CUBENS

A GPU-accelerated high-order solver for wall-bounded flows with non-ideal fluids

Boldini, Pietro Carlo; Hirai, Ryo; Costa, Pedro; Peeters, Jurriaan W.R.; Pecnik, Rene

DOI
10.1016/j.cpc.2025.109507

Publication date
2025

Document Version
Final published version

Published in
Computer Physics Communications

Citation (APA)
Boldini, P. C., Hirai, R., Costa, P., Peeters, J. W. R., & Pecnik, R. (2025). CUBENS: A GPU-accelerated high-order solver for wall-bounded flows with non-ideal fluids. *Computer Physics Communications*, *309*, Article 109507. https://doi.org/10.1016/j.cpc.2025.109507

Computer Physics Communications 309 (2025) 109507

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Computer Programs in Physics

CUBENS: A GPU-accelerated high-order solver for wall-bounded flows with non-ideal fluids ☆,☆☆

Pietro Carlo Boldini *, Ryo Hirai, Pedro Costa, Jurriaan W.R. Peeters, Rene Pecnik *

Process and Energy Department, Delft University of Technology, Leeghwaterstraat 39, Delft, 2628 CB, the Netherlands

ARTICLE INFO

Keywords:
Non-ideal compressible fluid dynamics
Direct numerical simulation
Non-ideal thermodynamics
High-order numerics
Wall-bounded flows
GPUs
OpenACC
Open source

ABSTRACT

We present a massively parallel GPU-accelerated solver for direct numerical simulations of transitional and turbulent flat-plate boundary layers and channel flows involving fluids in non-ideal thermodynamic states. While several high-fidelity solvers are currently available as open source, all of them are restricted to the ideal-gas region. In contrast, the CUBic Equation of state Navier-Stokes solver (CUBENS) can accurately model and simulate the non-ideal thermodynamics of single-phase compressible fluids in the vicinity of the vapor-liquid saturation line or the thermodynamic critical point. By employing high-order finite-difference schemes and convective terms in split, kinetic-energy-, and entropy-preserving form, the solver is numerically stable, and robust with minimal numerical dissipation, enabling it to capture the steep variations of non-ideal thermodynamic properties. For cost-effective high-fidelity simulations, in addition to MPI parallelization, CUBENS is GPU-accelerated using OpenACC directives for computation offloading, and asynchronous GPU-aware MPI for efficient GPU-GPU communication. Moreover, CUBENS is compatible with both NVIDIA and AMD GPU architectures, achieving significant performance results while ensuring energy-efficient simulations. For instance, using 64 NVIDIA A100 GPUs compared to 8192 CPUs at the same computational cost results in a speedup of approximately 130×. In multi-node and multi-GPU configurations ranging from 2 to 128 compute nodes (8 to 512 GPUs), a strong scaling efficiency of around 52% and a weak scaling efficiency of 0.88 with 1024^3 points per GPU, corresponding to approximately 5 billion degrees of freedom, are achieved. The CUBENS solver is validated against selected cases from the literature, covering transitional to turbulent ideal and non-ideal flows up to the transonic regime. In particular, we demonstrate the solver's suitability and applicability for direct numerical simulations of transitional boundary layers with fluids at supercritical pressure and with buoyancy effects. The development of this high-fidelity solver offers the potential for future fundamental research in non-ideal compressible fluid dynamics.

Program summary

Program Title: CUBic Equation of state Navier-Stokes (CUBENS)
CPC Library link to program files: https://doi.org/10.17632/6jfy758gyv.1
Developer's repository link: https://github.com/pcboldini/CUBENS
Licensing provisions: MIT
Programming language: Fortran 90, OpenACC, MPI, Python, MATLAB
Nature of problem: This code solves the three-dimensional Navier-Stokes equations for non-ideal gas flows in a Cartesian domain, applicable to boundary layers and channels.
Solution method: This code uses high-order central finite-differences with split-convective form, preserving kinetic energy and entropy (KEEP) and pressure-equilibrium-preserving (PEP) property, for spatial discretization. The time advancement is performed with a third-order Total Variation Diminishing low-storage Runge-Kutta scheme. Flow non-ideality is accounted for by cubic equations of state and complex transport-properties models. Alongside MPI parallelization, the solver is GPU-accelerated using OpenACC for computation offloading and CPU-GPU data transfer, along with GPU-aware MPI for GPU-GPU communication.

☆ Source code open and available under the terms of a MIT License on https://github.com/pcboldini/CUBENS.
☆☆ The review of this paper was arranged by Prof. Peter Vincent.
* Corresponding authors.
E-mail addresses: p.c.boldini@tudelft.nl (P.C. Boldini), r.pecnik@tudelft.nl (R. Pecnik).

https://doi.org/10.1016/j.cpc.2025.109507
Received 15 July 2024; Received in revised form 4 November 2024; Accepted 14 January 2025

Available online 17 January 2025

1. Introduction

The application of single-phase fluids in non-ideal thermodynamic states has rapidly expanded the relevance of a new branch of fluid mechanics called non-ideal compressible fluid dynamics [1]. Given the increasing number of industrial applications operating under non-ideal gas conditions, such as turbomachinery and heat exchangers [2], the development of more accurate theoretical, experimental, and numerical tools is essential [3]. A major challenge, however, is the lack of knowledge regarding transitional and turbulent boundary layers due to the difficulty in conducting experiments at high density and temperature conditions [4]. In contrast, high-fidelity simulations can significantly enhance and accelerate the design of new engineering systems that operate under non-ideal gas conditions. Fluids above their vapor-liquid critical point, i.e., in the supercritical-fluid region, play a key role in future energy conversion systems, see e.g., Zhao et al. [5].

In this region, if a high-pressure liquid-like or vapor-like fluid is isobarically heated or cooled across the pseudo-boiling or Widom line, defined as $\max\{c_p(T)\}$, a continuous phase transition with significant variations in thermophysical properties occurs. This transition can lead to either heat-transfer deterioration or enhancement [6], and may delay or promote the onset of turbulence [7,8]. Due to the non-ideal fluid behavior close to the thermodynamic vapor-liquid critical point, the ideal-gas equation of state $p = \rho R_g T$ (with p as pressure, ρ as density, R_g as specific gas constant, and T as temperature) is not applicable, requiring more complex equations of state. It is important to clarify that the term "complex" does not refer to the complexity of fluid molecules itself, e.g., fluids with high molecular complexity (HMC). Instead, non-ideal gas dynamic phenomena affect the flow of all fluids whose thermodynamic states fall within the non-ideal thermodynamic region, including those with simpler molecular structures, such as low molecular complexity (LMC) fluids [3].

The first high-fidelity simulations of non-ideal fluid flows have been conducted for only a limited number of specific configurations. For example, Kawai [9] investigated turbulent boundary layers with the LMC fluid parahydrogen at supercritical pressure and transcritical temperatures. Due to the abrupt density variation across the Widom lime, large density fluctuations were observed, which significantly altered the near-wall turbulence. In the case of dense vapor of a HMC fluid, Sciacovelli et al. [10] performed the first direct numerical simulation (DNS) of a supersonic turbulent channel configuration. The high isobaric heat capacity of HMC fluids results in negligible temperature variations, despite observing pressure fluctuations similar to those in ideal-gas cases. Conversely, supersonic flows of HMC fluid are found to exhibit both a "gas-like" (e.g., speed of sound) and "liquid-like" (e.g., viscosity) behavior [10]. With the semi-local scaling (see Patel et al. [11]), turbulence statistics for variable-property flows show remarkable agreement with incompressible-gas cases. In Patel et al. [12], Pecnik and Patel [13], turbulence modulation by non-ideal viscosity laws was investigated. Once again, the semi-local Reynolds number proved to quantify best turbulence in variable-property flows. Ma et al. [14] found that while semi-local scaling worked well for the bottom cooled wall, it was less suitable for the top-heated wall, where density fluctuations were significant. Following Patel et al. [12], an improved near-wall mean temperature formulation was proposed by Guo et al. [15] to better collapse the slope of the logarithmic region. Studies by Nemati et al. [16], Peeters et al. [17], Kim et al. [18] on turbulent flows with LMC fluids at supercritical pressure revealed that the thermal wall-boundary condition greatly influences heat transfer in supercritical fluids, and that property fluctuations considerably impact turbulent streak evolution. He et al. [19] and Cao et al. [20] demonstrated that supercritical CO_2 (LMC fluid) turbulent pipe flows are prone to laminarization, primarily due to the alternating dominance of buoyancy and inertia mechanisms.

Only in recent times have selected transition routes to turbulence in wall-bounded flows with non-ideal fluids have been explored. The first DNS of a spatially developing boundary layer with HMC fluids was conducted by Sciacovelli et al. [21] for PP11 and by Gloerfelt et al. [22] for Novec649, two HMC fluids. Dense-vapor boundary layers revealed large fluctuations in Mach number and dilatation compared to light-gas boundary layers at high Mach numbers. Additionally, by means of DNS, Boldini et al. [23] investigated the first controlled laminar-to-turbulent transition of a transcritical boundary layer with an LMC fluid at supercritical pressure. The H-type breakdown unveiled a combination of strong vortical structures, resembling Λ-vortices, as well as high- and low-speed and density streaks. Overall, the transition to turbulence was found to be much more abrupt than in the case at supercritical pressure with subcritical temperature.

All the aforementioned studies can be classified as high-fidelity simulations, where all temporal and spatial flow scales are fully resolved. A key requirement for conducting these simulations is the minimization of dispersion and dissipation errors in the calculation of convective terms [24]. Numerical dispersion at large wave numbers can be reduced by employing high-order finite-difference schemes [25], which, in turn, enhance spatial resolution [4]. To minimize numerical dissipation, it is essential to preserve not only the primary variables (mass, momentum, total energy) but also the secondary ones (energy and entropy). This approach also improves the numerical stability of the proposed scheme [26].

A first attempt to conserve kinetic energy was proposed by Kennedy and Gruber [27] using a cubic-split form of the convective terms. This approach, later rewritten by Pirozzoli [28] for computational efficiency, is now commonly known as the Kennedy-Gruber-Pirozzoli (KGP) scheme. An improvement of this kinetic-energy preserving (KEP) scheme was made by Kuya et al. [24] and Kuya and Kawai [29], who developed a high-order accurate kinetic energy and entropy preserving (KEEP) scheme that provides improved numerical robustness compared to the existing KEP scheme. Importantly, while the KEEP scheme enhances the entropy preservation of the KEP scheme, resulting in greater numerical robustness, it does not ensure a conservative calculation of the entropy convection term, meaning that entropy is not conserved to machine precision [30]. In a new study, Aiello et al. [31] developed a spatial discretization to achieve exact entropy-conservative numerical fluxes for an arbitrary equation of state.

Nonetheless, these classical preserving schemes are unable to maintain the pressure-equilibrium-preserving (PEP) property, which refers to the discrete scheme's ability to reproduce a traveling density wave with an initially constant velocity and pressure distribution. To suppress spurious pressure oscillations, several strategies have emerged. De Michele and Coppola [32] proposed asymptotically entropy conservative (AEC) schemes under the ideal-gas assumption, based on the harmonic mean for internal energy, which exhibit both PEP and KEP properties. In the KEEP-PEP version presented by Shima et al. [33], a variation of the split-form discretization for the internal-energy convective term (logarithmic mean) was derived, albeit under the assumption of an ideal gas. In fact, due to the linear relationship between pressure and internal energy in this case, the PEP condition was ensured. Yet, when non-ideal gas dynamic phenomena affect the flow, the strongly non-linear dependencies of the equation of state affect the relationship between pressure and internal energy, making the formulation of the internal-energy flux non-trivial. Although the ideal-gas PEP condition of [33] provides a first-order approximation of pressure equilibrium for any equation of state, spurious, non-physical pressure and velocity oscillations can occur in abrupt flow variations (e.g., under pseudo-boiling conditions close to the vapor-liquid critical point) [34,35]. These oscillations persist as grid-to-grid variations, even with mesh refinement or additional filtering, potentially leading to numerical instabilities [9]. A first attempt to specifically address this issue was made by Bernades et al. [36] with the KGP-Pt scheme, where the term "Pt" refers to the time derivative of pressure. In this approach, the pressure evolution equation is discretized directly instead of the total energy equation, resulting in better preservation of pressure equilibrium; however, this comes at the cost of conserving total energy and entropy. Despite these advancements, no numerical

method currently exists that is capable of discretely preserving both primary and secondary variables while simultaneously enforcing pressure equilibrium for non-ideal fluids. Consequently, performing DNS of supercritical flows near the thermodynamic critical point remains highly challenging from a numerical standpoint.

To summarize, the following numerical requirements are essential for high-fidelity simulations of transitional and turbulent wall-bounded flows of compressible single-phase non-ideal fluids: (i) a high-order scheme to achieve greater accuracy, (ii) appropriate modeling of non-linear thermodynamic and transport properties, (iii) accurate resolution of the strong property variations near the Widom line, and (iv) high-spatial resolution to resolve the smallest coherent thermal structures. Note that, in the proximity of the critical point and across the Widom line, the Prandtl number becomes $O(Pr) \gg 1$. This necessitates high-spatial resolution to resolve the smallest scale of the temperature field, such as the Batchelor length scale η_B, defined as $\eta_B = \eta_k/\sqrt{Pr}$, where η_k is the Kolmogorov scale. Consequently, substantial computational resources are necessary to perform stable and accurate scale-resolving simulations.

In the past, high-fidelity solvers relied on central processing unit (CPU) execution and message passing interface (MPI) parallelization. Yet, the advent of graphics processing units (GPUs) has significantly advanced the progress towards Exascale computing [37]. As a result, high-performance-computing (HPC) architectures are progressively adopting heterogeneous nodes that combine CPUs with GPUs as accelerators. For instance, approximately 40% of the current TOP500 [38] list's computing power comes from GPU-accelerated systems, a profound contrast to over a decade ago when such systems were absent from the list. Furthermore, among the most efficient supercomputers in the current GREEN500 list [39], all of the top 30 employ GPU-accelerated energy-efficient computing units. Notably, these systems are powered exclusively by either NVIDIA or AMD GPUs, although Intel has recently entered the competition with the Aurora supercomputer at Argonne National Laboratory, which currently ranks as the second-fastest in the world and is number 42 on the GREEN500 list. This paradigm shift has recently lead to many incompressible and compressible DNS solvers being GPU-accelerated or entirely developed for GPU-based architectures, yielding remarkable speedups of 10 to 20 times compared to CPU-only architectures [40].

Among them, we summarize a selection of open-source direct numerical simulation solvers. Two of the most common academic incompressible Navier-Stokes solvers are AFiD [41] and CaNS [42], which utilize CUDA Fortran and OpenACC GPU-acceleration, respectively. Conversely, in the compressible-flow community, high-order solvers such as STREAmS [43] for the ideal-gas regime and HTR [44] for hypersonic reacting flows employ CUDA Fortran and task-based Legion for GPU-based architectures, respectively. Another example is URANOS [45], which leverages the advantageous OpenACC paradigms for GPU acceleration. As a directive-based method, OpenACC is fully independent of the computing architecture and more user-friendly than CUDA Fortran, requiring minimal code adaptations. Significant performance improvements have recently been achieved for both STREAmS and URANOS, as outlined in Salvadore et al. [46] and De Vanna and Baldan [47], respectively. In the case of STREAmS, a comprehensive performance analysis on Intel GPUs demonstrated a 40% increase in performance compared to other available GPUs. Meanwhile, URANOS has been enhanced for portability to AMD GPUs and includes the implementation of a chunking strategy for flux calculations.

In the community of wall-bounded flows with compressible single-phase non-ideal fluids, the development of new high-fidelity GPU-accelerated solvers has only recently begun. Regarding high-fidelity numerical solvers for non-ideal compressible fluid dynamics, four in-house-codes are reported here: (i) MUSICAA [10], which employs high-order central finite-difference schemes and MPI parallelization for transitional and turbulent boundary layers of HMC fluids; (ii) 3DNS [48], which utilizes high-order compact finite-difference schemes and multi-block parallelization for turbomachinery studies; (iii) PadeLibs [49], which implements high-order compact finite-difference methods and GPU-acceleration for turbulent boundary flows with LMC fluids at supercritical pressure; and (iv) RHEA [50], which uses low-order central-difference schemes and GPU-acceleration via OpenACC for turbulent flows with LMC fluids at supercritical pressure. However, no publicly available GPU-accelerated high-order solver exists for simulating wall-bounded transitional and turbulent flows with compressible single-phase non-ideal fluids in canonical geometries. Therefore, we present a state-of-art solver featuring high-order energy-preserving schemes, achieving excellent numerical robustness with minimal numerical dissipation for the simulation of single-phase non-reacting flows involving highly non-ideal fluids.

Given the reasons outlined above, the open-source CUBic Equation of state Navier-Stokes (CUBENS) solver has been developed for Direct Numerical Simulations of transitional and turbulent wall-bounded flows with compressible non-ideal fluids. The CUBENS solver efficiently operates on the latest generations of GPU-accelerated clusters [38], such as Snellius (SURF, the Netherlands) and LUMI (CSC, Finland), and is fully compatible with the two leading and most powerful GPU architectures, NVIDIA and AMD, through the OpenACC standard [51]. With respect to the spatial discretization, high-order central finite-difference schemes are employed alongside the KEEP-PEP scheme for the convective fluxes and pressure-equilibrium conservation. The solver accounts for the strongly non-linear nature of thermodynamics and fluid properties by combining cubic equations of state with non-ideal transport-property models.

This manuscript provides a comprehensive description of the CUBENS solver, detailing its governing equations, validation, and performance. The paper is structured as follows: Section 2 introduces the conservation equations, non-ideal equations of state, and non-linear transport-property models. Section 3 discusses the numerical discretization of CUBENS, followed by its GPU implementation in Section 4. In Section 5, the solver is validated against various benchmark cases, including additional transitional boundary-layer flows characterized by (i) stable and unstable stratification and (ii) a strongly non-ideal fluid at supercritical pressure. Section 6 presents the code performance on both single and multi-CPU and -GPU architectures. Finally, conclusions are drawn in Section 7.

2. Governing equations

In the following, we present the fundamental governing equations and thermodynamic models for a single-phase, non-reacting compressible fluid suitable for the non-ideal thermodynamic region. The fluid is assumed to be Newtonian and in thermodynamic and chemical equilibrium.

