The Journal of Open Source Software

CÆLUM: An academic High-Order Finite-Volume solver for the compressible Euler equations and related models

Adrian Navas-Montilla [1*] **and Isabel Echeverribar** [2*]

1 Fluid Dynamics Technologies, i3A-University of Zaragoza, Spain. **2** Instituto Tecnológico de Aragón – ITA, Zaragoza, Spain. ***** These authors contributed equally.

DOI: 10.21105/joss.07777

Software

- Review
- Repository
- Archive

Editor: Philip Cardiff

Reviewers:

- @iBatistic
- @nasserma

Submitted: 09 October 2024
Published: 29 September 2025

Summary

In the last decade, high-order numerical methods have become very popular within the Computational Fluid Dynamics (CFD) and Numerical Weather Prediction (NWP) communities. Traditional low-order –i.e., equal or lower than 2– methods have been widely adopted by industry and academia to simulate fluid flows, but they exhibit important limitations due to high diffusion and dispersion errors, which may lead to non-physical solutions. By contrast, high-order schemes can provide accurate solutions, with small diffusion and dispersion errors, and feature a high computational efficiency (Ferrer et al., 2023; Wang et al., 2013). They are well-suited for multi-scale problems, e.g., turbulent flows, flows involving sharp gradients such as shocks, acoustics, etc. High-order solvers have gained attention due to their notable performance when running on modern High-Performance Computing (HPC) architectures.

`CÆLUM` is a hydrodynamic solver for compressible flows, which solves the Euler equations of gas dynamics and other related models by means of high-order essentially non-oscillatory finite volume schemes in Cartesian meshes. The core of the solver is written in C, but pre- and post-processing can be easily done using Jupyter Notebook, as provided in the examples. The Weighted Essentially Non-Oscillatory (WENO) (Jiang & Shu, 1996) and Targeted Essentially Non-Oscillatory (TENO) (Fu, 2019b) spatial reconstructions, in combination with Runge-Kutta integrators (Gottlieb et al., 2001), are used in `CÆLUM` to achieve a high resolution in smooth regions of the flow and to capture discontinuous solutions without spurious oscillations. These schemes also make it possible to adopt an Implicit Large Eddy Simulation (iLES) framework when computing turbulent flows (Grinstein et al., 2007; San & Kara, 2015; Solán-Fustero et al., 2021). The numerical diffusion plays the role of the explicit dissipative sub-grid model–a favorable choice when seeking simplicity and computational efficiency. In summary, this hydrodynamic solver can compute turbulent compressible flows, as well as low Mach number flows such as simplified dry atmospheric flows and other wave-propagation phenomena.

Statement of need

Given the limitations of traditional low-order numerical methods mentioned above, there is a critical need for advanced, high-order computational tools. The increasing complexity of engineering and scientific problems motivates a shift toward methods that can deliver both high accuracy and computational efficiency. Therefore, it is important to popularize such methods and make them available to the scientific community, in particular, for students and early-career researchers who are often limited by the accessibility of these tools.

`CÆLUM` is designed for use by both researchers and students in the fields of CFD, atmospheric flows, and related disciplines. In fact, the simplicity and compactness of the code make it particularly suitable for those taking their first steps in numerical flow simulation with high-order methods. It is designed from an academic perspective, where clarity and accessibility are prioritized. Therefore, it includes user-friendly pre-processing and post-processing tools based on Python and Jupyter Notebook. The repository comes with a series of Python scripts for the configuration and visualization of various example flows, ranging from simple scalar advection in 1D to more complex atmospheric or compressible flow cases in 3D. These scripts rely on standard libraries such as `numpy`, `matplotlib`, and `pyvista`, the latter being a powerful module for data visualization and rendering. Additionally, several Jupyter Notebooks are included, where the steps for configuring the simulation tool and visualizing the results are explained in detail.

A key feature of this software is that it also allows the simulation of simplified dry atmospheric flows in the meso- and micro-scale. For this, `CÆLUM` uses the compressible Euler equations with the gravity source term, composed of the equations for the conservation of mass, momentum, and energy (Ghosh & Constantinescu, 2016). This approach offers several benefits for the selected application, such as the ability to conserve mass and energy with machine accuracy when using a suitable discretization. Another benefit of this approach is that many numerical advances developed by the CFD community can be easily adapted (Giraldo & Restelli, 2008).

