The Journal of Open Source Software

HySoP: Hybrid Simulation with Particles

Jean-Matthieu Etancelin[1*¶], **Jean-Baptiste Keck**[3*], **Franck Pérignon**[3*], **Chloé Mimeau**[2*], **Nicolas Grima**[1*], **Christophe Picard**[3*], and **Georges-Henri Cottet**[3*]

1 Université de Pau et des Pays de l'Adour, E2S UPPA, CNRS, LMAP, Pau, France. **2** Laboratoire M2N, EA7340, CNAM, 2 rue Conté, Paris 75003, France. **3** Laboratoire Jean Kuntzmann, Grenoble INP, Université Grenobles Alpes and CNRS, Grenoble, France. ¶ Corresponding author * These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Philip Cardiff

Reviewers:

- @kburns
- @piyueh
- @lluisjofre

Submitted: 14 November 2024
Published: unpublished

Summary

In Computational Fluid Dynamics community, the most famous methods (finite difference, finite volume, finite element, spectral methods) deal with a purely Eulerian frameworks and have been extensively studied both from consistency/stability point of view as well as numerical dissipation characterization.

In parallel, particle approaches have met a large development in the context of incompressible flows and distinguish themselves from the Eulerian approaches by their intuitive and natural description of the fluid flow as well as their low numerical dissipation and by their ability to bypass the nonlinearities related to the advection phenomenon. Many efforts have been devoted to overcome the main intrinsic difficulties of purely Lagrangian particle methods. These efforts led in particular to the design of semi-Lagrangian approaches, also known as Remeshed Particle Method (RPM), where the particles discretizing the flow are regularly remeshed on a cartesian grid, thus capitalizing the strengths of both Eulerian and Lagrangian approaches (Mimeau & Mortazavi, 2021). The present numerical tool `HySoP` is a Python package dedicated to high performance numerical simulations of fluid-related problems based on these RPM semi-Lagrangian methods.

Statement of need

The `HySoP` library (Hybrid Simulation with Particles) is designed for distributed hybrid architectures, supporting CPU and GPU compute devices, via `MPI+OpenCL`. The high level functionalities and the user interface are mainly written in `Python`. In general, `HySoP` strives to propose a non-architecture-specific, performance-portable and easily reusable numerical code. One notes that `HySoP` is a scientific research software that is continuously evolving.

Most advanced open source softwares related to RPM and similar to `HySoP` are OpenFPM and Murphy. Both are parallel and accelerated libraries. OpenFPM (Incardona et al., 2019) is an open-source C++ framework for parallel particles-only and hybrid particle-mesh codes. Murhpy (Gillis & Rees, 2022) is a multiresolution adaptive grid framework for numerical simulations on 3D block-structured collocated grids with distributed computational architectures.

Features of the software

Description of the software design

HySoP has been designed on the basis of an uncoupling between the mathematical specifications of the problem to solve and the numerical methods and algorithms. The purpose is to let the user describing only the higher level specifications, in formulation quite close to the mathematical formalism:

- problem parameters;
- computational domain;
- variables defined on domain;
- operators;
- overall discretisation for the cartesian grid.

Specifying lower-level details like numerical methods, target architectures, and parallelism layout is optional and corresponding code is interchangeable and updgradable without any changes in user code.