2.1. Flow-conservation equations

The non-dimensional, fully-compressible Navier-Stokes (NS) equations are written as

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \left[\mathbf{F} + \mathbf{F}_{visc.}\right]}{\partial x} + \frac{\partial \left[\mathbf{G} + \mathbf{G}_{visc.}\right]}{\partial y} + \frac{\partial \left[\mathbf{H} + \mathbf{H}_{visc.}\right]}{\partial z} = \mathbf{S}, \tag{1}$$

and are integrated by the CUBENS solver over time t and in the three spatial dimensions $\left[x_1, x_2, x_3\right] = [x, y, z]$, using Cartesian coordinates in the streamwise, wall-normal, and spanwise directions. The state vector $\mathbf{Q}$ in Eq. (1) is defined as

$$\mathbf{Q} = [\rho, \rho u, \rho v, \rho w, \rho E]^T, \tag{2}$$

where ρ is the fluid density; ρu, ρv, ρw are the x-, y-, and z-momentum components with velocity $|\mathbf{u}| = [u, v, w]^T$, respectively; $E = e + |\mathbf{u}|^2/2$ is the specific total energy, where e is the specific internal energy (see Section 2.2). In Eq. (1), $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{H}$ are the Euler fluxes, specified as

$$\mathbf{F}=\begin{pmatrix}\rho u\\ \rho uu+p\\ \rho uv\\ \rho uw\\ u(\rho E+p)\end{pmatrix},\quad \mathbf{G}=\begin{pmatrix}\rho v\\ \rho vu\\ \rho vv+p\\ \rho vw\\ v(\rho E+p)\end{pmatrix},\quad \mathbf{H}=\begin{pmatrix}\rho w\\ \rho wu\\ \rho wv\\ \rho ww+p\\ w(\rho E+p)\end{pmatrix}, \tag{3}$$

where p is the pressure, and the term $\rho E+p$ is equal to ρH, with H being the specific total enthalpy. Likewise, the viscous fluxes are expressed as:

$$\mathbf{F}_{visc.}=\begin{pmatrix}0\\ -\tau_{xx}\\ -\tau_{xy}\\ -\tau_{xz}\\ -u\tau_{xx}-v\tau_{xy}-w\tau_{xz}+q_x\end{pmatrix},\quad \mathbf{G}_{visc.}=\begin{pmatrix}0\\ -\tau_{xy}\\ -\tau_{yy}\\ -\tau_{yz}\\ -u\tau_{xy}-v\tau_{yy}-w\tau_{yz}+q_y\end{pmatrix},\quad \mathbf{H}_{visc.}=\begin{pmatrix}0\\ -\tau_{xz}\\ -\tau_{yz}\\ -\tau_{zz}\\ -u\tau_{xz}-v\tau_{yz}-w\tau_{zz}+q_z\end{pmatrix}. \tag{4}$$

The viscous stress tensor τ_{ij} and the heat flux vector q_j are given as

$$\tau_{ij}=\frac{\lambda}{Re}\delta_{ij}\frac{\partial u_k}{\partial x_k}+\frac{\mu}{Re}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right),\qquad q_j=-\frac{\kappa}{ReEcPr}\frac{\partial T}{\partial x_j}, \tag{5}$$

where μ is the dynamic viscosity, $\lambda=-2/3\mu$ is the second viscosity coefficient, δ_{ij} is the Kronecker delta, κ is the thermal conductivity, and T is the fluid temperature. The source term $\mathbf{S}$ in Eq. (1) includes (i) the forcing term f_x to enforce constant mass-flow-rate in channel simulations, and (ii) the buoyancy force due to strong density variations along the wall-normal direction in supercritical fluids [52]. Thus, $\mathbf{S}$ is expressed as:

$$\mathbf{S}=\begin{pmatrix}0\\ f_x\\ -Ri_{unit}(\rho-1)\\ 0\\ -Ri_{unit}(\rho-1)v+uf_x\end{pmatrix}. \tag{6}$$

The non-dimensionalization of Eq. (1) is based on the following reference values

$$t=\frac{t^*u^*_{ref}}{L^*_{ref}},\quad x_i=\frac{x^*_i}{L^*_{ref}},\quad u_i=\frac{u^*_i}{u^*_{ref}},\quad \rho=\frac{\rho^*}{\rho^*_{ref}},\quad p=\frac{p^*}{\rho^*_{ref}u^{*2}_{ref}},$$
$$T=\frac{T^*}{T^*_{ref}},\quad E=\frac{E^*}{u^{*2}_{ref}},\quad \mu=\frac{\mu^*}{\mu^*_{ref}},\quad \kappa=\frac{\kappa^*}{\kappa^*_{ref}}, \tag{7}$$

where $(\cdot)^*$ denotes dimensional quantities, and $(\cdot)_{ref}$ corresponds to reference conditions based on the considered geometry configuration (see Section 3.3). As a result, the corresponding non-dimensional characteristic numbers are defined as

$$Re=\frac{\rho^*_{ref}u^*_{ref}L^*_{ref}}{\mu^*_{ref}},\quad Ec=\frac{u^{*2}_{ref}}{c^*_{p,ref}T^*_{ref}},$$
$$Pr=\frac{c^*_{p,ref}\mu^*_{ref}}{\kappa^*_{ref}},\quad Ri_{unit}=\frac{1}{Fr^2}=\frac{L^*_{ref}g^*}{u^{*2}_{ref}}, \tag{8}$$

where $c^*_{p,ref}$ is the specific heat capacity at constant pressure. The Reynolds number Re and the unit Richardson number Ri_{unit}, or the Froude number Fr, are based on a chosen length scale L^*_{ref}, and Ri_{unit} depends on the gravitational acceleration g^*. Moreover, in Eq. (8), Ec is the Eckert number, and Pr is the Prandtl number. The Mach number $M=u^*_{ref}/c^*_{ref}$, where c^*_{ref} is the speed of sound, can be obtained from Ec as a function of the selected equation of state.

2.2. Equations of state

In order to close the conservation equations in Eq. (1), thermal (i.e., $p=p(\rho,T)$) and caloric (i.e., $e=e(\rho,T)$) equations of state (EoS) need to be defined by satisfying the compatibility condition, which is defined as

$$e=e_{ref}+\int_{T_{ref}}^{T}c_{v,\infty}(T')\,dT'-\int_{\rho_{ref}}^{\rho}\left(T\frac{\partial p}{\partial T}\bigg|_\rho-\frac{p}{\rho'^2}\right)d\rho', \tag{9}$$

where $(\cdot)_{ref}$ denotes a reference state, $c_{v,\infty}$ is the specific heat capacity at constant volume in the ideal-gas limit, and $(\cdot)'$ indicates integration variables. Non-ideal gas EoS are expressed in reduced form, enabling the same departure from the vapor-liquid critical point for different fluids. Quantities at the thermodynamic critical point are indicated by the subscript $(\cdot)^*_c$. The speed of sound c, isobaric specific heat capacity c_p, and thermal expansion coefficient α_v in general form are:

$$c^2=\frac{\partial p}{\partial\rho}\bigg|_e+\frac{p}{\rho^2}\frac{\partial p}{\partial e}\bigg|_\rho,\quad c_p=c_v+\frac{T}{\rho^2}\frac{\partial\rho}{\partial p}\bigg|_T\frac{\partial p}{\partial T}\bigg|_\rho,\quad \alpha_v=\frac{1}{\rho}\frac{\partial\rho}{\partial T}\bigg|_p. \tag{10}$$

These quantities are necessary for the implementation of the non-reflecting boundary conditions, as discussed in Section 3.4. Hereafter, the different EoS implemented in CUBENS are reported. The ideal-gas law is included for completeness, as it is also considered for validation against previous ideal-gas direct numerical simulations found in the literature.

2.2.1. Ideal gas (calorically perfect)

The non-dimensional ideal-gas law $p=\rho R_gT$ holds, where the dimensionless specific gas constant R_g is defined as $1/(\gamma M^2_\infty)$. The heat capacity ratio $\gamma=c_p/c_v$ is constant. Consequently, the caloric equation state, given as $e=h-p/\rho=c_vT$, can be used to explicitly compute the temperature. Furthermore, the speed of sound c and thermal expansion coefficient α_v are determined as:

$$c=\sqrt{\gamma(\gamma-1)e},\quad \alpha_v=\frac{1}{T}. \tag{11}$$

2.2.2. Van der Waals

The reduced cubic Van der Waals EoS is expressed with the compressibility factor at the critical point $Z_c=p^*_c/(\rho^*_cR^*_gT^*_c)=3/8$. The thermal and caloric equation are given as

$$p_r=\frac{8\rho_rT_r}{3-\rho_r}-3\rho_r^2,\quad e_r=\frac{c_{v,r}T_r}{Z_c}-3\rho_r,\quad c_{v,r}=\frac{c^*_v}{R^*_g}=\frac{f}{2}, \tag{12}$$

where $p_r=p^*/p^*_c$ is the reduced pressure, $T_r=T^*/T^*_c$ is the reduced temperature, $\rho_r=\rho^*/\rho^*_c$ is the reduced density, $e_r=e^*\rho^*_c/p^*_c$ is the reduced energy, $c_{v,r}=c^*_v/R^*_g$ is the reduced isochoric specific heat capacity, and f is the number of active degrees of freedom of the fluid. Additionally, the reduced isobaric specific heat capacity $c_{p,r}=c^*_p/R^*_g$ and speed of sound $c^2_r=c^{*2}\rho^*_c/p^*_c$ are calculated as:

$$c_{p,r}=c_{v,r}+\left[1-\frac{\rho_r(3-\rho_r)^2}{4T_r}\right]^{-1},$$
$$c_r^2=\left(\frac{c_{v,r}+1}{c^2_{v,r}}\right)(e_r+3\rho_r)\left(\frac{3}{3-\rho_r}\right)^2-6\rho_r. \tag{13}$$

Moreover, the reduced thermal expansion coefficient $\alpha_{v,r}=\alpha^*_vT^*_c$ is expressed as

$$\alpha_{v,r} = \frac{1}{\rho_r}\frac{\partial p_r}{\partial T_r}\bigg|_{\rho_r} \Big/ \frac{\partial p_r}{\partial \rho_r}\bigg|_{T_r}, \tag{14}$$

with

$$\frac{\partial p_r}{\partial T_r}\bigg|_{\rho_r} = \frac{8\rho_r}{3-\rho_r}, \quad \frac{\partial p_r}{\partial \rho_r}\bigg|_{T_r} = \frac{24T_r}{(3-\rho_r)^2} - 6\rho_r. \tag{15}$$

2.2.3. Peng-Robinson

The reduced cubic Peng-Robinson EoS is presented hereafter. The thermal equation is given as

$$\begin{aligned} p_r &= \frac{Z_c^{-1}\rho_r T_r}{1-b_r Z_c^{-1}\rho_r} - \frac{\alpha a_r Z_c^{-2}\rho_r^2}{1+2b_r Z_c^{-1}\rho_r - b_r^2 Z_c^{-2}\rho_r^2}, \\ a_r &= 0.45724, \quad b_r = 0.07780, \quad \alpha = \left[1 + K\left(1-\sqrt{T_r}\right)\right]^2, \end{aligned} \tag{16}$$

where the constants a_r, b_r, and $Z_c = 0.3112$ are obtained by enforcing the critical-point conditions. The term accounting for inter-molecular forces, α, is temperature-dependent, and $K = 0.37464 + 1.54226\bar{\omega} - 0.26992\bar{\omega}^2$ is a function of the acentric factor $\bar{\omega}$. By applying the compatibility condition, the caloric equation of state becomes

$$\begin{aligned} e_r &= c_{v,r} T_r Z_c^{-1} + \frac{a_r Z_c^{-1}}{2b_r\sqrt{2}}\sqrt{\alpha}(K+1)\ln\left(\frac{1+b_r Z_c^{-1}\rho_r(1-\sqrt{2})}{1+b_r Z_c^{-1}\rho_r(1+\sqrt{2})}\right), \\ c_{v,r} &= \frac{f}{2} - \frac{a_r K(K+1)}{4b_r\sqrt{2T_r}}\ln\left(\frac{1+b_r Z_c^{-1}\rho_r(1-\sqrt{2})}{1+b_r Z_c^{-1}\rho_r(1+\sqrt{2})}\right), \end{aligned} \tag{17}$$

where f is the number of active degrees of freedom of the fluid. Additionally, the reduced isobaric heat capacity $c_{p,r}$, speed of sound c_r^2, and thermal expansion coefficient $\alpha_{v,r}$ are calculated as:

$$\begin{aligned} c_{p,r} &= c_{v,r} + \frac{Z_c T_r}{\rho_r^2}\frac{\partial p_r}{\partial T_r}\bigg|_{\rho_r}^2 \frac{\partial p_r}{\partial \rho_r}\bigg|_{T_r}, \quad c_r^2 = \frac{Z_c T_r}{c_{v,r}\rho_r^2}\frac{\partial p_r}{\partial T_r}\bigg|_{\rho_r}^2 + \frac{\partial p_r}{\partial \rho_r}\bigg|_{T_r}, \\ \frac{\partial p_r}{\partial T_r}\bigg|_{\rho_r} &= K\sqrt{\frac{\alpha}{T_r}}\frac{a_r\rho_r^2 Z_c^{-2}}{1+2b_r Z_c^{-1}\rho_r - b_r^2 Z_c^{-2}\rho_r^2} - \frac{\rho_r Z_c^{-1}}{b_r Z_c^{-1}\rho_r - 1}, \\ \frac{\partial p_r}{\partial \rho_r}\bigg|_{T_r} &= \frac{Z_c^{-1}T_r}{(\rho_r b_r Z_c^{-1}-1)^2} - \frac{a_r\alpha Z_c^{-2}(2\rho_r + 2b_r Z_c^{-1}\rho_r^2)}{\left[1+2b_r Z_c^{-1}\rho_r - \rho_r^2 b_r^2 Z_c^{-2}\right]^2}, \\ \alpha_{v,r} &= \frac{1}{\rho_r}\frac{\partial p_r}{\partial T_r}\bigg|_{\rho_r} \Big/ \frac{\partial p_r}{\partial \rho_r}\bigg|_{T_r}. \end{aligned} \tag{18}$$

2.3. Transport properties

Along with the equation of state, to close the Navier-Stokes equations, the fluid's dynamic viscosity and thermal conductivity need to be specified.

2.3.1. Power and Sutherland's law (calorically perfect gas)

The non-dimensional dynamic viscosity $\mu = \mu^*/\mu^*_\infty$ is calculated as

$$\mu = T^n; \quad \mu = T^{3/2}\frac{1 + T^*_{ref}/T^*_\infty}{T + T^*_{ref}/T^*_\infty}, \quad \text{with } T^*_{ref} = 110.4\,\text{K}, \tag{19}$$

for the power law (n is the exponent) and Sutherland's law [53], respectively. Simultaneously, the following relation $\kappa = \mu$ holds for the non-dimensional thermal conductivity κ.

2.3.2. Jossi, Stiel, and Thodos

Based on theoretical scalings and experimental findings, analytical expressions for dynamic viscosity and thermal conductivity for non-polar supercritical fluids were derived by Jossi et al. [54] and Stiel and Thodos [55], respectively, and are denoted as JST hereafter. The advantage of this model lies in its reliance on reduced quantities, providing a more universal and general representation, allowing for easier comparison across different species. The dynamic viscosity μ^* is defined implicitly as

$$\begin{aligned} F_\rho &= \left[(\mu^* - \mu_0^*)\xi^* + 10^{-4}\right]^{1/4} = \\ &= 0.10230 + 0.023364\rho_r + 0.058533\rho_r^2 \\ &- 0.040758\rho_r^3 + 0.0093324\rho_r^4, \end{aligned} \tag{20}$$

where ρ_r is the reduced density, $\xi^* = 10^3 T_c^{*1/6}/(M^{*1/2}p_c^{*2/3})$ is the viscosity parameter, and μ_0^* is the viscosity at moderate pressures ($0.1 < p^* < 5\,\text{atm}$), given as

$$\mu_0^*\xi^* = \begin{cases} 34\times10^{-5}\,T_r^{0.94}, & T_r \le 1.50, \\ 17.78\times10^{-5}\left(4.58T_r - 1.67\right)^{5/8}, & T_r > 1.50, \end{cases} \tag{21}$$

where T_r is the reduced temperature. The non-dimensional dynamic viscosity μ can then be formulated as:

$$\mu = \frac{\mu^*}{\mu^*_\infty} = \frac{F_\rho^4 - 10^{-4} + \mu_0^*\xi^*}{F_{\rho,\infty}^4 - 10^{-4} + \mu_{0,\infty}^*\xi^*}. \tag{22}$$

The thermal conductivity κ^* is

$$\begin{aligned} G_\rho &= (\kappa^* - \kappa_0^*)\lambda^* Z_c^5 = 4.1868\times10^{-6} \\ &\times\begin{cases} 14.0\left(\exp\left[0.535\rho_r\right] - 1\right), & \rho_r < 0.50, \\ 13.1\left(\exp\left[0.67\rho_r\right] - 1.069\right), & 0.50 \le \rho_r \le 2.0, \\ 2.976\left(\exp\left[1.155\rho_r\right] + 2.016\right), & 2.0 < \rho_r \le 2.80, \end{cases} \end{aligned} \tag{23}$$

where $\lambda^* = \xi^* M^*$ is the thermal conductivity parameter, and κ_0^* is the thermal conductivity at atmospheric pressures of a polyatomic gas [56], expressed as:

$$\kappa_0^* = \left(0.307\frac{c_v^*}{R_g^*} + 0.539\right)\kappa^*_{0,\text{mon}}, \quad \kappa^*_{0,\text{mon}} = \frac{15}{4}\frac{R_g^*}{M^*}\mu_0^*. \tag{24}$$

The non-dimensional thermal conductivity κ can then be formulated as:

$$\kappa = \frac{\kappa^*}{\kappa^*_\infty} = \frac{G_\rho + Z_c^5\kappa_0^*\lambda^*}{G_{\rho,\infty} + Z_c^5\kappa^*_{0,\infty}\lambda^*}. \tag{25}$$

2.3.3. Chung

A more accurate representation of the liquid-like regime is achieved with the generalized laws of Chung et al. [57]. Unlike JST, the considered fluid must be explicitly specified. Thus, the dimensional dynamic viscosity and thermal conductivity are calculated as

$$\mu^* = \mu_k^* + \mu_p^*, \quad \kappa^* = \kappa_k^* + \kappa_p^*, \tag{26}$$

where μ_k^* and κ_k^* are the modified dilute-gas viscosity and conductivity, respectively, and μ_p^* and κ_p^* are the respective correction terms for density dependency. For brevity, the details are omitted here but can be found in Appendix A.1.

3. Numerical methods

A summary of the numerical techniques employed to solve Eq. (1) is discussed hereafter. For more details, see the corresponding references.

3.1. Spatial discretization

A high-order explicit central finite-difference method is combined with split convective derivatives for local kinetic energy and entropy conservation. Here, the high-order KEEP scheme in Cartesian coordinates Kuya and Kawai [29] is combined with the variation of Shima et al. [33] for PEP conditions. By using this scheme, both primary (mass, momentum, and total energy) and secondary (kinetic energy and entropy) conservative quantities are conserved while maintaining numerical stability. Thus, aliasing errors are minimized, and the whole range of turbulence scales can be captured without any stabilization scheme, such as artificial diffusivity or filtering [26]. Note that the PEP

Table 1
Coefficients a_{q,i_s} for the central difference approximation of first derivative. q is the order of accuracy, and i_s is the stencil size.

q	$a_{q,1}$	$a_{q,2}$	$a_{q,3}$
2	1/2	-	-
4	2/3	-1/12	-
6	3/4	-3/20	1/60

discretization of internal energy exactly holds under the ideal-gas assumption. However, for non-ideal fluids, no current numerical scheme can fully conserve this property. Therefore, a compromise is made, with the ideal-gas PEP condition serving as a first approximation to satisfy pressure equilibrium for arbitrary equations of state (see "PEP-IG" scheme in Bernades et al. [36]).

By defining the numerical flux $\tilde{F}_\phi|_{(i\pm1/2)}$ for a scalar ϕ at the cell interfaces ($i\pm1/2$), we can write the spatial derivative of the convective term as

$$\frac{\partial \rho\phi u}{\partial x} \simeq \frac{\tilde{F}_\phi|_{(i+1/2)} - \tilde{F}_\phi|_{(i-1/2)}}{\Delta x}, \tag{27}$$

considering a second-order discretization. Thus, Eq. (1) in differential and inviscid form is discretized by the numerical fluxes at cell index i as:

$$\left.\frac{\partial \rho}{\partial t}\right|_i + \frac{\tilde{C}|_{(i+1/2)} - \tilde{C}|_{(i-1/2)}}{\Delta x_j} = 0, \tag{28}$$

$$\left.\frac{\partial \rho u_k}{\partial t}\right|_i + \frac{\tilde{M}_{u_k}|_{(i+1/2)} - \tilde{M}_{u_k}|_{(i-1/2)}}{\Delta x_j} + \frac{\tilde{\Pi}|_{(i+1/2)} - \tilde{\Pi}|_{(i-1/2)}}{\Delta x_j} = 0, \tag{29}$$

where $\tilde{C}$, $\tilde{M}_{u_k}$, $\tilde{\Pi}$ are the mass, momentum, and pressure-gradient ($k = 1,2,3$) numerical fluxes, respectively, and:

$$\left.\frac{\partial \rho E}{\partial t}\right|_i + \frac{\tilde{K}|_{(i+1/2)} - \tilde{K}|_{(i-1/2)}}{\Delta x_j} + \frac{\tilde{I}|_{(i+1/2)} - \tilde{I}|_{(i-1/2)}}{\Delta x_j} + \frac{\tilde{P}|_{(i+1/2)} - \tilde{P}|_{(i-1/2)}}{\Delta x_j} = 0, \tag{30}$$

where $\tilde{K}$, $\tilde{I}$, and $\tilde{P}$ are the kinetic energy, internal energy, and pressure-diffusion numerical fluxes, respectively. Their expressions can be found in Shima et al. [33], where a new internal energy numerical flux that maintains velocity and pressure equilibrium at the contact interface is defined. Note that, in the context of non-ideal fluids, a diffused interface occurs along the Widom line when the flow is at supercritical pressure and under transcritical conditions. This scenario is characterized by a continuous transcritical contact surface [34] between the high-density liquid-like and the low-density vapor-like regime. In CUBENS, we focus on fully resolving the diffuse interface and selecting the appropriate flow regime to minimize spurious oscillations, as the PEP condition of Shima et al. [33] strictly applies only under ideal-gas conditions.

