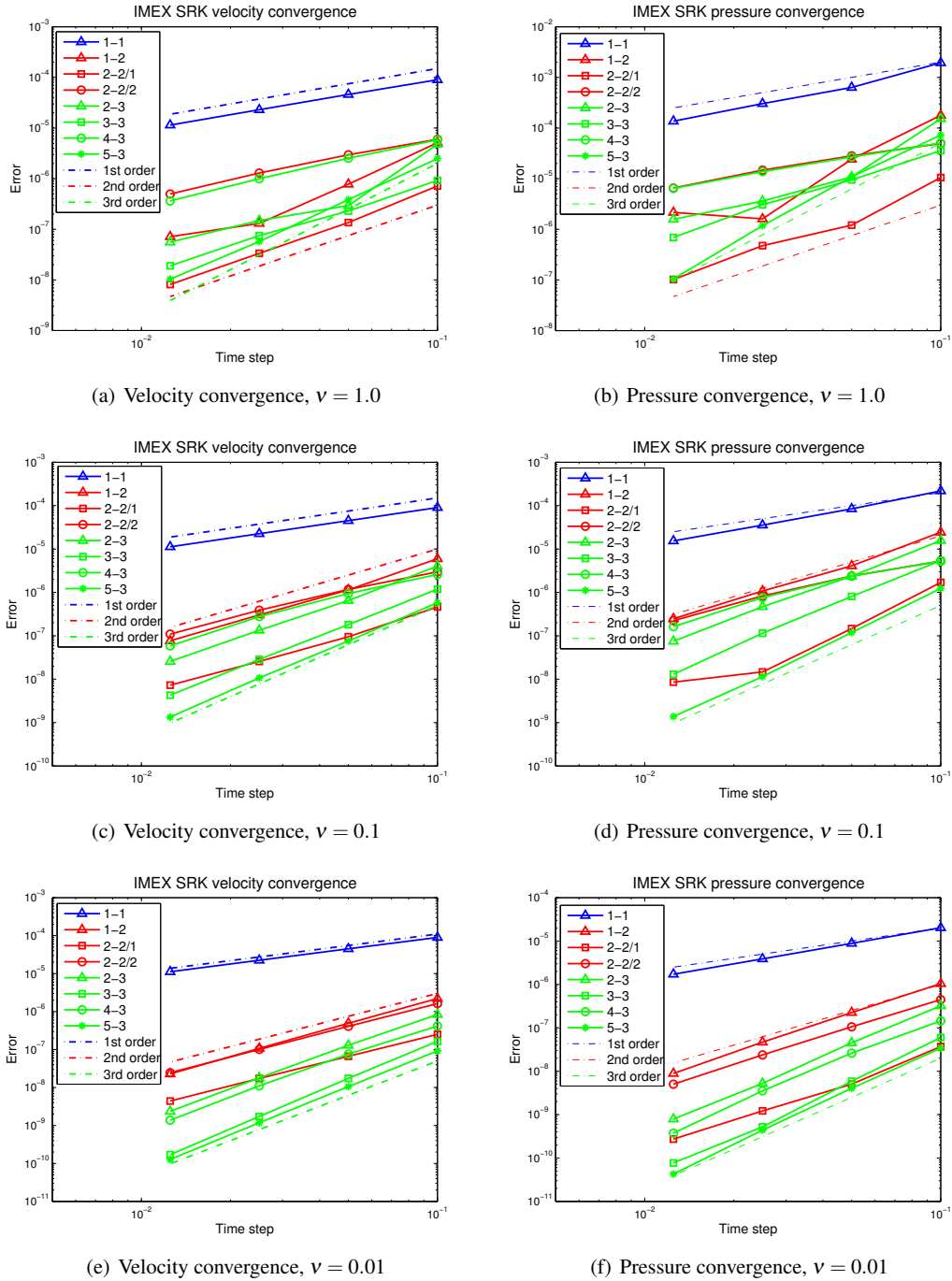
(a) Velocity convergence, $\nu = 1.0$

(b) Pressure convergence, $\nu = 1.0$

(c) Velocity convergence, $\nu = 0.1$

(d) Pressure convergence, $\nu = 0.1$

(e) Velocity convergence, $\nu = 0.01$

(f) Pressure convergence, $\nu = 0.01$

Figure 2. SRK convergence with convection integrated explicitly and diffusion integrated implicitly.

Fully explicit SRK.   Finally we test the fully explicit situation, which consist on sending all terms to the right-hand side of the equation. The operators $\mathscr{F}$ and $\mathscr{G}$ in Eqs. (23) and (24) will read

$$\mathscr{F}(\mathbf{U}) := 0, \qquad \text{and} \qquad \mathscr{G}(\mathbf{U}, \mathbf{P}) := \mathbf{F} - (K + C(\mathbf{U}))\mathbf{U} - G\mathbf{P}.$$

In this case the CFL number that limits the stability of the method is given by the parabolic one (CFL_ν). This is far more restrictive than CFL_u in the IMEX case, and for the setting described

above we get unstable results. Only when the viscosity is $\nu = 0.01$, the CFL_ν values are in the order or smaller than the critical value 1.0.

Fig. 3 depicts the convergence rate for this case, where we see that for the velocity (Fig. 3(a)) and the pressure (Fig. 3(b)) fields the order of convergence is the desired one for most of the schemes, except for the *(5-3)* scheme which converges with a higher order. Here does not appear the order reduction phenomena since it is not present for explicit schemes.
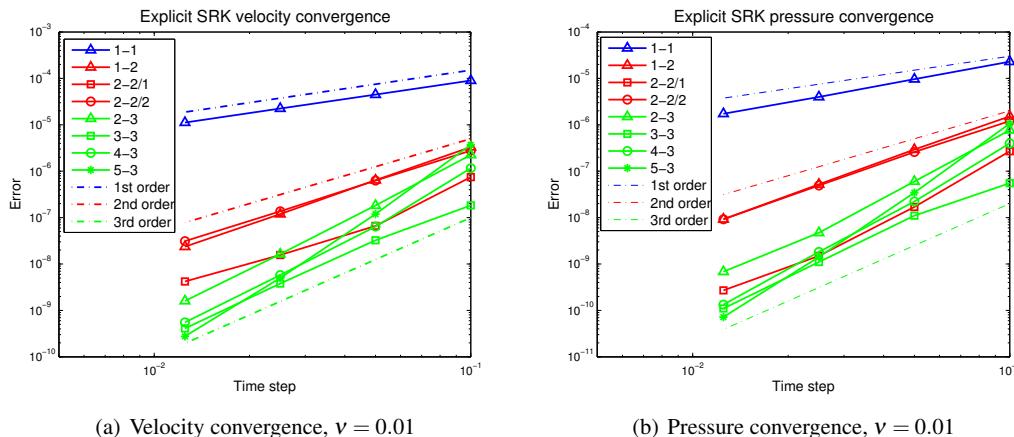

(a) Velocity convergence, $\nu = 0.01$ (b) Pressure convergence, $\nu = 0.01$

Figure 3. Fully explicit SRK convergence.

Discrete divergence constraint preservation. As stated in Proposition 3.2, the discrete divergence constraint is preserved when the strongly imposed Dirichlet data is a polynomial of order at most $p - 1$ in time. In order to show this phenomena we solve the same problem given by (26)-(27), but considering a second order polynomial for the time dependency. The analytical solution to be solved in this case will be

$$\mathbf{u}(x,y,t) = \begin{pmatrix} x \\ -y \end{pmatrix} t^2,$$

$$p(x,y) = x + y.$$

This problem is solved with the fully implicit SRK method from $t = 0$ to $t = 2.0$ using a time step size $\delta t = 1.0 \cdot 10^{-2}$ and a viscosity $\nu = 0.01$. The linear solver tolerance has been set equal to $1.0 \cdot 10^{-8}$, and for the implicit version, the nonlinear tolerance is $1.0 \cdot 10^{-6}$. In Fig. 4 the evolution of $\|\nabla \cdot \mathbf{u}\|$ is depicted for all schemes considered in Appendix A, for both the implicit and explicit versions of the SRK method. We see that for the first-order *(1-1)* scheme the discrete divergence constraint is not preserved as it was expected. Moreover, the second-order schemes seem to give really accurate results when evaluating the discrete divergence, even when the time dependence of the solution is of order 2. The *(2-2/2)* and *(4-3)* schemes, which do not satisfy the condition $b_i = \hat{b}_i$ for $i = 1, ..., s$, have the worst performance compared to the other methods of the same order. Although the errors are very small, we can observe the effect of the accumulation of solver error commented in Remark 3.4, which leads to an increasing value of $\|\nabla \cdot \mathbf{u}\|$. In any case, the third order schemes that have been proved to preserve the discrete divergence constraint (see Proposition 3.2) keep $\|\nabla \cdot \mathbf{u}\|$ below 10^{-9}.

