

An Efficient Semi-implicit Solver for Direct Numerical Simulation of Compressible Flows at All Speeds

Davide Modesti¹ · Sergio Pirozzoli²

Received: 9 February 2017 / Revised: 6 July 2017 / Accepted: 4 August 2017 /
Published online: 10 August 2017
© Springer Science+Business Media, LLC 2017

Abstract We develop a semi-implicit algorithm for time-accurate simulation of compressible shock-free flows, with special reference to wall-bounded flows. The method is based on partial linearization of the convective fluxes in such a way to suppress, or at least mitigate the acoustic time step limitation. Together with replacement of the total energy equation with the entropy transport equation, this approach avoids the inversion of block-banded matrices involved in classical methods, which is replaced by much less demanding inversion of standard banded matrices. The method is extended to deal with implicit integration of viscous terms and to multiple space dimensions through approximate factorization, and used as a building block of a semi-implicit Runge–Kutta scheme which guarantees third-order of accuracy in time (Nikitin in *Int J Numer Methods Fluids* 51:221–233, 2006). Numerical experiments are carried out for isotropic turbulence, plane channel flow, and flow in a square duct. All available data support higher computational efficiency than existing methods, and saving of computer resources ranging from 85% under low-subsonic flow conditions (down to $M_0 \sim 0.1$), to about 50% in supersonic flow.

Keywords Wall turbulence · Compressible flows · Implicit schemes

1 Introduction

Compressible wall-bounded flows play an important role in many aerospace applications of industrial and academic interest. The direct numerical solution (DNS) of the compressible Navier–Stokes equations for wall-bounded turbulent flows has recently become affordable owing to the large increase in available computer power, and canonical incompressible flows

✉ Davide Modesti
davide.modesti@lecnam.net

¹ Cnam-Laboratoire DynFluid, 151 Boulevard de L'Hopital, 75013 Paris, France

² Dipartimento di Ingegneria Meccanica e Aerospaziale, Sapienza Università di Roma, Via Eudossiana 18, 00184 Rome, Italy

have been simulated up to high Reynolds number [2]. However, it is known that the numerical solution of the compressible Navier–Stokes equations is significantly more time consuming than their incompressible counterpart, partly owing to the inherently higher number of floating point operations (flops) per grid point, but mainly because of the much smaller time step imposed by the acoustic stability restriction. In free-shear flows, conventional explicit algorithms can still be used efficiently as long as the typical Mach number is of the order of unity. However, wall-bounded flows inevitably include regions with near stagnant flow and tiny grid spacing adjacent to solid surfaces, which makes the acoustic time step limitation in the wall-normal direction dominant, even at high bulk Mach numbers. Besides being dictated by stability considerations, time step limitations in turbulent flows also have a physical interpretation, as in order to capture the relevant physics of transport phenomena with given speed (say U) on a mesh with given size (say Δ), time steps no larger than Δ/U should be used. Hence, CFL numbers (defined as the ratio of the time advancement step to the maximum allowed time step for explicit time integration) should always be of the order of unity for genuine DNS. In compressible flows, information simultaneously propagate at the hydrodynamic and at the acoustic speed. However, acoustic waves typically make a negligible contribution to the overall energetics of turbulent flows [3]. Hence, with the obvious exception of cases where acoustic instabilities play an important role, such as in certain combustion applications [4] or in direct simulation of aerodynamic noise [5], using a time step which allows to resolve the hydrodynamic (vortical) mode while giving up accurate representation of acoustic phenomena may be a legitimate choice, which actually subtends much of the research carried out for low-speed solvers.

It is the goal of this paper to develop a numerical algorithm for direct numerical simulation of compressible flow which is capable of seamless efficient operation throughout the Mach number range, down to nearly incompressible conditions. The algorithm is at the same time meant to remove or at least alleviate the acoustic time step limitation in the presence of solid boundaries. To gain a clearer perception for the problem, we refer to a canonical compressible boundary layer flow over a flat surface, or flow in a planar channel. Let Δx , Δz be the mesh spacings in the streamwise and spanwise directions, respectively, and let Δy be the minimum mesh spacing in the wall-normal direction, assuming unit CFL number, the time step limitations associated with the discretization of the convective terms in the coordinate directions are

$$\begin{aligned} \Delta t_x^+ &= \frac{\Delta x^+}{\max(u_0^+ + c_0^+, c_w^+)} = \Delta x^+ M_0 \sqrt{C_f/2} \min\left(1, \frac{1}{1 + M_0} \sqrt{T_w/T_0}\right), \\ \Delta t_y^+ &= \frac{\Delta y^+}{c_w^+} = \Delta y^+ M_0 \sqrt{C_f/2} \\ \Delta t_z^+ &= \frac{\Delta z^+}{\max(c_0^+, c_w^+)} = \Delta z^+ M_0 \sqrt{C_f/2} \min\left(1, \sqrt{T_w/T_0}\right), \end{aligned} \tag{1}$$

where the ‘+’ superscript is used to denote quantities made nondimensional with respect to local wall units, namely the friction velocity $u_\tau = (\tau_w/\rho_w)^{1/2}$, and the viscous length scale $\delta_v = \nu_w/u_\tau$, the subscript 0 is used to denote flow properties at the centerline (for channels) and at the free-stream (for boundary layers), and w to denote wall properties, with $C_f = 2\tau_w/(\rho_0 u_0^2)$ the friction coefficient. ρ , u , T and ν denote the density, velocity, temperature and kinematic viscosity, whereas M , c and τ_w are the Mach number, speed of sound and wall stress, respectively. It should be noted that if acoustic waves are suppressed, as is the case of strictly incompressible flow, the time step is controlled by the streamwise

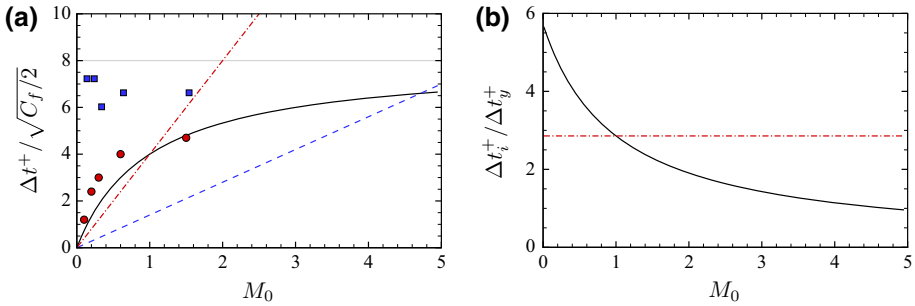


Fig. 1 Inviscid time step limitation in the coordinate directions as from Eq. (1) as a function of the reference Mach number M_0 . In panel **a** we show Δt_x (solid), Δt_y (dashed), Δt_z (dot-dashed). In panel **b** we show the ratios $\Delta t_x/\Delta t_y$ (solid), $\Delta t_z/\Delta t_y$ (dot-dashed). For reference, in panel **a** we report with a grey line the ‘incompressible’ time limitation given in Eq. (2). The symbols denote the time step limits for the present semi-implicit algorithm as dictated by accuracy (circles) and stability (squares), as discussed in Sect. 3.2

direction, and

$$\Delta t_f^+ = \Delta x^+ \sqrt{C_f/2}. \tag{2}$$

The viscous time step limitation is mainly effective in the wall-normal direction, and in wall units one has

$$\Delta t_{yv}^+ = \Delta y^+{}^2. \tag{3}$$

For the sake of graphical representation of the above formulas, we assume: (i) the distance of the first point from the wall is $\Delta y_w^+ \approx 0.7$, which is the maximum value for which accurate turbulence statistics are obtained [6]; (ii) the minimum mesh spacing in the wall-normal direction is $\Delta y = 2\Delta y_w$, which can be achieved by staggering the mesh in the vertical direction, thus alleviating the stability restrictions [6]; (iii) the wall-parallel mesh spacings are $\Delta x^+ = 8$, $\Delta z^+ = 4$, which is typical for DNS; iv) the wall is isothermal, with $T_w = T_0$. Figure 1 shows the inviscid time step restrictions according to Eq. (1) as a function of the reference Mach number M_0 , scaled by $\sqrt{C_f}/2$ (panel a), and as a fraction of the wall-normal allowed time step (panel b). Inefficiency of explicit compressible solvers is apparent in the low-Mach-number regime, where vanishingly small time steps are required. Time steps comparable to those achievable in incompressible flow are only possible starting at $M_0 \approx 3$. With the exception of hypersonic flow, the most restrictive time limitation is that associated with the vertical direction, and an increase by at least a factor of two can be gained by removing it (see panel b). It is also interesting to note that the acoustic time limitation in the spanwise size is more restrictive than the streamwise limitation up to $M_0 \approx 1$, whereas at supersonic Mach numbers the convective limitation in x is controlling. Removing the wall-normal acoustic time limitation in supersonic flow is sufficient to achieve a similar time step as in incompressible flow, whereas in subsonic flow it is also necessary to remove the acoustic time restriction in the wall-parallel directions. We further note that the normalized viscous time limitation $\Delta t_{yv}^+/\sqrt{C_f}/2$, with Δt_{yv}^+ given in Eq. (3) is always much weaker than the convective ones, provided $\Delta y^+ \sim 1$, and considering that the range of friction coefficients typically accessed by DNS is $2 \times 10^{-3} \leq C_f \leq 6 \times 10^{-3}$. The above estimates are deduced for typical DNS mesh spacings, but the case of wall-resolved RANS, LES and DES is even more severe, as the aspect ratio of near-wall cells is substantially higher, hence making suppression of the wall-normal time step restriction mandatory for any practical calculation. This is even more important in the case of curvilinear coordinate

systems with singular metrics. For instance, in the case of cylindrical coordinates the mesh spacing (hence, the allowed time step for explicit stepping) becomes exceedingly small in the azimuthal direction, even in the absence of walls.

