The Journal of Open Source Software

RHEA: an open-source Reproducible Hybrid-architecture flow solver Engineered for Academia

Lluís Jofre[1], **Ahmed Abdellatif**[1], **and Guillermo Oyarzun**[2]

1 Department of Fluid Mechanics, Universitat Politècnica de Catalunya · BarcelonaTech (UPC), Barcelona 08019, Spain **2** Department of Computer Applications in Science & Engineering, Barcelona Supercomputing Center, Barcelona 08034, Spain

DOI: 10.21105/joss.04637

Software

- Review
- Repository
- Archive

Editor: Patrick Diehl

Reviewers:

- @ctdegroot
- @thomasgillis

Submitted: 24 June 2022
Published: 13 January 2023

Summary

The study of complex multiscale flows (Groen et al., 2014), like for example the motion of small-scale turbulent eddies over large aerodynamic structures (Jofre & Doostan, 2022), microconfined high-pressure supercritical fluids for enhanced energy transfer (Bernades & Jofre, 2022), or hydrodynamic focusing of microorganisms in wall-bounded flows (Palacios et al., 2022), greatly benefits from the combination of interconnected theoretical, computational and experimental approaches. This manifold methodology provides a robust framework to corroborate the phenomena observed, validate the modeling assumptions utilized, and facilitates the exploration of wider parameter spaces and extraction of more sophisticated insights. These analyses are typically encompassed within the field of Predictive Science & Engineering (Njam, 2009), which has attracted attention in the Fluid Mechanics community and is expected to exponentially grow as computational studies transition from (mostly) physics simulations to active vectors for scientific discovery and technological innovation with the advent of Exascale computing (Alowayyed et al., 2017). In this regard, the computational flow solver presented aims at bridging the gap between studying complex multiscale flow problems and utilizing present and future state-of-the-art supercomputing systems in academic environments. The solver presented is named RHEA, which stands for *open-source **R**eproducible **H**ybrid-architecture flow solver **E**ngineered for **A**cademia*, and is available as an open-source Git repository at https://gitlab.com/ProjectRHEA/flowsolverrhea.

Statement of need

The computational study and optimization of multiscale flow problems, like for example turbulence, requires the utilization of powerful supercomputers. In particular, the computational cost of studying wall turbulence, in terms of total number of grid points N, by means of direct numerical simulation (DNS) and/or wall-modeled large eddy simulation (LES) strategies scales with the Reynolds number as $N \sim Re^{37/14}$ and $N \sim Re$ (Choi & Moin, 2012), respectively. Therefore, flows at high Reynolds numbers require large computational meshes solved on substantial node counts. As a result, the meshes used to discretize the computational domain on which the equations of fluid motion are solved need to be partitioned (distributed) among parallel tasks such that each of these only holds a local portion of the global mesh. The local portion of each task is composed of a set of grid points that it owns, and a set of off-processor grid points (owned by remote processors) connected with their corresponding local grid points. This overlapped mesh partition is used to exchange data among nearest neighbors by means of message passage interface (MPI) (Forum, 2022) instructions. In addition, present (and future) supercomputers are equipped with accelerators, like for example graphics processing units

(GPU), interconnected with the central processing units (CPU) and the main memory at the node level. In this regard, the flow solver RHEA uses OpenACC (Organization, 2022) pragmas to efficiently speedup the main kernels. This computational framework has been designed with the objective to facilitate the portability of the flow solver from small- and mid-size workstations to top-tier supercomputers, while maintaining notable levels of computational performance.

There is only a handful of full-fledged accelerated open-source flow solvers available in the literature. Some selected examples for compressible flows include (i) popular multi-purpose open-source packages for simulation (Cantwell et al., 2015) and design (Economon et al., 2015), (ii) OpenSBLI (Jacobs et al., 2017) dedicated to the automated derivation of finite-difference solvers for hybrid (CPU-GPU) architectures based on a Python framework, (iii) HTR (Renzo et al., 2020) constructed from the task-based Legion programming paradigm (Stanford University, 2022) and targeted to simulate hypersonic reacting flows, and (iv) STREAmS (Bernardini et al., 2021) resulting from the extension of a CPU-based solver to run on multi-GPU clusters and focused on supersonic ideal-gas flows. In this regard, RHEA has been designed and developed with the aim to cover this void for the academic community, specifically focusing on (i) solver expressivity, (ii) generality of flow problems that can be studied in terms of thermodynamic and flow regimes, (iii) rapid portability to different architectures, (iv) computational performance, and (v) reproducibility through uploading to the repository the parameters and input files required to regenerate the data from simulations.