4.2. Beltrami flow

In the manufactured analytical solution stated in (26)-(27) the spatial error is not present since the solution belongs to the FE space. In order to check the behaviour of SRK methods when solving problems with spatial error originated by the discretization in space, we consider an analytical

(a) Implicit SRK   (b) Explicit SRK
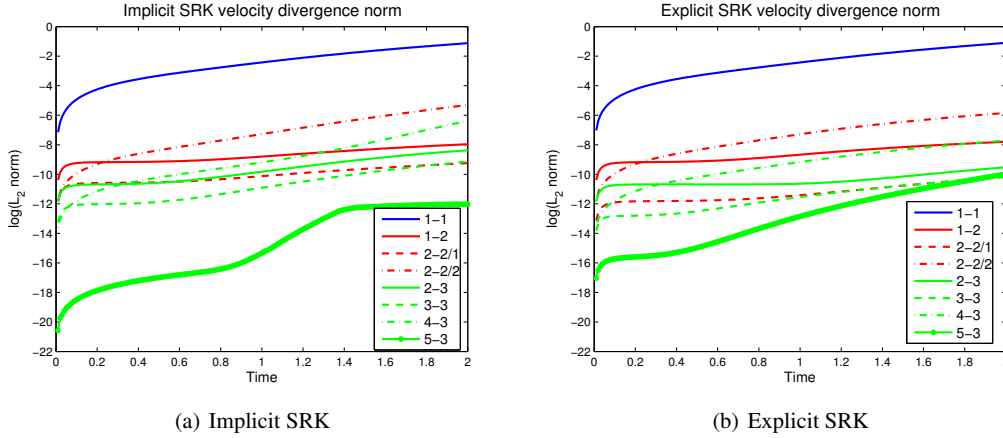
Figure 4. $\|\nabla \cdot \mathbf{u}\|$ for the implicit and explicit SRK schemes.

solution that does not belong to the FE space. A 3D Beltrami flow like the one defined in [27] is used in this subsection, but in this case a pressure with no dependence in time is defined. The flow is solved in a cube centered on $(x,y,z) = (0,0,0)$ and with a edge size $L = 2$, the viscosity of the problem in this case is set as $\nu = 0.01$.

$$\mathbf{u}(x,y,z,t) = \begin{pmatrix} -a\left[e^{ax}\sin(ay+dz)+e^{az}\cos(ax+dy)\right] \\ -a\left[e^{ay}\sin(az+dx)+e^{ax}\cos(ay+dz)\right] \\ -a\left[e^{az}\sin(ax+dy)+e^{ay}\cos(az+dx)\right] \end{pmatrix} e^{-d^2 t}, \tag{28}$$

$$p(x,y,z) = -\frac{a^2}{2}\left[e^{2ax}+e^{2ay}+e^{2ay}+2\sin(ax+dy)\cos(az+dx)e^{a(y+z)}\right. \tag{29}$$

$$\left. +2\sin(ay+dz)\cos(ax+dy)e^{a(z+x)}+2\sin(az+dx)\cos(ay+dz)e^{a(x+y)}\right].$$

To analyze the effect of the spatial error we solve the problem with the analytical solution (28)-(29) refining both in time and space keeping the ratio $\delta t / h$ constant. The problem is solved using $Q2 - Q1$ elements with the IMEX SRK method, for all schemes stated in Appendix A. A first result is obtained setting the $\delta t / h = 0.01$, with the time step sizes $\delta t = \{5.0 \cdot 10^{-3}, 2.5 \cdot 10^{-3}, 1.25 \cdot 10^{-3}, 6.25 \cdot 10^{-4}\}$ and solving from $t = 0$ till $T = 5.0 \cdot 10^{-3}$. A second test is done with a smaller ratio, $\delta t / h = 0.002$, being the time steps sizes $\delta t = \{1.0 \cdot 10^{-3}, 5.0 \cdot 10^{-4}, 2.5 \cdot 10^{-4}, 1.25 \cdot 10^{-4}\}$ from $t = 0$ till $T = 1.0 \cdot 10^{-3}$. In Fig. 5 the error convergence for both velocity and pressure fields is depicted. The results for the case with $\frac{\delta t}{h} = 0.01$ are shown in Fig. 5(a), where we clearly see that the convergence rate for the pressure field is two, the one prescribed by the spatial discretization error, so here all schemes give the same result since the spatial error prevails over the temporal error. But looking at the velocity field results at the same figure, it is seen that the third-order of convergence given by the theoretical spatial error convergence is reduced. Here we see how the first order scheme *(1-1)* starts loosing the convergence rate given by the spatial error and exhibits the convergence rate prescribed by the temporal discretization. All the other schemes converge with a second order slope, even the third-order schemes. This order reduction phenomena is also observed in [24] when both δt and h are refined simultaneously. When we select a smaller ratio of $\delta t / h$, see Fig. 5(b), the temporal error is masked by the spatial error, giving a third-order convergence for the velocity field and a second-order convergence for the pressure field. A little reduction of the order is observed at the smallest time step sizes for the velocity field in Fig. 5(b).

4.3. 2D Laminar flow around a cylinder

Once studied the behaviour of the different methods proposed in Section 3 for a manufactured analytical solution, we study a widely used laminar flow benchmark which is the flow around a

(a) Error convergence, $\frac{\delta t}{h} = 0.01$ (b) Error convergence, $\frac{\delta t}{h} = 0.002$
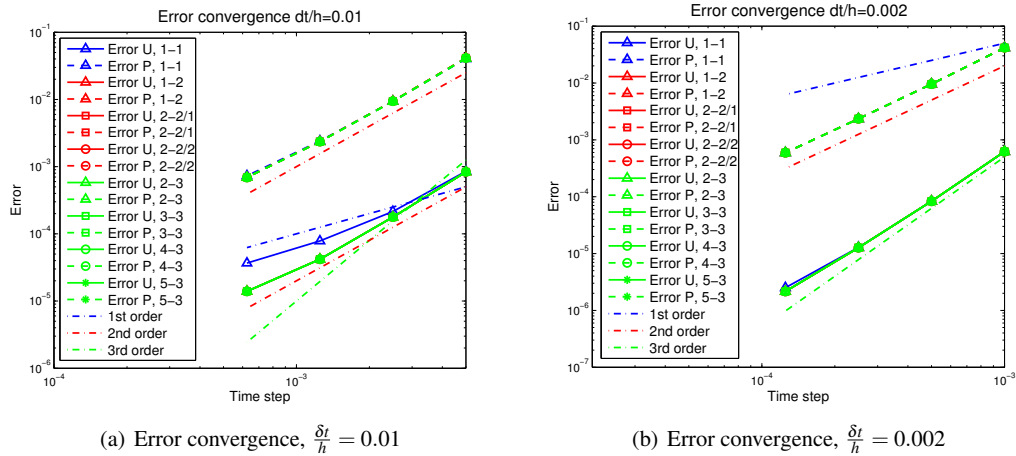
Figure 5. Beltrami flow error convergence (red and blue lines below green line).

cylinder for a low Reynolds number $Re = 100$. A detailed overview of benchmark computations of laminar flow around a cylinder are given in [28]. The test performed in the current work is called *2D-2* in that paper and is defined as shown in Fig. 6. It basically consists in a rectangular channel with a cylinder located near the inflow boundary. A non-slip condition is imposed in the cylinder wall and the channel walls that are perpendicular to the flow direction (x). The inflow condition is
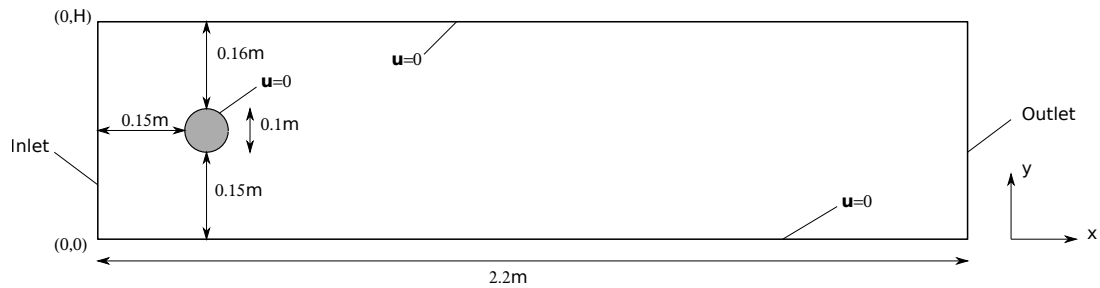

Figure 6. Flow around a cylinder test geometry.

$$\mathbf{u}(0,y,t) = \begin{pmatrix} u_x \\ u_y \end{pmatrix} = \begin{pmatrix} 4U_m \frac{y(H-y)}{H^2} \\ 0 \end{pmatrix},$$

with the maximum velocity $U_m = 1.5$ m/s and $H = 0.41$ the channel height.

