



Low Mach preconditioned density-based methods with implicit Runge–Kutta schemes in physical-time

Leonardo Santos de Brito Alves¹ · Ricardo Dias dos Santos¹ · Carlos Eduardo Guex Falcão²

Received: 20 August 2020 / Accepted: 28 May 2021 / Published online: 12 June 2021
© The Brazilian Society of Mechanical Sciences and Engineering 2021

Abstract

As far as the authors are aware, low Mach preconditioned density-based methods found in the peer-reviewed literature only employ multi-step schemes for physical-time integration. This essentially limits the maximum achievable temporal accuracy-order of these methods to two, since the multi-step schemes of order higher than two are conditionally stable. However, the present paper shows how these methods can employ multi-stage schemes in physical-time using the same low Mach preconditioning techniques developed over the past few decades. In doing so, it opens up the rich field of Runge–Kutta time integration schemes to low Mach preconditioned density-based methods. One and two-dimensional test cases are used to demonstrate the capabilities of this novel approach. The former simulates the propagation of marginally stable entropy perturbations superposed on a uniform flow whereas the latter simulates the temporal growth of vorticity perturbations superposed on an absolutely unstable planar mixing-layer. A novel procedure is employed to generate highly accurate initial conditions for the two-dimensional test case, it minimizes receptivity regions as well as deleterious interactions with artificial boundary conditions due to the numerical error introduced by approximate initial conditions. These test cases show that second, third and fourth order multi-stage schemes with strong linear numerical stability can be successfully utilized for the physical-time integration of low Mach preconditioned density-based methods.

Keywords Unsteady compressible flows · Very low Mach number flows · Multi-stage schemes · High-order physical-time integration · Strong linear numerical stability · Dual time stepping

1 Introduction

After decades of evolution, modern computers can now perform very complex unsteady three-dimensional numerical simulations. Even so, selecting a time-marching scheme is still of great importance being a compromise between accuracy and efficiency. Furthermore, both are controlled

by stiffness to a great extent, which is caused by disparities in characteristic time and/or length scales. One of the major difficulties faced by researchers performing simulations of very low speed compressible flows is the stiffness due to large differences between convective and acoustic velocities. It should be noted that these speeds are not only found in incompressible flows. Compressibility can still present in very low Mach number flows in the presence of, for instance, strong temperature variations [50]. Nevertheless, stiffness decreases efficiency. Realistic computer times are only possible in such cases with numerical schemes that are at least A-stable, a mathematical property defined as unconditional stability when solving a linear and homogeneous standard test problem. An additional difficulty associated with low Mach number flows is the fact that pressure is no longer a thermodynamic state variable, but only a Lagrange multiplier. Hence, one must solve for it directly, instead of recovering it from an equation of state, to avoid significant round-off error propagation. In other words, stiffness decreases accuracy. Arguably the two main approaches advocated to

Technical Editor: Monica Carvalho.

✉ Leonardo Santos de Brito Alves
lsbalves@id.uff.br

Ricardo Dias dos Santos
ricardos@id.uff.br

Carlos Eduardo Guex Falcão
carlosfalcao@mecanica.ufsm.br

¹ PGMEC/TEM/UFF, Rua Passo da Pátria 156, Bloco E, Sala 211, São Domingos, Niterói, RJ 24210-240, Brazil

² PGMEC/UFSM, Cidade Universitária - Camobi, Av. Roraima 1000, Santa Maria, RS 97105-900, Brazil

deal with these difficulties and simulate the compressible Navier-Stokes equations at low Mach numbers are known as pressure-based and density-based methods [39].

Pressure-based methods are extensions of incompressible flow solvers towards the high Mach regime. Their origins can be traced back to the marker in cell (MAC) [27] and pressure correction [16] methods for incompressible flow simulations. SIMPLE is arguably the most used variant known to date [52]. They were transformed into compressible flow solvers in the early seventies [28]. However, research on this subject laid dormant for more than fifteen years, until the creation of PISO [34] and SIMPLEC [36]. Recent work has addressed many different issues, such as zero Mach number singularity [7, 30], low Mach number stiffness [56] and discrete conservation of mass, momentum and energy equations at arbitrary Mach numbers [33]. Both incompressible and compressible versions mentioned so far utilize a time-marching approach known as projection or fractional step method, which is based on a Helmholtz-Hodge decomposition [19]. An extensive review including all different variants of projection methods [26] has shown that the pressure-velocity decomposition is unconditionally stable only up to second-order accuracy. Recent attempts to reach higher orders have utilized stable pressure extrapolation schemes [22], but accuracy-order was problem dependent and barely above two. Simulations were limited to short time spans.

On the other hand, density-based methods are extensions of compressible flow solvers towards the low Mach regime. Two popular variants exist. The first one was derived in the early eighties and is known today as either acoustic filtering [53] or low Mach number asymptotic Navier-Stokes equations [40]. It employs a perturbation expansion procedure to obtain asymptotic forms of the compressible flow equations using the Mach number to the first [25] or second [47] powers as a small parameter. They are used to this day for complex problems [11] and extensive reviews exist in the literature [49], but these methodologies apply only to low Mach number flows. They are not an arbitrary Mach compressible flow solver such as pressure-based methods. However, their time-marching schemes are also based on projection methods. Hence, they suffer from the same low temporal resolution limitations described earlier.

Preconditioned density-based methods provide another alternative. Their origins trace back to the artificial compressibility for incompressible flows [15]. This idea was extended towards compressible flows twenty years later [65], allowing traditional compressible solvers to simulate low Mach number flows as well. An important contribution was the separation of static (thermodynamic) and dynamic (hydrodynamic) pressure contributions, allowing much lower Mach numbers to be reached [48]. These methods have been optimized for high Reynolds number flows [14],

reacting flows with non-equilibrium chemistry [60], simulations with highly stretched meshes [9], implicit upwind solvers [10], different variable sets [68], reduced eigenvector orthogonality losses [18, 29], supercritical heat transfer [64] and improved energy conservation accuracy at very low Mach numbers [43, 44]. Extensive literature reviews can be found elsewhere [66, 67, 69]. The introduction of low Mach preconditioning into the governing equations alters their physical-time evolution, although the correct steady-state is always obtained. In order to recover the time accuracy of low Mach preconditioned equations, dual time stepping [32] must be used as well. This technique was originally introduced with artificial compressibility [46] and is still used to this day for unsteady flows [21]. However, low Mach number preconditioned density-based methods are usually limited to second-order accuracy in physical-time [62]. In fact, the authors have not been able to find a low Mach preconditioned density-based method in the peer-reviewed journal literature that has an accuracy-order higher than two in physical-time [3]. This restriction is due to the fact that these methods always employ multi-step schemes for an accurate temporal resolution, but there are no A-stable multi-step schemes with accuracy-orders greater than two [17]. Higher order versions are available, but they are all conditionally stable [45].

The limitation on the stability of high-order implicit multi-step schemes led the study of multi-stage schemes to minimize the CFL restrictions imposed by stiffness, [8, 35, 70], since these schemes, known as implicit Runge-Kutta (IRK) methods [12], were capable of simulating unsteady compressible flows with high Mach numbers [6, 20, 38, 51, 73–75]. IRK methods can achieve high accuracy-order with strong numerical stability increasing the number of intermediate stages within the physical-time step. These methods are self-starting as well, and easy to implement with variable time-step sizes, which is not the case for multi-step schemes.

However, not even the stronger numerical stability of IRK methods is able to control the stiffness found in very low Mach number flows, where $M \ll 1$, evidenced by the fact that lower bounds in previously mentioned studies were limited to $M \sim 0.05$. The main goal of the present study is to demonstrate how the well-known low Mach preconditioning techniques can be employed with these multi-stage schemes as well. This is achieved in such a way that no modifications whatsoever are required of the preconditioning matrix and preconditioned artificial dissipation for spatial resolution, allowing a straightforward use of all scientific developments related to preconditioned density-based methods achieved over the past twenty-five years. The present work provides the framework for low Mach preconditioned density-based methods to be employed with multi-stage schemes, allowing high-order unsteady simulations with strong numerical stability of

very low Mach number compressible flows. Several test cases simulated with Mach numbers as low as $M = 10^{-5}$ are presented, providing strong evidence for the improved efficiency and accuracy of this novel approach when compared to non-preconditioned multi-stage methods and low Mach preconditioned multi-step schemes.

2 Governing equations

2.1 Unsteady governing equations

Consider the 2D compressible unsteady Euler equations from fluid dynamics

$$\frac{\partial \mathbf{q}}{\partial t} = \mathbf{f}(\mathbf{q}) \quad \text{where} \quad \mathbf{f}(\mathbf{q}) = -\left(\frac{\partial \mathbf{e}_i}{\partial \xi} + \frac{\partial \mathbf{f}_i}{\partial \eta}\right), \quad (1)$$

where t is the physical-time independent variable and ξ and η are the independent spatial variables in computational space. Their differentials are related to their counterparts in physical space, x and y , through the Jacobian determinant inverse

$$\frac{1}{\mathcal{J}} = \frac{\partial(\xi, \eta)}{\partial(x, y)} = \begin{vmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{vmatrix} = \frac{\partial \xi}{\partial x} \frac{\partial \eta}{\partial y} - \frac{\partial \xi}{\partial y} \frac{\partial \eta}{\partial x}. \quad (2)$$

Furthermore, the generalized dependent variable \mathbf{q} and inviscid fluxes \mathbf{e}_i and \mathbf{f}_i in the steady-state residue $\mathbf{f}(\mathbf{q})$ are related to their counterparts in physical space through

$$\mathbf{q} = \mathcal{J}\hat{\mathbf{Q}}, \quad (3)$$

$$\mathbf{e}_i = \mathcal{J}\mathbf{E}_i \frac{\partial \xi}{\partial x} + \mathcal{J}\mathbf{F}_i \frac{\partial \xi}{\partial y}, \quad (4)$$

$$\mathbf{f}_i = \mathcal{J}\mathbf{E}_i \frac{\partial \eta}{\partial x} + \mathcal{J}\mathbf{F}_i \frac{\partial \eta}{\partial y}, \quad (5)$$

which, in turn, are defined in conservative form as

$$\mathbf{Q} = (\rho, \rho u, \rho v, \rho E)^T, \quad (6)$$

$$\mathbf{E}_i = (\rho u, \rho u^2 + P, \rho uv, (\rho E + P)u)^T, \quad (7)$$

$$\mathbf{F}_i = (\rho v, \rho uv, \rho v^2 + P, (\rho E + P)v)^T, \quad (8)$$

where ρ stands for density, u for stream wise velocity, v for cross stream velocity, $E = e + (u^2 + v^2)/2$ for total energy per unit mass, e for thermal internal energy per unit mass and P as pressure.

2.2 Controlling roundoff error propagation

Low Mach number simulations using compressible solvers bring some limitations. In the low Mach number limit, density becomes independent from pressure. Therefore one has to solve for the latter instead of the former. Furthermore, pressure should be decomposed into $P = P_T + P_H$, which is the sum of its hydrodynamic ($P_H \sim \rho u^2$) and thermodynamic ($P_T \sim \rho c^2$) contributions, since the latter is essentially constant and orders of magnitude higher than the former when $M \ll 1$. This is essential in order to control the propagation of pressure roundoff errors in the momentum conservation equations, where P must be replaced by P_H . It should be noted that P_T is constant in space, but allowed to vary in time as is often the case in fully bounded flows. Nevertheless, here P_T is assumed to be constant. Hence, a simple chain rule is applied to Eq. (1), leading to

$$\mathbf{T} \frac{\partial \hat{\mathbf{q}}}{\partial t} = \mathbf{f}(\mathbf{q}), \quad (9)$$

which solves for the primitive dependent variable vectors

$$\hat{\mathbf{q}} = \mathcal{J}\hat{\mathbf{Q}} \quad \text{and} \quad \hat{\mathbf{Q}} = (P_H, u, v, T)^T, \quad (10)$$

where the conservative to primitive variable Jacobian \mathbf{T} that allows this change in variables is given by

$$\mathbf{T} = \frac{\partial \mathbf{q}}{\partial \hat{\mathbf{q}}} = \frac{\partial \mathbf{Q}}{\partial \hat{\mathbf{Q}}} = \begin{pmatrix} \frac{\partial \rho}{\partial P} & 0 & 0 & \frac{\partial \rho}{\partial T} \\ u \frac{\partial \rho}{\partial P} & \rho & 0 & u \frac{\partial \rho}{\partial T} \\ v \frac{\partial \rho}{\partial P} & 0 & \rho & v \frac{\partial \rho}{\partial T} \\ H \frac{\partial \rho}{\partial P} + \rho \frac{\partial h}{\partial P} - 1 & \rho u & \rho v & H \frac{\partial \rho}{\partial T} + \rho \frac{\partial h}{\partial T} \end{pmatrix}, \quad (11)$$

where T stands for temperature, $H = h + (u^2 + v^2)/2$ total enthalpy per unit mass and $h = e + P/\rho$ enthalpy per unit mass. In $\hat{\mathbf{Q}}$ temperature can be replaced by enthalpy or entropy, among others. Density and enthalpy dependencies on pressure and temperature are determined from an equation of state and only thermally perfect gases are going to be considered here, although arbitrary equations of state can be used.