Regarding numerics, the simulation code uses WENO (Jiang & Shu, 1996), TENO (Fu, 2019b), and optimal polynomial reconstructions on Cartesian meshes. For simplicity, a discretization strategy based on the midpoint rule that employs independent 1D reconstructions in each Cartesian direction is used (Zhang et al., 2011). This approach avoids performing multi-dimensional reconstructions and Gaussian integration at the cell faces, thereby drastically reducing the complexity of the algorithms as well as computational expenses. This is done at the cost of not achieving a genuinely high order of accuracy, which is not critical when computing shocked problems, underresolved flows, or flows with discontinuities and sharp gradients (Fu, 2019a; San & Kara, 2015; Zhang et al., 2011). As a result, we provide a simple and versatile computational code that can be applied to a wide variety of problems and enables iLES.

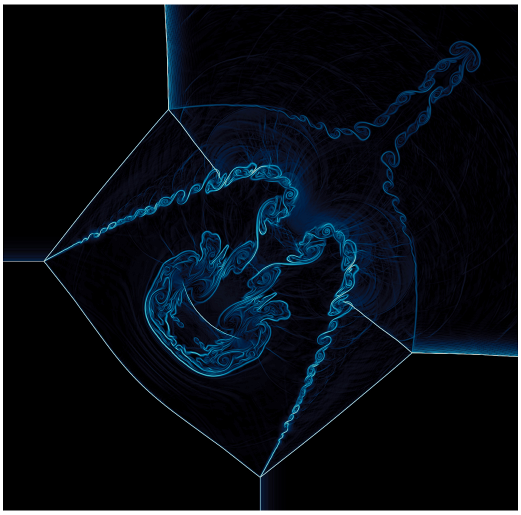

Figure 1: Density gradient (schlieren-like) representation of a 2D Riemann Problem from Lax & Liu (1998). Solution computed by a 7th-order WENO scheme.

As an example, Figure 1 displays the solution provided by `CÆLUM` of a 2D Riemann Problem from Lax & Liu (1998), which is a typical benchmark for this type of simulation code. The figure shows the density gradient representation, which reveals the main features of the flow. Shock waves, as well as slip lines (i.e., contact lines) where Kelvin-Helmholtz vortices form, are accurately captured.

Regarding the implementation of the solver, CÆLUM uses OpenMP, a directive-based threading library that allows parallel computing on multi-core CPUs. The modification of the base code by the inclusion of pragmas is minimal, preserving the readability and clarity of the code.

We can find other open-source simulation codes using high-order non-oscillatory finite volume and finite difference schemes for the more general Navier-Stokes equations, aimed at similar applications, which show a superior performance than most state-of-the-art commercial codes. Some examples are OpenSBLI (Lusher et al., 2021), a powerful framework for code generation and HPC execution of high order solvers for CFD; HTR Solver (Di Renzo et al., 2020), which is a GPU-accelerated, TENO-based Direct Numerical Simulation (DNS) solver of transitional and turbulent hypersonic turbulent flows at high enthalpies, and accounts for thermochemical effects; JAX-Fluids (Bezgin et al., 2023), offering a user-friendly fully-differentiable CFD solver for 3D, compressible single-phase and two-phase flows, using WENO and TENO schemes and being GPU- and TPU-accelerated; UCNS3D solver (Antoniadis et al., 2022), a powerful high-order, finite-volume CFD solver that allows an extensive choice of numerical methods and sub-grid models and runs in HPC facilities; URANOS (De Vanna et al., 2023), a massively parallel GPU-accelerated compressible flow solver for high-fidelity modeling of compressible wall flows (DNS, LES and WMLES); STREAmS (Bernardini et al., 2021), a GPU-accelerated high-fidelity solver for direct numerical simulations (DNS) of canonical compressible wall-bounded flows; HyPar (Ghosh, 2015), a finite-difference solver for the computation of hyperbolic-parabolic partial differential equations with source terms on Cartesian grids, which considers (dry) atmospheric applications; and Cerisse (Patel et al., 2024), a GPU-accelerated solver for LES and DNS of high-speed compressible turbulent reactive flows, with customizable thermodynamics and complex chemistry. We can also find finite volume solvers aimed exclusively at atmospheric applications, such as the OpenFOAM-based GEA (Girfoglio et al., 2023) and PortUrb (M. Norman et al., 2025; M. R. Norman et al., 2023), a flexible HPC, WENO-based model for the moist atmosphere.