Our aim is not only to see the order of convergence of the time integration schemes proposed, but also to compare the results with a detailed benchmark that has been used to test different algorithmic approaches. Then, we compute the parameters needed for this comparison, namely the drag coefficient c_D, the lift coefficient c_L, and the pressure difference ΔP as functions of time for one period $[t_0, t_0 + 1/f]$, f being the frequency of separation. The values that we will use in the comparison are the maximum drag coefficient $c_{D_{\max}}$, the maximum lift coefficient $c_{L_{\max}}$, the Strouhal number St and the pressure difference $\Delta P(t)$ at $t = t_0 + 0.5/f$. The initial time t_0 should correspond to the flow state with $c_{L_{\max}}$. The drag coefficient c_D and the lift coefficient c_L are given by

$$c_D = \frac{2F_D}{\rho \bar{U}^2 D}, \qquad c_L = \frac{2F_L}{\rho \bar{U}^2 D},$$

being $\rho = 1.0$ the fluid density, $D = 0.1$ the cylinder diameter, and $\bar{U} = 1.0$ the mean velocity. The drag F_D and lift F_L forces are defined as

$$F_D = \int_S \left(\rho \nu \frac{\partial v_t}{\partial n} n_y - p n_x \right) \mathrm{d}S, \qquad F_L = -\int_S \left(\rho \nu \frac{\partial v_t}{\partial n} n_x - p n_y \right) \mathrm{d}S,$$

with S the cylinder surface and n the normal vector on S, with n_x and n_y the x-component and y-component, respectively. v_t is the tangential velocity on S for the tangent vector $t = (n_y, -n_x)$. These surface integrals are computed as the residual of the weak form on boundary nodes, as advocated in [29] for accuracy reasons.

In order to reduce computational cost, we first compute the flow from $t = 0$ to $t = 8.0$ with a monolithic fully implicit Crank-Nicolson scheme with a time step equal to $\delta t = 0.05$. In Fig. 7 we show the vorticity field at $t = 8.0$. It can be seen that at that time the flow is fully developed. So, at $t = 8.0$ we can start the computations with the different schemes studied in this work. The computations are performed from $t = 8.0$ to $t = 8.4$, ensuring that we have two maximums of the lift coefficient, so we have a complete period of data after the first lift coefficient maximum.

Figure 7. Vorticity field at $t = 8.0$.

As exposed in the discussion of results in [28], the use of explicit schemes for the time integration of laminar flows is not an efficient approach. The restriction on the time step size to ensure stability of the method is critical since the physical time scale may be much larger. As a result, we will only focus on the fully implicit and IMEX approaches. In this test we consider two different situations: 1) Fully implicit SRK scheme; 2) SRK scheme with diffusive term integrated implicitly and convective term explicitly.

This test is solved using a mesh with 13886 $Q2 - Q1$ elements, for all the schemes defined by the Butcher tableaus exposed in Appendix A and for several time step sizes. In particular, the problem is solved with $\delta t = \{2.0 \cdot 10^{-2}, 1.0 \cdot 10^{-2}, 5.0 \cdot 10^{-3}, 2.5 \cdot 10^{-3}, 1.25 \cdot 10^{-3}\}$ for each scheme. Furthermore, an extra computation for the *(3-3)* scheme is done with $\delta t = 3.125 \cdot 10^{-4}$ in order to have a more accurate result from which we can compare to do the convergence analysis.

In [28] there is not a prescribed correct value for the benchmark quantities, but there is a range within which most of the reported values are located. Then, we expect that our computation results will fit into these bounds.

Fully implicit SRK. Here, as the flow is laminar, we do not expect the nonlinearity of the convective term to play a decisive role in the computational cost. So, it seems natural to consider an implicit treatment of this term, especially taking into account that the explicit treatment of this term involves time stepping restrictions. Thus, the schemes used in this first case are given by defining the operators $\mathscr{F}$ and $\mathscr{G}$ in Eqs. (23) and (24) as

$$\mathscr{F}(\mathbf{U}) := \mathbf{F} - (K + C(\mathbf{U}))\mathbf{U}, \qquad \text{and} \qquad \mathscr{G}(\mathbf{U}, \mathbf{P}) := -G\mathbf{P}.$$

Let us point out that for this test the *(1-2)* and *(2-3)* schemes are unstable for the largest time step sizes, i.e. $\delta t \geq 1.25 \cdot 10^{-3}$ for the *(1-2)* scheme and $\delta t \geq 2.5 \cdot 10^{-3}$ for the *(2-3)* scheme. Therefore, these two schemes will not be taken into account in the results shown in this first case. The benchmark quantities of the different computations are given in Table II, together with the

lower and upper bounds of the results in [28]. It is seen that most of the computed quantities fit into the benchmark bounds; the Strouhal number seems to be a little bit greater than the upper bound in some cases. Note that aside the Strouhal number, the other quantities converge to a value that is inside the benchmark range. As it is expected, the schemes with higher order have more results in the correct range and give very accurate results earlier when we refine the time step.

Table II. Benchmark 2D-2 results with implicit convection.

Scheme	δt	$c_{D_{max}}$	$c_{L_{max}}$	St	ΔP
$(1-1)$	2.000e-02	3.1984	0.8261	0.2941	2.4291
$(1-1)$	1.000e-02	3.2024	0.8704	0.2941	2.4398
$(1-1)$	5.000e-03	3.2108	0.9158	0.3030	2.4568
$(1-1)$	2.500e-03	3.2182	0.9441	0.3030	2.4700
$(1-1)$	1.250e-03	3.2230	0.9598	0.3042	2.4780
$(2-2/1)$	2.000e-02	3.2423	1.0121	0.3125	2.5082
$(2-2/1)$	1.000e-02	3.2329	1.0039	0.3030	2.4867
$(2-2/1)$	5.000e-03	3.2304	1.0008	0.3077	2.4852
$(2-2/1)$	2.500e-03	3.2298	1.0009	0.3053	2.4884
$(2-2/1)$	1.250e-03	3.2296	1.0007	0.3065	2.4883
$(2-2/2)$	2.000e-02	3.2418	1.0819	0.3125	2.4954
$(2-2/2)$	1.000e-02	3.2323	1.0063	0.3030	2.4835
$(2-2/2)$	5.000e-03	3.2301	0.9973	0.3077	2.4845
$(2-2/2)$	2.500e-03	3.2297	0.9997	0.3053	2.4883
$(2-2/2)$	1.250e-03	3.2296	1.0003	0.3065	2.4883
$(3-3)$	2.000e-02	3.2361	1.0138	0.3125	2.5027
$(3-3)$	1.000e-02	3.2304	1.0013	0.3030	2.4849
$(3-3)$	5.000e-03	3.2298	0.9999	0.3077	2.4848
$(3-3)$	2.500e-03	3.2296	1.0007	0.3053	2.4883
$(3-3)$	1.250e-03	3.2296	1.0006	0.3065	2.4883
$(4-3)$	2.000e-02	3.2321	1.1003	0.3125	2.4788
$(4-3)$	1.000e-02	3.2285	0.9948	0.3030	2.4802
$(4-3)$	5.000e-03	3.2285	0.9954	0.3077	2.4839
$(4-3)$	2.500e-03	3.2294	1.0000	0.3053	2.4881
$(4-3)$	1.250e-03	3.2296	1.0005	0.3065	2.4882
$(5-3)$	2.000e-02	3.2249	0.9770	0.3125	2.4951
$(5-3)$	1.000e-02	3.2290	0.9977	0.3030	2.4843
$(5-3)$	5.000e-03	3.2295	0.9995	0.3077	2.4847
$(5-3)$	2.500e-03	3.2296	1.0006	0.3053	2.4883
$(5-3)$	1.250e-03	3.2296	1.0006	0.3065	2.4883
lower bound		3.2200	0.9900	0.2950	2.4600
upper bound		3.2400	1.0100	0.3050	2.5000

To check the convergence in time of the different time integration schemes, we perform a convergence analysis computing the ℓ^∞-norm of the velocity and the pressure errors, e_u and e_p. In this case, the solution is not compared against an analytical one but computed with a finer time step ($\delta t = 3.125 \cdot 10^{-4}$) with the *(3-3)* scheme. The reason of this choice is related to the efficiency of this scheme, discussed below.

In Fig. 8 we show the convergence rate for both velocity and pressure fields, Figs. 8(a) and 8(b), respectively. A first conclusion that we can make is that for the largest time steps the order of convergence is not the prescribed one, especially for the first order scheme. It is explained by the fact that the largest time step sizes are greater than the one required for stability purposes when a fully-explicit scheme is used. For the time steps smaller than $2.0 \cdot 10^{-2}$ the convergence rates are the expected ones, with the exception of *(1-1)* and *(4-3)* schemes, with a convergence rate lower than the expected one.

