High-order implicit time integration for unsteady incompressible flows

A. Montlaur[1,2], S. Fernandez-Mendez[1,3] and A. Huerta[1,3,*,†]

[1] *Laboratori de Càlcul Numèric (LaCàN), Universitat Politècnica de Catalunya - BarcelonaTech, Jordi Girona 1-3, 08034 Barcelona, Spain*
[2] *Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels, Universitat Politècnica de Catalunya, Spain*
[3] *E.T.S. d'Enginyers de Camins, Canals i Ports de Barcelona, Universitat Politècnica de Catalunya, Spain*

SUMMARY

The spatial discretization of unsteady incompressible Navier–Stokes equations is stated as a system of differential algebraic equations, corresponding to the conservation of momentum equation plus the constraint due to the incompressibility condition. Asymptotic stability of Runge–Kutta and Rosenbrock methods applied to the solution of the resulting index-2 differential algebraic equations system is analyzed. A critical comparison of Rosenbrock, semi-implicit, and fully implicit Runge–Kutta methods is performed in terms of order of convergence and stability. Numerical examples, considering a discontinuous Galerkin formulation with piecewise solenoidal approximation, demonstrate the applicability of the approaches and compare their performance with classical methods for incompressible flows.

KEY WORDS: differential algebraic equations; incompressible Navier–Stokes; high-order time integrators; Runge–Kutta; Rosenbrock; discontinuous Galerkin

1. INTRODUCTION

Because of constraints of computing costs, in the past, development of numerical techniques for flow simulations has focused mainly on steady state calculations. However, many physical phenomena of interest are inherently unsteady, creating the need for efficient numerical formulations for unsteady problems, a few examples being separated flows, wake flows, fluid actuators, and maneuvering. Good stability properties and high orders of accuracy in time as well as in space are critical requirements, especially when studying boundary layers, high Reynolds number flows, or flows with high vorticity.

An important difficulty for the numerical simulation of incompressible flows is that velocity and pressure are coupled by the incompressibility constraint. Interest in using projection methods to overcome this difficulty in time-dependent viscous incompressible flows started with the introduction of fractional-step methods for the incompressible Navier–Stokes equations [1, 2]. Following the original ideas of Chorin and Temam, numerous authors have successfully used fractional-step methods for incompressible flows (see, for instance, [3–6]). The pressure/incompressibility terms have to be treated implicitly, whereas the remaining terms, viscous and convective, can be treated either explicitly, semi-implicitly, or fully implicitly. Nevertheless, although explicit schemes are used at much lower cost, the number of realistic problems that are amenable to explicit formulation is very small. In common situations, large variations in element size, required to solve multiple spatial scales occurring in high Reynolds number flow or in boundary layers, make the use of explicit

[*] Correspondence to: A. Huerta, Departament de Matemàtica Aplicada III, E.T.S. Ingenieros de Caminos, Universitat Politècnica de Catalunya, Jordi Girona 1, E-08034 Barcelona, Spain.
[†] E-mail: antonio.huerta@upc.edu

time-integration techniques impractical. In such cases, implicit schemes have to be considered such as implicit fractional step methods, Crank–Nicolson (CN) [7], or generalized-α methods [8]. Unfortunately, these classical methods for incompressible flow are at most second-order accurate in time.

By contrast, high-order time integrators are widely used for compressible flows, such as backward difference multistep methods [9] or high-order Runge–Kutta (RK) methods. In particular, it is well known that for high-order accurate computations, RK methods present two major advantages in front of multistep methods: larger stability regions and straightforward implementation of variable time step. Thus, high-order RK methods have been successfully applied to compressible flow problems, whose spatial discretization (for example, with finite elements or finite volumes) leads to a system of ODEs [10, 11]. Finally, another alternative to solve the resulting ODE system is to use mixed explicit/implicit time-integration schemes. These schemes, using an explicit advection and implicit diffusion, exhibit much broader stability regions compared, for example, with Adams family schemes, typically used in splitting methods [12].

In this work, the possibilities of using high-order implicit RK (IRK) methods, as well as an alternative to RK methods, the Rosenbrock methods, also called Kaps and Rentrop methods [13], for incompressible flow computations are explored. To that end, the space discretization of the unsteady incompressible Navier–Stokes equations is interpreted as an index-2 system of differential algebraic equations (DAE) [14]; that is, a system of ODEs corresponding to the conservation of momentum equation plus algebraic constraints corresponding to the incompressibility condition. This interpretation has already been considered in [15, 16] for the implementation of third-order and fifth-order implicit RK methods and in [17] for Rosenbrock.

This paper proposes and thoroughly analyzes a set of time-integration methods especially suited for incompressible flows. A critical and general comparison, in terms of accuracy and stability, of semi-implicit and fully implicit RK and Rosenbrock methods is performed: the orders of convergence of these methods for index-2 DAEs are recalled, and a stability analysis for the solution of the unsteady incompressible Navier–Stokes equations is presented. Among the proposed methods, specific Rosenbrock and RK methods are recommended for incompressible flow computations, with both unconditional stability and high-order accuracy. Furthermore, this paper shows that these recommended methods also stand out from an efficiency point of view when compared with standard methods, such as CN.

The paper is structured as follows. In Section 2, the discontinuous Galerkin (DG) interior penalty method (IPM) proposed in [18, 19] for the steady Stokes and Navier–Stokes equations is extended to the solution of the unsteady Navier–Stokes equations. Section 3 then recalls the basic concepts of implicit RK and Rosenbrock methods for the solution of index-2 DAEs, motivated by their application to the solution of incompressible flow problems. The methodology proposed in [20, 21] is then particularized to the stability analysis of RK and Rosenbrock methods applied to the solution of index-2 DAEs. This index-2 DAE matrix structure arises from the previous spatial discretization of the incompressible Navier–Stokes equations. Numerical examples are presented in Section 4. They show the applicability of the proposed methods, compare accuracy and relative computational cost of RK and Rosenbrock methods for index-2 DAEs with a classical CN scheme, and allow to recommend specific efficient and highly accurate time-integration methods.

2. DISCONTINUOUS GALERKIN FORMULATION FOR THE UNSTEADY INCOMPRESSIBLE NAVIER–STOKES PROBLEM

Let $\Omega \subset \mathbb{R}^{\mathtt{n}_{\mathtt{sd}}}$ be an open-bounded domain, with boundary $\partial\Omega$ and $\mathtt{n}_{\mathtt{sd}}$ the number of spatial dimensions. The strong form of the unsteady incompressible Navier–Stokes problem can be written as

$$\frac{\partial \boldsymbol{u}}{\partial t} - 2\nabla\cdot(\nu\nabla^{\mathrm{s}}\boldsymbol{u}) + \nabla p + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = \boldsymbol{f} \qquad \text{in } \Omega\times]0,T[, \tag{1a}$$

$$\nabla\cdot\boldsymbol{u} = 0 \qquad \text{in } \Omega\times]0,T[, \tag{1b}$$

$$\boldsymbol{u} = \boldsymbol{u}_D \qquad \text{on } \Gamma_D \times]0,T[, \tag{1c}$$

$$-p\boldsymbol{n} + 2\nu(\boldsymbol{n}\cdot\nabla^{\mathrm{s}})\boldsymbol{u} = \boldsymbol{t} \qquad \text{on } \Gamma_N \times]0,T[, \tag{1d}$$

$$\boldsymbol{u}(\boldsymbol{x},0) = \boldsymbol{u}_0(\boldsymbol{x}) \qquad \text{in } \Omega, \tag{1e}$$

where $\partial\Omega = \overline{\Gamma}_D \cup \overline{\Gamma}_N$, $\Gamma_D \cap \Gamma_N = \emptyset$, $\boldsymbol{f} \in \mathcal{L}_2(\Omega)$ is a source term, $\boldsymbol{t}$ the boundary tractions, $\boldsymbol{u}$ the flux velocity, p its pressure, ν the kinematic viscosity, and $\nabla^{\mathrm{s}} = \frac{1}{2}(\nabla + \nabla^T)$. Note that, in Equation (1a), the constant density has been absorbed into the pressure and, in Equation (1e), the initial velocity field $\boldsymbol{u}_0$ is assumed solenoidal.

The discretization of problem (1) following a DG interior penalty formulation [18,19] is presented in this section. To that purpose, suppose that Ω is partitioned in $\mathtt{n}_{\mathtt{el}}$ disjoint subdomains Ω_i, with piecewise linear boundaries $\partial\Omega_i$, which define an internal interface Γ. The *jump* $[\![\cdot]\!]$ and *mean* $\{\cdot\}$ operators are defined along the interface Γ using values from the elements to the left and to the right of the interface (say, Ω_i and Ω_j) and are also extended along the exterior boundary (only values in Ω are employed), namely

$$[\![\circledcirc]\!] = \begin{cases} \circledcirc_i + \circledcirc_j & \text{on } \Gamma, \\ \circledcirc & \text{on } \partial\Omega, \end{cases} \quad \text{and} \quad \{\circledcirc\} = \begin{cases} \kappa_i \circledcirc_i + \kappa_j \circledcirc_j & \text{on } \Gamma, \\ \circledcirc & \text{on } \partial\Omega. \end{cases}$$

Usually $\kappa_i = \kappa_j = 1/2$, but in general, these two scalars are only required to verify $\kappa_i + \kappa_j = 1$; see, for instance, [18,22,23] for more details on the jump and mean definitions.

The following discrete finite element spaces are also introduced

$$\begin{aligned} \boldsymbol{\mathcal{V}}^h &= \{\boldsymbol{v} \in [\boldsymbol{\mathcal{L}}_2(\Omega)]^{\mathtt{n}_{\mathtt{sd}}}; \boldsymbol{v}|_{\Omega_i} \in [\mathcal{P}^k(\Omega_i)]^{\mathtt{n}_{\mathtt{sd}}} \quad \forall \Omega_i\} \\ \mathcal{Q}^h &= \{q \in [\mathcal{L}_2(\Omega)]; q|_{\Omega_i} \in [\mathcal{P}^{k-1}(\Omega_i)] \quad \forall \Omega_i\} \end{aligned}$$

where $\mathcal{P}^k(\Omega_i)$ is the space of polynomial functions of degree at most $k \geqslant 1$ in Ω_i. Finally, in the following equations, $(\cdot,\cdot)$ and $(\cdot,\cdot)_\Upsilon$ respectively denote the $\mathcal{L}_2$ scalar products in Ω and in $\Upsilon \subset \Gamma \cup \partial\Omega$ [18].

In [19], an IPM was derived for the steady Navier–Stokes equations. Its extension for an unsteady formulation becomes the following: find $\boldsymbol{u}_h \in \boldsymbol{\mathcal{V}}^h \times]0,T[$ and $p_h \in \mathcal{Q}^h \times]0,T[$ such that $\forall\ \boldsymbol{v} \in \boldsymbol{\mathcal{V}}^h$, $\forall\ q \in \mathcal{Q}^h$ and $\forall\ t \in]0,T[$

$$\begin{cases} \left(\dfrac{\partial \boldsymbol{u}_h}{\partial t}, \boldsymbol{v}\right) + a(\boldsymbol{u}_h,\boldsymbol{v}) + c(\boldsymbol{u}_h;\boldsymbol{u}_h,\boldsymbol{v}) + b(\boldsymbol{v},p_h) + (\{p_h\},[\![\boldsymbol{n}\cdot\boldsymbol{v}]\!])_{\Gamma\cup\Gamma_D} = l(\boldsymbol{v}) \\ \qquad\qquad b(\boldsymbol{u}_h,q) + (\{q\},[\![\boldsymbol{n}\cdot\boldsymbol{u}_h]\!])_{\Gamma\cup\Gamma_D} = (q,\boldsymbol{n}\cdot\boldsymbol{u}_D)_{\Gamma_D}, \end{cases} \tag{2}$$

where

$$\begin{aligned} a(\boldsymbol{u},\boldsymbol{v}) := {} & (2\nu\nabla^{\mathrm{s}}\boldsymbol{u},\nabla^{\mathrm{s}}\boldsymbol{v}) + C_{11}\,([\![\boldsymbol{n}\otimes\boldsymbol{u}]\!],[\![\boldsymbol{n}\otimes\boldsymbol{v}]\!])_{\Gamma\cup\Gamma_D} \\ & - (2\nu\{\nabla^{\mathrm{s}}\boldsymbol{u},[\![\boldsymbol{n}\otimes\boldsymbol{v}]\!])_{\Gamma\cup\Gamma_D} - ([\![\boldsymbol{n}\otimes\boldsymbol{u}]\!],2\nu\{\nabla^{\mathrm{s}}\boldsymbol{v}\})_{\Gamma\cup\Gamma_D}, \end{aligned} \tag{3a}$$

$$l(\boldsymbol{v}) := (\boldsymbol{f},\boldsymbol{v}) + (\boldsymbol{t},\boldsymbol{v})_{\Gamma_N} + C_{11}\,(\boldsymbol{u}_D,\boldsymbol{v})_{\Gamma_D} - (\boldsymbol{n}\otimes\boldsymbol{u}_D,2\nu\nabla^{\mathrm{s}}\boldsymbol{v})_{\Gamma_D}, \tag{3b}$$

$$\begin{aligned} c(\boldsymbol{w};\boldsymbol{u},\boldsymbol{v}) := \frac{1}{2}\Bigg[& -((\boldsymbol{w}\cdot\nabla)\boldsymbol{v},\boldsymbol{u}) + ((\boldsymbol{w}\cdot\nabla)\boldsymbol{u},\boldsymbol{v}) \\ & + \sum_{i=1}^{\mathtt{n}_{\mathtt{el}}} \int_{\partial\Omega_i\setminus\Gamma_N} \frac{1}{2}\left[(\boldsymbol{w}\cdot\boldsymbol{n}_i)(\boldsymbol{u}^{ext}+\boldsymbol{u}) - |\boldsymbol{w}\cdot\boldsymbol{n}_i|\,(\boldsymbol{u}^{ext}-\boldsymbol{u})\right]\cdot\boldsymbol{v}\,\mathrm{d}\Gamma + \int_{\Gamma_N}(\boldsymbol{w}\cdot\boldsymbol{n})\boldsymbol{u}\cdot\boldsymbol{v}\,\mathrm{d}\Gamma\Bigg] \end{aligned} \tag{4a}$$

and

$$b(\boldsymbol{v},p) := -\int_\Omega q\,\nabla\cdot\boldsymbol{v}\,\mathrm{d}\Omega. \tag{4b}$$