The high-order extension is achieved by following the approach proposed in Pirozzoli [58] for KEP schemes. Hence, the convective numerical flux $\tilde{F}_\phi|_{(i\pm1/2)}$ is extended to an arbitrary order of accuracy as

$$\tilde{F}_\phi|_{(i+1/2)} = 2\sum_{i_s=1}^{q/2} a_{q,i_s} \times \sum_{j=0}^{i_s-1} \frac{\rho_{(i\mp j)} + \rho_{(i\mp j\pm i_s)}}{2} \frac{\phi_{(i\mp j)} + \phi_{(i\mp j\pm i_s)}}{2} \frac{u_{(i\mp j)} + u_{(i\mp j\pm i_s)}}{2}, \tag{31}$$

where a_{q,i_s} are the coefficients used for the central difference approximations of the first derivative, with an order of accuracy q and stencil size i_s (see Table 1 for up to 6th-order). In CUBENS, the order of accuracy of the convective terms ($a_{q,i_s} = 2,4,6$) can be explicitly selected by the user. The diffusion fluxes in Eq. (4) are discretized using a central finite-difference approximation both for first- and second-order derivatives, with CUBENS supporting 2nd- and 4th-order finite differences.

3.2. Time discretization

After spatial discretization, the system in Eq. (1) can be explicitly advanced in time. This results in

$$\frac{\partial \mathbf{Q}}{\partial t} = \mathbf{R}(\mathbf{Q}, t), \tag{32}$$

where $\mathbf{R}$ is the non-linear spatial differential operator and $\mathbf{Q}$ is the state vector as in Eq. (2). Subsequently, a third-order Total Variation Diminishing (TVD) low-storage Runge-Kutta scheme [59] is applied to advance $\mathbf{Q}^n$ from time t^n to $\mathbf{Q}^{n+1}$ at t^{n+1}, with time step $\Delta t = t^{n+1} - t^n$. This yields:

$$\begin{aligned} \mathbf{Q}^{(1)} &= \mathbf{Q}^n + \Delta t \mathbf{R}(\mathbf{Q}^n), \\ \mathbf{Q}^{(2)} &= \frac{3}{4}\mathbf{Q}^n + \frac{1}{4}\left[\mathbf{Q}^{(1)} + \Delta t \mathbf{R}\left(\mathbf{Q}^{(1)}\right)\right], \\ \mathbf{Q}^{(n+1)} &= \frac{1}{3}\mathbf{Q}^n + \frac{2}{3}\left[\mathbf{Q}^{(2)} + \Delta t \mathbf{R}\left(\mathbf{Q}^{(2)}\right)\right]. \end{aligned} \tag{33}$$

To ensure numerical stability [60], the Courant-Friedrichs-Lewy (CFL) and Fourier criteria need to hold according to

$$\Delta t_{\mathrm{CFL}} \propto \left(\frac{\Delta x_j}{|\mathbf{u}_j + c|}\right), \quad \Delta t_\mu \propto \left(\frac{\Delta x_j^2}{\mu}\right), \quad \Delta t_\kappa \propto \left(\frac{\Delta x_j^2}{\kappa}\right), \tag{34}$$

where Δx_j is the grid size in the corresponding j-direction, and c is the speed of sound (see Section 2.3). The global time step Δt to advance $\mathbf{Q}^n$ is then calculated from Eq. (32) as a minimum condition from:

$$\Delta t = \mathrm{CFL} \cdot \min(\Delta t_{\mathrm{CFL}}, \Delta t_\nu, \Delta t_\alpha). \tag{35}$$

For CFL < 1, the 3rd-order Runge-Kutta method is numerically stable [59]. Thus, the CFL number can be set arbitrarily by the user, typically between 0.5 and 0.8.

In numerical simulations of controlled transitional boundary layers [7,61,62], it is also common practice to specify the time step Δt based on the frequency of the primary disturbance ω_0 as follows

$$\Delta t = \frac{2\pi}{\omega_0}\frac{1}{LP}, \tag{36}$$

where LP is a multiple of the number of samples saved during each forcing period. The parameter LP can be adjusted by the user by, for instance, constraining CFL to values not larger than 0.8.

3.3. Computational domains

Two canonical geometries can be considered in CUBENS: channel flow and flat-plate boundary layer. While the former configuration can be utilized in alignment with DNS studies under supercritical pressure conditions [14,18], the canonical flat-plate geometry remains largely unexplored for transitional and turbulent boundary layers with LMC fluids, particularly under the influence of density stratification.

3.3.1. Channel

The channel configuration has a wall-normal height L_y of $2H^*$, where H^* is the half channel height and represents the reference length. For the non-dimensionalization of Eq. (1), the following reference values are used

$$\begin{aligned} t &= \frac{t^* u^*_{ref}}{H^*}, \quad x_i = \frac{x_i^*}{H^*}, \quad u_i = \frac{u_i^*}{u^*_{ref}}, \quad \rho = \frac{\rho^*}{\rho^*_w}, \quad p = \frac{p^*}{\rho^*_w u^{*2}_{ref}}, \\ T &= \frac{T^*}{T^*_w}, \quad E = \frac{E^*}{u^{*2}_{ref}}, \quad \mu = \frac{\mu^*}{\mu^*_w}, \quad \kappa = \frac{\kappa^*}{\kappa^*_w}, \end{aligned} \tag{37}$$

where $(\cdot)^*_w$ indicates the selected dimensional reference wall values (note that non-isothermal conditions can also be considered in CUBENS

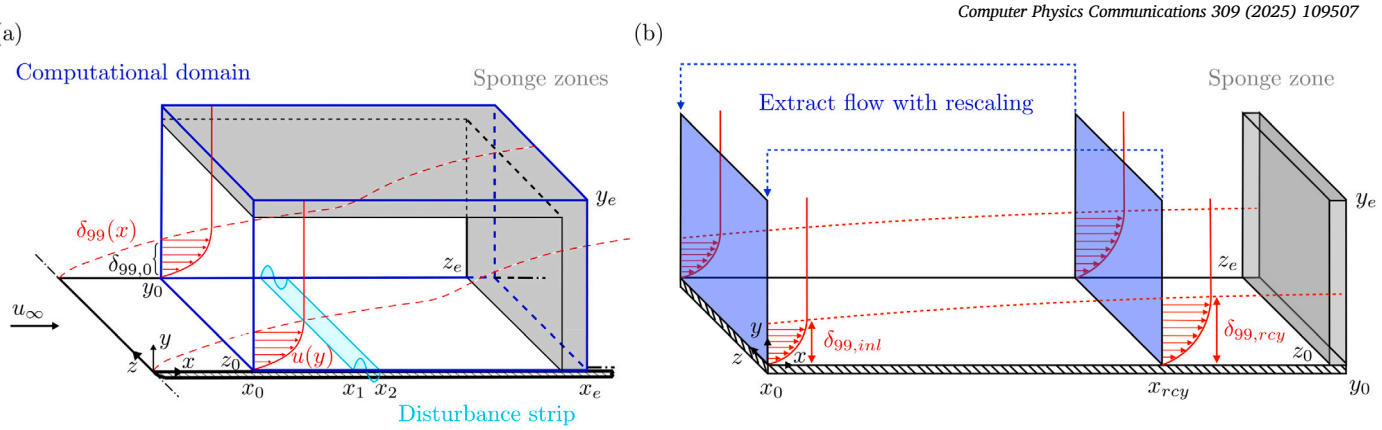

Fig. 1. DNS computational domain for (a) transitional boundary layer (see Section 3.3.2) and (b) turbulent boundary layer (see Section 3.3.3). All the reference values are explained in Sections 3.3.2, 3.3.3, and 3.4.

by imposing a temperature difference between the top and bottom walls), and u^*_{ref} is the reference velocity, e.g., centerline velocity. In streamwise and spanwise directions, the mesh is uniform, while in the wall-normal direction, it follows a hyperbolic tangent function given by

$$y = y_e \left[\left(1 + \frac{\tanh(0.5\sigma(\eta - 1))}{\tanh(0.5\sigma)} \right) \right], \tag{38}$$

where the wall-normal coordinate η extends from 0 to 2.

3.3.2. Transitional boundary layer

The computational domain for a laminar boundary layer transitioning to turbulence is illustrated in Fig. 1(a). The inflow is located at x_0 and the outflow at x_e. The domain height is given by y_e, and the spanwise domain extends from 0 to z_e. To achieve laminar-to-turbulent transition, disturbances are triggered by a wall-normal blowing and suction strip ($x_1 < x < x_2$), see Section 3.4. The reference length L^*_{ref} in Eqs. (7) and (8) is chosen as the boundary-layer thickness at the start of the computational domain, $\delta^*_{99,0}(u = 0.99)$. The other reference quantities in Eqs. (7) and (8) correspond to free-stream flow conditions and are denoted with $(\cdot)^*_\infty$. After the re-scaling of all dimensions, the dimensionless boundary-layer thickness $\delta_{99,0}$ at the domain inlet $Re_{\delta,0}$ is equal to unity. The corresponding local Reynolds number Re_δ, based on the Blasius length scale δ^*, and Reynolds number Re_x, based on the streamwise coordinate x^*, are dependent on $\delta^*_{99,0}$ and can be written as:

$$Re_\delta = \frac{\rho^*_\infty u^*_\infty \delta^*}{\mu^*_\infty} = \sqrt{Re_x} = Re_{\delta,0} \frac{\delta^*}{\delta^*_{99,0}}, \quad \delta^* = \sqrt{\frac{\mu^*_\infty x^*}{\rho^*_\infty u^*_\infty}}. \tag{39a}$$

In the spanwise z-direction, an equidistant mesh is applied with constant grid spacing Δz. The same applies for the streamwise x-direction, although streamwise stretching with $\Delta x \propto \tanh(x)$ is available. In the wall-normal y-direction, a grid stretching towards the wall is applied. The wall-normal distribution y with N_y grid points is then expressed as

$$\begin{aligned} y &= y_e \left[K_1 \eta + (1 - K_1) \left(1 + \frac{\tanh(0.5\sigma(\eta - 1))}{\tanh(0.5\sigma)} \right) \right], \\ \eta &= 0, \ldots, 1, \quad K_1 = \frac{0.6}{Re_{\tau,0}} \frac{N_y - 1}{y_e}, \end{aligned} \tag{40}$$

where $Re_{\tau,0} = \delta^*_{99,0}/\delta^*_{99,\upsilon}$ is the prescribed friction Reynolds number at the domain inlet, with $\delta^*_{99,\upsilon} = \mu^*_w/(\rho^*_w u^*_\tau)$, and σ is the stretching factor for near-wall region refinement.

3.3.3. Turbulent boundary layer

The computational domain in CUBENS for a turbulent boundary layer is illustrated in Fig. 1(b). The same numerical setup and reference quantities are chosen as for a transitional boundary layer in Fig. 1(a). However, the following differences need to be taken into account. The reference length is selected as the mean turbulent boundary-layer thickness at the inlet, defined as $\delta^*_{99,inl}$, which remains constant by fixing the mean inflow profiles. Inflow turbulence fluctuations are generated using the recycling-rescaling (RR) method of Urbin and Knight [63]. First, the mean velocity and temperature profiles, either known a priori or obtained from a transitional boundary-layer simulation, are extracted at the streamwise recycling position, x^*_{rcy}. The rescaling parameter β, which is the ratio of the inlet and recycling friction velocities, is estimated from the ratio of the mean turbulent boundary-layer thicknesses as

$$\beta \equiv \frac{u^*_{\tau,inl}}{u^*_{\tau,rcy}} \approx \left(\frac{\delta^*_{99,rcy}}{\delta^*_{99,inl}} \right)^{0.1}, \tag{41}$$

where $\delta_{99,rcy}$ is the mean turbulent boundary-layer thickness at the recycling position. The recycled velocity profiles are then decomposed into a mean, denoted as $\langle\cdot\rangle$, and fluctuating component, denoted as $(\cdot)'$. The log- and defect-laws are applied to the inner- and outer-layer scaling of the velocity, respectively [63]. Note that this method can be further improved, particularly for strongly cooled or heated boundary layers and high Mach number flows, by applying the approach proposed by Hasan et al. [64]. This yields the following relationships between the mean velocity profiles at the inlet and the recycling position as

$$\begin{aligned} \langle u \rangle^{inn}_{VD,inl} &= \beta \langle u \rangle^{inn}_{VD,rcy}, \\ \langle u \rangle^{out}_{VD,inl} &= u_{\infty,VD} - \beta \left(u_{\infty,VD} - \langle u \rangle^{out}_{VD,rcy} \right), \end{aligned} \tag{42}$$

where superscripts *inn* and *out* denote the inner- and outer-layer scaled quantities, respectively, and the subscript *VD* refers to the van Driest's transformation [65], which is defined as

$$\langle u \rangle_{VD} = \frac{u_\infty}{A} \sin^{-1} \left(A \frac{\langle u \rangle}{u_\infty} \right), \quad A = \sqrt{\frac{M^2_\infty Pr_t (\gamma - 1)/2}{1 + M^2_\infty Pr_t (\gamma - 1)/2}}, \tag{43}$$

where the turbulent Prandtl number Pr_t is constant and equal to 0.89 (ideal gas). The inner- and outer-layer fluctuating components are rescaled using the same parameter β. Finally, the mean and fluctuating components are combined using a weighted average [66] to determine the new inflow profile as

$$u_{inl} = \left(\langle u \rangle^{inn}_{inl} + u'^{inn}_{inl} \right) (1 - W(y)) + \left(\langle u \rangle^{out}_{inl} + u'^{out}_{inl} \right) W(y), \tag{44}$$

where the weighting function $W(y)$ is:

$$W(y) = \frac{1}{2} \left(1 + \left\{ \tanh \left[\frac{4(y - B)}{(1 - 2B)y + B} \right] / \tanh(4) \right\} \right), \quad B = 0.2. \tag{45}$$

This same weighting procedure is applied consistently to all other flow variables.

Table 2
Summary of the non-reflecting boundary conditions for flat-plate boundary layer (BL) and channel configuration. $\mathcal{L}_1$ is the amplitude of the incoming wave with constant K according to Rudy and Strikwerda [70].

Case	Inlet	Top	Outlet	Bottom
BL	subsonic	$\mathcal{L}_1 = K(p - p_\infty)$	$\mathcal{L}_1 = 0$	no-slip
Channel	periodic	no-slip	periodic	no-slip

3.4. Boundary conditions

Based on the computational domain described in Section 3.3, non-reflecting boundary conditions are applied at the domain boundaries according to the inviscid characteristic wave analysis of Thompson [67] and Poinsot and Lele [68]. To account for gas non-ideality, the extension of Okong'o and Bellan [69] for single-phase flow is implemented. While the local associated one-dimensional inviscid (LODI) problem is not affected by gas non-ideality, the speed of sound c, thermal expansion coefficient α_v, and isobaric specific heat capacity c_p are modified according to the corresponding EoS in Section 2.2. Additionally, the inclusion of a wall-normal buoyant force, as indicated by the source term in Eq. (6), should be considered due to the large wall-normal density gradients that may arise within the boundary layer, particularly at supercritical pressure under transcritical conditions. Thus, an inhomogeneous term, i.e., $\mathbf{C}$ in Eq. (46), appears on the right-side of the characteristic equations [67]. The one-dimensional characteristic form of the Euler equations reads:

$$\frac{\partial \mathbf{Q}^*}{\partial t} + \mathbf{A}\frac{\partial \mathbf{Q}^*}{\partial x} + \mathbf{C} = 0, \quad \text{with}$$
$$\mathbf{Q}^* = \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} u & \rho & 0 \\ 0 & u & 1/\rho \\ 0 & \rho c^2 & u \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 0 \\ Ri_{unit}(\rho - 1)/\rho \\ 0 \end{pmatrix}. \tag{46}$$

Both wave speeds λ_i and wave amplitudes $\mathcal{L}_i$ are identical to those in [69], except for $\mathcal{L}_5$ at the wall, which becomes due to the buoyancy force:

$$\mathcal{L}_{5,new} = \mathcal{L}_5 - 2cRi_{unit}(\rho - 1). \tag{47}$$

A summary of the non-reflecting boundary conditions for the aforementioned computational domains is presented in Table 2. For the boundary condition at the bottom (and at the top in the case of channel flow), CUBENS can switch between an adiabatic or isothermal no-slip, non-reflecting wall-boundary condition. Regardless of the considered geometry, periodic boundary conditions are employed in the spanwise direction.

For boundary-layer calculations, numerical sponge zones can additionally be applied to the boundaries to minimize acoustic reflections [71]. The reference solution inside the sponge region is the similarity solution of the compressible laminar boundary layer [2], combined with the chosen EoS (see [7,8] for more details).

To simulate a transitional boundary layer and to achieve a controlled laminar-to-turbulent transition, disturbances are triggered by a wall-normal blowing and suction strip. This approach represents a vibrating ribbon, in agreement with the controlled experimental studies of Kachanov et al. [72]. Two different disturbance strips between x_1 and x_2, $x_{mid} = 0.5(x_1 + x_2)$, can be selected by the user. Both are dependent on the non-dimensional frequency ω, given as

$$\omega = FRe_\delta, \quad \text{with} \quad F = \frac{2\pi f^* \mu^*_\infty}{\rho^*_\infty u^{*2}_\infty}, \tag{48}$$

where f^* is the dimensional frequency. In [73–76], the three-dimensional (3-D) wall-normal velocity distribution is prescribed as

$$v(y=0) = f(x)\left[A_{2\text{-D}}\sin(\omega_{2\text{-D}}t) + A_{3\text{-D}}\sin(\omega_{3\text{-D}}t)\cos(\beta z)\right], \tag{49}$$

where $A_{2\text{-D}}$ is the amplitude of the primary two-dimensional (2-D) and $A_{3\text{-D}}$ of the 3-D wave, with $\beta = 2\pi/z_e$ being the spanwise wavenumber. The variation of the disturbance strip in the x-direction is equal to:

$$f(x) = 15.1875\xi^5 - 35.4375\xi^4 + 20.25\xi^3,$$
$$\xi = \begin{cases} \dfrac{x - x_1}{x_{mid} - x_1}, & x_1 < x < x_{mid}, \\ \dfrac{x_2 - x}{x_2 - x_{mid}}, & x_{mid} < x < x_2. \end{cases} \tag{50}$$

Conversely, in agreement with [21,77], additional N_{mod} modes can be excited via the 3-D blowing and suction of the form

$$v(y=0) = f(x)g(z)\sum_{i=1}^{N_{mod}} A_i \sin(\omega_i t - \beta_i z),$$
$$f(x) = \exp\left[\frac{-(x - x_{mid})^2}{\sigma^2}\right], \tag{51}$$

where A_i and β_i are the wave amplitude and spanwise wavenumber of mode i, respectively. The spanwise variation of the disturbance strip is given by

$$g(z) = 1 + 0.1\exp\left[-\left(\frac{z - z_{mid} - z_w}{z_w}\right)^2\right] - 0.1\exp\left[-\left(\frac{z - z_{mid} + z_w}{z_w}\right)^2\right], \tag{52}$$

where z_{mid} and z_w are the mid-point and width of the disturbance strip, respectively.