2.3 Removing convective/acoustic time scale stiffness

Despite these modifications, Eq. (9) becomes increasingly stiff as the Mach number is decreased below $M \sim 0.1$. The same is true for Eq. (1). This means that the time steps required to march either equation in time would have to be on the order of the acoustic time scales, even though

acoustics generally does not affect the flow at these very low Mach numbers. Hence, a prohibitively large number of time steps would be required to reach steady-state. In order to prevent such a strong time step restriction and accelerate convergence towards steady-state, a second technique is required, i.e., low Mach preconditioning. It works by replacing conservative to primitive variable Jacobian \mathbf{T} by a preconditioning matrix $\mathbf{\Gamma}$, leading to

$$\mathbf{\Gamma} \frac{\partial \hat{\mathbf{q}}}{\partial \tau} = \mathbf{f}(\mathbf{q}), \tag{12}$$

where $\mathbf{\Gamma}$ can be generally defined as

$$\mathbf{\Gamma} = \begin{pmatrix} \rho_P & 0 & 0 & \rho_T \\ u \rho_P & \rho & 0 & u \rho_T \\ v \rho_P & 0 & \rho & v \rho_T \\ H \rho_P + \rho h_P - 1 & \rho u & \rho v & H \rho_T + \rho h_T \end{pmatrix}, \tag{13}$$

whose density and enthalpy dependencies on pressure and temperature are modified to

$$\begin{aligned} \rho_P &= \frac{1}{c_p^2} - \rho_T \left(\frac{1 - \rho h_P}{\rho h_T} \right), & \rho_T &= \delta \frac{\partial \rho}{\partial T}, \\ h_P &= \frac{\partial h}{\partial P} & \text{and} & \quad h_T = \frac{\partial h}{\partial T}, \end{aligned} \tag{14}$$

with the pseudo-speed of sound c_p taken from [69] and $\delta = 1$ [71]. Doing so modifies the eigenvalues λ_i of Eq. (9), e.g. given by $\lambda_1 = \lambda_2 = u$ and $\lambda_{3,4} = u \pm c$ in the x-direction, to $\tilde{\lambda}_i \simeq O(u)$ in Eq. (12) even when $M \ll 1$, i.e. $c \gg |u|$. The exact expressions for $\tilde{\lambda}_i$ can be found in these references. It should be noted that the physical-time t in Eq. (9) was replaced by the pseudo-time τ in Eq. (12) to highlight that the time evolution in the latter is no longer accurate. This is not relevant when one is searching for steady-states that satisfy $\mathbf{f}(\mathbf{q}) = 0$, since they are not modified by low Mach preconditioning. Doing so by integrating Eq. (12) instead of Eq. (9), however, is significantly more efficient since the latter is stiff but the former is not.

2.4 Recovering time accuracy

If time accurate simulations are still desired when $M \ll 1$, Eq. (12) must be modified. In other words, if low Mach stiffness must be removed and time accuracy must be maintained, one needs to re-introduce the original physical-time derivative that was present in Eq. (1) into Eq. (12), yielding

$$\mathbf{\Gamma} \frac{\partial \hat{\mathbf{q}}}{\partial \tau} + \frac{\partial \mathbf{q}}{\partial t} = \mathbf{f}(\mathbf{q}), \tag{15}$$

a technique known as dual time stepping (DTS). It should be noted that the physically correct unsteady-state that satisfies Eq. (1) is only recovered from Eq. (15) at pseudo-time

steady-state. Doing so with Eq. (15) requires reaching the pseudo-time steady-state within each physical-time step. Equation (15) is the classical equation all low Mach number preconditioned density-based methods solve to simulate unsteady flows. It must be employed instead of Eqs. (1) or (9), otherwise the same stiffness issues due to the large disparity between acoustic and convective time scales, already discussed in the previous subsection, will also occur. Hence, using Eq. (15) instead is much more efficient, despite the additional pseudo-time iterations per physical-time step. These additional iterations can be minimized by using special marching schemes designed for the calculation of steady-states [5] when performing each pseudo-time integration.

3 Temporal resolution

3.1 Physical-time integration with multi-step schemes

A careful review of the literature shows that Eq. (15) is always solved by relocating the physical-time derivative to the right hand side of the equation and treating it as a source term,

$$\mathbf{\Gamma} \frac{\partial \hat{\mathbf{q}}}{\partial \tau} = \tilde{\mathbf{f}}(\mathbf{q}) = \mathbf{f}(\mathbf{q}) - \frac{\partial \mathbf{q}}{\partial t}, \tag{16}$$

and applying multi-step schemes to the unsteady residue $\tilde{\mathbf{f}}(\mathbf{q})$. In the vast majority of cases, a second-order backwards differentiation formula (BDF2) is employed,

$$\tilde{\mathbf{f}}(\mathbf{q}^{n+1}) \simeq \mathbf{f}(\mathbf{q}^{n+1}) - \frac{3 \mathbf{q}^{n+1} - 4 \mathbf{q}^n + \mathbf{q}^{n-1}}{2 \Delta t}, \tag{17}$$

although the second-order Crank-Nicholson (CN2) scheme,

$$\tilde{\mathbf{f}}(\mathbf{q}^{n+1}) \simeq \frac{1}{2} (\mathbf{f}(\mathbf{q}^{n+1}) + \mathbf{f}(\mathbf{q}^n)) - \frac{\mathbf{q}^{n+1} - \mathbf{q}^n}{\Delta t}, \tag{18}$$

has been used as well. In both approximations, n (as well as $n - 1$) and $n + 1$ represent the known and unknown indexes in physical-time, respectively, and Δt is the physical-time step. BDF2 is the preferred choice because of its L-stability. Despite being only A-stable, CN2 can still be an interesting choice because of its non-dissipative nature. L-stability is also called strong A-stability, since it is equivalent to A-stability but with the added property of zero, as opposed to bounded, gain at infinite CFL numbers. In general, $\tilde{\mathbf{f}}(\mathbf{q})$ can be approximated with any arbitrary multi-step scheme,

$$\tilde{\mathbf{f}}(\mathbf{q}^{n+1}) \simeq \sum_{i=-1}^{N_b-2} b_{i+1} \mathbf{f}(\mathbf{q}^{n-i}) - \sum_{i=-1}^{N_a-2} \frac{a_{i+1} \mathbf{q}^{n-i}}{\Delta t}, \tag{19}$$

where coefficients a_i , N_a , b_i and N_b are selected *a priori* to impose a pre-determined physical-time accuracy-order and numerical stability.

Applying an implicit Euler scheme to march Eq. (16) in pseudo-time leads to

$$\Gamma^p \frac{\hat{\mathbf{q}}^{p+1} - \hat{\mathbf{q}}^p}{\Delta\tau} \simeq \tilde{\mathbf{f}}(\mathbf{q}^{p+1}), \tag{20}$$

where p and $p + 1$ represent the known and unknown indexes in pseudo-time, respectively, and $\Delta\tau$ is the pseudo-time step. Equation (20) can be linearized using the $O(\Delta\tau^2)$ approximation

$$\begin{aligned} \tilde{\mathbf{f}}^{p+1} &\simeq \tilde{\mathbf{f}}^p + \left. \frac{\partial \tilde{\mathbf{f}}}{\partial \tau} \right|^p \Delta\tau = \tilde{\mathbf{f}}^p + \left(\frac{\partial \tilde{\mathbf{f}}}{\partial \hat{\mathbf{q}}} \frac{\partial \hat{\mathbf{q}}}{\partial \tau} \right)^p \Delta\tau \\ &\simeq \tilde{\mathbf{f}}^p + \left. \frac{\partial \tilde{\mathbf{f}}}{\partial \hat{\mathbf{q}}} \right|^p \Delta\hat{\mathbf{q}}, \end{aligned} \tag{21}$$

without incurring in additional pseudo-time accuracy loss, to yield

$$\left\{ \frac{\Gamma}{\Delta\tau} - \frac{\partial \tilde{\mathbf{f}}}{\partial \hat{\mathbf{q}}} \right\}^p \Delta\hat{\mathbf{q}} \simeq \tilde{\mathbf{f}}(\mathbf{q}^p), \tag{22}$$

where $\partial \tilde{\mathbf{f}}/\partial \hat{\mathbf{q}}$ includes the transient and inviscid Jacobians with respect to the primitive variable vector $\hat{\mathbf{q}}$ and $\hat{\mathbf{q}}^{p+1} = \hat{\mathbf{q}}^p + \Delta\hat{\mathbf{q}}$. Combining Eqs. (19) and (22) yields

$$\begin{aligned} &\left\{ \frac{\Gamma}{\Delta\tau} + a_0 \frac{\mathbf{T}}{\Delta t} - b_0 \frac{\partial \mathbf{f}}{\partial \hat{\mathbf{q}}} \right\}^p \Delta\hat{\mathbf{q}} \\ &\simeq b_0 \mathbf{f}^p \\ &+ \sum_{i=0}^{N_b-2} b_{i+1} \mathbf{f}^{n-i} - \left(\frac{a_0 \mathbf{q}^p}{\Delta t} + \sum_{i=0}^{N_a-2} \frac{a_{i+1} \mathbf{q}^{n-i}}{\Delta t} \right), \end{aligned} \tag{23}$$

where $\partial \mathbf{f}/\partial \hat{\mathbf{q}}$ contains the inviscid Jacobians.

There is a difficulty that, at this point, it is something worth discussing [68]. Eq. (19) yields \mathbf{q}^{n+1} but Eq. (20) requires \mathbf{q}^{p+1} . Since they are not related, $\mathbf{q}^{n+1} \simeq \mathbf{q}^{p+1}$ is required as an additional approximation during the derivation of Eq. (23). If an explicit scheme for pseudo-time marching was employed instead, the approximation would be $\mathbf{q}^{n+1} \simeq \mathbf{q}^p$. Both approximations are only rigorously true at pseudo-time steady-state, since $\tilde{\mathbf{f}}(\mathbf{q}) \simeq 0$ and $\Delta\hat{\mathbf{q}} \simeq 0$ once this limit is reached. Nevertheless, they must be employed when using DTS with multi-step schemes in physical-time to solve Eq. (16). Their true impact on accuracy and efficiency of preconditioned density-based methods is not yet known, but can be determined from the use of DTS with multi-stage schemes in physical-time.

3.2 Physical-time integration with multi-stage schemes

In order to enable low Mach preconditioned density-based methods to use multi-stage schemes in physical-time, one must first treat the preconditioned pseudo-time derivative in Eq. (15) as a source term, leading to

$$\frac{\partial \mathbf{q}}{\partial t} = \bar{\mathbf{f}}(\mathbf{q}) = \mathbf{f}(\mathbf{q}) - \Gamma \frac{\partial \hat{\mathbf{q}}}{\partial \tau}, \tag{24}$$

where $\bar{\mathbf{f}}(\mathbf{q})$ is the pseudo-unsteady residue. Then, the above equation is marched in physical-time by a general IRK method using

$$\mathbf{q}^{n+1} = \mathbf{q}^n + \Delta t \sum_{i=1}^k \gamma_i \bar{\mathbf{f}}(\mathbf{k}_i), \tag{25}$$

where the k intermediate stage variable vectors \mathbf{k}_i are obtained from

$$\mathbf{k}_i = \mathbf{q}^n + \Delta t \sum_{j=1}^k \beta_{i,j} \bar{\mathbf{f}}(\mathbf{k}_j) \quad \text{for } 1 \leq i \leq k \tag{26}$$

defined at specific times $t_n + \alpha_i \Delta t$ between t_n and t_{n+1} , where γ , α and $\beta_{i,j}$ are determined *a priori* by accuracy and stability requirements. Explicit RK (ERK) methods are obtained with $\beta_{i,j} = 0$ if $j \geq i$, diagonally implicit RK (DIRK) methods with $\beta_{i,j} = 0$ if $j > i$, singly DIRK (SDIRK) methods with $\beta_{i,j} = 0$ if $j > i$ and all $\beta_{i,i}$ equal to the same constant, and fully implicit RK (FIRK) methods with all coefficients being generally nonzero. Steady-state residue and preconditioned pseudo-time derivative inside the pseudo-unsteady residue are separated in Eq. (26) to yield

$$\sum_{j=1}^k \beta_{i,j} \Gamma_j \frac{\partial \hat{\mathbf{k}}_j}{\partial \tau} = \frac{\mathbf{q}^n - \mathbf{k}_i}{\Delta t} + \sum_{j=1}^k \beta_{i,j} \mathbf{f}(\mathbf{k}_j) \tag{27}$$

for $1 \leq i \leq k$. Furthermore, $\hat{\mathbf{k}}_i$ are the intermediate variable vectors in primitive form. They are related to their conservative counterparts \mathbf{k}_i in the same way $\hat{\mathbf{q}}$ is related to \mathbf{q} . Nevertheless, this equation is written in such a way that its numerical solution is obtained very inefficiently. This is caused by the existence of multiple pseudo-time derivatives in each intermediate stage. One may note, however, that the summations on both sides have the same coefficients. Hence, this equation can be written in the diagonal form

$$\Gamma_i \frac{\partial \hat{\mathbf{k}}_i}{\partial \tau} = \tilde{\mathbf{f}}(\mathbf{k}_i) = \mathbf{f}(\mathbf{k}_i) + \sum_{j=1}^k \beta_{i,j}^{-1} \left(\frac{\mathbf{q}^n - \mathbf{k}_j}{\Delta t} \right), \tag{28}$$

where $\beta_{i,j}^{-1}$ represents the coefficients of the inverse of the coefficient matrix determined *a priori* [4]. Once pseudo-time steady-state is reached for all intermediate stage equations within each physical-time step, Eq. (28) converges to its original non-preconditioned version $\tilde{\mathbf{f}}(\mathbf{k}_i) \simeq 0$. Equation (25) can be then utilized for the physical-time update of Eq. (1) with $\tilde{\mathbf{f}}(\mathbf{k}_i)$ replaced by $\mathbf{f}(\mathbf{k}_i)$, since they become approximately equal when pseudo-time steady-state is reached.