The penalty parameter, a positive scalar C_{11} of order $\mathcal{O}(h^{-1})$, must be large enough to ensure coercivity of the bilinear form $a\,(\cdot,\cdot)$ [18]. The characteristic mesh size is denoted by h. A standard upwind numerical flux, see for instance [24], is used for the stabilization of the convective term $c\,(\cdot;\cdot,\cdot)$. In Equation (4a), $\boldsymbol{u}^{\text{ext}}$ denotes the exterior trace of $\boldsymbol{u}$ taken over the side/face under consideration, that is,

$$\boldsymbol{u}^{\text{ext}}(\boldsymbol{x}) = \lim_{\varepsilon \to 0^+} \boldsymbol{u}(\boldsymbol{x} + \varepsilon \boldsymbol{n}_i) \qquad \text{for } \boldsymbol{x} \in \partial\Omega_i.$$

Remark 1
Note that in [19], the convective term was defined as

$$\begin{aligned} c\,(\boldsymbol{w};\boldsymbol{u},\boldsymbol{v}) :=& -((\boldsymbol{w}\cdot\nabla)\boldsymbol{v},\boldsymbol{u}) + \sum_{i=1}^{\mathtt{n_{el}}} \int_{\partial\Omega_i\backslash\Gamma_N} \frac{1}{2}\left[(\boldsymbol{w}\cdot\boldsymbol{n})(\boldsymbol{u}^{\text{ext}}+\boldsymbol{u}) - |\boldsymbol{w}\cdot\boldsymbol{n}_i|(\boldsymbol{u}^{\text{ext}}-\boldsymbol{u})\right]\cdot\boldsymbol{v}\,\mathrm{d}\Gamma \\ &+ \int_{\Gamma_N} (\boldsymbol{w}\cdot\boldsymbol{n})\boldsymbol{u}\cdot\boldsymbol{v}\,\mathrm{d}\Gamma. \end{aligned} \tag{5}$$

Nevertheless, when solving the unsteady incompressible Navier–Stokes equations, the original convective term of the strong form $(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}$ can be replaced by $(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} - \frac{1}{2}(\nabla\cdot\boldsymbol{u})\boldsymbol{u}$, which is a legitimate modification for a divergence-free velocity field [2] and leads to the trilinear convective term defined in Equation (4a). This guarantees unconditional stability, in the case of an implicit or semi-implicit time integration [25]. The importance of the choice of this skew-symmetric form will be commented in Section 3.3.

Following [18, 26, 27], the velocity space $\boldsymbol{\mathcal{V}}^h$ is now split into direct sum of a solenoidal part and an irrotational part $\boldsymbol{\mathcal{V}}^h = \boldsymbol{\mathcal{S}}^h \oplus \boldsymbol{\mathcal{I}}^h$, where

$$\begin{aligned} \boldsymbol{\mathcal{S}}^h &= \left\{\boldsymbol{v} \in [\mathcal{H}^1(\Omega)]^{\mathtt{n_{sd}}} \mid \boldsymbol{v}|_{\Omega_i} \in [\mathcal{P}^k(\Omega_i)]^{\mathtt{n_{sd}}},\ \nabla\cdot\boldsymbol{v}|_{\Omega_i} = 0 \text{ for } i = 1,\dots,\mathtt{n_{el}}\right\}, \\ \boldsymbol{\mathcal{I}}^h &\subset \left\{\boldsymbol{v} \in [\mathcal{H}^1(\Omega)]^{\mathtt{n_{sd}}} \mid \boldsymbol{v}|_{\Omega_i} \in [\mathcal{P}^k(\Omega_i)]^{\mathtt{n_{sd}}},\ \nabla\times\boldsymbol{v}|_{\Omega_i} = \boldsymbol{0} \text{ for } i = 1,\dots,\mathtt{n_{el}}\right\}, \end{aligned}$$

see [19] for examples of these spaces and [28] for further details on their construction.

Under these circumstances, IPM problem (2) can be split in two *uncoupled* problems. The first one solves for *divergence-free* velocities and *hybrid pressures*: find $\boldsymbol{u}_h \in \boldsymbol{\mathcal{S}}^h \times 0, T[$ and $\tilde{p}_h \in \boldsymbol{P}^h \times]0, T[$ solution of

$$\begin{cases} \left(\dfrac{\partial \boldsymbol{u}_h}{\partial t}, \boldsymbol{v}\right) + a_{\mathtt{IP}}\,(\boldsymbol{u}_h, v) + c\,(\boldsymbol{u}_h;\boldsymbol{u}_h, v) + (\tilde{p}_h, [\![\boldsymbol{n}\cdot\boldsymbol{v}]\!])_{\Gamma\cup\Gamma_D} = l_{\mathtt{IP}}\,(\boldsymbol{v}) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad (\tilde{q}, [\![\boldsymbol{n}\cdot u_h]\!])_{\Gamma\cup\Gamma_D} = (\tilde{q}, n\cdot\boldsymbol{u}_D)_{\Gamma_D}, \end{cases} \tag{6}$$

$\forall \boldsymbol{v} \in \boldsymbol{\mathcal{S}}^h$, $\forall \tilde{q} \in \boldsymbol{P}^h$, $\forall\ t \in]0, T[$, with the forms defined in Equations (3), (4a), and (4b). Note that this problem, which has to be solved at each time step, shows an important reduction in the number of degrees of freedom (DOF) with respect to problem (2), as explained in [19].

The space of hybrid pressures (pressures along the sides in 2D or faces in 3D) is simply the following:

$$\boldsymbol{P}^h := \left\{\tilde{p} \mid \tilde{p} : \Gamma\cup\Gamma_D \longrightarrow \mathbb{R} \text{ and } \tilde{p} = [\![\boldsymbol{n}\cdot\boldsymbol{v}]\!] \text{ for some } \boldsymbol{v} \in \boldsymbol{\mathcal{S}}^h\right\}.$$

In fact, reference [26] demonstrates that $\boldsymbol{P}^h$ corresponds to piecewise polynomial pressures in the element sides in 2D or faces in 3D.

The second problem, which requires the solution of the previous one, evaluates *interior* pressures: find $p_h \in \mathcal{Q}^h \times]0, T[$ such that $\forall \boldsymbol{v} \in \boldsymbol{\mathcal{I}}^h$ and $\forall t \in]0, T[$

$$b\,(\boldsymbol{v}, p_h) = l_{\mathtt{IP}}\,(\boldsymbol{v}) - \left(\frac{\partial \boldsymbol{u}_h}{\partial t}, \boldsymbol{v}\right) - a_{\mathtt{IP}}\,(\boldsymbol{u}_h, \boldsymbol{v}) - (\tilde{p}_h, [\![\boldsymbol{n}\cdot\boldsymbol{v}]\!])_{\Gamma\cup\Gamma_D} - c\,(\boldsymbol{u}_h;\boldsymbol{u}_h,\boldsymbol{v}). \tag{7}$$

It is important to note that Equation (7) can be solved element by element and pressure is its only unknown. The second problem, Equation (7), is a postprocessing that allows to compute pressure in the elements interior, usually at the end of the computation or after the iterations in each time step. For example, if interior pressure p_h needs to be calculated at time t^n, Equation (7) is solved at t^n, where $\partial \boldsymbol{u}/\partial t$ can be approximated using

$$\left.\frac{\partial \boldsymbol{u}}{\partial t}\right|^n = \frac{-\boldsymbol{u}^{n+2} + 8\boldsymbol{u}^{n+1} - 8\boldsymbol{u}^{n-1} + \boldsymbol{u}^{n-2}}{12\Delta t}, \text{ for fourth-order accuracy in time.} \tag{8}$$

This choice preserves, for interior pressure recovery, the high order of convergence of hybrid pressure obtained with some RK or Rosenbrock methods (Section 3). Note that here, when interior pressure is calculated as a postprocessing using Equation (8), two additional iterations are needed to compute $\boldsymbol{u}^{n+1}$ and $\boldsymbol{u}^{n+2}$. If interior pressure needs to be calculated at each time step, interior pressure is always computed two steps later than velocity and hybrid pressure. Else, other approximations can be used, such as

$$\left.\frac{\partial \boldsymbol{u}}{\partial t}\right|^n = \frac{\boldsymbol{u}^{n} - \boldsymbol{u}^{n-1}}{\Delta t}, \text{ for first-order accuracy in time,} \tag{9a}$$

$$\left.\frac{\partial \boldsymbol{u}}{\partial t}\right|^n = \frac{3\boldsymbol{u}^{n} - 4\boldsymbol{u}^{n-1} + \boldsymbol{u}^{n-2}}{2\Delta t}, \text{ for second-order accuracy in time.} \tag{9b}$$

3. IMPLICIT METHODS FOR UNSTEADY INCOMPRESSIBLE FLOWS

Here, the space discretization of the incompressible Navier–Stokes equations is carried out using a DG IPM with solenoidal piecewise approximations, as detailed in Section 2. Nevertheless, the algorithms discussed in this work would be equally applicable to other types of discretization schemes, for example, using classical DG (with non-divergence-free elements or hybrid pressure) or continuous Galerkin. In any case, the space discretization of the unsteady incompressible Navier–Stokes problem (1) can be written as

$$\begin{cases} \mathbf{M}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} + \mathbf{C}(\mathbf{u})\mathbf{u} + \mathbf{G}\mathbf{p} = \mathbf{f_1} \\ \mathbf{G}^T\mathbf{u} = \mathbf{f_2} \end{cases} \tag{10}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{K}$ the diffusion matrix, $\mathbf{C}$ the convection matrix, $\mathbf{G}$ the discrete gradient/divergence matrix, $\mathbf{u}$ and $\mathbf{p}$ the vectors of nodal values or approximation coefficients of velocity and pressure, respectively, $\dot{\mathbf{u}}$ denotes the time derivative, and $\mathbf{f_1}$ and $\mathbf{f_2}$ vectors taking into account force term and boundary conditions; see Appendix 5 for the implementation of the semidiscretized forms. This system of $\mathtt{n}_{\mathtt{DOF}}$ DOF can also be written as

$$\begin{cases} \mathbf{M}\dot{\mathbf{u}} = \mathcal{F}(t, \mathbf{u}, \mathbf{p}) \\ 0 = \mathcal{G}(t, \mathbf{u}) \end{cases} \tag{11}$$

with $t \in]0, T[$ and where

$$\begin{aligned} \mathcal{F}(t, \mathbf{u}, \mathbf{p}) &= \mathbf{f_1} - \mathbf{K}\mathbf{u} - \mathbf{C}(\mathbf{u})\mathbf{u} - \mathbf{G}\mathbf{p}, \\ \mathcal{G}(t, \mathbf{u}) &= \mathbf{G}^T\mathbf{u} - \mathbf{f_2}. \end{aligned} \tag{12}$$

Note that $\dfrac{\partial \mathcal{G}}{\partial \mathbf{u}}\dfrac{\partial \mathcal{F}}{\partial \mathbf{p}} = \mathbf{G}^T\mathbf{M}^{-1}\mathbf{G}$ is invertible ($\mathbf{M}$ is regular and $\mathbf{G}$ has full rank), therefore, Equation (11) is a Hessenberg index-2 DAE system [29].

Differential algebraic equations originate in the modeling of various physical or chemical phenomena and have been deeply studied during the last years [29, 30]. They are classified by their differential index, that is, the minimum number of times that a DAE system must be differentiated to obtain an ODE. For instance, the discrete incompressible Stokes, Oseen, and Navier–Stokes equations are index-2 DAE systems.

Many numerical methods initially defined for ODEs have been adapted to DAEs, as for example, multistep backward differentiation formulae [31] or RK methods [30]. At first, RK methods were regarded as poor competitors to multistep methods. The reason was consistent: for most DAEs and RK methods, the order of convergence obtained was less than the order obtained for ODEs, and the higher the index, the larger the reduction. Then, however, reference [14] showed that proper RK methods can form the basis of a competitive code because they are unconditionally stable and can reach orders of convergence as high as when applied to ODE. An alternative to RK methods are the Rosenbrock methods, for which order reduction is avoided by using more stages and satisfying further order conditions, allowing to reach up to third order of convergence for index-2 DAEs [14, 32–34].