4. Solver implementation

Given the current transformation of HPC systems towards Exascale supercomputing, it is desirable to develop a solver that can run on different modern architectures, with performance portability. In this context, two different supercomputing architectures are chosen: the Dutch national supercomputer Snellius, based at SURF (165th and 240th in the TOP500 list for CPU and GPU cores, respectively [38]) with NVIDIA A100 GPUs, and the pre-exascale EuroHPC JU LUMI supercomputer, based at CSC in Finland (5th in the TOP500 [38]), with AMD MI250X GPUs. Portability to Intel GPUs is currently not supported due to limited access to this architecture.

At the coarse-grained level, the message-passing interface (MPI) is adopted for parallel computations in distributed memory. The geometry is divided into several computational subdomains using the 2DECOMP&FFT library [78]. Specifically, the domain decomposition is performed in the streamwise and spanwise direction, with ghost cells at the domain boundaries. Thus, stencils for finite-difference schemes can be calculated with the support of data exchange to adjacent subdomains. When migrating to a system with different computing units, such as multicore processors and accelerators, the two main requisites are to have a low-cost code development that is easy to maintain for domain experts. The use of directives is desirable here, as they are naturally masked by the compiler and treated as comments for the standard CPU implementation. The OpenACC [51] standard is adopted to accelerate the solver at the fine-grained (thread-level) parallelization on GPUs and to manage host-device data movement. As a directive-based programming model designed for GPU computations, OpenACC uses directives to instruct the compiler on how to offload computations to the GPU, or on how to perform data transfer. Note that the two different architectures used in this work required different compilers that support OpenACC. On Snellius, we use the NVIDIA compiler shipped in the NVIDIA HPC SDK [79], whereas, for LUMI – and specifically its GPU partition, LUMI-G – we use the Cray compiler form of the Cray Compiler Environment (CCE). For the compilation of CUBENS on the two architectures, the solver uses a standard build system based on GNU make, tailored to easily switch

```
program main_CUBENS
...
call init() ! Subroutines initialization
!$acc enter data copyin (variables)
call init_solution() ! Initial solution initialization
!$ acc update device (variables)
do istep=istart+1,istart+nsteps ! Time loop
    ...
    call calc_timestep() ! Calculation of the time-step
    do RK_step=1,RK_order ! Loop over Runge-Kutta substeps
        call MPI_comm() ! ghost cells communication
        call set_BC() ! Boundary conditions
        call calc_Euler_flux() ! Euler fluxes
        call calc_Visc_flux() ! Viscous fluxes
        call calc_Update_variables() ! Update primary variables
        call calc_RK_step() ! Runge-Kutta step
    enddo
    !$ acc update host(variables) async(n)
    !$ acc wait
    call output()! Write output/restart files
enddo
...
!$ acc end data
...
end program
```

Algorithm 1: Simplified code structure of the main operations in CUBENS.

```
program calc_RK_step
...
!$acc parallel default(present)
!$acc loop gang, vector collapse(3)
do k=1,k_max ! Loop over the streamwise direction
    do j=1,j_max ! Loop over the spanwise direction
        do i=1,i_max ! Loop over the wall-normal direction
            rho(i,j,k) =  rho_Old(i,j,k) + dt*0.5_rp*RHS_rho1(i,j
     ,k) ! rho_Old: density at         former time step (istep-1)
            ... ! other non-conservative variables
        enddo
    enddo
enddo
!$ acc end parallel loop
...
end program
```

Algorithm 2: Calculation of the first Runge-Kutta step (RK_step = 1) for the density variable. Values at the former time step (at istep-1) are indicated with index _Old.

```
...
call timer_tic("start calc_RK_step",1)
call calc_RK_step()
call timer_toc("end calc_RK_step",1)
...
```

Algorithm 3: Profiling example of calc_RK_step with timer module.

from one architecture to the other. With the flag ARCH in the CUBENS makefile, the user can choose between the NVHPC and Cray compilers. For CPU partitions, CUBENS supports GNU, Cray, and Intel compilers.

4.1. GPU-porting

Before presenting the details of the GPU porting, Algorithm 1 shows the simplified code structure of CUBENS. CUBENS runs its GPU-accelerated cases in a so-called GPU-resident fashion. After the initialization of the main variables on the host (CPU), the data associated with variables required on the device (GPU) are copied using the `!$acc enter data copyin(list)` directive. Note that any host-to-device (H2D) or device-to-host (D2H) operation is associated with a performance loss due to limited host-device communication bandwidth and should therefore be minimized. Afterward, the initial solution is prescribed on the host before the values on the device are updated using `!$acc update device(list)`. At this point, we identify the most time-consuming part of the code: the `nsteps` time loop, which contains Runge-Kutta substeps at each `istep`, and especially the calculation of the Navier-Stokes Right-Hand Side (RHS). The RHS consists of (a) inviscid Euler fluxes in `calc_Euler_flux()` (see Eq. (3)), (b) viscous diffusive fluxes in `calc_Visc_flux()` (see Eq. (4)), (c) the source term (see Eq. (6)), and (d) boundary conditions or `set_BC()` in Section 3.4.

Next, we discuss the acceleration of operations performed within each Runge-Kutta substep, specifically contained in `calc_RK_step()`, as shown in Algorithm 1. In this regard, the OpenACC acceleration of the temporal advancement of the density variable in the first Runge-Kutta step (`RK_step=1`) is illustrated in Algorithm 2. The `parallel` construct of OpenACC is used for a prescriptive approach to the compiler. Within a parallel region of the code, the `parallel loop` construct can be applied to specifically parallelize a nested loop, as displayed.

OpenACC supports up to three levels of parallelism, ranging from coarse to fine: `gang`, `worker`, and `vector` (see [51]). In Algorithm 2, the OpenACC directive parallelizes three nested loops by distributing iterations across multiple gangs and vectorizing them for efficient parallel execution. If the level of parallelism is not specified, the compiler independently schedules the loop iterations on the target accelerator device. Furthermore, another level of parallelism can be added using the `collapse` clause, which exposes the three nested loops to thread-level parallelization. This transforms the nested loops over k_{max}, j_{max}, and i_{max} into a single loop of size $k_{max} \times j_{max} \times i_{max}$, to be distributed among parallel threads.

Once the Runge-Kutta substeps are completed, the solution on the device can be transferred to the host with the `update host` clause. This D2H data movement is again associated with efficiency loss. However, the generation of output and restart files can be controlled by the user by specifying the frequency of data writing. Another way to mitigate the problem is to exploit asynchrony between operations using OpenACC's `async(n)` clause, where `n` represents the so-called OpenACC *queue* number, similar to concurrent, independent streams in CUDA Fortran. This technique has been applied to treat the inviscid fluxed in Section 4.2.

An essential task in porting the code to GPU is profiling to assess performance. To address this, we use either (a) the NVIDIA Nsight Systems performance tool together with the NVIDIA Tools Extension (NVTX) library, which is a C-based library previously used for optimizing other GPU-based codes, see e.g., [42,45], or (b) the ROCm open-source stack for AMD GPUs [80]. Both libraries are linked to a dedicated profiling module in CUBENS, i.e., `timer.f90`. The timing module reports the average time per task for each tagged region, and it also supports more detailed reporting, including minimum and maximum times per call and per task. An example of the tagged Runge-Kutta substep calculation in CUBENS is presented in Algorithm 3.

4.2. Treatment of the inviscid fluxes

Since the host and device have physically distinct memories, which are accessible simultaneously, one can leverage both memories by performing data movement while executing parallel operations on the device. With the `async(n)` clause, independent asynchronous operations can be executed in parallel before the `wait` clause ensures the completion of all parallel operations. We employ this technique in CUBENS to speed up the calculation of the RHS. Specifically in Algorithm 1, the modules `MPI_comm`, `set_BC`, and `Euler_flux` can be effectively parallelized. In fact, `MPI_comm` module is only responsible for updating the ghost cells at the domain boundaries as a function of the finite-difference stencils. On the other hand, as displayed in Fig. 2(a), the calculation of the `Euler_flux` section can be split into two parts: the fluxes at the interior nodes, i.e., `Euler_internal`, and those at the boundaries `Euler_BC`. Consequently, the execution of `Euler_internal` is independent of the operations at the domain boundaries, and thus can be

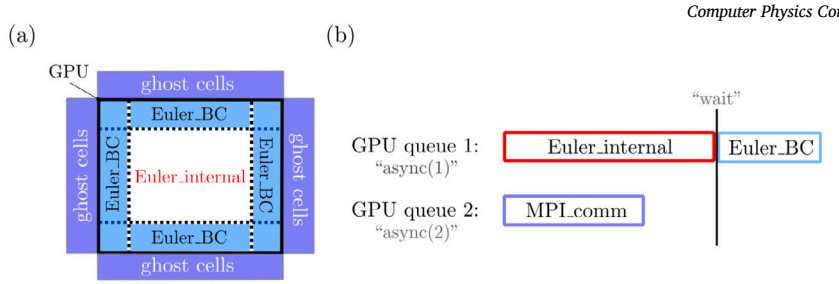

Fig. 2. CUBENS asynchronous operations: (a) GPU domain with fluxes at the internal cells and boundary cells, and ghost cells; (b) sketch of `async` clauses for `Euler_internal` and `MPI_comm` before `wait` clause.

```
...
#if defined(_GPU_DIRECT)
!$acc host_data use_device(buff_send9_k,buff_send9_j,buff_recv9_j
    ,buff_recv9_k)
call MPI_SENDRECV(buff_send9_k(:,:,:,1:nHalo),xs(1)*xs(2)*9*nHalo
    , &
                  real_type,neigh%kp,0, &
                  buff_recv9_k(:,:,:,1:nHalo),xs(1)*xs(2)*9*nHalo
    , &
                  real_type,neigh%km,0, &
                  MPI_COMM_WORLD,istat,ierr)
...
!$acc end host_data
#endif
...
```

Algorithm 4: Code snippet from the `MPI_comm` module for GPU-aware MPI data transfer. `-D_GPU_DIRECT` is a pre-processing flag.

run in parallel with `MPI_comm`. Simultaneously, the execution of `Euler_BC` can only be performed after the ghost cells have been updated. As illustrated in Fig. 2(b), a `wait` clause ensures the correct evaluation of the boundary conditions in time, and thus of the fluxes at the boundaries. With this strategy, the workload distribution of `MPI_comm` is overlapped with `Euler_internal`.

4.3. Multi-GPU implementation

For high-fidelity simulations, multiple processing units need to be exploited. In a multi-GPU implementation, two communication approaches can be identified. While in the traditional approach, accelerators are connected via Host-based MPI, leading to inefficient H2D&D2H data transfer, a GPU-aware MPI implementation can be extremely beneficial. With Unified Virtual Addressing (UVA), the host and all devices' memory are unified into a large (virtual) address space, improving communication efficiency [45]. In CUBENS, the MPI initialization is conducted in accordance with [42,45]. Each GPU is assigned an MPI rank (`rank`) along with a device identification number (`dev_id`). GPU-aware MPI can then be activated by using the `-D_GPU_DIRECT` compiler option. In Algorithm 4, an example of GPU-to-GPU data transfer is shown. The `host_data` clause is used to share the address of the device data to the host. The list of arrays, e.g., `use_device(variables_list)`, which are already present in the device memory, are directly passed to the traditional MPI routine `MPI_SENDRECV`. Note that in CUBENS, the transfer data – both sending (see `buff_send9_k`) and receiving (see `buff_recv9_k`) – contains a four-dimensional array, where 9 is the number of transferred variables and `k` is the streamwise direction with `nHalo` ghost cells.

5. Validation and additional case studies

In this section, the CUBENS solver is first validated using a set of benchmark cases. These include: (a) the laminar-to-turbulent transition of an ideal-gas decaying vortex, (b) the ideal-gas flat-plate H-type breakdown, (c) a turbulent boundary layer using the recycling-rescaling method, and (d) the linear small-amplitude disturbance evolution in a laminar boundary layer with an LMC fluid at supercritical pressure.

Subsequently, this section showcases the CUBENS solver's capability for the direct numerical simulations of strongly-stratified wall-bounded flows with compressible, single-phase, non-ideal fluids. Additional transitional flat-plate boundary layers at zero pressure gradient are considered: stably and unstably stratified (see Section 5.4), as well as a transcritical transitional boundary layer of an LMC fluid with strong property variations (see Section 5.5.2). It is important to note that all flow cases in this section employ 6th-order of accuracy for the convective fluxes and 4th-order accuracy for the diffusive fluxes. Overall, these simulations pave the way for future investigations with the CUBENS solver, with detailed analyses planned.

5.1. Compressible Taylor-Green vortex

The accuracy of the numerical scheme is assessed using the canonical Taylor-Green Vortex (TGV) problem. Originally formulated for incompressible flows by DeBonis [81], the TGV problem has been recently extended to strongly compressible regimes [82–84]. The flow is initialized in a triply periodic cubic domain with a side length L of 2π as

$$\begin{cases} \rho(x,y,z,t=0) = \rho_0, \\ u(x,y,z,t=0) = u_0 \sin(x)\cos(y)\cos(z), \\ v(x,y,z,t=0) = -u_0 \cos(x)\sin(y)\cos(z), \\ w(x,y,z,t=0) = 0, \\ p(x,y,z,t=0) = p_0 + \dfrac{\rho_0 u_0^2}{16}\left[\cos(2x)+\cos(2y)\right]\left[2+\cos(2z)\right], \end{cases} \tag{53}$$

where p_0 is computed using the non-dimensional ideal-gas law as $1/(\gamma M_0^2)$, and ρ_0 and u_0 are set to unity. The initial temperature field is derived from the ideal-gas law (see Section 2.2.1) using p_0 and ρ_0. The Reynolds number Re_L is set to 1600, and additional flow parameters are listed in Table A.6. Simulations are conducted on grids ranging from 64^3 to 1024^3 grid points, with a constant CFL number of 0.5, up to a dimensionless time of $t = 15$. For validation purposes, the kinetic energy K and enstrophy Ω are calculated as

$$K = \frac{1}{\mathcal{V}}\int_{\mathcal{V}} \rho u_i u_i \, d\mathcal{V}, \quad \Omega = \frac{1}{Re}\int_{\mathcal{V}} \mu \omega_i \omega_i \, d\mathcal{V}, \quad i = 1,2,3 \tag{54}$$

where $\omega_i = \epsilon_{ijk}\partial u_k/\partial x_j$ is the vorticity, and $\mathcal{V}$ is the computational domain.

Figure 3 shows K and Ω as a function of dimensionless time for three different Mach numbers. Results are compared with those from Sciacovelli et al. [83] on a $N = 1024^3$ mesh for all Mach numbers, with the reference solution (spectral method) reported in DeBonis [81] for $M_0 = 0.1$, and with Lusher and Sandham [84] on a $N = 512^3$ mesh for $M_0 = 1.0$. Note that for $M_0 = 1.0$, the results shown in Fig. 3(e,f) are obtained exclusively using the KEEP scheme, as the KEEP-PEP scheme leads to numerical instability. In contrast, for all other subsonic cases, both the KEEP and KEEP-PEP schemes produce identical results.

First, for $M_0 = 0.1$, the kinetic energy agrees very well with Sciacovelli et al. [83] and DeBonis [81], except on the coarsest grid of

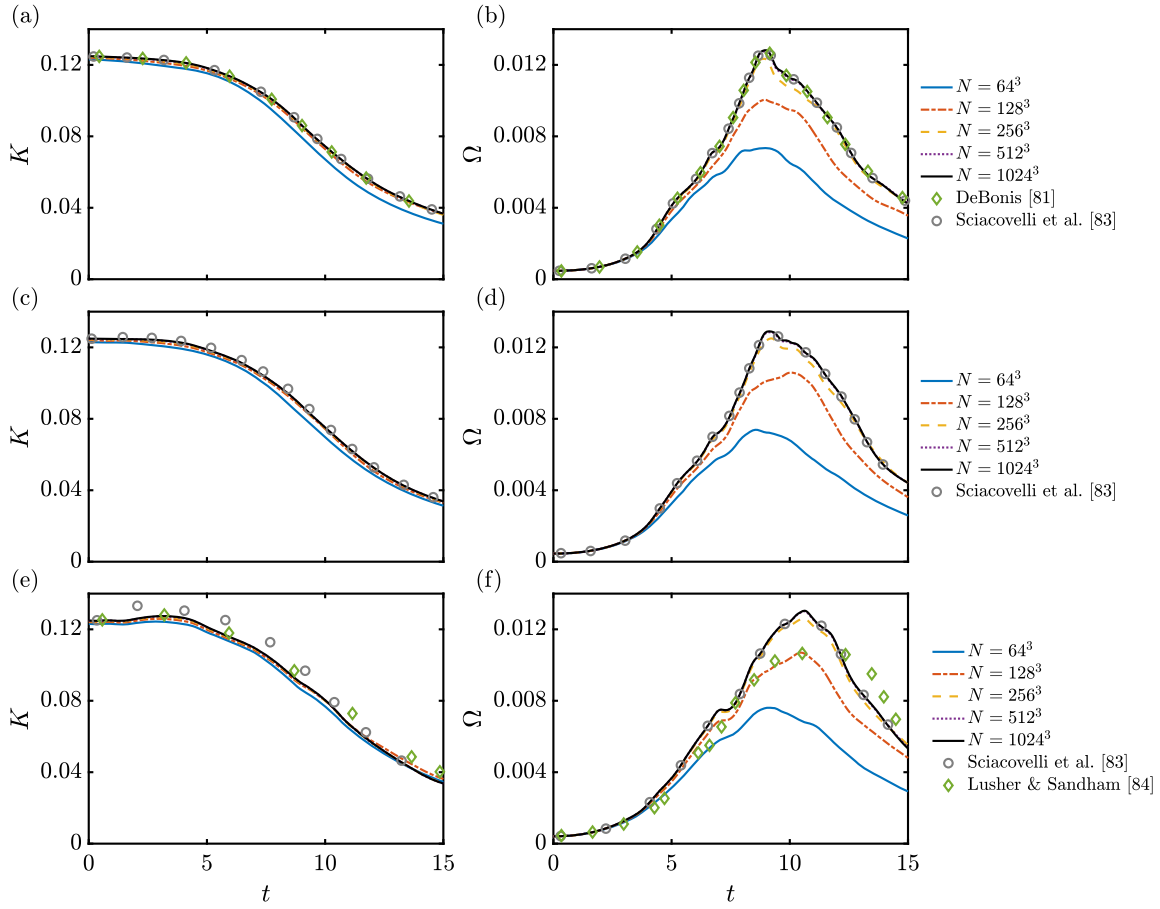

Fig. 3. Influence of grid resolution on the Taylor-Green Vortex problem: (a,c,e) kinetic energy K and (b,d,f) enstrophy Ω as a function of dimensionless time t. The reference Mach numbers are as follows: (a,b) $M_0 = 0.1$, (c,d) $M_0 = 0.5$, and (e,f) $M_0 = 1.0$. Grey circles represent the results from Sciacovelli et al. [83]. In (a,b), green diamonds correspond to the spectral results from DeBonis [81], and in (e,f) to Lusher and Sandham [84].

64^3 mesh points. Enstrophy, however, exhibits a stronger dependency on the grid resolution, showing remarkable convergence to the literature for meshes of $N = 512^3$ points and larger. Second, at higher Mach number ($M_0 > 0.5$), the effect of increasing pressure work becomes evident at early stages. While kinetic energy shows a slight increase in Fig. 3(c,e) due to internal-energy conversion, the growth of enstrophy in Fig. 3(d,f) is slower compared to Fig. 3(b) at $M_0 = 0.1$. For $M_0 = 1.0$, the peak Mach number reaches up to 1.95, leading to the formation of shocklets and supersonic flow regions (not shown here). It is worth noting that the KEEP and KEEP-PEP schemes implemented in the CUBENS solver were designed for shock-free compressible flows. To robustly and accurately resolve discontinuities, shock-capturing methods employing shock sensors and localized artificial diffusivity would be necessary [24]. However, this approach would compromise energy and pressure-equilibrium conservation, necessitating a balance between shock-free and shock-capturing methods.