It is important to note that this new explicit coupling in DIRK schemes retains the same triangular structure of the implicit coupling from the original versions in Eq. (26). In fact, all preconditioned implicit RK (PIRK) methods maintain the original nature of their respective implicit couplings, although in an explicit manner. Hence, it becomes quite difficult to generate a preconditioned ERK (PERK) method for physical-time marching. The reason is that the re-definition of $\mathbf{f}(\mathbf{q})$ as $\tilde{\mathbf{f}}(\mathbf{q})$ in Eq. (24) no longer introduces the pseudo-time derivative of \mathbf{k}_i at the intermediate stage i , i.e. $\beta_{i,i} = 0$ in Eq. (27), preventing \mathbf{k}_i from being marched forward in pseudo-time in the respective version of Eq. (28).

In order to maintain consistency among all preconditioned density-based methods utilized in this paper and ensure a proper comparative analysis, Eq. (28) is marched in pseudo-time with the same implicit Euler method applied to multi-step variant (20), yielding

$$\Gamma_i^p \frac{\hat{\mathbf{k}}_i^{p+1} - \hat{\mathbf{k}}_i^p}{\Delta\tau} \simeq \tilde{\mathbf{f}}(\mathbf{k}_i^{p+1}) \quad \text{for } 1 \leq i \leq k, \quad (29)$$

which, upon linearization with expansion (21), becomes

$$\left\{ \frac{\Gamma_i}{\Delta\tau} - \frac{\partial \tilde{\mathbf{f}}_i}{\partial \hat{\mathbf{k}}_i} \right\}^p \Delta \hat{\mathbf{k}}_i \simeq \tilde{\mathbf{f}}(\mathbf{k}_i^p) \quad \text{for } 1 \leq i \leq k, \quad (30)$$

where $\hat{\mathbf{k}}_i^{p+1} = \hat{\mathbf{k}}_i^p + \Delta \hat{\mathbf{k}}_i$ and $\partial \tilde{\mathbf{f}}_i / \partial \hat{\mathbf{k}}_i$ is a general Jacobian equivalent to $\partial \tilde{\mathbf{f}} / \partial \hat{\mathbf{q}}$. Combining Eqs. (28) and (30) yields

$$\begin{aligned} & \left\{ \frac{\Gamma_i}{\Delta\tau} - \frac{\partial \tilde{\mathbf{f}}_i}{\partial \hat{\mathbf{k}}_i} \right\}^p \Delta \hat{\mathbf{k}}_i + \sum_{j=1}^k \beta_{i,j}^{-1} \frac{\mathbf{T}_j^p}{\Delta t} \Delta \hat{\mathbf{k}}_j \simeq \mathbf{f}(\mathbf{k}_i^p) \\ & + \sum_{j=1}^k \beta_{i,j}^{-1} \left(\frac{\mathbf{q}^n - \mathbf{k}_j^p}{\Delta t} \right), \end{aligned} \quad (31)$$

for $1 \leq i \leq k$, where $\mathbf{T}_j = \partial \mathbf{k}_j / \partial \hat{\mathbf{k}}_j$ is now the conservative to primitive intermediate stage variable vector Jacobian and $\partial \tilde{\mathbf{f}}_i / \partial \hat{\mathbf{k}}_i$ contains the remaining Jacobians, just as $\partial \tilde{\mathbf{f}} / \partial \hat{\mathbf{q}}$ does. This development consider the general case of a fully implicit RK scheme. At any given stage in diagonally implicit RK schemes, the equation above can be re-written as

$$\begin{aligned} & \left\{ \frac{\Gamma_i}{\Delta\tau} + \beta_{i,i}^{-1} \frac{\mathbf{T}_i}{\Delta t} - \frac{\partial \tilde{\mathbf{f}}_i}{\partial \hat{\mathbf{k}}_i} \right\}^p \Delta \hat{\mathbf{k}}_i \simeq \mathbf{f}(\mathbf{k}_i^p) \\ & + \beta_{i,i}^{-1} \frac{\mathbf{q}^n - \mathbf{k}_i^p}{\Delta t} + \sum_{j=1}^{i-1} \beta_{i,j}^{-1} \left(\frac{\mathbf{q}^n - \mathbf{k}_j^{p+1}}{\Delta t} \right), \end{aligned} \quad (32)$$

for $1 \leq i \leq k$. SDIRK schemes of second and third-order accurate derived by [1] and fourth-order accurate derived by [13] that are either A-stable (AS) or strongly S-stable (SSS) were chosen for the present study. The latter property is equivalent to L-stability (LS), but when applied to a linear but nonhomogeneous standard test problem. FIRK schemes were also tested but are not shown because they are not competitive with or without preconditioning. Specific details about the form taken by Eq. (32) for each IRK scheme and its respective preconditioned version are given elsewhere [4].

The main advantage of using either Eq. (31) or (32), obtained from this novel PIRK methodology presented here, is their consistency with Eq. (15). It follows the traditional DTS approach employed by preconditioned density-based methods to simulate unsteady low Mach number flows. Hence, all the low Mach preconditioning techniques employed during the past decades can now be directly used with multi-stage schemes as well without any modifications whatsoever to either the preconditioning matrix or the preconditioned spatial discretization.

It should be mentioned that no additional approximation correlating intermediate stage vectors in physical (\mathbf{k}_i^{n+1}) and pseudo-times (\mathbf{k}_i^{p+1} or \mathbf{k}_i^p) is necessary, which is not the case for preconditioned density-based methods with multi-step schemes in physical-time, as discussed at the end of the previous subsection. This is simply due to the absence of a physical-time derivative in the original (non-preconditioned) version of intermediate stage Eq. (28). The importance of this fact will become clear when comparing multi-step and multi-stage schemes later on in the results section.

4 Spatial resolution

In every test case reported next, numerical resolution is dominated by temporal instead of spatial errors. This is assured by increasing the spatial grid until it becomes sufficiently large, combined with a high-order spatial discretization, using the fifth-order version [41, 59] of the well-known flux-difference method with a preconditioned artificial dissipation matrix [10]. Hence, one-dimensional test cases render Eq. (32) block tri-diagonal whereas two-dimensional ones make it block penta-diagonal. For this reason, a diagonally dominant symmetric relaxation technique is applied to the latter case with line sweeps in both directions at each

pseudo-time step [10], where two block tri-diagonal matrices are inverted per step.

Finally, all selected model problems are periodic in the stream wise direction in order to eliminate a source of errors, originated by the boundary closure scheme. Periodic boundary conditions are exactly enforced in the explicit residue of Eqs. (23) and (32), while approximated in their implicit operator to avoid a cyclic matrix structure. Cross stream boundary conditions for the two-dimensional test case use a zero derivative approximation for all variables but pressure, which is fixed. No buffer zones are needed in these boundaries because initial conditions are specially constructed to satisfy the steady governing equations, essentially eliminating initial condition noise. However, grid stretching was employed in this direction away from the physically relevant domain region to damp any perturbations that reach the cross stream boundaries. Furthermore, the modified artificial dissipation was altered near these boundaries to reduce the scheme to fourth-order, enhancing numerical stability of the overall method.

5 Results and discussion

In the following sections, all density-based methods using both multi-stage and multi-step schemes in physical-time will have the capital letter P added to their acronym whenever low Mach preconditioning is used. Among multi-step schemes, these acronyms are BDF and CN for backwards differentiation formula and Crank-Nicholson, respectively. The acronym for all multi-stage schemes is SDIRK, which stands for singly diagonally implicit Runge-Kutta, since they are all triangular schemes with the same coefficient in the diagonal. There is only one exception, which is IRK (implicit Runge-Kutta), because it refers to a single stage scheme. These acronyms are followed a number representing the accuracy-order of the scheme. A dash is introduced at this point, followed by another acronym that represents the linear stability of the scheme. These are AS, LS, and SSS standing for A Stable, L Stable and Strongly S Stable, respectively.

Unsteady simulations with third-order multi-step schemes Adams-Moulton (AM3) and backwards differentiation formula (BDF3) in physical-time were also tested, but without success. The former did not pass the first periods for any Mach number. Although more stable, the latter could only finish a simulation in each problem when the number of grid points per period was too large for any practical purposes. As mentioned in the introduction, this is due to the fact that all multi-step schemes with accuracy-orders higher than two are no longer unconditionally stable. Hence, their results will not be shown here.

5.1 Convection of one-dimensional entropy perturbations

This test case has a very specific purpose. The idea here is to show that the present approach to introducing low Mach preconditioning into multi-stage schemes does not change the accuracy of their results. In other words, their diffusive/dispersive error characteristics as well as accuracy-order remain the same. On the other hand, low Mach preconditioning significantly reduces the number of iterations per physical-time step as the Mach number decreases, i.e. it significantly improves efficiency.

5.1.1 Initial conditions

Entropy waves were simulated solving the one-dimensional version of Eq. (1) at Mach numbers ranging from $M = 10^{-1}$ to 10^{-5} . They are generated with density perturbations at constant pressure and velocity. Compressible flows can sustain three distinct perturbation fields according to Perturbation Theory: entropy, vorticity and sound waves. Whenever fluctuations are sufficiently small they don't interact. As a consequence, small density perturbations are convected downstream without any modifications in amplitude, frequency or phase. Arbitrating a dimensionless function $f(x)$ as initial condition for density, its transient behavior is described by $\rho(x, t)/\rho_0 = f(x - u_0 t)$. The wave form

$$f(x) = 1 + \delta_0 \sin[2\pi x/l_0], \quad \text{with } 0 \leq x \leq l_0, \quad (33)$$

is chosen here in order to keep mean values unaltered. The terms ρ_0 and u_0 are the average density and constant flow speed, and $\delta_0 = 1\%$ of ρ_0 , l_0 and $t_0 = l_0/u_0$ are the perturbation amplitude, wavelength and period, respectively. Reference pressure and temperature are given by $P_0 = 101325 \text{ Pa}$ and $T_0 = 300 \text{ K}$, respectively.

5.1.2 Analysis framework

As physical-time advances density values are monitored in location $x/l_0 = 0$. Time-steps are defined as $\Delta t = t_0/N_T$, where N_T is the number of points per period of oscillation and one hundred periods were simulated in each run with $N_T = 4, 8, 16, 32, 64$ and 128 . In order to evaluate dissipative (or anti-dissipative) and dispersive (either lagging or leading) errors, peak to peak changes of the expected maximum or minimum values and deviations from the expected periodic location of each one of these peaks after each period were verified.

In order to demonstrate the capabilities of this novel PIRK methodology for density-based methods, two separate comparative analyzes are made. First, all PIRK schemes selected

are compared to their respective (non-preconditioned) IRK schemes in order to demonstrate they can simulate very low Mach number flows more efficiently and accurately. This can be achieved by changing the preconditioning matrix Γ back to the conservative to primitive variable Jacobian \mathbf{T} in either Eqs. (31) or (32). Nevertheless, τ can still be referred to as pseudo-time since the explicit and implicit sides of both equations utilize approximations with different spatial accuracy-orders [55]. Such a comparison is actually very conservative, since standard compressible flow codes solve for conservative variables instead. As was discussed earlier, solving for pressure directly as opposed to density improves both convergence and accuracy. Nevertheless, such a path still shows the advantages of PIRK methods while avoiding excessive coding at the same time. The second comparative analysis is between preconditioned density-based methods using multi-step and multi-stage schemes in physical-time, focusing on efficiency and accuracy as well.