Computational design

RHEA solves the three-dimensional (3-D) compressible equations of fluid motion utilizing non-uniform Cartesian second-order discretizations (Moin, 2010) in combination with kinetic energy preserving (Coppola et al., 2019) convection schemes, or different Harten-Lax-van-Leer-type (HLL) Riemann solvers (Toro, 2009), and uses explicit Runge-Kutta methods (Gottlieb et al., 2001) for time integration. The set of conservation equations is closed by means of ideal- or real-gas thermodynamics to target, respectively, subcritical and supercritical fluid regimes (Jofre & Urzay, 2021). The solver can be utilized for a wide range of fluid mechanics problems as it allows (i) to flexibly overwrite most high-level kernels to set, for example, specific initial conditions and source terms, and (ii) select between Dirichlet, Neumann, periodic, and subsonic & supersonic inflow-outflow (Poinsot & Lele, 1992) boundary conditions. RHEA is written in C++ (Stroustrup, 2013), using object-oriented programming, utilizes YAML (YAML Web Site, 2022) and HDF5 (Group, 2022) for input/output operations, and targets hybrid (CPU-GPU) supercomputing architectures based on a state-of-the-art parallel and scalable MPI (Forum, 2022) + OpenACC (Organization, 2022) accelerated (managed-memory) computational framework.

The computational performance of RHEA is analyzed by carrying out performance and scalability tests based on 100 time iterations of the 3-D turbulent channel flow configuration described in the next section. In this regard, to assess the portability of RHEA to different computing systems, the tests have been performed on (i) a local hybrid machine (results are referred to as Hybrid) and (ii) the Barcelona Supercomputer Center (Barcelona Supercomputing Center, 2022) (results are indicated as BSC) without performing any particular tuning of the solver. The Hybrid machine is composed of a node with 1 AMD Ryzen 9 3900XT 12-core CPU and 2 NVIDIA Quadro RTX 4000 GPUs, while the BSC supercomputer contains a CPU-GPU cluster with 3 racks formed of 54 IBM POWER9 nodes, each containing 2 Witherspoons CPUs (20 cores of 3.1 GHz each), 4 Volta NVIDIA GPUs (16 GB each), and 6.4 TB of NVMe (Non-Volatile Memory). The number of grid points utilized for the strong (total quantity) and weak (quantity per node) scalability tests correspond to $256 \times 256 \times 256$ (Hybrid) and $384 \times 384 \times 384$ (BSC). Three main observations can be inferred from the results. First, focusing on the time-to-solution when using CPUs+GPUs with respect to only CPUs, the solver is accelerated approximately $2.5\times$ and $5\times$ on the Hybrid and BSC computers. Second, as shown in Figure 1(left), on BSC for a fixed-problem size and noting that on CPUs+GPUs the solver is roughly $5\times$ faster, the solver presents strong scalability speedups above 80% and 60%, respectively, when running on CPUs and CPUs+GPUs up to 32 nodes (640 cores and 128 GPUs). Third, maintaining the same ratio of problem size to number of nodes, RHEA is able to preserve weak scalability efficiency above 90% on CPUs and CPUs+GPUs up to 32 nodes on BSC as depicted in Figure 1.

