

GPU-native extension of a high-order compressible solver to multicomponent reacting flows

Edoardo Forti

`edoardo.forti@uniroma1.it`

Sapienza, University of Rome

Marco Fratini

Sapienza, University of Rome

Agostino Neri

European Space Agency

Mauro Valorani

Sapienza, University of Rome

Fulvio Stella

Sapienza, University of Rome

Matteo Bernardini

Sapienza, University of Rome

Pietro Paolo Ciottoli

Sapienza, University of Rome

Research Article

Keywords: GPUs, DNS, Exascale computing, Compressible reacting flows, High-order numerics

Posted Date: October 20th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11112065/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

GPU-native extension of a high-order compressible solver to multicomponent reacting flows

Edoardo Forti^{1*}, Marco Fratini¹, Agostino Neri², Mauro Valorani¹,
Fulvio Stella¹, Matteo Bernardini¹, Pietro Paolo Ciottoli¹

^{1*}Department of Mechanical and Aerospace Engineering, Sapienza,
University of Rome, Via Eudossiana, 18, Rome, 00184, Lazio, Italy.

²European Space Agency, Via Galileo Galilei, 1, Frascati, 00044, Lazio,
Italy.

*Corresponding author(s). E-mail(s): edoardo.forti@uniroma1.it;

Contributing authors: marco.fratini@uniroma1.it; agostino.neri@esa.int;
mauro.valorani@uniroma1.it; fulvio.stella@uniroma1.it;
matteo.bernardini@uniroma1.it; pietropaolo.ciottoli@uniroma1.it;

Abstract

We present an extension of STREAmS-2.0, a GPU-accelerated solver for direct numerical simulation of compressible turbulent flows, to multicomponent chemically reacting gas mixtures. Three principal numerical contributions are introduced. First, the energy-preserving split discretization of the convective fluxes is reformulated for variable-composition mixtures, with an eigenvalue structure extended to the additional species advective branches; species diffusion is modeled via the Hirschfelder–Curtiss approximation with a mass-conservation corrector. Second, a chemical kinetics module supporting Arrhenius, three-body, and pressure-dependent fall-off reactions is integrated through a custom interface with Cantera, with transport and kinetic properties pre-tabulated at runtime for GPU execution. Third, chemical stiffness is handled through an operator-splitting strategy coupling the explicit three-stage Runge–Kutta flow integrator with a four-stage fourth-order implicit Rosenbrock method. The extended solver is validated against nine standardized benchmark cases, from zero- to three-dimensional configurations, covering multicomponent transport and chemical kinetics, showing excellent agreement with reference solutions. Weak-scaling efficiency exceeds 99% up to 512 NVIDIA A100 GPUs, while strong scaling up to 64 GPUs is governed by per-GPU memory usage, with a critical threshold near 10% of memory capacity below which efficiency degrades sharply. Profiling shows that

chemical source-term evaluation accounts for approximately 40% of the per-step wall time.

Keywords: GPUs, DNS, Exascale computing, Compressible reacting flows, High-order numerics

1 Introduction

High-performance computing (HPC) has become an indispensable pillar of modern scientific research, enabling simulations of unprecedented fidelity and scale. Within computational fluid dynamics (CFD), the emergence of heterogeneous multi-node architectures equipped with high-performance GPU accelerators has fundamentally transformed the landscape of large-scale flow simulation. As documented in the most recent TOP500 list ([TOP500 2026](#)), eight of the ten most powerful supercomputers in the world rely on GPU acceleration, with the leading systems now exceeding the exascale threshold, making GPU-native solver design a prerequisite for state-of-the-art high-fidelity simulations.

Among the most computationally demanding problems in CFD, Direct Numerical Simulation (DNS) of compressible reacting flows occupies a distinct position of difficulty. The challenge stems from the simultaneous presence of several physically disparate phenomena: compressible turbulence, multicomponent molecular transport, and chemical kinetics, each introducing its own range of spatial and temporal scales that must be resolved without modelling assumptions. In particular, chemical kinetics introduces two fundamental numerical difficulties. First, the governing equations become stiff: the timescales associated with individual reaction steps can span several orders of magnitude and may be several orders of magnitude smaller than the fluid-dynamic timescale, precluding the use of explicit time integrators without prohibitive time-step restrictions ([Sportisse 2000](#)). Second, multicomponent chemistry substantially increases the number of variables: for a mixture of N_s species evolving through N_R reactions, the chemical source term vector $\dot{\omega} \in \mathbb{R}^{N_s}$ requires evaluating species-specific Jacobians at every grid point and at every sub-step of the chemistry integrator, incurring a computational cost that grows quadratically (and at worst cubically) with the size of the kinetic mechanism. These difficulties, combined with the fine-grid requirements imposed by both turbulence and flame-front resolution, make DNS of reacting flows one of the canonical grand-challenge problems of computational physics. Such simulations are also directly relevant to aerospace propulsion and combustion engineering, where the predictive fidelity of numerical models is critical to the development of cleaner and more efficient propulsion systems.

A number of strategies have been proposed to address the stiffness problem. Implicit time-stepping methods applied globally to the coupled system avoid the time-step restriction but require the inversion of large, sparse Jacobians at every stage, making them computationally prohibitive at the mesh sizes required for DNS. Operator splitting decouples the flow and chemistry sub-problems, allowing each to be integrated with a method suited to its own timescale ([Sportisse 2000](#)). Within this framework, the

chemistry sub-system reduces to a set of pointwise initial-value problems, one per grid cell, which are intrinsically independent and therefore well-matched to parallel GPU architectures. The choice of ODE solver for the chemistry sub-step remains a critical design decision: CVODE-type multistep methods (Hindmarsh et al. 2005) offer high accuracy but are ill-suited to massively parallel execution due to their variable-order, variable-step history requirements; explicit Runge–Kutta methods are simple but ineffective for stiff systems; while Rosenbrock methods provide an attractive compromise, offering L -stability and fixed-stage structure amenable to GPU parallelisation (Stone et al. 2018; Utkarsh et al. 2024).

Several GPU-accelerated compressible flow solvers have been developed in recent years to address these challenges. URANOS (De Vanna et al. 2023) employs OpenACC directives for DNS, LES, and wall-modeled LES of compressible wall-bounded flows. PyFR (Witherden et al. 2014) and ZEFR (Romero et al. 2020) exploit CUDA on unstructured meshes via high-order flux reconstruction. HTR (Di Renzo and Pirozzoli 2021), based on the Legion programming model (Bauer et al. 2012), targets hypersonic reacting flows on heterogeneous architectures. For incompressible flows, AFiD-GPU (Zhu et al. 2018) and CaNS (Costa et al. 2021) have demonstrated strong GPU performance using CUDA-Fortran and OpenACC, respectively. Community frameworks such as AMReX (Zhang et al. 2021) provide GPU-capable infrastructure for adaptive mesh refinement, while YALES2 (Moureau et al. 2011) targets turbulent combustion LES. HiPSTAR (Sandberg 2013) is a CUDA-based solver suited for DNS and LES of compressible reacting flows. Despite this progress, combining high-order shock-capturing numerics for the compressible Euler equations, a fully general multi-component chemistry module, and an implicit stiff chemistry integrator within a single solver that demonstrates near-ideal weak scaling at the hundreds-of-GPUs scale has not been comprehensively addressed in the open literature.

Within this context, the Supersonic TuRbulEnt Accelerated Navier-Stokes Solver (STREAmS), developed at Sapienza University of Rome, was originally designed for DNS of single-species compressible turbulent flows (Salvadore et al. 2013; Bernardini et al. 2021), exploiting the CUDA Fortran programming model on NVIDIA GPU architectures. The subsequent release, STREAmS-2.0 (Bernardini et al. 2023), introduced an object-oriented design, extended support to AMD GPUs (Sathyanarayana et al. 2025), and added energy- and entropy-preserving convective flux schemes, a recycling/rescaling inflow boundary condition, and a thermally perfect gas model with variable specific heats. The solver has been successfully employed in DNS studies of supersonic internal flows (Modesti and Pirozzoli 2016), roughness-induced boundary-layer transition (Bernardini et al. 2012), and supersonic and hypersonic wall-bounded turbulence (Pirozzoli and Bernardini 2011, 2013).

The present work extends STREAmS-2.0 to multicomponent, chemically reacting compressible flows. The principal computational contributions are:

1. a multicomponent reformulation of the convective and diffusive flux discretizations, extending the existing energy-preserving scheme to variable-composition mixtures, with species diffusion modeled via the Hirschfelder–Curtiss approximation and a mass-conservation corrector;

2. a chemical kinetics module supporting Arrhenius, three-body, and pressure-dependent fall-off reactions following the [ANSYS \(2015\)](#) CHEMKIN standard, with thermochemical data managed through a custom interface with Cantera ([Goodwin et al. 2025](#)) and pre-tabulated at runtime to ensure compatibility with GPU execution;
3. an operator-splitting strategy coupling the explicit three-stage Runge–Kutta flow integrator with a four-stage, fourth-order Rosenbrock method for the stiff pointwise chemistry sub-system ([Press et al. 1993](#)), with adaptive step-size control based on an embedded error estimator and classical safety-factor step-size adaptation.

The extended solver is validated against a comprehensive suite of nine benchmark cases following the procedures of [Abdelsamie et al. \(2021\)](#) and [Martínez Ferrer et al. \(2014\)](#). Furthermore, its performance is assessed on NVIDIA A100 GPUs, conducting strong scaling analyses up to 64 GPUs and weak scaling assessments up to 512 GPUs. The remainder of the paper is organized as follows. Section 2 presents the governing equations and physical models. Section 3 details the numerical discretization, time integration, and chemistry coupling strategies. Sections 4 and 5 present the validation results and scalability analysis, respectively. Concluding remarks are offered in Section 6.

2 Governing equations

The governing equations and physical models for the single-species, compressible, and non-reactive version of the solver are described in detail in previous works ([Bernardini et al. 2021](#)). Those have been reformulated to account for a chemically reacting mixture of ideal gases, leading to the following form of the dimensional and fully compressible Navier–Stokes conservation equations:

$$\frac{\partial}{\partial t} (\rho Y_m) + \frac{\partial}{\partial x_i} [\rho(u_i + V_{m,i})Y_m] = \dot{\omega}_m, \quad m = 1, \dots, N_s, \quad (1)$$

$$\frac{\partial}{\partial t} (\rho u_i) + \frac{\partial}{\partial x_j} (\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j}, \quad (2)$$

$$\frac{\partial}{\partial t} (\rho E) + \frac{\partial}{\partial x_j} (\rho u_j H) = -\frac{\partial q_j}{\partial x_j} + \frac{\partial \tau_{ij} u_i}{\partial x_j}, \quad (3)$$

where u_i denotes the velocity component in the i -th direction, ρ represents the density, and $V_{m,i}$ is the diffusion velocity modeled via the Hirschfelder and Curtiss approximation ([Hirschfelder et al. 1954](#)). $\dot{\omega}_m$ denotes the chemical mass production rate of species $m = 1, \dots, N_s$ per unit volume, while p is the pressure. The specific total energy, E , is defined as

$$E = \int_{T_0}^T c_v dT - \frac{R_0 T_0}{W} + \sum_{m=1}^{N_s} \Delta h_{f,m}^o Y_m + \frac{1}{2} u_i u_i, \quad (4)$$

with R_0 the universal gas constant, $c_v = \sum_{m=1}^{N_s} c_{v,m} Y_m$ is the mean specific heat capacity at constant volume, $\Delta h_{f,m}^o$ is the enthalpy of formation, and W denotes

the mean molecular weight. Finally, the specific total enthalpy is given by $H = E + p/\rho$ (Poinsot and Veynante 2011). The components of the energy flux vector q_j and of the viscous stress tensor τ_{ij} are

$$q_j = -\lambda \frac{\partial T}{\partial x_j} + \rho \sum_{m=1}^{N_s} h_m Y_m V_{m,j}, \quad (5)$$

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right). \quad (6)$$

The second term in Eq. (5) is associated with the diffusion of species with different enthalpies, while λ and μ are the thermal conductivity and viscosity of the mixture, respectively.

The chemical source term in Eq. (1) is modeled using the Arrhenius formulation, with reaction types defined by the ANSYS (2015) CHEMKIN standard. STREAMS-2.0 supports Arrhenius reactions, three-body reactions, and fall-off reactions modeled using the Lindemann (Lindemann et al. 1922), Troe (Gilbert et al. 1983), and SRI (Stewart et al. 1989) formulations. The source term is evaluated as follows:

$$\dot{\omega}_m = \sum_{l=1}^{N_R} \dot{\omega}_{ml} = W_m \sum_{l=1}^{N_R} \nu_{ml} \mathcal{Q}_l \quad \text{with} \quad \frac{\dot{\omega}_{ml}}{W_m \nu_{ml}} = \mathcal{Q}_l. \quad (7)$$

Here, W_m is the molecular weight of species m , $\nu_{ml} = \nu''_{ml} - \nu'_{ml}$, with ν'_{ml} and ν''_{ml} molar stoichiometric coefficients of species m in reaction l for reactants and products respectively, while $\mathcal{Q}_l$ is the rate of progress of reaction $l = 1, \dots, N_R$, calculated as:

$$\mathcal{Q}_l = K_{f,l} \prod_{m=1}^{N_s} [X_m]^{\nu'_{ml}} - K_{r,l} \prod_{m=1}^{N_s} [X_m]^{\nu''_{ml}}, \quad (8)$$

where $[X_m]$ denotes the molar concentration of species m , while $K_{f,l}$ and $K_{r,l}$ denote the forward and reverse rate constants for the reaction l . In STREAMS-2.0, $K_{f,l}$ is modeled using the Arrhenius law, while $K_{r,l}$ is computed from the reaction equilibrium constant $K_{c,l}$:

$$K_{f,l} = A_{f,l} T^{\beta_l} \exp \left(-\frac{E_l}{R_0 T} \right) = A_{f,l} T^{\beta_l} \exp \left(-\frac{T_{a,l}}{T} \right), \quad (9)$$

$$K_{r,l} = \frac{K_{f,l}}{K_{c,l}}, \quad (10)$$

where $A_{f,l}$ is the pre-exponential factor, β_l the temperature exponent, $T_{a,l}$ the activation temperature, and $E_l = R_0 T_{a,l}$ the activation energy.

The evaluation of transport properties and chemical source terms requires a large set of thermochemical data, including species thermodynamic polynomials, pure-species transport parameters, and reaction rate coefficients. These are managed through a

custom C++/Fortran interface to Cantera (Goodwin et al. 2025), which parses the chemical mechanism, thermodynamic, and transport databases in the standard YAML format. Since Cantera operates in a CPU environment, it cannot be invoked within GPU kernels. Its use is therefore confined to a preprocessing stage: at initialization, all species and reaction quantities are pre-tabulated over a temperature range, namely the pure-species dynamic viscosities $\mu_m(T)$ and thermal conductivities $\lambda_m(T)$, the binary mass diffusivities $\mathcal{D}_{mn}(T)$ (evaluated at a reference pressure p_{ref}), and the equilibrium constants $K_{c,l}(T)$ of Eq. (10). The resulting tables are transferred to GPU memory at startup. At runtime, the GPU kernels retrieve the properties by linear interpolation and reconstruct the local mixture properties from the instantaneous thermodynamic state: the mixture viscosity μ via Wilke’s rule (Wilke 1950), the thermal conductivity λ via the Mathur et al. (1967) formulation, and the mixture-averaged diffusion coefficients entering the Hirschfelder–Curtiss approximation from the tabulated binary diffusivities, dynamically scaled by p_{ref}/p . Likewise, the reverse rate constants $K_{r,l}$ are obtained from Eq. (10) using the tabulated $K_{c,l}$. Pre-tabulating $K_{c,l}$ avoids the evaluation of computationally expensive NASA thermodynamic polynomials inside the GPU kernels, whereas the forward rate constants $K_{f,l}$ are efficiently evaluated directly from the Arrhenius law, Eq. (9). This strategy avoids costly kinetic-theory evaluations inside the kernels and keeps the entire time integration fully GPU-resident, with no CPU–GPU data exchange during the computation.