Convergence criterion is based on the velocity field's L_∞ -norm and set to $M \times 10^{-7}$. Inner iterations were performed until maxima pseudo-time increment (either $\Delta \hat{\mathbf{q}}$ or $\Delta \hat{\mathbf{k}}_i$) and residue (either $\hat{\mathbf{f}}(\mathbf{q})$ or $\hat{\mathbf{f}}(\mathbf{k}_i)$) based on the velocity field's L_∞ -norm were below that limit, which is close to machine precision. Average values change very little over several periods, although they may have a considerable standard deviation within a given period. When large dissipative errors are present, they force convergence to an artificial physical-time steady-state and reduce the number of iterations per period in the process. Therefore, it becomes useful to establish the control parameter \bar{m} , which is the iteration count averaged over all simulated periods and utilized as a global measure of convergence. When multiple stages i are used, $\bar{m} = \sum \bar{m}_i$. A procedure similar to Brent's method was implemented to determine respective optimal CFL numbers and it was stopped when the average quadratic difference between bounds was below 5×10^{-3} [54]. Lower tolerance values were not feasible as they created convergence problems caused by the oscillatory nature of the iteration count and were not considered.

5.1.3 Validation of multi-step and multi-stage methods

The final periods of density values with $M = 10^{-4}$ at $x/l_0 = 0$ for all second-order methods are shown in Fig. 1. For each curve, the solid points represent the position of the maximum density value after 95 periods of oscillation. Their asymptotic location as N_T increases is $t/t_0 = 95 - 1/4$ because the first peak occurs at $t/t_0 = 3/4$. As presented, PBDF2-LS with $N_T = 16$ generate an artificial steady-state solution close to $t/t_0 = 100$. As a result of dissipative and dispersive errors, there are significant vertically and horizontally distance between points. By doing a similar analysis, PCN2-AS, PIRK2-AS and IRK2-AS shows to be

non-dissipative as predicted by numerical linear stability, and have much smaller dispersive errors than the previous one. Dispersive errors in all cases are of a lagging type because it takes longer for each point to reach its expected location when N_T decreases. All these results are Mach number independent and converge towards the analytical solution. Higher-order multi-stage schemes provide essentially similar results with expected dissipative and/or dispersive numerical behavior and, hence, do not need to be shown here as well.

5.1.4 Verification of second-order methods

Code verification can now be performed, first we compare standard multi-step schemes PBDF2-LS and PCN2-AS with single-stage PIRK-AS and IRK-AS schemes, all second-order. The reference of the numerical stability in the methods names may be omitted in the text but not in the figures. To compute a single pseudo-time iteration, the computational cost for a multi-step and a single-stage scheme are nearly the same, therefore, the average number of iterations per period can be taken as a measure of the overall costs.

For all four methods, the dimensionless density absolute error presented in Fig. 2 are computed using a reference numerical solution with twice as many points per period. To confirm second-order accuracy, a slope is also presented. The reason why PBDF2-LS's absolute error decreases dramatically at larger physical-time steps is given by Fig. 1, which shows that this method leads to an artificial steady-state in this limit. As expected, PCN2, PIRK2 and IRK2 generate machine precision equal solutions, which also shows that low Mach preconditioning has no effect on temporal accuracy. They also maintain accuracy-order marching long periods of time, improving solutions accuracy, an issue not addressed for short term simulations [8]. Results generated in a dimensionless error form for Mach numbers decreasing up to $M = 10^{-5}$ are graphically identical and hence not shown here.

In order to compare all second-order methods at different Mach numbers, Fig. 3 shows the computational cost based on the average number of pseudo-time iterations per physical-time step \bar{m} . The preconditioning matrix chosen for all schemes here presented was developed applying PBDF2-LS, therefore, fewer iterations are needed and it decreases for larger physical-time step sizes due to dissipation as shown in Fig. 1, additionally is the closest one to Mach number independence. For PCN2-AS and PIRK2-AS, even though generate essentially identical solutions over time, the convergence rates are very different. When $M = 10^{-1}$, PIRK2-AS requires from 1.09 to 1.53 times as many pseudo-time iterations per physical-time step as PCN2-AS, while when $M = 10^{-2}$ these numbers go from 1.35 to 1.90. This discrepancy in convergence

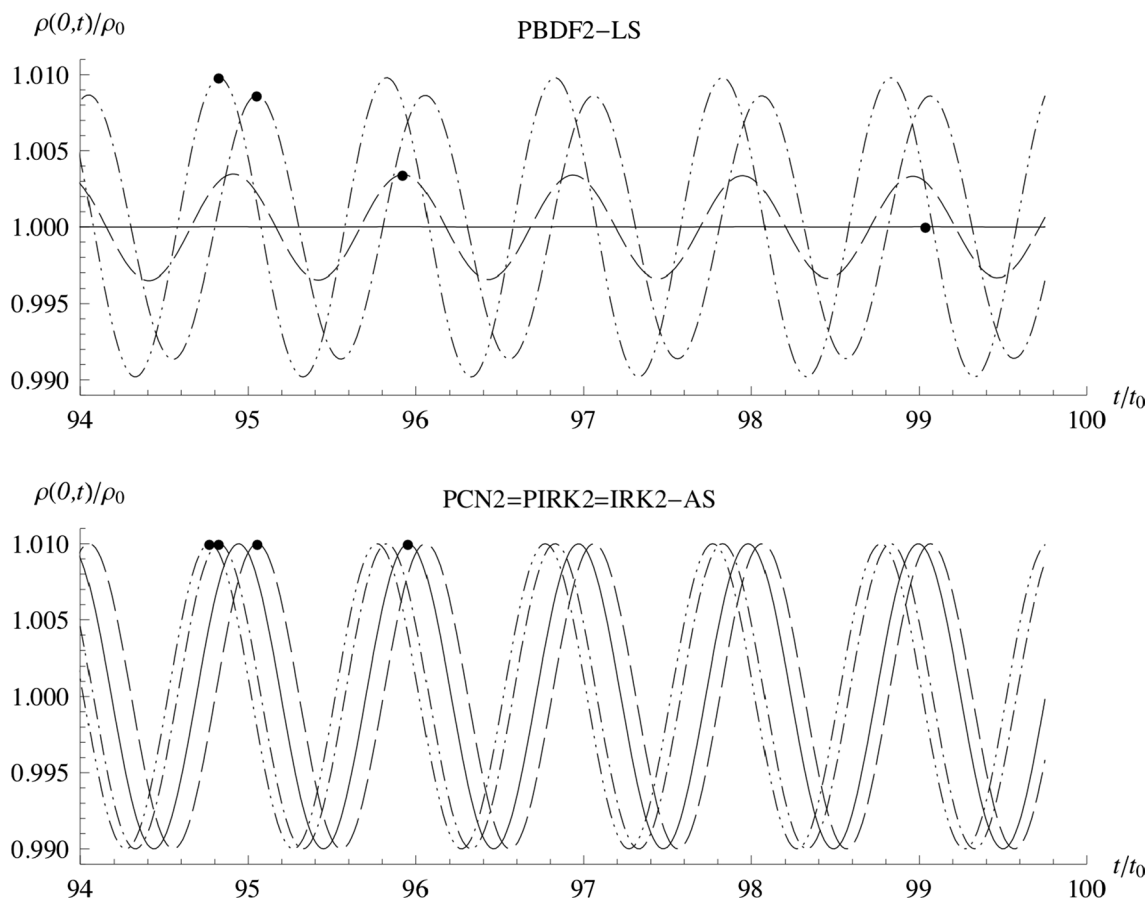


Fig. 1 Dimensionless density measured at $x = 0$ versus physical-time from PBDF2-LS and PCN2 = PIRK2 = IRK2-AS with $N_x = 102$ and $M = 10^{-4}$. Curves generated with $N_T = 16$ (—), 32 (---),

64 (-----) and 128 (.....). Solid points are the 95th maximum peak of each curve

rates continue to increase for lower Mach numbers, Fig. 3 does not show the results of PIRK2-AS at $M = 10^{-5}$ since $N_T \geq 32$ was the minimum number of points per period to generate results. PCN2-AS’s better convergence rates may be explained in the context of numerical stiff accuracy, which the pseudo-time acts on \mathbf{q}^{n+1} instead of the halfway \mathbf{k}_1 in PIRK2-AS. As discussed at the end of subsections 3.1 and 3.2, the development of low Mach preconditioned multi-stage schemes enabled the understanding of the role played by the additional approximation required to write Eq. (23) but not Eq. (32). A comparison between PIRK2-AS and PCN2-AS schemes shows that this approximation has no effect on either order or accuracy, but improves convergence by enhancing numerical stability.

As a result of no preconditioning on multi-stage schemes, even at $M = 10^{-1}$ in unsteady problems, \bar{m} values for IRK2-AS are approximately 1 to 4.81 times higher than its PIRK2-AS counterparts. As the Mach number decreases, the computational cost increases dramatically, for $M = 10^{-2}$ from 1.55 to 32.1 times more iterations are

needed. IRK2-AS simulations with even lower Mach numbers are outside the range shown in Fig. 3. Without low Mach preconditioning, poorly resolved acoustic waves deteriorate convergence dramatically.

5.1.5 Verification of higher-order methods

Figure 4 shows the absolute error per physical-time step, both show density error in dimensionless form for all schemes tested for $M = 10^{-5}$ at $t/t_0 = 1$ and 25. With one period of oscillation, the designed accuracy-order of all schemes are maintained. However, accuracy-order loss is observed after twenty-five periods of oscillation. At this later time, PIRK schemes with stronger numerical stability can maintain their order and higher accuracy but the same number of stages. The same reasoning does not apply to BDF2-LS and CN2-AS schemes, but that is probably due to the latter’s non-dissipative nature. Finally, all multi-stage schemes with and without preconditioning yield the same results, as observed in the previous subsection, confirming

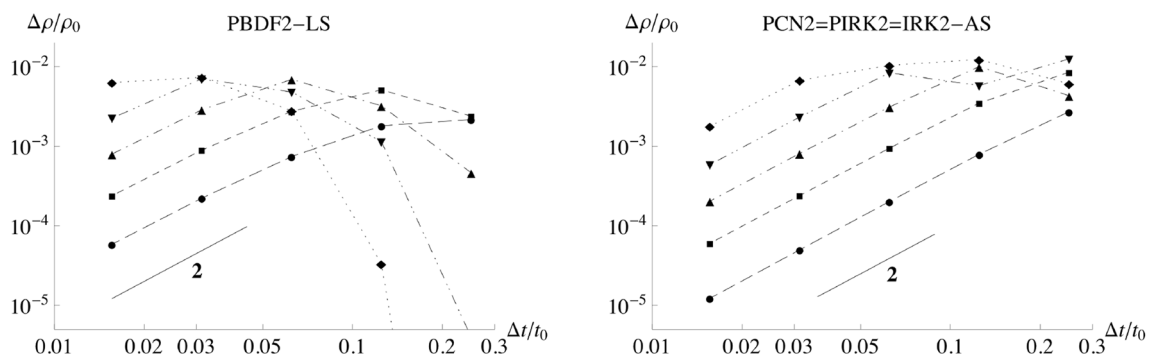


Fig. 2 Dimensionless density's absolute error per physical-time step from PBDF2-LS (left) and PCN2 = PIRK2 = IRK2-AS (right) with $N_x = 102$ and $M = 10^{-1}$. Curves generated at $t/t_0 = 1$ (—◆—), 3

(--■--), 9 (---▲---), 25 (---▼---) and 75 (---◆---). Solid line is $O(\Delta t^2)$ slope

that preconditioning does not affect accuracy. For this reason, figures such as Fig. 1 will not be shown any longer, since dissipative and dispersive errors of all IRK schemes selected are well known already [12].

Figure 5 is the equivalent of Fig. 3, but for three-stage schemes PSDIRK3-SSS, SDIRK3-SSS, PSDIRK4-AS and SDIRK4-AS. The former second-order Runge–Kutta schemes are equivalent to PCN2 in terms of pseudo-time iterations per physical-time step and more accurate than multi-step counterpart, which make a competitive scheme if a non-dissipative one is necessary. On the other hand, approximately twice as many iterations than both are required by PSDIRK4-AS to obtain more accurate results only for short integration times. The impact of preconditioning is analyzed once again for high-order schemes. For $M = 10^{-1}$ SDIRK3-SSS is 1 to 5.32 times slower than PSDIRK3-SSS, which increase to 1.61 and 51.1 when $M = 10^{-2}$. The results are similar for fourth-order accurate schemes, SDIRK4-AS is 1.04 to 5.72 times slower than PSDIRK4-AS when $M = 10^{-1}$, and 2.50 to 50.3 when $M = 10^{-2}$. Whenever there is no need to simulate acoustic waves in flow, if preconditioning techniques are used, physical-time steps based on convective wave speeds can be employed. Two, four and five-stage schemes yield qualitatively similar results and, hence, will not be discussed any further here.