The above-mentioned difficulties are well-known to the CFD community, and a variety of techniques have been developed to cope with the numerical stiffness of the compressible Navier–Stokes equations. The chief choice in this respect has traditionally been the use of (semi-)implicit time integration schemes. A landmark contribution in this sense was given by [7,8], who proposed a time-implicit algorithm for the solution of the Navier–Stokes equations in conservative form based on linearization of the convective and viscous flux vectors, coupled with approximate factorization [9] to handle multiple space dimensions. However, the method is computationally expensive as it requires the inversion of 5×5 block-banded systems of equations, which is more expensive than, e.g. standard banded systems. In this respect we note that, whereas the classical Thomas algorithm for tridiagonal matrices requires a number of floating point operations (flops) of $O(6N)$ (where N is the number of grid points in a given coordinate direction), its block-tridiagonal version requires $O(3N(M^3 + M^2))$ flops, where M ($=5$ in the Beam–Warming algorithm) is the size of each block [10]. The computational cost is about twice as much in the case of periodic boundary conditions [11]. Pulliam and Chaussee [12] developed a variant of the Beam–Warming algorithm which involves the inversion of standard tridiagonal systems rather than block matrices, with large saving of computer time, but with loss of accuracy and stability in the case of unsteady simulations [13]. Algorithms of the Beam–Warming family are at the heart of highly successful aerospace CFD software [14,15]. For instance, [16] have recently developed an implicit fourth-order Runge–Kutta scheme, using residual smoothing based on a bilaplacian operator. The algorithm requires the inversion of one scalar pentadiagonal system in each space direction per sub-step, allowing for net speedup of 3–5 with respect to the explicit case. Algorithms which avoid inversion of banded systems of equations have also been designed [17], which may be useful for efficient parallel implementation. However, those algorithms require point-wise iterative procedures whereby the right-hand-side of the equations must be evaluated several times per time step, with unclear outcome in terms of overall efficiency.

Alternative approaches to circumvent the stiffness of compressible Navier–Stokes equations rely on the use of pre-conditioning techniques, based on the attempt to change the eigenvalues of the system of equations in order to remove the large disparity of wave speeds. This is accomplished by pre-multiplying the time derivatives by a matrix that slows the speed of the acoustic waves down toward the fluid speed [18,19]. Preconditioning is the choice of election for steady-state application, however its extension to unsteady flow problem is not straightforward, requiring the use of dual time stepping techniques, namely inner iterations in terms of a pseudo-time [20–22]. However, the number of iterations per physical time step can be very large, with subsequent loss of computational efficiency.

An important class of algorithms for low-Mach-number flows relies on the concept of asymptotic consistency [23]. The main idea is that the discrete compressible Navier–Stokes equations should automatically reduce to their discrete incompressible counterpart as the Mach number approaches zero. Examples of this approach include use of the compressible isentropic Navier–Stokes equations which are then discretized with a semi-implicit scheme, yielding a formulation similar to the classical projection method for incompressible flow [24,25]. Asymptotic-preserving methods are certainly interesting as they allow to entirely remove the acoustic time step limitation, but they suffer from the same drawback of incompressible solvers, as in the incompressible limit they require the solution of Poisson equation, which is computationally expensive in multiple space dimensions.

Specialized algorithms for the Navier–Stokes equations have been also developed for the low-Mach number regime, which allow to account for temperature-dependent density variations, as is typically the case in combustion. All these variable-density algorithms are based on the idea the only the terms which bring an acoustic contribution should be advanced implicitly in time, in such a way that the acoustic time limitation is removed. Numerical schemes of this kind were pioneered by [26], who proposed to treat implicitly only the pressure term in the momentum equation and the dilatation term in the internal energy equation, which results in having to solve an elliptic equation for pressure, with large incurred overhead. [27,28] extended the classical pressure-correction method [29] to variable-density flows by solving a Helmholtz equation for the pressure correction, and the use of sub-iterations. LES results were carried out in which a time step forty times larger than the explicit case was achieved, with modest computational cost overhead. Moureau et al. [30] developed an implicit scheme for the removal of the acoustic limitation which also relies on the solution of a Helmholtz equation, however without reverting to sub-iterations, with an overhead CPU time of about 25% with respect to standard incompressible solvers. Hence it appears that, in one way or another, algorithms tailored for the near-incompressible regime involve either iterative procedures and/or the inversion of elliptic systems of equations. The latter can only be carried out efficiently in the case that periodic directions are present, which allows for the use of direct solvers [31].

In this paper we develop a novel semi-implicit algorithm for the compressible Navier–Stokes equations based on a modification of the basic Beam–Warming linearization, thus avoiding any iterative procedure. The algorithm is presented in Sect. 2, which also includes a discussion of the treatment of viscous terms, accurate time integration by the use of a third-order semi-implicit Runge–Kutta scheme, and extension to multiple space dimensions. Numerical examples are given in Sect. 3, which include DNS of turbulent flows from the low subsonic to the supersonic regime. Final remarks and suggestions for future work are given in Sect. 4.

2 Formulation of the Algorithm

The Navier–Stokes equations for a compressible perfect gas are considered in which the total energy equation is replaced with the entropy equation

$$\frac{\partial \mathbf{w}}{\partial t} = - \sum_{i=1}^3 \frac{\partial \mathbf{f}_i}{\partial x_i} + \sum_{i=1}^3 \frac{\partial \mathbf{f}_i^v}{\partial x_i} + \mathbf{S} = \mathbf{R}, \tag{4}$$

where \mathbf{R} is the right hand side of Eq. 4, \mathbf{w} is the vector of the conserved variables, \mathbf{f}_i and \mathbf{f}_i^v are the convective and viscous fluxes in the i th direction, with x, y, z the streamwise, wall normal and spanwise directions and \mathbf{S} the source terms in the entropy equation,

$$\mathbf{w} = \begin{bmatrix} \rho \\ \rho u_j \\ \rho s \end{bmatrix}, \quad \mathbf{f}_i = \begin{bmatrix} \rho u_i \\ \rho u_i u_j + p \delta_{ij} \\ \rho u_i s \end{bmatrix}, \quad \mathbf{f}_i^v = \begin{bmatrix} 0 \\ \sigma_{ij} \\ -q_i/T \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{\sigma_{\ell m}}{T} \frac{\partial u_\ell}{\partial x_m} - \frac{q_\ell}{T^2} \frac{\partial T}{\partial x_\ell} \end{bmatrix}, \tag{5}$$

where ρ is the density, p is the pressure, T is the temperature and $u_i, i = 1, 2, 3$ the velocity components in the i th direction (also denoted as u, v, w in the following), $s = c_v \ln(p\rho^{-\gamma})$

is the entropy per unit mass, q_i and σ_{ij} are the components of the viscous stress tensor and heat flux,

$$\sigma_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right), \quad q_i = -k \frac{\partial T}{\partial x_i}, \tag{6}$$

where μ is the dynamic viscosity, $k = \mu c_p / Pr$ the thermal conductivity and $Pr = 0.72$ the molecular Prandtl number.

As shown in the following, the use of the entropy equation is instrumental to achieving efficient implicit treatment of the acoustic terms, and also yield benefits in terms of increased robustness as compared to algorithms solving for the energy equation [32,33]. On the other hand, this setting prevents correct capturing of shock waves, as the entropy equation cannot be used as conservation law [33,34], hence in the following we restrict ourselves to discussing the case of smooth compressible flows.

2.1 Implicit Treatment of Acoustic Waves

In order to remove (or at least alleviate) the time acoustic time step limitation in the generic coordinate direction (say, y), we proceed by splitting the convective flux vector into a purely advective part, and a part which supports acoustic fluctuations, namely

$$\mathbf{f}_y = \mathbf{f}_y^c + \mathbf{f}_y^a, \quad \mathbf{f}_y^c = \begin{bmatrix} 0 \\ \rho uv \\ \rho v^2 \\ \rho vw \\ \rho vs \end{bmatrix}, \quad \mathbf{f}_y^a = \begin{bmatrix} \rho v \\ 0 \\ p \\ 0 \\ 0 \end{bmatrix}. \tag{7}$$

In a linearized setting, this splitting yields full decoupling of the acoustic, vortical and entropy modes [35]. Of course, such decoupling does not directly extend to the nonlinear case, and the ansatz 7 is mainly instrumental to try to suppress the acoustic time limitation, its validity resting in success, to be judged a-posteriori. The main advantage for numerical purposes is that the acoustic partial flux Jacobian has a simple structure,

$$\mathbf{A}_y^a = \frac{\partial \mathbf{f}_y^a}{\partial \mathbf{w}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{p}{\rho} \left(\gamma - \frac{s}{C_v} \right) & 0 & 0 & 0 & \frac{p}{\rho C_v} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{8}$$

Splitting of the flux vectors into pressure and velocity contributions was previously considered by [36,37], based on the attempt to reduce the block size in the implicit operator as compared to the Beam–Warming algorithm. In essence, these decompositions amounted [14] to isolating the pressure gradient in the momentum equation and the pressure flux in the total energy equation. However, besides being consistent with wave decomposition in a linear setting, we find the splitting (8) to be vastly more robust in practice.

We proceed to discretize Eq. (4) between two consecutive time levels n and $n + 1$, by evaluating explicitly the advective partial flux, and evaluating the acoustic partial flux implicitly, upon linearization about time level n , namely

$$\mathbf{f}_y^{an+1} = \mathbf{f}_y^{an} + \mathbf{A}_y^{an} (\mathbf{w}^{n+1} - \mathbf{w}^n) + O(\Delta t^2), \tag{9}$$

thus obtaining

$$\left(\mathbf{I} + \Delta t \frac{\partial}{\partial y} \mathbf{A}_y^{an} \right) \Delta \mathbf{w}^n = -\Delta t \frac{\partial \mathbf{f}_y^n}{\partial y} + \Delta t \mathbf{F}_{xz}^n = \Delta t \mathbf{R}^n, \tag{10}$$

where $\Delta \mathbf{w}^n = \mathbf{w}^{n+1} - \mathbf{w}^n$, and where terms containing transverse flux derivatives and viscous terms are lumped together into \mathbf{F}_{xz} . It is important to note that, because of the special structure of the acoustic flux Jacobian, the inversion of Eq. (10) is much simpler than for the standard Beam–Warming algorithm, which relies on linearization of the full convective flux. Component-wise, Eq. (10) reads

$$\begin{cases} \Delta w_1^n + \Delta t \frac{\partial}{\partial y} \Delta w_3^n = \Delta t R_1^n & (11a) \\ \Delta w_2^n = \Delta t R_2^n & (11b) \\ \Delta w_3^n + \Delta t \frac{\partial}{\partial y} (A_{y31}^{an} \Delta w_1^n) + \Delta t \frac{\partial}{\partial y} (A_{y35}^{an} \Delta w_5^n) = \Delta t R_3^n & (11c) \\ \Delta w_4^n = \Delta t R_4^n & (11d) \\ \Delta w_5^n = \Delta t R_5^n. & (11e) \end{cases}$$

Hence, the time increments of entropy and of the transverse velocity components can be evaluated explicitly, thus effectively reducing the system of equations to be solved to

$$\begin{cases} \Delta w_1^n + \Delta t \frac{\partial}{\partial y} \Delta w_3^n = \Delta t R_1^n & (12a) \\ \Delta w_3^n + \Delta t \frac{\partial}{\partial y} (A_{y31}^{an} \Delta w_1^n) = \Delta t R_3^n - \Delta t \frac{\partial}{\partial y} (A_{y35}^{an} \Delta w_5^n) = \Delta t \widehat{R}_3^n, & (12b) \end{cases}$$

which, upon discretization of the space derivative operators, yields a 2×2 block-banded system of equations, whose solution returns the time increments of ρ and ρv . Equation (12) can be further rearranged by formally solving for Δw_1^n in (12a), to obtain

$$\left(1 - \Delta t^2 A_{y31}^{an} \frac{\partial^2}{\partial y^2} - \Delta t^2 \frac{\partial A_{y31}^{an}}{\partial y} \frac{\partial}{\partial y} \right) \Delta w_3^n = \Delta t \widehat{R}_3^n - \Delta t^2 \frac{\partial}{\partial y} \left(A_{y31}^{an} R_1^n \right), \tag{13}$$

whose solution requires the inversion of a single ordinary banded system of equations, with bandwidth depending on the accuracy in the approximation of the first and second space derivative operators. Back substitution into (12a) then returns the time increment of density. Although apparently cumbersome, we find the latter formulation to be more computationally efficient than the solution of the 2×2 block system given by Eq. (12), while the accuracy is nearly identical. Hence, Eq. (13) is used in all the forthcoming numerical applications.