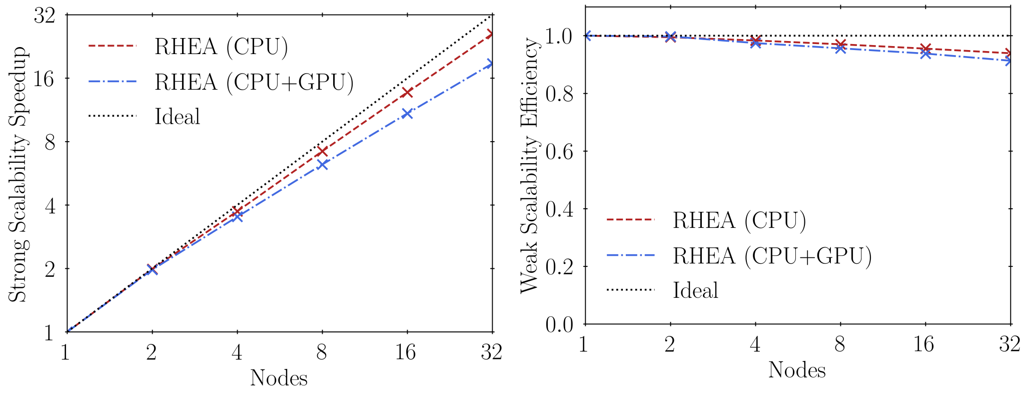

Figure 1: Strong scalability speedup (left) and weak scalability efficiency (right) of RHEA on BSC supercomputer using CPUs and CPUs+GPUs.

Application example

The ability of RHEA to configure a wide range of computational flow problems and run them efficiently on powerful supercomputing systems is demonstrated by simulating the canonical 3-D turbulent channel flow problem (Smits et al., 2011) on 2 nodes of the CTE POWER9 cluster of the Barcelona Supercomputing Center (Barcelona Supercomputing Center, 2022). The friction Reynolds number selected is $Re_\tau = u_\tau \delta/\nu = 180$, where $u_\tau = 1$ m/s is the friction velocity, $\delta = 1$ m is the channel half-height, and $\nu = \mu/\rho$ is the kinematic viscosity of the fluid with μ the dynamic viscosity and $\rho = 1$ kg/m^3 the density. The Prandtl number of the problem is $Pr = c_p \mu/\kappa = 0.71$ with c_p the isobaric specific heat capacity, and the Mach number is $Ma = u_b/\sqrt{\gamma P_b/\rho} = 0.3$ with u_b and P_b the bulk velocity and pressure, respectively, and $\gamma = 1.4$ the heat capacity ratio. The mass flow rate in the streamwise direction is imposed through a body force equal to $\mathbf{f} = [\tau_w/\delta, 0, 0]^\top$, where τ_w is the wall shear stress.

The computational domain is $4\pi\delta \times 2\delta \times 4/3\pi\delta$ in the streamwise (x), wall-normal (y), and spanwise (z) directions, respectively. The streamwise and spanwise boundaries are set periodic, and no-slip conditions are imposed on the horizontal boundaries (x-z planes). The KGP convection scheme (Coppola et al., 2019) is utilized for solving the transport equations together with a third-order Runge-Kutta time-integration method. The grid is uniform in the streamwise and spanwise directions with resolutions in wall units equal to $\Delta x^+ = 9$ and $\Delta z^+ = 6$, and stretched toward the walls in the vertical direction with the first grid point at $y^+ = y u_\tau/\nu = 0.1$ and with sizes in the range $0.2 \lesssim \Delta y^+ \lesssim 4$. This grid arrangement corresponds to a DNS of size $256 \times 128 \times 128$ grid points. The simulation strategy starts from a linear velocity profile with random fluctuations (Nelson & Fringer, 2017), which is advanced in time utilizing the KGP convection scheme (Coppola et al., 2019) with $\mathrm{CFL} = 0.9$ to reach turbulent steady-state conditions after approximately 20 flow-through-time (FTT) units; based on the bulk velocity u_b and the length of the channel $L_x = 4\pi\delta$, a FTT is defined as $t_b = L_x/u_b \sim \delta/u_\tau$. Flow statistics are collected for roughly 30 FTTs once steady-state conditions are achieved, and compared against reference results (Moser et al., 1999).