3.1. Implicit and singly diagonally implicit Runge–Kutta methods

An s-stage RK method for the index-2 DAE (11) reads

$$\begin{aligned} \mathbf{u}^{n+1} &= \mathbf{u}^n + \Delta t \sum_{i=1}^{s} b_i \mathbf{l}_i \\ \mathbf{p}^{n+1} &= \mathbf{p}^n + \Delta t \sum_{i=1}^{s} b_i \mathbf{k}_i \end{aligned} \tag{13}$$

where $\mathbf{l}_i$ and $\mathbf{k}_i$ are defined as the solution of the system

$$\mathbf{M}\mathbf{l}_i = \mathcal{F}\left(t^n + c_i \Delta t, \mathbf{u}^n + \Delta t \sum_{j=1}^{s} a_{ij} \mathbf{l}_j, \mathbf{p}^n + \Delta t \sum_{j=1}^{s} a_{ij} \mathbf{k}_j\right) \tag{14a}$$

$$0 = \mathcal{G}\left(t^n + c_i \Delta t, \mathbf{u}^n + \Delta t \sum_{j=1}^{s} a_{ij} \mathbf{l}_j\right) \tag{14b}$$

for $i = 1, ..., s$, [14]. Coefficients a_{ij}, b_i, and c_i come from the Butcher array, whose general form is seen in Table I. Depending on the specific form of the Butcher array, implicit, semi-implicit, or explicit RK methods are obtained. An RK method is said to be explicit if its Butcher array is strictly lower triangular, that is, $a_{ij} = 0$ for $j \geqslant i$. Otherwise, the method is implicit (IRK). In particular, an implicit method is said to be semi-implicit or diagonally implicit (DIRK), if $a_{ij} = 0$ for $j > i$ and $a_{ii} \neq 0$ for some i. If, in addition, all diagonal coefficients (a_{ii}) are identical, the method is called singly diagonally implicit (SDIRK). SDIRK are of special interest for a linear problem, as for example, the Stokes problem, because one may hope to use repeatedly the stored LU-factorization of the matrix. For nonlinear problems, this can also be an advantage if a simplified Newton method (conserving the same Jacobian) is used.

This work focuses on fully implicit and semi-implicit methods because of their stability properties. In fact, explicit RK methods cannot even be used in the form of Equations (13) and (14) for Hessenberg index-2 DAEs because the resulting system (14) is underdetermined to solve for $\mathbf{l}_i$ and $\mathbf{k}_i$. Nevertheless, in [35], explicit RK methods are applied to DAE, using a different formulation than Equations (13) and (14). In this case, the order of convergence of explicit RK methods is less

Table I. Butcher array.

c_1	a_{11}	a_{12}	$\cdots$	a_{1s}
c_2	a_{21}	a_{22}	$\cdots$	a_{2s}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
c_s	a_{s1}	a_{s2}	$\cdots$	a_{ss}
	b_1	b_2	$\cdots$	b_s

Table II. Orders of convergence for s-stage implicit Runge–Kutta methods for index-2 DAEs and for ODEs [14, 37].

Method	DAE: $\boldsymbol{u}$ error	DAE: p error	ODE error
Radau IA	$(\Delta t)^{s}$	$(\Delta t)^{s-1}$	$(\Delta t)^{2s-1}$
Radau IIA	$(\Delta t)^{2s-1}$	$(\Delta t)^{s}$	$(\Delta t)^{2s-1}$
Lobatto IIIC	$(\Delta t)^{2s-2}$	$(\Delta t)^{s-1}$	$(\Delta t)^{2s-2}$

DAE, differential algebraic equation; ODE, ordinary differential equation.

Table III. Butcher array for Radau IIA implicit Runge–Kutta (IRK) methods.

(a) 2-stage IRK

$\frac{1}{3}$	$\frac{5}{12}$	$-\frac{1}{12}$
1	$\frac{3}{4}$	$\frac{1}{4}$
	$\frac{3}{4}$	$\frac{1}{4}$

(b) 3-stage IRK

$\frac{4-\sqrt{6}}{10}$	$\frac{88-7\sqrt{6}}{360}$	$\frac{296-169\sqrt{6}}{1800}$	$\frac{-2+3\sqrt{6}}{225}$
$\frac{4+\sqrt{6}}{10}$	$\frac{296+169\sqrt{6}}{1800}$	$\frac{88+7\sqrt{6}}{360}$	$\frac{-2-3\sqrt{6}}{225}$
1	$\frac{16-\sqrt{6}}{36}$	$\frac{16+\sqrt{6}}{36}$	$\frac{1}{9}$
	$\frac{16-\sqrt{6}}{36}$	$\frac{16+\sqrt{6}}{36}$	$\frac{1}{9}$

than the one reached for a regular ODE. For example, the four-stage explicit RK scheme applied to the incompressible Navier–Stokes equations in [35] only leads to second-order accuracy for velocity and pressure when no pressure correction is applied [36].

Table II shows the order of convergence for index-2 DAE (such as the discrete incompressible Navier–Stokes problem) and for ODE, when considering s-stage Radau IA, IIA, and Lobatto IIIC methods. Other methods, such as Gauss or Lobatto IIIA, are dismissed because they present higher order reduction when applied to DAEs with respect to ODEs. As shown in Table II, the best orders of convergence for velocity and pressure are obtained for a Radau IIA-IRK method, keeping the order of convergence of velocity for DAEs as high as for ODEs.

Table III shows Butcher diagrams for two-stage and three-stage Radau IIA-IRK methods. Radau IIA-IRK methods are a special case of IRK methods satisfying the additional property $b_j = a_{sj}$ for $j = 1, \cdots, s$. These methods are called IRK(DAE), and they stand out from all IRK methods in view of their applicability to DAE because at the last stage, $\mathbf{u}^{n+1}$ directly satisfies the constraint $\mathcal{G}(t^{n+1}, \mathbf{u}^{n+1}) = 0$. Because of this additional property, two-stage and three-stage Radau IIA-IRK are selected among fully implicit RK methods to be compared from accuracy and cost points of view in Section 4.1.

Note that the solution of an index-2 DAE system, such as Equation (11), with a fully implicit s-stage RK method requires solving a nonlinear system of equations of dimension $s\mathrm{n}_{\mathrm{DOF}}$ at each time step, where $\mathrm{n}_{\mathrm{DOF}}$ is the number of DOF in Equation (11). An alternative to reduce the computational cost would be to use an SDIRK method.

For instance, Table IV shows the Butcher diagram for two-stage, three-stage, and five-stage SDIRK methods. The computational effort in implementing semi-implicit methods is substantially less than for a fully implicit method, indeed s systems of dimension $\mathrm{n}_{\mathrm{DOF}}$ must be solved, instead of a problem of dimension $s\mathrm{n}_{\mathrm{DOF}}$ for the fully implicit scheme.

Unfortunately, SDIRK methods do not reach high orders of convergence, as illustrated in Table V. Unlike for ODE problems, increasing the number of stages of SDIRK methods does not improve the order of convergence for index-2 DAE systems. The order of convergence of SDIRK methods for an index-2 DAE system is limited to 2 for velocity and 1 for pressure, for two, three, or five stages.[‡] Note that, this is exactly the same order as the classical CN scheme, which has considerably

[‡] Norsett [38] conjectured and presented some evidence indicating that for any s even number greater than two, no SDIRK method exists with order $s + 1$ for an ODE. That is why no four-stage method appears in Table V.

Table IV. Butcher array for singly diagonally implicit Runge–Kutta (SDIRK) methods.

(a) 2-stage SDIRK, $\gamma = \frac{3+\sqrt{3}}{6}$

$$\begin{array}{c|cc} \gamma & \gamma & \\ 1-\gamma & 1-2\gamma & \gamma \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

(b) 3-stage SDIRK, $\gamma = \frac{1}{\sqrt{3}}\cos(\frac{\pi}{18}) + \frac{1}{2}$, $\delta = \frac{1}{6(2\gamma-1)^2}$

$$\begin{array}{c|ccc} \gamma & \gamma & & \\ \frac{1}{2} & \frac{1}{2}-\gamma & \gamma & \\ 1-\gamma & 2\gamma & 1-4\gamma & \gamma \\ \hline & \delta & 1-2\delta & \delta \end{array}$$

(c) 5-stage SDIRK

$$\begin{array}{c|ccccc} \frac{1}{4} & \frac{1}{4} & & & & \\ \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & & & \\ \frac{11}{20} & \frac{17}{50} & -\frac{1}{25} & \frac{1}{4} & & \\ \frac{1}{2} & \frac{371}{1360} & -\frac{137}{2720} & \frac{15}{544} & \frac{1}{4} & \\ 1 & \frac{25}{24} & -\frac{49}{48} & \frac{125}{16} & -\frac{85}{12} & \frac{1}{4} \\ \hline & \frac{25}{24} & -\frac{49}{48} & \frac{125}{16} & -\frac{85}{12} & \frac{1}{4} \end{array}$$

Table V. Orders of convergence for singly diagonally implicit Runge–Kutta methods of Table IV for index-2 DAEs and for ODEs [14, 37].

Number of stages	DAE: $\boldsymbol{u}$ error	DAE: p error	ODE error
2	2	1	3
3	2	1	4
5	2	1	4

DAE, differential algebraic equation; ODE, ordinary differential equation.

less computational cost (one system of dimension $\mathtt{n_{DOF}}$ at each time step), so the choice of SDIRK methods is dismissed.

Although standard SDIRK methods cannot satisfy high-order conditions for index-2 problems, their reduced cost make them very interesting compared with fully implicit schemes. To increase the order of the first calculated stage, it is possible to use the solution from the previous step to provide additional information. That is, an explicit first stage with $c_1 = 0$ and $a_{11} = 0$ is added [39–41]. With this modification, formally, this method should be classified as a DIRK scheme. Table VI shows the coefficients satisfying the order conditions for a four-stage DIRK method to reach third order for velocity and second order for pressure.

Thus, in this work, fully implicit two-stage Radau IIA-IRK method and four-stage DIRK methods, both third-order methods, are first chosen among other RK methods for the solution of incompressible flow problems. Section 4.1 explores if a two-stage method involving systems of equations

Table VI. Butcher array for four-stage diagonally implicit Runge–Kutta method.

$$\begin{array}{c|cccc} 0 & 0 & & & \\ 1 & \frac{1}{2} & \frac{1}{2} & & \\ \frac{3}{2} & \frac{5}{8} & \frac{3}{8} & \frac{1}{2} & \\ 1 & \frac{7}{18} & \frac{1}{3} & -\frac{2}{9} & \frac{1}{2} \\ \hline & \frac{7}{18} & \frac{1}{3} & -\frac{2}{9} & \frac{1}{2} \end{array}$$

of size $2\mathtt{n}_{\mathtt{DOF}}$ is more or less efficient than a four-stage method involving systems of equations of size $\mathtt{n}_{\mathtt{DOF}}$. Furthermore, three-stage Radau IIA-IRK method, which is expected to be fifth order, is contemplated to study if the extra cost of the extra stage is compensated by the higher accuracy obtained. These methods are all compared in terms of accuracy and relative computational cost with the classical CN method in Section 4.1. In the following section, a third-order Rosenbrock formulation is presented, completing the panel of third-order methods.

3.2. Rosenbrock methods

Originally thought for stiff problems, Rosenbrock methods can be derived from SDIRK methods and avoid the solution of nonlinear systems. At each time step, $\mathbf{u}$ and $\mathbf{p}$ are updated using the same formulation as in standard RK method (Equation (13)), but in this case, $\mathbf{l}_i$ and $\mathbf{k}_i$ are solution of a linearized system of equations. To derive Rosenbrock formulation, consider first the non-autonomous system (11); it is made autonomous by adding $\partial t/\partial t = 1$, then the SDIRK method is applied

$$\begin{cases} \mathbf{M}\mathbf{l}_i = \mathcal{F}\left(t^n + \Delta t \displaystyle\sum_{j=1}^{s} a_{ij} t_j, \mathbf{u}^n + \Delta t \sum_{j=1}^{s} a_{ij}\mathbf{l}_j, \mathbf{p}^n + \Delta t \sum_{j=1}^{s} a_{ij}\mathbf{k}_j\right) \\ 0 = \mathcal{G}\left(t^n + \Delta t \displaystyle\sum_{j=1}^{s} a_{ij} t_j, \mathbf{u}^n + \Delta t \sum_{j=1}^{s} a_{ij}\mathbf{l}_j\right) \\ t_i = 1 \end{cases} \tag{15}$$

for $i = 1, \dots, s$, where s is the number of stages.