2.2 Implicit Treatment of Viscous Terms

If needed, viscous terms can also be handled implicitly, using approximate factorization. For that purpose, we split the viscous flux derivatives in Eq. (4) into a Laplacian term and a difference thereof

$$\frac{\partial \mathbf{f}_y^v}{\partial y} = \mu \frac{\partial^2 \mathbf{v}}{\partial y^2} + \varphi_y^v, \tag{14}$$

where \mathbf{v} is the vector of primitive variables, $\mathbf{v} = [\rho, u, v, w, T]$, and $\boldsymbol{\mu}$ is the viscosity matrix,

$$\boldsymbol{\mu} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & \frac{\mu C_p}{PrT} \end{bmatrix}. \tag{15}$$

Freezing for simplicity the viscosity matrix at time step n , the following linearization is considered,

$$\left(\boldsymbol{\mu} \frac{\partial^2 \mathbf{v}}{\partial y^2} \right)^{n+1} \approx \left(\boldsymbol{\mu} \frac{\partial^2 \mathbf{v}}{\partial y^2} \right)^n + \boldsymbol{\mu}^n \frac{\partial^2 \mathbf{P} \Delta \mathbf{w}^n}{\partial y^2}, \tag{16}$$

where \mathbf{P} is the Jacobian of the conservative-to-primitive variables transformation

$$\mathbf{P} = \frac{\partial \mathbf{v}}{\partial \mathbf{w}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -\frac{u}{\rho} & \frac{1}{\rho} & 0 & 0 & 0 \\ -\frac{v}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 \\ -\frac{w}{\rho} & 0 & 0 & \frac{1}{\rho} & 0 \\ -\frac{T_s}{\rho c_v} & 0 & 0 & 0 & \frac{T}{\rho c_v} \end{bmatrix}. \tag{17}$$

Following similar steps as done to arrive at Eq. (10), the previous linearization yields

$$\left(\mathbf{I} + \Delta t \frac{\partial}{\partial y} \mathbf{A}_y^{an} - \Delta t \boldsymbol{\mu}^n \frac{\partial^2}{\partial y^2} \mathbf{P}^n \right) \Delta \mathbf{w}^n = \Delta t \mathbf{R}^n, \tag{18}$$

which can be approximately factorized as follows

$$\left(\mathbf{I} + \Delta t \frac{\partial}{\partial y} \mathbf{A}_y^{an} \right) \left(\mathbf{I} - \Delta t \boldsymbol{\mu}^n \frac{\partial^2}{\partial y^2} \mathbf{P}^n \right) \Delta \mathbf{w}^n = \Delta t \mathbf{R}^n. \tag{19}$$

Inversion of Eq. (19) can be then carried out into two sequential sub-steps,

$$\left(\mathbf{I} + \Delta t \frac{\partial}{\partial y} \mathbf{A}_y^{an} \right) \widetilde{\Delta \mathbf{w}}^n = \Delta t \mathbf{R}^n, \tag{20}$$

$$\left(\mathbf{I} - \Delta t \boldsymbol{\mu}^n \frac{\partial^2}{\partial y^2} \mathbf{P}^n \right) \Delta \mathbf{w}^n = \widetilde{\Delta \mathbf{w}}^n, \tag{21}$$

whereby the provisional time increment $\widetilde{\Delta \mathbf{w}}^n$ is first evaluated through the inversion procedure for the convective fluxes described in Sect. 2.1. The actual time increment $\Delta \mathbf{w}^n$ is then evaluated by inverting the viscous implicit operator at the left-hand-side of Eq. (21) which, in light of the special structure of the Jacobian matrix given in Eq. (17), can be carried out sequentially, as follows

$$\left\{ \begin{array}{l} \Delta w_1^n = \widetilde{\Delta w}_1^n \tag{22a} \\ \left(1 - \mu_{22}^n \Delta t \frac{\partial^2}{\partial y^2} P_{22}^n \right) \Delta w_2^n = \widetilde{\Delta w}_2^n + \mu_{22}^n \Delta t \frac{\partial^2}{\partial y^2} (P_{21}^n \Delta w_1^n) \tag{22b} \\ \left(1 - \mu_{33}^n \Delta t \frac{\partial^2}{\partial y^2} P_{33}^n \right) \Delta w_3^n = \widetilde{\Delta w}_3^n + \mu_{33}^n \Delta t \frac{\partial^2}{\partial y^2} (P_{31}^n \Delta w_1^n) \tag{22c} \\ \left(1 - \mu_{44}^n \Delta t \frac{\partial^2}{\partial y^2} P_{44}^n \right) \Delta w_4^n = \widetilde{\Delta w}_4^n + \mu_{44}^n \Delta t \frac{\partial^2}{\partial y^2} (P_{41}^n \Delta w_1^n) \tag{22d} \\ \left(1 - \mu_{55}^n \Delta t \frac{\partial^2}{\partial y^2} P_{55}^n \right) \Delta w_5^n = \widetilde{\Delta w}_5^n + \mu_{55}^n \Delta t \frac{\partial^2}{\partial y^2} (P_{51}^n \Delta w_1^n) \tag{22e} \end{array} \right.$$

The inversion of four standard narrow-banded systems of equations is thus required for the purpose. We point out that the present procedure is again different than the original Beam–Warming procedure, which relies on linearization of the full viscous flux vectors, hence requiring the inversion of block-banded systems. However, we have found that numerical robustness is very weakly affected by the approximations herein made.

2.3 Multiple Space Dimensions

As done for the case of a single space dimension, the acoustic and viscous time limitations can be removed in more than one direction through direction-wise factorization of the implicit operators. For instance, assuming that all space directions are handled in semi-implicit fashion, Eq. (19) is replaced by

$$\mathbf{L}^n \Delta \mathbf{w}^n = \mathbf{R}^n, \tag{23}$$

where

$$\begin{aligned} \mathbf{L}^n = & \left(\mathbf{I} + \Delta t \frac{\partial}{\partial x} \mathbf{A}_x^{an} \right) \left(\mathbf{I} + \Delta t \frac{\partial}{\partial y} \mathbf{A}_y^{an} \right) \left(\mathbf{I} + \Delta t \frac{\partial}{\partial z} \mathbf{A}_z^{an} \right) \cdot \\ & \left(\mathbf{I} - \Delta t \mu^n \frac{\partial^2}{\partial x^2} \mathbf{P}^n \right) \left(\mathbf{I} - \Delta t \mu^n \frac{\partial^2}{\partial y^2} \mathbf{P}^n \right) \left(\mathbf{I} - \Delta t \mu^n \frac{\partial^2}{\partial z^2} \mathbf{P}^n \right). \end{aligned} \tag{24}$$

Hence, repeated application of the procedures developed in the previous two sections is sufficient. Practical application of Eq. (24) requires some caution, as the order in which the various inversions are carried out is not immaterial. We have found that, in order to remove possible spurious anisotropies, it is a good practice to shuffle the order of the implicit left-hand-side operators.

2.4 Time Integration

Time accuracy and stability enhancement is typically obtained by Runge–Kutta schemes as wrapper to one-step implicit procedures outlined in the previous paragraphs. Low-storage algorithms are a popular choice, and here we consider for example Wray’s three-stage, third-order scheme [38], adapted to semi-implicit integration of the convective terms,

$$\mathbf{L}^{(\ell)} \Delta \mathbf{w}^{(\ell)} = \alpha_\ell \Delta t \mathbf{R}^{(\ell-1)} + \beta_\ell \Delta t \mathbf{R}^{(\ell)}, \quad \ell = 0, 1, 2, \tag{25}$$

where $\Delta \mathbf{w}^{(\ell)} = \mathbf{w}^{(\ell+1)} - \mathbf{w}^{(\ell)}$, $\mathbf{w}^{(0)} = \mathbf{w}^n$, $\mathbf{w}^{n+1} = \mathbf{w}^{(3)}$, the left-hand-side implicit operator is a generalization of Eq. (24), namely

$$\begin{aligned} \mathbf{L}^{(\ell)} = & \left(\mathbf{I} + \gamma_\ell \Delta t \frac{\partial}{\partial x} \mathbf{A}_x^{a(\ell)} \right) \left(\mathbf{I} + \gamma_\ell \Delta t \frac{\partial}{\partial y} \mathbf{A}_y^{a(\ell)} \right) \left(\mathbf{I} + \gamma_\ell \Delta t \frac{\partial}{\partial z} \mathbf{A}_z^{a(\ell)} \right) \\ & \cdot \left(\mathbf{I} - \gamma_\ell \Delta t \mu^{(\ell)} \frac{\partial^2}{\partial x^2} \mathbf{P}^{(\ell)} \right) \left(\mathbf{I} - \gamma_\ell \Delta t \mu^{(\ell)} \frac{\partial^2}{\partial y^2} \mathbf{P}^{(\ell)} \right) \left(\mathbf{I} - \gamma_\ell \Delta t \mu^{(\ell)} \frac{\partial^2}{\partial z^2} \mathbf{P}^{(\ell)} \right), \end{aligned}$$

and the integration coefficient are $\alpha_\ell = (0, 17/60, -5/12)$, $\beta_\ell = (8/15, 5/12, 3/4)$, $\gamma_\ell = \alpha_\ell + \beta_\ell$. We have found this time stepping scheme to work well in practice, however because of the partial flux linearization, the method is only formally first-order accurate in time.