5.1.6 Efficiency analysis

In engineering applications of compressible flows where small error tolerances are required, a more efficient choice is high-order versions of multi-step schemes, since there is a limitation in terms of accuracy-order for the latter one. It is important to note that efficiency was defined as the computer time required to generate a solution with a prescribed tolerance. Here, the total iteration count $\bar{m}N_T$ is used instead of

computer time because all schemes have the same computer cost per pseudo-time iteration.

Results presented in Fig. 6 show the same conclusion is valid for low speed compressible flows when PIRK methods are employed and, considering the data from previous figures, it shows the absolute error data as a fraction of the perturbation amplitude versus the number of pseudo-time iterations per physical-time period for $M = 10^{-5}$ with $t/t_0 = 1$ and 25. For $t/t_0 = 1$, PBDF2 is the optimal choice when errors higher than 10% can be tolerated. Below this value, PSDIRK2-SSS has a better performance up to a certain point. PSDIRK4-SSS outperforms all other methods if errors below 0.1% are required. Performance-wise PSDIRK3-SSS is similar to the latter two methods in both cases. Considering the scenario where $t/t_0 = 25$, i.e., later physical-times, the differences are even more pronounced. In order to better exemplify these characteristics, a hypothetical scenario given by the imposition of a 1% tolerance for both times is shown in Table 1. Excessive dissipation can lead to an artificial steady-state solution, which is the reason of the drop-off as $\bar{m}N_T$ decreases. In low Mach number flows, preconditioned k -stage methods with higher accuracy-orders and stronger numerical stabilities despite to solve k times more equations per physical-time step outperform their multi-step counterparts. Moreover, PIRK methods are very close to physical-time step and Mach number independent convergence rates. The results shown and characteristics identified are responsible for making these high-order time integration techniques more efficient than their multi-step counterparts for low Mach number compressible flows.

5.2 Absolutely unstable planar mixing-layer

The previous test case showed that the present approach to applying low Mach preconditioning to multi-stage schemes maintains their accuracy while drastically reducing the required number of iterations per physical-time step as

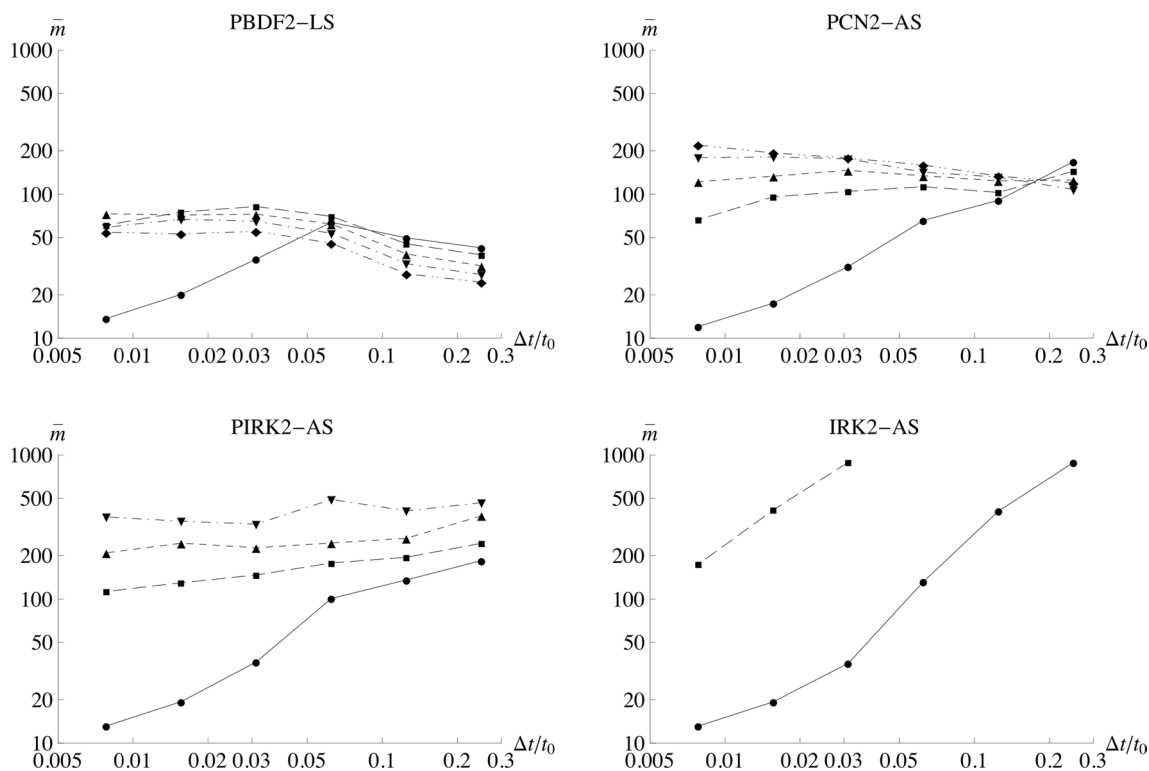


Fig. 3 Optimal average number of pseudo-time iterations per physical-time step from PBDF2-LS, PCN2-AS, PIRK2-AS and IRK2-AS with $N_x = 52$. Curves generated with $M = 10^{-1}$ (—●—), 10^{-2} (-■-), 10^{-3} (--▲--), 10^{-4} (-▼-) and 10^{-5} (---◆---)

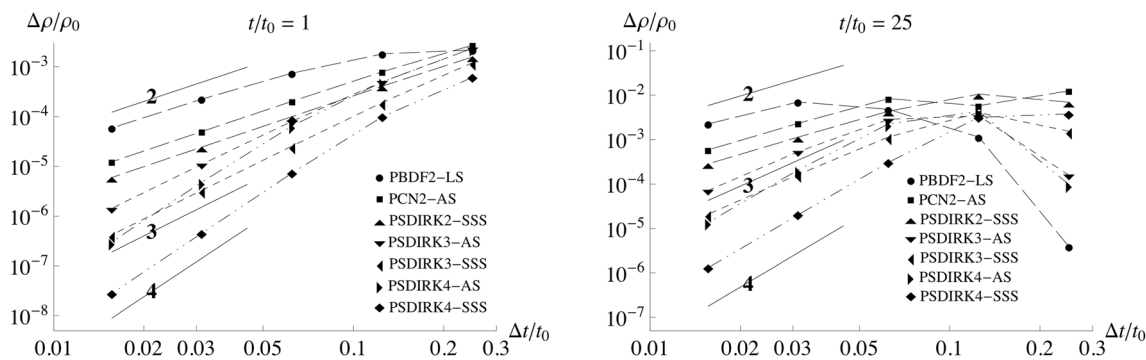


Fig. 4 Dimensionless density's absolute error per dimensionless physical-time step generated with $M = 10^{-5}$ at $t/t_0 = 1$ (left) and 25 (right) for all preconditioned density-based methods tested

the Mach number decreases. This was verified, however, in a rather simple problem. From a linear stability point of view, it is marginally stable and non-dispersive. In other words, small disturbances are simply convected downstream by the mean flow without changing their amplitude. Hence, the goal of this second test case is to show that the proposed approach also works in a much more complex flow, where disturbances grow in amplitude as they propagate until nonlinear saturation occurs. Once again, the non-preconditioned versions yield the same results but

with a much higher number of pseudo-time iterations and, hence, are not shown.

5.2.1 Initial conditions

The two-dimensional linear and nonlinear temporal evolution of inviscid perturbations in a planar mixing-layer with stream wise periodicity are simulated. They are superposed on an absolutely unstable base flow of air modeled as an ideal gas. It is approximated by the traditional

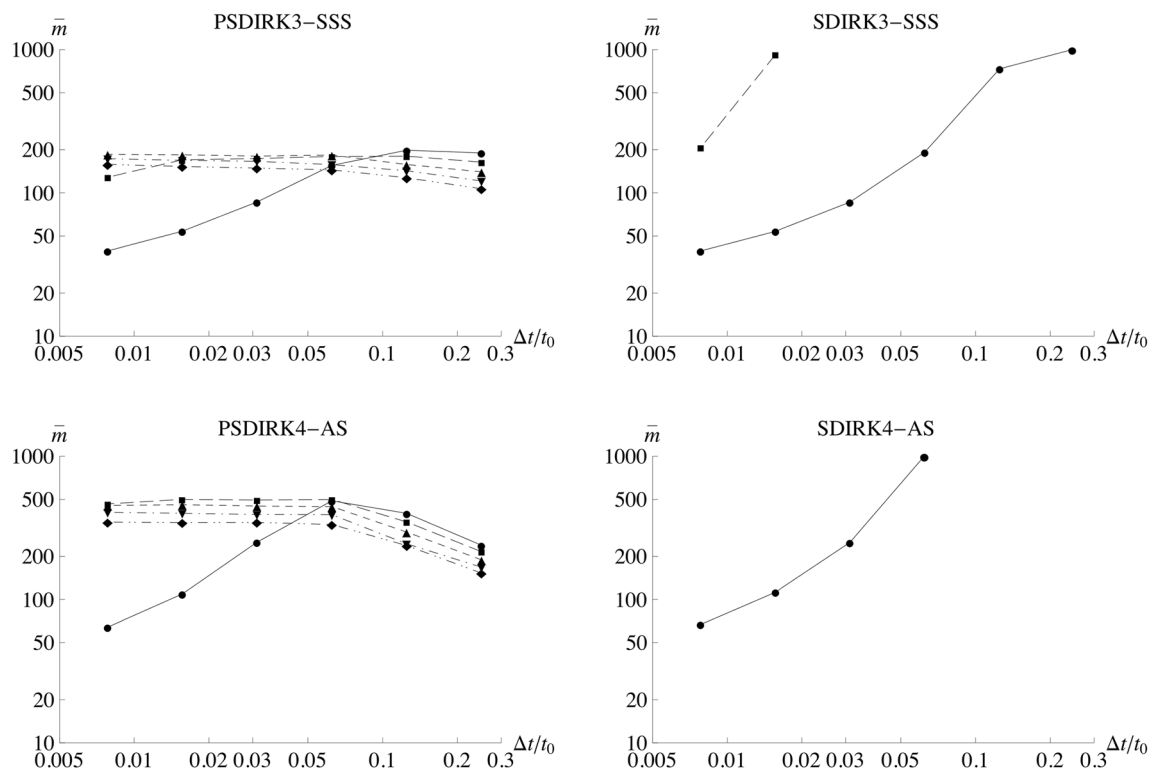


Fig. 5 Optimal average number of pseudo-time iterations per physical-time step from PSDIRK3-SSS, SDIRK3-SSS, PSDIRK4-AS and SDIRK4-AS with $N_X = 52$. Curves generated with $M = 10^{-1}$ (—●—), 10^{-2} (---■---), 10^{-3} (···▲···), 10^{-4} (-·-▼-·-) and 10^{-5} (-·-◆-·-)

hyperbolic tangent function with $0.05 m$ momentum thickness, $101325 Pa$ pressure, $300 K$ temperature and 0.5 velocity ratio, where the Mach number based on the slower layer is fixed at 10^{-3} .

Although such a base flow profile is quite commonly used in the literature [23, 42, 57, 58], these studies differ in their choice of perturbation method. While some superpose perturbations obtained from linear stability analysis [57, 58], other studies approximate the perturbation being superposed as well [23]. Regardless of the chosen approach, exact solutions of the Navier-Stokes equations are not used. Hence, these profiles introduce unwanted numerical perturbations in the simulation even in the absence of additional perturbations superposed onto the base flow [42]. Such errors can create long receptivity regions before the behavior predicted by linear stability theory can be detected or even mask this behavior to a large extent, greatly jeopardizing its detection. This is one major reason why most numerical simulations of absolutely unstable flows [57, 58] and globally unstable flows [8, 35] provide perturbation oscillation frequencies but not their temporal growth rates for comparisons with linear stability theory results, since growth rate calculations are significantly more sensitive to numerical errors than frequency calculations.