Applying one iteration of Newton–Raphson method leads to

$$\begin{aligned} \begin{pmatrix} \mathbf{l}_i^1 \\ \mathbf{k}_i^1 \\ t_i^1 \end{pmatrix} = \begin{pmatrix} \mathbf{l}_i^0 \\ \mathbf{k}_i^0 \\ t_i^0 \end{pmatrix} & \\ - \begin{pmatrix} \mathbf{M} - \Delta t a_{ii} \dfrac{\partial \mathcal{F}(t_i^0, \mathbf{u}_i^0, \mathbf{p}_i^0)}{\partial \mathbf{l}_i} & -\Delta t a_{ii} \dfrac{\partial \mathcal{F}(t_i^0, \mathbf{u}_i^0, \mathbf{p}_i^0)}{\partial \mathbf{k}_i} & -\Delta t a_{ii} \dfrac{\partial \mathcal{F}(t_i^0, \mathbf{u}_i^0, \mathbf{p}_i^0)}{\partial t_i} \\ -\Delta t a_{ii} \dfrac{\partial \mathcal{G}(t_i^0, \mathbf{u}_i^0)}{\partial \mathbf{l}_i} & 0 & -\Delta t a_{ii} \dfrac{\partial \mathcal{G}(t_i^0, \mathbf{u}_i^0)}{\partial t_i} \\ 0 & 0 & 1 \end{pmatrix}^{-1} & \\ \times \begin{pmatrix} \mathbf{l}_i^0 - \mathcal{F}(t_i^0, \mathbf{u}_i^0, \mathbf{p}_i^0) \\ -\mathcal{G}(t_i^0, \mathbf{u}_i^0) \\ t_i^0 - 1 \end{pmatrix} & \end{aligned} \tag{16}$$

where $t_i^0 = t^n + \Delta t \left(\displaystyle\sum_{j=1}^{i-1} a_{ij} t_j + a_{ii} t_i^0\right)$, $\mathbf{u}_i^0 = \mathbf{u}^n + \Delta t \left(\displaystyle\sum_{j=1}^{i-1} a_{ij}\mathbf{l}_j + a_{ii}\mathbf{l}_i^0\right)$, $\mathbf{p}_i^0 = \mathbf{p}^n + \Delta t \left(\displaystyle\sum_{j=1}^{i-1} a_{ij}\mathbf{k}_j + a_{ii}\mathbf{k}_i^0\right)$.

Considering $\mathbf{l}_i^0 = 0$, $\mathbf{k}_i^0 = 0$, $t_i^0 = 0$, and $t_i^1 = 1$ and iterating Newton–Raphson only once, Equation (16) is equivalent to solve

$$\left(\begin{bmatrix}\mathbf{M} & 0\\ 0 & 0\end{bmatrix} - \Delta t a_{ii}\begin{bmatrix}\dfrac{\partial\mathcal{F}(t_i,\mathbf{u}_i,\mathbf{p}_i)}{\partial\mathbf{l}_i} & \dfrac{\partial\mathcal{F}(t_i,\mathbf{u}_i,\mathbf{p}_i)}{\partial\mathbf{k}_i}\\ \dfrac{\partial\mathcal{G}(t_i,\mathbf{u}_i)}{\partial\mathbf{l}_i} & 0\end{bmatrix}\right)\begin{bmatrix}\mathbf{l}_i\\ \mathbf{k}_i\end{bmatrix}$$

$$= \begin{bmatrix}\mathcal{F}\,(t_i,\mathbf{u}_i,\mathbf{p}_i)\\ \mathcal{G}\,(t_i,\mathbf{u}_i)\end{bmatrix} + \Delta t a_{ii}\begin{bmatrix}\dfrac{\partial\mathcal{F}(t_i,\mathbf{u}_i,\mathbf{p}_i)}{\partial t_i}\\ \dfrac{\partial\mathcal{G}(t_i,\mathbf{u}_i)}{\partial t_i}\end{bmatrix} \tag{17}$$

with $t_i = t^n + \Delta t \sum_{j=1}^{i-1} a_{ij} t_j$, $\mathbf{u}_i = \mathbf{u}^n + \Delta t \sum_{j=1}^{i-1} a_{ij}\mathbf{l}_j$, $\mathbf{p}_i = \mathbf{p}^n + \Delta t \sum_{j=1}^{i-1} a_{ij}\mathbf{k}_j$.

Correction terms are then added in order to keep the same Jacobian and time derivative in all stages, see [13, 32] for details. The resulting linear system of equations solving for $\mathbf{l}_i$ and $\mathbf{k}_i$ is

$$\left(\begin{bmatrix}\mathbf{M} & 0\\ 0 & 0\end{bmatrix} - \Delta t\gamma\begin{bmatrix}\dfrac{\partial\mathcal{F}}{\partial\mathbf{u}}(t^n,\mathbf{u}^n,\mathbf{p}^n) & \dfrac{\partial\mathcal{F}}{\partial\mathbf{p}}(t^n,\mathbf{u}^n,\mathbf{p}^n)\\ \dfrac{\partial\mathcal{G}}{\partial\mathbf{u}}(t^n,\mathbf{u}^n) & 0\end{bmatrix}\right)\begin{bmatrix}\mathbf{l}_i\\ \mathbf{k}_i\end{bmatrix}$$

$$= \begin{bmatrix}\mathcal{F}\left(t^n + \Delta t\, a_i, \mathbf{u}^n + \Delta t \displaystyle\sum_{j=1}^{i-1} a_{ij}\mathbf{l}_j, \mathbf{p}^n + \Delta t \sum_{j=1}^{i-1} a_{ij}\mathbf{k}_j\right)\\ \mathcal{G}\left(t^n + \Delta t\, a_i, \mathbf{u}^n + \Delta t \displaystyle\sum_{j=1}^{i-1} a_{ij}\mathbf{l}_j,\right)\end{bmatrix}$$

$$+ \Delta t\left(\sum_{j=1}^{i-1}\gamma_{ij}\begin{bmatrix}\dfrac{\partial\mathcal{F}}{\partial\mathbf{u}}(t^n,\mathbf{u}^n,\mathbf{p}^n) & \dfrac{\partial\mathcal{F}}{\partial\mathbf{p}}(t^n,\mathbf{u}^n,\mathbf{p}^n)\\ \dfrac{\partial\mathcal{G}}{\partial\mathbf{u}}(t^n,\mathbf{u}^n) & 0\end{bmatrix}\begin{bmatrix}\mathbf{l}_j\\ \mathbf{k}_j\end{bmatrix} + \gamma_i\begin{bmatrix}\dfrac{\partial\mathcal{F}}{\partial t}(t^n,\mathbf{u}^n,\mathbf{p}^n)\\ \dfrac{\partial\mathcal{G}}{\partial t}(t^n,\mathbf{u}^n)\end{bmatrix}\right) \tag{18}$$

for $i = 1,\ldots,s$, where γ, γ_i, γ_{ij}, a_i, and a_{ij} are new parameters, where a_i and γ_i are then calculated as follows [42]

$$a_1 = 0, \quad a_i = \sum_{j=1}^{i-1} a_{ij} \text{ for } i = 2,\ldots,s,$$

$$\text{and } \gamma_1 = \gamma, \quad \gamma_i = \sum_{j=1}^{i-1}\gamma_{ij} + \gamma \text{ for } i = 2,\ldots,s.$$

Note that this approach only requires the solution of s linear systems of $\mathtt{n}_{\mathtt{DOF}}$ equations with the same matrix at each time step. Because the left-hand side matrix is independent of the stage number, the same LU factorization can be used repeatedly for each stage.

Parameters γ, γ_i, γ_{ij}, a_i, a_{ij}, and b_i have to be calculated for each Rosenbrock scheme, fulfilling some order conditions to obtain a sufficient consistency order. In [42], it has been proved that three-stage Rosenbrock method can not reach third order of accuracy for index-2 DAE, so four-stage methods have to be used. Table VII gives the values of the parameters for a four-stage ROSI2Pw method that reaches third order of accuracy for index-2 DAE.

Table VII. Set of coefficients for the four-stage ROSI2Pw method [42].

$\gamma = 4.3586652150845900e^{-1}$	
$a_{21} = 8.7173304301691801e^{-1}$	$\gamma_{21} = -8.7173304301691801e^{-1}$
$a_{31} = 7.8938917169345013e^{-1}$	$\gamma_{31} = -8.4175599602920992e^{-1}$
$a_{32} = -3.9389171693450180e^{-2}$	$\gamma_{32} = -1.2977652642309580e^{-2}$
$a_{41} = 6.2787416864263046e^{-1}$	$\gamma_{41} = -3.7964867148089526e^{-1}$
$a_{42} = 6.9295440480994763$	$\gamma_{42} = -8.3490231248017537$
$a_{43} = -6.5574182167421071$	$\gamma_{43} = 8.2928052747741905$
$b_1 = 2.4822549716173517e^{-1}$	
$b_2 = -1.4194790767022774$	
$b_3 = 1.7353870580320832$	
$b_4 = 4.3586652150845900e^{-1}$	

Note that six-stage Rosenbrock methods have also been developed [43]. These schemes, for an ODE and index-1 DAE problems, may attain order four, but they only reach third order for index-2 DAE. This is why in here, a four-stage Rosenbrock method is considered, and it will be further compared with DIRK and IRK methods.

3.3. *Asymptotic stability*

The goal of this section is to recall the asymptotic stability properties of Rosenbrock and RK methods, particularize them to the case of the incompressible Navier–Stokes equations, and then justify the choice of the skew-symmetric convective term defined in Equation (4a). It will actually be seen that even though Rosenbrock and RK methods are supposed to be unconditionally stable, if the space discretization is not correctly performed, the stability of the time-integration scheme may be limited.

3.3.1. Analytical stability regions. As a step towards the study of the asymptotic properties of the solution of the incompressible Navier–Stokes equations, the linear homogeneous Oseen equations are now considered. The results of asymptotic stability for this scheme can then be extended to the nonlinear Navier–Stokes equations. The strong form of the unsteady incompressible homogeneous Oseen problem is

$$\frac{\partial \boldsymbol{u}}{\partial t} - 2\nabla\cdot(\nu\nabla^{s}\boldsymbol{u}) + \nabla p + (\boldsymbol{w}\cdot\nabla)\boldsymbol{u} = \boldsymbol{0} \qquad \text{in } \Omega\times]0,T[, \tag{19a}$$

$$\nabla\cdot\boldsymbol{u} = 0 \qquad \text{in } \Omega\times]0,T[, \tag{19b}$$

where $\boldsymbol{w}$ is a given velocity field, with boundary and interface conditions (1c)–(1d) and initial condition (1e). Its discretized form is

$$\dot{\mathbf{u}} + \mathbf{M}^{-1}(\mathbf{K}+\mathbf{C})\,\mathbf{u} + \mathbf{M}^{-1}\mathbf{G}\mathbf{p} = \mathbf{0}, \tag{20a}$$

$$\mathbf{G}^{T}\mathbf{u} = \mathbf{0}. \tag{20b}$$

where, now, all matrices $\mathbf{M}$, $\mathbf{K}$, $\mathbf{C}$, and $\mathbf{G}$ are constant. Following the discussion of asymptotic properties of solutions of general linear DAEs and, in particular, the cases of index-2 DAEs in [20, 21], system (19) is rewritten as a simpler equivalent DAE system. For this purpose, let $\mathbf{A} = \mathbf{M}^{-1}(\mathbf{K}+\mathbf{C})$ and $\mathbf{H} = \mathbf{M}^{-1}\mathbf{G}\left(\mathbf{G}^{T}\mathbf{M}^{-1}\mathbf{G}\right)^{-1}\mathbf{G}^{T}$. From Equation (20b), we get

$$\mathbf{H}\mathbf{u} = \mathbf{0}. \tag{21}$$

Multiplying Equation (20a) by $\mathbf{G}^{T}$ and solving for $\mathbf{p}$, we get

$$\mathbf{p} = -\left(\mathbf{G}^{T}\mathbf{M}^{-1}\mathbf{G}\right)^{-1}\mathbf{G}^{T}\left[\dot{\mathbf{u}} + \mathbf{A}\mathbf{u}\right]. \tag{22}$$

Substituting Equation (22) in Equation (20a)

$$\dot{\mathbf{u}} + \mathbf{A}\mathbf{u} - \mathbf{M}^{-1}\mathbf{G}\left(\mathbf{G}^T\mathbf{M}^{-1}\mathbf{G}\right)^{-1}\mathbf{G}^T\left[\dot{\mathbf{u}} + \mathbf{A}\mathbf{u}\right] = \mathbf{0},$$

which can be written as

$$(\mathbf{I} - \mathbf{H})\,\dot{\mathbf{u}} + (\mathbf{I} - \mathbf{H})\,\mathbf{A}\mathbf{u} = \mathbf{0}$$

or using Equation (21) and the fact that $\mathbf{H}$ is constant in time

$$\dot{\mathbf{z}} = -\,(\mathbf{I} - \mathbf{H})\,\mathbf{A}\mathbf{z}, \tag{23}$$

with $\mathbf{z} = (\mathbf{I} - \mathbf{H})\,\mathbf{u}$. Thus, the solution of Equation (20) consists of three parts: one ODE (23) for the variable $\mathbf{z} = (\mathbf{I} - \mathbf{H})\,\mathbf{u}$ and two algebraic equations (21) and (22), for $\mathbf{z}$, $\mathbf{u}$, and $\mathbf{p}$. Studying the eigenvalues of $(\mathbf{I} - \mathbf{H})\,\mathbf{A}$ and using stability functions for RK or Rosenbrock schemes give a necessary condition for the asymptotic stability of the solution.