3 Numerical methods

This section presents a brief summary of the spatiotemporal discretization algorithms implemented in the solver for numerically integrating the conservation equations. The reader is referred to the cited references for further details on the formulation.

3.1 Spatial discretization

To properly handle multicomponent flows, the treatment of the convective terms was revised with respect to the single-component baseline. The solver evolves the full set of partial densities, and the thermal Equation of State (EoS) is consequently defined as:

$$p := p(\rho_1, \dots, \rho_{N_s}, e), \quad (11)$$

where p is the pressure and e is the the sum of the sensible and chemical specific internal energy, i.e. the specific total energy E minus the kinetic contribution, $e = E - u_i u_i / 2$. Accordingly, the set of conservative variables is

$$\mathbf{C} = [\rho_1, \dots, \rho_{N_s}, \rho u, \rho v, \rho w, \rho E]^T, \quad (12)$$

with mixture density ρ obtained as $\rho = \sum_{m=1}^{N_s} \rho_m$. The eigenvalue structure of the extended Jacobian matrix comprises two acoustic modes, with eigenvalues $u \pm c$ where c is the frozen speed of sound, and $N_s + 2$ advective modes with eigenvalue u . The corresponding right and left eigenvectors, derived following the methodology of Shuen et al. (1990) (who, however, used the total density as a conservative variable), are

reported in Appendix A.

The convective fluxes are treated using a hybrid energy-preserving and shock-capturing scheme in locally conservative form, following the approach of Pirozzoli (2010). In smooth (shock-free) regions, the numerical flux of species m at the cell interface $i+1/2$ takes the split form

$$\hat{f}_{Y_m, i+1/2} = \frac{1}{4} \sum_{r=1}^R a_r \sum_{s=0}^{r-1} (\rho_{i-s} + \rho_{i-s+r}) (u_{i-s} + u_{i-s+r}) (Y_{m, i-s} + Y_{m, i-s+r}), \quad (13)$$

where a_r are the standard central finite-difference coefficients yielding order of accuracy $2R$, and $Y_m = \rho_m/\rho$ is the mass fraction of species m . The same operator is applied consistently to the momentum and energy equations, ensuring discrete conservation of kinetic energy in the incompressible inviscid limit (Pirozzoli et al. 2010). Optionally, the arithmetic density average appearing in Eq. (13), together with the analogous density and internal-energy averages in the momentum and total-energy fluxes, can be treated using a KEEP- n scheme (Tamaki et al. 2022), which substantially improves the discrete entropy-conservation properties of the scheme for inviscid smooth flows, approaching exact local entropy conservation as n increases.

The shock-capturing capability is achieved by hybridizing the central scheme with an upwind flux, computed in the flagged regions by means of a $(2r-1)$ th-order WENO reconstruction (Jiang and Shu 1996). The local smoothness of the solution is assessed using the dilatation-based sensor of Ducros et al. (1999),

$$\theta = \max \left(\frac{-\nabla \cdot \mathbf{u}}{\sqrt{(\nabla \cdot \mathbf{u})^2 + |\nabla \times \mathbf{u}|^2 + u_0^2/L_0^2}}, 0 \right) \in [0, 1], \quad (14)$$

where u_0 and L_0 are suitable reference velocity and length scales. Cells satisfying $\theta > \theta_{\text{thr}}$ undergo shock-capturing reconstruction, whereas the remaining cells employ a central energy-preserving flux. The upwind flux is evaluated either through a characteristic-wise Lax–Friedrichs flux-vector splitting, based on the characteristic decomposition at the Roe-averaged state, where the WENO reconstruction is applied to the split characteristic fluxes, or via the HLLC (Harten–Lax–van Leer Contact) (Toro 2009) approximate Riemann solver, where the WENO reconstruction provides the left and right states at the cell interface. The characteristic decomposition and the HLLC solver are detailed in Appendices A and B. The implementation of the HLLC Riemann solver described in Appendix B was developed with the assistance of a large language model (Gemini, Google). All generated code was reviewed, tested, and validated by the authors, who repeated the tests reported below and obtained equivalent results. No other part of the solver was developed with LLM assistance.

A further extension with respect to the single-component baseline is the treatment of species diffusion: the diffusion velocity of species m appearing in Eq. (1) is modeled using the Hirschfelder–Curtiss approximation (Hirschfelder et al. 1954) with a mass-conservation corrector following the formulation described by Poinso and Veynante

(2011):

$$\mathbf{V}_m = -D_m \nabla \ln X_m + \sum_{n=1}^{N_s} Y_n D_n \nabla \ln X_n, \quad (15)$$

where X_m and D_m are the molar fraction and the mixture-averaged diffusion coefficient of species m , respectively. The first term is the Fickian driving force; the second enforces zero net diffusion mass flux, guaranteeing global mass conservation at the discrete level. The resulting diffusive flux for species m in the x -direction at face $i + 1/2$ is evaluated using arithmetic averaging of cell-centered quantities as

$$\mathcal{F}_{Y_m}|_{i+1/2} = -(\overline{\rho D})_m \frac{W_m}{\tilde{W}} \nabla X_m + \bar{Y}_m \sum_{n=1}^{N_s} (\overline{\rho D})_n \frac{W_n}{\tilde{W}} \nabla X_n, \quad (16)$$

where overbars denote arithmetic averages of the two cell-centered values on either side of the face and $\tilde{W}$ is the harmonic mean of the local mixture molecular weights. The enthalpy flux associated with species diffusion, $\sum_m \mathcal{F}_{Y_m} \bar{h}_m$, is accumulated face-by-face during the diffusion-flux kernel and added to the total energy equation.

3.2 Time integration

As in the baseline STREAMS-2.0 solver, the discretization of the spatial derivatives yields a semi-discrete system of ordinary differential equations,

$$\frac{d\mathbf{C}}{dt} = \mathbf{R}(\mathbf{C}), \quad (17)$$

where $\mathbf{C} = [\rho Y_1, \dots, \rho Y_{N_s}, \rho u, \rho v, \rho w, \rho E]^T$ is the vector of conservative variables and $\mathbf{R}$ the residual vector. For non-reactive simulations, the system is advanced using Wray's three-stage, third-order Runge-Kutta scheme (Spalart et al. 1991),

$$\mathbf{C}^{(\ell+1)} = \mathbf{C}^{(\ell)} + \alpha_\ell \Delta t \mathbf{R}^{(\ell-1)} + \beta_\ell \Delta t \mathbf{R}^{(\ell)}, \quad \ell = 0, 1, 2, \quad (18)$$

with $\mathbf{C}^{(0)} = \mathbf{C}^n$, $\mathbf{C}^{n+1} = \mathbf{C}^{(3)}$, and coefficients $\alpha_\ell = (0, -17/60, -5/12)$, $\beta_\ell = (8/15, 5/12, 3/4)$. The addition of a chemical source term introduces severe stiffness, since reaction timescales can be several orders of magnitude smaller than the fluid-dynamic ones. Rather than reducing the global time step to meet the stability constraint of the explicit scheme, an operator-splitting strategy is adopted. Specifically, two different splitting techniques are implemented: the second-order Strang splitting (Sportisse 2000) and the Simpler Balanced Splitting (Wu et al. 2019). To this end, the residual is decomposed as $\mathbf{R}(\mathbf{C}) = \mathbf{T}(\mathbf{C}) + \mathbf{\Omega}(\mathbf{C})$, where $\mathbf{T}$ collects the convective and diffusive contributions and $\mathbf{\Omega}$ the chemical source terms. Denoting by $\mathcal{L}_{\text{flow}}^\tau$ and $\mathcal{L}_{\text{chem}}^\tau$ the solution operators of the sub-problems $d\mathbf{C}/dt = \mathbf{T}(\mathbf{C})$ and $d\mathbf{C}/dt = \mathbf{\Omega}(\mathbf{C})$ over a time interval τ , the Strang scheme reads

$$\mathbf{C}^{n+1} = \mathcal{L}_{\text{chem}}^{\Delta t/2} \circ \mathcal{L}_{\text{flow}}^{\Delta t} \circ \mathcal{L}_{\text{chem}}^{\Delta t/2} (\mathbf{C}^n). \quad (19)$$

Concerning the Simpler Splitting, at each time level the two operators are balanced through the constant $\mathbf{c}^n = \mathbf{T}(\mathbf{C}^n)$,

$$\frac{d\mathbf{C}}{dt} = \underbrace{[\mathbf{T}(\mathbf{C}) - \mathbf{c}^n]}_{\tilde{\mathbf{T}}(\mathbf{C})} + \underbrace{[\mathbf{\Omega}(\mathbf{C}) + \mathbf{c}^n]}_{\tilde{\mathbf{\Omega}}(\mathbf{C})}, \quad (20)$$

and the symmetric composition $\tilde{\mathcal{L}}_{\text{flow}}^{\Delta t/2} \circ \tilde{\mathcal{L}}_{\text{chem}}^{\Delta t} \circ \tilde{\mathcal{L}}_{\text{flow}}^{\Delta t/2}$ is applied to the balanced sub-problems. Since $\mathbf{C}^n$ is an equilibrium point of $\tilde{\mathbf{T}}$, the first half-step reduces to the identity and the scheme takes the two-stage form

$$\mathbf{C}^{n+1} = \tilde{\mathcal{L}}_{\text{flow}}^{\Delta t/2} \circ \tilde{\mathcal{L}}_{\text{chem}}^{\Delta t} (\mathbf{C}^n), \quad (21)$$

where $\tilde{\mathcal{L}}_{\text{chem}}^{\Delta t}$ integrates $d\mathbf{C}/dt = \tilde{\mathbf{\Omega}}(\mathbf{C})$ over the full time step and $\tilde{\mathcal{L}}_{\text{flow}}^{\Delta t/2}$ integrates $d\mathbf{C}/dt = \tilde{\mathbf{T}}(\mathbf{C})$ over $\Delta t/2$. The balancing constant is the fully discrete transport residual, evaluated once at the beginning of the step with the same spatial discretization and boundary treatment as the flow operator, and stored. The resulting scheme retains second-order accuracy and, compared to Eq. (19), it requires a single implicit chemistry solution per time step in place of two.

In both formulations the flow sub-problems are advanced with the explicit Runge–Kutta scheme, while the chemistry sub-problems employ an implicit Rosenbrock solver. At each grid point, the chemistry sub-step formally integrates the full state vector $\mathbf{C} \in \mathbb{R}^{N_s+4}$ over the splitting interval τ by solving the autonomous system of $N_s + 4$ equations:

$$\frac{d\mathbf{C}}{dt} = \mathbf{f}(\mathbf{C}), \quad (22)$$

where $\mathbf{f} = \mathbf{\Omega}$ for the Strang scheme and $\mathbf{f} = \tilde{\mathbf{\Omega}}$ for the simpler splitting. The chemical source vector contains the species net production rates $\dot{\omega}_m$ in the first N_s components, while the four components corresponding to momentum and total energy are identically zero; the balanced source is offset by the components of the frozen transport residual $\mathbf{c}^n = [c_1^n, \dots, c_{N_s}^n, c_{\rho u}^n, c_{\rho v}^n, c_{\rho w}^n, c_{\rho E}^n]^T$:

$$\mathbf{\Omega}(\mathbf{C}) = \begin{bmatrix} \dot{\omega}_1 \\ \vdots \\ \dot{\omega}_{N_s} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \tilde{\mathbf{\Omega}}(\mathbf{C}) = \begin{bmatrix} \dot{\omega}_1 + c_1^n \\ \vdots \\ \dot{\omega}_{N_s} + c_{N_s}^n \\ c_{\rho u}^n \\ c_{\rho v}^n \\ c_{\rho w}^n \\ c_{\rho E}^n \end{bmatrix}. \quad (23)$$

The equation system (22) is integrated pointwise with the four-stage, fourth-order Rosenbrock method (Press et al. 1993; Rosenbrock 1963). Each stage requires solving a linear system of the form $(\mathbf{I} - \gamma h \mathbf{J})\mathbf{k}_i = \mathbf{r}_i$, where $\mathbf{J} = \partial \mathbf{f} / \partial \mathbf{C}$ is the $(N_s + 4) \times (N_s + 4)$ Jacobian matrix of the chemical subsystem, evaluated via central finite differences,

and h is the substep size. Since the balancing constant $\mathbf{c}^n$ is frozen during the step, the two formulations share the same Jacobian matrix $\mathbf{J}$. Furthermore, due to the absence of chemical source terms for the momentum and energy equations, $\mathbf{J}$ exhibits a highly asymmetric block structure:

$$\mathbf{J} = \begin{pmatrix} \frac{\partial \dot{\omega}_1}{\partial(\rho Y_1)} & \cdots & \frac{\partial \dot{\omega}_1}{\partial(\rho Y_{N_s})} & \frac{\partial \dot{\omega}_1}{\partial(\rho u)} & \frac{\partial \dot{\omega}_1}{\partial(\rho v)} & \frac{\partial \dot{\omega}_1}{\partial(\rho w)} & \frac{\partial \dot{\omega}_1}{\partial(\rho E)} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \dot{\omega}_{N_s}}{\partial(\rho Y_1)} & \cdots & \frac{\partial \dot{\omega}_{N_s}}{\partial(\rho Y_{N_s})} & \frac{\partial \dot{\omega}_{N_s}}{\partial(\rho u)} & \frac{\partial \dot{\omega}_{N_s}}{\partial(\rho v)} & \frac{\partial \dot{\omega}_{N_s}}{\partial(\rho w)} & \frac{\partial \dot{\omega}_{N_s}}{\partial(\rho E)} \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (24)$$

As shown in Eq. (24), the bottom four rows of the Jacobian are identically zero, since the momentum and energy components of $\mathbf{f}$ do not depend on the state: they vanish for the Strang scheme and reduce to the frozen transport residual for the simpler splitting. Conversely, the last four columns are generally non-zero; perturbing the momentum $(\rho u, \rho v, \rho w)$ or the total energy (ρE) alters the mixture internal energy and, consequently, the mixture temperature. This variation directly impacts the reaction rate coefficients, thereby changing the species net production rates $\dot{\omega}_m$.

The Jacobian $\mathbf{J}$ is computed once per substep and factorized via LU decomposition with partial pivoting; the same factorization is reused across all four stages through forward and backward substitution. The method computes two solution estimates from the same set of stage vectors: a fourth-order primary solution $\mathbf{y}$ and an embedded third-order solution $\hat{\mathbf{y}}$, satisfying

$$\mathbf{y}_{\text{exact}} = \mathbf{y} + \mathcal{O}(h^5), \quad \hat{\mathbf{y}}_{\text{exact}} = \hat{\mathbf{y}} + \mathcal{O}(h^4), \quad |\mathbf{y} - \hat{\mathbf{y}}| = \mathcal{O}(h^4). \quad (25)$$

Their difference yields the local error vector $\boldsymbol{\varepsilon} = \sum_{i=1}^4 e_i \mathbf{k}_i$, with e_i the coefficients of the method (Press et al. 1993), and the substep is accepted when

$$\text{err}_{\max} = \max_m \left(\frac{|\varepsilon_m|}{\max(1, |y_m|) \eta} \right) < 1, \quad m = 1, \dots, N_s, \quad (26)$$

where η is a user-prescribed tolerance. Rejected substeps are retried with a reduced step size; accepted ones are followed by a step-size increase, both governed by the classical safety-factor power-law adaptation of Press et al. (1993). The pointwise nature of the integration makes the solver intrinsically parallelizable across grid cells, a structure that allows for full exploitation of the GPU potential.