Higher order of accuracy in time can be achieved using a third-order accurate semi-implicit Runge–Kutta scheme [1], which can be conveniently cast as follows

$$\begin{cases} \mathbf{L}^n \Delta \mathbf{w}^{(1)} = \frac{2}{3} \Delta t \mathbf{R}^n & (26a) \\ \mathbf{L}^{(1)} \Delta \mathbf{w}^{(2)} = -(\mathbf{w}^{(1)} - \mathbf{w}^n) + \frac{1}{3} \Delta t \mathbf{R}^n + \frac{1}{3} \Delta t \mathbf{R}^{(1)} & (26b) \\ \Delta \mathbf{w}^{(3)} = \frac{1}{2} (\mathbf{w}^{(2)} - \mathbf{w}^n) - \frac{3}{2} \alpha \Delta \mathbf{w}^{(2)} & (26c) \\ \mathbf{L}^{(3)} \Delta \mathbf{w}^{(4)} = -(\mathbf{w}^{(3)} - \mathbf{w}^n) + \frac{1}{4} \Delta t \mathbf{R}^n + \frac{3}{4} \Delta t \mathbf{R}^{(1)} & (26d) \\ \mathbf{L}^{(4)} \Delta \mathbf{w}^{(5)} = -(\mathbf{w}^{(4)} - \mathbf{w}^n) + \frac{1}{4} \Delta t \mathbf{R}^n + \frac{3}{4} \Delta t \mathbf{R}^{(2)}, & (26e) \end{cases}$$

where $\gamma_\ell = \gamma$ is the same for all sub-steps, and α are free parameters (hereafter, we assume $\alpha = 1, \gamma = 0.6$). With respect to Wray’s algorithm, Eq. (26) is not in low-storage form (although it can be implemented using three arrays only), and it involves an additional inversion, to achieve third-order regardless of the linear operator $\mathbf{L}^{(\ell)}$ in Eq. (26), but no additional evaluation of the explicit operator. Extensive accuracy tests of the method are reported in the original reference.

2.5 Stability Analysis

The stability of the semi-implicit algorithm herein developed is here analyzed within the simplified setting of the linearized inviscid acoustic equations in the presence of a mean flow u_0 , which can be cast as

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{A} \frac{\partial \mathbf{v}}{\partial x} = 0, \quad \mathbf{v} = \begin{bmatrix} \rho' \\ u' \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} u_0 & \rho_0 \\ c_0^2/\rho_0 & u_0 \end{bmatrix}, \quad (27)$$

where the subscript 0 refers to the unperturbed state, and primes to fluctuations thereof. A semi-implicit discretization of (27) can be obtained by considering the linearized counterpart of the partial flux Jacobian (8), namely

$$\mathbf{A}^a = \begin{bmatrix} u_0 & \rho_0 \\ c_0^2/\rho_0 & 0 \end{bmatrix}. \quad (28)$$

Backward Euler discretization of Eq. (27) then yields

$$\left(\mathbf{I} - \Delta t \mathbf{A}^a \frac{\partial}{\partial x} \right) \Delta \mathbf{v}^n = -\Delta t \mathbf{A} \frac{\partial \mathbf{v}^n}{\partial x}. \quad (29)$$

Transforming Eq. (29) to Fourier space with the token $\mathbf{v}(x, t) = \hat{\mathbf{v}}(t)e^{ikx}$ yields the amplification matrix of the scheme

$$\mathbf{G} = \mathbf{I} - \left(\mathbf{I} - i \Delta t \tilde{k} \mathbf{A}^a \right)^{-1} i \Delta t \tilde{k} \mathbf{A}, \quad (30)$$

where $\mathbf{v}^{n+1} = \mathbf{G} \mathbf{v}^n$, and \tilde{k} is the modified wavenumber corresponding to the discretization of the space first derivative operator [39]. Von Neumann’s stability condition requires that both eigenvalues of \mathbf{G} are no larger than unity in modulus. Assuming for instance second-order central differencing (i.e. $\tilde{k}h = \sin(kh)$), it turns out that the scheme (29) is unconditionally stable for $M_0 = u_0/c_0 \lesssim 1$. A similar analysis can be carried out (details are omitted) for the Runge–Kutta time stepping scheme of Eq. (26). In the case of explicit time integration

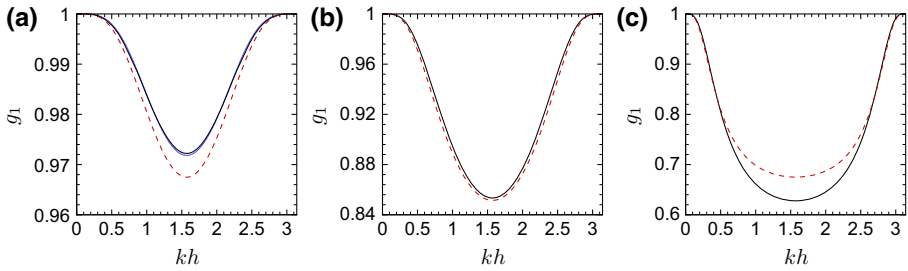


Fig. 2 Smallest eigenvalue of amplification matrix at CFL = 1 (a), CFL = 2 (b), CFL = 5 (c), for explicit Runge–Kutta time integration (dotted lines), semi-implicit time integration (with $\alpha = 1, \gamma = 0.6$, solid lines), and fully implicit Beam–Warming scheme (dashed lines), at Mach number $M_0 = 0.3$. Curves are only shown for stable schemes

(i.e. $\gamma = 0$) the scheme is stable for $CFL \lesssim \sqrt{3}$, where $CFL = (u_0 + c_0)\Delta t/h$. In the case of semi-implicit time integration (with $\gamma = 0.6, \alpha = 1$) unconditional stability is achieved for $M_0 \lesssim 0.525$.

To provide an idea of the accuracy of the algorithm, in Fig. 2 we show the smallest eigenvalues of the amplification matrix at various Courant numbers for explicit and semi-implicit Runge–Kutta time integration. For reference, the amplification factor of the baseline Beam–Warming algorithm is also shown. At CFL numbers lower than the stability limit for explicit discretization [panel (a)], the semi-implicit and the fully explicit algorithms have similar performance, whereas the Beam–Warming algorithm has somewhat higher diffusion. At higher Courant numbers the explicit scheme goes unstable, and semi-implicit and fully implicit scheme have similar performance, with slightly less diffusive behavior of Beam–Warming at higher CFL. Notably, all schemes have unit amplification factor at the Niquist limit ($kh = \pi$), hence they are not dissipative in the sense of Kreiss. This is the reason why schemes of the Beam–Warming family are typically used with explicit addition of artificial diffusion terms [7, 13].

2.6 Spatial Discretization

All the convective derivatives at the right-hand-side operator defined in Eq. (4) are discretized using conservative, energy-preserving formulas [40], based on application of standard central difference approximations to the fully expanded form of the convective derivatives [41]. In the explicit case this discretization allows to exactly preserve the total kinetic energy from convection, and conserve the entropy variance in the inviscid limit, hence providing strong nonlinear stability to the algorithm without introducing any numerical diffusion [33, 42]. We have found that this feature is very important to prevent nonlinear divergence caused by accumulation of aliasing errors, especially in light of the fact that the semi-implicit algorithms herein dealt with have zero numerical diffusion at the highest resolved wavenumbers. Hence, no explicit addition of artificial diffusion is needed for the semi-implicit algorithm herein developed. Viscous terms are also expanded to Laplacian form and discretized by means of central formulas [43].

Previous studies [44–46] have shown that second-order spatial accuracy is sufficient for DNS of turbulence, provided energy-consistent discretizations are used. Hence, in the present work we focus on second-order space discretizations, which only require the inversion of standard tridiagonal matrices. Extension of the algorithm to higher-order spatial accuracy is straightforward, and it can be achieved by either reverting to compact-

Table 1 Computational cost for implicit schemes compared to fully explicit discretization

Scheme	CPU/CPU _{EXPL}
EXPL	1
ATI	1.14
ATI-CYC	1.16
AVTI	1.32
AVTI-CYC	1.37
BW	1.67
BW-CYC	2.21
BWV	1.87
BWV-CYC	2.33

Figures refer to implicit treatment of a single space direction

difference approximations [13], or by widening the discretization stencil. In the latter case, higher-order accuracy in space can be achieved at the price of inverting wider banded matrices.

2.7 Computational Efficiency

Achieving higher computational efficiency is obviously the main motivation for using implicit algorithms, which are inherently more computationally intensive than explicit ones. Computational cost figures for the present semi-implicit algorithm and for the Beam–Warming scheme are listed in Table 1, as a fraction of the cost for the baseline explicit algorithm. Cost estimates are given for implicit treatment of convective terms only, and for simultaneous treatment of convective and viscous terms, referring to a single space direction. Also for ease of later reference, we use the following notation to distinguish the various schemes. The semi-implicit scheme herein developed is referred to as either acoustic terms-implicit (ATI, as in Eq. (10)), or ATVI in the case that both convective and viscous terms are handled implicitly (Eq. 18). As a basis of comparison, cost figures for the Beam–Warming (BW) scheme, also with implicit treatment of the viscous terms (BWV) are reported. Cost figures are provided for both the case of periodic (CYC) and non-periodic boundary conditions. It should be noted that the cost estimates refer to actual parallel computations, and also include the computational overhead for data transposition across processors in non-contiguous space directions. Of course, precise figures may change depending on the specific implementation of the algorithm and/or machine architecture, but we trust that the numbers listed in the table provide a reasonably robust estimate. It appears that the computational overhead of the ATI algorithm is rather limited, hence implicit treatment of a given space direction is computationally advantageous provided the attainable time step is at least 20% higher than for fully explicit. Substantial improvement of computational efficiency over standard Beam–Warming discretization is also apparent, for comparable expected accuracy (recalling Fig. 2).

3 Numerical Results

The performance of the semi-implicit algorithm herein developed is tested through application to a series of canonical compressible turbulent flows, in order of increasing physical complexity.