In this regard, for $M_0 = 1.0$, the enstrophy computed by CUBENS matches well with Sciacovelli et al. [83], who used a high-order upwind scheme with high-order artificial dissipation, while the kinetic energy computed by CUBENS aligns with Lusher and Sandham [84], who applied a hybrid central non-oscillatory high-order scheme. However, discrepancies arise in the comparison of K against [83] and Ω against [84], especially at later times when the latter appears to be more dissipative than CUBENS. The variations in the value of K may originate from the significant contribution of the pressure-work term in the kinetic energy equation at higher Mach numbers. This non-conservative term, along with the pressure-dilatation term in the internal energy equation, ensures consistency in the pressure-diffusion term of the total energy equation, as outlined by Kuya et al. [24]. However, their discretization in non-conservative form results in significantly reduced numerical robustness compared to the KEEP scheme employed in CUBENS Kuya and Kawai [29]. Concerning the conservation of total entropy, inviscid TGV simulations from Kuya and Kawai [29] at a lower reference Mach number of 0.4 show that this quantity is well preserved and the density fluctuations converge to constant values, regardless of the pressure-work contribution. Ultimately, achieving stable solutions requires a trade-off between enforcing primary and secondary conservation variables, depending on the flow regime and flow physics [36].

5.2. Transitional boundary layer of an ideal gas

Here, CUBENS is tested for a three-dimensional flat-plate boundary layer with the classical subharmonic H-type breakdown investigated by Sayadi et al. [76]. To provide a visual representation of the results obtained with CUBENS, the instantaneous flow structures are displayed using isocontours of the Q-criterion in Fig. 4.

The dimensions of the computational domain are $0 \leq x/\delta_{99,0} \leq 515$ in the streamwise direction, $0 \leq x/\delta_{99,0} \leq 20$ in the wall-normal direction, and $0 \leq x/\delta_{99,0} \leq 9.63$ in the periodic spanwise direction. The inlet boundary-layer thickness $\delta_{99,0}$ is based on the inlet Reynolds number of $Re_{\delta,0} = \sqrt{10^5}$. Non-reflecting boundary conditions, along with numerical sponge zones, are applied at the inlet, top, and outflow boundaries, whereas the domain is periodic in the spanwise direction. At the wall, the no-slip and fully-reflective conditions are applied. In order to trigger transition, "controlled" disturbances are introduced at the disturbance strip between $x_1/\delta_{99,0} = 41.8$ and $x_2/\delta_{99,0} = 51.8$. The wall-normal velocity distribution in Eq. (49) is selected. It induces a

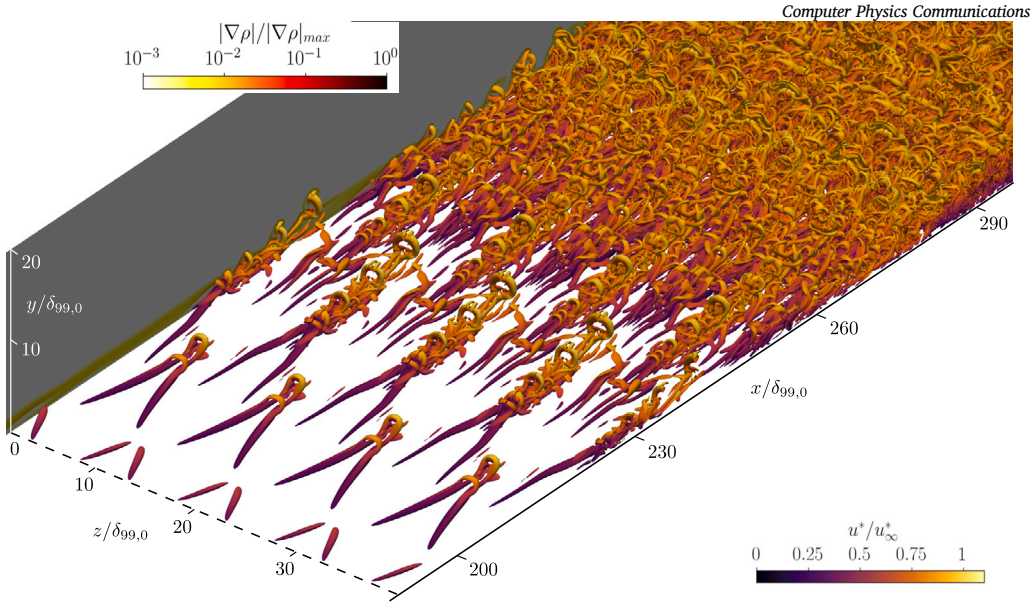

Fig. 4. Flat-plate boundary-layer H-type breakdown with CUBENS. Visualization of the instantaneous flow structures using isosurfaces of the Q-criterion ($Q = 0.015$), colored by the streamwise velocity. The side plane is colored by the normalized density gradient. Note that the domain is copied four times in the spanwise direction for better visualization.

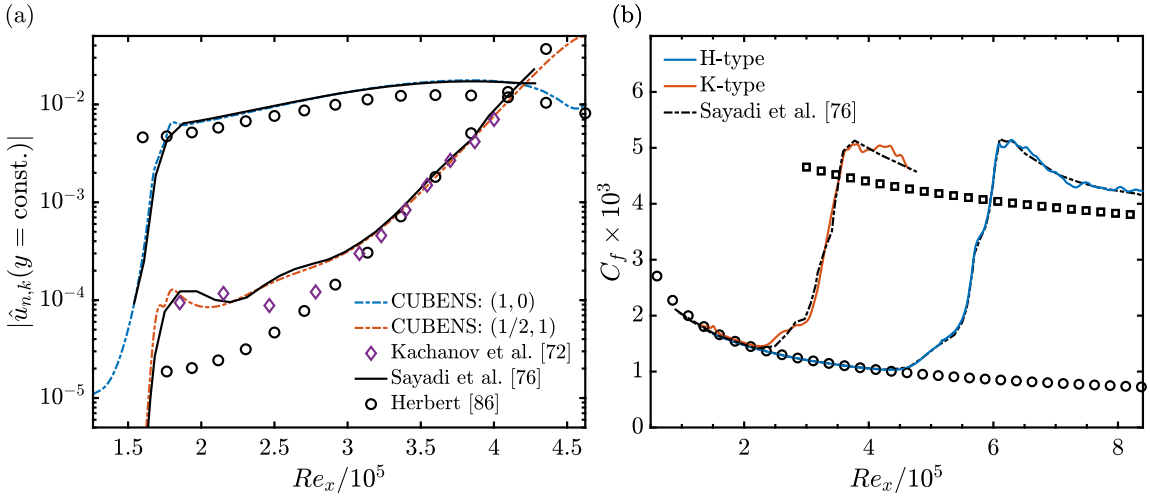

Fig. 5. H-type breakdown with CUBENS. (a) downstream development of the streamwise disturbance amplitude at constant $y/\delta_{99,0} = 0.26$ (purple diamonds from Kachanov et al. [72], black lines from Sayadi et al. [76], black circles from Herbert [86]); (b) time- and spanwise-averaged skin-friction coefficient (black dash-dotted line from Sayadi et al. [76], black circles from the laminar self-similar solution, black squares from turbulent correlation of White [87]). In (b), red line refers to the K-type breakdown of Sayadi et al. [76] simulated in CUBENS.

Tollmien-Schlichting (TS) wave ($A_{2\text{-D}} = 7.5 \times 10^{-3}$) at a reduced frequency of $F_{2\text{-D}} = F_0 = 124 \times 10^{-6}$, along with a pair of oblique waves ($A_{2\text{-D}} = 8.5 \times 10^{-5}$ with $\beta = \beta_0 = 0.65$) at half the frequency of the primary wave, i.e., $F_{3\text{-D}} = 62 \times 10^{-6}$. The computational domain is discretized with $N_x \times N_y \times N_z = 4000 \times 600 \times 150$ grid points. While an equidistant grid-point distribution is prescribed in the streamwise and spanwise directions, stretching in the wall-normal direction is applied using Eq. (40). The stretching factor σ is selected such that a Δy^+ (superscript + for wall units) of 0.59 in the first grid cell is obtained, normalized in wall units at $\max\{c_f\}$. For the other directions, the grid spacing in wall units is $\Delta x^+ = 10.0$ and $\Delta z^+ = 4.9$.

Once the DNS has reached a time-periodic solution, flow snapshots are extracted within two forcing periods and fast-Fourier transformed (FFT) in time and spanwise direction. Note that the spanwise FFT uses the FFTW package [85], and it is fully MPI-parallelized in the CUBENS solver. The Fourier components are denoted with (h, k), where h indicates a wave with frequency $h \cdot F_0$ and k denotes the spanwise wavenumber $k \cdot \beta_0$. In Fig. 5(a), the spatial development of the FFT streamwise velocity disturbance $|\hat{u}|$ at constant $y/\delta_{99,0} = 0.26$ is displayed. Excellent agreement between the CUBENS simulation and Sayadi et al. [76] is observed for both the primary and subharmonic growth at later stages. Only in the proximity of the disturbance strip, small quantitative differences are found for the oblique modes. This can be attributed to a slight variation in the receptivity mechanism at the disturbance strip caused by the different numerical scheme and setup. For the subharmonic disturbances, very good agreement is also achieved with the experiments of Kachanov et al. [72]. Below $Re_x < 3 \times 10^5$, the values obtained from secondary-instability theory (see Herbert [86]) diverge because the receptivity mechanism of the oblique waves is still active, and the primary-wave amplitude is not large enough to trigger the onset of secondary instability. For a more complete comparison of the H-type breakdown, the time- and spanwise-averaged distribution of the skin-friction coeffi-

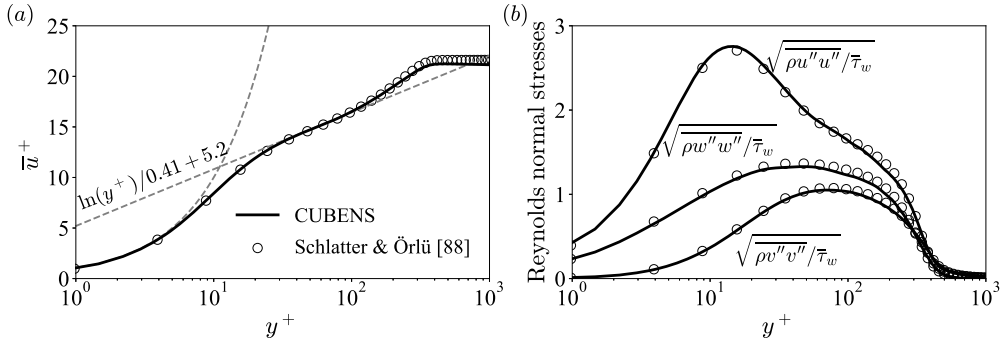

Fig. 6. Validation of the incompressible turbulent boundary layer simulations by Schlatter and Örlü [88] (black circles) with CUBENS. Over the dimensionless wall-normal coordinate y^+, (a) mean streamwise velocity profile $\bar{u}^+$ and (b) the three components of the Reynolds normal stress.

cient is plotted over the Reynolds number Re_x in Fig. 5(b). We observe good agreement with the results of Sayadi et al. [76], including for the K-type breakdown. While the transition location and the increase in C_f are very well matched, small misalignments are visible in the C_f-peak (overshoot). This could be due to a different time-averaging procedure. Note that transition to turbulence is an extremely non-linear process and is very sensitive to the numerical setup.

5.3. Turbulent boundary layer of an ideal gas

To validate the recycling-rescaling method in Section 3.3.3, we compare the incompressible turbulent boundary of Schlatter and Örlü [88] at $M_\infty = 0.2$. The flow parameters are listed in Table A.6. The inlet friction Reynolds number and momentum-thickness Reynolds number are $Re_{\tau,inl} = 112.9$ and $Re_{\theta,inl} = 234.0$, respectively. The grid spacings in wall units are $\Delta x^+ = \Delta z^+ = 4.4$ and $\Delta y^+ = 0.73$, which are comparable to those in [88]. The computational domain dimensions are $L_x = 144.0\,\delta_{99,inl}$, $L_y = 14.5\,\delta_{99,inl}$, and $L_z = 7.0\,\delta_{99,inl}$ in the streamwise, wall-normal, and spanwise directions, respectively. A sponge region of $14.5\,\delta_{99,inl}$ is placed at the outlet. The recycling position is set to $x_{rcy} = 112.3\,\delta_{99,inl}$. Note that $\delta_{99,inl}$ is the mean turbulent boundary-layer thickness at the inlet.

Fig. 6 illustrates the profiles of the time- and spanwise-averaged turbulent statistics obtained at $x = 123.9\,\delta_{99,inl}$ over the dimensionless wall-normal coordinate $y^+ = y/\delta_v$, with $\delta_v = \nu_w/u_\tau$ and $u_\tau = \sqrt{\tau_w/\rho_w}$. The CUBENS solver successfully replicates the mean streamwise velocity profile and Reynolds stresses in agreement with Schlatter and Örlü [88].

It is worth mentioning that the current recycling-rescaling method fails when applied to a turbulent boundary layer of a non-ideal compressible fluid with heat transfer. Employing the classical van Driest's velocity transformation in Eq. (43) can lead to an inaccurate prediction of the velocity profile under strong property variations near the wall, and particularly when crossing the Widom line ("pseudo-boiling phenomena"), as noted by Bai et al. [89]. A recent successful attempt to go beyond van Driest's approach in non-ideal fluid flows was proposed by Hasan et al. [90] in channel flow at supercritical pressure. However, in transcritical (pseudo-boiling) boundary layers, large mean-density ratios and density fluctuations can induce a near-wall convective flux in the stress balance equation, making the new scaling law transformation inaccurate. Overall, the generation of inflow turbulent profiles for strongly non-ideal flows remains an open research question.

5.4. Stably and unstably stratified boundary layer of an ideal gas

This section validates the wall-normal buoyancy force in the source term of Eq. (6) and in the boundary conditions of Eq. (47). As a test case, we consider the controlled laminar-to-turbulent transition of a flat-plate boundary layer with positive (unstably stratified) and negative (stably stratified) buoyancy force. As in Section 5.2, the H-type breakdown is selected.

To exclusively investigate the buoyancy effect on the transition to turbulence, we opt for the ideal-gas assumption. While the temperature at the wall is fixed, the sign of the gravity acceleration g^* is modified to obtain a positive or negative Richardson number in Eq. (8). In this way, the buoyancy force is either directed against (unstable stratification, positive gravity) or aligned with the temperature gradient (stable stratification, negative gravity). The flow parameters are shown in Table A.6, which also includes a reference neutrally buoyant case, i.e., $Ri_{w,0} = 0$. Similar boundary conditions to those in Section 5.2 are applied. The dimensions of the computational domain are $0 \le x/\delta_{99,0} \le 541$, $0 \le y/\delta_{99,0} \le 20$, and $0 \le z/\delta_{99,0} \le 9.63$. The disturbance strip, as described in Eq. (49), is located between $x_1/\delta_{99,0} = 21.4$ and $x_2/\delta_{99,0} = 31.4$. For this comparison, we choose the perturbation frequency and amplitude of the fundamental mode $(1,0)$ to match the growth of the primary instability across all three cases. Note that before activating the disturbance strip, a converged laminar solution is obtained, as the boundary layer departs from self-similarity when a buoyancy force is present. The computational domain is discretized with $N_x \times N_y \times N_z = 7420 \times 390 \times 150$ grid points, along with wall-normal stretching ($\Delta y^+ = 0.44$ at the wall). For the other directions, the grid spacing in wall units is $\Delta x^+ = \Delta z^+ = 4.9$.

In Fig. 7(a), the spatial development of the FFT streamwise velocity disturbance $|\hat{u}|$ at constant $y/\delta_{99,0} = 0.26$ is displayed. Note that, recalling Eq. (8), the unit Richardson number is proportional to the local Blasius length scale δ^*. This implies that when moving downstream, the Ri_{unit} increases according to

$$Ri_{unit} = Ri_{unit,0}\frac{Re_\delta}{Re_{\delta,0}}, \tag{55}$$

where $Ri_{unit,0}$ is the inlet Richardson number based on the boundary-layer thickness $\delta_{99,0}$. The same scaling applies to the wall Richardson number, defined as $Ri_w = Ri_{unit}(\rho_w - 1)$, where ρ_w is the density at the wall. Additionally, the growth rate according to the in-house LST solver is included to validate the linear growth of the fundamental mode. In all three scenarios, a good quantitative agreement between DNS and LST is observed for the primary growth. With regard to the subharmonic mode $(1/2,1)$, the unstable stratification enhances the secondary instability, shifting the transition location farther upstream; see Fig. 8. Here, snapshots of the streamwise velocity inside the boundary layer, i.e., x-z plane at a constant wall-normal height of $y/\delta_{99,0} = 0.4$, are presented. Staggered patterns of Λ-vortices, characteristic of the H-type transition, are visible. The transition location shifts considerably downstream as $Ri_w > 0$ (negative gravity).

5.5. Transitional boundary layer of a fluid at supercritical pressure

5.5.1. Boundary-layer instability

The influence of strongly non-linear thermodynamics on the stability of a two-dimensional zero pressure gradient flat-plate boundary

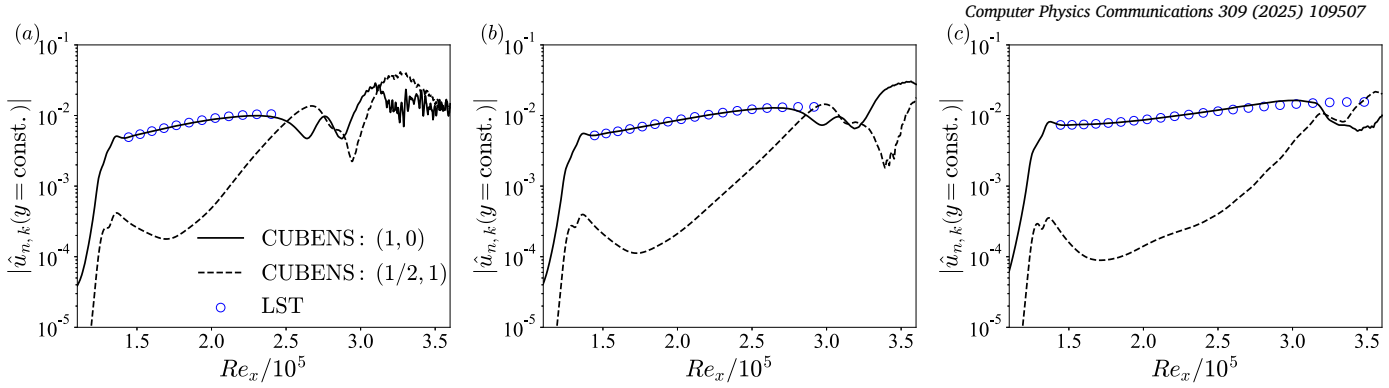

Fig. 7. Downstream development of the $|u'|$-disturbance amplitude at constant $y/\delta_{99,0} = 0.26$ for modes (1,0) and (1/2,1) in (a) unstably-stratified ($Ri_{w,0} = -0.01$), (b) neutrally-buoyant ($Ri_{w,0} = 0$), and (c) stably-stratified ($Ri_{w,0} = 0.01$) boundary layer. The wall Richardson number is defined as $Ri_w = Ri_{unit}(\rho_w - 1)$. The LST solution is represented by blue circles.