Fig. 8 shows that the most accurate schemes are *(3-3)* and *(5-3)*. Now, let us analyze the computational cost of these methods based on the CPU time for a given target error. In Fig. 9 it is clearly seen that the most efficient scheme is *(3-3)*, even for relatively high error ($\sim 4 \cdot 10^{-2}$). For
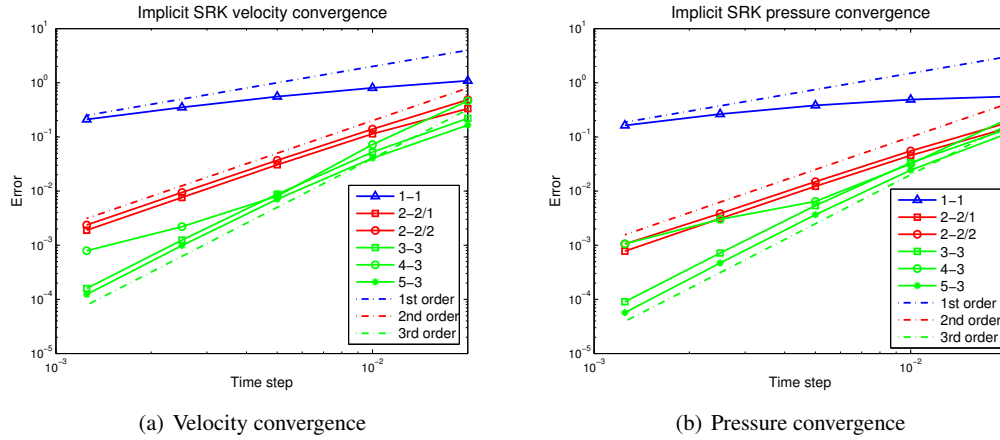
(a) Velocity convergence (b) Pressure convergence

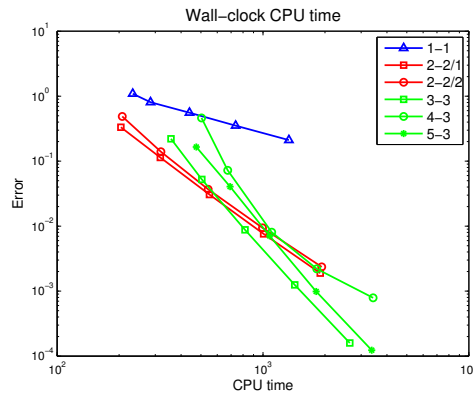Figure 8. Fully implicit SRK convergence.

Figure 9. Fully implicit SRK CPU time efficiency.

larger target errors, the *(2-2/1)* schemes is competitive. We note that the first order *(1-1)* scheme is not competitive at all, since the error is too large and there is no significant gain in CPU time.

IMEX-SRK. In this case, we consider an explicit time integration approach of the convective term. That is to define the operators $\mathscr{F}$ and $\mathscr{G}$ in Eqs. (23) and (24) as

$$\mathscr{F}(\mathbf{U}) := \mathbf{F} - K\mathbf{U}, \qquad \text{and} \qquad \mathscr{G}(\mathbf{U}, \mathbf{P}) := C(\mathbf{U}) - G\mathbf{P}.$$

As it has been exposed above, the explicit treatment of the convective term implies time stepping restrictions that are given by the condition on the hyperbolic CFL_u number. As the mesh is not homogeneous and the velocity is not the same over all the domain, we can obtain a bound for the maximum CFL_u number taking the maximum value over all the elements. With these parameters, the maximum CFL_u depending on the time step size will be of the order shown in Table III.

Looking at Table III we see that in this case the time step needs to be much smaller than in the previous case if we want to guarantee a stable solution. For the implicit treatment of the convective term, for time step sizes below or equal to $\delta t \leq 1.0 \cdot 10^{-2}$, the method is stable and gives good convergence rates. Here, for this time step we see that we have a $\mathrm{CFL}_u \sim 3.4$, which is grater than the critical ~ 1.0.

In effect, the largest time step size for which all the schemes give stable results for this case is $\delta t = 3.125 \cdot 10^{-4}$. Some schemes also are stable for $\delta t = 6.25 \cdot 10^{-4}$ or even for $\delta t = 1.25 \cdot 10^{-3}$, but no one is stable for $\delta t = 2.5 \cdot 10^{-3}$, which is of the order of the critical time step size. Thus, it

Table III. CFL_u values.

δt	CFL_u
$2.0 \cdot 10^{-2}$	6.81
$1.0 \cdot 10^{-2}$	3.48
$5.0 \cdot 10^{-3}$	1.79
$2.5 \cdot 10^{-3}$	0.91
$1.25 \cdot 10^{-3}$	0.46
$6.25 \cdot 10^{-4}$	0.21
$3.125 \cdot 10^{-4}$	0.11
$1.5625 \cdot 10^{-4}$	0.05
$7.8125 \cdot 10^{-5}$	0.03

is clear that for this type of problems, an explicit time integration of the convective term implies the use of much smaller time steps than for an implicit time integration of this term.

In Table IV we show the benchmark quantities of the cases that attained convergence, till $\delta t = 1.5625 \cdot 10^{-4}$. It is clearly seen that for such small time step sizes all the results are very similar, showing that the IMEX-SRK scheme also gives good results considering the explicit integration of the convective term, whenever the time step is sufficiently small to give a stable result. The results that are out of the range are in italics.

Table IV. Benchmark 2D-2 results with explicit convection.

Scheme	δt	$c_{D_{max}}$	$c_{L_{max}}$	St	ΔP
$(1-1)$	3.125e-04	3.2272	*0.9882*	*0.3059*	2.4862
$(1-1)$	1.563e-04	3.2280	0.9944	*0.3062*	2.4879
$(1-2)$	3.125e-04	3.2296	1.0006	*0.3068*	2.4891
$(1-2)$	1.563e-04	3.2296	1.0006	*0.3065*	2.4893
$(2-2/1)$	6.250e-04	3.2296	1.0006	*0.3065*	2.4891
$(2-2/1)$	3.125e-04	3.2296	1.0006	*0.3065*	2.4895
$(2-2/1)$	1.563e-04	3.2296	1.0006	*0.3065*	2.4893
$(2-2/2)$	6.250e-04	3.2296	1.0006	*0.3065*	2.4892
$(2-2/2)$	3.125e-04	3.2296	1.0006	*0.3065*	2.4896
$(2-2/2)$	1.563e-04	3.2296	1.0006	*0.3065*	2.4893
$(2-3)$	3.125e-04	3.2296	1.0006	*0.3065*	2.4895
$(2-3)$	1.563e-04	3.2296	1.0006	*0.3065*	2.4893
$(3-3)$	6.250e-04	3.2296	1.0006	*0.3065*	2.4891
$(3-3)$	3.125e-04	3.2296	1.0006	*0.3065*	2.4895
$(3-3)$	1.563e-04	3.2296	1.0006	*0.3065*	2.4893
$(4-3)$	6.250e-04	3.2296	1.0006	*0.3065*	2.4891
$(4-3)$	3.125e-04	3.2296	1.0006	*0.3065*	2.4895
$(4-3)$	1.563e-04	3.2296	1.0006	*0.3065*	2.4893
$(5-3)$	6.250e-04	3.2296	1.0006	*0.3065*	2.4891
$(5-3)$	3.125e-04	3.2296	1.0006	*0.3065*	2.4895
$(5-3)$	1.563e-04	3.2296	1.0006	*0.3065*	2.4893
lower bound		3.2200	0.9900	0.2950	2.4600
upper bound		3.2400	1.0100	0.3050	2.5000

In order to compare the efficiency between the implicit and explicit approaches for the time integration of the convective term for this particular test, we perform the convergence analysis for both methods at $t = 8.01$ s. We restrict this test only to the most efficient schemes for each order of approximation, which are *(1-1), (2-2/1)*, and *(3-3)* schemes. The results of the convergence in time for this comparison are shown in Fig. 10. Unstable results for the explicit convective term integration are not plotted in this figure.

The fully implicit versions of the SRK schemes considered here show a good agreement with the prescribed convergence rate. Furthermore, when the time step is sufficiently small, the explicit version seems to give more accurate results. This is especially remarkable for the *(2-2/1)* scheme, where the differences are bigger. Same conclusions can be pointed out for both velocity and pressure fields, since there are not significant differences between their convergence rates, see Figs. 10(a) and 10(b). Fig. 10 shows that the *(3-3)* scheme is the most accurate one. Apart from the *(2-2/1)* scheme,

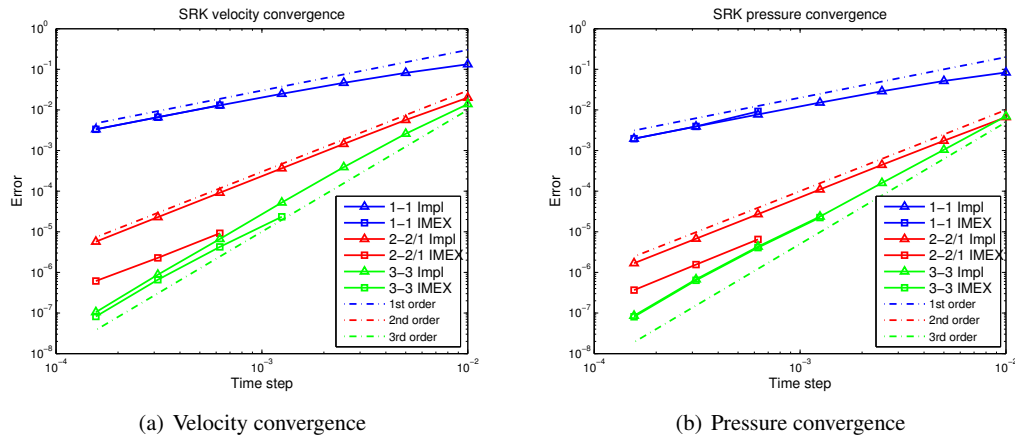(a) Velocity convergence   (b) Pressure convergence

Figure 10. Fully implicit and IMEX-SRK convergence rate comparison.

implicit and explicit versions give similar results when the time step is sufficiently small.