Some final remarks concern the enforcement of mass conservation and the computational effort during chemistry integration. During the chemical substep, partial densities may locally drop below zero as a result of the integration of stiff production terms. Since the mixture density is not an independent variable but is recovered as

$\sum_m(\rho Y_m)$, negative values are eliminated by clipping and the clipped composition is rescaled by the factor $\alpha = \rho / \sum_m \max(\rho Y_m, 0)$, which restores the pre-clip mixture density exactly. This procedure enforces $Y_m \geq 0$ and $\sum_m Y_m = 1$, and conserves total mass by construction. Cells with no undershoot are not affected, so the additional cost is negligible. In terms of computational effort, the number of substeps required to cover the splitting interval is almost invariably limited to a few. In smooth regions of the flow a single substep is typically accepted, whereas in the most reactive cells, the adaptive step-size control settles on three to four substeps on average, with a maximum of 18 attempted substeps observed in the tests reported below. These figures naturally depend on the stiffness of the adopted kinetic mechanism and on the ratio between the fluid-dynamic time step and the local chemical timescales.

4 Validation

Following the implementation of the methodologies outlined in the previous sections, a comprehensive validation campaign was conducted to assess the accuracy and robustness of the extended solver. The validation strategy was based on the benchmark suites introduced by [Abdelsamie et al. \(2021\)](#) and [Martínez Ferrer et al. \(2014\)](#), who provided sets of standardized test cases ranging from 0D to 3D to assess the performance of multi-species reacting CFD solvers. A total of nine test cases were selected, covering both multicomponent transport and chemical kinetics. The multicomponent validation included: (i) the one-dimensional Sod shock tube problem, (ii) a pure diffusion problem, (iii) the propagation of acoustic waves in a binary mixture, (iv) a 2D single-species Taylor–Green Vortex (TGV), and (v) a 3D multicomponent TGV. The chemical kinetics module was tested using: (vi) a zero-dimensional constant-volume batch reactor, (vii) a one-dimensional reactive shock tube, (viii) a 1D laminar premixed flame, and (ix) a 3D multicomponent reacting TGV.

The batch reactor simulations were directly compared against results obtained with Cantera, while the other chemically reacting cases were validated against the reference data provided in [Abdelsamie et al. \(2021\)](#) and [Martínez Ferrer et al. \(2014\)](#).

The results reported are obtained with Strang splitting and characteristic-wise Lax–Friedrichs flux-vector splitting. The same tests were repeated using the simpler splitting and the HLLC solver, obtaining similar results.

4.1 Sod’s shock tube

The Sod shock tube is a standard benchmark for validating compressible flow solvers. This one-dimensional Riemann problem involves the solution of the Euler equations, where the evolution of the flow leads to the formation of a shock wave, a contact discontinuity, and a rarefaction wave.

The computational domain is defined by a unit length ($L_x = 1$), with the initial interface located at the center ($x = 0.5$). The initial states for the left (L) and right (R) regions are set to $(\rho_L, u_L, p_L) = (1, 0, 1)$ and $(\rho_R, u_R, p_R) = (0.125, 0, 0.1)$, respectively. The domain is discretized using 400 grid points, with a Courant-Friedrichs-Lewy (CFL) number of 0.5.

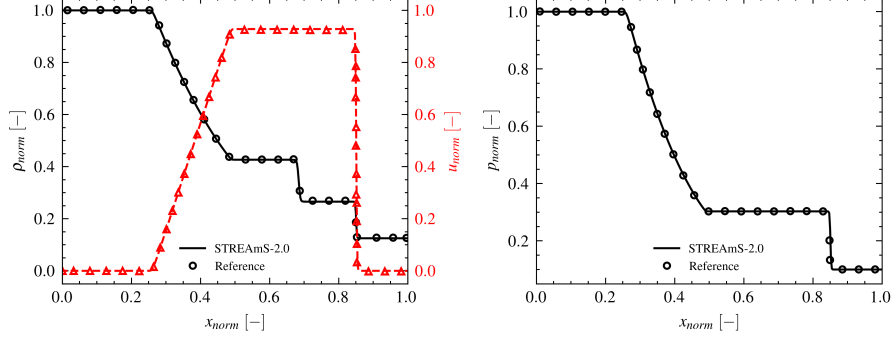


Fig. 1 Sod shock tube results at $t = 0.2$: comparison between the analytical solution reported by [Martínez Ferrer et al. \(2014\)](#) (markers) and STREAmS-2.0 results (lines). Density and velocity are shown on the left; pressure is reported on the right

Fig. 1 compares the numerical results obtained via STREAmS-2.0 with the analytical solution at $t = 0.2$, demonstrating excellent agreement across all physical variables.

4.2 Multicomponent diffusion

This test case is designed to validate the implementation of molecular transport terms within the Navier–Stokes equations for multicomponent gas mixtures. The simulation is conducted on a one-dimensional domain of length 0.05 m, discretized with 200 uniformly spaced grid points. A CFL number of 0.5 is maintained, and periodic boundary conditions are applied at both ends of the domain.

The initial velocity is set to zero, and the pressure is initialized at 101325 Pa. The initial temperature and species mass fractions follow a smooth spatial distribution governed by the following relations:

$$Y_k(x) = Y_{k,o} + (Y_{k,f} - Y_{k,o})f(x), \quad k = \text{O}_2, \text{N}_2, \text{H}_2\text{O}, \text{CH}_4, \quad (27)$$

$$T(x) = T_o + (T_f - T_o)f(x), \quad (28)$$

$$f(x) = 1 - \exp\left[-\frac{(x - x_0)^2}{d^2}\right], \quad (29)$$

where $d = 2.5 \cdot 10^{-3}$ m and $x_0 = 25 \cdot 10^{-3}$ m. Table 1 summarizes the temperature and mass fraction values at the boundaries.

The simulation results at $t = 50$ ms and 500 ms are presented in Fig. 2. The excellent agreement between the numerical results and the reference data confirms that the implemented multicomponent diffusion model is accurate and reliable.

4.3 Propagation of an acoustic wave

This benchmark is used to assess the solver’s ability to capture wave propagation phenomena in multicomponent gas mixtures. The test case simulates the evolution of a small-amplitude acoustic perturbation in a one-dimensional periodic domain of

Variable	Fuel side	Oxidizer side
T [K]	320	1350
Y_{O_2}	0.195	0.142
Y_{N_2}	0.591	0.758
$Y_{\text{H}_2\text{O}}$	0.0	0.1
Y_{CH_4}	0.214	0.0

Table 1 Temperature and mass fractions for the fuel and oxidizer regions

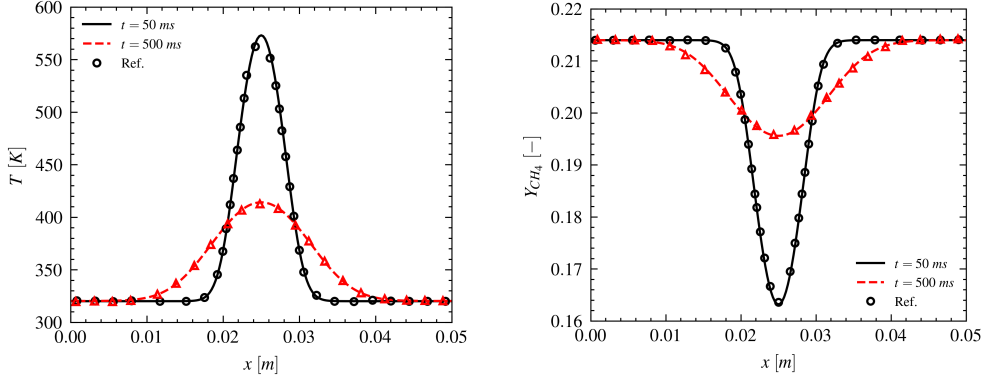


Fig. 2 Multicomponent diffusion test results. Comparison between [Martínez Ferrer et al. \(2014\)](#) data (markers) and STREAMS-2.0 output (lines). Temperature is shown on the left, and methane mass fraction is shown on the right

length $L = 5$ mm, initially containing the oxidizer mixture specified in Table 1. The initial disturbance is defined by the following relations:

$$u = 0.01a_0 \exp \left[-\frac{(x - x_0)^2}{d^2} \right], \quad \rho = \rho_0 + \frac{\rho_0}{a_0} u, \quad (30)$$

$$p = p_0 + \rho_0 a_0 u, \quad T = \frac{pW}{\rho R_0}, \quad (31)$$

where $\rho_0 = 0.24$ kg/m³, $a_0 = 734.6$ m/s, and $p_0 = 1$ atm denote the density, speed of sound, and pressure of the unperturbed state, respectively. The perturbation is centered at $x_0 = L/2$, with the spatial width parameter set to $d = L/10$.

The domain is discretized using 200 grid points with a CFL number of 0.5. Fig. 3 illustrates the temporal evolution of density and velocity at the fixed location $x = 0$ m. The results obtained with STREAMS-2.0 are compared against the reference data from [Martínez Ferrer et al. \(2014\)](#), showing good agreement.

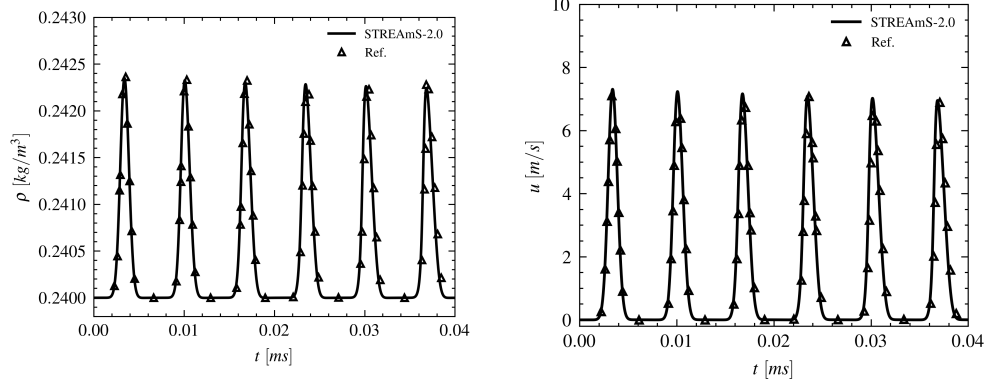


Fig. 3 Acoustic wave propagation results. Comparison between [Martínez Ferrer et al. \(2014\)](#) data (markers) and STREAmS-2.0 predictions (solid lines). Temporal evolution of density (left) and velocity (right) at $x = 0$ m

4.4 Two-dimensional single-species non-reacting TGV

This test case involves the well-documented 2D TGV ([Abdelsamie et al. 2016](#); [Laizet and Lamballais 2009](#)). This configuration serves as a fundamental verification step for 2D domains due to the availability of an analytical solution ([Taylor and Green 1937](#)). To ensure consistency with the initial conditions in [Abdelsamie et al. \(2021\)](#), a square domain of size $[0, L]^2$ with periodic boundary conditions is employed, where $L = 2\pi L_0$ and $L_0 = 1$ m. The domain is assumed to contain air as a single species with an initial temperature $T_0 = 288.15$ K and pressure $p_0 = 2368.5$ Pa, resulting in a kinematic viscosity of $\nu = 6.25 \cdot 10^{-4}$ m²/s. The initial velocity components are defined as:

$$\begin{cases} u(x, y, 0) = u_0 \sin(x/L_0) \cos(y/L_0), \\ v(x, y, 0) = -u_0 \cos(x/L_0) \sin(y/L_0), \end{cases} \quad (32)$$

where $u_0 = 1$ m/s. The initial flow field consists of four vortices, one in each quadrant of the domain, as illustrated in Fig. 4. The reference vortex turnover time is $\tau_{ref} = L_0/u_0 = 1$ s. The physical parameters L_0 , u_0 , and ν were selected to yield a Reynolds number $Re = u_0 L_0/\nu = 1600$, ensuring that the flow remains laminar over the entire simulation. The numerical results are compared against the following analytical solution:

$$\begin{cases} u(x, y, t) = u(x, y, 0) e^{-2\nu t/L_0^2}, \\ v(x, y, t) = v(x, y, 0) e^{-2\nu t/L_0^2}. \end{cases} \quad (33)$$

A uniform grid of 64^2 cells was employed with a constant time step of $\Delta t = 5 \cdot 10^{-4}$ s, covering a total integration time of $t = 10\tau_{ref}$. The velocity profiles along the domain centerlines at the final time ($t = 10\tau_{ref}$) are illustrated in Fig. 5, showing the spatial distribution of the u and v components. To further assess the accuracy of the solver, Table 2 provides a quantitative comparison of the maximum velocity magnitude. The

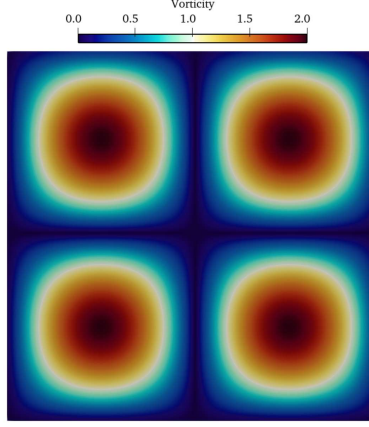


Fig. 4 2D TGV: initial vorticity at $t = 0$

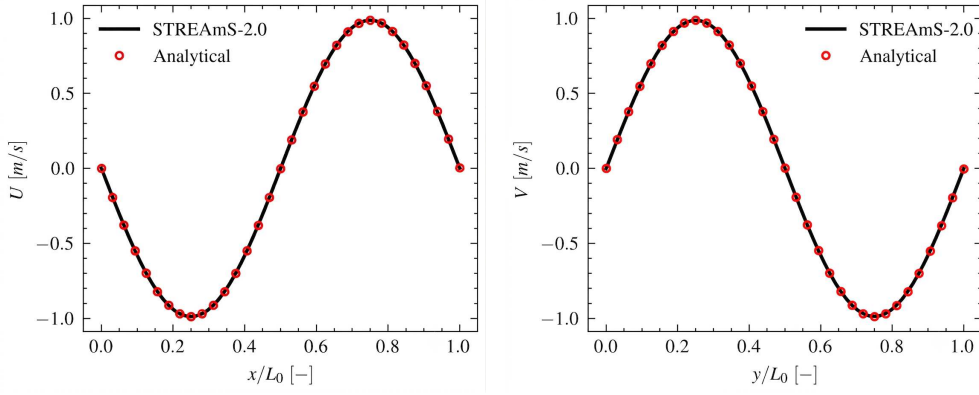


Fig. 5 Velocity profiles along the domain centerlines: u -velocity component on the left and v -velocity component on the right at $t = 10\tau_{ref}$

results obtained with STREAmS-2.0 are evaluated against both the analytical solution and the YALES2 code, with the corresponding relative errors reported to highlight the solver's accuracy.

As illustrated in Fig. 5, the STREAmS-2.0 results show good alignment with the analytical solution. The high fidelity of the current implementation is further confirmed by the data in Table 2, where the relative error for the maximum velocity remains low, specifically on the order of $10^{-3}\%$.

4.5 3D multicomponent non-reacting TGV

The objective of this configuration is to extend the physical complexity of the previous test case by considering a 3D multicomponent gaseous mixture with varying local compositions, while maintaining frozen chemistry. This setup allows for a rigorous

	Analytical	STREAmS-2.0	YALES2
U_{max}	0.987577	0.987604	0.987583
ϵ_{rel} [%]	(Ref)	$2.6 \cdot 10^{-3}$	$6.1 \cdot 10^{-4}$

Table 2 Quantitative comparison of maximum velocity and relative error between the analytical solution, STREAmS-2.0, and YALES2

assessment of the numerical models governing species transport and heat diffusion. Thermodynamic and transport properties are derived from the 9-species mechanism of Boivin et al. (2011), although chemical source terms remain suppressed. The computational domain consists of a cubic box of size $[0, L]^3$ with $L = 2\pi L_0$, where $L_0 = 1$ mm. This reduced length scale, compared to the previous single-species case, is selected to ensure adequate resolution of the hydrogen oxidation reaction fronts that will be introduced in subsequent investigations. Periodic boundary conditions are applied in all spatial directions. The initial velocity field is imposed as follows:

$$\begin{cases} u(x, y, z, 0) = u_0 \sin(x/L_0) \cos(y/L_0) \cos(z/L_0), \\ v(x, y, z, 0) = -u_0 \cos(x/L_0) \sin(y/L_0) \cos(z/L_0), \\ w(x, y, z, 0) = 0, \end{cases} \quad (34)$$

with $u_0 = 4$ m/s as reference velocity.