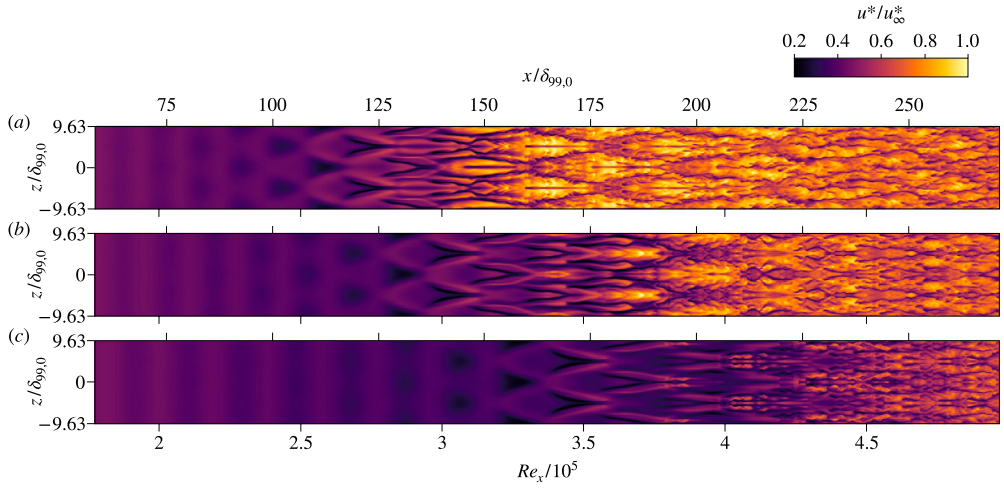

Fig. 8. Contours of instantaneous streamwise velocity (x-z plane at $y/\delta_{99,0} = 0.4$) of (a) unstably-stratified ($Ri_{w,0} = -0.01$), (b) neutrally-buoyant ($Ri_{w,0} = 0$), and (c) stably-stratified ($Ri_{w,0} = 0.01$), boundary layer. The wall Richardson number is defined as $Ri_w = Ri_{unit}(\rho_w - 1)$. Note that the results are copied once in the spanwise direction for better visualization.

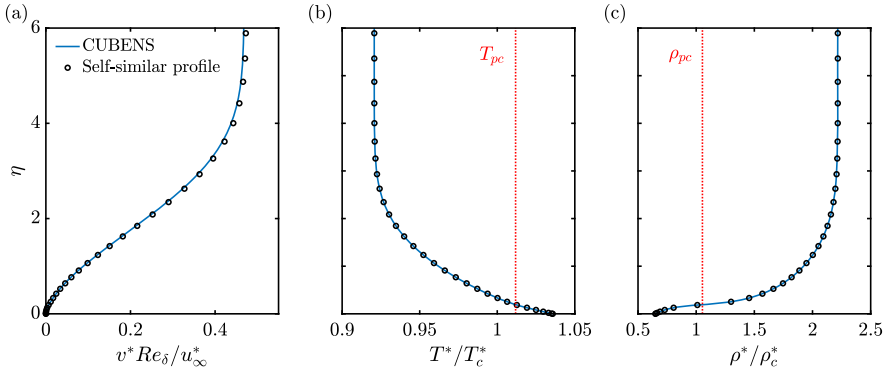

Fig. 9. Laminar solution with CUBENS (blue lines) for (a) wall-normal velocity, (b) temperature, and (c) density profile. The self-similar solution is marked with black circles. In (b,c), the pseudo-boiling temperature and density are indicated in red, respectively.

layer is investigated. To achieve the strongest departure from the ideal-gas behavior, we consider an LMC fluid at supercritical pressure that is heated such that the boundary layer crosses the Widom line (transcritical regime). The Peng-Robinson equation of state with Chung's model for the transport properties is selected.

The flow parameters are reported in Table A.7. This setup corresponds to supercritical CO_2 at $T^*_\infty = 280\,\text{K}$ (liquid-like free stream) and pressure $p^* = 80\,\text{bar}$, while the wall is heated into the vapor-like regime. The inflow Reynolds number is $Re_{\delta,0} = 700$, with a streamwise domain extension of $x/\delta_{99,0} = 575$. Comparisons between the fully developed DNS solution and the self-similar solution of the compressible boundary-layer equations (see [7,8]) are provided in Fig. 9. Significant consistency with the self-similar profiles is observed, especially for the temperature, density, and streamwise velocity (not shown here). However, a minimal deviation of about 1% in the wall-normal velocity is noted, as the assumption of a perfectly constant pressure in the wall-normal direction is not valid in the supercritical fluid region near the critical point (not shown here). Therefore, we conclude that a minor non-zero wall-normal

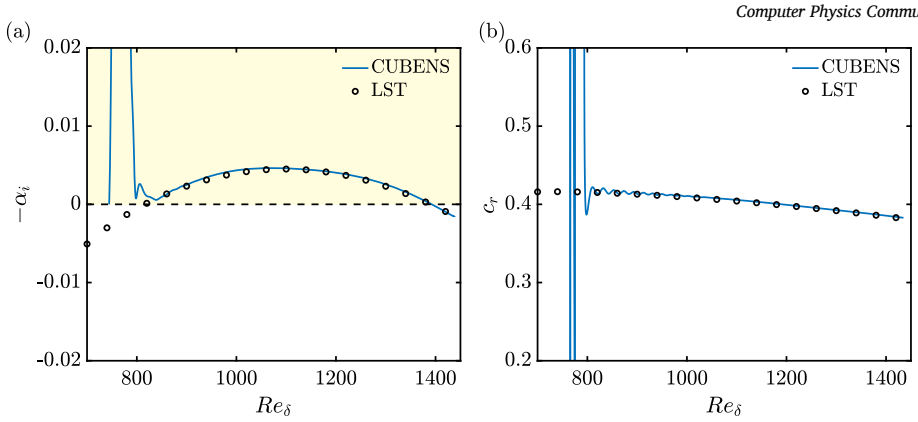

Fig. 10. Boundary-layer instability in the transcritical regime with CUBENS (blue lines): (a) growth rate $-\alpha_i$, (b) phase speed c_r as a function of the local Reynolds number Re_δ. The LST solution is marked with black circles.

pressure gradient, which is neglected in the canonical self-similar approximation, remains in the numerical integration of the conservation equations, resulting in a minimal deviation in the wall-normal velocity profile.

In order to compare the linear disturbance evolution with the in-house linear-stability theory (LST) solver (see [7,8]), the 2-D disturbance strip in Eq. (49) is activated at $x_{mid}/\delta_{99,0} = 38$ with $\beta = 0$ and a small linear forcing amplitude of about $A_{2\text{-D}} = 10^{-5} M_\infty$. A non-dimensional frequency of $F = 80 \times 10^{-6}$ is selected according to the LST results. Once the simulation has reached a time-periodic solution, a similar procedure as in Section 5.2 is performed. The results are Fourier transformed in time with the fundamental frequency of $\Omega = \omega_{2\text{-D}}/2$, yielding in complex Fourier coefficients $\hat{q}_j$ for a given harmonic j and flow variable q. For the sake of comparison with the LST reference solution, the normalized disturbance growth rate and phase speed of the first harmonic are calculated as

$$\alpha_i(x) = -\frac{Re_\delta}{Re_{\delta,0}} \frac{1}{\hat{q}_1^{max}} \frac{\partial \hat{q}_1^{max}}{\partial x}, \quad c_r(x) = \frac{\omega Re_{\delta,0}}{Re_\delta} \left(\frac{\partial \hat{\phi}_1}{\partial x} \right)^{-1}, \tag{56}$$

with $\hat{q}_1^{max}(x) = \max\{|\hat{q}_1(x = const., y)|\}$ being the wall-normal maximum amplitude of the fundamental harmonic $\hat{q}_1$ and $\hat{\phi}_1 = \arg(\hat{q}_1)$ being the phase angle. Given the transcritical boundary-layer profiles in Fig. 9, the streamwise evolutions of the growth rate and phase speed are displayed in Fig. 10. In this case, it is Mode I of Tollmien-Schlichting-type, the primary unstable mode. Conversely, if the wall temperature is further increased, as presented in Boldini et al. [23], the only modal instability present in the boundary layer is caused by the transcritical Mode II (see e.g., [7] for more information on Mode II). Overall, good agreement with the reference results is observed for both considered quantities. The minimal deviations in the growth rate are attributed to non-parallel effects, which are not included in the LST framework.

5.5.2. H-type breakdown

Next, the DNS of a transcritical transitional boundary-layer is computed using CUBENS. Analogous to the ideal-gas case in Section 5.2, the controlled H-type breakdown is selected. The base-flow parameters are reported in Table A.7. Note that the Van der Waals EoS is chosen with transport properties according to the JST model. The free-stream compressibility factor is equal to $Z_\infty = 0.254$. The laminar boundary layer, which also serves as DNS initial condition, exhibits similar profiles to those in the transcritical case of Section 5.5.1. A large wall-normal density stratification, i.e., $\rho_w^*/\rho_\infty^* = 0.328$, is noticeable. Yet, a wall-normal buoyancy force is neglected for this particular simulation. After performing a linear stability analysis of the laminar profiles, the DNS is set up such that a modal instability is triggered at the same primary frequency $\omega_{2\text{-D}}$ of the ideal-gas case in Section 5.2. Therefore, in a disturbance strip at $Re_{\delta,mid} = 415$, the two-dimensional wave $(1,0)$ and a pair of oblique subharmonic waves $(1/2, \pm 1)$ with $\omega_{3\text{-D}} = 0.5\omega_{2\text{-D}}$ and $A_{2\text{-D}}/A_{3\text{-D}} = 88$ are introduced. Note that the primary instability here is caused by the transcritical Mode II (see [7,8,52]). The computational setup is analogous to the one described in Section 5.2, except for the following aspects: the streamwise and wall-normal lengths are $0 \le x/\delta_{99,0} \le 668$ and $0 \le y/\delta_{99,0} \le 40$, respectively. A grid discretization of $N_x \times N_y \times N_z = 10240 \times 700 \times 160$ is obtained. The stretching factor σ of Eq. (40) is selected such that a Δy^+ of 0.89 in the first grid cell is obtained (normalization in wall units at the turbulent region domain exit). For the other directions, the grid spacing in wall units is $\Delta x^+ = 5.2$ and $\Delta z^+ = 4.8$. Instantaneous flow structures are displayed in Fig. 11 using isocontours of the Q-criterion. Clearly, developed staggered patterns of Λ-structures (location A) are absent, in contrast to the ideal-gas case in Fig. 4. Instead, secondary vortex systems (location B) are observed at the valley planes of the primary Λ-vortices, half a spanwise length apart. Here, high-low-speed streaks with high-low-density fluid are induced. Compared to the H-type breakdown of Fig. 4, the transition to turbulence here is much more violent, with a stronger structure breakdown.

Overall, a detailed analysis of the flow variables (not shown) reveals an oscillation-free flow field and a stable transition location, achieved without the need for any artificial diffusion, dissipation, or filtering.

6. Solver performance

In this section, the performance of the CUBENS solver is assessed by conducting strong and weak scaling tests for the Taylor-Green Vortex presented in Section 5.1. The tests are carried out on both Snellius and LUMI. On Snellius, a single CPU node is equipped with 128 AMD EPYC 7H12 CPUs, while a GPU node consists of 4 NVIDIA A100 GPUs paired with 72 Intel Xeon Platinum 8360Y 36C 2.4GHz processors. On LUMI, a single CPU node contains 128 AMD EPYC 7763 CPUs, while a GPU node consists of 4 AMD MI250X GPUs, each featuring two Graphics Compute Dies (GCDs), along with a 64-core AMD EPYC 7A53 CPU. All tests run warm-up 100 time steps on both computing units for each architecture, before calculating an average wall time per step (every 10 steps). Data I/O to disk is discarded.

In the following, we analyze the performance of a single CPU and a GPU. This choice is motivated by: (a) the memory limits of the graphic cards employed in these supercomputing architectures, and (b) the need to understand the relationship between memory occupancy and performance speedup for both partition units. These results serve as the baseline for extending the computational approach to a multi-GPU setup.

6.1. Single CPU and single GPU performance

In this analysis, we examine the wall-clock time per step of a single CPU core (Snellius: 1 AMD EPYC 7H12, LUMI: 1 AMD EPYC 7763) and

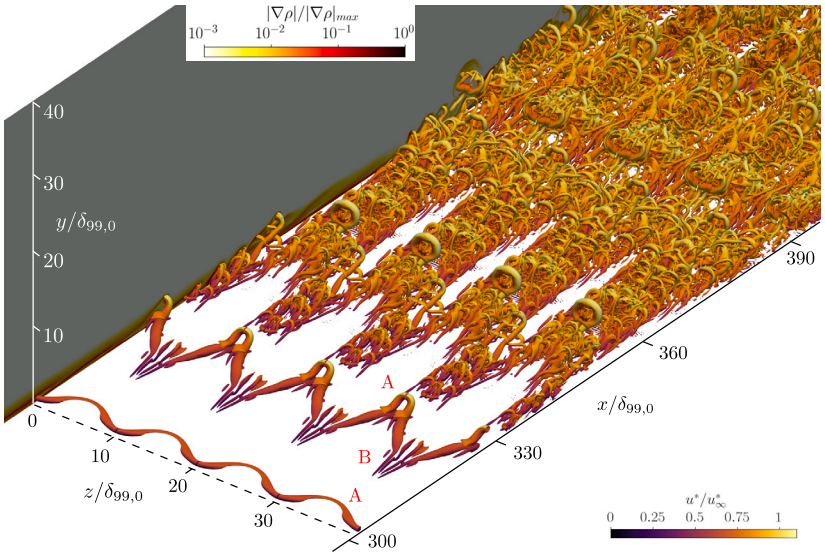

Fig. 11. Transcritical flat-plate boundary layer H-type breakdown with CUBENS. Visualization of the instantaneous flow structures using isosurfaces of the Q-criterion ($Q = 0.025$), colored by the streamwise velocity. The side plane is colored by the normalized density gradient. Note that the domain is copied four times in the spanwise direction for better visualization.

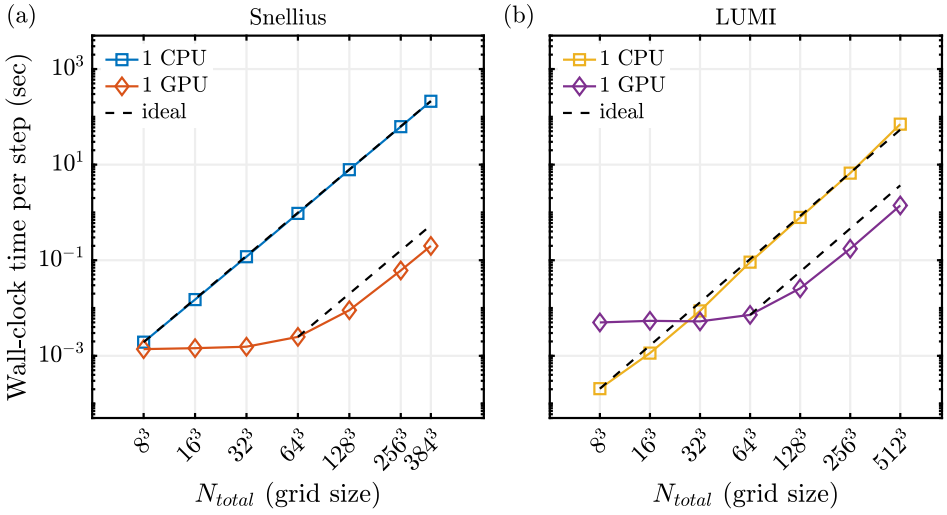

Fig. 12. Performance of CUBENS on a single CPU and GPU. Wall-clock time per step in seconds over grid size N_{total} for the TGV problem: (a) Snellius, (b) LUMI. The dashed black lines refer to the ideal solutions.

a single GPU (Snellius: 1 NVIDIA A100, LUMI: 1 AMD MI250X GPU), as the grid size increases by a factor of 8. The performance results in Fig. 12 are presented without intra- and inter-node communication, as MPI communication between multiple computing units and nodes could bias the results. On both systems, the increase in CPU wall-clock time is proportional to the increase in grid size, whereas, up to $N = 64^3$, the GPU remains underutilized, resulting in only a slight increase in execution time per time step. However, once the GPU is sufficiently utilized ($N > 64^3$), the wall-clock time increases proportionally to the eightfold increase in grid size. The calculations are constrained by the available memory, which is 40 GB for the NVIDIA A100 GPUs and 64 GB for AMD MI250X. Therefore, CUBENS can run on a single GPU up to a grid size of $N = 384^3$ on Snellius and up to $N = 512^3$ on LUMI. Notably, one LUMI CPU is approximately 10 times faster than one Snellius CPU (33.2M vs. 3.8M transistors). Simultaneously, we observe a performance drop for CUBENS on one AMD MI250X compared to one NVIDIA A100 GPU, with a slowdown factor of 2.9× for the benchmark simulation scenario with 50 million grid points presented in Section 5.1.

Given these results, it is crucial to understand which parts of the code consume the most time in order to optimize the multi-GPU implementation.

The workload distribution is shown in percentage for a single EPYC 7H12 CPU in Fig. 13(a) and for a single A100 NVIDIA GPU in Fig. 13(b). Note that a similar code profiling is observed for the corresponding partitions on the LUMI cluster. The case considered is the TGV problem at $M_0 = 0.1$ with $N = 256^3$ grid points, and the main relevant sections inside the time loop are tracked using the `MPI_WTIME` function on the CPU and the `-D_USE_NVTX` flag on the GPU. Note that the wall-clock time per step for a single GPU is $t = 62.3$ sec, while for a single GPU it is $t = 0.06$ sec.

For both computing units, the calculation of the inviscid (Euler) fluxes (see Eq. (3)) and viscous fluxes (see Eq. (4)) are the most time-consuming operations, accounting for over 70% of the total workload across the three Runge-Kutta substeps. Copying data into the ghost cells, which corresponds to exchanging data at the periodic boundary conditions in the TGV problem, is negligible for a single partition, regardless

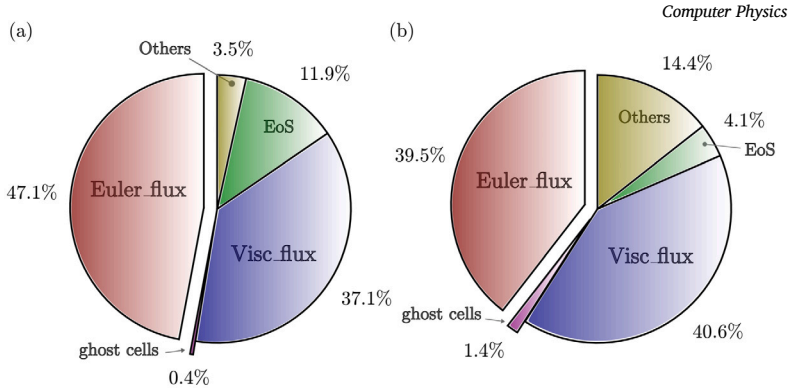

Fig. 13. Code profiling on Snellius for (a) single CPU and (b) single GPU with $N = 256^3$ grid points. Workload distribution of: Euler fluxes (`Euler_flux`, Eq. (3)), viscous fluxes (`Visc_flux`, Eq. (4)), updating ghost cells (included in `MPI_comm` module), equation of state and transport properties models (`EoS`), and other calculations (`Others`).

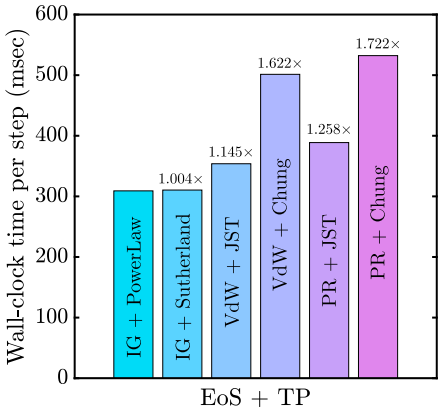

Fig. 14. Impact of thermodynamics (EoS) and transport properties (TP) on the wall-clock time per step for the TGV problem with $N = 256^3$ grid points. The different EoS and TP models are reported in Sections 2.2 and 2.3.

of the hardware architecture (module `set_BC` in Algorithm 1 does not apply here).

The computational time required to calculate both caloric and thermal EoS, along with transport properties (TP), as shown in Fig. 13, is relatively small, accounting for approximately 12% of the total time. Notably, the impact of thermodynamics calculations on the total workload decreases when using a single GPU.

With respect to the GPU architecture used in Fig. 13(b), we further investigate the kernel-level performance of the two most computationally expensive kernels – Euler and viscous fluxes – through a roofline analysis (not shown here). This preliminary performance analysis revealed that both kernels are compute-bound, indicating high computational efficiency relative to data movement. Moreover, the Euler- and viscous-fluxes kernels achieve 62.5% and 64.7% of the GPU's FP64 FLOPS peak performance, respectively.