A necessary condition for a stable scheme is that all the eigenvalues of $(\mathbf{I} - \mathbf{H})\,\mathbf{A}$ lie in the stability region of the chosen RK or Rosenbrock scheme. For s-stage Rosenbrock and implicit RK methods, the stability function is

$$R(z) = 1 + z\mathbf{b}^T(\mathbf{I} - z\mathbf{B})^{-1}\mathbf{e}_n$$

where $z = \lambda\Delta t$, $\mathbf{b}^T = (b_1, b_2, ..., b_s)$, $\mathbf{B}_{ij} = a_{ij} + \gamma_{ij}$ and $\mathbf{e}_n$ is a $n \times 1$ vector of 1 [14]. Note that for IRK, all $\gamma_{ij} = 0$. For instance, for the two-stage and three-stage Radau IIA-IRK methods, the stability functions are

$$R(z) = \frac{6 + 2z}{6 - 4z + z^2} \qquad \text{for two-stage Radau IIA-IRK,}$$

$$R(z) = \frac{60 + 24z + 3z^2}{60 - 36z + 9z^2 - z^3} \qquad \text{for three-stage Radau IIA-IRK,}$$

with $z = \lambda\Delta t$ [14].

A necessary condition for stability is $|R(\lambda\Delta t)| \leqslant 1$, which must hold for all λ, that is, all eigenvalues of $(\mathbf{I} - \mathbf{H})\,\mathbf{A}$ and any given time step Δt. That is, the method is A-stable. Furthermore, recall that an A-stable method with $|R(z)| \to 0$ when $z \to \infty$ is L-stable. Radau-IIA methods and ROSI2Pw are examples of L-stable methods; their stability regions are shown in Figure 1. Note that

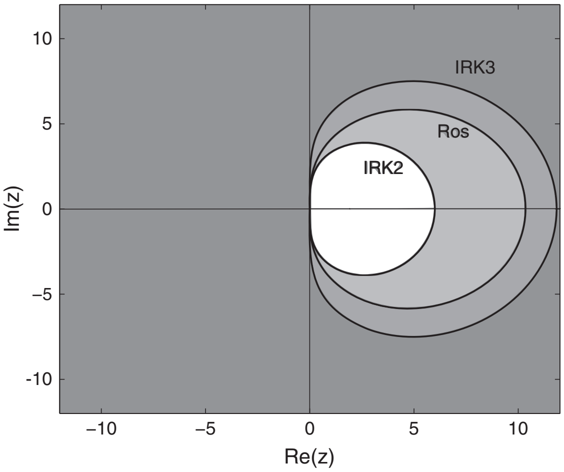


Figure 1. Stability regions in the complex plane for Radau IIA implicit Runge–Kutta with two stages (IRK2), three stages (IRK3), and ROSI2Pw Rosenbrock (Ros) methods. The stable regions correspond to the areas outside the stability borders, that is, the gray areas.

the four-stage DIRK method presented in Section 3.1 is also A-stable but not L-stable; see [41] for further details.

Next, this analysis is applied to the Navier–Stokes or the Oseen equations. The conclusions are that the discretization of the Navier–Stokes or the Oseen equations always leads to systems of DAEs such that the eigenvalues of $(\mathbf{I}-\mathbf{H})\,\mathbf{A}$ have negative real part provided that the skew-symmetric form of the convective term is used (Remark 1). Thus, any method containing the left-hand side of the complex plane ($Re(z) \leqslant 0$) in its stability region, such as the Radau IIA-IRK, four-stage DIRK, and Rosenbrock methods, satisfies the necessary stability condition.

3.3.2. Numerical validation. The theoretical asymptotic stability study is developed for the Oseen equations. Numerical examples are used to validate the extension of these results to the Navier–Stokes equations and justify the choice of the skew-symmetric convective term defined in Equation (4a). Here, a two-stage Radau IIA-IRK method is considered, but a similar rationale can be applied to other RK or Rosenbrock methods.

An example with analytical solution proposed in [5] is considered. The incompressible Navier–Stokes equations are solved in a 2D square domain $\Omega =]0,1/2[\times]0,1/2[$ with Dirichlet boundary conditions on three sides and Neumann boundary condition on the fourth side $\{x=0\}$. A body force

$$\boldsymbol{f} = \begin{pmatrix} 2\nu\sin(x+t)\sin(y+t)+\cos(x-y+t) \\ \qquad\qquad +\sin(x+y+2t)+\sin(x+t)\cos(x+t) \\ 2\nu\cos(x+t)\cos(y+t)-\cos(x-y+t) \\ \qquad\qquad -\sin(x+y+2t)-\sin(y+t)\cos(y+t) \end{pmatrix}$$

is imposed to have the analytical solution

$$\begin{aligned} \boldsymbol{u} &= \begin{pmatrix} \sin(x+t)\sin(y+t) \\ \cos(x+t)\cos(y+t) \end{pmatrix}, \\ p &= \sin(x-y+t). \end{aligned} \tag{24}$$

A third-order approximation for velocities and a second-order for pressure ($k=3$) are considered with a characteristic mesh size $h=0.25$.

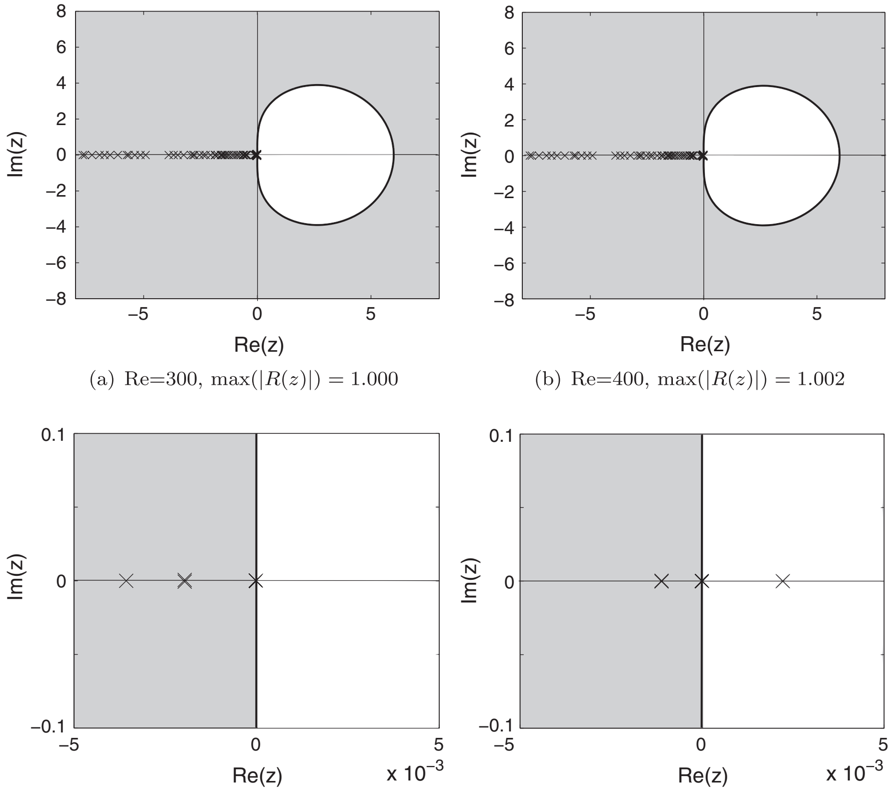
First, a non-skew-symmetric form is considered for the convective term (Equation (5)). Figure 2 shows the distribution of $\lambda\Delta t$, with a small time-increment $\Delta t = 0.001$, where λ are the eigenvalues of $(\mathbf{I}-\mathbf{H})\,\mathbf{A}$, for two Reynolds numbers, $Re = 300$ and $Re = 400$.

When a non-skew-symmetric convective term is used, some eigenvalues become positive when the Reynolds number is increased, entering the unstable zone of the two-stage Radau IIA-IRK method, as seen in Figure 2(d). Note that it is also possible to check whether the eigenvalues are all in the stable region by computing $\max(|R(z)|)$, where R is the stability function of the method; when $\max(|R(z)|) > 1$, the method is unstable.

Figure 3 shows the velocity vectors obtained at time $t=1$ for $Re = 300, 400$. Note that for $Re = 300$, the solution is stable because all the eigenvalues are in the stable part, $\max(|R(z)|) = 1.000$. Whereas for $Re = 400$, the solution obtained is not accurate because some eigenvalues are positive. In this case, $\max(|R(z)|) = 1.002$. This suggests that, in practice, the stability condition previously stated is a necessary and sufficient condition.

The skew-symmetric form (4a) is now used. A higher Reynolds number is considered, $Re = 1000$, to show that the scheme obtained is now unconditionally stable. Indeed, it can be seen in Figure 4 that all $\lambda\Delta t$, with a large time-increment $\Delta t = 0.1$, have negative real part and therefore remain in the stability region. In that case, the solution of the incompressible Navier–Stokes equations is unconditionally stable for any Reynolds number for the two-stage Radau IIA-IRK scheme.

Although the theoretical stability analysis provides a necessary condition for the asymptotic stability of the solution of the incompressible Oseen equations, numerical experiments show that, in



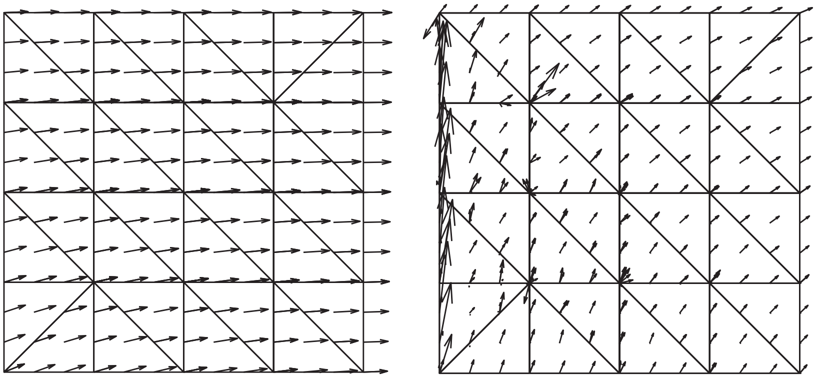
(a) Re=300, $\max(|R(z)|) = 1.000$

(b) Re=400, $\max(|R(z)|) = 1.002$

(c) Detail for Re=300

(d) Detail for Re=400

Figure 2. Distribution of $\lambda\Delta t$ marked with ×, using a non-skew-symmetric convective term, for $Re = 300$ (left), $Re = 400$ (right), $k = 3$, $h = 0.25$, $\Delta t = 0.001$, and stability border for two-stage Radau IIA implicit Runge–Kutta scheme. The gray part is the stability region.

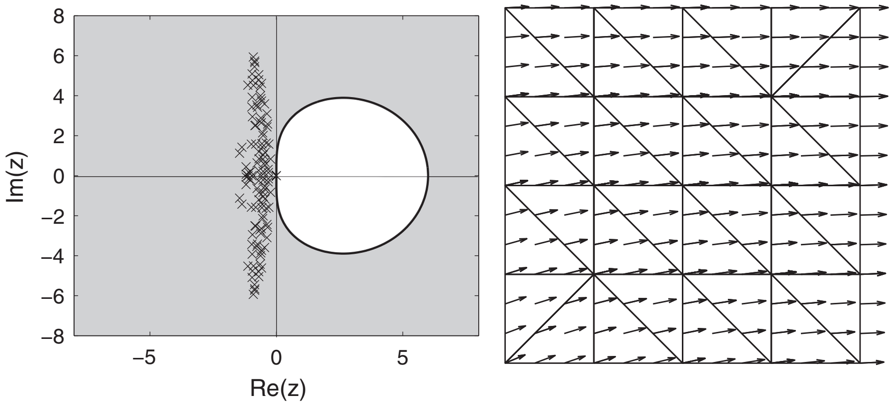


(a) $Re = 300$

(b) $Re = 400$

Figure 3. Velocity vectors at $t = 1$ for $Re = 300$, 400, for $k = 3$, $h = 0.25$, $\Delta t = 0.001$.

practice, this is actually also a sufficient condition. Moreover, the same results stand when applied to the incompressible Navier–Stokes equations, as long as the space discretization of the convective term is correctly implemented. Note that because the stability region of Rosenbrock methods includes the entire half complex plane with negative real part, the same unconditional stability properties apply.



(a) (b)

Figure 4. Distribution of $\lambda\Delta t$ marked with × (a) and velocity vectors (b) at $t = 1$, using a skew-symmetric convective term, for $Re = 1000$, $k = 3$, $h = 0.25$, and $\Delta t = 0.1$.