Next, we look at the overall computational time needed for each approach. This is shown in Fig. 11, where the error e_u is plotted against the CPU time for each scheme and for each approach. We see
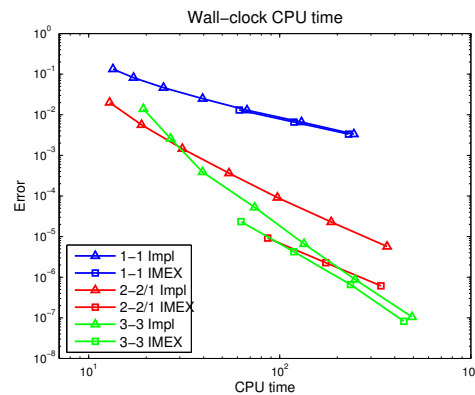

Figure 11. Fully implicit and IMEX-SRK CPU time efficiency comparison.

that, as expected, the approach with explicit time integration of the convective term is more efficient than the implicit one when the time step is sufficiently small. For large time steps, we cannot use the explicit versions since the CFL_u condition is limiting the stability of the method, and therefore its accuracy. Note that there is not a big difference on the efficiency for *(1-1)* and *(3-3)* schemes, since the gain of treating the convective term explicitly is not too much relevant in laminar problem types. As exposed at the beginning of this subsection, explicit approaches involve time step size restrictions, which may be much smaller than the physical time scale. Furthermore, as it has also been exposed before, the nonlinearity of the convective term when it is integrated implicitly does not increase the cost too much. Despite of that, for errors smaller than $\sim 10^{-4}$, the most efficient scheme is the *(3-3)* using explicit time integration of the convective term, at least for this test.

4.3.1. Adaptive time stepping technique When solving generic transient flow problems, it is very useful to use an adaptive time stepping technique that will automatically provide a (dynamic) value of the time step size for a target accuracy. Time stepping techniques allow us to adapt the time step size to the flow conditions, that is to change the time step when the physical scales of the flow change, e.g., transition to turbulence. Adaptive time-stepping techniques have been implemented to satisfy accuracy requirements for the incompressible Navier-Stokes equations and several successful tests can be found in the literature, as it can be shown in [30, 31, 32, 33].

These adaptive time stepping techniques can straightforwardly be applied to SRK methods. In fact, for all multi-step or multi-stage methods like SRK schemes, the implementation of an adaptive time step technique is widely used, since we only need a different evaluation of the final unknown at each step that can be done using a different Butcher tableau, see for instance [34, 35]. Here we use the so called *PI11 controller* by [36] and suggested in [35], which computes the time step size as follows

$$\delta t_{n+1} = \left(\frac{\varepsilon}{r_{n+1}}\right)^{1/k} \left(\frac{r_n}{r_{n+1}}\right)^{1/k} \frac{\delta t_n}{\delta t_{n+1}} \delta t_n,$$

with $\varepsilon = 0.8 \cdot TOL$, where TOL is a given tolerance that we take as $1 \cdot 10^{-6}$ and 0.8 is a safety factor. The local error is $r_{n+1} = \|\mathbf{U} - \hat{\mathbf{U}}\|$ if the error per step (EPS) is controlled or $r_{n+1} = \|(\mathbf{U} - \hat{\mathbf{U}})/\delta t_n\|$ if the error per unit step (EPUS) is controlled. In the former case $k = p + 1$ (EPS) and for the second one (EPUS) $k = p$, p being the order of the time integration scheme which has been used to compute the estimated velocity $\hat{\mathbf{U}}$.

We solve the problem from $t = 8.0$ to $t = 8.4$ using the SRK *(3-3)* scheme, considering the implicit time integration of the convective term. We compare the solutions against the one obtained with the same scheme, but using a fixed time step size $\delta t = 3.125 \cdot 10^{-4}$, the smallest time step considered in previous analysis. Let us note that an adaptive time stepping technique for the explicit time integration of the convective term does not make sense in this case. We see in Table III that the CFL_u number of the SRK with explicit versions of the convective term with a time step of $\delta t = 3.125 \cdot 10^{-4}$ is already above the critical value of 1.0. This means that if we increase the time step size, we will have stability problems due to the hyperbolic CFL condition.

To compute $\hat{\mathbf{U}}$ we have used the 2nd order Butcher tableau referred as *Embedded Formula of Order 2* for the *Third Order Strongly S-Stable Formula* defined in [37], which is 3rd order and corresponds to the *(3-3)* scheme defined in Appendix A. The initial time step size is set to be $\delta t_0 = 1.0 \cdot 10^{-5}$, small enough to get an accurate first solution. In Fig. 12(a) we show the time step evolution for the two different cases considered in this subsection. We see that the time step size for the scheme with adaptive time stepping is increasing with a variable rate and seems to converge to an optimal one. Fig. 12(b) depicts a close up view of the lift coefficient evolution. We

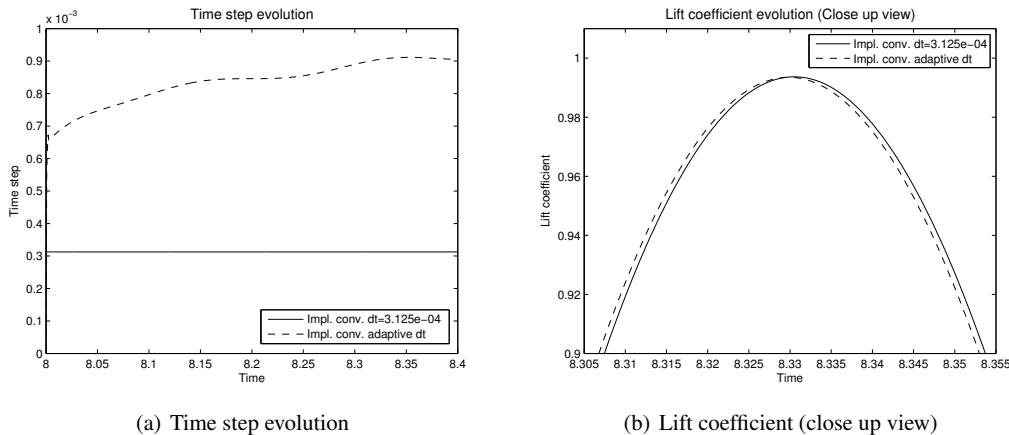(a) Time step evolution

(b) Lift coefficient (close up view)

Figure 12. Adaptive time stepping.

see that for the implicit adaptive time step case, the results are really close to those obtained with the same scheme with a fixed time step of $\delta t = 3.125 \cdot 10^{-4}$. Note that the adaptive time step size are twice or even three times larger than the fixed one. The total elapsed CPU time for the implicit scheme with adaptive time step is 3615 s, while the total time for the implicit scheme with fixed time step is 8453 s, which supposes a reduction of a 42.8% of time consumption with a very little difference in the result.

4.4. Taylor-Green vortex flow

The next step of this work is to check the performance of the SRK methods for turbulent incompressible flows. The use of IMEX-SRK methods is usually favoured for turbulent flows, since the time step size required to satisfy the CFL_u condition is of the same order as the physical one needed for accuracy purposes. The Taylor-Green vortex (TGV) problem is a typical and widely used problem in turbulence numerical simulations, in which we can see the basic turbulence decay mechanisms in a relatively simple flow. Here, the computational domain is the cube $(0, 2\pi)^3$ with periodical boundary conditions. The initial analytical condition for this problem is given by (see, e.g., [38])

$$\mathbf{u}(x,y,z,0) = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} = \begin{pmatrix} u_0 \cos(x) \sin(y) \sin(z) \\ -u_0 \sin(x) \cos(y) \sin(z) \\ 0 \end{pmatrix} \qquad (30)$$

$$p(x,y,z,0) = p_0 + \frac{1}{16} \left(\cos(2x) + \cos(2y) \right) \left(\cos(2z) + 2 \right),$$

with

$$u_0 = \frac{2}{\sqrt{3}} \sin\left(\gamma + \frac{2\pi}{3} \right).$$

We choose $\gamma = 0$, which gives the mean initial velocity $u_0 = 1$. We solve the TGV problem using a Reynolds number $\mathrm{Re} = 1600$, but in the literature the same test using different Reynolds numbers (e.g., $\mathrm{Re} = 800$ and $\mathrm{Re} = 3000$) can be found (see, e.g., [39, 40]).