As the previous profiling was performed using the most computationally efficient ideal-gas equation of state, i.e., power law, it is relevant to assess how a more complex thermodynamics affect the computational time. Therefore, an investigation into the effect of different EoS and TP models on the wall-clock time per step is conducted using the Taylor-Green-Vortex benchmark case as in Fig. 13, while varying the EoS and TP models and adjusting thermodynamic conditions accordingly (e.g., reduced temperature and pressure are $T_r = 1.50$ and $p_r = 1.50$, respectively, for non-ideal thermodynamic states). The total number of grid points is $N = 256^3$. The results are reported in Fig. 14, with the reference case being the IG EoS and a power law for viscosity. As expected, the largest slow down of 1.72× is attributed to the most detailed combination of EoS and TP, i.e., the PR EoS with Chung's model. The latter is slower than the PR EoS with the JST model for transport properties by approximately 40%.

6.2. Multi-GPU performance on different GPU architectures

The parallel performance of CUBENS is assessed through strong- and weak-scaling tests. As the number of GPU partitions increases, the communication-to-computation ratio rises from 9% (4 GPUs) to 70% (16 GPUs). This ratio further increases to 79% when using the traditional Host-based MPI, due to greater latency and overhead with larger datasets. Thus, the use of GPU-aware MPI (see Section 4.3), combined with the asynchronous treatment of the Euler fluxes (see Section 4.2), is preferred for the strong- and weak-scaling analysis. Fig. 15(a) reports the strong scaling of CUBENS. It is important to note that the GPU architecture available on the Snellius cluster is limits usage to a maximum of 64 GPUs. While the number of GPUs is doubled in each step, two different grid sizes are selected for each supercomputing architecture. The first corresponds to the grid size required for full memory occupancy ($N = 384^3$ on Snellius and $N = 512^3$ on LUMI), and the second targets high-fidelity simulations that resolve both the Kolmogorov and Batchelor scales ($N = 1024^3$, or approximately 5 billion degrees of freedom).

For the Snellius cluster, excellent scaling is achieved with $N = 384^3$ points up to 4 GPUs. However, when 8 GPUs are used, a considerable performance slowdown occurs, resulting in approximately the same wall-clock time per step as with 4 GPUs. This slowdown corresponds to the transition from a single-node to a multi-node configuration. In the Snellius cluster's inter-node communication, the InfiniBand HDR100 bandwidth is limited by the Peripheral Component Interconnect Express (PCIe), with approximately 12.5 GB/s bandwidth per GPU. This inevitable performance drop is not unique to the current supercomputer architecture, but can be even more severe for algorithms with more complex (i.e., collective) communication patterns; see Romero et al. [91]. In contrast, no slowdown is observed on LUMI's GPU partitions, thanks to the high memory bandwidth of the Slingshot-11 network, providing 50 GB/s bidirectional bandwidth per GPU, along with double the intra-node communication (Infinity Fabric) compared to the Snellius cluster. As a result, a speedup of 7.2× is achieved for $N = 512^3$ grid points, scaling from 1 to 8 LUMI GPUs.

With 8 GPUs (2 nodes) on Snellius and 16 GPUs (2 nodes) on LUMI, full memory occupancy per GPU is reached with a larger grid size of $N = 1024^3$. On Snellius, increasing the compute nodes eightfold, from 2 to 16, results in a strong-scaling speedup of about 3.9. On LUMI, increasing the computer nodes by a factor 64, from 2 to 128, yields a strong-scaling speedup of about 33. Similar strong-scaling efficiencies, around 50%, were also observed in other open-source DNS solvers for wall-bounded flow [43–45]. Comparing Host-based MPI data transfer to GPU-aware MPI, the latter reveals a speedup on Snellius between 2× (2 GPUs with 384^3 grid points) and 2.5× (64 GPUs with 1024^3 grid

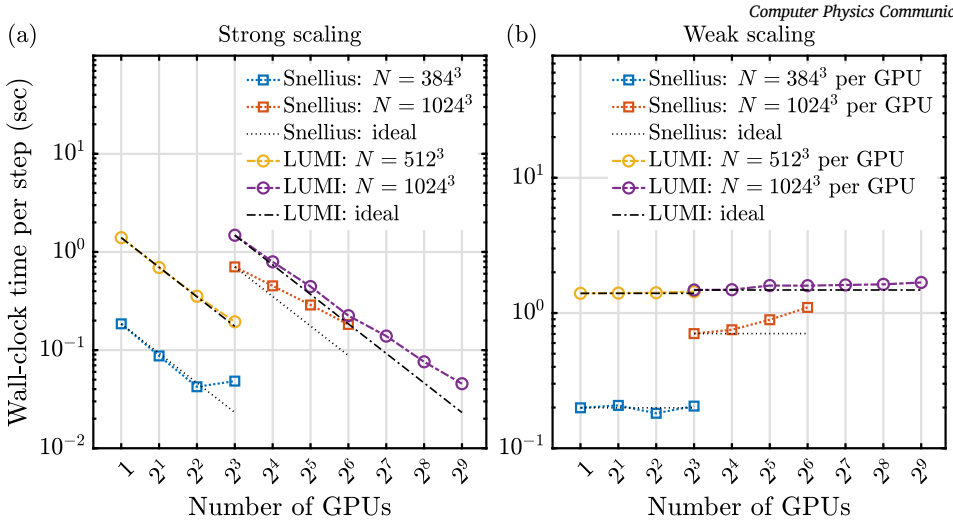

Fig. 15. Wall-clock time per step in seconds for (a) strong scaling and (b) weak scaling using NVIDIA A100 GPUs on Snellius and AMD MI250X GPUs on LUMI. The dashed black line represents the ideal scaling. Note the different y-axis range between (a) and (b).

Table 3
Energy consumption on Snellius for CPU-based and GPU-accelerated architectures with $N = 256^3$ grid points. The acronyms CU and AWF stand for computing unit and accounting weight factor, respectively.

CU	Number of cores	Snellius AWF	Wall-clock time per step	Power per core	Energy
CPU	128	1	0.71 sec	4.46 W	5.43 MJ (1.51 kWh)
GPU	1	128	0.013 sec	98.3 W	68 kJ (0.019 kWh)

points), respectively. On LUMI, the speedup ranges from 1.1× (2 GPUs with 512^3 grid points) to 1.7× (64 GPUs with 1024^3 grid points). The asynchronous treatment of the inviscid fluxes becomes more advantageous as the number of nodes increases, e.g., a speedup of 1.1× with 2 nodes compared to 1.3× with 16 nodes on both architectures.

If we only consider a CPU-based approach, we can compare the performance drop to the GPU-accelerated version of CUBENS, given the same computational cost in Standard Billing Units (SBUs). For example, on the Snellius cluster, with a total of 8192 CPUs (64 CPU nodes), the wall-clock time per step is $\Delta t = 24.18$ sec for $N = 1024^3$ grid points. Compared to the value of 0.183 sec achieved with 16 GPU nodes, the computational speedup of the GPU-accelerated system over the CPU-based system is approximately 130×.

Fig. 15(b) shows the weak scaling wall-clock time per step, calculated by fixing the grid points per GPU to 384^3 and 1024^3 on Snellius, and to 512^3 and 1024^3 on LUMI. The number of GPU domains is progressively doubled in the respective grid direction, ensuring that the ratio between grid points in one direction and GPU partitions remains constant. A stable weak-scaling efficiency of around 1 is observed for the smaller number of grid points per GPU ($N = 384^3$ and $N = 512^3$), regardless of the computing architecture. However, for larger grid sizes ($N = 1024^3$), increasing inter-node communication significantly impacts weak scaling on the Snellius architecture, with the efficiency dropping to 0.64 from 8 to 64 GPUs. On LUMI, the weak-scaling analysis for $N = 1024^3$ grid points per GPU yields impressive results, achieving an efficiency of 0.88 from 16 to 512 GPUs. This result is comparable to the weak-scaling efficiencies reported in [43,45], but with more than 30 times the number of grid points per GPU. Notably, with 2^3 GPUs, LUMI achieves the same performance for $N = 512^3$ and $N = 1024^3$ grid points, once again highlighting LUMI's superior memory capacity and bandwidth compared to the Snellius cluster.

6.3. Energy consumption of CUBENS

Finally, we relate the performance improvements of the GPU-accelerated version of CUBENS, discussed in Sections 6.1 and 6.2, to the power consumption of the simulations. In fact, greater emphasis is need on enhancing the energy efficiency of high-fidelity simulations to reduce their carbon footprint [92]. To investigate this, the TGV problem from Section 5.1 with $N = 256^3$ is simulated on the Snellius architecture using both CPUs and GPUs until $t = 15$. For a fair comparison, we consider the same computational price of 1 SBU, i.e., 128 CPUs vs. 1 GPU. Energy management and consumption are monitored by the Energy Aware Runtime (EAR) library. By enabling the flag `-ear=on`, power usage of the active computing cores is tracked throughout the runtime. The results are summarized in Table 3. The reported power output accounts for all power usage by the motherboard (computing unit, memory, network, etc.), but GPU cooling process, which typically exceed those of CPUs, are not captured by the EAR library. Regarding energy usage, CUBENS consumed 1.51 kWh to advance the solution on 128 CPUs, compared to only 0.019 kWh on a single GPU. In other words, the energy consumption on a single graphics processing unit is 80× more energy efficient than on traditional processors, for the same computational cost. Note that if input/output operations were active, the GPU energy consumption would likely be higher due to the continuous, time-consuming, D2H data transfer.

7. Conclusions

We have introduced the open-source DNS code, CUBic Equation of state Navier-Stokes (CUBENS). The solver is GPU-accelerated for pre-exascale supercomputing architectures, employs high-order central finite-difference schemes, ensures kinetic energy and entropy conservation, and accounts for the strongly non-linear behavior of thermodynamic and fluid properties. CUBENS is designed to address both fundamental research and industrial applications operating under non-ideal gas conditions.

In addition to the canonical single-phase, non-reacting compressible Navier-Stokes equations, the solver incorporates a wall-normal buoyant force relevant in strongly stratified wall-bounded flows. Special attention is given to the implementation of non-ideal thermodynamic laws, including cubic equations of state and modified transport-property mod-

els. CUBENS utilizes high-order explicit central finite-difference schemes for both convective and diffusive fluxes. In particular, its KEEP split convective form discretization is non-dissipative, enhances numerical stability with minimized aliasing errors, and conserves kinetic energy and entropy. This approach enables stable direct numerical simulations of shock-free, high-Re flows with strong density variations. Additionally, the pressure-equilibrium-preserving (PEP) property is ensured under linear approximation for non-ideal fluids, providing further robustness under transcritical conditions.

CUBENS offers flexibility in simulating different canonical flow geometries, such as (a) transitional boundary layers with controlled disturbance excitation, (b) turbulent boundary layers using the recycling and rescaling method, and (c) channel flows. Non-reflecting inviscid boundary conditions with viscous numerical sponges are adapted for non-ideal flow conditions, also accounting for a wall-normal buoyant force. The solver has been successfully tested against various benchmark cases, including the compressible Taylor-Green Vortex, H-type transition to turbulence in a compressible boundary layer, ideal-gas turbulent boundary layer, stable and unstable stratified transitional boundary layer, and a laminar boundary layer with a non-ideal fluid at supercritical pressure and its linear instability.

Considerable attention is dedicated to accelerating the solver for massively parallel simulations, targeting the next generation of Exascale supercomputers. CUBENS supports both NVIDIA and AMD GPU architectures, with the OpenACC directive-based method chosen for architecture interoperability and ease of code extension. Further speedup is achieved when using multiple GPUs by (a) overlapping inviscid flux calculations at interior cells with MPI communication at boundary cells, and (b) employing GPU-aware MPI communication to avoid data transfer to the host CPU. For example, with 64 NVIDIA A100 GPUs compared to 8192 CPUs at the same computational price, a speedup of approximately 130× is achieved. Notably, the strong and weak scaling performance has been outstanding, especially on AMD-based supercomputing architecture. For instance, in a multi-GPU-node configuration ranging from 8 to 512 GPUs, the strong scaling speedup reaches a factor of 33, while a weak scaling efficiency of 0.88 is achieved for $N = 1024^3$ grid points per GPU.

The capability of CUBENS to simulate computationally expensive transitional and turbulent boundary layers in non-ideal fluid flows under stratified conditions has been thoroughly investigated and successfully demonstrated in Section 5. This represents a significant advancement in understanding the impact of complex thermodynamic effects on the dynamics of wall-bounded flows.

Significant improvements and expansions of CUBENS are currently in progress. Key areas of development include: (a) addressing the supersonic and hypersonic flow regime with appropriate shock-capturing techniques, (b) implementing curvilinear coordinates to handle more realistic geometries found in turbomachinery, and (c) developing a new recycling and rescaling method for diabatic boundary layers with strongly non-ideal fluids. Additionally, achieving exact discrete conservation of the PEP property for arbitrary equations of state would allow for the extension of CUBENS to high-fidelity simulations at the critical point and within the two-phase region.

The open-source version of CUBENS is available at the following link: https://github.com/pcboldini/CUBENS.

CRediT authorship contribution statement

Pietro Carlo Boldini: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Ryo Hirai:** Writing – review & editing, Visualization, Validation, Software, Methodology, Investigation, Formal analysis. **Pedro Costa:** Writing – review & editing, Supervision, Software, Funding acquisition. **Jurriaan W.R. Peeters:** Writing – review & editing, Supervision, Conceptualization. **Rene Pecnik:** Writing – review & editing, Supervision, Project administration, Methodology, Software, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The European Research Council (grant no. ERC-2019-CoG-864660, Critical) has supported the investigations presented in this study. The authors acknowledge the use of computational resources of the Dutch National Supercomputer Snellius (SURF) (grant no. 2022/ENW/01251049) and on LUMI based at CSC in Finland (via the EuroHPC Benchmark and Development Access Grant no. DEV-2024D04-039).

Appendix A. Transport properties

A.1. Chung

The modified dilute-gas viscosity, μ_k^*, and the correction term, μ_p^*, are written as

$$\mu_k^* = \mu_0^* \left[\frac{1}{G_2} + A_6 Y \right],$$
$$\mu_p^* = 3.6344 \times 10^{-6} (M^* T_c^*)^{1/2} \rho_c^{*2/3} A_7 Y^2 G_2 \exp\left[A_8 + \frac{A_9}{T_\epsilon} + \frac{A_{10}}{T_\epsilon^2} \right], \tag{A.1}$$

where

$$G_2 = \frac{A_1 \left[1 - \exp(-A_4 Y)\right] / Y + A_2 G_1 \exp(A_5 Y) + A_3 G_1}{A_1 A_4 + A_2 + A_3}, \tag{A.2}$$
$$G_1 = \frac{1 - 0.5Y}{(1-Y)^3},$$

with $Y = \rho_r / 6$ and $T_\epsilon = k_B^* T^* / \epsilon$. The constants $A_{i=1,\dots,10}$ are linear functions of the form

$$A_i = a_0(i) + a_1(i)\bar{\omega} + a_2(i)\xi_r^4 + a_3(i)\varpi, \quad i \in \{1, \dots, 10\}, \tag{A.3}$$
$$\varpi = 0.0682 + 4.704 \frac{[\text{number of -OH groups}]}{M^*},$$

where $\bar{\omega}$ is the acentric factor, $\xi_r = 131.3 \xi \rho_c^{*1/2} / T_c^{*1/2}$ is the reduced dipole moment [93], ϖ is an association parameter [57], and values for the coefficients a_0, a_1, a_2, and a_3 are given in Table A.4. Note that for non-polar fluids, $\xi_r = 0$ and $\varpi = 0$. The dilute gas viscosity, μ_0^*, is obtained as

$$\mu_0^* = 4.0785 \times 10^{-6} \frac{(M^* T^*)^{1/2} \rho_c^{*2/3}}{\Omega_v} F_c, \tag{A.4}$$
$$F_c = 1 - 0.2756\bar{\omega} + 0.059035\xi_r^4 + \varpi,$$

Table A.4
Dynamic viscosity: a_i-coefficients [94].

i	a_0	a_1	a_2	a_3
1	6.32402	50.41190	-51.68010	1189.0200
2	0.0012102	-0.0011536	-0.0062571	0.037283
3	5.28346	254.20900	-168.48100	3898.27000
4	6.62263	38.09570	-8.46414	31.41780
5	19.74540	7.63034	-14.35440	31.52670
6	-1.89992	-12.53670	4.98529	-18.15070
7	24.27450	3.44945	-11.29130	69.34660
8	0.79716	1.11764	0.012348	-4.11661
9	-0.23816	0.067695	-0.81630	4.02528
10	0.068629	0.34793	0.59256	-0.72663

Table A.5
Thermal conductivity: b_j-coefficients [94].

j	b_0	b_1	b_2	b_3
1	2.41657	0.74824	-0.91858	121.72100
2	-0.50924	-1.50936	-49.99120	69.98340
3	6.61069	5.62073	64.75990	27.03890
4	14.54250	-8.91387	-5.63794	74.34350
5	0.79274	0.82019	-0.69369	6.31734
6	-5.86340	12.80050	9.58926	-65.52920
7	81.17100	114.15800	-60.84100	466.75500

where Ω_v is the reduced collision integral, calculated as

$$\Omega_v = \frac{A_\Omega}{T_\epsilon^{B_\Omega}} + \frac{C_\Omega}{\exp(D_\Omega T_\epsilon)} + \frac{E_\Omega}{\exp(F_\Omega T_\epsilon)} + G_\Omega T_\epsilon^{B_\Omega} \sin(S_\Omega T_\epsilon^{W_\Omega} - H_\Omega), \tag{A.5}$$

with constants $A_\Omega = 1.16145$, $B_\Omega = 0.14874$, $C_\Omega = 0.52487$, $D_\Omega = 0.77320$, $E_\Omega = 2.16178$, $F_\Omega = 2.43787$, $G_\Omega = -6.435 \times 10^{-4}$, $H_\Omega = 7.27371$, $S_\Omega = 18.0323$, and $W_\Omega = -0.76830$ and

$$k_B = 1.38064852 \times 10^{-23} \frac{J}{K}, \qquad \epsilon = \frac{k_B^* T_c^*}{1.2593}. \tag{A.6}$$

The modified dilute-gas thermal conductivity, κ_k^*, and the correction term, κ_p^*, are written as

$$\kappa_k^* = \kappa_0^* \left[\frac{1}{H_2} + B_6 Y \right], \tag{A.7}$$

$$\kappa_p^* = 3.586 \times 10^{-3} \frac{T_c^{*1/2} \rho_c^{*2/3}}{M^{*1/2}} B_7 Y^2 H_2 T_r^{1/2}, \tag{A.8}$$

with the parameter $Y = \rho_r/6$, and

$$H_2 = \frac{B_1 \left[1 - \exp(-B_4 Y)\right]/Y + B_2 G_1 \exp(B_5 Y) + B_3 G_1}{B_1 B_4 + B_2 + B_3}, \qquad G_1 = \frac{1 - 0.5Y}{(1-Y)^3}. \tag{A.9}$$

The constants $B_{j=1,\dots,7}$ are linear functions given by

$$B_j = b_0(j) + b_1(j)\bar{\omega} + b_2(j)\xi_r^4 + b_3(j)\varpi, \quad j \in \{1, \dots, 7\}, \tag{A.10}$$

with the aforementioned parameters ω, ξ_r, and ϖ. Values for the coefficients b_0, b_1, b_2, and b_3 are listed in Table A.5. The same rules that apply to the dilute dynamic viscosity μ_0^* also apply here. The dilute-gas conductivity κ_0^* is equal to

$$\kappa_0^* = 31.2 \frac{\mu_0^*}{M^*} \Psi, \qquad \Psi = 1 + \alpha \frac{0.215 + 0.28288\alpha - 1.061\beta + 0.26665Z}{0.6366 + \beta Z + 1.061\alpha\beta}, \tag{A.11}$$

with

$$\alpha = \frac{c_v^*}{R_g^*} - \frac{3}{2}, \quad \beta = 0.7862 - 0.7109\bar{\omega} + 1.3168\bar{\omega}^2, \qquad Z = 2.0 + 10.5 T_r^2. \tag{A.12}$$

A.2. Validation and new case studies: flow parameters

Flow parameters in Section 5 used for the direct numerical simulations with CUBENS are reported in Table A.6 and A.7.