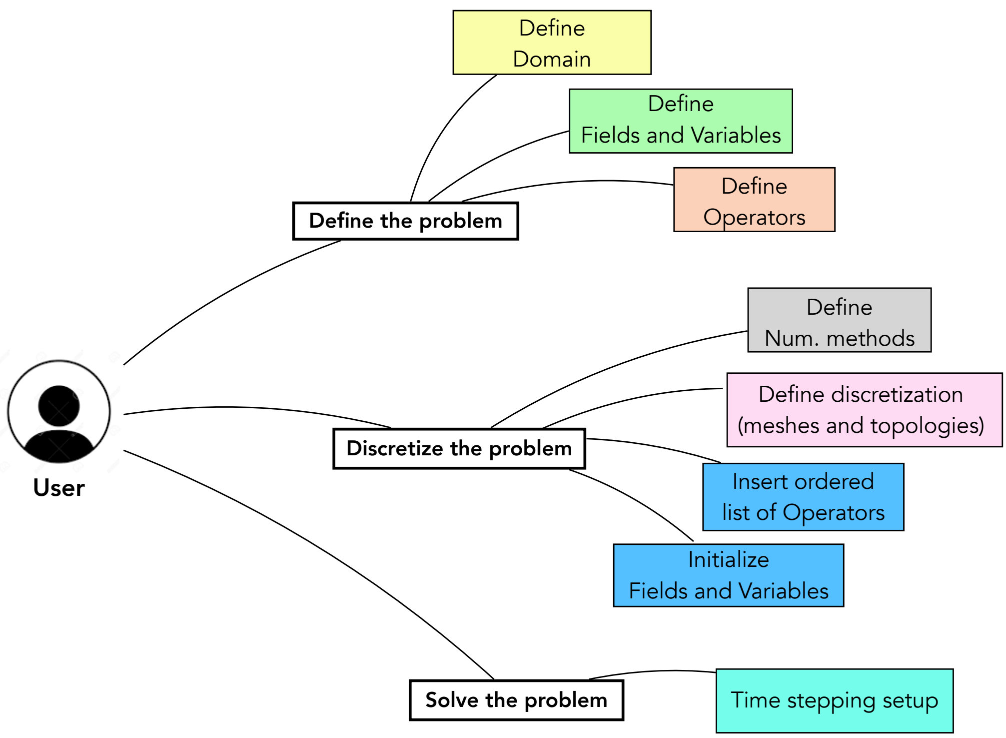

Figure 1: Use case diagram

From the user point of view, the main usecase will decompose into three main steps (cf Figure 1):

1. Problem description: as mathematical PDE formalism using domain, variables and operators;
2. Problem initialisation: after describing numerical methods and their parameters, the user specifies the main grid resolution, the mesh decomposition for parallel runs, and the compute backend. Then, the user enforces operator ordering and data initialization. At this stage, all memory allocations are completed, generated code is optimized and compiled, and communication layout is established.

3. Problem solving: with additional parameters for time-dependent problems (i.e. time steps), computations proceed by applying operators sequentially.

Programming languages and external dependencies

The main drawback of pure Python programs is performance, mitigated by using arrays and tools from the `numpy` module. Additional performance is achieved through optimized external libraries like `fftw` for fast Fourier transforms. `HySoP` is also using the `f2py` python module to use a Fortran implementation of RPM ((Lagaert et al., 2014)). Further speedup is obtained via just-in-time compiling using `numba` or `OpenCL`. `Numba` translates Python code into compiled code at runtime, while `OpenCL` handles multi-core and heterogeneous architectures. Contrary to `numba`, the `OpenCL` code is explicitly generated from formal representation of the numerical methods. Micro-benchmarks are also performed at initialization for optimization of the parameters of generated OpenCL code.

The interaction between backends and the base `Python` layer is summarized in Figure 2. Our software architecture allows easy integration of new backends, with inter-operability being the main challenge.

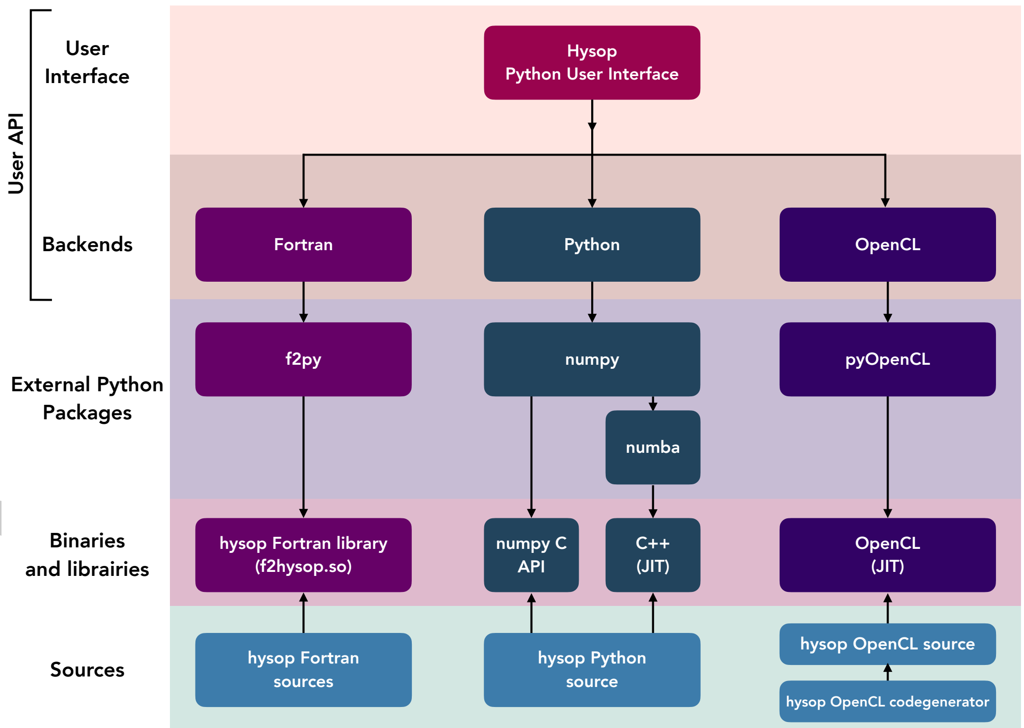

Figure 2: Computing backends in `HySoP`

Parallelism

The software supports distributed memory parallelism through domain decomposition using `MPI` parallelism, implemented via the `mpi4py` interface without restrictions on the `MPI` library provider. As a result, `HySoP` may be considered as a Python-based solver utilizing hybrid `MPI+OpenCL` programming for heterogeneous computing platforms.