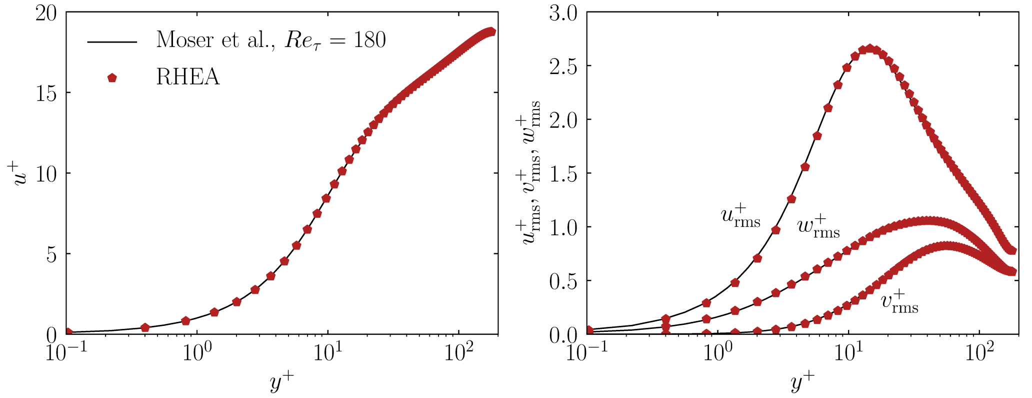

Figure 2: Comparison against reference results of the time-averaged streamwise velocity u^+ (left) and velocity fluctuations u^+_{rms}, v^+_{rms} and w^+_{rms} (right) along the wall-normal direction y^+ in wall units.

The time-averaged mean streamwise velocity u^+ and root-mean-squared (rms) velocity fluctuations u^+_{rms}, v^+_{rms}, w^+_{rms} along the wall-normal direction y^+ in wall units provided by RHEA and compared to reference results (Moser et al., 1999) are depicted in Figure 2. As shown in the figure, the results from RHEA accurately reproduce the first- and second-order flow statistics by (i) properly capturing the inner (viscous sublayer, buffer layer and and log-law region) and outer layers, and (ii) the flow fluctuations peaking around $y^+ \approx 15$ for the streamwise velocity. Additionally, a snapshot of the instantaneous streamwise velocity in wall units u^+ on a x^+-y^+ slice is displayed in Figure 3 to provide qualitative information of the wall-bounded turbulence computed by RHEA.

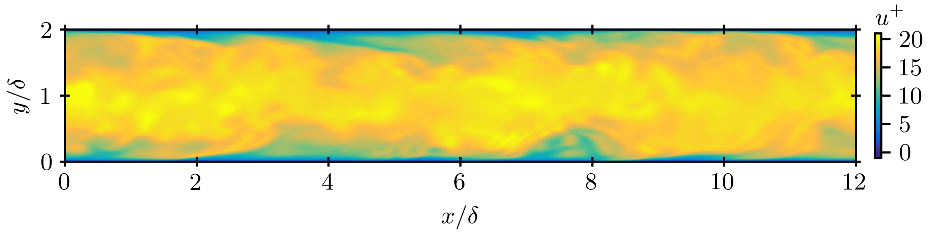

Figure 3: Snapshot of the instantaneous streamwise velocity u^+ on a x^+-y^+ slice from RHEA.

Acknowledgements

This work is supported by the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (grant agreement No. 101040379 - SCRAMBLE), and the *Beatriz Galindo* programme (Distinguished Researcher, BGP18/00026) of the *Ministerio de Ciencia, Innovación y Universidades* (Spain).

References

Alowayyed, S., Groen, D., Coveney, P. V., & Hoekstra, A. G. (2017). Multiscale computing in the Exascale era. *J. Comput. Sci.*, *22*, 15–25. https://doi.org/10.1016/j.jocs.2017.07.004

Barcelona Supercomputing Center. (2022). *BSC-CNS | Barcelona Supercomputing Center*. https://www.bsc.es

Bernades, M., & Jofre, L. (2022). Thermophysical analysis of microconfined turbulent flow regimes at supercritical fluid conditions in heat transfer applications. *J. Heat Transfer*, *144*, 082501. https://doi.org/10.1115/1.4054554

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

Cantwell, C., Moxey, D., Comerford, A., Bolis, A., Rocco, G., Mengaldo, G., Grazia, D. D., Yakovlev, S., Lombard, J.-E., Ekelschot, D., Jordi, B., Hu, H., Mohamied, Y., Eskilsson, C., Nelson, B., Vos, P., Biotto, C., Kirby, R., & Sherwin, S. (2015). Nektar++: An open-source spectral/hp element framework. *Comput. Phys. Commun.*, *192*, 205–209. https://doi.org/10.1016/j.cpc.2015.02.008