Data availability

Data will be made available on request.

References

[1] S. aus der Wiesche, Experimental investigation techniques for non-ideal compressible fluid dynamics, Int. J. Turbomach. Propuls. Power 8 (2023).
[2] T.M. White, G. Bianchi, L. Chai, S.A. Tassou, A.I. Sayma, Review of supercritical CO2 technologies and systems for power generation, Appl. Therm. Eng. 185 (2021) 116447.
[3] A. Guardone, P. Colonna, M. Pini, A. Spinelli, Nonideal compressible fluid dynamics of dense vapors and supercritical fluids, Annu. Rev. Fluid Mech. 56 (2024) 241–269.
[4] P. Cinnella, X. Gloerfelt, Insights into the turbulent flow of dense gases through high-fidelity simulations, Comput. Fluids 267 (2023) 106067.
[5] A. Zhao, R. Pecnik, J.W.R. Peeters, Thermodynamic analysis and heat exchanger calculations of transcritical high-temperature heat pumps, Energy Convers. Manag. 303 (2024) 118172.
[6] J.Y. Yoo, The turbulent flows of supercritical fluids with heat transfer, Annu. Rev. Fluid Mech. 45 (2013) 495–525.
[7] J. Ren, O. Marxen, R. Pecnik, Boundary-layer stability of supercritical fluids in the vicinity of the Widom line, J. Fluid Mech. 871 (2019) 831–864.
[8] P.C. Boldini, B. Bugeat, J.W.R. Peeters, M. Kloker, R. Pecnik, Transient growth in diabatic boundary layers with fluids at supercritical pressure, Phys. Rev. Fluids 9 (2024) 083901.
[9] S. Kawai, Heated transcritical and unheated non-transcritical turbulent boundary layers at supercritical pressures, J. Fluid Mech. 865 (2019) 563–601.
[10] L. Sciacovelli, P. Cinnella, X. Gloerfelt, Direct numerical simulations of supersonic turbulent channel flows of dense gases, J. Fluid Mech. 821 (2017) 153–199.
[11] A. Patel, J.W.R. Peeters, B.J. Boersma, R. Pecnik, Semi-local scaling and turbulence modulation in variable property turbulent channel flows, Phys. Fluids 27 (2015) 095101.
[12] A. Patel, B.J. Boersma, R. Pecnik, The influence of near-wall density and viscosity gradients on turbulence in channel flows, J. Fluid Mech. 809 (2016) 793–820.
[13] R. Pecnik, A. Patel, Scaling and modelling of turbulence in variable property channel flows, J. Fluid Mech. 823 (2017) R1.
[14] P.C. Ma, X.I.A. Yang, M. Ihme, Structure of wall-bounded flows at transcritical conditions, Phys. Rev. Fluids 3 (2018) 034609.
[15] J. Guo, X.I.A. Yang, M. Ihme, Structure of the thermal boundary layer in turbulent channel flows at transcritical conditions, J. Fluid Mech. 934 (2022) A45.
[16] H. Nemati, A. Patel, B.J. Boersma, R. Pecnik, The effect of thermal boundary conditions on forced convection heat transfer to fluids at supercritical pressure, J. Fluid Mech. 800 (2016) 531–556.
[17] J.W.R. Peeters, R. Pecnik, M. Rohde, T.H.J.J. van der Hagen, B.J. Boersma, Turbulence attenuation in simultaneously heated and cooled annular flows at supercritical pressure, J. Fluid Mech. 799 (2016) 505–540.
[18] K. Kim, J.-P. Hickey, C. Scalo, Pseudophase change effects in turbulent channel flow under transcritical temperature conditions, J. Fluid Mech. 871 (2019) 52–91.
[19] J. He, R. Tian, P.X. Jiang, S. He, Turbulence in a heated pipe at supercritical pressure, J. Fluid Mech. 920 (2021) A45.
[20] L.Y. Cao, R.N. Xu, J.J. Yan, S. He, X.P. Jiang, Direct numerical simulation of convective heat transfer of supercritical pressure CO2 in a vertical tube with buoyancy and thermal acceleration effect, J. Fluid Mech. 927 (2021) A29.
[21] L. Sciacovelli, X. Gloerfelt, D. Passiatore, P. Cinnella, F. Grasso, Numerical investigation of high-speed turbulent boundary layers of dense gases, Flow Turbul. Combust. 105 (2020) 555–579.
[22] X. Gloerfelt, A. Bienner, P. Cinnella, High-subsonic boundary-layer flows of an organic vapour, J. Fluid Mech. 971 (2023) A8.
[23] P.C. Boldini, R. Gaspar, B. Bugeat, P. Costa, J.W.R. Peeters, R. Pecnik, Direct numerical simulation of H-type transition in a flat-plate boundary layer with supercritical fluids, in: 14th International ERCOFTAC Symposium on Engineering Turbulence Modelling and Measurements (ETMM14), Barcelona, Spain, September 6–8, 2023.
[24] Y. Kuya, K. Totani, S. Kawai, Kinetic energy and entropy preserving schemes for compressible flows by split convective forms, J. Comput. Phys. 375 (2018) 823–853.
[25] C. Kok, A high-order low-dispersion symmetry-preserving finite-volume method for compressible flow on curvilinear grids, J. Comput. Phys. 228 (2009) 6811–6832.
[26] G. Coppola, F. Capuano, S. Pirozzoli, L. De Luca, Numerically stable formulations of convective terms for turbulent compressible flows, J. Comput. Phys. 382 (2019) 86–104.
[27] C.A. Kennedy, A. Gruber, Reduced aliasing formulations of the convective terms within the Navier–Stokes equations for a compressible fluid, J. Comput. Phys. 227 (2008) 1676–1700.
[28] S. Pirozzoli, Generalized conservative approximations of split convective derivative operators, J. Comput. Phys. 229 (2010) 7180–7190.
[29] Y. Kuya, S. Kawai, High-order accurate kinetic-energy and entropy preserving (KEEP) schemes on curvilinear grids, J. Comput. Phys. 442 (2021) 110482.
[30] Y. Tamaki, Y. Kuya, S. Kawai, Comprehensive analysis of entropy conservation property of non-dissipative schemes for compressible flows: KEEP scheme redefined, J. Comput. Phys. 468 (2022) 111494.
[31] A. Aiello, C. De Michele, G. Coppola, Entropy conservative discretization of compressible Euler equations with an arbitrary equation of state, preprint, arXiv:2408.01235, 2024.
[32] C. De Michele, G. Coppola, Novel pressure-equilibrium and kinetic-energy preserving fluxes for compressible flows based on the harmonic mean, J. Comput. Phys. 518 (2024) 113338.

Table A.6
Base-flow properties for ideal-gas cases of Section 5. The inlet Richardson number at the wall is defined as $Ri_{w,0} = Ri_{unit,0}(\rho_w - 1)$.

Case	TP	M_∞	Ec_∞	Pr_∞	$Ri_{w,0}$	γ	T_w^*/T_∞^*
Section 5.1	power law	[0.1, 0.5, 1.0]	[0.004, 0.1, 0.4]	0.75	-	1.4	-
Section 5.2	Sutherland's law	0.2	0.016	0.75	-		adiabatic
Section 5.3	Sutherland's law	0.2	0.016	0.75	-		adiabatic
Section 5.4	Sutherland's law	0.2	0.016	0.72	[-0.01,0,0.01]		1.05

Table A.7
Base-flow properties for non-ideal gas cases of Section 5. The free-stream compressibility factor is defined as $Z_\infty = p_{r,\infty}/(\rho_{r,\infty} R_g T_{r,\infty})$.

Case	EoS	TP	$p_{r,\infty}$	$T_{r,\infty}$	M_∞	Ec_∞	Pr_∞	Z_∞	c_v^*/R_g^*	T_w^*/T_∞^*
Section 5.5.1	PR	Chung	1.084	0.92	0.2	0.01	2.39	0.165	9/2	1.125
Section 5.5.2	VdW	JST	1.1	0.90	0.2	0.016	1.0	0.254	9/2	1.222

[33] N. Shima, Y. Kuya, Y. Tamaki, S. Kawai, Preventing spurious pressure oscillations in split convective form discretization for compressible flows, J. Comput. Phys. 427 (2021) 110060.
[34] S. Kawai, H. Terashima, H. Negishi, A robust and accurate numerical method for transcritical turbulent flows at supercritical pressure with an arbitrary equation of state, J. Comput. Phys. 300 (2015) 116–135.
[35] J.-P. Hickey, P.C. Ma, M. Ihme, S. Thakur, Large eddy simulation of shear coaxial rocket injector: real fluid effects, AIAA Paper 2013-4071, 2013.
[36] M. Bernades, L. Jofre, F. Capuano, Kinetic-energy- and pressure-equilibrium-preserving schemes for real-gas turbulence in the transcritical regime, J. Comput. Phys. 493 (2023) 112477.
[37] S. Alowayyed, D. Groen, P.V. Coveney, A.G. Hoekstra, Multiscale computing in the Exascale era, J. Comput. Sci. 22 (2017) 15–25.
[38] TOP500.org, TOP500 the list, https://top500.org/lists/top500/list/2024/06/, 2024. (Accessed 18 October 2024).
[39] TOP500.org, GREEN500 the list, https://top500.org/lists/green500/2024/06/, 2024. (Accessed 18 October 2024).
[40] Y. Kim, D. Ghosh, E.M. Constantinescu, R. Balakrishnan, GPU-accelerated DNS of compressible turbulent flows, Comput. Fluids 251 (2023) 105744.
[41] X. Zhu, E. Phillips, V. Spandan, J. Donners, G. Ruetsch, J. Romero, R. Ostilla-Mónico, Y. Yang, D. Lohse, R. Verzicco, M. Fatica, R. Stevens, AFiD-GPU: a versatile Navier–Stokes solver for wall-bounded turbulent flows on GPU clusters, Comput. Phys. Commun. 229 (2018) 199–210.
[42] P. Costa, E. Phillips, L. Brandt, M. Fatica, GPU acceleration of CaNS for massively-parallel direct numerical simulations of canonical fluid flows, Comput. Math. Appl. 81 (2021) 502–511.
[43] M. Bernardini, D. Modesti, F. Salvadore, S. Pirozzoli, STREAmS 2.0: supersonic turbulent accelerated Navier-Stokes solver version 2.0, Comput. Phys. Commun. 285 (2023) 108644.
[44] M. Di Renzo, HTR-1.3 solver: predicting electrified combustion using the hypersonic task-based research solver, Comput. Phys. Commun. 272 (2022) 108247.
[45] F. De Vanna, F. Avanzi, M. Cogo, S. Sandrin, M. Bettencourt, F. Picano, E. Benini, URANOS: a GPU accelerated Navier-Stokes solver for compressible wall-bounded flows, Comput. Phys. Commun. 287 (2023) 108717.
[46] F. Salvadore, G. Rossi, S. Sathyanarayana, M. Bernardini, OpenMP offload toward the exascale using intel®GPU max 1550: evaluation of STREAmS compressible solver, J. Supercomput. 80 (2024) 21094–21127.
[47] F. De Vanna, G. Baldan, URANOS-2.0: improved performance, enhanced portability, and model extension towards exascale computing of high-speed engineering flows, Comput. Phys. Commun. 303 (2024) 109285.
[48] A.P.S. Wheeler, High fidelity simulation of dense vapours with thermodynamic consistent modelling, Comput. Fluids 268 (2024) 106088.
[49] H. Song, A.S. Ghate, S. Dai, A. Chandra, S.K. Lele, Conservative and robust compact finite difference approach for simulations of dense gas flows, AIAA AVIATION 2023 Forum, 2023.
[50] L. Jofre, A. Abdellatif, G. Oyarzun, RHEA: an open-source reproducible hybrid-architecture flow solver engineered for academia, J. Open Sour. Softw. 8 (2023) 4637.
[51] OpenACC Architecture Review Board, OpenACC application program interface version 3.2, https://www.openacc.org/sites/default/files/inline-images/Specification/OpenACC-3.2-final.pdf, 2021.
[52] B. Bugeat, P.C. Boldini, A.M. Hasan, R. Pecnik, Instability in strongly stratified plane Couette flow with application to supercritical fluids, J. Fluid Mech. 984 (2024) A31.
[53] O. Marxen, G. Iaccarino, E.S.G. Shaqfeh, Numerical simulation of hypersonic instability using different gas models, Center Turbul. Res. Rep. CTR (2007) 15–27.
[54] A.J. Jossi, L.I. Stiel, G. Thodos, The viscosity of pure substances in the dense gaseous and liquid phases, AIChE J. 8 (1962) 59–63.
[55] L.I. Stiel, G. Thodos, The thermal conductivity of nonpolar substances in the dense gaseous and liquid regions, AIChE J. 10 (1964) 26–30.
[56] J. Hirschfelder, C. Curtiss, R. Bird, Molecular Theory of Gases and Liquids, Wiley, 1954.
[57] T.H. Chung, M. Ajlan, L.L. Lee, K.E. Starling, Generalized multiparameter correlation for nonpolar and polar fluid transport properties, Ind. Eng. Chem. Res. 27 (1988) 671–679.
[58] S. Pirozzoli, Stabilized non-dissipative approximations of Euler equations in generalized curvilinear coordinates, J. Comput. Phys. 230 (2011) 2997–3014.
[59] S. Gottlieb, C.W. Shu, Total variation diminishing Runge–Kutta schemes, Math. Comput. 67 (1998) 73–85.
[60] G.N. Coleman, R.D. Sandberg, A primer on direct numerical simulation of turbulence - methods, procedures and guidelines, Technical Report, University of Southampton, 2010.
[61] O. Marxen, T.E. Magin, E.S.G. Shaqfeh, G. Iaccarino, A method for the direct numerical simulation of hypersonic boundary-layer instability with finite-rate chemistry, J. Comput. Phys. 255 (2013) 572–589.
[62] O. Marxen, G. Iaccarino, T.E. Magin, Direct numerical simulations of hypersonic boundary-layer transition with finite-rate chemistry, J. Fluid Mech. 755 (2014) 35–49.
[63] G. Urbin, D. Knight, Large-eddy simulation of a supersonic boundary layer using an unstructured grid, AIAA J. 39 (2001) 1288–1295.
[64] A.M. Hasan, J. Larsson, S. Pirozzoli, R. Pecnik, Estimating mean profiles and fluxes in high-speed turbulent boundary layers using inner/outer-layer scalings, AIAA J. 62 (2024) 848–853.
[65] E.R. van Driest, Turbulent boundary layer in compressible fluids, J. Aeronaut. Sci. 18 (1951) 145–160.
[66] T.S. Lund, X. Wu, K.D. Squires, Generation of turbulent inflow data for spatially-developing boundary layer simulations, J. Comput. Phys. 140 (1998) 233–258.
[67] K.W. Thompson, Time dependent boundary conditions for hyperbolic systems, J. Comput. Phys. 68 (1987) 1–24.
[68] T.J. Poinsot, S.K. Lele, Time dependent boundary conditions for hyperbolic systems, J. Comput. Phys. 101 (1992) 104–129.
[69] N. Okong'o, J. Bellan, Consistent boundary conditions for multicomponent real gas mixtures based on characteristic waves, J. Comput. Phys. 176 (2002) 330–344.
[70] D.H. Rudy, J.C. Strikwerda, A nonreflecting outflow boundary condition for subsonic Navier-Stokes calculations, J. Comput. Phys. 36 (1980) 55–70.
[71] A. Mani, Analysis and optimization of numerical sponge layers as a nonreflective boundary treatment, J. Comput. Phys. 231 (2012) 704–716.
[72] Y.S. Kachanov, V.V. Kozlov, V.Y. Levchenko, Nonlinear development of a wave in a boundary layer, Fluid Dyn. 12 (1977) 383–390.
[73] X. Huai, R.D. Joslin, U. Piomelli, Large-eddy simulation of transition to turbulence in boundary layers, Theor. Comput. Fluid Dyn. 9 (1997) 149–163.
[74] U. Rist, H. Fasel, Direct numerical simulation of controlled transition in a flat-plate boundary layer, J. Fluid Mech. 298 (1995) 211–248.
[75] H.F. Fasel, U. Rist, U. Konzelmann, Numerical investigation of the three-dimensional development in boundary-layer transition, AIAA J. 28 (1990) 29–37.
[76] T. Sayadi, C.W. Hamman, P. Moin, Direct numerical simulation of complete H-type and K-type transitions with implications for the dynamics of turbulent boundary layers, J. Fluid Mech. 724 (2013) 480–509.
[77] K.J. Franko, S.K. Lele, Breakdown mechanisms and heat transfer overshoot in hypersonic zero pressure gradient boundary layers, J. Fluid Mech. 730 (2013) 491–532.
[78] N. Li, S. Laizet, 2DECOMP&FFT - a highly scalable 2D decomposition library and FFT interface, in: Cray User Group 2010 Conference, 2010, pp. 1–13.
[79] NVIDIA, NVIDIA HPC SDK Version 23.1 Documentation, https://docs.nvidia.com/hpc-sdk/archive/23.1/index.html, 2023.
[80] AMD, ROCm™ Software 6.1.2, https://rocm.docs.amd.com/en/latest/, 2024.
[81] J.R. DeBonis, Solutions of the Taylor-Green vortex problem using high-resolution explicit finite difference methods, in: 51st AIAA Aerospace Sciences Meeting: The New Horizons Forum and Aerospace Exposition (AIAA, 2013), 2013, 2013-0382.

[82] N. Peng, Y. Yang, Effects of the Mach number on the evolution of vortex-surface fields in compressible Taylor-Green flows, Phys. Rev. Fluids 3 (2018) 013401.
[83] L. Sciacovelli, D. Passiatore, P. Cinnella, G. Pascazio, Assessment of a high-order shock-capturing central-difference scheme for hypersonic turbulent flow simulations, Comput. Fluids 230 (2021) 105134.
[84] D.J. Lusher, N.D. Sandham, Assessment of low-dissipative shock-capturing schemes for the compressible Taylor–Green vortex, AIAA J. 59 (2021) 533–545.
[85] M. Frigo, S.G. Johnson, The design and implementation of FFTW3, Proc. IEEE 93 (2005) 216–231.
[86] T. Herbert, Secondary instability of boundary layers, Annu. Rev. Fluid Mech. 20 (1988) 487–526.
[87] F.M. White, Viscous Fluid Flow, 3rd edition, McGraw-Hill, Boston, 2006.
[88] P. Schlatter, R. Örlü, Assessment of direct numerical simulation data of turbulent boundary layers, J. Fluid Mech. 659 (2010) 116–126.
[89] T. Bai, K.P. Griffin, L. Fu, Compressible velocity transformations for various non-canonical wall-bounded turbulent flows, AIAA J. 60 (2022) 4325.
[90] A.M. Hasan, J. Larsson, S. Pirozzoli, R. Pecnik, Incorporating intrinsic compressibility effects in velocity transformations for wall-bounded turbulent flows, Phys. Rev. Fluids 8 (2023) L112601.
[91] J. Romero, P. Costa, M. Fatica, Distributed-memory simulations of turbulent flows on modern GPU systems using an adaptive pencil decomposition library, in: Proceedings of the Platform for Advanced Scientific Computing Conference, June 2022, ACM, 2022.
[92] J.A.K. Horwitz, Estimating the carbon footprint of computational fluid dynamics, Phys. Fluids 36 (2024) 045109.
[93] R.C. Reid, J.M. Prausnitz, T.K. Sherwood, The Properties of Gases and Liquids, 5th ed., McGraw-Hill Education, New York, 2001.
[94] L. Duan, Q. Zheng, Z. Jiang, J. Wang, Dense gas effect on small-scale structures of compressible isotropic turbulence, Phys. Fluids 33 (2021) 115113.