The problem is solved in a 64^3 $Q2 - Q1$ elements mesh, and no additional sub-grid modelling is being used. Our concern is about the time integration, therefore the spatial accuracy does not take a crucial role in this work. Anyway, we analyze some physical quantities, like the global kinetic energy or the kinetic energy dissipation rate, that are typically used for this test to calibrate different methods.

We only use the IMEX-SRK method with the convection treated explicitly, avoiding the need of nonlinear iterations. This situation is given by defining the operators $\mathscr{F}$ and $\mathscr{G}$ in Eqs. (23) and (24) as

$$\mathscr{F}(\mathbf{U}) := -K\mathbf{U}, \qquad \text{and} \qquad \mathscr{G}(\mathbf{U}, \mathbf{P}) := \mathbf{F} - C(\mathbf{U})\mathbf{U} - G\mathbf{P}.$$

First of all, we perform a time step convergence analysis at the beginning of the simulation, where the flow is still laminar. This convergence analysis consists in solving the problem from $t = 0.0$ to $t = 0.1$ for all the schemes proposed in Appendix A for several time step sizes. In particular, we solve the problem with four different time step sizes ($\delta t = \{0.1, 0.05, 0.025, 0.0125\}$) and we compare the solution against the one obtained with the *(5-3)* scheme with a time step equal to $\delta t = 6.25 \cdot 10^{-3}$. The L^2-norm of the kinetic energy error compared against the reference kinetic energy solution given by a DNS computation can be found in [41]. Our approach is to compute the L^∞-norm of the solution, comparing against a solution computed with a finer time step, but with the same spatial discretization. With this approach we are eliminating the spatial error and we are using a more restrictive error norm.

We show in Fig. 13 the order of convergence in time for both velocity (Fig. 13(a)) and pressure (Fig. 13(b)) fields. We can see that the order of convergence in this regime follows the predicted rate for almost all schemes; the *(5-3)* schemes seems to converge with a higher rate than the expected one, being this performance especially remarkable for the velocity field convergence.
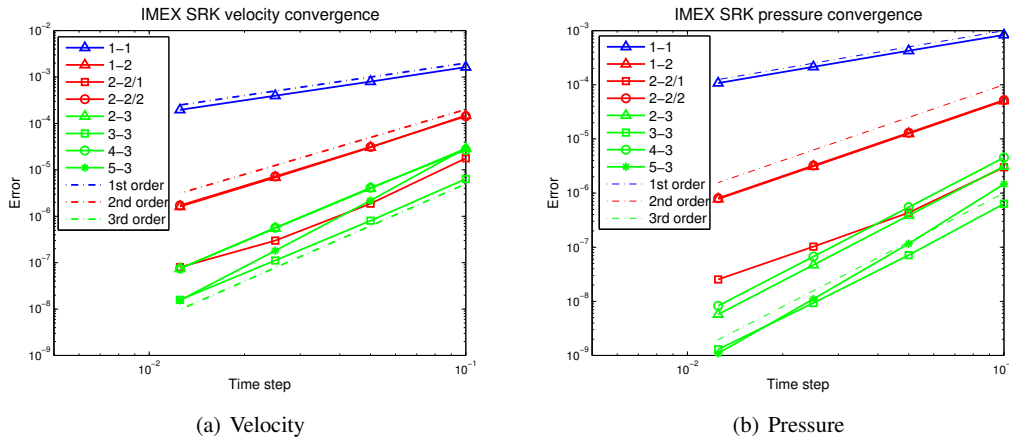
Figure 13. IMEX-SRK convergence for the laminar regime.

Another point to be highlighted in Fig. 13 is that the *(2-2/1)* scheme shows a much lower error compared to the other second-order schemes. This performance is also observed on the previous tests when an IMEX-SRK (with explicit treatment of the convection) is used.

As it has been done for the 2D laminar flow around a cylinder test (see Subsection 4.3) we analyze the efficiency of the methods comparing error against CPU time. This comparison is in Fig. 14, where the error is plotted in terms of the averaged elapsed CPU time per processor needed to complete the simulation from $t = 0$ to $t = 0.1$.
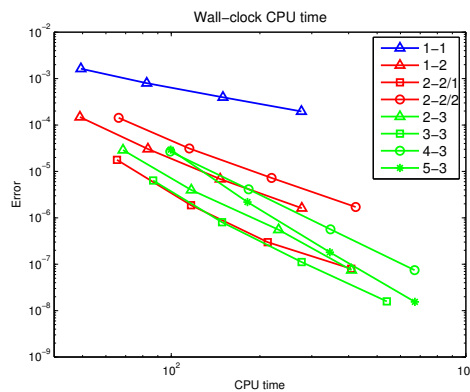

Figure 14. IMEX-SRK efficiency for the laminar regime.

It is seen in Fig. 14 that the most efficient schemes are the *(2-2/1)* and *(3-3)* ones, which for errors greater than $\sim 10^{-6}$ give the same error-CPU time ratio. As it is natural, below this threshold, the second-order scheme *(2-2/1)* starts loosing efficiency and the *(3-3)* scheme becomes the most efficient one for errors smaller than $\sim 10^{-6}$.

Once we have determined that the most efficient schemes for this test are the *(2-2/1)* and *(3-3)* ones, we now want to see their performance for a larger time interval. The aim is to solve the problem until the turbulence becomes fully developed, which for this Reynolds number case takes place around $t \sim 9$. Following [38], the time interval will go from $t = 0.0$ to $T = 10.0$, so we will be able to compare the results with their DNS. For this computation, we select the time step size that has an error at $t = 0.1$ in Fig. 13(a) of the order $\sim 1 \cdot 10^{-6}$, i.e., $\delta t = 5.0 \cdot 10^{-2}$ for the *(3-3)* scheme and $\delta t = 3.6 \cdot 10^{-2}$ for the *(2-2/1)* scheme.

Fig. 15 depicts the total energy evolution (Fig. 15(a)) and the kinetic energy dissipation rate of the resolved scales (Fig. 15(b)) compared against the DNS provided by [38]. The result is exactly the

same for both schemes, despite the fact that the third-order scheme uses a larger time step. Looking at Fig. 15(a), we see that the energy evolution of the solution is not far from the DNS, but there is a gap after $t = 6$, when the turbulence is developed. This is caused by the lack of any turbulent model which would capture the small scales proper dissipation. For the same reason we see big differences between our solutions and the DNS results in Fig. 15(b) after $t = 6$.

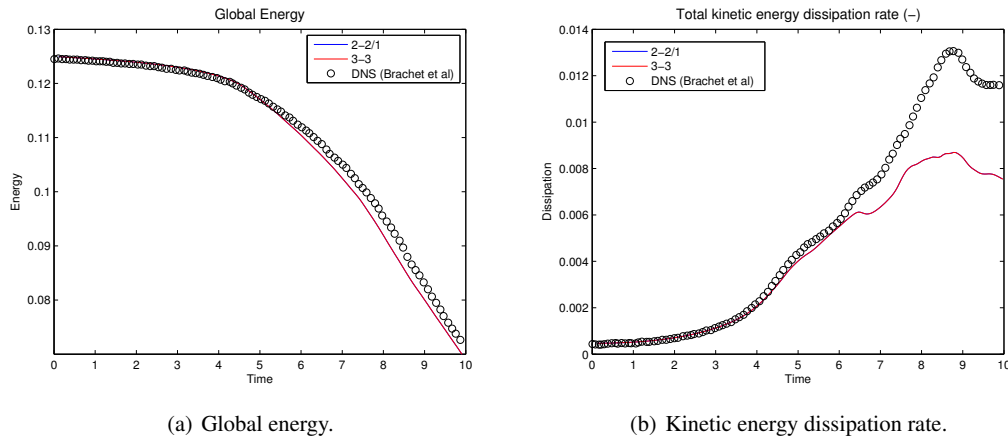(a) Global energy.   (b) Kinetic energy dissipation rate.

Figure 15. Global energy and Kinetic energy dissipation rate evolution (blue line below red line).

Looking at Fig. 15(b) we clearly see that the error given by the spatial discretization when the flow becomes turbulent is too large to appreciate significant differences between time integration schemes. We want to highlight here that an adaptive time step technique could be used in order to efficiently solve the transient problem. However, the use of an adaptive time stepping technique without an accurate solution in space does not make sense. Future work in this direction involves the introduction of LES models within the IMEX-SRK approach proposed in this work.