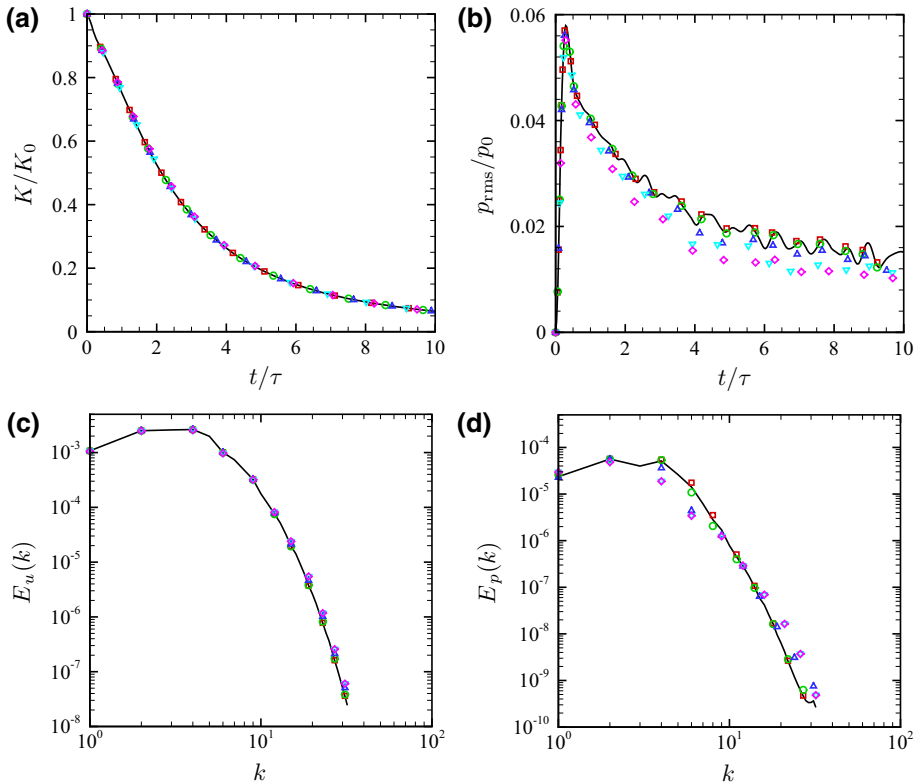


Fig. 3 Numerical simulations of homogeneous isotropic turbulence at $M_t = 0.3$, $k_0 = 4$, $Re_\lambda = 30$, with ATI-XYZ scheme. Time history of turbulence kinetic energy (a), and pressure variance (b), and spectra of velocity (c) and pressure fluctuations (d) at $t/\tau = 5$. Solid lines denote reference results obtained with explicit time discretization at CFL = 1. Symbols denote results obtained with ATI scheme at CFL = 1 (squares), CFL = 2 (circles), CFL = 3 (triangles), CFL = 4 (down-triangles), CFL = 5 (diamonds)

3.1 Isotropic Turbulence

Numerical simulations of homogeneous isotropic turbulence have been frequently carried out to evaluate the properties of numerical schemes for turbulent flows [47]. DNS are here carried out in a triply periodic $(2\pi)^3$ box, discretized with 64^3 collocation points. At the initial time pressure and density are taken to be uniform, and solenoidal velocity perturbations are added according to the procedure introduced by [48], with prescribed three-dimensional energy spectrum

$$E(k) = 16\sqrt{\frac{2}{\pi}} \frac{u_0^2}{k_0} \left(\frac{k^4}{k_0}\right)^4 e^{-2(k/k_0)^2}, \tag{31}$$

where $k_0 = 4$ is the most energetic mode. The initial turbulent Mach number is given by $M_{t0} = \sqrt{3}u_0/c_0 = 0.3$, and the Reynolds number based on the Taylor microscale is $Re_\lambda = 2\rho_0u_0/(\mu_0k_0) = 30$. Time is made nondimensional with respect to the eddy turnover time $\tau = 2\sqrt{3}/(k_0M_{t0}c_0)$.

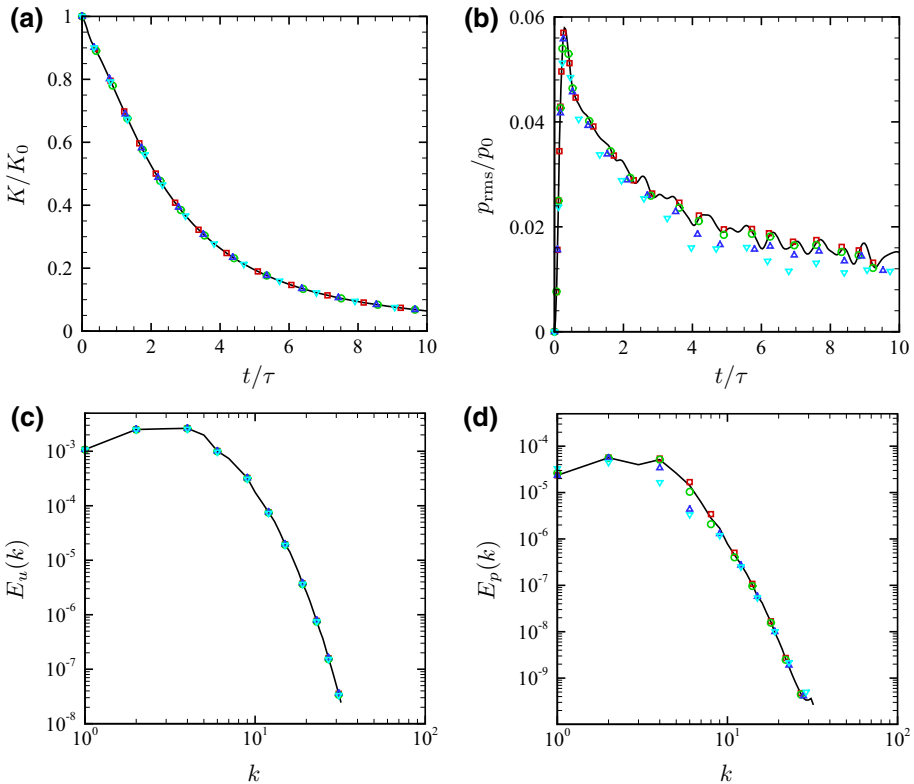


Fig. 4 Numerical simulations of homogeneous isotropic turbulence at $M_t = 0.3$, $k_0 = 4$, $Re_\lambda = 30$, with BW-XYZ scheme. Time history of turbulence kinetic energy (a), and pressure variance (b), and spectra of velocity (c) and pressure fluctuations (d) at $t/\tau = 5$. *Solid lines* denoted reference results obtained with explicit time discretization at CFL = 1. *Symbols* denote results obtained with BW scheme at CFL = 1 (squares), CFL = 2 (circles), CFL = 3 (triangles), CFL = 4 (down-triangles)

The results obtained with ATI and BW discretization in all space directions are shown in Figs. 3 and 4, respectively, at various Courant numbers. Stable results are obtained for $CFL \lesssim 5.1$ for ATI, and $CFL \lesssim 4.8$ for BW. Loss of stability at larger time steps is due to flux linearization and/or factorization errors, which prevent unconditional stability in practical computations [13]. The time behavior of turbulence kinetic energy [panel (a)] is well predicted at all Courant numbers up to the stability limit, whereas pressure fluctuations [panel (b)] are overdamped starting at $CFL \approx 3$, in both ATI and BW. The different behavior is caused by the fact that pressure receives contributions of both hydrodynamic and acoustic nature. As seen in the previous Section, acoustic waves undergo significant damping at high Courant number. This is even clearer in the velocity and pressure spectra, shown in panels (c) and (d), respectively. Although velocity spectra are perfectly captured at all Courant numbers, pressure spectra undergo numerical damping, especially at intermediate wavenumbers, which is easily understood based on the amplification factors shown in Fig. 2. Given the similar performance of the two implicit methods for this test case, ATI is certainly preferable owing to its lower computational cost, which allows to achieve an effective speed-up over the explicit case (see Table 1) of about a factor of three, whereas BW yields almost the same efficiency.

3.2 Turbulent Flow in Plane Channel

Channel flow is the simplest prototype of wall-bounded flows, and it has been studied by many authors in the incompressible [2, 45, 49], as well as in the compressible regime [6, 50, 51]. The controlling parameters are the bulk Mach number $M_b = u_b/c_w = 1.5$ (where u_b is the average velocity across the channel thickness, and c_w the sound speed at the wall temperature), and the bulk Reynolds number $Re_b = 2\rho_b u_b h/\mu_w = 6000$ (where ρ_b is the bulk density, μ_w the dynamic viscosity at the wall, and h the channel half height). All DNS are initialized with a parabolic velocity profile with superposed small perturbations, whereas density and pressure are uniform. Periodic boundary conditions are applied in the streamwise (x) and spanwise (z) coordinate directions, and no-slip, isothermal boundary conditions are applied at the walls. A spatially uniform forcing is applied to the streamwise momentum equation, and dynamically adjusted in time to maintain constant mass flow rate [6]. Favre density-weighted decomposition is applied to separate mean values from fluctuations, namely $\phi = \tilde{\phi} + \phi''$, with $\tilde{\phi} = \overline{\rho\phi}/\bar{\rho}$.

The main flow parameters are listed in Table 2. Three flow cases have been considered, one at $M_b = 0.1$ (denoted as CH01), and two at $M_b = 1.5$ (denoted as CH15a-b), the latter two only differing in the distance of the first grid point from the wall. Reference DNS have been carried out with fully explicit time discretization, at $CFL \approx 1$, which are used as a basis of comparison for the ATI and BW algorithms. In order to understand the effectiveness of the (semi-)implicit algorithms, in Table 2 we report the time step restrictions associated with the three coordinate directions, as estimated from Eqs. (1) and (3), as well as the actual time step used in the DNS, all in wall units. As expected, in all flow cases the time step limitation in the wall-normal direction is the most restrictive. Although larger time steps are allowed on grounds of sole numerical stability, all DNS have been carried out at the maximum time step for which accurate results are obtained, which corresponds to $CFL \approx 1$ for the fully explicit simulations. For ease of reference, the maximum time steps associated with accuracy and stability restrictions are also reported in Fig. 1a with circle and square symbols, respectively.