A few steps have been taken in the present study to avoid these difficulties and present both frequency and growth rate results. First, the compressible Euler equations have been solved instead of Navier-Stokes equations in its compressible form, since the hyperbolic tangent profile satisfies the former but not the latter. Such an approach is justified by the fact that the instability mechanism governing perturbation behavior in this flow field is inviscid [31]. Second, unwanted numerical oscillations introduced by the perturbations imposed [23] are minimized using Physical-Time Damping or PTD [63]. Since the perturbation imposed will have a higher energy content than any other undesired perturbations, it is the last one to be damped. The solution obtained by interrupting PTD when the cross stream velocity is one order of magnitude higher than its steady-state value represents a highly accurate perturbed initial condition. Relative errors based on perturbation wavenumber are on the order of 0.001% . Frequency spectra of the cross stream velocity data from initial conditions with and without PTD reveals that unwanted perturbations have an amplitude up to 5 times smaller than the one with the targeted frequency.

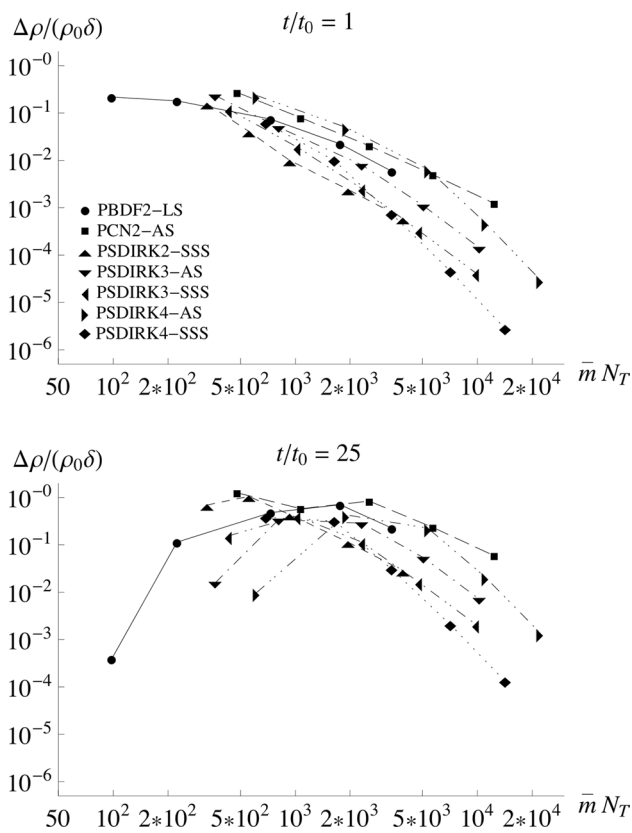


Fig. 6 Dimensionless density’s absolute error as a fraction of the perturbation amplitude versus the number of pseudo-time iterations per physical-time period generated with $M = 10^{-5}$ at $t/t_0 = 1$ (top) and 25 (bottom)

Table 1 CPU time ratio of each scheme and the respective fastest one for each time evaluated in Fig. 6. Data taken at a 1% tolerance

	$t/t_0 = 1$	$t/t_0 = 25$
PSDIRK2-SSS	1.00	1.53
PSDIRK3-SSS	1.46	1.22
PSDIRK4-SSS	1.76	1.00
PSDIRK3-AS	2.25	1.99
PBDF2-LS	2.89	4.41
PCN2-AS	4.16	6.57
PSDIRK4-AS	4.93	2.88

5.2.2 Analysis framework

Linear stability analysis (LSA) data, obtained with a temporal version of a code originally developed for a spatial analysis [2, 37], was employed to properly set up the numerical simulations. Once the perturbation wave number k_R was chosen, the domain stream wise length was set to its wave length l_0 and LSA revealed the related perturbation period t_0 and

cross stream width $2h_0$. The former guided the physical-time step selection, with the number of points per period fixed at $N_T = 4, 8, 16, 32, 64$ and 128 , whereas the latter fixed the domain width, chosen as the cross stream location where the maximum perturbation amplitude is 1000 times smaller. Furthermore, grid resolution studies were performed beforehand to guarantee this data is dominated by temporal errors instead of spatial ones. They led to a uniform grid in the periodic stream wise direction with $N_X = 102$ points. On the other hand, they required a non-uniform grid in the non-homogeneous cross stream direction with $N_Y = 301$. A uniform grid was employed in the vorticity containing region, where the cross stream velocity difference over two consecutive grid points at the center divided by the maximum cross stream velocity is equal to 0.114164 , approximately, for all tests performed. This grid was stretched towards the lateral boundaries, with the last three grid points clustered to enhance numerical stability [61, 76]. The function

$$y/(2\theta_0) = a\eta \left(1 + b \exp[-c(\eta \pm 1)^2] \right), \tag{34}$$

with $\mp\eta \geq 0$ and for $-1 \leq \eta \leq +1$, was used to generate this grid, where a, b and c are adjustable coefficients.

All numerical simulation data was fitted to the Fourier mode functional form known from linear stability analysis using a least squares based nonlinear regression procedure programmed in the *Mathematica* built-in function FindFit [72]. This allowed the extraction of perturbation wave numbers, frequencies and temporal growth rates from these simulations. Several numerical data sets were fed to this fitting procedure, with all possible initial and final simulation time combinations leading to a data set size of at least 5 periods. The one yielding the smallest standard deviation for frequency was selected, although doing so for the temporal growth rate instead led to the same result in almost all cases tested. Since the initial dimensionless cross stream velocity perturbation amplitude was approximately 10^{-7} , the perturbation behavior remained linear for a long enough time. Hence, a large enough data set was generated to maintain these standard deviations small. For instance, when an initial condition with the targeted dimensionless perturbation wave number $k_R^* = 0.4$ is used, this fitting procedure selects unsteady simulation data within $3 \lesssim t/t_0 \lesssim 8$. At both limits, $k_R^* \simeq 0.400002$ and 0.400447 , obtained with standard deviations of 8.07344×10^{-5} and 1.11019×10^{-3} , respectively. Hence, wave number relative errors vary from $5 \times 0.0005\%$ to $1.1175 \times 0.1\%$, providing strong evidence that the targeted perturbation is indeed the one being amplified during linear growth. The dimensionless frequency and temporal growth rate during this period are $\omega_R^* \simeq 0.300076$ and $\omega_I^* \simeq 0.0466667$, after minimizing their respective standard deviations to 7.62142×10^{-6} and 8.4249×10^{-6} . Their

relative errors are $2.53333 \times 0.01\%$ and $7.86504 \times 0.1\%$, respectively. Even though the temporal growth rate error is larger than its frequency counterpart, as expected, both are relatively small and indicate a good agreement with LSA data.

5.2.3 Validation of multi-step and multi-stage methods

The processes described in the previous two sections were repeated for $k_R^* = 0.1, 0.2, \dots, 0.8$ and 0.9 . Frequencies (left) and temporal growth rates (right) extracted from these simulations are compared to LSA data (solid lines) in Fig. 7. There is an excellent agreement for the frequency data whereas non-negligible but still relatively small differences can be observed for the temporal growth rates. It is important to note that such a higher error for the growth rate, compared to its respective frequency, is observed even in the LSA calculations. These errors are sensitive to the spatial location chosen for the extraction of the temporal data points, but they always stay approximately within the minimum and maximum values presented in the legend of this figure. Nevertheless, temporal growth rate relative errors do appear to systematically increase with wave number, which is consistent with the dissipative error expected of biased upwind schemes such as the one employed for the present spatial discretization. Despite these issues, the present results are well within the accuracy bounds found in the cited literature and, hence, provide sufficient validation for the novel methodology applied to the planar mixing-layer. Furthermore, all the other multi-step and multi-stage schemes tested provide similar results.

5.2.4 Verification of second-order methods

In order to present an accuracy-order verification in a concise manner, only results for the $k_R^* = 0.4$ case are discussed. As was done for the previous test case, errors are estimated using the numerical solution obtained with twice as many points per period. Figure 8 shows dimensionless frequency and temporal growth rate absolute errors, which are the real and imaginary parts of $\Delta\omega/\omega_0$, respectively, versus dimensionless physical-time step for all four second-order-methods analyzed. It is important to note that this analysis is even more stringent than the one presented for the first test case in Figs. 2 and 4, since density is one of the simulated variables whereas frequency and growth rates are extracted after post-processing the simulated cross stream velocity component variable. The numerical results presented in Fig. 8 show that all schemes do follow their theoretical second-order slope, at least approximately. This is true for both frequency and growth rate curves, although the former has a smaller error than the latter as expected. Furthermore, PCN2-AS and PIRK2-AS generate essentially identical results, which

is another expected result. However, there are differences among these schemes. Arguably, PBDF2-LS results have a higher error whereas PSDIRK2-SSS results have a lower error. Once again, stronger linear numerical stability leads to lower errors in general. This is not true when comparing PBDF2-LS with PCN2-AS, but this particular behavior is caused by the excessive dissipative error introduced by the former scheme with the latter one being non-dissipative. Such an excessive dissipative error also decreases the order of this scheme, whose slope is the furthest away from second-order among the schemes in Fig. 8. These results are consistent with the previous test case.

5.2.5 Verification of higher-order methods

A similar analysis is performed for the higher-order multi-stage schemes and their results are shown in Fig. 9. Most curves in this figure are further away from straight lines than their counterparts in Fig. 8. These oscillations are likely

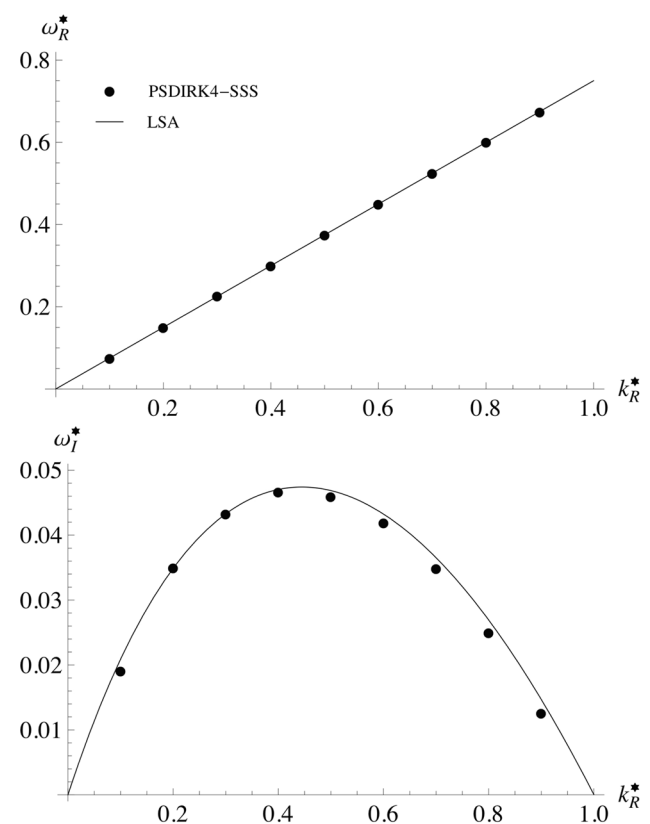


Fig. 7 Frequency (top) and temporal growth rate (bottom) as functions of wave number. Frequency relative errors in percentage values are 4.35798×10^{-1} , 7.46618×10^{-2} , 7.79251×10^{-1} , 2.52349×10^{-2} , 4.99809×10^{-2} , 9.02838×10^{-2} , 1.30255×10^{-1} , 1.26476×10^{-1} and 1.2845×10^{-1} , whereas temporal growth rate errors in percentage values are 8.62826 , 3.27743×10^{-1} , 8.19625×10^{-2} , 8.04703×10^{-1} , 1.91886 , 3.03887 , 4.45349 , 7.04191 and 14.0822 . Solid lines represent the results from linear stability analysis

caused by the increased complexity of this 2D test case, which often leads to this behavior in high-order multi-stage schemes when compared to simpler flows such as the 1D test case [24]. One major difference is the fact that error measurements are not taken directly from the simulated data, but are instead based on frequencies and temporal growth rates calculated from the simulated data. When temporal accuracy-order is increased, error measurements decrease towards their respective minimal standard deviations found in the nonlinear fitting procedure ($\sim O(10^{-5})$). This problem could be avoided by employing larger data samples, which would reduce the minimal standard deviations. However, this would require either the use of initial perturbations with even smaller amplitudes or the simulation of linearized equations instead. The former would increase much further the CPU time required for the present study whereas the latter is beyond the scope of the present study. Neither approach is necessary though, given the reasonable qualitative agreement between theoretical and numerical slopes observed in Fig. 9.

5.2.6 Efficiency analysis

Finally, as was done for the one-dimensional test case in Fig. 6, an efficiency analysis for the two-dimensional test case is shown in Fig. 10. Once again, efficiency is defined as the total iteration count $\bar{m} N_T$ required to generate a solution for $\Delta\omega/\omega_0$ with a prescribed tolerance, be it frequency or temporal growth rate, since the computer time per pseudo-time iteration is essentially the same for all multi-step and multi-stage schemes studied here. In general, Fig. 10 for this two-dimensional test case indicates that higher-order and/or stronger numerical stability lead to more efficient methods. Although the same general statement was made based on Fig. 6, some important differences exist.