4. NUMERICAL EXAMPLES

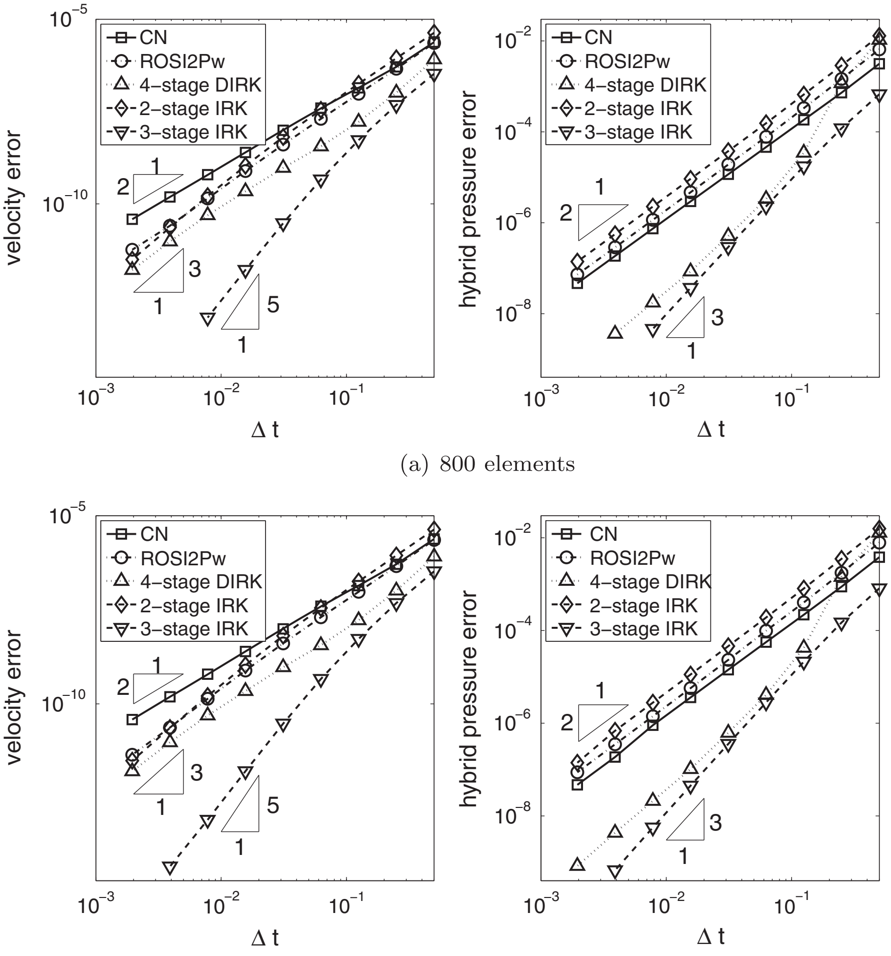
After the stability analysis, two numerical examples are considered to show the applicability of the proposed methods. An example with analytical solution is used first to compare Rosenbrock, DIRK, and IRK methods with a classical CN method from accuracy and cost points of view. The flow past a cylinder example is then used to further compare the two selected methods, four-stage Rosenbrock and three-stage IRK. In both examples, the IPM-DG formulation with piecewise solenoidal approximations described in Section 2 is employed. The goal of this section is to recommend high-order time-integration schemes, matching the spatial high accuracy obtained, thanks to the chosen DG formulation.

4.1. Runge–Kutta, Rosenbrock, and Crank–Nicolson accuracy and cost comparison

The unsteady example with analytical solution proposed in Section 3.3 is now used to compare the accuracy and relative cost of the proposed methods. Here, third-order methods, such as two-stage Radau IIA-IRK, four-stage DIRK, and four-stage Rosenbrock (ROSI2Pw), are compared with the three-stage Radau IIA-IRK and with a classical second-order CN method. Note that, as commented in Section 3.3, all methods are unconditionally stable for incompressible Navier–Stokes problems for the chosen discretization scheme. The goal of this section is thus to determine which method is more suitable to solve incompressible flow problems with high accuracy. Several third-order methods are compared with them and also with a fifth-order IRK method to see if its extra cost is balanced by the extra precision obtained.

Polynomial interpolation of degree $k = 4$ for velocity and 3 for pressure is chosen, and two uniform meshes are used, one of 800 elements (20,880 DOFs), where $h = 0.025$, and another one of 1800 elements (46,920 DOFs), where $h = 0.0167$. To avoid numerical error, the calculation is made until a final time $t = 1$. The initial condition prescribes the analytical solution (24) on the whole domain.

Figure 5 shows the evolution of the normalized $\mathcal{L}_2$-error (that is, the $\mathcal{L}_2$-error divided by the $\mathcal{L}_2$-norm of the exact velocity) under Δt refinement when solving Equation (6) for velocity and hybrid pressure for 800-element and 1800-element meshes. CN exhibits its theoretical convergence rate, 2 for both velocity and pressure. For velocity, slightly suboptimal convergence rates are obtained with two-stage IRK, four-stage DIRK, and Rosenbrock methods. Nevertheless, by decreasing the mesh size h from 0.025 to 0.0167, the slope of the convergence curves increases as Δt is refined, and consequently, convergence rates get closer to the optimal third order of convergence. Although the four-stage Rosenbrock method used here is expected to reach third order of accuracy for hybrid pressure [42], numerical examples only show second order, which is the same order as



(a) 800 elements

(b) 1800 elements

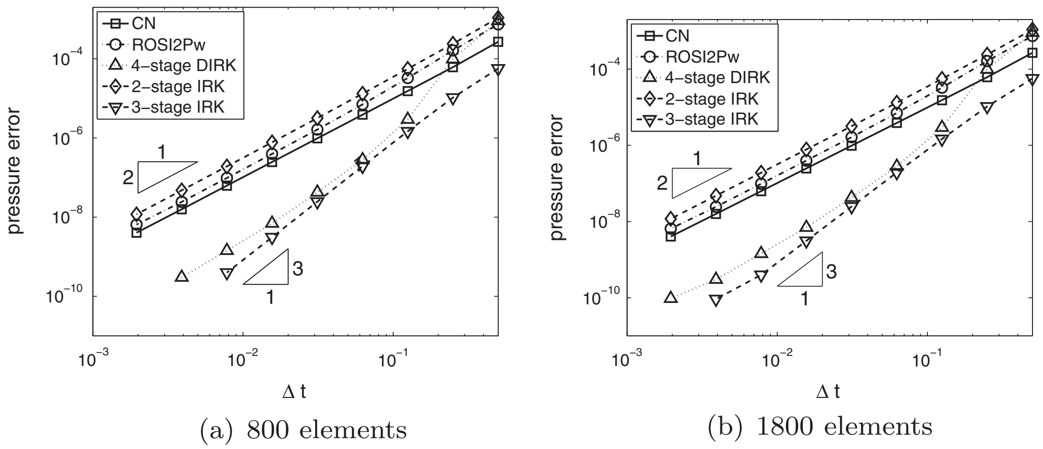
Figure 5. Unsteady analytical example: velocity and hybrid pressure $\mathcal{L}_2$-errors, for Crank–Nicolson (CN), four-stage Rosenbrock, four-stage diagonally implicit Runge–Kutta (DIRK), and two-stage and three-stage implicit Runge–Kutta (IRK) methods, $k = 4$.

the one expected and obtained for two-stage IRK and four-stage DIRK. Three-stage IRK shows the expected fifth order of convergence in velocity and third in pressure.

Figure 6 shows the evolution of the normalized $\mathcal{L}_2$-error of interior pressure under Δt refinement when solving the post-processing (7) using the time derivative fourth-order approximation (8). Note that for the different methods, the orders of convergence obtained for interior pressure are the same as the hybrid pressure ones.

In any case, as expected from the theoretical orders of convergence, for the same time step, clearly higher accuracy and convergence rate are obtained with the three-stage IRK method, and among the third-order methods, four-stage DIRK method is the most accurate.

Figure 5 shows how, for the same time step, the high-order three-stage IRK method provides higher accuracy compared with classical CN or any third-order methods. Nevertheless, it is also the most expensive method. Compared with CN, the three-stage IRK method requires three times more evaluations of the convective residue and leads to a three-time larger linear system of equations to be solved at each iteration. It is important to note that CN, four-stage DIRK, and two-stage and three-stage IRK methods all require computing the iterations of a nonlinear solver, here Broyden method, at each time step. This is not the case for Rosenbrock methods, which require four times more evaluations of the convective residue than CN and in which a linear system is to be solved at



(a) 800 elements

(b) 1800 elements

Figure 6. Unsteady analytical example: interior pressure $\mathcal{L}_2$-errors, for Crank–Nicolson (CN), four-stage Rosenbrock, four-stage diagonally implicit Runge–Kutta (DIRK), and two-stage and three-stage implicit Runge–Kutta (IRK) methods, $k = 4$.

each time step. Furthermore, although four-stage Rosenbrock methods require solving four linear systems in each time step, they have the same matrix; thus, the same factorization can be used with a computational time similar to the solution of the linear system to be solved for CN in each iteration and time step. Thus, Rosenbrock methods are promising in front of CN and may be competitive in front of DIRK and IRK. This is why after performing an accuracy study, it is now necessary to compare the computational costs of each method.

Figure 7 compares the normalized $\mathcal{L}_2$-errors of velocity and hybrid pressures obtained with CN, four-stage DIRK, four-stage Rosenbrock, and two-stage and three-stage Radau IIA-IRK methods as a function of cost. Results are depicted for both the 800-element (above) and the 1800-element (below) meshes. A Broyden method is used for the solution of the nonlinear system for CN, DIRK, and IRK, iterating until fulfilling a convergence criteria where the tolerance parameter for the stopping criteria is $\epsilon = c\Delta t^p$, with p the theoretical order of convergence of the method and c a positive constant. A direct solver is used for solving linear systems. Note that the cost here is defined as follows

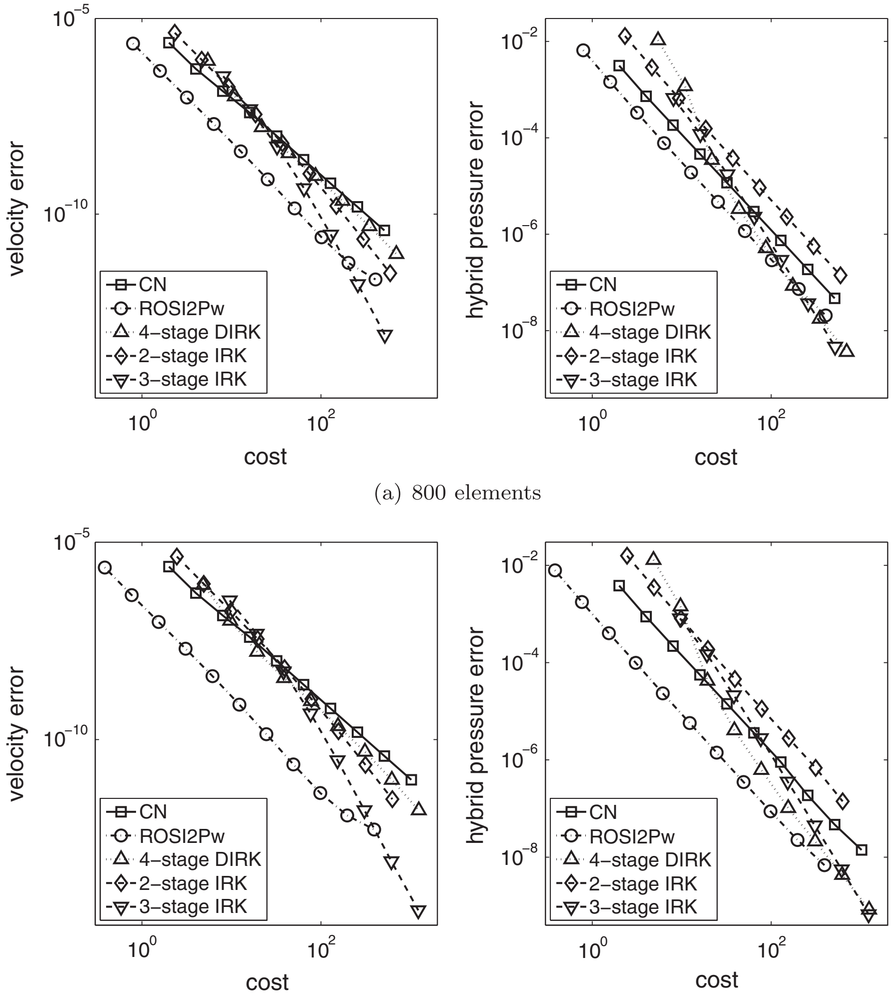
$$\text{cost} = \frac{\text{CPU time for 1 iteration with a given method}}{\text{CPU time for 1 iteration with CN}} \times \text{number of iterations}$$

It should be emphasized that the computing time depends on the implementation of the methods. The code used here is a research/development code, which surely can be further optimized. Nevertheless, all routines for the solution process (matrix and vector generation and assembly, linear solver) are the same for every method. Thus, it is expected that the correlation of CPU times is a fair comparison for the relative cost of each studied method. Moreover, results, which are consistent, are shown for the two used meshes.

Figure 7 shows that among the third-order methods, Rosenbrock is clearly the method performing with the highest efficiency, both for velocity and hybrid pressure. When comparing it with the three-stage Radau IIA-IRK method, it can be seen that at lower accuracy, Rosenbrock is also more efficient. But at high accuracy, the higher order of convergence for the three-stage Radau IIA-IRK balances its increased cost per iteration, and it becomes the most efficient method.

As the number of unknowns increases, for instance for a finer mesh, the size of the matrix also grows. This increment in the system size is much more important for the three-stage IRK compared with Rosenbrock method. Consequently, Rosenbrock methods become even more efficient at low accuracies. Only at high accuracy, high-order three-stage IRK outperforms the other methods giving a better precision-to-cost ratio.