As a first test, we consider flow at low subsonic Mach number (CH01), for which the explicit time advancement step is very small, hence we apply implicit treatment in all coordinate directions (XYZ). We find that, although the wall-normal time step restrictions can be removed, the allowed time step for accurate calculations cannot be substantially larger than the streamwise convective restriction (see Fig. 1a). This is probably due to inherent mesh anisotropy in DNS of wall-bounded flows. In fact, mesh spacing is over-resolved in the wall-normal direction, hence the relevant values of the reduced wavenumber kh are small, which allows to operate at high values of CFL with little error, recalling (see Fig. 2) that the dissipation error grows with both kh and CFL. On the other hand, the typical wall-parallel mesh spacings used in DNS are barely sufficient to resolve the smallest scales of turbulence, hence the typical reduced wavenumbers are higher, and time accuracy is a factor in that case. We find that both ATI and BW are capable of boosting the time step by about a factor of ten, with efficiency gain of 85% for ATI, and results almost indistinguishable from the fully explicit case (see below). Still, the time step is far from that allowed by incompressible solvers (again, see Fig. 1a). This issue will be further recalled in the concluding discussion.

To show effectiveness in removing the wall-normal acoustic time limitation in supersonic flow calculations, in flow case CH15a the first grid point is placed sufficiently far from the wall that the viscous limitation is ineffective. Hence, the implicit algorithms are applied only in the wall-normal direction (Y), and viscous terms are handled explicitly. The ATI and BW algorithms are both found to effectively suppress the wall-normal acoustic time step limitation, and achieve the same maximum time step for accurate flow resolution, corresponding to

Table 2 Flow parameters for DNS of plane channel flow (CH)

Case	M_b	M_0	Re_b	Re_τ	Δy_w^+	Δx^+	Δz^+	Δt_x^+	Δt_y^+	Δt_z^+	$\Delta t_{y_w}^+$	Δt^+	CPU
CH01-EXPL	0.1	0.1	5790	180	0.60	8.80	3.90	0.053	0.0077	0.026	1.3	0.0077	1
CH01-ATI-XYZ	0.1	0.1	5790	180	0.60	8.80	3.90	0.053	0.0077	0.026	1.3	0.077	0.15
CH01-BW-XYZ	0.1	0.1	5790	180	0.60	8.80	3.90	0.053	0.0077	0.026	1.3	0.077	0.82
CH15a-EXPL	1.5	1.28	6000	220	0.70	10.8	4.80	0.32	0.11	0.27	1.2	0.099	1
CH15a-ATI-Y	1.5	1.28	6000	220	0.70	10.8	4.80	0.32	0.11	0.27	1.2	0.24	0.48
CH15a-BW-Y	1.5	1.28	6000	220	0.70	10.8	4.80	0.32	0.11	0.27	1.2	0.24	0.70
CH15b-EXPL	1.5	1.28	6000	220	0.15	10.8	4.08	0.32	0.11	0.27	0.062	0.021	1
CH15b-AVTI-Y	1.5	1.28	6000	220	0.15	10.8	4.80	0.32	0.11	0.27	0.062	0.21	0.13
CH15b-BWV-Y	1.5	1.28	6000	220	0.15	10.8	4.80	0.32	0.11	0.27	0.062	0.21	0.19

M_b and Re_b are the bulk Reynolds and Mach number, respectively, $M_0 = M_b \sqrt{T_w/T_b}$ is the reference Mach number, introduced when discussing Eq. (1). The computational box dimension is $4\pi h \times 2h \times 4/3\pi$ for all flow cases. Δy_w^+ is the distance of the first grid point from the wall, and Δx^+ , Δz^+ are the streamwise and spanwise grid spacings. The Δt_i^+ are the allowable time steps in the coordinate directions, estimated according to Eqs. (1) and (3). Δt^+ is the time step actually used in the simulations. CPU is the cost to cover a unit time interval, compared to the standard fully explicit algorithm (EXPL)

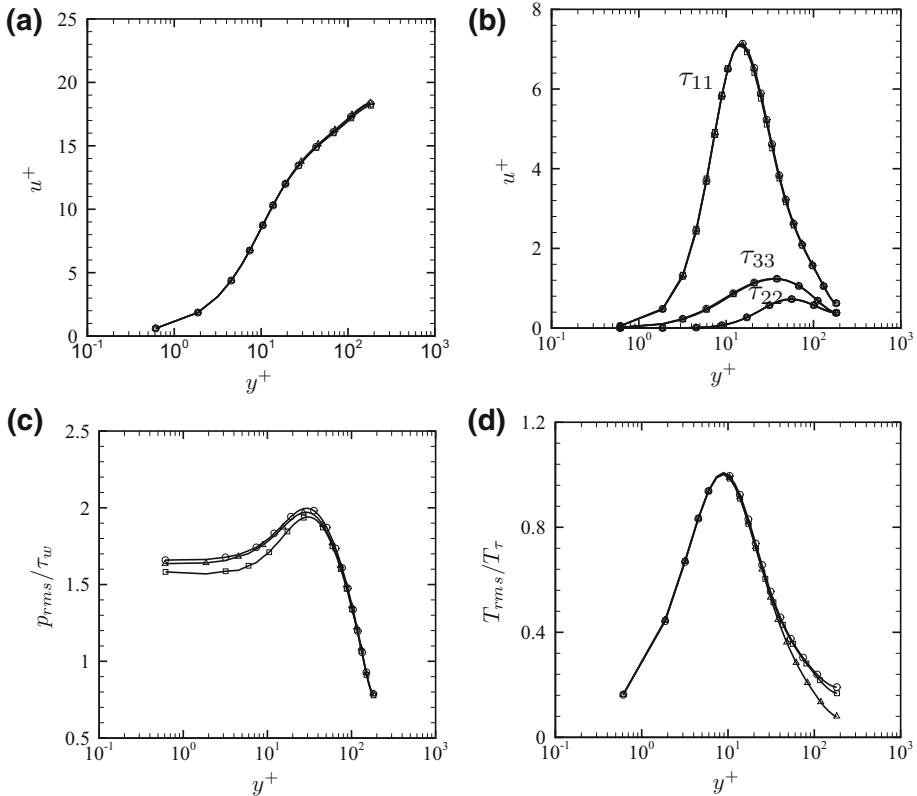


Fig. 5 Flow statistics for DNS of flow case CH01 (see Table 2): mean velocity (a), Reynolds stresses (b), r.m.s. pressure (c) and r.m.s. temperature (d), for CH01-EXPL (squares), CH01-ATI-XYZ (circles), CH01-BW-XYZ (triangles). $T_\tau = q_w/(\rho_w c_p u_\tau)$ is the friction temperature

about CFL = 2.4. Hence, accounting for the cost figures given in Table 1, we find a speed-up of about a factor of two for the ATI algorithm, and 30% gain with BW.

To prove effectiveness of the implicit treatment of the viscous terms proposed in Sect. 2.2, in flow case CH15b the first grid point is placed closer to the wall, in such a way that the viscous time limitation also becomes relevant, after the acoustic one. Both wall-normal time step restrictions are suppressed through use of the AVTI and BWV algorithms, hence the achieved time step is similar to flow case CH15a. Both algorithms here achieve CFL ≈ 10 , at a cost which is a small fraction of the fully explicit algorithm.

For the sake of comparison, in Figs. 5, 6 and 7 we show the main statistics for the flow cases listed in Table 2. As anticipated, excellent agreement is observed between implicit algorithms and the reference explicit solution, including pressure and temperature fluctuations, which is especially satisfactory.

3.3 Turbulent Flow in Square Duct

As a further step in complexity we consider the flow inside a straight duct with square cross-section. This flow has been the subject of several DNS studies in the incompressible regime [52–54], all limited to low Reynolds number. One of the main difficulties that arise

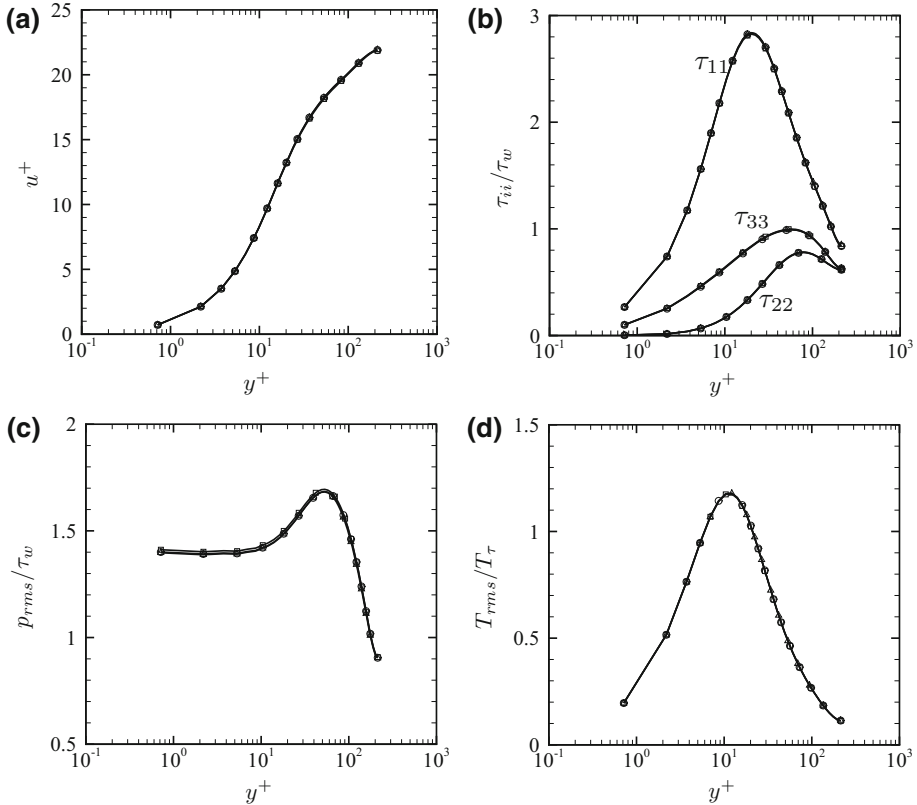


Fig. 6 Flow statistics for DNS of flow case CH15a (see Table 2): mean velocity (a), Reynolds stresses (b), r.m.s. pressure (c) and r.m.s. temperature (d), for CH15a-EXPL (*squares*), CH15a-ATI-Y (*circles*), CH15a-BW-Y (*triangles*)

when dealing with square duct flows is the long averaging time necessary to attain convergence of even the basic mean flow statistics, caused by the extremely long typical time scales of secondary corner eddies. In fact, [54] reported that an averaging time of about $8000h/u_b$ was needed to have symmetric statistics in the four quadrants of the cross section. Hence, it is clear that efficient numerical methods are needed to study turbulent compressible flow in ducts. Numerical simulations have been here carried out (see Table 3 for the main flow parameters) at the same Reynolds number as [54], and sufficiently low Mach number ($M_b = 0.2$) that direct comparison with incompressible data is possible. The duct length $L_x = 8h$ (where $2h$ is the length of each side of the duct), and the time window for collecting the flow statistics is the same used by [54]. As in plane channel flow, a spatially uniform forcing is applied to the momentum equation to maintain a time constant mass flow rate. Note that, unlike in channel flow, the mesh is also non-uniformly spaced in the z direction, hence a range of mesh spacings is reported in Table 3. A reference fully explicit numerical simulation has been carried out and used as a basis of reference for the ATI algorithm, here applied to all coordinate directions. As seen in Table 3, the corresponding CFL number is about unity. As in the case of plane channel, DNS were carried out at increasing values of CFL, until deviations from the reference data were found, to determine the maximum allowed time step