5. CONCLUSIONS

The segregated Runge-Kutta methods proposed in this work enjoy two nice features, namely the velocity and pressure segregation at the time integration level (without the need to perform additional fractional step techniques that spoil high orders of accuracy) and the provable same order of accuracy for both velocities and pressures. These methods have been motivated as an implicit-explicit Runge-Kutta time integration of the projected Navier-Stokes system onto the discrete divergence-free space. The terms in this system that involve the inverse of a discrete Laplacian $DM^{-1}G$ are treated explicitly in all cases, in order to make the resulting method numerically feasible. Viscous and convection terms can be treated using implicit, combined implicit/explicit, or fully explicit schemes, leading to implicit, IMEX, or explicit SRK schemes, respectively. The pressure can be recovered by using a discrete pressure Poisson equation, and the SRK scheme can be finally recast in a velocity-pressure formulation.

Explicit SRK methods have been proved to be equivalent to existing half-explicit RK methods. Further, these methods exactly satisfy the divergence-constraint equation in most situations of interest; in all cases when weakly enforcing Dirichlet boundary conditions, and for fixed or at most a point-wise p-th order polynomial variation in time (for a p-th order method) of the Dirichlet data for a strong enforcement. Further, it is easy to check that the error for the pressure is of the same order as the one for the velocity.

We have performed a wide set of numerical experiments to evaluate the segregated Runge-Kutta algorithms; first to third order schemes have been implemented and analyzed. They include convergence tests for problems with manufactured solutions for time-dependent Dirichlet boundary data. This way, we can also evaluate the well-known order reduction effect of RK methods. We

have also performed numerical tests for the laminar flow around a cylinder (evaluating drag and lift coefficients) and the turbulent Taylor-Green vortex flow. The different methods have been compared in terms of CPU cost for a target error. Fully implicit, implicit viscous/explicit convective, and fully explicit methods have been evaluated, considering their respective CFL conditions. Fully implicit SRK schemes have shown a remarkably strong stability and high accuracy till about 100 times the explicit CFL condition. Further, segregated Runge-Kutta schemes with adaptive time stepping have been proposed and analyzed numerically.

The use of SRK schemes is very appealing for large scale computations of incompressible flows, since the monolithic indefinite system is replaced by segregated positive-definite velocity and pressure blocks. The pressure block involves a Poisson solver, whereas the velocity block is a vector-Laplacian or elasticity matrix when the convective term is treated explicitly. Massively parallel solvers for these problems can be found in the literature (see, e.g., [14]) and are at the user's disposal, e.g., within the FEMPAR scientific computing software.

APPENDIX A. BUTCHER TABLEAUS

We define different settings of the Butcher tableaus, depending on the stages and the order of accuracy of the time integration scheme. For the DIRK tableaus we use the following notation:

$$
\frac{\mathbf{c} \mid A}{\mid \mathbf{b}} =
\begin{array}{c|ccccc}
c_1 & a_{11} & 0 & ... & 0 \\
c_2 & a_{21} & a_{22} & ... & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
c_s & a_{s1} & a_{s2} & ... & a_{ss} \\
\hline
 & b_1 & b_2 & ... & b_s
\end{array}
$$

with the convention $c_i = \sum_{j=1}^{s} a_{ij}$.

In the explicit case the tables are similar, but with zeros on the diagonal. A r-stage explicit scheme can be written as follows.

$$
\frac{\hat{\mathbf{c}} \mid \hat{A}}{\mid \hat{\mathbf{b}}} =
\begin{array}{c|cccccc}
0 & 0 & 0 & 0 & ... & 0 \\
\hat{c}_2 & \hat{a}_{21} & 0 & 0 & ... & 0 \\
\hat{c}_3 & \hat{a}_{31} & \hat{a}_{32} & 0 & ... & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\hat{c}_r & \hat{a}_{r1} & \hat{a}_{r2} & \hat{a}_{r3} & ... & 0 \\
\hline
 & \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & ... & \hat{b}_r
\end{array}
$$

Note that in general the number of implicit stages (s) are not the same than the explicit ones (r). In fact, usually we have one additional stage for the explicit Butcher Tableau ($r = s + 1$). Then, in order to facilitate the IMEX schemes implementation, we pad the implicit s-stage scheme with zeroes, obtaining a new tableau with the same size than the explicit one. We can easily see that with this procedure we will always have $\mathbf{U}_1 = \mathbf{U}_n$ since the first rows for the explicit and the implicit tableaus are filled with zeros. Given that, no computation is needed at the first stage, so we will say that the IMEX scheme has s stages when the associated Butcher tableaus are of the size $(s+1)$.

We select from the literature some Butcher table settings for IMEX RK schemes that have been already tested for convection-diffusion type problems, mostly from [42], but also from [26]. It is important to keep in mind that the properties described below are given for the ODE system in time that arises from the convection-diffusion equation and may not be preserved for an index 2 DAE system like the semi-discrete incompressible Navier-Stokes problem.

- **1 stage, 1st order (1-1):** This is the so called *Forward-Backward Euler* and was proposed by Ascher *et al.* in [42]. There are two different versions of this scheme, one that satisfies $\hat{\mathbf{b}} = \mathbf{b}$ and another that does not satisfy this condition. We use the first version, just to be consistent with the following schemes, although it requires an extra evaluation per step of the explicit

counterpart. Note that this scheme has stability problems when the implicit operator is zero. Furthermore, the choice $\hat{\mathbf{b}} = \mathbf{b}$ preserves the incompressibility constraint under the conditions stated in Proposition 3.2.

Implicit:

0	0	0
1	0	1
	0	1

Explicit:

0	0	0
1	1	0
	0	1

- **1 stage, 2nd order (1-2):** This scheme was also proposed in [42], where they called it as *Implicit-explicit midpoint*. It is second-order accurate and as stated in [42] performs comparably to the popular Crank-Nicolson Adams-Bashforth method, with the addition that it has better symmetry properties. It is important to remark that it does not have any attenuation at the stiffness limit.

Implicit:

0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$
	0	1

Explicit:

0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0
	0	1

- **2 stages, 2nd order, case 1 (2-2/1):** Another second-order IMEX RK scheme was proposed in [42], but in this case it has two stages. This scheme turns out to be *L*-stable and it is based on a stiffly accurate two-stage, second-order DIRK scheme.

Implicit:

0	0	0	0
γ	0	γ	0
1	0	$1-\gamma$	γ
	0	$1-\gamma$	γ

Explicit:

0	0	0	0
γ	γ	0	0
1	δ	$1-\delta$	0
	0	$1-\gamma$	γ

with $\gamma = (2-\sqrt{2})/2$ and $\delta = -2\sqrt{2}/3$.

- **2 stages, 2nd order, case 2 (2-2/2):** An alternative definition of the previous scheme is also proposed in [42], where the explicit table is defined satisfying the condition $\hat{b}_j = \hat{a}_{sj}$ for all $j = 1,...,s$ instead of satisfying that $\hat{\mathbf{b}} = \mathbf{b}$. The resulting scheme is also *L*-stable and second-order accurate in time. Note that this scheme will not satisfy the incompressibility constrain under the assumptions exposed in Proposition 3.2.

Implicit:

0	0	0	0
γ	0	γ	0
1	0	$1-\gamma$	γ
	0	$1-\gamma$	γ

Explicit:

0	0	0	0
γ	γ	0	0
1	δ	$1-\delta$	0
	δ	$1-\delta$	0

with $\gamma = (2-\sqrt{2})/2$ and $\delta = 1 - 1/(2\gamma)$.

- **2 stages, 3rd order (2-3):** A two-stage combination, third-order accurate in time IMEX scheme is defined in [42]. In this case, the implicit table is based on a two-stage, third-order DIRK scheme.

Implicit:

0	0	0	0
γ	0	γ	0
$1-\gamma$	0	$1-2\gamma$	γ
	0	$\frac{1}{2}$	$\frac{1}{2}$

Explicit:

0	0	0	0
γ	γ	0	0
$1-\gamma$	$\gamma-1$	$2(1-\gamma)$	0
	0	$\frac{1}{2}$	$\frac{1}{2}$

with $\gamma = (3+\sqrt{3})/6$.

- **3 stages, 3rd order (3-3):** A stiffly accurate third-order, three-stage DIRK scheme combined with a four-stage third-order explicit RK scheme is proposed in [42]. The resulting scheme is third-order accurate in time and *L*-stable.

Implicit:

0	0	0	0	0
0.4358665215	0	0.4358665215	0	0
0.7179332608	0	0.2820667392	0.4358665215	0
1	0	1.208496649	−0.644363171	0.4358665215
	0	1.208496649	−0.644363171	0.4358665215

Explicit:

0	0	0	0	0
0.4358665215	0.4358665215	0	0	0
0.7179332608	0.3212788860	0.3966543747	0	0
1	−0.105858296	0.5529291479	0.5529291479	0
	0	1.208496649	−0.644363171	0.4358665215

- **4 stages, 3rd order (4-3):** Finally, it is propoed in [42] a four-stage third-order accurate in time IMEX scheme.