The initial field is characterized by a central fuel region composed of a H_2/N_2 mixture with a hydrogen mass fraction of $Y_{H_2}^0 = 0.0556$ at $T = 300$ K. The remainder of the domain is filled with an oxidizer (air) consisting of $Y_{O_2}^0 = 0.233$ at $T = 300$ K. Nitrogen is employed as the species complement throughout the domain, such that $Y_{N_2} = 1 - Y_{H_2} - Y_{O_2}$. To prevent numerical instabilities associated with sharp gradients, the interfaces between the fuel and oxidizer regions are smoothed using hyperbolic tangent function:

$$R_d(x) = |x - 0.5L|, \quad (35)$$

$$\psi(x) = 0.5 \left[1 + \tanh \left(\frac{\sigma(R_d(x) - R)}{R} \right) \right], \quad (36)$$

$$Y_{H_2}(x) = Y_{H_2}^0 (1 - \psi(x)), \quad (37)$$

$$Y_{O_2}(x) = Y_{O_2}^0 \psi(x). \quad (38)$$

where $R = \pi/4$ mm and $\sigma = 3$ denote the half-width of the central slab and the stiffness parameter, respectively. Consequently, the initial field contains a thin transition region where fuel and oxidizer coexist.

Finally, a non-homogeneous temperature profile is imposed to evaluate the solver's performance regarding heat diffusion, utilizing the following procedure:

1. At each grid point, the equilibrium temperature for the local mixture is computed under constant pressure and enthalpy conditions using Cantera.

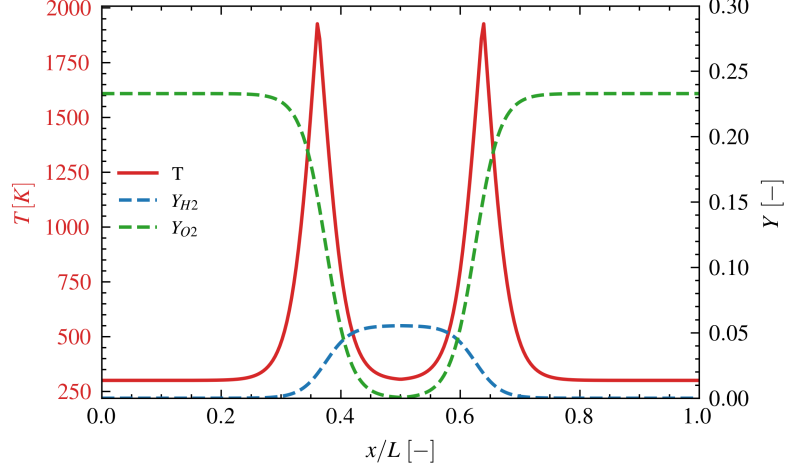


Fig. 6 Initial profiles of temperature and species mass fractions along the domain centerline ($y = 0.5L, z = 0.5L$) for the 3D non-reacting TGV case

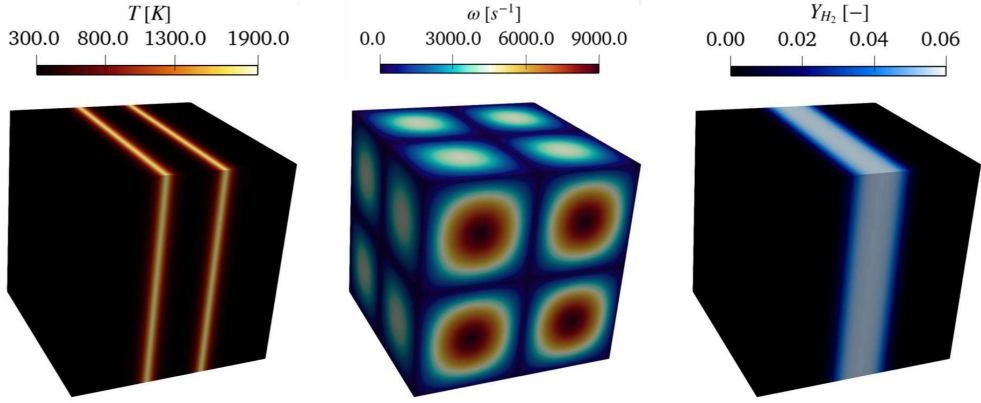


Fig. 7 Initial conditions for the 3D multicomponent TGV: spatial distribution of temperature (left), vorticity magnitude (center), and H_2 mass fraction (right)

2. The resulting temperature profile is enforced throughout the domain, while the species mass fractions are kept at their initial values, as defined in Eqs. (37) and (38).

Following this initialization, the peak adiabatic temperature at the fuel/air interface is found to be $T_{ad} = 1927.94$ K. This value exhibits a slight deviation from the one reported in [Abdelsamie et al. \(2021\)](#), likely due to differences in the underlying thermodynamic databases or specific implementation details of the equilibrium calculation. The initial profiles for this configuration are illustrated in Fig. 6 and Fig. 7.

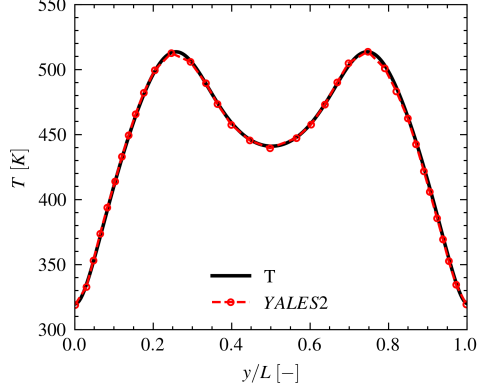


Fig. 8 Temperature profile at $t = 2\tau_{ref}$ along the domain centerline ($x = 0.5L, z = 0.5L$) for the 3D non-reacting 9-species case

The initial configuration replicates the complexities of a real flame, characterized by multi-species distribution, high temperature gradients, and sharp scalar profiles. As previously noted, the chemical source terms are suppressed for this simulation to isolate and assess the convection and diffusion processes.

To ensure a consistent benchmark with the results reported in [Abdelsamie et al. \(2021\)](#), which utilizes a constant Lewis number approach to facilitate comparison with DNS codes lacking an advanced transport model, a dedicated diffusion model was implemented. The diffusion coefficient for each species m is defined as follows:

$$D_m = \frac{\lambda}{\rho C_p Le_m}, \quad (39)$$

where λ and C_p represent the thermal conductivity and the specific heat at constant pressure of the mixture, respectively, while Le_m denotes the constant Lewis number for the m -th species, as specified in [Abdelsamie et al. \(2021\)](#) and listed in Table 3. Regarding the numerical setup, a CFL number of 0.25 was adopted. The domain is discretized using a uniform grid of 256^3 points, resulting in a spatial resolution of $\Delta x \approx 25 \mu\text{m}$. The reference time scale is defined as $\tau_{ref} = L_0/u_0 = 0.25$ ms, and the simulation was integrated over a total physical time of $t = 0.5$ ms (equivalent to $2\tau_{ref}$).

The simulation results are compared against reference data from the YALES2 solver. Fig. 8 and Fig. 9 illustrate the temperature and velocity profiles, respectively, while Fig. 10 presents the distributions of the mass fractions O_2 and H_2 along the centerline of the domain.

The comparison reveals excellent agreement between the two datasets, particularly in species concentrations, validating the solver's ability to accurately handle 3D multi-component flows. Minor oscillations observed in the velocity profiles are attributed to the compressible nature of the current solver in a low-Mach regime.

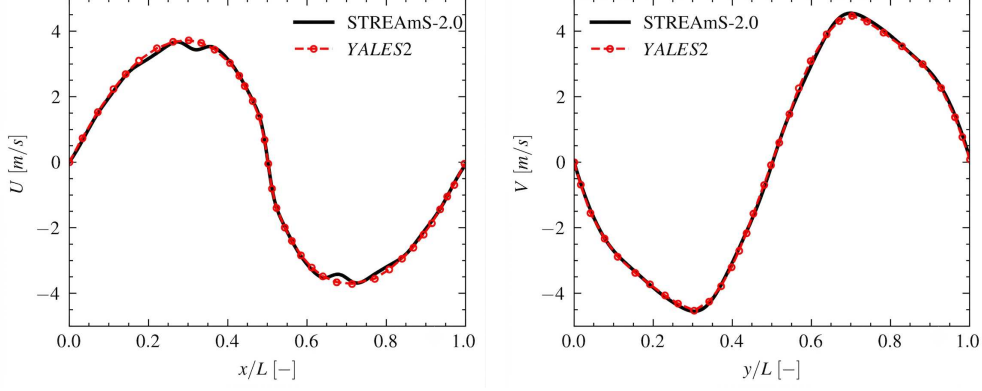


Fig. 9 Velocity profiles at $t = 2\tau_{ref}$ for the 3D non-reacting 9-species case: u -velocity along the centerline ($y = 0.5L, z = 0.5L$) on the left, and v -velocity along the centerline ($x = 0.5L, z = 0.5L$) on the right

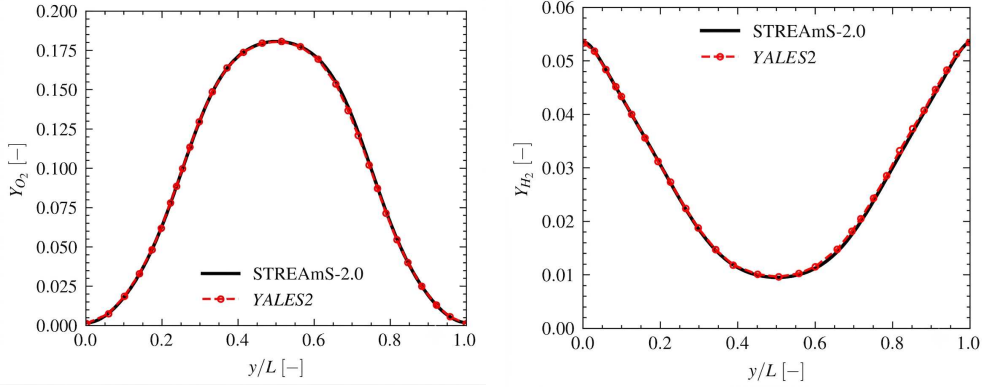


Fig. 10 Species mass fraction profiles at $t = 2\tau_{ref}$ along the centerline ($x = 0.5L, z = 0.5L$) for the 3D non-reacting 9-species case: oxygen on the left and hydrogen mass fraction on the right

4.6 Constant-volume batch reactor

This test case is designed to validate the implementation of the chemical source terms $\dot{\omega}_m$ within the Navier–Stokes equations by isolating the chemical kinetics from transport phenomena. The simulation reproduces the constant-volume combustion of a hydrogen-oxygen mixture, and the results obtained with STREAmS-2.0 are compared against a reference solution generated using Cantera.

The initial conditions and the kinetic mechanism follow the setup described by [Kee et al. \(1996\)](#), involving 11 species and 23 reactions. The mixture consists of $\text{H}_2/\text{O}_2/\text{N}_2$ with a molar ratio of 2/1/7, respectively, at an initial temperature of $T = 1000$ K and

Species	H_2	H	O_2	OH	O	H_2O	HO_2	H_2O_2	N_2
Le	0.3290	0.2228	1.2703	0.8279	0.8128	1.0741	1.2582	1.2665	1.8268

Table 3 Lewis numbers for the species involved in the kinetic scheme of [Boivin et al. \(2011\)](#)

a pressure of $p = 1$ atm. The simulation is carried out up to a final time of $t = 0.25$ ms. Fig. 11 illustrates the temporal evolution of the temperature (left) and the mass fractions of a representative subset of species (right), including the primary reactant (H_2), the main product (H_2O), and two key reactive intermediates (O and OH). The STREAMS-2.0 results match the data obtained from Cantera, confirming the accuracy and reliability of the chemical kinetics module in the absence of convective and diffusive transport.

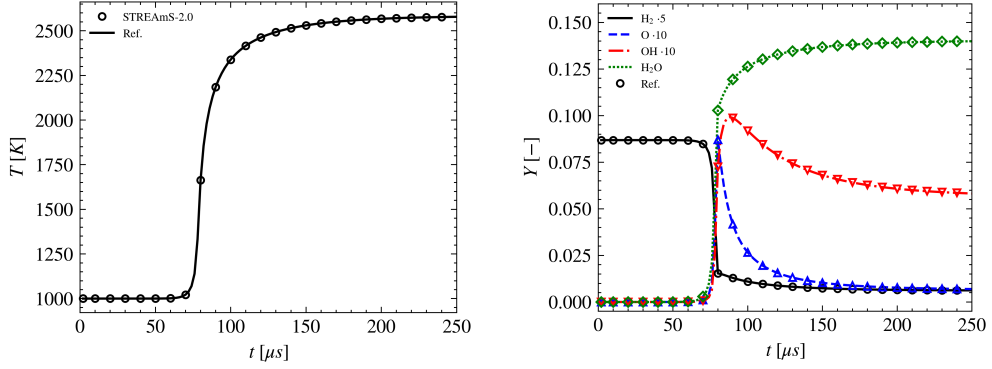


Fig. 11 Constant-volume batch reactor results. Comparison between STREAMS-2.0 predictions (lines) and [Goodwin et al. \(2025\)](#) data (symbols). Left: temperature evolution; Right: mass fraction profiles of selected species ($H_2 \cdot 10$, H_2O , O , and $OH \cdot 10$)

4.7 Multi-species reactive shock tube

This test case is intended to validate the solver’s ability to capture the coupling between high-speed convection and chemical kinetics. Furthermore, it assesses the robustness of the numerical schemes in the presence of strong pressure gradients within a reactive medium. The scenario involves shock reflection in a closed tube, which triggers a reaction wave that subsequently merges with the reflected shock.

The reactive mixture consists of $H_2/O_2/Ar$ with a molar ratio of 2/1/7. The initial states are defined as $(\rho, u, p)_L = (0.072 \text{ kg/m}^3, 0 \text{ m/s}, 7173 \text{ Pa})$ and $(\rho, u, p)_R = (0.18075 \text{ kg/m}^3, -487.34 \text{ m/s}, 35594 \text{ Pa})$. The 12 cm-long one-dimensional domain is discretized into 400 grid points, with the initial discontinuity located at the center.