Similarly, there are other open-source solvers based on the Discontinuous Galerkin (DG) method such as HORSES3D (Ferrer et al., 2023), which is a very powerful multiphysics environment that solves the Navier-Stokes equations, the Cahn–Hilliard equation and their entropy–stable variants and can be applied to a wide range of applications including industrial turbulent flows, multi-phase flows and aeroacustics; GALÆEXI (Kempf et al., 2024), a GPU-enabled, high-order numerical framework for solving PDEs, with a special focus on turbulent flows and shock-turbulence interaction; Nek5000 (Markidis et al., 2015), a well-known computational fluid dynamics solver that is very efficient, scalable and ideally suited to wall-bound and complex geometry problems; and NebulaSEM (Abdi, 2024), a recent open-source code that combines finite volume and DG methods with adaptive mesh refinement to construct a high-order non-hydrostatic dynamical core for atmospheric simulations and other applications.

Compared to DG-finite-element (FE) approaches (and hybrid variants), high-order finite volume methods generally offer lower algorithmic and implementation complexity, retaining robustness and simplicity in handling shocks and discontinuities. This motivates the choice of this family of methods in CÆLUM. This is done at the cost of a lower formal accuracy per degree of freedom and a non-local nature when compared with DG-FE methods, which may sometimes compromise computational efficiency –i.e., they rely on wider stencils that hinder parallel scalability.

CÆLUM represents an alternative to the open-source codes listed above, offering the features detailed earlier in this section. It is targeted at those seeking a simpler, more academic tool, as it is designed to be pedagogical and versatile in terms of applications, and it includes explanatory notebooks. It can also serve as a testbed for evaluating novel Riemann solvers and spatial reconstruction methods. A key feature not found in other packages is the implementation of well-balanced TENO schemes for the Euler equations with gravity, used to simulate simplified atmospheric flows, including different approaches for integrating source terms, each with a distinct impact on energy conservation (Navas-Montilla et al., 2024). As an academic code, it may have some limitations when applied to realistic cases, particularly due to its reliance on a uniform structured mesh.

The model equations

We herein consider the compressible Euler equations with a gravitational source term, given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{Continuity}$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v} + p\mathbf{I}) = \rho \mathbf{g} \tag{Momentum}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot ((E + p)\mathbf{v}) = \rho \mathbf{v} \cdot \mathbf{g} \tag{Energy}$$

where ρ is density, $\mathbf{v}$ is the velocity vector, p is pressure, $\mathbf{g} = (0, 0, -g)^T$ is the specific gravity force and $\mathbf{I}$ is the identity matrix. The energy is defined as the sum of kinetic and internal energy $E = \rho(\frac{1}{2}\mathbf{v} \cdot \mathbf{v} + e)$.

Additionally, scalar transport can also be considered as the following equation is implemented

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) = 0$$

where u is the transported quantity and $\mathbf{v}$ is the advection velocity. It is also possible to compute the Burgers equation when setting $\mathbf{v} = 1/2(u, u, u)^T$.

Further details on the model equations and numerical resolution methods can be found in (Navas-Montilla et al., 2024; Navas-Montilla & Echeverribar, 2023).

Previous work using the software

`CÆLUM` was developed and first used in Navas-Montilla & Echeverribar (2023), presenting a novel methodology to construct high-order well-balanced schemes for the computation of the Euler equations with gravitational source term, with application to NWP. The objective of this work was twofold: first, to assess the use of augmented solvers in computing the Euler equations with gravity, and second, to evaluate the performance of the novel TENO reconstruction for NWP. Afterwards, a more traditional implementation of the Euler equations with gravity based on the use of fluctuation variables was implemented in different forms to also allow for the conservation of total energy. The resulting schemes were evaluated in terms of their spectral resolution for the computation of turbulent flows in the dry atmosphere. The results were presented in Navas-Montilla et al. (2024), with the objective of determining whether or not these models can be used to build an iLES framework, shedding light on their potential advantages or limitations in representing under-resolved atmospheric processes in the meso- and micro-scales.