When focusing on frequency data, shown in Fig. 10 (left), PCN2-AS is the most efficient choice when errors higher than 3% can be tolerated. Otherwise, PSDIRK2-SSS is the optimal choice, but only when errors higher than 0.6% can be tolerated. If smaller errors are required, PSDIRK3-SSS is the best available choice among the schemes analyzed. Similar conclusions can be reached from temporal growth rate data as well, which is shown in Fig. 10 (right). PCN2-AS is still the most efficient choice, but only when tolerances higher than 9% are imposed. When smaller errors are required,

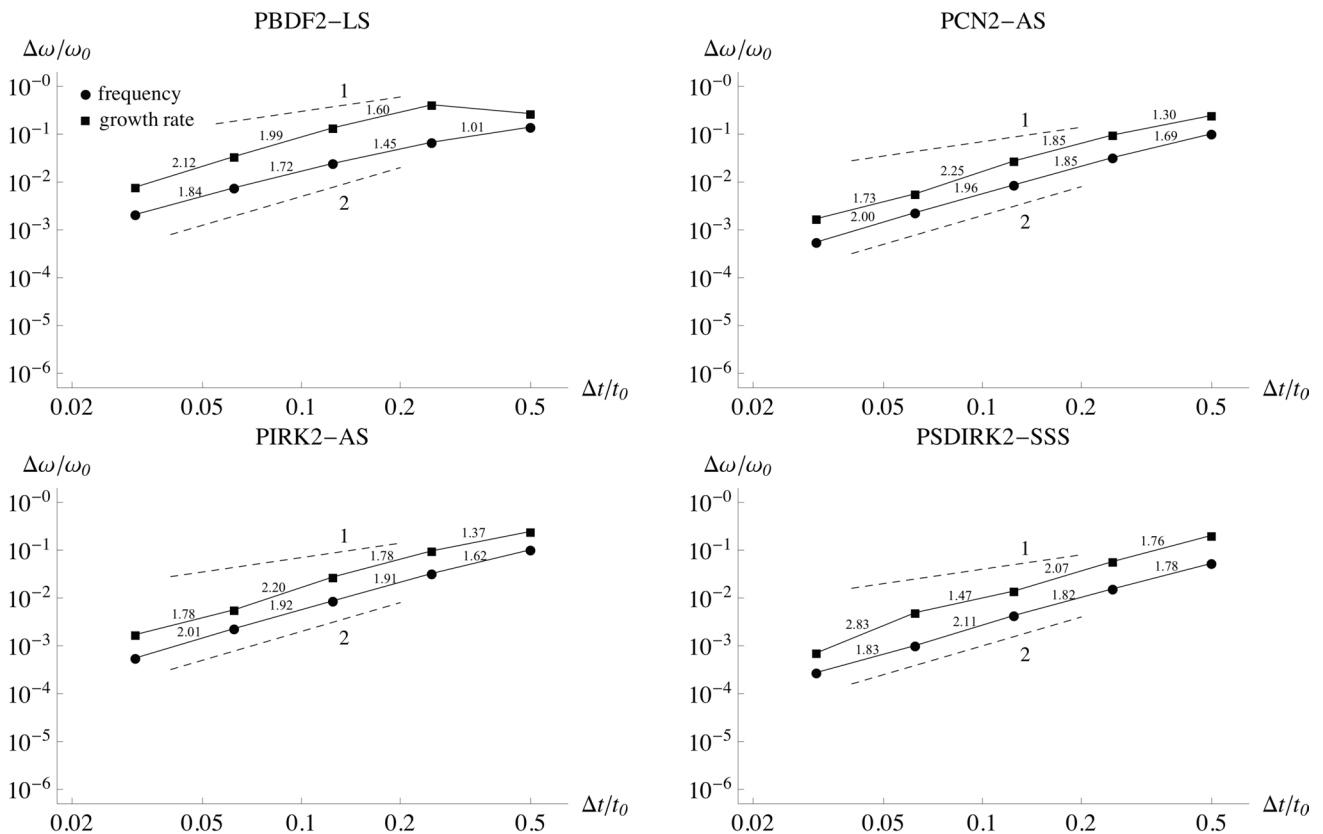


Fig. 8 Dimensionless frequency and temporal growth rate absolute errors per dimensionless physical-time step for all second-order accurate schemes in physical-time. Dashed lines indicate theoretical orders

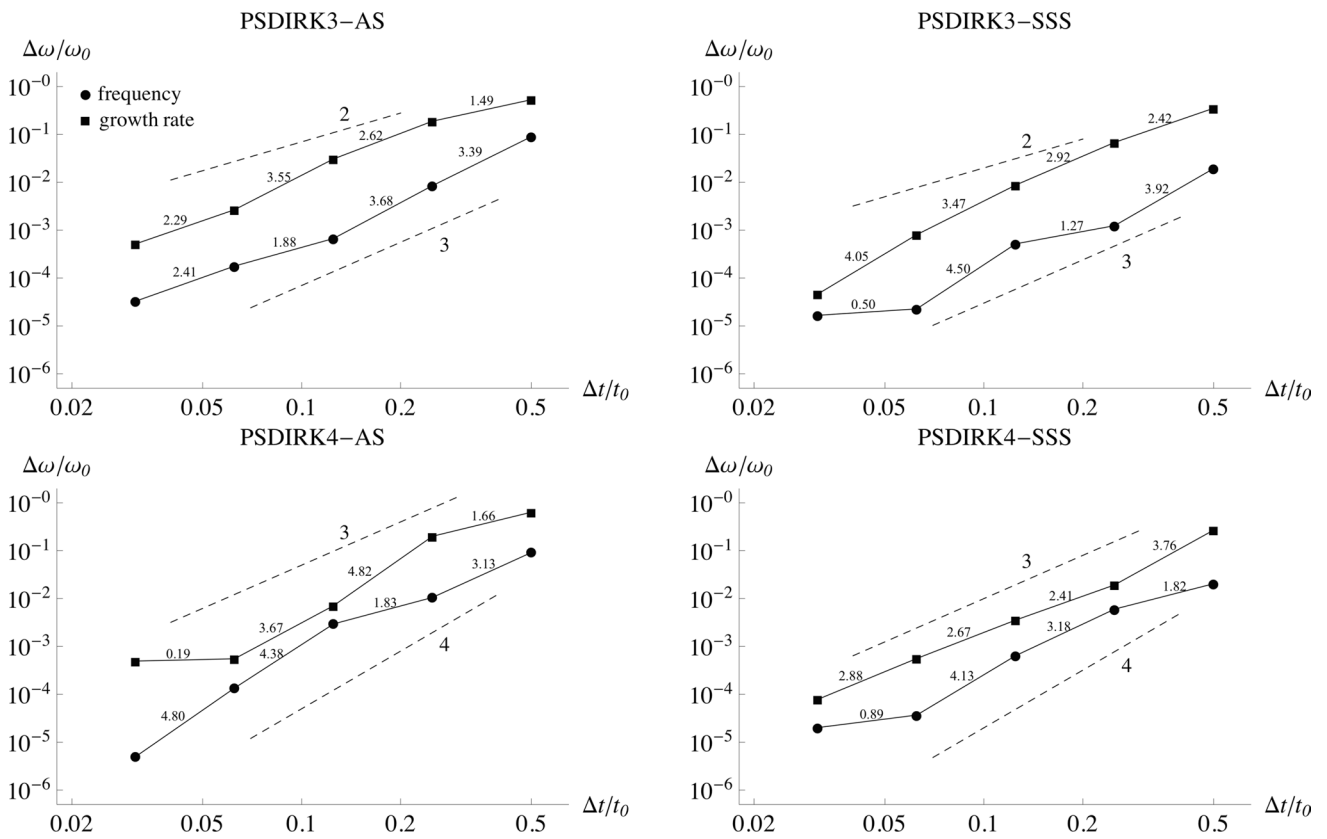


Fig. 9 Same as Fig. 8, but for third and fourth-order accurate schemes in physical-time

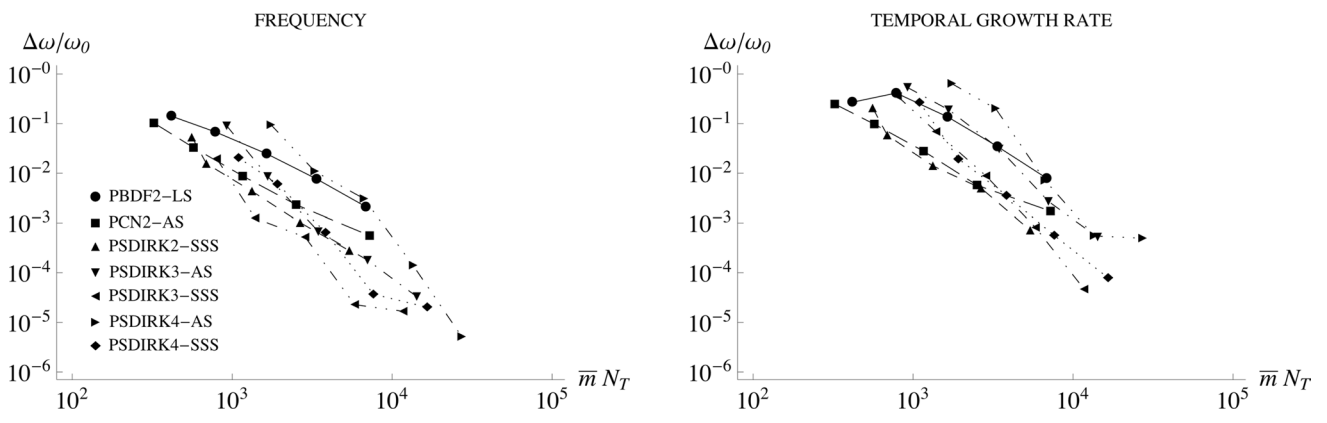


Fig. 10 Dimensionless frequency (left) and temporal growth rate (right) absolute errors versus the number of pseudo-time iterations per physical-time period for the conditions reported in Fig. 8

PSDIRK2-SSS is the preferred choice within the range analyzed. Nevertheless, PCN2-AS takes a close second place for tolerances down to 0.5%, where it is replaced by PSDIRK3-SSS when smaller tolerances are needed. A more quantitative comparison amongst all schemes analyzed here can be performed by imposing a 1% tolerance. This is shown in Table 2 when applied to both the frequency and temporal growth rate data. In contrast with the one-dimensional test

case results, PCN2-AS performs significantly better than its multi-step counterpart PBDF2-LS and PSDIRK4-SSS is not the most efficient scheme when imposing the smallest tolerances. The latter is likely due to the stronger error propagation caused by the combined effects of an additional spatial dimension and an additional intermediate stage equation. Nevertheless, multi-stage schemes with improved numerical stability and/or higher accuracy-order still outperform

Table 2 CPU time ratio of each scheme and the respective fastest one for real and complex frequencies taken from Fig. 10. Data taken at a 1% tolerance

	$\text{Re}[\Delta\omega/\omega_0]$	$\text{Im}[\Delta\omega/\omega_0]$
PSDIRK2-SSS	1.00	1.00
PSDIRK3-SSS	1.10	1.47
PSDIRK4-SSS	1.68	1.59
PSDIRK3-AS	1.65	3.18
PBDF2-LS	3.11	3.42
PCN2-AS	1.13	1.17
PSDIRK4-AS	3.64	3.45

the multi-step schemes whenever the required tolerances are small enough.

6 Conclusions

Since its creation approximately thirty years ago, low Mach number preconditioned density-based methods have employed multi-step schemes for physical-time marching. In most cases, a second-order Backwards Differencing Scheme (BDF2) is employed, although a second-order Crank-Nicholson (CN2) has been used as well. Both are unconditionally stable, but this preference for the former comes from its ability to reach zero gain in the infinite time step limit (L-stable) whereas the gain is only bounded in the same limit (A-stable) for the latter. All higher-order multi-step schemes are conditionally stable, which makes them unsuitable for problems with even a moderate level of stiffness.

In recent years, temporal integration in high Mach number compressible flow simulations has achieved higher accuracy orders and/or stronger numerical stability by using multi-stage schemes. The present paper shows that the same can be done for low Mach preconditioned density-based methods. Furthermore, this is done using the same low Mach preconditioned dual-time-stepping procedure traditionally employed with multi-step schemes, but without introducing any additional approximations. This means that all low Mach preconditioning matrices and all low Mach preconditioned artificial dissipation schemes used for spatial resolution developed over the past three decades can now be used with multi-stage physical-time integration schemes, requiring no modifications or adjustments whatsoever. In doing so, the large field of research devoted to Runge–Kutta schemes is now open to low Mach preconditioned density-based methods as well. One and two-dimensional test cases are simulated to demonstrate the capabilities of this novel approach.