An adiabatic wall boundary condition is applied at $x = 0$ cm, while a zero-gradient (outflow) condition is imposed at $x = 12$ cm. The chemical kinetics are modeled using the mechanism by [Kee et al. \(1988\)](#), comprising 9 species and 18 reactions. Fig. 12 and

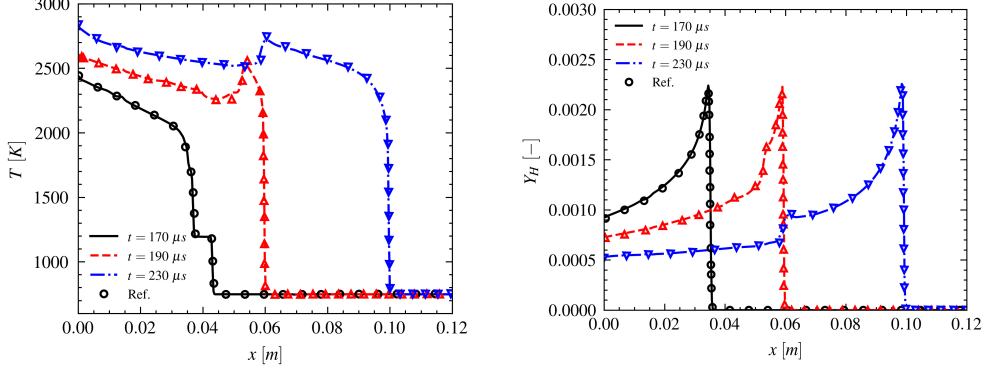


Fig. 12 Reactive shock tube results: comparison between STREAMS-2.0 (lines) and Martínez Ferrer et al. (2014) data (markers). Left: temperature; Right: H_2 mass fraction. Results are shown at $t = 170 \mu s$ (black), $t = 190 \mu s$ (red), and $230 \mu s$ (blue)

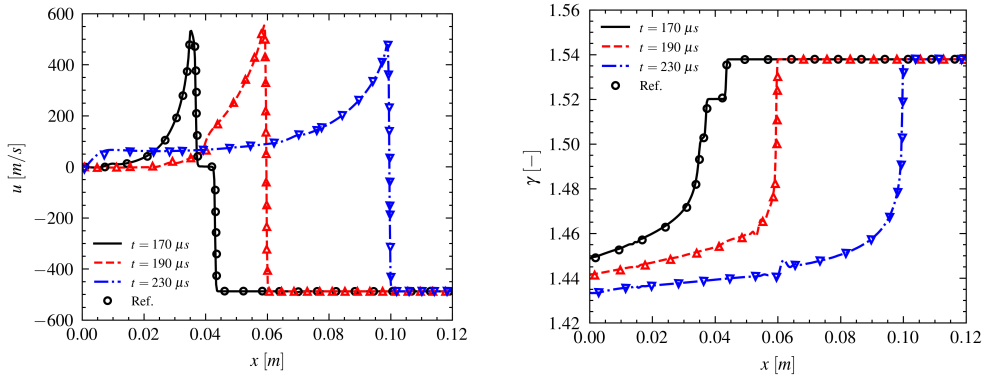


Fig. 13 Reactive shock tube results: comparison between STREAMS-2.0 (lines) and Martínez Ferrer et al. (2014) data (markers). Left: velocity; Right: specific heat ratio. Results are shown at $t = 170 \mu s$ (black), $t = 190 \mu s$ (red), and $230 \mu s$ (blue)

Fig. 13 present a comparison between the STREAMS-2.0 predictions and the reference data from Martínez Ferrer et al. (2014). The analysis includes the spatial distribution of temperature, hydrogen mass fraction, velocity, and the ratio of specific heats (γ) at three time instances: $t = 170 \mu s$, $190 \mu s$, and $230 \mu s$. The numerical results, show excellent agreement with the reference data, confirming the solver's accuracy and stability in simulating reactive, shock-dominated flows.

4.8 One-dimensional hydrogen/oxygen laminar premixed flame

This benchmark aims to evaluate the solver's performance in scenarios where convection, molecular diffusion, and chemical reaction rates are of comparable magnitudes. We consider the combustion of a lean hydrogen-oxygen mixture with an equivalence

ratio of $\phi = 0.6$, using H_2 as fuel and an O_2/N_2 mixture as the oxidizer. The chemical kinetics are modeled using the mechanism of [O Conaire et al. \(2004\)](#), which includes 10 species and 21 reactions.

The fresh reactants are set at 298 K and 1 atm. The 3-cm-long computational domain is discretized using a uniform mesh with a resolution of $\Delta x = 15 \mu\text{m}$. Initial profiles for temperature, velocity, and species composition are derived from a steady-state solution obtained via Cantera. At the inlet ($x = 0 \text{ cm}$), cold reactant properties are imposed with an inflow velocity matching the laminar flame speed, while zero-gradient conditions are applied at the outlet ($x = 3 \text{ cm}$).

The simulation was conducted for a physical time of $t = 1 \text{ ms}$. Fig. 14 displays the resulting temperature, density, and species profiles as a function of the progress variable $c = (T - T_u)/(T_b - T_u)$, where T_u and T_b denote the temperatures of the unburnt and burnt mixtures, respectively. The results indicate that the flame front is fully stabilized within the domain, with all profiles reaching a steady state. This confirms the solver’s ability to sustain a steady laminar flame when the inflow velocity balances the flame speed.

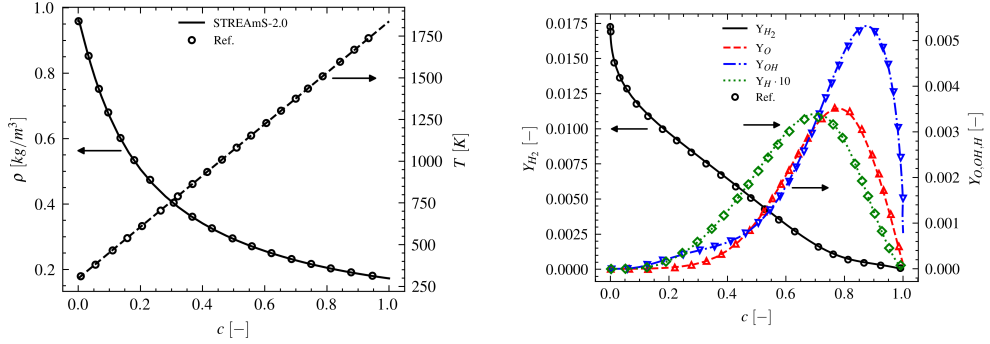


Fig. 14 Laminar premixed flame results at $t = 1 \text{ ms}$. Left: temperature and density; Right: mass fractions of selected species. STREAmS-2.0 results are shown as lines, while markers represent the reference data reported by [Martínez Ferrer et al. \(2014\)](#). Arrows indicate the reference axes for the respective curves

4.9 3D multicomponent reacting TGV

In this final benchmark, the 3D multi-species configuration is repeated with chemical kinetics activated. The chemical evolution is governed by the mechanism of [Boivin et al. \(2011\)](#), which includes 9 species and 12 reversible reactions. The initial state for this test case is illustrated in Fig. 15. While these conditions are qualitatively similar to the non-reacting case shown in Fig. 6, the species profiles at the fuel/oxidizer interface are fundamentally different.

Specifically, the local species concentrations are initialized based on the thermochemical equilibrium values corresponding to the mixtures defined by Eqs. (37) and (38) at $T_0 = 300 \text{ K}$ and atmospheric pressure. This initialization strategy ensures that the

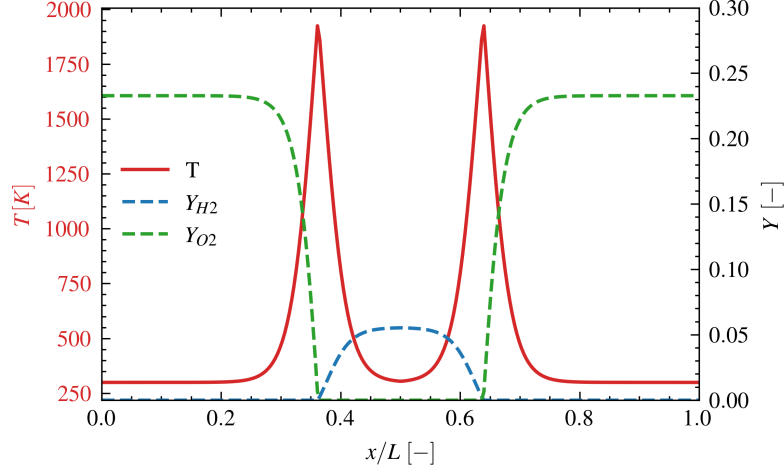


Fig. 15 Initial profiles of temperature and species mass fractions along the domain centerline ($y = 0.5L, z = 0.5L$) for the 3D reacting TGV case

starting conditions closely mimic a real flame front, featuring a locally consistent composition and temperature field. As noted in the previous section, slight discrepancies in the peak temperature were observed when using Cantera compared to the reference values in [Abdelsamie et al. \(2021\)](#).

Periodic boundary conditions are maintained in all directions, effectively treating the domain as a closed system with no net flux of mass, momentum, or energy. To ensure a consistent comparison with the reference DNS data, the transport model remains based on the constant Lewis numbers reported in [Table 3](#). All other physical and numerical parameters, including the grid resolution and the CFL number, set to 0.25, remain identical to the non-reacting setup.

The simulation was integrated for a physical time of $t = 0.5$ ms ($2\tau_{ref}$). The results obtained with the current solver are presented in [Fig. 16](#), where temperature, velocity components, and the mass fractions of Y_{O_2} , Y_{H_2} , and Y_{OH} are compared against the results from the YALES2 code. Finally, the three-dimensional fields of temperature, vorticity, and H_2 mass fraction after $t = 2\tau_{ref}$ at $z = 0.5L$ are shown in [Fig. 17](#).

The comparison reveals a good agreement between the two datasets across all measured variables. Minor discrepancies are observed in the velocity profiles, consistent with the compressible nature of the current solver as previously noted.

5 Performance analysis and profiling

Scalability is a critical performance metric for GPU-accelerated solvers, directly impacting their execution efficiency in large-scale scientific computing. In this context, rigorous scaling assessments are essential to verify that the solver fully exploits modern multi-GPU hardware architectures while minimizing parallelization overheads. Performance degradation in such parallel systems is typically attributed to resource underutilization, computational bottlenecks, or communication latencies.

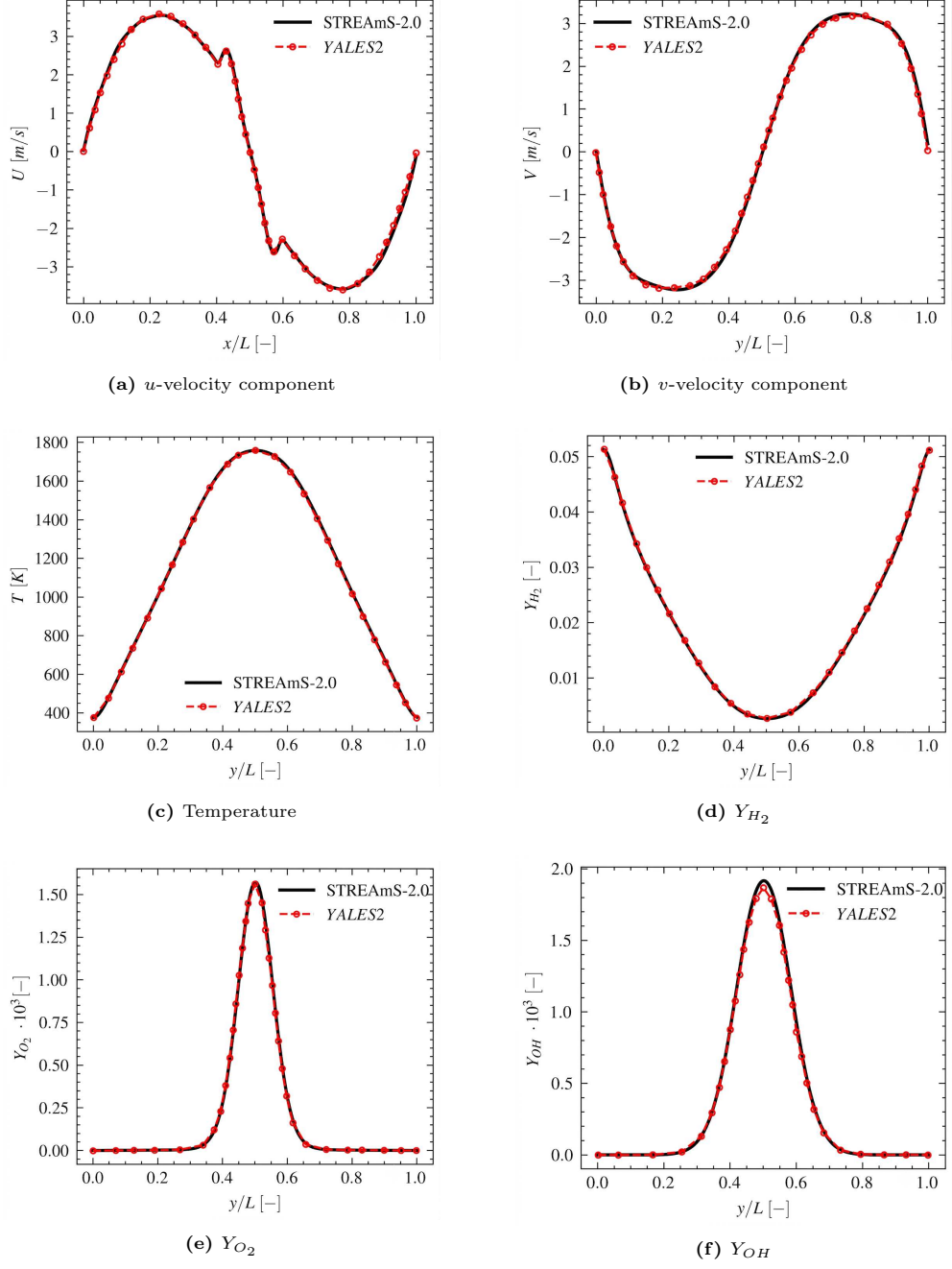


Fig. 16 Results of the 3D reacting multicomponent TGV at $t = 2\tau_{ref}$ compared with YALES2 data: (a) u -velocity at $y = 0.5L, z = 0.5L$, (b) v -velocity, (c) temperature, (d) Y_{H_2} , (e) Y_{O_2} , and (f) Y_{OH} at $x = 0.5L, z = 0.5L$

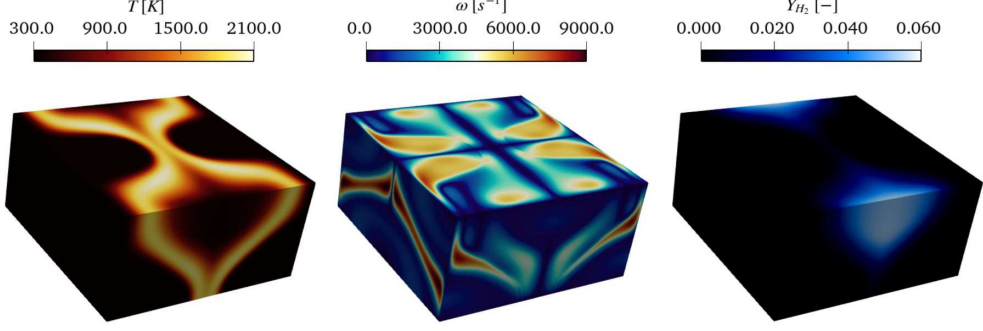


Fig. 17 Three-dimensional view of temperature, vorticity and Y_{H_2} fields at $z = 0.5L$ at $t = 2\tau_{ref}$

5.1 Strong scaling analysis

To evaluate the strong scalability of STREAmS-2.0, a series of simulations was performed on the Leonardo supercomputer by varying the computational resources from 1 to 64 NVIDIA A100 SXM4 64GB GPUs, with 4 GPUs allocated per compute node. Two distinct configurations were considered: a non-reacting case featuring multicomponent transport only, and a reacting case including full chemical kinetics. Strong scaling evaluates the reduction in execution time as the number of computational units increases for a fixed total problem size. In these tests, the mesh size was fixed at $68 \cdot 10^6$ cells for the non-reacting configuration and $53.4 \cdot 10^6$ cells when chemical kinetics were enabled. This difference in grid density was introduced to maintain a comparable percentage of memory usage between the two cases; indeed, the evaluation of chemical source terms $\dot{\omega}_m$ requires additional storage for intermediate parameters, thereby increasing both the memory footprint and the volume of data transferred per cell. Parallel performance is quantified using the speed-up (S_s) and the parallel efficiency (E_S), defined as:

$$S_s = \frac{t_1}{t_N}, \quad (40)$$

$$E_S = \frac{S_s}{N} \cdot 100 = \frac{t_1}{t_N \cdot N} \cdot 100, \quad (41)$$

where t_1 and t_N represent the execution times on 1 and N GPUs, respectively. In the ideal scenario, S_s scales linearly with N ($S_s = N$) and E_S remains constant at 100%. The speed-up, parallel efficiency, and GPU memory utilization for both the multi-component and reactive configurations are summarized in Tables 4 and 5, and further illustrated in Fig. 18 and Fig. 19.