Example cases

The example cases provided in the repository showcase a diverse suite of test problems that are commonly used to validate and demonstrate the capabilities of computational fluid dynamics codes. These include scalar problems like linear advection and Burgers' equation in 1D, 2D, and 3D, compressible flow simulations such as Riemann problems (single- and multi-component flow), shock–bubble interactions, and a Richtmyer–Meshkov instability for turbulent mixing. Additionally, atmospheric flow cases feature colliding thermal bubbles and descending thermal plumes over complex terrain. Each case is supported by Python scripts (or Jupyter notebooks) and corresponding visual animations, making them useful for verifying the numerical solvers and exploring the underlying physical phenomena. Apart from this, the code also provides an automated testing script that performs a convergence rate test, executes four Riemann problems, and runs an atmospheric flow benchmark.

Acknowledgements

Part of this work has been funded by the Spanish Ministry of Science, Innovation and Universities - Agencia Estatal de Investigación (10.13039/501100011033) and FEDER-EU under project-nr. PID2022-141051NA-I00.

References

Abdi, D. S. (2024). NebulaSEM: A high-order discontinuous galerkin spectral element code for atmospheric modeling. *Journal of Open Source Software*, *9*(96), 6448. https://doi.org/10.21105/joss.06448

Antoniadis, A. F., Drikakis, D., Farmakis, P. S., Fu, L., Kokkinakis, I., Nogueira, X., Silva, P. A. S. F., Skote, M., Titarev, V., & Tsoutsanis, P. (2022). UCNS3D: An open-source high-order finite-volume unstructured CFD solver. *Computer Physics Communications*, *279*, 108453. https://doi.org/10.1016/j.cpc.2022.108453

Bernardini, M., Modesti, D., Salvadore, F., & Pirozzoli, S. (2021). STREAmS: A high-fidelity accelerated solver for direct numerical simulation of compressible turbulent flows. *Computer Physics Communications*, *263*, 107906. https://doi.org/10.1016/j.cpc.2021.107906

Bezgin, D. A., Buhendwa, A. B., & Adams, N. A. (2023). JAX-fluids: A fully-differentiable high-order computational fluid dynamics solver for compressible two-phase flows. *Computer Physics Communications*, *282*, 108527. https://doi.org/10.1016/j.cpc.2022.108527

De Vanna, F., Avanzi, F., Cogo, M., Sandrin, S., Bettencourt, M., Picano, F., & Benini, E. (2023). URANOS: A GPU accelerated navier-stokes solver for compressible wall-bounded flows. *Computer Physics Communications*, *287*, 108717. https://doi.org/10.1016/j.cpc.2023.108717

Di Renzo, M., Fu, L., & Urzay, J. (2020). HTR solver: An open-source exascale-oriented task-based multi-GPU high-order code for hypersonic aerothermodynamics. *Computer Physics Communications*, *255*, 107262. https://doi.org/10.1016/j.cpc.2020.107262

Ferrer, E., Rubio, G., Ntoukas, G., Laskowski, W., Mariño, O. A., Colombo, S., Mateo-Gabín, A., Marbona, H., Manrique de Lara, F., Huergo, D., Manzanero, J., Rueda-Ramírez, A. M., Kopriva, D. A., & Valero, E. (2023). A high-order discontinuous galerkin solver for flow simulations and multi-physics applications. *Computer Physics Communications*, *287*, 108700. https://doi.org/10.1016/j.cpc.2023.108700

Fu, L. (2019a). A low-dissipation finite-volume method based on a new TENO shock-capturing scheme. *Computer Physics Communications*, *235*, 25–39. https://doi.org/10.1016/j.cpc.2018.10.009

Fu, L. (2019b). A very-high-order TENO scheme for all-speed gas dynamics and turbulence. *Computer Physics Communications*, *244*, 117–131. https://doi.org/10.1016/j.cpc.2019.06.013

Ghosh, D. (2015). *HyPar Repository (https://github.com/debog/hypar)*.

Ghosh, D., & Constantinescu, E. M. (2016). Well-balanced, conservative finite difference algorithm for atmospheric flows. *AIAA Journal*, *54*(4), 1370–1385. https://doi.org/10.2514/1.J054580

Giraldo, F. X., & Restelli, M. (2008). A study of spectral element and discontinuous galerkin methods for the navier–stokes equations in nonhydrostatic mesoscale atmospheric modeling: Equation sets and test cases. *Journal of Computational Physics*, *227*(8), 3849–3877. https://doi.org/10.1016/j.jcp.2007.12.009

Girfoglio, M., Quaini, A., & Rozza, G. (2023). Validation of an OpenFOAM®-based solver for the euler equations with benchmarks for mesoscale atmospheric modeling. *AIP Advances*, *13*(5). https://doi.org/10.1063/5.0147457