Here, academic problems are used in the numerical examples, and consequently, direct solvers have been employed. However, to solve problems of practical engineering interest, iterative solvers



(a) 800 elements

(b) 1800 elements

Figure 7. Unsteady analytical example: velocity and hybrid pressure $\mathcal{L}_2$-errors, as a function of cost for Crank–Nicolson (CN), four-stage Rosenbrock, four-stage diagonally implicit Runge–Kutta (DIRK), and two-stage and three-stage implicit Runge–Kutta (IRK) methods, $k = 4$. Cost is defined as the ratio between the cost of one iteration of a given method and the cost of one iteration of CN multiplied by the number of time iterations.

are required. The efficiency of iterative solvers depends on the condition number of the resulting matrices. A comparison of condition numbers, depending on the method used and for various Reynolds numbers, is presented in Figure 8. These results have been obtained for the 1800-element mesh, $k = 4$, with $\Delta t = 0.01$ and at a time $t = 1$. For the Rosenbrock method, the considered matrix is the one resulting from Equation (18). For DIRK and IRK methods, the resulting nonlinear systems are solved using Broyden's method. The condition number considered is thus the one of the approximated Jacobian of the resulting system of equations. For DIRK, the matrix resulting from one stage, $i = 2, 3$, or 4, in Equation (14) is considered (recall that for DIRK methods, approximated Jacobians are independent of the stage number) and for IRK, the matrix resulting from the coupled system of equations, that is, $i = 1, 2$ for two-stage IRK and $i = 1, 2, 3$ for three-stage IRK. Similar results are obtained when considering the exact Jacobian instead of the approximated one. Figure 8 shows that the condition number for Rosenbrock and IRK methods decreases when the Reynolds number increases. The resulting matrix for three-stage IRK, which is the largest one, is the worst conditioned. DIRK presents better conditioning at low Reynolds number, but it then increases with

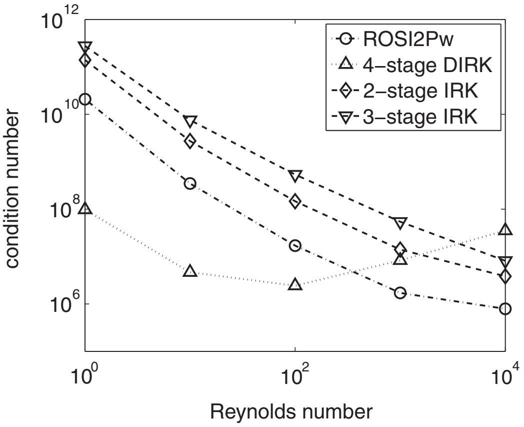


Figure 8. Condition number for Rosenbrock, diagonally implicit Runge–Kutta (DIRK), and implicit Runge–Kutta (IRK) methods, for 1800 elements, $k = 4$, $\Delta t = 0.01$ and at a time $t = 1$.

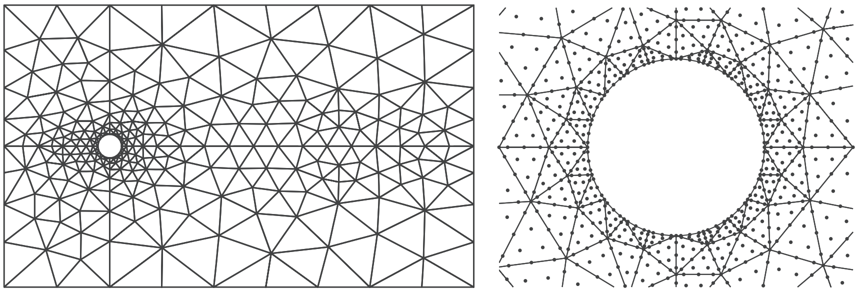
higher values of Reynolds number. On the whole, Rosenbrock thus exhibits the best conditioning: it is not negatively affected by the increase of the Reynolds number, and it shows lower values than IRK methods.

This example shows that three-stage IRK gives the most accurate solution but that among the proposed methods, four-stage Rosenbrock is the most performant. It is, in general, more efficient than three-stage IRK, in particular, when the size of the problem is increased, and its resulting matrix is also better conditioned. It is obvious that these are preliminary results for a 2D analytical case; further studies in 3D should confirm these conclusions. Meanwhile, to compare more deeply these two selected methods (four-stage Rosenbrock and three-stage IRK), the classical flow past a circle example is studied next.

4.2. Flow past a circle

In the present section, we consider a mixed Dirichlet/Neumann problem simulating the flow past a circle, with diameter $D = 1$, in a uniform stream. In this example, a high-order mesh generator EZ4U is used [44] because of its high-order export feature, which generates middle edge nodes over curves of the domain and inner face nodes that follow curved edges of the elements.

An unstructured mesh of 472 fourth-order elements is used, as seen in Figure 9 . These fourth-order elements are used for numerical integration and in the postprocessing. Fourth-order piecewise solenoidal approximation for the velocity ($k = 4$) and third-order for pressure are used in this computation. Dirichlet boundary condition $\boldsymbol{u}_D = (1, 0)$ is imposed on the inlet and no-slip condition, $\boldsymbol{u}_D = (0, 0)$, on the circle. Homogeneous Neumann conditions are imposed on the three other sides. Initial conditions prescribe a unitary velocity field $\boldsymbol{u}_0 = (1, 0)$ on the whole domain, except on the



(a) Mesh of the total domain (b) Zoom in the vicinity of the circle

Figure 9. Flow past a circle: unstructured mesh of 472 fourth-order elements.

circle boundary where $\boldsymbol{u}_0 = (0,0)$. The flow pattern depends on the Reynolds number defined here as $Re = u_\infty D/\nu$, where u_∞ is the mean fluid velocity, here $u_\infty = 1$.

For low Reynolds number ($1 \leqslant Re \leqslant 50$), it is well known that the solution reaches a stationary state. Here, a Reynolds number of $Re = 100$ is considered, leading to a periodic solution. First, a reference solution is calculated, using three-stage Radau IIA-IRK with a time step $\Delta t = 0.025$ on the time interval $[0,100]$ and a smaller $\Delta t = 0.005$ on $[100,120]$, to better capture the period of the periodic flow pattern. Once the flow passes the transient phase and reaches a periodic solution, vortex shedding is observed, that is, the flow detaches successively from the top and from the bottom of the circle, creating vortices. This happens in an alternating manner and this non-symmetric flow pattern is known as Von Karman vortex. Note that Figure 10 shows that these vortices are correctly captured even on a rather coarse mesh.

The periodic behavior of the solution can also be captured by the evolution of the lift coefficient C_L, which is defined by the following integral along the circle

$$C_L = \int_0^{2\pi} \boldsymbol{\tau}_y \mathrm{d}\boldsymbol{\theta}$$

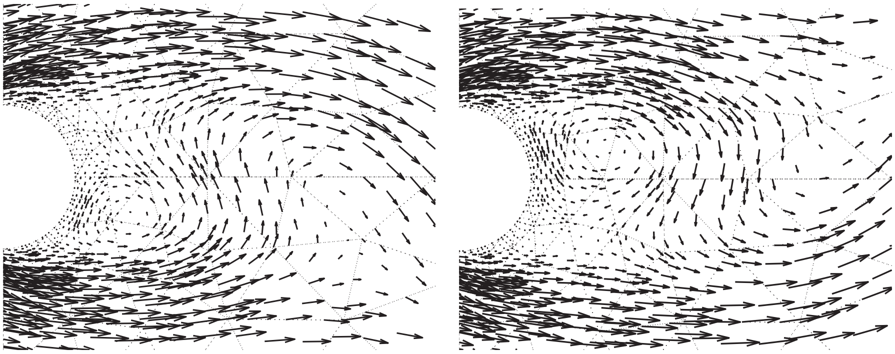
where $\boldsymbol{\tau}_y$ is the y-component of the normal component of the Cauchy stress tensor $\boldsymbol{\tau} = -p\boldsymbol{n} + 2\nu(\boldsymbol{n} \cdot \boldsymbol{\nabla}^{\mathrm{s}})\boldsymbol{u}$. Roshko [45] experimentally established the relation between the Strouhal number and the Reynolds number for flows past a circle and for Reynolds numbers between 90 and 150 as

$$S = 0.212\left(1 - \frac{21.2}{Re}\right). \tag{25}$$

The Strouhal number is a dimensionless number describing oscillating flow mechanisms, defined from the frequency of vortex shedding f_S as

$$S = \frac{f_S D}{u_\infty},$$

with D and u_∞ the characteristic length and velocity of the problem previously defined. Here, the measured period is $T = 5.96$, which corresponds to $S = 0.168$. Thus, it is in good agreement with experimental results and reported numerical simulations [45, 46], as shown in Table VIII.


(a) t=96

(b) t=105

Figure 10. Flow past a circle: velocity vectors in the vicinity of the circle for $Re = 100$, periodic phase, for the three-stage Radau IIA implicit Runge–Kutta reference solution, with a time step $\Delta t = 0.025$.

Table VIII. Flow past a circle: Strouhal number results for $Re = 100$.

	Three-stage IRK	Roshko (25)	Simo [46]
S	0.168	0.1671	0.167

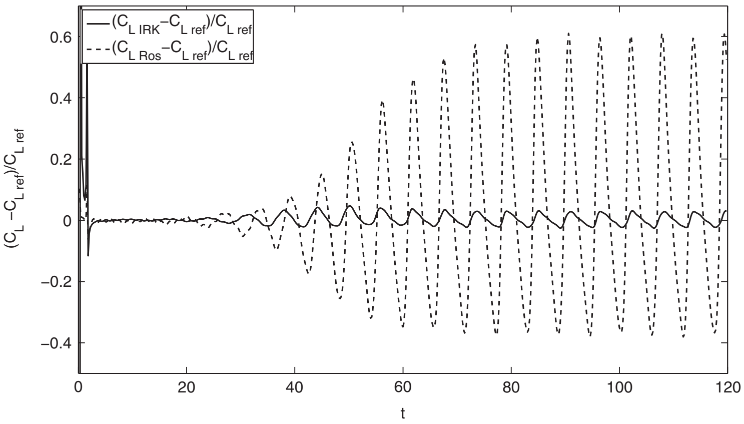


Figure 11. Flow past a circle: evolution of the relative error of lift coefficient with time, with respect to the reference implicit Runge–Kutta (IRK) solution with $\Delta t = 0.025$, for the IRK method with $\Delta t = 0.2$ and for the Rosenbrock method with $\Delta t = 0.05$.

Now, to further compare four-stage Rosenbrock and three-stage IRK methods, these two methods are considered for an equivalent computational cost. That is, a four times larger value of Δt is chosen for the three-stage IRK ($\Delta t = 0.2$) compared with a four-stage Rosenbrock ($\Delta t = 0.05$). Figure 11 shows, for the same computational cost, the relative error obtained with both methods relative to the reference solution. It can be seen that the Rosenbrock solution shows more noise in the transient phase and also presents a small phase shift once the periodic state is reached. Furthermore, the relative error for the Rosenbrock in the periodic solution is approximately 60%, whereas the error of IRK method is 5%. Note that if a reference solution is taken using a Rosenbrock method with $\Delta t = 0.01$, the same conclusions as with the IRK reference solution apply. In this example, using a three-stage IRK method allows obtaining higher accuracy, even with relatively larger values of Δt and at a competitive cost compared with Rosenbrock methods. Nevertheless, in larger or in 3D problems, Rosenbrock would compensate its lower accuracy by its higher efficiency.

5. CONCLUSIONS

The incompressible Navier–Stokes equations are interpreted as a system of DAE, that is, a system of ODEs corresponding to the conservation of momentum equation, plus algebraic constraints corresponding to the incompressibility condition. A high-order DG formulation with solenoidal approximations is used for space discretization, aiming to reach high orders of accuracy in space. Time-integration methods are then proposed to reach similar high order of accuracy in time. Within available RK methods, semi-implicit (DIRK) and fully implicit RK (IRK) methods are considered to solve this index-2 DAE system. In particular, between the available IRK schemes, Radau IIA-IRK methods are chosen because, for a given number of stages, they reach the highest order of convergence with the same order of convergence for velocity as for ODEs, and within the DIRK methods, a four-stage one is contemplated, reaching third order of convergence. A third-order four-stage Rosenbrock method, which avoids the solution of nonlinear systems at each time step, is also considered.

The unconditionally asymptotic stability of IRK and Rosenbrock schemes for DAE systems for incompressible Navier–Stokes problem is theoretically contemplated and then confirmed, as long as the space discretization is correctly implemented, through a numerical example.

A numerical example with analytical solution shows that four-stage Rosenbrock method stands out between third-order methods (such as four-stage DIRK and two-stage IRK) and that it is more efficient than CN method for the solution of incompressible Navier–Stokes problems. Although three-stage IRK performs very well when higher accuracy is needed, four-stage Rosenbrock shows an increasing efficiency with respect to three-stage IRK when the size of the problem gets bigger. The classical benchmark example of the flow past a circle confirms these results. It is obvious that these are preliminary results for a 2D analytical case. Further studies in 3D should confirm these conclusions.