A notable observation is the strict dependence of GPU parallel performance on memory utilization. In both test cases, high parallel efficiency is sustained as long as the per-GPU memory usage remains elevated. However, performance degrades noticeably when the number of GPUs exceeds 8. This decline is primarily driven by the progressive reduction of the local workload assigned to each device. As reported in Table 4, when allocating the problem across 64 GPUs, each unit utilizes only about

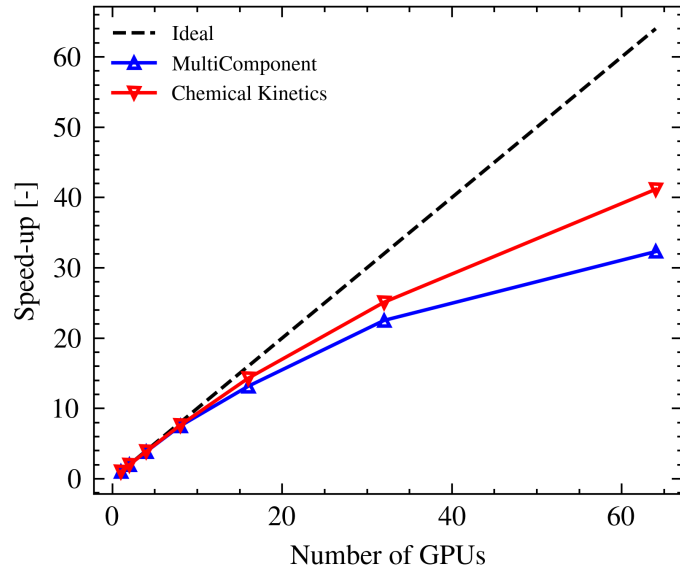


Fig. 18 Strong scaling speed-up with disabled (blue) and enabled (red) chemistry

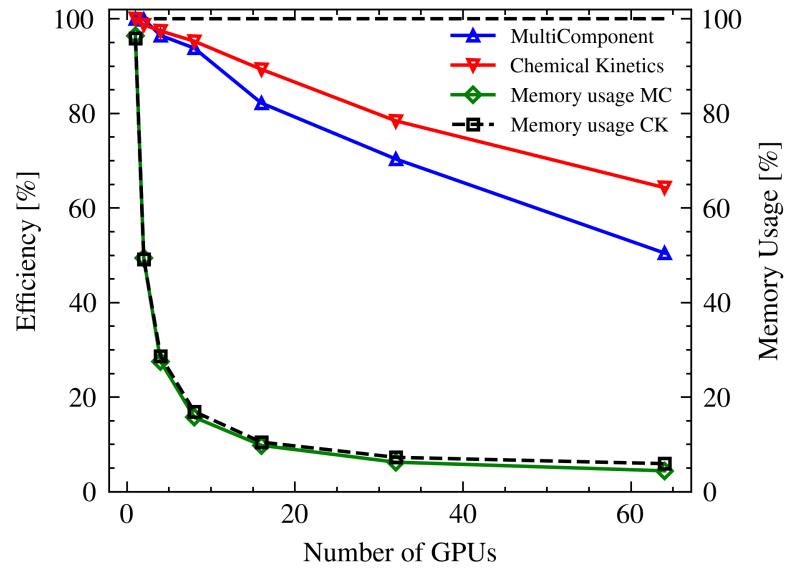


Fig. 19 Strong scaling efficiency and memory usage with disabled chemistry in blue and green lines, and with enabled chemistry using red and dotted black lines

GPUs	Avg time [s]	Speed-up [-]	Id. speed-up [-]	Eff. [%]	Mem. usage [%]
1	14.86	1	1	100	96.40
2	7.44	1.99	2	99.86	49.39
4	3.85	3.86	4	96.49	27.53
8	1.98	7.50	8	93.81	15.75
16	1.13	13.15	16	82.19	9.80
32	0.66	22.51	32	70.36	6.25
64	0.46	32.30	64	50.47	4.43

Table 4 Strong scaling speed-up, efficiency and memory usage with disabled chemistry

GPUs	Avg time [s]	Speed-up [-]	Id. speed-up [-]	Eff. [%]	Mem. usage [%]
1	20.58	1	1	100	95.89
2	10.42	1.97	2	98.75	49.18
4	5.28	3.90	4	97.44	28.66
8	2.70	7.62	8	95.27	16.97
16	1.44	14.21	16	89.32	10.49
32	0.82	25.10	32	78.43	7.28
64	0.50	41.16	64	64.31	5.94

Table 5 Strong scaling speed-up, efficiency and memory usage with enabled chemistry

4% of its available memory capacity. At such low occupancy levels, the computational workload within individual kernels is insufficient to hide data transfer latencies and communication overheads, resulting in a substantial drop in efficiency. These trends align with the established expectation that GPU-accelerated solvers require a minimum workload per device to achieve optimal streaming multiprocessor throughput. This analysis allows for the identification of a critical memory usage threshold below which resource exploitation becomes inefficient. Specifically, this threshold lies near 10% of the maximum memory capacity; below this limit, the communication-to-computation ratio becomes unfavorable, significantly compromising solver performance.

5.2 Weak scaling analysis

Weak scaling assesses the solver’s capacity to handle larger problem sizes scaled proportionally with the computational resources, thereby keeping the workload per GPU constant. In this framework, each device processes the same number of computational cells across all allocations. To evaluate weak scalability, performance was tested on the Leonardo supercomputer by scaling up to 512 NVIDIA A100 SXM4 64GB GPUs, with 4 GPUs allocated per compute node. To guarantee full memory utilization on each device, the total mesh size was scaled from $68 \cdot 10^6$ to $34.8 \cdot 10^9$ cells for the non-reacting cases, and from $53.4 \cdot 10^6$ to $27.3 \cdot 10^9$ cells when chemical kinetics were included.

GPUs	Avg time [s]	Ideal avg time [s]	Efficiency [%]
1	14.862	14.862	100
2	14.871	[-]	99.94
4	14.902	[-]	99.73
8	14.937	[-]	99.49
16	14.950	[-]	99.41
32	14.954	[-]	99.38
64	14.962	[-]	99.33
128	14.970	[-]	99.27
256	14.973	[-]	99.25
512	14.975	[-]	99.24

Table 6 Weak scaling efficiency with disabled chemistry

GPUs	Avg time [s]	Ideal avg time [s]	Efficiency [%]
1	20.583	20.583	100
2	20.522	[-]	100.29
4	20.602	[-]	99.90
8	20.622	[-]	99.81
16	20.635	[-]	99.74
32	20.642	[-]	99.71
64	20.647	[-]	99.69
128	20.655	[-]	99.65
256	20.659	[-]	99.63
512	20.671	[-]	99.57

Table 7 Weak scaling efficiency with enabled chemistry

The performance is quantified via the weak scaling parallel efficiency (E_W), defined as:

$$E_W = \frac{t_1}{t_N} \cdot 100, \quad (42)$$

where t_1 represents the average execution time on a single GPU and t_N is the corresponding time measured on N GPUs. The results are summarized in Tables 6 and 7, and graphically illustrated in Fig. 20.

As reported in Tables 6 and 7, the weak scaling efficiency consistently exceeds 99% across the entire investigated range, confirming the excellent scaling properties of the STREAmS-2.0 solver.

Furthermore, the results indicate a minor yet measurable improvement in parallel efficiency when chemical kinetics are activated. This trend is justified by the increased computational intensity associated with the evaluation of chemical source terms. When the per-GPU workload is substantial and inherently compute-bound, the hardware execution units are exploited more effectively because the ratio of floating-point operations to data movement increases. Consequently, this leads to an enhanced parallel performance, as the latency of memory transactions and interconnect communication is successfully masked by active thread computation.

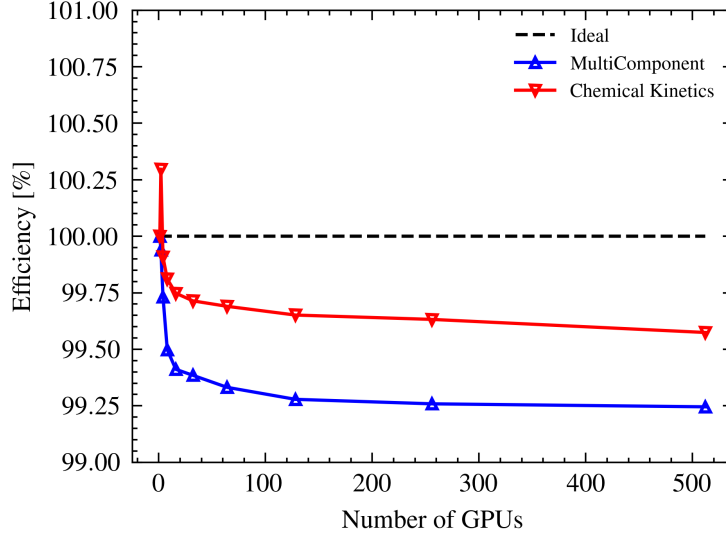


Fig. 20 Weak scaling efficiency with disabled (blue) and enabled (red) chemistry

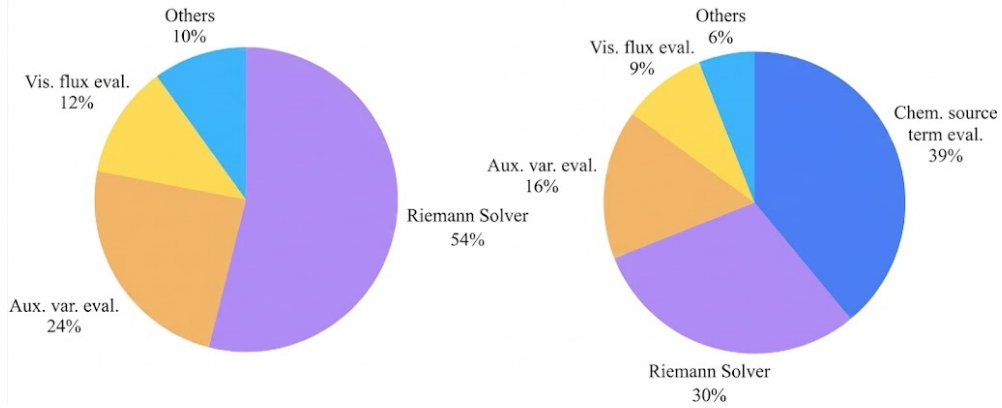


Fig. 21 Profiling results: chemistry disabled (left), chemistry enabled (right)

5.3 Profiling

Finally, code profiling was conducted under the same conditions as the scalability analysis to provide a comprehensive characterization of the solver's performance. Fig. 21 illustrates the percentage of computational time allocated to various subroutines per time step, comparing cases with and without active chemistry.

In the non-reacting case, most of the computational effort is spent on the Riemann solver for the Euler equations. Conversely, when chemistry is enabled, approximately 40% of the total simulation time is spent evaluating species production rates. Given the

inherent stiffness of the reacting Navier–Stokes equations and the need for an implicit integration scheme, these results are fully consistent with theoretical expectations and reflect the dominant role of chemical source-term computation in reactive flow simulations.

6 Conclusions

This work presented the extension of the in-house solver STREAMS-2.0 to handle multicomponent, chemically reacting compressible flows. The implementation was rigorously validated following the benchmark suites of [Abdelsamie et al. \(2021\)](#) and [Martínez Ferrer et al. \(2014\)](#), which provide established frameworks for the verification of reactive CFD solvers. The numerical results demonstrate excellent agreement with both analytical solutions and reference data, thereby confirming the reliability and accuracy of the newly implemented physical models. Furthermore, a comprehensive scalability analysis was conducted on the Leonardo supercomputer to evaluate the solver’s performance on modern HPC architectures. Strong-scaling tests on up to 64 NVIDIA A100 GPUs showed that high parallel efficiency is retained provided an adequate per-GPU workload is sustained, with a critical memory-occupancy threshold identified at approximately 10% of the device capacity. Weak-scaling tests demonstrated near-ideal behaviour, with parallel efficiency consistently above 99% on up to 512 GPUs and grids of up to 3.5×10^{10} cells. These results confirm that the combination of high-order shock-capturing numerics, general multicomponent chemistry, and implicit stiff integration can be achieved without compromising scalability on large-scale GPU systems. Future work will focus on a fully resolved DNS of a laboratory jet flame configuration, to be validated against experimental data as a final assessment step. Preliminary simulations, currently on coarse grids with reduced kinetic mechanisms, are underway to establish the computational setup and identify the required model refinements.

Declarations

Acknowledgements. The authors gratefully acknowledge the financial support provided by the European Space Agency (ESA) under the Technical Support Activities for the STS PNRR Project (PNRR CUP J58G21000020007).

The view expressed herein can in no way be taken to reflect the official opinion of the European Space Agency, the Presidency of the Council of Ministers of the Italian Republic, and the European Commission. Views and opinions expressed are those of the author(s) only, and the European Union/European Commission/ESA/Presidency of Council of Ministers of the Italian Republic cannot be held responsible for any use which may be made of the information contained therein.

Authors’ contribution. **E. F:** Conceptualization, Methodology, Software, Validation, Investigation, Visualization, and Writing - original draft. **M. F:** Conceptualization, Software, Validation. **A. N:** Project administration, Conceptualization, Funding acquisition. **M. V:** Project administration, Conceptualization, Supervision, Funding acquisition. **F. S:** Project administration, Conceptualization, Supervision, Funding

acquisition. **M. B:** Conceptualization, Software, Resources, Writing – review & editing. **P.P. C:** Conceptualization, Supervision, Writing – review & editing. All authors read and approved the final manuscript.

Funding. Funded by the European Union - NextGenerationEU and by the Presidency of the Council of Ministers pursuant to Article 1, paragraph 254, of Law 160/2019 and by the Presidency of the Council of Ministers from the Complementary Fund.

Data availability. The data that support the findings of this study are available from the corresponding author upon reasonable request.