Acknowledgements The first author would like to thank CNPq for the financial support through grants 480599/2007-6, 305031/2010-4 and 312255/2013-6 as well as FAPERJ through grant E-26/103.254/2011. Some of the data in this paper has been presented at the AIAA Aviation conference (2014-3085).

References

- Alexander R (1977) Diagonally implicit Runge–Kutta methods for stiff O.D.E. *SIAM J Numer Anal* 14(6):1006–1021
- Alves LSB, Kelly RE, Karagozian AR (2008) Transverse jet shear layer instabilities. part ii: linear analysis for large jet-to-crossflow velocity ratios. *J Fluid Mech* 602:383–401
- Alves LSB (2009) Review of numerical methods for the compressible flow equations at low mach numbers. In: XII Encontro de Modelagem Computacional, Rio de Janeiro, RJ, Brazil
- Alves LSB (2010) Dual time stepping with multi-stage schemes in physical time for unsteady low mach number compressible flows. In: VII Escola de Primavera de Transição e Turbulência, ABCM, Ilha Solteira, São Paulo, Brasil
- Teixeira RSD, Alves LSDB (2017) Minimal gain marching schemes: searching for unstable steady-states with unsteady solvers. *Theo Comput Fluid Dyn* 31(5–6):607–621
- Ascher UM, Ruuth SJ, Spiteri RJ (1997) Implicit-explicit Runge–Kutta methods for time-dependent partial differential equations. *Appl Numer Math* 25:151–167
- Bijl H, Wesseling P (1998) A unified method for computing incompressible and compressible flows in boundary-fitted coordinates. *J Comput Phys* 141(2):153–173
- Bijl H, Carpenter MH, Vatsa VN, Kennedy CA (2002) Implicit time integration schemes for the unsteady compressible Navier–Stokes equations: laminar flow. *J Comput Phys* 179:313–329
- Buelow PEO, Venkateswaran S, Merkle CL (1994) Effect of grid aspect ratio on convergence. *AIAA J* 32(12):2402–2408
- Buelow PEO, Venkateswaran S, Merkle CL (2001) Stability and convergence analysis of implicit upwind schemes. *Comput Fluids* 30:961–988
- Bukhvostova A, Kuerten JGM, Geurts BJ (2015) Low Mach number algorithm for droplet-laden turbulent channel flow including phase transition. *J Comput Phys* 295:420–437
- Butcher JC (2008) *Numerical methods for ordinary differential equations*. John Wiley & Sons Inc, England
- Cash JR (1979) Diagonally implicit Runge–Kutta formulae with error estimates. *IMA J Appl Math* 24(3):293–301
- Choi YH, Merkle CL (1993) The application of preconditioning in viscous flows. *J Comput Phys* 105:207–223
- Chorin AJ (1967) A numerical method for solving incompressible viscous flow problems. *J Comput Phys* 2:12–26
- Chorin AJ (1968) Numerical solution of the Navier–Stokes equations. *Math Comput* 22(104):745–762
- Dahlquist G (1963) A special stability problem for linear multistep methods. *BIT Numer Math* 3:27–43
- Darmofal DL, Schmid PJ (1996) The importance of eigenvectors for local preconditioners of the Euler equations. *J Comput Phys* 127:346–362
- Denaro FM (2003) On the application of the helmholtz–hodge decomposition in projection methods for incompressible flows with general boundary conditions. *Int J Numer Methods Fluids* 43:43–69
- Ferracina L, Spijker MN (2008) Strong stability of singly-diagonally-implicit runge–kutta methods. *Appl Numer Math* 58:1675–1686

21. Folkner D, Katz A, Sankaran V (2015) An unsteady preconditioning scheme based on convective-upwind split-pressure artificial dissipation. *Int J Numer Methods Fluids* 78(1):1–16
22. Frochte J, Heinrichs W (2009) A splitting technique of higher order for the Navier-Stokes equations. *J Comput Appl Math* 228:373–390
23. Germanos RAC, Souza LF, Medeiros MAF (2009) Numerical investigation of the three-dimensional secondary instabilities in the time-developing compressible mixing layer. *J Brazil Soc Mech Sci Eng* 31(2):125–136
24. Greene PT, Eldredge JD, Zhong X, Kim J (2016) A high-order multi-zone cut-stencil method for numerical simulations of high-speed flows over complex geometries. *J Comput Phys* 316:652–681
25. Guerra J, Gustafsson B (1986) A numerical method for incompressible and compressible flow problems with smooth solutions. *J Comput Phys* 63:377–397
26. Guermond JL, Mineev P, Shen J (2006) An overview of projection methods for incompressible flows. *Comput Methods Appl Mech Eng* 195:6011–6045
27. Harlow FH, Welch JE (1965) Numerical calculation of time-dependent viscous incompressible flow of fluids with free surface. *Phys Fluids* 8:2182–2189
28. Harlow FH, Amsden AA (1971) A numerical fluid dynamics calculation method for all flow speeds. *J Comput Phys* 8(2):197–213
29. Hejranfar K, Kamali-Moghadam R (2012) Assessment of three preconditioning schemes for solution of the two-dimensional Euler equations at low Mach number flows. *Int J Numer Methods Eng* 89:20–52
30. van der Heul DR, Vuik C, Wesseling P (2003) A conservative pressure-correction method for flow at all speeds. *Comput Fluids* 32:1113–1132
31. Ho CM, Huerre P (1984) Perturbed free shear layers. *Ann Rev Fluid Mech* 16:365–424
32. Hosangadi A, Merkle CL, Turns SR (1990) Analysis of forced combusting jets. *AIAA J* 28(8):1473–1480
33. Hou Y, Mahesh K (2005) A robust, collocated, implicit algorithm for direct numerical simulation of compressible, turbulent flows. *J Comput Phys* 205:205–221
34. Issa RI, Gosman AD, Watkins AP (1986) The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme. *J Comput Phys* 62(1):66–82
35. Jothiprasad G, Mavriplis DJ, Caughey DA (2003) Higher-order time integration schemes for the unsteady Navier-Stokes equations on unstructured meshes. *J Comput Phys* 191(2):542–566
36. Karki KC, Patankar SV (1989) Pressure-based calculation procedure for viscous flows at all speeds in arbitrary configurations. *AIAA J* 27(9):1167–1174
37. Kelly RE, Alves LSB (2008) A uniformly valid asymptotic solution for the transverse jet and its linear stability analysis, *Philosophical Transactions of the Royal Society of London. Series A Math Phys Sci* 366:2729–2744
38. Kennedy CA, Carpenter MH (2003) Additive Runge-Kutta schemes for convection-diffusion-reaction equations. *Appl Numer Math* 44:139–181
39. Keshtiban IJ, Belblidia F, Webster MF (2004) Compressible flow solvers for low Mach number flows: a review, Department of computer science at university of wales, swansea. Institute of Non-Newtonian Fluid Mechanics, u.k.
40. Klainerman S, Majda A (1982) Compressible and incompressible fluids. *Commun Pure Appl Math* 35:629–651
41. Laney CB (1998) *Computational gas dynamics*. Cambridge University Press, United Kingdom
42. Lardjane N, Fedioun I, Gokalp I (2004) Accurate initial conditions for the direct numerical simulation of temporal compressible binary shear layers with high density ratio. *Comput Fluids* 33:549–576
43. Lee SH (2005) Convergence characteristics of preconditioned Euler equations. *J Comput Phys* 208(1):266–288
44. Lee SH (2007) Cancellation problem of preconditioning method at low Mach numbers. *J Comput Phys* 225(2):1199–1210
45. Lomax H, Pulliam TH, Zingg DW (2001) *Fundamentals of computational fluid dynamics*. Scientific computation. Springer & Verlag, Berlin
46. Merkle CL, Athavale M (1987) A time accurate unsteady incompressible algorithm based on artificial compressibility. *AIAA Conf Paper* 87–1137:397–407
47. Merkle CL, Choi YH (1987) Computation of low-speed flow with heat addition. *AIAA J* 25:831–838
48. Merkle CL, Choi YH (1988) Computation of low speed compressible flows with time-marching methods. *Int J Numer Methods Eng* 25:293–311
49. Müller B (1999) *Low Mach Number Asymptotics of the Navier-Stokes Equations and Numerical Implications*, lecture series 1999-03 Edition, von Karman Institute for Fluid Dynamics
50. Panton RL (2003) *Incompressible flow*, 4th edn. Wiley & Sons, Hoboken
51. Pareschi L, Russo G (2005) Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation. *J Sci Comput* 25(1):129–155
52. Patankar SV, Spalding DB (1972) A calculation procedure for heat, mass, and momentum transfer in three-dimensional parabolic flow. *Int J Heat Mass Trans* 15:1787–1806
53. Paolucci S (1982) On the filtering of sound from the Navier-Stokes equations, Tech. Rep. Technical Report SAND 82-8257, Sandia National Laboratories
54. Press WH, Flannery BP, Teukolsky SA, Vetterling WT (1992) *Numerical recipes in FORTRAN 77: the art of scientific computing*, 2nd edn. Cambridge University Press, Cambridge
55. Rai MM (1987) Navier-stokes simulations of blade-vortex interaction using high-order accurate upwind schemes. *AIAA Conf Paper* 87–0543:1–25
56. Rossow CC (2003) A blended pressure/density based method for the computation of incompressible and compressible flows. *J Comput Phys* 185(2):375–398
57. Sandham ND, Reynolds WC (1989) Compressible mixing layer: linear theory and direct simulation. *AIAA J* 28(4):618–624
58. Sandham ND, Reynolds WC (1991) Three-dimensional simulations of large eddies in the compressible mixing layer. *J Fluid Mech* 224:133–158
59. Schwer DA (1999) Numerical study of unsteadiness in non-reacting and reacting mixing layers, Ph.D. thesis, Pennsylvania State University
60. Shuen JS, Chen KH, Choi YH (1993) A coupled implicit method for chemical non-equilibrium flows at all speeds. *J Comput Phys* 106:306–318
61. Shukla RK, Zhong X (2005) Derivation of high-order compact finite difference schemes for non-uniform grid using polynomial interpolation. *J Comput Phys* 204:404–429
62. Suh J, Frankel SH, Mongeau L, Plesniak MW (2006) Compressible large eddy simulations of wall-bounded turbulent flows using a semi-implicit numerical scheme for low Mach number aeroacoustics. *J Comput Phys* 215(2):526–551
63. Teixeira RS, Alves LSB (2012) Modeling far field entrainment in compressible flows. *Int J Comput Fluid Dyn* 26:67–78
64. Teixeira PC, Alves LSB (2015) Modeling supercritical heat transfer in compressible fluids. *Int J Therm Sci* 88:267–278
65. Turkel E (1987) Preconditioned methods for solving the incompressible and low speed compressible equations. *J Comput Phys* 72(2):277–298

66. Turkel E (1993) Review of preconditioning methods for fluid dynamics. *Appl Numer Math* 12:257–284
67. Turkel E (1999) Preconditioning techniques in computational fluid dynamics. *Ann Rev Fluid Mech* 31:385–416
68. Turkel E, Vatsa VN (2005) Local preconditioners for steady and unsteady flow applications. *ESAIM Math Model Numer Anal* 39(3):515–535
69. Venkateswaran S, Merkle CL (1999) *Analysis of Preconditioning Methods for the Euler and Navier-Stokes Equations*, Von Karman Institute Lecture Series
70. Wang L, Mavriplis DJ (2007) Implicit solution of the unsteady Euler equations for high-order accurate discontinuous Galerkin discretizations. *J Comput Phys* 225(2):1994–2015
71. Weiss JM, Smith WA (1995) Preconditioning applied to variable and constant density flows. *AIAA J* 33(11):2050–2057
72. Wolfram S (2003) *The mathematica book*, 5th edn. Cambridge University Press, New York, Wolfram Media
73. Yoh JJ, Zhong X (2004) New hybrid Runge-Kutta methods for unsteady reactive flow simulation. *AIAA J* 42(8):1593–1600
74. Yoh JJ, Zhong X (2004) New hybrid Runge-Kutta methods for unsteady reactive flow simulation: applications. *AIAA J* 42(8):1601–1611
75. Zhong X (1996) Additive semi-implicit Runge-Kutta methods for computing high-speed nonequilibrium reactive flows. *J Comput Phys* 128:19–31
76. Zhong X, Tatineni M (2003) High-order non-uniform grid schemes for numerical simulation of hypersonic boundary-layer stability and transition. *J Comput Phys* 190(2):419–458

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.