`HySoP` also employs parallelism through the operator splitting approach, allowing parallelism by coarse tasks via the parallel execution of distinct operators that are weakly coupled in the mathematical formulation.

Evaluating HySoP's computational performance is challenging; however, references like (Cottet et al., 2014), (Keck, 2019), and (Keck et al., 2021) provide performance insights on multi-GPU and heterogeneous platform simulations.

Applications

The following list illustrates the successful use of the HySoP library in various domains of applications, implying a large range of governing equations, thus highlighting its versatility and flexibility:

Applications	Involved equations	Reference
- Bluff body flows	Navier-Stokes	(Mimeau et al., 2016, 2021)
- Large-Eddy Simulations (sub-grid scale modeling)	Filtered Navier-Stokes	(Crouy-Chanel et al., 2024)
- Transport of passive scalar at high Schmidt number	Navier-Stokes and a passive scalar advection-diffusion	(Cottet et al., 2014)
- Sedimentation in high Schmidt number flows	Navier-Stokes coupled with scalars advection-diffusion	(Keck et al., 2021)
- Passive flow control using porous media	Brinkman-Navier-Stokes	(Mimeau et al., 2017)
- Porous media dissolution at pore-scale	Darcy-Brinkman-Stokes coupled with reactive transport	(Etancelin et al., 2020)

Acknowledgements

We acknowledge contributions from Jean-Baptiste Lagaert during the genesis of this project.

The development of HySoP has been supported in parts by french ANR MPARME (ANR-17-CE23-0024) and ANR HAMM (ANR-10-COSI-0009)

References

Cottet, G.-H., Etancelin, J.-M., Pérignon, F., & Picard, C. (2014). High order Semi-Lagrangian particle methods for transport equations: numerical analysis and implementation issues. *ESAIM: Mathematical Modelling and Numerical Analysis*, *48*(4), 1029–1060. https://doi.org/10.1051/m2an/2014009

Crouy-Chanel, M. de, Mimeau, C., Mortazavi, I., Mariotti, A., & Salvetti, M. V. (2024). Large-eddy simulations with remeshed vortex methods: An assessment and calibration of subgrid-scale models. *Computers & Fluids*, *277*, 106287. https://doi.org/10.1016/j.compfluid.2024.106287

Etancelin, J.-M., Moonen, P., & Poncet, P. (2020). Improvement of remeshed Lagrangian methods for the simulation of dissolution processes at pore-scale. *Advances in Water Resources*, *146*, 103780. https://doi.org/10.1016/j.advwatres.2020.103780

Gillis, T., & Rees, W. M. van. (2022). *MURPHY – a scalable multiresolution framework for scientific computing on 3D block-structured collocated grids*. https://doi.org/10.48550/arXiv.2112.07537

Incardona, P., Leo, A., Zaluzhnyi, Y., Ramaswamy, R., & Sbalzarini, I. F. (2019). OpenFPM: A scalable open framework for particle and particle-mesh codes on parallel computers.

Computer Physics Communications, *241*, 155–177. https://doi.org/10.1016/j.cpc.2019.03.007

Keck, J.-B. (2019). *Numerical modelling and High Performance Computing for sediment flows* (Theses No. 2019GREAM067, Université Grenoble Alpes). https://tel.archives-ouvertes.fr/tel-02433509

Keck, J.-B., Cottet, G.-H., Meiburg, E., Mortazavi, I., & Picard, C. (2021). Double-diffusive sedimentation at high Schmidt numbers: Semi-Lagrangian simulations. *Physical Review Fluids*, *6*(2), L022301. https://doi.org/10.1103/PhysRevFluids.6.L022301

Lagaert, J.-B., Balarac, G., & Cottet, G.-H. (2014). Hybrid spectral-particle method for the turbulent transport of a passive scalar. *Journal of Computational Physics*, *260*, 127–142. https://doi.org/10.1016/j.jcp.2013.12.026

Mimeau, C., Cottet, G.-H., & Mortazavi, I. (2016). Direct numerical simulations of three-dimensional flows past obstacles with a vortex penalization method. *Computers and Fluids*, *136*, 331–347. https://doi.org/10.1016/j.compfluid.2016.06.020

Mimeau, C., Marié, S., & Mortazavi, I. (2021). A comparison of semi-Lagrangian Vortex method and Lattice Boltzmann method for incompressible flows. *Computers and Fluids*, *224*, 104946. https://doi.org/10.1016/j.compfluid.2021.104946

Mimeau, C., & Mortazavi, I. (2021). A Review of Vortex Methods and Their Applications: From Creation to Recent Advances. *Fluids*, *6*(2), 68. https://doi.org/10.3390/fluids6020068

Mimeau, C., Mortazavi, I., & Cottet, G.-H. (2017). Passive control of the flow around a hemisphere using porous media. *European Journal of Mechanics - B/Fluids*, *65*, 213–226. https://doi.org/10.1016/j.euromechflu.2017.03.002