Appendix A Eigenstructure at the Roe-averaged state

This appendix reports the derivation of the right and left eigenvectors of the Euler–Jacobian matrix used by the characteristic-wise flux-vector splitting of Sec. 3. The derivation follows the approach of [Shuen et al. \(1990\)](#), adapted to the formulation of Eqs. (11) and (12). The x -component of the convective flux reads

$$\mathbf{F} = \{\rho_1 u, \dots, \rho_{N_s} u, \rho u^2 + p, \rho uv, \rho uw, \rho u H\}^T, \quad (\text{A1})$$

where $\rho_i = \rho Y_i$ is the species mass density, $H = E + p/\rho$ is the total enthalpy, $E = e + q$ the specific total energy and $q = (u^2 + v^2 + w^2)/2$ the kinetic energy. To derive the Jacobian $A = \partial \mathbf{F} / \partial \mathbf{C}$, the internal energy must be expressed as a function of the conservative variables,

$$e = \frac{\rho E}{\sum_{i=1}^{N_s} \rho_i} - \frac{1}{2} \frac{(\rho u)^2 + (\rho v)^2 + (\rho w)^2}{\left(\sum_{i=1}^{N_s} \rho_i\right)^2}. \quad (\text{A2})$$

Introducing the pressure derivatives $p_e = (\partial p / \partial e)_{\rho_i}$ and $p_{\rho_i} = (\partial p / \partial \rho_i)_{\rho_{j \neq i}, e}$, the chain rule yields

$$\frac{\partial p}{\partial \rho u} = -p_e \frac{u}{\rho}, \quad \frac{\partial p}{\partial \rho v} = -p_e \frac{v}{\rho}, \quad \frac{\partial p}{\partial \rho w} = -p_e \frac{w}{\rho}, \quad \frac{\partial p}{\partial \rho E} = \frac{p_e}{\rho}, \quad (\text{A3})$$

$$\widehat{p}_{\rho_i} := \frac{\partial p}{\partial \rho_i} = p_{\rho_i} + p_e \frac{u^2 + v^2 + w^2 - E}{\rho}, \quad (\text{A4})$$

all derivatives being taken with the remaining conservative variables held fixed. Setting $N_s = 3$ for compactness, the Jacobian then reads

$$A = \begin{pmatrix} u(1-Y_1) & -Y_1u & -Y_1u & Y_1 & 0 & 0 & 0 \\ -Y_2u & u(1-Y_2) & -Y_2u & Y_2 & 0 & 0 & 0 \\ -Y_3u & -Y_3u & u(1-Y_3) & Y_3 & 0 & 0 & 0 \\ -u^2 + \hat{p}_{\rho_1} & -u^2 + \hat{p}_{\rho_2} & -u^2 + \hat{p}_{\rho_3} & u\left(2 - \frac{p_e}{\rho}\right) & -v\frac{p_e}{\rho} & -w\frac{p_e}{\rho} & \frac{p_e}{\rho} \\ -uv & -uv & -uv & v & u & 0 & 0 \\ -uw & -uw & -uw & w & 0 & u & 0 \\ u(\hat{p}_{\rho_1} - H) & u(\hat{p}_{\rho_2} - H) & u(\hat{p}_{\rho_3} - H) & H - u^2\frac{p_e}{\rho} & -uv\frac{p_e}{\rho} & -uw\frac{p_e}{\rho} & u\left(1 + \frac{p_e}{\rho}\right) \end{pmatrix}. \quad (\text{A5})$$

The derivatives p_e and p_{ρ_i} depend on the assumptions made on the mixture components. For a mixture of thermally perfect gases, the thermal EoS is

$$p = \left(\sum_{i=1}^{N_s} \frac{\rho_i}{W_i} \right) R_0 T = \rho \left[\left(\sum_{i=1}^{N_s} \frac{Y_i}{W_i} \right) R_0 \right] T = \rho R T, \quad (\text{A6})$$

with $R = R_0/\overline{W}$. Each species obeys the caloric EoS $de_i = C_{v,i}(T) dT$, and a frozen specific heat at constant volume can be defined for the mixture,

$$C_{v,f} = \left(\frac{\partial e}{\partial T} \right)_{\rho_i} = \sum_{i=1}^{N_s} Y_i C_{v,i}. \quad (\text{A7})$$

Exploiting the dependence on the temperature, we obtain

$$p_e = \left(\frac{\partial p}{\partial T} \right)_{\rho_i} \left(\frac{\partial T}{\partial e} \right)_{\rho_i} = \frac{\rho R}{C_{v,f}}. \quad (\text{A8})$$

As for p_{ρ_i} , differentiating the internal energy of the mixture, $\rho e = \sum_{k=1}^{N_s} \rho_k e_k(T)$, with respect to ρ_i at constant e and $\rho_{j \neq i}$ gives $(\partial T / \partial \rho_i)_{\rho_{j \neq i}, e} = (e - e_i) / (\rho C_{v,f})$, where e_i denotes the specific internal energy of the i -th species. Hence,

$$p_{\rho_i} = \frac{R_0 T}{W_i} + \rho R \left(\frac{\partial T}{\partial \rho_i} \right)_{\rho_{j \neq i}, e} = \frac{R_0 T}{W_i} + \frac{R}{C_{v,f}} (e - e_i). \quad (\text{A9})$$

The Jacobian has eigenvalues $(u - c, u, \dots, u, u + c)$, the species and shear modes being associated with the repeated eigenvalue u , where c is the frozen speed of sound, given for a generic EoS $p = p(\rho_i, e)$ by

$$c^2 = \sum_{i=1}^{N_s} Y_i p_{\rho_i} + \frac{p}{\rho^2} p_e, \quad (\text{A10})$$

which reduces to $c^2 = \gamma RT$, with $\gamma = C_p/C_v$ for a mixture of thermally perfect gases. In the flux-vector splitting, the eigenvector decomposition is evaluated at a Roe-type average of the two states adjacent to the interface. The velocity components, the total enthalpy, the internal energy and the mass fractions are averaged with weights $\sqrt{\rho}$,

$$\bar{\phi} = \frac{\phi_L + \sqrt{\rho_R/\rho_L} \phi_R}{1 + \sqrt{\rho_R/\rho_L}}, \quad \phi = u, v, w, H, e, Y_i, \quad (\text{A11})$$

the average density is $\bar{\rho} = \sqrt{\rho_L \rho_R}$, and the remaining quantities are derived from the averaged ones: $\bar{E} = \bar{e} + \bar{q}$, $\bar{p}/\bar{\rho} = \bar{H} - \bar{E}$, while temperature $\bar{T}$ is obtained by inverting the caloric EoS. The same average state is used in Appendix B, again with the overbar for averaged variables. The right and left eigenvectors can be expressed in terms of the frozen speed of sound c , the total enthalpy H , the total energy E and the pressure derivatives p_e and p_{ρ_i} , in the same form adopted in the HTR solver of [Di Renzo and Pirozzoli \(2021\)](#), which keeps the matrices valid for a generic EoS. For the sake of brevity, we only report the matrices for convection along x when $N_s = 3$ species are considered. With the eigenvalues ordered as above, and all quantities evaluated at the averaged state with the overbars omitted for readability, the right and left eigenvector matrices read

$$R_1 = \begin{pmatrix} Y_1 & 1 & 0 & 0 & 0 & 0 & Y_1 \\ Y_2 & 0 & 1 & 0 & 0 & 0 & Y_2 \\ Y_3 & 0 & 0 & 1 & 0 & 0 & Y_3 \\ u - c & u & u & u & 0 & 0 & u + c \\ v & v & v & v & 1 & 0 & v \\ w & w & w & w & 0 & 1 & w \\ H - uc & E - \frac{p_{\rho_1} \rho}{p_e} & E - \frac{p_{\rho_2} \rho}{p_e} & E - \frac{p_{\rho_3} \rho}{p_e} & v & w & H + uc \end{pmatrix}, \quad (\text{A12})$$

$$L_1 = \begin{pmatrix} \frac{1}{2} \left(\frac{u}{c} + \frac{\hat{p}_{\rho_1}}{c^2} \right) & \frac{1}{2} \left(\frac{u}{c} + \frac{\hat{p}_{\rho_2}}{c^2} \right) & \frac{1}{2} \left(\frac{u}{c} + \frac{\hat{p}_{\rho_3}}{c^2} \right) & -\frac{1}{2c} - \frac{p_e u}{2\rho c^2} & -\frac{p_e v}{2\rho c^2} & -\frac{p_e w}{2\rho c^2} & \frac{p_e}{2\rho c^2} \\ 1 - \frac{\hat{p}_{\rho_1} Y_1}{c^2} & -\frac{\hat{p}_{\rho_2} Y_1}{c^2} & -\frac{\hat{p}_{\rho_3} Y_1}{c^2} & \frac{p_e u Y_1}{\rho c^2} & \frac{p_e v Y_1}{\rho c^2} & \frac{p_e w Y_1}{\rho c^2} & -\frac{p_e Y_1}{\rho c^2} \\ -\frac{\hat{p}_{\rho_1} Y_2}{c^2} & 1 - \frac{\hat{p}_{\rho_2} Y_2}{c^2} & -\frac{\hat{p}_{\rho_3} Y_2}{c^2} & \frac{p_e u Y_2}{\rho c^2} & \frac{p_e v Y_2}{\rho c^2} & \frac{p_e w Y_2}{\rho c^2} & -\frac{p_e Y_2}{\rho c^2} \\ -\frac{\hat{p}_{\rho_1} Y_3}{c^2} & -\frac{\hat{p}_{\rho_2} Y_3}{c^2} & 1 - \frac{\hat{p}_{\rho_3} Y_3}{c^2} & \frac{p_e u Y_3}{\rho c^2} & \frac{p_e v Y_3}{\rho c^2} & \frac{p_e w Y_3}{\rho c^2} & -\frac{p_e Y_3}{\rho c^2} \\ -v & -v & -v & 0 & 1 & 0 & 0 \\ -w & -w & -w & 0 & 0 & 1 & 0 \\ \frac{1}{2} \left(\frac{\hat{p}_{\rho_1}}{c^2} - \frac{u}{c} \right) & \frac{1}{2} \left(\frac{\hat{p}_{\rho_2}}{c^2} - \frac{u}{c} \right) & \frac{1}{2} \left(\frac{\hat{p}_{\rho_3}}{c^2} - \frac{u}{c} \right) & \frac{1}{2c} - \frac{p_e u}{2\rho c^2} & -\frac{p_e v}{2\rho c^2} & -\frac{p_e w}{2\rho c^2} & \frac{p_e}{2\rho c^2} \end{pmatrix}, \quad (\text{A13})$$

which satisfy $L_1 = R_1^{-1}$ and $A = R_1 \Lambda L_1$, with $\Lambda = \text{diag}(u - c, u, u, u, u, u, u + c)$. The matrices for convection along y and z are obtained by exchanging the role of u with v and w , respectively, and rearranging the shear rows and columns accordingly. For a single-species flow the matrices reduce to their classical 5×5 counterparts; in particular, for a calorically perfect gas, $\hat{p}_\rho = (\gamma - 1)q$ and $E - \rho p_\rho/p_e = q$, and the familiar expressions are recovered.

The matrices L and R evaluated at the Roe-averaged state of the interface are then used as follows: at each node of the $2r$ -point stencil centered at the interface $i + 1/2$,

the conservative variables and the physical flux are projected onto the characteristic fields, $w_m = L_m \mathbf{C}$ and $g_m = L_m \mathbf{F}$, and employing a local Lax–Friedrichs splitting we obtain

$$g_m^\pm = \frac{1}{2} (g_m \pm \lambda_m w_m), \quad (\text{A14})$$

where λ_m is the maximum over the stencil of the modulus of the eigenvalue associated with the m -th field, evaluated at the nodes ($|u \mp c|$ for the acoustic fields, $|u|$ for the species and shear fields). The split fluxes g_m^+ and g_m^- are reconstructed at the interface with the upwind-biased WENO procedure of order $2r - 1$, using the left- and right-biased $(2r - 1)$ -point sub-stencils respectively, and the numerical flux is finally recovered in conservative form as

$$\hat{F}_{i+1/2} = R \hat{g}, \quad \hat{g}_m = \hat{g}_m^+ + \hat{g}_m^-. \quad (\text{A15})$$

Appendix B HLLC solver

As an alternative to the characteristic-wise flux-vector splitting of Appendix A, the upwind flux at the shock-flagged interfaces can be computed with the HLLC approximate Riemann solver (Toro 2009), which restores the contact and shear waves lost in the original HLL scheme. The procedure reported here is based on the work of Johnsen and Colonius (2006), Abgrall (1996), and Karni (1994). The left and right input states of the Riemann problem are obtained from a fifth-order WENO-Z reconstruction (Borges et al. 2008) of the primitive variables, carried out in locally characteristic form to reduce spurious oscillations across discontinuities. On the six-point stencil centered at the interface, the vector $[\rho, u, v, w, p]$ is projected onto the characteristic fields of the Euler equations linearized about the Roe-averaged interface state, denoted here by an overbar, as in Appendix A,

$$\begin{aligned} w^{(1)} &= \frac{1}{2} (p - \bar{\rho} \bar{c} u), & w^{(2)} &= \rho - \frac{p}{\bar{c}^2}, \\ w^{(3)} &= v, & w^{(4)} &= w, & w^{(5)} &= \frac{1}{2} (p + \bar{\rho} \bar{c} u). \end{aligned} \quad (\text{B16})$$

Each field is reconstructed at the interface with upwind bias from either side, and the reconstructed values are mapped back to the primitive variables through the relations

$$\begin{aligned} p_K &= w_K^{(1)} + w_K^{(5)}, & u_K &= \frac{w_K^{(5)} - w_K^{(1)}}{\bar{\rho} \bar{c}}, & \rho_K &= w_K^{(2)} + \frac{p_K}{\bar{c}^2}, \\ v_K &= w_K^{(3)}, & w_K &= w_K^{(4)}, \end{aligned} \quad (\text{B17})$$

with $K = L, R$. The mass fractions are reconstructed component-wise with the same WENO-Z operator, clipped to $[0, 1]$ and normalized to unit sum. The thermodynamic state is then re-evaluated from the reconstructed values: the temperature follows from $T_K = p_K / (\rho_K R_K)$, with $R_K = \sum_{m=1}^{N_s} Y_{m,K} R_0 / W_m$, and the internal energy, the total energy, the frozen specific-heat ratio γ_K and the speed of sound $c_K = \sqrt{\gamma_K R_K T_K}$ are computed from the caloric EoS, so that the two input states are thermodynamically

consistent. A first-order variant, in which the left and right states coincide with the two adjacent nodes states, is also available. The extreme wave speeds are estimated following [Einfeldt \(1988\)](#) as

$$S_L = \min(u_L - c_L, \tilde{u} - \tilde{c}), \quad S_R = \max(u_R + c_R, \tilde{u} + \tilde{c}), \quad (\text{B18})$$

where the tilde denotes a second Roe average, taken between the reconstructed states L and R, rather than between adjacent nodes, $\tilde{\phi} = (\phi_L + \sqrt{\rho_R/\rho_L} \phi_R) / (1 + \sqrt{\rho_R/\rho_L})$ for $\phi = u, v, w, H, c_p, R$, and the average speed of sound is evaluated as $\tilde{c}^2 = (\tilde{\gamma} - 1) [\tilde{H} - \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2)]$, with $\tilde{\gamma} = \tilde{c}_p / (\tilde{c}_p - \tilde{R})$. The speed of the contact wave is computed following [Batten et al. \(1997\)](#),

$$S_M = \frac{p_R - p_L + \rho_L u_L (S_L - u_L) - \rho_R u_R (S_R - u_R)}{\rho_L (S_L - u_L) - \rho_R (S_R - u_R)}. \quad (\text{B19})$$

In the star region between the two acoustic waves, the normal velocity is constant and equal to S_M and the pressure is continuous, whereas the transverse velocities and the mass fractions are unchanged across the outer waves. The star states on the two sides of the contact are therefore

$$\begin{aligned} \rho_K^* &= \rho_K \frac{S_K - u_K}{S_K - S_M}, & (\rho_m)_K^* &= \rho_K^* Y_{m,K}, & (\rho u)_K^* &= \rho_K^* S_M, \\ (\rho v)_K^* &= \rho_K^* v_K, & (\rho w)_K^* &= \rho_K^* w_K, \end{aligned} \quad (\text{B20})$$

$$(\rho E)_K^* = \frac{S_K - u_K}{S_K - S_M} \left[\rho_K E_K + (S_M - u_K) \left(\rho_K S_M + \frac{p_K}{S_K - u_K} \right) \right], \quad (\text{B21})$$

and the numerical flux reads

$$\hat{\mathbf{F}}_{i+1/2} = \begin{cases} \mathbf{F}_L, & \text{if } 0 \leq S_L, \\ \mathbf{F}_L + S_L(\mathbf{C}_L^* - \mathbf{C}_L) & \text{if } S_L \leq 0 \leq S_M, \\ \mathbf{F}_R + S_R(\mathbf{C}_R^* - \mathbf{C}_R) & \text{if } S_M \leq 0 \leq S_R, \\ \mathbf{F}_R, & \text{if } S_R \leq 0, \end{cases} \quad (\text{B22})$$

where $\mathbf{C}_K$ and $\mathbf{F}_K$ denote the conservative states and the corresponding fluxes evaluated from the reconstructed values. Since the mass fractions do not jump across the acoustic waves, the species are transported through the interface by the contact wave only, rather than being diffused across the whole Riemann fan as in the HLL scheme. The expressions above refer to convection along x ; the fluxes along y and z follow by exchanging the role of u with v and w , respectively, in the characteristic projection, in the wave-speed estimates and in the star states.