APPENDIX A

A.1. Implementation of the semi-discretized system

In the following, solenoidal vector functions are discretized in each element Ω_k (for $k = 1, \dots, \mathtt{n_{el}}$) with a solenoidal vector basis $\boldsymbol{\phi}_i^k$ defined as

$$\boldsymbol{v} = \sum_{i=1}^{\mathtt{n_{bfu}}} \boldsymbol{\phi}_i^k \mathrm{v}_i^k \quad \text{in } \Omega_k,$$

with some scalar coefficients v_i^k, where $\mathtt{n_{bfu}}$ is the number of basis functions for the interpolation of the velocity in each element. The solenoidal discrete space in Ω_k is denoted as $\boldsymbol{\mathcal{S}}(\Omega_k) :=< \boldsymbol{\phi}_i^k >_{i=1}^{\mathtt{n_{bfu}}}$. Hybrid pressure is discretized on each side Γ_e (for $e = 1, \dots, \mathtt{n_{edge}}$) as

$$\tilde{p} = \sum_{i=1}^{\mathtt{n_{bfp}}} \psi_i^e \tilde{\mathrm{p}}_i^e \quad \text{on } \Gamma_e.$$

Moreover, for every side Γ_e or face in 3D, $\ell(e,1)$ and $\ell(e,2)$ respectively denote the numbers of the first element (left element) and the second element (right element) sharing the side. Figure A.I shows an example where side Γ_{13} is shared by elements Ω_{37} and Ω_{22}, thus for this side $\ell(13,1) = 37$ and $\ell(13,2) = 22$.

Implementation of Equation (6) is performed using this notation. For example, $\mathbf{M}^{\Omega_k}$, $\mathbf{K}^{\Omega_k}$ (for $k = 1, \dots, \mathtt{n_{el}}$), $\mathbf{K}^{\ell(e,\alpha),\ell(e,\beta)}$ and $\mathbf{G}^{\ell(e,\alpha),\Gamma_e}$ (for $\alpha, \beta = 1, 2$ and $e = 1, \dots, \mathtt{n_{edge}}$) are block matrices given by

$$\begin{aligned}
\left[\mathbf{M}^{\Omega_k}\right]_{ij} &= \int_{\Omega_k} \boldsymbol{\phi}_i^k \cdot \boldsymbol{\phi}_j^k \, \mathrm{d}\Omega, \quad \text{for } i, j = 1 \dots \mathtt{n_{bfu}}, \\
\left[\mathbf{K}^{\Omega_k}\right]_{ij} &= \int_{\Omega_k} 2\nu \nabla^{\mathrm{s}} \boldsymbol{\phi}_i^k \boldsymbol{:} \nabla^{\mathrm{s}} \boldsymbol{\phi}_j^k \, \mathrm{d}\Omega, \quad \text{for } i, j = 1 \dots \mathtt{n_{bfu}}, \\
\left[\mathbf{K}^{\ell(e,\alpha),\ell(e,\beta)}\right]_{ij} &= C_{11} \int_{\Gamma_e} \mathbf{n}_{\ell(e,\alpha)} \otimes \boldsymbol{\phi}^{\ell}(e,\alpha)_i \boldsymbol{:} (\boldsymbol{n}_{\ell(e,\beta)} \otimes \boldsymbol{\phi}_j^{\ell(e,\beta)}) \, \mathrm{d}\Gamma \\
&\quad - \int_{\Gamma_e} \nu (\nabla^{\mathrm{s}} \boldsymbol{\phi}_i^{\ell(e,\alpha)}) \boldsymbol{:} (\boldsymbol{n}_{\ell(e,\beta)} \otimes \boldsymbol{\phi}_j^{\ell(e,\beta)}) \, \mathrm{d}\Gamma \\
&\quad - \int_{\Gamma_e} \nu \mathbf{n}_{\ell(e,\alpha)} \otimes \boldsymbol{\phi}_i^{\ell(e,\alpha)} \boldsymbol{:} (\nabla^{\mathrm{s}} \boldsymbol{\phi}_j^{\ell(e,\beta)}) \, \mathrm{d}\Gamma \quad \text{for } i, j = 1 \dots \mathtt{n_{bfu}}, \\
\left[\mathbf{G}^{\ell(e,\alpha),\Gamma_e}\right]_{ij} &= \int_{\Gamma_e} (\boldsymbol{n}_{\ell(e,\alpha)} \otimes \boldsymbol{\phi}_i^{\ell(e,\alpha)}) \psi_j \quad \text{for } i = 1 \dots \mathtt{n_{bfu}}, j = 1 \dots \mathtt{n_{bfp}}.
\end{aligned}$$

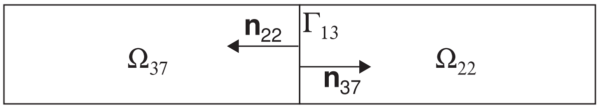


Figure A.I. Elements Ω_{37} and Ω_{22} share face Γ_{13}; $\boldsymbol{n}_{37}$ and $\boldsymbol{n}_{22}$ are respectively exterior unit normals to Ω_{37} and Ω_{22}.

Matrices $\mathbf{M}$, $\mathbf{K}$, and $\mathbf{G}$ of respective sizes $(\mathtt{n_{el}n_{bfu}} \times \mathtt{n_{el}n_{bfu}})$, $(\mathtt{n_{el}n_{bfu}} \times \mathtt{n_{el}n_{bfu}})$, and $(\mathtt{n_{el}n_{bfu}} \times \mathtt{n_{edge}n_{bfp}})$ are then assembled. Note that the convection matrix $\mathbf{C}$ and the vectors of nodal values, $\mathbf{f_1}$ and $\mathbf{f_2}$, or the approximation coefficients of velocity and pressure respectively, are computed in a similar way to obtain the discretized form (10).

ACKNOWLEDGEMENTS

This research is funded by Ministerio de Educación y Ciencia grant no. DPI2011-27778-C02-02 and Generalitat de Catalunya AGAUR grant no. 2009SGR875.

REFERENCES

1. Chorin A. Numerical solution of the Navier-Stokes equations. *Mathematics of Computation* 1968; **22**(104):745–762.
2. Temam R. *Navier-Stokes Equations: Theory and Numerical Analysis*. AMS Chelsea Publishing: Providence, RI, 2001.
3. Donea J, Giuliani S, Laval H, Quartapelle L. Finite element solution of the unsteady Navier-Stokes equations by a fractional step method. *Computer Methods in Applied Mechanics and Engineering* 1981; **30**:53–73.
4. Kim J, Moin P. Application of a fractional-step method to incompressible Navier-Stokes equations. *Journal of Computational Physics* 1985; **59**:308–323.
5. Guermond J, Minev P, Shen J. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:6011–6045.
6. Houzeaux G, Vázquez M, Aubry R, Cela JM. A massively parallel fractional step solver for incompressible flows. *Journal of Computational Physics* 2009; **228**(17):6316–6332.
7. Kim K, Baek SJ, Sung HJ. An implicit velocity decoupling procedure for the incompressible Navier-Stokes equations. *International Journal of Numerical Methods in Fluids* 2002; **38**(2):125–138.
8. Jansen K, Whiting C, Hulbert G. A generalized-α method for integrating the filtered Navier-Stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering* 2000; **190**(3):305–319.
9. Persson PO, Peraire J. Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier-Stokes equations. *SIAM Journal on Scientific Computing* 2008; **30**(6):2709–2733.
10. Bassi F, Rebay S. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics* 1997; **131**(2):267–279.
11. Wang L, Mavriplis DJ. Implicit solution of the unsteady Euler equations for high-order accurate discontinuous Galerkin discretizations. *Journal of Computational Physics* 2007; **225**(2):1994–2015. DOI: 10.1016/j.jcp.2007.03.002.
12. Karniadakis GE, Israeli M, Orszag S. High-order splitting methods for the incompressible Navier-Stokes equations. *Journal of Computational Physics* 1991; **97**:414–443.
13. Kaps P, Rentrop P. Generalized Runge-Kutta methods of order four with stepsize control for stiff ordinary differential equations. *Numerische Mathematik* 1979; **33**:55–68.
14. Hairer E, Wanner G. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer-Verlag: Berlin, 1996.
15. Étienne S, Garon A, Pelletier D. Perspective on the geometric conservation law and finite element methods for ALE simulations of incompressible flow. *Journal of Computational Physics* 2009; **228**:2313–2333.
16. Montlaur A, Fernandez-Mendez S, Huerta A. Métodos Runge-Kutta implícitos de alto orden para flujo incompresible. *Revista Internacional Métodos numéricos para cálculo y diseño en ingeniería* 2011; **27**(1):77–91.
17. John V, Rang J. Adaptive time step control for the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**:514–524.
18. Montlaur A, Fernandez-Mendez S, Huerta A. Discontinuous Galerkin methods for the Stokes equations using divergence-free approximations. *International Journal for Numerical Methods in Fluids* 2008; **57**(9):1071–1092.
19. Montlaur A, Fernandez-Mendez S, Peraire J, Huerta A. Discontinuous Galerkin methods for the Navier-Stokes equations using solenoidal approximations. *International Journal for Numerical Methods in Fluids* 2010; **64**(5):549–564.
20. Hanke M, März R. On asymptotics in case of DAE's. *Zeitschrift für Angewandten Mathematik und Mechanik (ZAMM)* 1996; **76**(Suppl.1):99–102.
21. Hanke M, Izquierdo E, März R. On asymptotics in case of linear index-2 differential-algebraic equations. *SIAM Journal on Numerical Analysis* 1998; **35**(4):1326–1346.
22. Stenberg R. Mortaring by a method of J. A. Nitsche. *Computational mechanics (Buenos Aires, 1998)*, Centro Internac. Métodos Numér. Ing.: Barcelona, 1998; CD–ROM file.
23. Hansbo A, Hansbo P. A finite element method for the simulation of strong and weak discontinuities in solid mechanics. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**(33-35):3523–3540.
24. Kanschat G, Schötzau D. Energy norm a posteriori error estimation for divergence-free discontinuous Galerkin approximations of the Navier-Stokes equations. *International Journal for Numerical Methods in Fluids* 2008; **57**(9):1093–1113.
25. Donea J, Huerta A. *Finite Element Methods for Flow Problems*. John Wiley & Sons: Chichester, 2003.

26. Cockburn B, Gopalakrishnan J. Incompressible finite elements via hybridization. Part I: the Stokes system in two space dimensions. *SIAM Journal on Numerical Analysis* 2005; **43**(4):1627–1650.
27. Carrero J, Cockburn B, Schötzau D. Hybridized globally divergence-free LDG methods. Part I: the Stokes problem. *Mathematics of Computation* 2005; **75**(254):533–563.
28. Baker GA, Jureidini WN, Karakashian OA. Piecewise solenoidal vector fields and the Stokes problem. *SIAM Journal on Numerical Analysis* 1990; **27**(6):1466–1485.
29. Hairer E, Lubich C, Roche M. *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*. Springer-Verlag: Berlin, 1989.
30. Brenan K, Campbell S, Petzold L. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. SIAM: Philadelphia, PA, 1996.
31. Brenan KE, Engquist BE. Backward differentiation approximations of nonlinear differential/algebraic systems. *Mathematics of Computation* 1988; **51**(184):659–676. (Available from: http://www.jstor.org/stable/2008768).
32. John V, Matthies G, Rang J. A comparison of time-discretization/linearization approaches for the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**(44-47):5995–6010.
33. Ostermann A, Roche M. Rosenbrock methods for partial differential equations and fractional orders of convergence. *SIAM Journal on Numerical Analysis* 1993; **30**(4):1084–1098.
34. Lubich C, Ostermann A. Linearly implicit time discretization of non-linear parabolic equations. *IMA Journal of Numerical Analysis* 1995; **15**(4):555–583. DOI: 10.1093/imanum/drm038.
35. Pereira JMC, Kobayashi M, Pereira JCF. A fourth-order-accurate finite volume compact method for the incompressible Navier-Stokes solutions. *Journal of Computational Physics* 2001; **167**(1):217–243.
36. Montlaur A. High-order discontinuous Galerkin methods for incompressible flows. *Ph.D. Thesis*, Universitat Politènica de Catalunya, 2009. (Available from: http://www.tesisenxarxa.net/TDX-0122110-183128).
37. Butcher J. *The Numerical Analysis of Ordinary Differential Equations*. Wiley: Chichester, 1987.
38. Norsett S. One-step methods of hermite type for numerical integration of stiff systems. *BIT* 1974; **14**:63–77.
39. Alexander R. Diagonally implicit Runge-Kutta methods for stiff O.D.E.'s. *SIAM Journal on Numerical Analysis* 1977; **14**(6):1006–1021.
40. Alexander R. Design and implementation of DIRK integrators for stiff systems. *Applied Numerical Mathematics* 2003; **46**(1):1–17.
41. Williams R, Burrage K, Camerona I, Kerr M. A four-stage index 2 diagonally implicit Runge–Kutta method. *Applied Numerical Mathematics* 2002; **40**:415–432.
42. Rang J, Angermann L. New Rosenbrock methods of order 3 for PDAES of index 2. *Proceedings of Equadiff-11*, Bratislava, Slovakia, 2005; 385–394.
43. Steinebach G. Order-reduction of ROW-methods for DAEs and Method of Lines Applications. Preprint-Nr 1741, Technische Universität Darmstadt, 1995.
44. Roca X, Sarrate J, Ruiz-Gironès E. A graphical modeling and mesh generation environment for simulations based on boundary representation data. *Congresso de Métodos Numéricos em Engenharia*, Porto, Portugal, 2007 ; CD-ROM file.
45. Roshko A. On the development of turbulent wakes from vortex streets. *NACA Report 1191 (formerly TN-2913)*, 1954.
46. Simo J, Armero F. Unconditional stability and long-term behaviour of transient algorithms for the incompressible Navier–Stokes and Euler equations. *Computer Methods in Applied Mechanics and Engineering* 1994; **111**:111–154.