Gottlieb, S., Shu, C.-W., & Tadmor, E. (2001). Strong stability-preserving high-order time discretization methods. *SIAM Review*, *43*(1), 89–112. https://doi.org/10.1137/S003614450036757X

Grinstein, F. F., Margolin, L. G., & Rider, W. J. (2007). *Implicit large eddy simulation* (Vol. 10). Cambridge university press Cambridge. https://doi.org/10.1017/CBO9780511618604

Jiang, G.-S., & Shu, C.-W. (1996). Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, *126*(1), 202–228. https://doi.org/10.1006/jcph.1996.0130

Kempf, D., Kurz, M., Blind, M., Kopper, P., Offenhäuser, P., Schwarz, A., Starr, S., Keim, J., & Beck, A. (2024). GALÆXI: Solving complex compressible flows with high-order discontinuous galerkin methods on accelerator-based systems. *arXiv Preprint arXiv:2404.12703*. https://doi.org/10.48550/arXiv.2404.12703

Lax, P. D., & Liu, X.-D. (1998). Solution of two-dimensional riemann problems of gas dynamics by positive schemes. *SIAM Journal on Scientific Computing*, *19*(2), 319–340. https://doi.org/10.1137/S1064827595291819

Lusher, D. J., Jammy, S. P., & Sandham, N. D. (2021). OpenSBLI: Automated code-generation for heterogeneous computing architectures applied to compressible fluid dynamics on structured grids. *Computer Physics Communications*, *267*, 108063. https://doi.org/10.1016/j.cpc.2021.108063

Markidis, S., Gong, J., Schliephake, M., Laure, E., Hart, A., Henty, D., Heisey, K., & Fischer, P. (2015). OpenACC acceleration of the Nek5000 spectral element code. *The International Journal of High Performance Computing Applications*, *29*(3), 311–319. https://doi.org/10.1177/1094342015576846

Navas-Montilla, A., & Echeverribar, I. (2023). A family of well-balanced WENO and TENO schemes for atmospheric flows. *Journal of Computational Physics*, *489*, 112273. https://doi.org/10.1016/j.jcp.2023.112273

Navas-Montilla, A., Guallart, J., Solán-Fustero, P., & García-Navarro, P. (2024). Exploring the potential of TENO and WENO schemes for simulating under-resolved turbulent flows in the atmosphere using euler equations. *Computers & Fluids*, *280*, 106349. https://doi.org/10.1016/j.compfluid.2024.106349

Norman, M. R., Eldred, C., & Gopalakrishnan Meena, M. (2023). Investigating inherent numerical stabilization for the moist, compressible, non-hydrostatic euler equations on collocated grids. *Journal of Advances in Modeling Earth Systems*, *15*(10), e2023MS003732. https://doi.org/10.1029/2023MS003732

Norman, M., Gopalakrishnan Meena, M., Gottiparthi, K., Koukpaizan, N., & Nichols, S. (2025). PortUrb: A performance portable, high-order, moist atmospheric large eddy simulation model with variable-friction immersed boundaries. *EGUsphere*, *2025*, 1–36. https://doi.org/10.5194/egusphere-2025-1135

Patel, Monal, Un, E., & Navarro-Martinez, S. (2024). *Cerisse Repository (https://github.com/salvadornm/cerisse)*.

San, O., & Kara, K. (2015). Evaluation of riemann flux solvers for WENO reconstruction schemes: Kelvin–helmholtz instability. *Computers & Fluids*, *117*, 24–41. https://doi.org/10.1016/j.compfluid.2015.04.026

Solán-Fustero, P., Navas-Montilla, A., Ferrer, E., Manzanero, J., & García-Navarro, P. (2021). Application of approximate dispersion-diffusion analyses to under-resolved burgers turbulence using high resolution WENO and UWC schemes. *Journal of Computational Physics*, *435*, 110246. https://doi.org/10.1016/j.jcp.2021.110246

Wang, Z. J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H. T., & others. (2013). High-order CFD methods: Current status and perspective. *International Journal for Numerical Methods in Fluids*, *72*(8), 811–845. https://doi.org/10.1002/fld.3767

Zhang, R., Zhang, M., & Shu, C.-W. (2011). On the order of accuracy and numerical performance of two classes of finite volume WENO schemes. *Communications in Computational Physics*, *9*(3), 807–827. https://doi.org/10.4208/cicp.291109.080410s