To summarize, the algorithm implemented in STREAmS-2.0 to find an approximate solution to the Euler equations at the shock-flagged interfaces follows the steps reported below:

1. Evaluate the local Roe-averaged state to define the characteristic transformation at the cell interface.
2. Project the primitive variables onto the local characteristic space.
3. Use the WENO scheme to reconstruct the characteristic and primitive variables, obtaining the left and right interface states.
4. Evaluate the thermodynamics of the multicomponent mixture to build the conservative variables ($\mathbf{C}_L$ and $\mathbf{C}_R$) and compute the corresponding physical fluxes ($\mathbf{F}_L$ and $\mathbf{F}_R$).
5. Estimate the wave speeds (S_L , S_R , S_M) and use the HLLC Riemann solver to compute the numerical flux, $\hat{\mathbf{F}}_{i+1/2}$.
6. March the nodal conservative variables forward in time using the computed numerical fluxes.

References

- Abdelsamie, A., Fru, G., Oster, T., Dietzsch, F., Janiga, G., Thévenin, D.: Towards direct numerical simulations of low-mach number turbulent reacting and two-phase flows using immersed boundaries. *Computers & Fluids* **131** (2016) <https://doi.org/10.1016/j.compfluid.2016.03.017>
- Abdelsamie, A., Lartigue, G., Frouzakis, C.E., Thévenin, D.: The taylor–green vortex as a benchmark for high-fidelity combustion simulations using low-mach solvers. *Computers & Fluids* **223** (2021) <https://doi.org/10.1016/j.compfluid.2021.104935>
- Abgrall, R.: How to prevent pressure oscillations in multicomponent flow calculations: A quasi conservative approach. *Journal of Computational Physics* **125** (1996) <https://doi.org/10.1006/jcph.1996.0085>
- ANSYS: Chemkin Theory Manual 17.0 (15151). Reaction Design: San Diego (2015)
- Batten, P., Clarke, N., Lambert, C., Causon, D.M.: On the choice of wavespeeds for the hllc riemann solver. *SIAM Journal on Scientific Computing* **18** (1997) <https://doi.org/10.1137/S1064827593260140>
- Bauer, M., Treichler, S., Slaughter, E., Aiken, A.: Legion: Expressing locality and independence with logical regions. In: SC '12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, pp. 1–11 (2012). <https://doi.org/10.1109/SC.2012.71>
- Bernardini, M., Pirozzoli, S., Orlandi, P.: Compressibility effects on roughness-induced boundary layer transition. *International Journal of Heat and Fluid Flow* **35**, 45–51 (2012) <https://doi.org/10.1016/j.ijheatfluidflow.2012.02.007>
- Bernardini, M., Modesti, D., Salvatore, F., Pirozzoli, S.: STREAmS: A high-fidelity accelerated solver for direct numerical simulation of compressible turbulent flows. *Computer Physics Communications* **263**, 107906 (2021) <https://doi.org/10.1016/j.cpc.2021.107906>

- Bernardini, M., Modesti, D., Salvatore, F., Sathyanarayana, S., Della Posta, G., Pirozzoli, S.: STREAmS-2.0: Supersonic turbulent accelerated Navier-Stokes solver version 2.0. *Computer Physics Communications* **285**, 108644 (2023) <https://doi.org/10.1016/j.cpc.2022.108644>
- Boivin, P., Jiménez, C., Sánchez, A.L., Williams, F.A.: An explicit reduced mechanism for h₂-air combustion. *Proceedings of the Combustion Institute* **33** (2011) <https://doi.org/10.1016/j.proci.2010.05.002>
- Borges, R., Carmona, M., Costa, B., Don, W.S.: An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. *Journal of Computational Physics* **227** (2008) <https://doi.org/10.1016/j.jcp.2007.11.038>
- Costa, P., Phillips, E., Brandt, L., Fatica, M.: Gpu acceleration of CaNS for massively-parallel direct numerical simulations of canonical fluid flows. *Computers & Mathematics with Applications* **81**, 502–511 (2021) <https://doi.org/10.1016/j.camwa.2020.01.002>
- De Vanna, F., Avanzi, F., Cogo, M., Sandrin, S., Bettencourt, M., Picano, F., Benini, E.: URANOS: A GPU accelerated Navier-Stokes solver for compressible wall-bounded flows. *Computer Physics Communications* **287**, 108717 (2023) <https://doi.org/10.1016/j.cpc.2023.108717>
- Di Renzo, M., Pirozzoli, S.: HTR-1.2 solver: Hypersonic task-based research solver version 1.2. *Computer Physics Communications* **261**, 107733 (2021) <https://doi.org/10.1016/j.cpc.2020.107733>
- Ducros, F., Ferrand, V., Nicoud, F., Weber, C., Darracq, D., Gacherieu, C., Poinso, T.: Large-eddy simulation of the shock/turbulence interaction. *Journal of Computational Physics* **152**(2), 517–549 (1999) <https://doi.org/10.1006/jcph.1999.6238>
- Einfeldt, B.: On godunov-type methods for gas dynamics. *SIAM Journal on Numerical Analysis* **25** (1988) <https://doi.org/10.1137/0725021>
- Gilbert, R.G., Luther, K., Troe, J.: Theory of thermal unimolecular reactions in the fall-off range. ii. weak collision rate constants. *Berichte der Bunsengesellschaft für physikalische Chemie* **87** (1983) <https://doi.org/10.1002/bbpc.19830870218>
- Goodwin, D.G., Moffat, H.K., Schoegl, I., Speth, R.L., Weber, B.W.: Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes (2025) <https://doi.org/10.5281/zenodo.17620923>
- Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D.E., Woodward, C.S.: Sundials: Suite of nonlinear and differential/algebraic equation solvers. *ACM Trans. Math. Softw.* **31** (2005) <https://doi.org/10.1145/1089014.1089020>

- Hirschfelder, J.O., Curtiss, C.F., Bird, R.B.: Molecular Theory of Gases and Liquids. John Wiley & Sons (1954)
- Jiang, G.-S., Shu, C.-W.: Efficient implementation of weighted eno schemes. *Journal of Computational Physics* **126**(1), 202–228 (1996) <https://doi.org/10.1006/jcph.1996.0130>
- Johnsen, E., Colonius, T.: Implementation of weno schemes in compressible multi-component flow problems. *Journal of Computational Physics* **219** (2006) <https://doi.org/10.1016/j.jcp.2006.04.018>
- Karni, S.: Multicomponent flow calculations by a consistent primitive algorithm. *Journal of Computational Physics* **112** (1994) <https://doi.org/10.1006/jcph.1994.1080>
- Kee, R.J., Rupley, F.M., Meeks, E.: CHEMKIN-III: a Fortran Chemical Kinetics Package for the Analysis of Gas-phase Chemical and Plasma Kinetics. Sandia National Laboratories (1996)
- Kee, R.J., Grcar, J.F., Smooke, M.D., Miller, J.A.: A FORTRAN Program for Modeling Steady Laminar One-dimensional Premixed Flames. SAND (Series) (Albuquerque, N.M.). Sandia National Laboratories, Livermore, CA (1988)
- Laizet, S., Lamballais, E.: High-order compact schemes for incompressible flows: A simple and efficient method with quasi-spectral accuracy. *Journal of Computational Physics* **228** (2009) <https://doi.org/10.1016/j.jcp.2009.05.010>
- Lindemann, F.A., Arrhenius, S., Langmuir, I., Dhar, N.R., Perrin, J., Lewis, W.C.M.: Discussion on the radiation theory of chemical action. *Transactions of the Faraday Society* **17** (1922) <https://doi.org/10.1039/TF9221700598>
- Martínez Ferrer, P.J., Buttay, R., Lehnasch, G., Mura, A.: A detailed verification procedure for compressible reactive multicomponent Navier–Stokes solvers. *Computers & Fluids* **89**, 88–110 (2014) <https://doi.org/10.1016/j.compfluid.2013.10.014>
- Mathur, S., Tondon, P.K., Saxena, S.C.: Thermal conductivity of binary, ternary and quaternary mixtures of rare gases. *Molecular Physics* **12**(6), 569–579 (1967) <https://doi.org/10.1080/00268976700100731>
- Modesti, D., Pirozzoli, S.: Reynolds and mach number effects in compressible turbulent channel flow. *International Journal of Heat and Fluid Flow* **59**, 33–49 (2016) <https://doi.org/10.1016/j.ijheatfluidflow.2016.01.007>
- Moureau, V., Domingo, P., Vervisch, L.: Design of a massively parallel CFD code for complex geometries. *Comptes Rendus Mécanique* **339**(2), 141–148 (2011) <https://doi.org/10.1016/j.crme.2010.12.001>

- Ó Conaire, M., Curran, H.J., Simmie, J.M., Pitz, W.J., Westbrook, C.K.: A comprehensive modeling study of hydrogen oxidation. *International Journal of Chemical Kinetics* **36**(11), 603–622 (2004) <https://doi.org/10.1002/kin.20036>
- Pirozzoli, S.: Generalized conservative approximations of split convective derivative operators. *Journal of Computational Physics* **229**(19), 7180–7190 (2010) <https://doi.org/10.1016/j.jcp.2010.06.006>
- Pirozzoli, S., Bernardini, M.: Turbulence in supersonic boundary layers at moderate reynolds number. *Journal of Fluid Mechanics* **688**, 120–168 (2011) <https://doi.org/10.1017/jfm.2011.368>
- Pirozzoli, S., Bernardini, M.: Probing high-reynolds-number effects in numerical boundary layers. *Physics of Fluids* **25**, 021704 (2013) <https://doi.org/10.1063/1.4792164>
- Pirozzoli, S., Bernardini, M., Grasso, F.: Direct numerical simulation of transonic shock/boundary layer interaction under conditions of incipient separation. *Journal of Fluid Mechanics* **657**, 361–393 (2010) <https://doi.org/10.1017/S0022112010001710>
- Poinsot, T., Veynante, D.: *Theoretical and Numerical Combustion*, 3rd edn. R. T. Edwards (2011)
- Press, W.H., Flannery, P., Teukolsky, S.A., Vetterling, W.T.: *Numerical Recipes in FORTRAN 77*, 2nd edn. Press Syndicate of the University of Cambridge (1993)
- Romero, J., Crabill, J., Watkins, J.E., Witherden, F.D., Jameson, A.: ZEFR: A GPU-accelerated high-order solver for compressible viscous flows using the flux reconstruction method. *Computer Physics Communications* **250**, 107169 (2020) <https://doi.org/10.1016/j.cpc.2020.107169>
- Rosenbrock, H.H.: Some general implicit processes for the numerical solution of differential equations. *The Computer Journal* **5**(4), 329–330 (1963) <https://doi.org/10.1093/comjnl/5.4.329>
- Salvadore, F., Bernardini, M., Botti, M.: GPU accelerated flow solver for direct numerical simulation of turbulent flows. *Journal of Computational Physics* **235**, 129–142 (2013) <https://doi.org/10.1016/j.jcp.2012.10.012>
- Sandberg, R.D.: Direct numerical simulations for flow and noise studies. *Procedia Engineering* **61**, 356–362 (2013) <https://doi.org/10.1016/j.proeng.2013.08.027>
- Sathyanarayana, S., Bernardini, M., Modesti, D., Pirozzoli, S., Salvatore, F.: High-speed turbulent flows towards the exascale: STREAmS-2 porting and performance. *Journal of Parallel and Distributed Computing* **196**, 104993 (2025) <https://doi.org/10.1016/j.jpdc.2024.104993>

- Shuen, J.-S., Liou, M.-S., Leer, B.V.: Inviscid flux-splitting algorithms for real gases with non-equilibrium chemistry. *Journal of Computational Physics* **90**(2), 371–395 (1990) [https://doi.org/10.1016/0021-9991\(90\)90172-W](https://doi.org/10.1016/0021-9991(90)90172-W)
- Spalart, P.R., Moser, R.D., Rogers, M.M.: Spectral methods for the Navier-Stokes equations with one infinite and two periodic directions. *Journal of Computational Physics* **96**(2), 297–324 (1991) [https://doi.org/10.1016/0021-9991\(91\)90238-G](https://doi.org/10.1016/0021-9991(91)90238-G)
- Sportisse, B.: An analysis of operator splitting techniques in the stiff case. *Journal of Computational Physics* **161**(1), 140–168 (2000) <https://doi.org/10.1006/jcph.2000.6495>
- Stewart, P.H., Larson, C.W., Golden, D.M.: Pressure and temperature dependence of reactions proceeding via a bound complex. 2. Application to $2\text{CH}_3 \rightarrow \text{C}_2\text{H}_5 + \text{H}$. *Combustion and Flame* **75** (1989) [https://doi.org/10.1016/0010-2180\(89\)90084-9](https://doi.org/10.1016/0010-2180(89)90084-9)
- Stone, C.P., Alferman, A.T., Niemeyer, K.E.: Accelerating finite-rate chemical kinetics with coprocessors: Comparing vectorization methods on gpus, mics, and cpus. *Computer Physics Communications* **226**, 18–29 (2018) <https://doi.org/10.1016/j.cpc.2018.01.015>
- Tamaki, Y., Kuya, Y., Kawai, S.: Comprehensive analysis of entropy conservation property of non-dissipative schemes for compressible flows: Keep scheme redefined. *Journal of Computational Physics* **468**, 111494 (2022) <https://doi.org/10.1016/j.jcp.2022.111494>
- Taylor, G.I., Green, A.E.: Mechanism of the production of small eddies from large ones. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* **158** (1937) <https://doi.org/10.1098/rspa.1937.0036>
- TOP500: TOP500 Supercomputer Sites. <https://top500.org/lists/top500/2026/06/>. Accessed: 1 September 2026 (2026)
- Toro, E.: *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*, (2009). <https://doi.org/10.1007/b79761>
- Utkarsh, U., Churavy, V., Ma, Y., Besard, T., Srisuma, P., Gymnich, T., Gerlach, A.R., Edelman, A., Barbastathis, G., Braatz, R.D., Rackauckas, C.: Automated translation and accelerated solving of differential equations on multiple gpu platforms. *Computer Methods in Applied Mechanics and Engineering* **419** (2024) <https://doi.org/10.1016/j.cma.2023.116591>
- Wilke, C.R.: A viscosity equation for gas mixtures. *The Journal of Chemical Physics* **18**(4), 517–519 (1950) <https://doi.org/10.1063/1.1747673>
- Witherden, F.D., Farrington, A.M., Vincent, P.E.: PyFR: An open source framework for solving advection–diffusion type problems on streaming architectures using the

- flux reconstruction approach. *Computer Physics Communications* **185**(11), 3028–3040 (2014) <https://doi.org/10.1016/j.cpc.2014.07.011>
- Wu, H., Ma, P.C., Ihme, M.: Efficient time-stepping techniques for simulating turbulent reactive flows with stiff chemistry. *Computer Physics Communications* **243** (2019) <https://doi.org/10.1016/j.cpc.2019.04.016>
- Zhang, W., Myers, A., Gott, K., Almgren, A., Bell, J.: AMReX: Block-structured adaptive mesh refinement for multiphysics applications. *The International Journal of High Performance Computing Applications* **35**, 508–526 (2021) <https://doi.org/10.1177/10943420211022811>
- Zhu, X., Phillips, E., Spandan, V., Donners, J., Ruetsch, G., Romero, J., Ostilla-Monico, R., Yang, Y., Lohse, D., Verzicco, R., Fatica, M., Stevens, R.J.A.M.: Afid-gpu: A versatile Navier–Stokes solver for wall-bounded turbulent flows on gpu clusters. *Computer Physics Communications* **229**, 199–210 (2018) <https://doi.org/10.1016/j.cpc.2018.03.026>