Implicit:

0	0	0	0	0	0
1/2	0	1/2	0	0	0
2/3	0	1/6	1/2	0	0
1/2	0	−1/2	1/2	1/2	0
1	0	3/2	−3/2	1/2	1/2
	0	3/2	−3/2	1/2	1/2

Explicit:

0	0	0	0	0	0
1/2	1/2	0	0	0	0
2/3	11/18	1/18	0	0	0
1/2	5/6	−5/6	1/2	0	0
1	1/4	7/4	3/4	−7/4	0
	1/4	7/4	3/4	−7/4	0

- **5 stages, 3rd order (5-3):** Boscarino developed in [26] a third-order five-stage IMEX RK scheme that does not suffer from order reduction phenomena when stiff problems containing a non-stiff term and a stiff term are solved. Note that in this case we say that the method has five stages because the first column of the implicit table is not zero, although the first row is still full of zeros so the first stage will give $\mathbf{U}_1 = \mathbf{U}_n$.

Implicit:

0	0	0	0	0	0
c_2	γ	γ	0	0	0
c_3	a_{31}	a_{32}	γ	0	0
c_4	a_{41}	a_{42}	a_{43}	γ	0
1	b_1	b_2	b_3	b_4	γ
	b_1	b_2	b_3	b_4	γ

Explicit:

0	0	0	0	0	0
c_2	γ	0	0	0	0
c_3	$\hat{a}_{31}$	$\hat{a}_{32}$	0	0	0
c_4	$\hat{a}_{41}$	$\hat{a}_{42}$	$\hat{a}_{43}$	0	0
1	$\hat{a}_{51}$	$\hat{a}_{52}$	$\hat{a}_{53}$	$\hat{a}_{54}$	0
	b_1	b_2	b_3	b_4	γ

The parameter values can be found in the appendix of [26].

ACKNOWLEDGMENTS

REFERENCES

1. Kim J, Moin P. Application of a fractional-step method to incompressible Navier-Stokes equations. *Journal of Computational Physics* 1985; **59**(2):308–323.
2. Le H, Moin P. An improvement of fractional step methods for the incompressible Navier-Stokes equations. *Journal of Computational Physics* 1991; **92**(2):369–379.

3. Codina R, Badia S. On some pressure segregation methods of fractional-step type for the finite element approximation of incompressible flow problems. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**(23-24):2900–2918.

4. Badia S, Codina R. Algebraic pressure segregation methods for the incompressible Navier-Stokes equations. *Archives of Computational Methods in Engineering* 2008; **15**(3):343–369.

5. Hairer E, Lubich C, Roche M. *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*, vol. 1409. Springer Verlag, 1989.

6. de Swart JJ, Söderlind G. On the construction of error estimators for implicit Runge-Kutta methods. *Journal of Computational and Applied Mathematics* 1997; **86**(2):347–358.

7. Marx YP. Time integration schemes for the unsteady incompressible Navier-Stokes equations. *Journal of Computational Physics* 1994; **112**(1):182–209.

8. Nikitin N. Third-order-accurate semi-implicit Runge-Kutta scheme for incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids* 2006; **51**(2):221–233.

9. Knikker R. Study of a staggered fourth-order compact scheme for unsteady incompressible viscous flows. *International Journal for Numerical Methods in Fluids* 2009; **59**(10):1063–1092.

10. Kampanis NA, Ekaterinaris JA. A staggered grid, high-order accurate method for the incompressible Navier-Stokes equations. *Journal of Computational Physics* 2006; **215**(2):589–613.

11. Sanderse B, Koren B. Accuracy analysis of explicit Runge-Kutta methods applied to the incompressible Navier-Stokes equations. *Journal of Computational Physics* 2012; **231**(8):3041–3063.

12. Hairer E, Wanner G. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer, 1993.

13. Sanderse B. Energy-conserving Runge-Kutta methods for the incompressible Navier-Stokes equations. *Journal of Computational Physics* 2013; **233**:100–131.

14. Badia S, Martín AF, Principe J. A highly scalable parallel implementation of balancing domain decomposition by constraints. *SIAM Journal on Scientific Computing* 2014; :C190–C218.

15. Elman HC, Silvester DJ, Wathen AJ. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Oxford University Press, 2005.

16. Brasey V, Hairer E. Half-explicit Runge-Kutta methods for differential-algebraic systems of index 2. *SIAM J. Numer. Anal.* 1993; **30**(2):538–552.

17. Arnold M. Half-explicit Runge-Kutta methods with explicit stages for differential-algebraic systems of index 2. *BIT Numerical Mathematics* 1998; **38**(3):415–438.

18. Murua A. Partitioned half-explicit Runge-Kutta methods for differential-algebraic systems of index 2. *Computing* 1997; **59**(1):43–61.

19. Verstappen RWCP, Veldman AEP. Symmetry-preserving discretization of turbulent flow. *Journal of Computational Physics* 2003; **187**(1):343–368.

20. Vreman AW, Kuerten JGM. Comparison of direct numerical simulation databases of turbulent channel flow at $Re_\tau = 180$. *Physics of Fluids (1994-present)* 2014; **26**(1):015 102.

21. Badia S, Martín AF, Principe J. Implementation and scalability analysis of balancing domain decomposition methods. *Archives of Computational Methods in Engineering* 2013; **20**(3):239–262.

22. Badia S, Martín AF, Principe J. On an overlapped coarse/fine implementation of balancing domain decomposition with inexact solvers. *Submitted* 2014; .

23. Hairer E, Nrsett SP, Wanner G. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer Science & Business Media, 2008.

24. Sanz-Serna JM, Verwer JG, Hundsdorfer WH. Convergence and order reduction of Runge-Kutta schemes applied to evolutionary problems in partial differential equations. *Numerische Mathematik* 1986; **50**(4):405–418.

25. Verwer JG. Convergence and order reduction of diagonally implicit Runge-Kutta schemes in the method of lines. *Numerical Analysis: Proceedings of the Dundee Conference on Numerical Analysis, 1975, Pitman Research Notes in Mathematics*, vol. 140, 1986; 220–237.

26. Boscarino S. On an accurate third order implicit-explicit Runge-Kutta method for stiff problems. *Applied Numerical Mathematics* 2009; **59**(7):1515–1528.

27. Ethier CR, Steinman DA. Exact fully 3d Navier-Stokes solutions for benchmarking. *International Journal for Numerical Methods in Fluids* 1994; **19**(5):369–375.

28. Schäfer M, Turek S, Durst F, Krause E, Rannacher R. Benchmark computations of laminar flow around a cylinder. *Flow Simulation with High-Performance Computers II*, Hirschel PDEH (ed.). No. 48 in Notes on Numerical Fluid Mechanics (NNFM), Vieweg+Teubner Verlag, 1996; 547–566.

29. Brezzi F, Hughes TJ, Süli E. Variational approximation of flux in conforming finite element methods for elliptic partial differential equations: a model problem. *Atti della Accademia Nazionale dei Lincei. Classe di Scienze Fisiche, Matematiche e Naturali. Rendiconti Lincei. Matematica e Applicazioni* 2001; **12**(3):159–166.

30. Feng Y, Peri D. A time-adaptive space-time finite element method for incompressible lagrangian flows with free surfaces: computational issues. *Computer Methods in Applied Mechanics and Engineering* 2000; **190**(5-7):499–518.

31. John V, Rang J. Adaptive time step control for the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(9-12):514–524.

32. Kay DA, Gresho PM, Griffiths DF, Silvester DJ. Adaptive time-stepping for incompressible flow part II: Navier-Stokes equations. *SIAM Journal on Scientific Computing* 2010; **32**(1):111–128.

33. Veneziani A, Villa U. ALADINS: An ALgebraic splitting time ADaptive solver for the incompressible Navier-Stokes equations. *Journal of Computational Physics* 2013; **238**:359–375.

34. Gustafsson K. Control theoretic techniques for stepsize selection in explicit Runge-Kutta methods. *ACM Trans. Math. Softw.* 1991; **17**(4):533–554.

35. Gustafsson K. Control-theoretic techniques for stepsize selection in implicit Runge-Kutta methods. *ACM Trans. Math. Softw.* 1994; **20**(4):496–517.

36. Söderlind G. Automatic control and adaptive time-stepping. *Numerical Algorithms* 2002; **31**(1-4):281–310.
37. Cash JR. Diagonally implicit Runge-Kutta formulae with error estimates. *IMA Journal of Applied Mathematics* 1979; **24**(3):293–301.
38. Brachet ME. Direct simulation of three-dimensional turbulence in the taylorgreen vortex. *Fluid Dynamics Research* 1991; **8**(1-4):1–8.
39. Gassner GJ, Beck AD. On the accuracy of high-order discretizations for underresolved turbulence simulations. *Theoretical and Computational Fluid Dynamics* 2013; **27**(3-4):221–237.
40. Hickel S, Adams NA, Domaradzki JA. An adaptive local deconvolution method for implicit LES. *Journal of Computational Physics* 2006; **213**(1):413–436.
41. Boom P, Zingg D. Time-accurate flow simulations using an efficient newton-krylov-schur approach with high-order temporal and spatial discretization. *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics, 2013.
42. Ascher UM, Ruuth SJ, Spiteri RJ. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics* 1997; **25**(2-3):151–167.