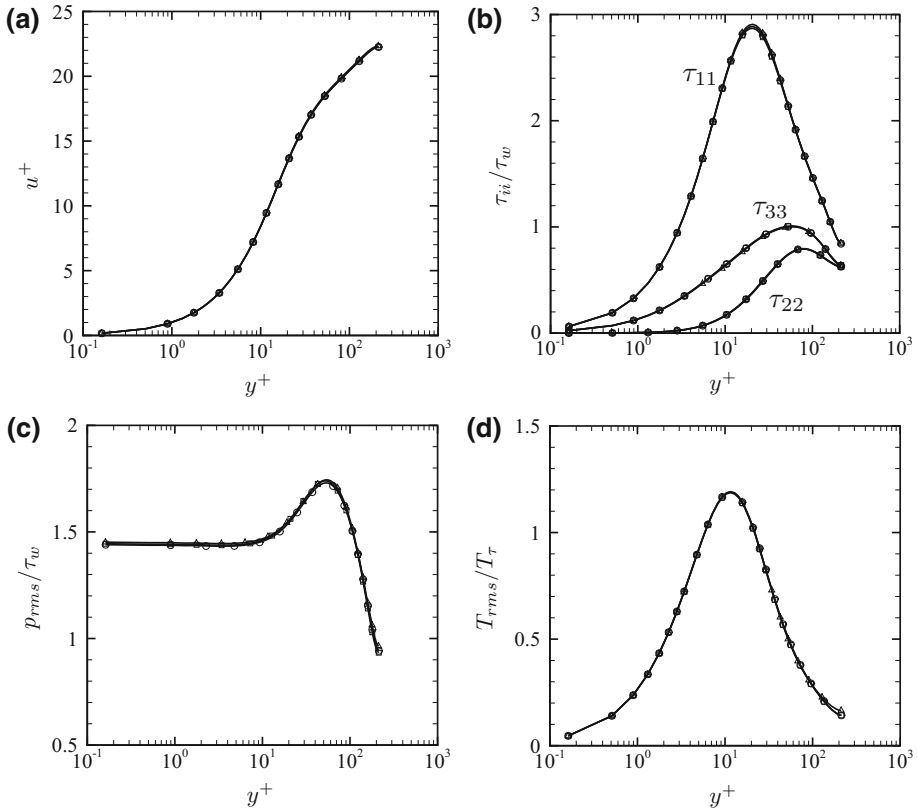


Fig. 7 Flow statistics for DNS of flow case CH15b (see Table 2): mean velocity (a), Reynolds stresses (b), r.m.s. pressure (c) and r.m.s. temperature (d), for CH15b-EXPL (squares), CH15b-AVTI-Y (circles), CH15b-BWV-Y (triangles)

for accuracy. It appears that accurate results of the semi-implicit algorithm are recovered up to $CFL \approx 10$. Again, implicit treatment of the x direction is not capable of fully suppressing the corresponding time step limitation, owing to the emergence of accuracy issues. Similar to channel flow, use of the ATI algorithm allows for about 85% cost reduction. Figure 8 confirms that excellent matching of the flow statistics is found among DU02-ATI, DU02-EXPL and the data of [54], except for some differences in the wall-normal Reynolds stress and the pressure r.m.s., which may be due to the greater importance of acoustic waves in the presence of a fully confined flow geometry.

4 Conclusions

A novel semi-implicit algorithm for time-accurate solution of the compressible Navier–Stokes equations has been developed, which is capable to operate efficiently all the way from low subsonic to supersonic flow conditions. The main features of the algorithm are as follows: (i) use of the entropy transport equation instead of total energy conservation; (ii) Beam–Warming-like linearization of the partial convective flux associated with acoustic

Table 3 DNS dataset for square duct (DU) flow

Case	M_b	M_0	Re_b	Re_τ	Δy_w^+	Δx^+	Δz^+	Δx^+	Δy^+	Δz^+	Δy_w^+	Δt^+	CPU
DU02-EXPL	0.2	0.2	4410	150	0.66	8.40	0.66–3.20	0.094	0.019	0.019	1.69	0.018	1
DU02-ATI-XYZ	0.2	0.2	4410	150	0.66	8.40	0.66–3.20	0.094	0.019	0.019	1.69	0.18	0.15

M_b and Re_b are the bulk Reynolds and Mach number, respectively. $M_0 = M_b \sqrt{T_w/T_b}$ is the reference Mach number, introduced when discussing Eq. (1). The computational box dimension is $8\pi h \times 2h \times 2h$. Δy_w^+ is the distance of the first grid point from the wall, and $\Delta x^+, \Delta z^+$ are the streamwise and spanwise grid spacings. The Δt^+ are the allowable time steps in the coordinate directions, estimated according to Eqs. (1) and (3). Δt^+ is the time step actually used in the simulations. CPU is the cost to cover a unit time interval, compared to the standard fully explicit algorithm (EXPL)

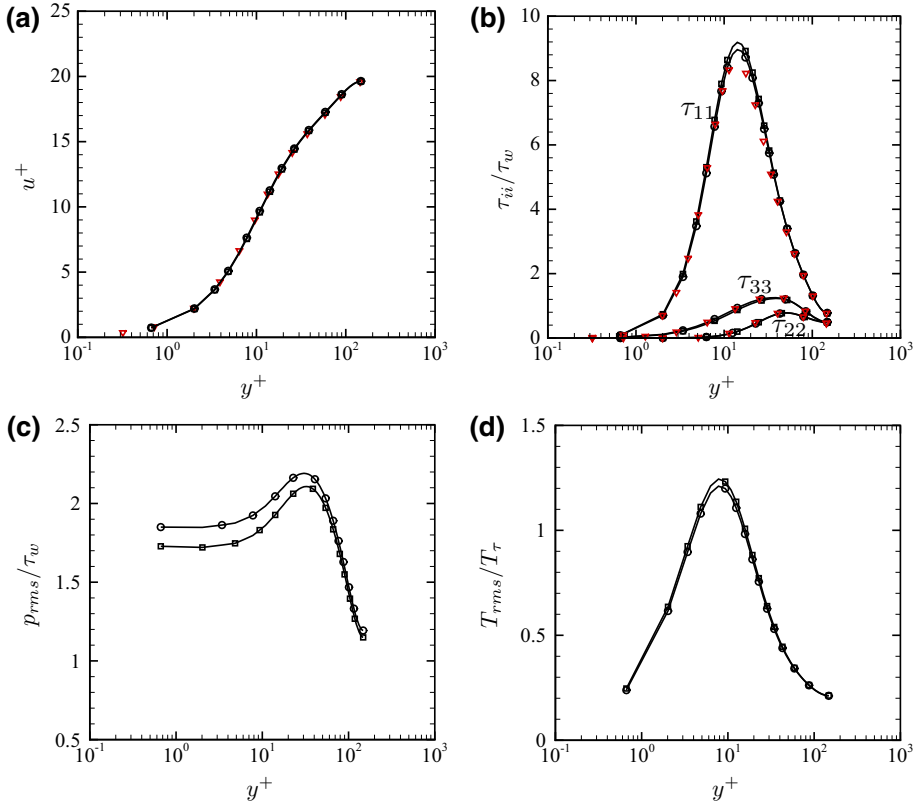


Fig. 8 DNS of flow in square duct (see Table 3): mean velocity (a), Reynolds stresses (b), r.m.s. pressure (c) and r.m.s. temperature (d), for DU02-EXPL (squares), DU02-ATI-XYZ (circles). Triangle symbols denote reference incompressible DNS data [54]

propagation; (iii) energy-consistent discretization of the convective derivatives in the explicit part of the time-advancement operator; (iv) semi-implicit treatment of viscous fluxes based on isolation of Laplacian terms; (v) approximate factorization for implicit treatment of multiple space directions; (vi) third-order accurate Runge–Kutta time integration, according to the algorithm proposed by [1]. The main advantage of the algorithm is that, unlike the classical Beam–Warming scheme, it avoids the computationally expensive inversion of 5×5 block-banded matrices, but rather of standard banded matrices (tridiagonal matrices in the case of second-order accurate space discretization). Specifically, a single banded matrix inversion is needed for implicit treatment of the convective terms, whereas five matrix inversions are needed if viscous terms are also handled implicitly. The cost overhead with respect to standard explicit algorithms (see Table 1) is quite modest, ranging from 20 to 30%, for each space direction to be handled implicitly. Modification of existing compressible flow solvers to incorporate the present method is straightforward, as the explicit part of the algorithm is unchanged.

The method nominally allows unconditional stability for low-Mach-number flows. However, flux linearization and approximate factorization reduce the stability margins, and CFL number of the order of 5–10 are achieved in practical computations, which is probably less than achievable with iterative methods. However, compared with compressible flow algo-

rithms based on pre-conditioning, the present method avoids the use of inner time iterations, whose computational cost is difficult to estimate a-priori. The other shortcoming of the method is the use of the entropy equation, which is instrumental to achieve (approximate) separation of hydrodynamic from acoustic effects, but which precludes correct capture of shock waves, as the equation is not in conservation form. Efforts are in progress to achieve shock-capturing capabilities through local replacement of the entropy equation with the total energy conservation equation. Although the algorithm herein developed has in principle much wider range of applications, the main focus of this paper was on DNS of compressible wall-bounded flows, which is notoriously plagued by severe time step restrictions inherited from the wall-normal acoustic and viscous stability conditions. We have found that the wall-normal acoustic time limitation can be effectively removed through semi-implicit treatment. The same conclusion also applies to the viscous time step restriction, although the most efficient way to remove it is placing the first grid point sufficiently away from the wall $y^+ \approx 0.5 - 0.7$, and using suitable staggering [6], with no effect on accuracy. The wall-parallel stability restrictions can also be suppressed through semi-implicit treatment. However, accuracy considerations lead to the practical rule (see Fig. 1) that the time step cannot be much larger than the one stemming from the streamwise time limitation. Hence, we suggest that in low-subsonic flow both the wall-normal and the spanwise convective terms are handled implicitly, whereas the streamwise terms can be evaluated explicitly. The resulting saving of computer time can then be of the order of 85% with respect to a fully explicit solver. In high subsonic or supersonic flow, implicit treatment of the wall-normal convective derivatives is sufficient, with typical savings of to order of 50%, in line with theoretical estimates.