Choi, H., & Moin, P. (2012). Grid-point requirements for large eddy simulation: Chapman's estimates revisited. *Phys. Fluids*, *24*, 011702. https://doi.org/10.1063/1.3676783

Coppola, G., Capuano, F., Pirozzoli, S., & Luca, L. de. (2019). Numerically stable formulations of convective terms for turbulent compressible flows. *J. Comput. Phys.*, *382*, 86–104. https://doi.org/10.1016/j.jcp.2019.01.007

Economon, T., Palacios, F., Copeland, S., Lukaczyk, T., & Alonso, J. (2015). An open-source suite for multiphysics simulation and design. *AIAA J.*, *54*, 828–846. https://doi.org/10.2514/1.J053813

Forum, T. M. (2022). *Message Passing Interface*. https://www.mpi-forum.org

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

Groen, D., Zasada, S. J., & Coveney, P. V. (2014). Survey of multiscale and multiphysics applications and communities. *Comput. Sci. Eng.*, *16*, 34–43. https://doi.org/10.1109/MCSE.2013.47

Group, T. H. (2022). *Hierarchical Data Format 5*. https://www.hdfgroup.org

Jacobs, C., Jammy, S., & Sandham, N. (2017). OpenSBLI: A framework for the automated derivation and parallel execution of finite difference solvers on a range of computer architectures. *J. Comput. Sci.*, *18*, 12–23. https://doi.org/10.1016/j.jocs.2016.11.001

Jofre, L., & Doostan, A. (2022). Rapid aerodynamic shape optimization under uncertainty using a stochastic gradient approach. *Struct. Multidiscipl. Optim.*, *65*, 196. https://doi.org/10.1007/s00158-022-03293-y

Jofre, L., & Urzay, J. (2021). Transcritical diffuse-interface hydrodynamics of propellants in high-pressure combustors of chemical propulsion systems. *Prog. Energ. Combust.*, *82*, 100877. https://doi.org/10.1016/j.pecs.2020.100877

Moin, P. (2010). *Fundamentals of engineering numerical analysis* (2nd ed.). Cambridge University Press. https://doi.org/10.1017/CBO9780511781438

Moser, R. D., Kim, J., & Mansour, N. N. (1999). Direct numerical simulation of turbulent channel flow up to $Re_\tau = 590$. *Phys. Fluids*, *11*, 943. https://doi.org/10.1063/1.869966

Nelson, K. S., & Fringer, O. 4B. (2017). Reducing spin-up time for simulations of turbulent channel flow. *Phys. Fluids*, *29*, 105101. https://doi.org/10.1063/1.4993489

Njam, H. N. (2009). Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annu. Rev. Fluid Mech.*, *41*, 35–52. https://doi.org/10.1146/annurev.fluid.010908.165248

Organization, T. O. (2022). *Open Accelerators*. https://www.openacc.org

Palacios, C., Jofre, M., Jofre, L., Romeu, J., & Jofre-Roca, L. (2022). Superheterodyne microwave system for the detection of bioparticles with coplanar electrodes on a microfluidic platform. *IEEE Trans. Instrum. Meas.*, *71*, 8002910. https://doi.org/10.1109/TIM.2022.3165790

Poinsot, T. J., & Lele, S. K. (1992). Boundary conditions for direct simulations of compressible viscous flows. *J. Comput. Phys.*, *101*, 104–129. https://doi.org/10.1016/0021-9991(92)90046-2

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

Smits, A. J., McKeon, B. J., & Marusic, I. (2011). High-Reynolds number wall turbulence. *Annu. Rev. Fluid Mech.*, *43*, 353–375. https://doi.org/10.1146/annurev-fluid-122109-160753

Stanford University. (2022). *Legion Programming System*. https://legion.stanford.edu

Stroustrup, B. (2013). *The C++ programming language* (4th ed.). Addison-Wesley.

Toro, E. F. (2009). *Riemann solvers and numerical methods for fluid dynamics* (3rd ed.). Springer. https://doi.org/10.1007/b79761

YAML Web Site, T. Official. (2022). *YAML: YAML Ain't Markup Language*. https://yaml.org