We foresee that the present technique can be fruitfully extended to numerical simulation of wall-bounded turbulent flows with time-accurate models, such as LES or DES [55]. In that case, given the higher aspect ratio of near-wall cells, higher gains are expected. Advantages with respect to classical algorithms based on Beam–Warming linearization are also expected for steady RANS applications. Indeed, although the present algorithm is in principle only capable of suppressing the acoustic time step limitation, it is found to be at least as stable as Beam–Warming in practical computations.

Acknowledgements We acknowledge that most of the results reported in this paper have been achieved using the PRACE Research Infrastructure resource FERMI based at CINECA, Casalecchio di Reno, Italy.

References

1. Nikitin, N.: Third-order-accurate semi-implicit Runge–Kutta scheme for incompressible Navier–Stokes equations. *Int. J. Numer. Methods Fluids* **51**, 221–233 (2006)
2. Lee, M., Moser, R.: Direct simulation of turbulent channel flow layer up to $Re_\tau = 5200$. *J. Fluid Mech.* **774**, 395–415 (2015)
3. Lele, S.: Compressibility effects on turbulence. *Annu. Rev. Fluid Mech.* **26**, 211–254 (1994)
4. Poinso, T., Trounev, A., Veynante, D., Candel, S., Esposito, E.: Vortex-driven acoustically coupled combustion instabilities. *J. Fluid Mech.* **177**, 265–292 (1987)
5. Colonius, T., Lele, S.K.: Computational aeroacoustics: progress on nonlinear problems of sound generation. *Prog. Aerosp. Sci.* **40**, 345–416 (2004)
6. Modesti, D., Pirozzoli, S.: Reynolds and Mach number effects in compressible turbulent channel flow. *Int. J. Heat Fluid Flow* **59**, 33–49 (2016)
7. Beam, R., Warming, R.: An implicit finite-difference algorithm for hyperbolic systems in conservation-law form. *J. Comput. Phys.* **22**, 87–110 (1976)
8. Beam, R., Warming, R.: An implicit factored scheme for the compressible Navier–Stokes equations. *AIAA J.* **16**, 393–402 (1978)

9. Douglas, J.: On the numerical integration of $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{\partial u}{\partial t}$ by implicit methods. *J. Soc. Ind. Appl. Math.* **3**, 42–65 (1955)
10. Isaacson, E., Keller, H.: *Analysis of Numerical Methods*. Courier Corporation, Mineola (1994)
11. Batista, M.: A cyclic block-tridiagonal solver. *Adv. Eng. Softw.* **37**, 69–74 (2006)
12. Pulliam, T., Chaussee, D.: A diagonal form of an implicit approximate-factorization algorithm. *J. Comput. Phys.* **39**, 347–363 (1981)
13. Hirsch, C.: *Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics*. Butterworth-Heinemann, Oxford (2007)
14. Pulliam, T.: *Solution methods in computational fluid dynamics*. In: von Karman Institute for Fluid Mechanics Lecture Series, Numerical Techniques for Viscous Flow Computations in Turbomachinery (1986)
15. Buning, P., Jespersen, D., Thomas, H., Chan, W., Slotnick, J., Krist, S., Renze, K.: *OVERFLOW User's Manual version 1.8f*, Unpublished NASA Report (1998)
16. Cinnella, C.C.P.: High-order implicit residual smoothing time scheme for direct and large eddy simulations of compressible flows. *J. Comput. Phys.* **326**, 1–29 (2016)
17. Martín, M., Candler, G.: A parallel implicit method for the direct numerical simulation of wall-bounded compressible turbulence. *J. Comput. Phys.* **215**, 153–171 (2006)
18. Turkel, E.: Review of preconditioning methods for fluid dynamics. *Appl. Numer. Math.* **12**, 257–284 (1993)
19. Turkel, E.: Preconditioning techniques in computational fluid dynamics. *Annu. Rev. Fluid Mech.* **31**, 385–416 (1999)
20. Venkateswaran, S., Merkle, C.: Dual time stepping and preconditioning for unsteady computations. *AIAA Paper 1995-0078* (1995)
21. Pandya, S., Venkateswaran, S., Pulliam, T.: Implementation of preconditioned dual-time procedures in overflow. *AIAA paper 2003-0072* (2003)
22. Palma, P.D., Tullio, M.D., Pascasio, G., Napolitano, M.: An immersed-boundary method for compressible viscous flows. *Comput. Fluids* **35**, 693–702 (2006)
23. Jin, S.: Efficient asymptotic-preserving (ap) schemes for some multiscale kinetic equations. *SIAM J. Sci. Comp.* **21**(2), 441–454 (1999)
24. Degond, P., Tang, M.: All speed scheme for the low mach number limit of the isentropic Euler equations. *Commun. Comput. Phys.* **10**(1), 1–31 (2011)
25. Haack, J., Jin, S., Liu, J.: An all-speed asymptotic-preserving method for the isentropic Euler and Navier–Stokes equations. *Commun. Comput. Phys.* **12**(04), 955–980 (2012)
26. Casulli, V., Greenspan, D.: Pressure method for the numerical solution of transient, compressible fluid flows. *Int. J. Numer. Methods Fluids* **4**, 1001–1012 (1984)
27. Pierce, C.: *Progress-variable approach for large-eddy simulation of turbulent combustion*. Ph.D. thesis, Citeseer (2001)
28. Wall, C., Pierce, C., Moin, P.: A semi-implicit method for resolution of acoustic waves in low Mach number flows. *J. Comput. Phys.* **181**, 545–563 (2002)
29. Kim, J., Moin, P.: Application of a fractional-step method to incompressible Navier–Stokes equations. *J. Comput. Phys.* **59**, 308–323 (1985)
30. Moureau, V., Bérat, C., Pitsch, H.: An efficient semi-implicit compressible solver for large-eddy simulations. *J. Comput. Phys.* **226**, 1256–1270 (2007)
31. Spalart, P., Moser, R., Rogers, M.: Spectral methods for the Navier–Stokes equations with one infinite and two periodic directions. *J. Comput. Phys.* **96**, 297–324 (1991)
32. Sesterhenn, J.: A characteristic-type formulation of the Navier–Stokes equations for high order upwind schemes. *Comput. Fluids* **30**, 37–67 (2000)
33. Honein, A., Moin, P.: Higher entropy conservation and numerical stability of compressible turbulence simulations. *J. Comput. Phys.* **201**, 531–545 (2004)
34. Salas, M., Iollo, A.: Entropy jump across an inviscid shock wave. *Theor. Comput. Fluid Dyn.* **8**, 365–375 (1996)
35. Kovásznay, L.: Turbulence in supersonic flow. *J. Aeronaut. Sci.* **20**, 657–674 (1953)
36. Steger, J.: Coefficient matrices for implicit finite difference solution of the inviscid fluid conservation law equations. *Comput. Methods Appl. Mech. Eng.* **13**, 175–188 (1978)
37. Barth, T., Steger, J.: A fast efficient implicit scheme for the gasdynamics equations using a matrix reduction technique. *AIAA paper 1985-0085* (1985)
38. Orlandi, P.: *Fluid Flow Phenomena: A Numerical Toolkit*. Kluwer, Dordrecht (2000)
39. Vichnevetsky, R., Bowles, J.B.: *Fourier Analysis of Numerical Approximations of Hyperbolic Equations*. SIAM, Philadelphia (1982)
40. Pirozzoli, S.: Generalized conservative approximations of split convective derivative operators. *J. Comput. Phys.* **229**, 7180–7190 (2010)

41. Kennedy, C., Gruber, A.: Reduced aliasing formulations of the convective terms within the Navier–Stokes equations for a compressible fluid. *J. Comput. Phys.* **227**, 1676–1700 (2008)
42. Pirozzoli, S.: Numerical methods for high-speed flows. *Annu. Rev. Fluid Mech.* **43**, 163–194 (2011)
43. Lele, S.K.: Compact finite difference schemes with spectral-like resolution. *J. Comput. Phys.* **103**, 16 (1992)
44. Verzicco, R., Orlandi, P.: A finite-difference scheme for three-dimensional incompressible flows in cylindrical coordinates. *J. Comput. Phys.* **123**(2), 402–414 (1996)
45. Bernardini, M., Pirozzoli, S., Orlandi, P.: Velocity statistics in turbulent channel flow up to $Re_\tau = 4000$. *J. Fluid Mech.* **742**, 171–191 (2014)
46. Moin, P., Verzicco, R.: On the suitability of second-order accurate discretizations for turbulent flow simulations. *Eur. J. Mech. B/Fluids* **55**, 242–245 (2016)
47. Shoeybi, M., Svård, M., Ham, F., Moin, P.: An adaptive implicit–explicit scheme for the DNS and LES of compressible flows on unstructured grids. *J. Comput. Phys.* **229**(17), 5944–5965 (2010)
48. Blaisdell, G., Mansour, N., Reynolds, W.: Numerical simulation of compressible homogeneous turbulence. Report TF-50, Thermosciences Division, Department of Mechanical Engineering, Stanford University (1991)
49. Kim, J., Moin, P., Moser, R.: Turbulence statistics in fully developed channel flow at low Reynolds number. *J. Fluid Mech.* **177**, 133–166 (1987)
50. Coleman, G., Kim, J., Moser, R.: A numerical study of turbulent supersonic isothermal-wall channel flow. *J. Fluid Mech.* **305**, 159–183 (1995)
51. Lechner, R., Sesterhenn, J., Friedrich, R.: Turbulent supersonic channel flow. *J. Turbul.* **2**, 1–25 (2001)
52. Gavrilakis, S.: Numerical simulation of low-Reynolds-number turbulent flow through a straight square duct. *J. Fluid Mech.* **244**, 101–129 (1992)
53. Huser, A., Biringen, S.: Direct numerical simulation of turbulent flow in a square duct. *J. Fluid Mech.* **257**, 65–95 (1993)
54. Pinelli, A., Uhlmann, M., Sekimoto, A., Kawahara, G.: Reynolds number dependence of mean flow structure in square duct turbulence. *J. Fluid Mech.* **644**, 107–122 (2010)
55. Spalart, P.: Strategies for turbulence modelling and simulations. *Int. J. Heat Fluid Flow* **21**, 252–263 (2000)