Final projects 2023

Scientific computing tools for advanced mathematical modelling

Goal: investigate a specific mathematical and numerical model for a given problem in a short time (4 weeks). Students will implement a numerical strategy in a branch of the main repository.

Students will self-assign checkpoints and discuss their progress with the professor, assistant, and tutor. There will be two (midterm and final pitch) presentations on Thursday, May 18, and Monday, May 29.

Contacts:

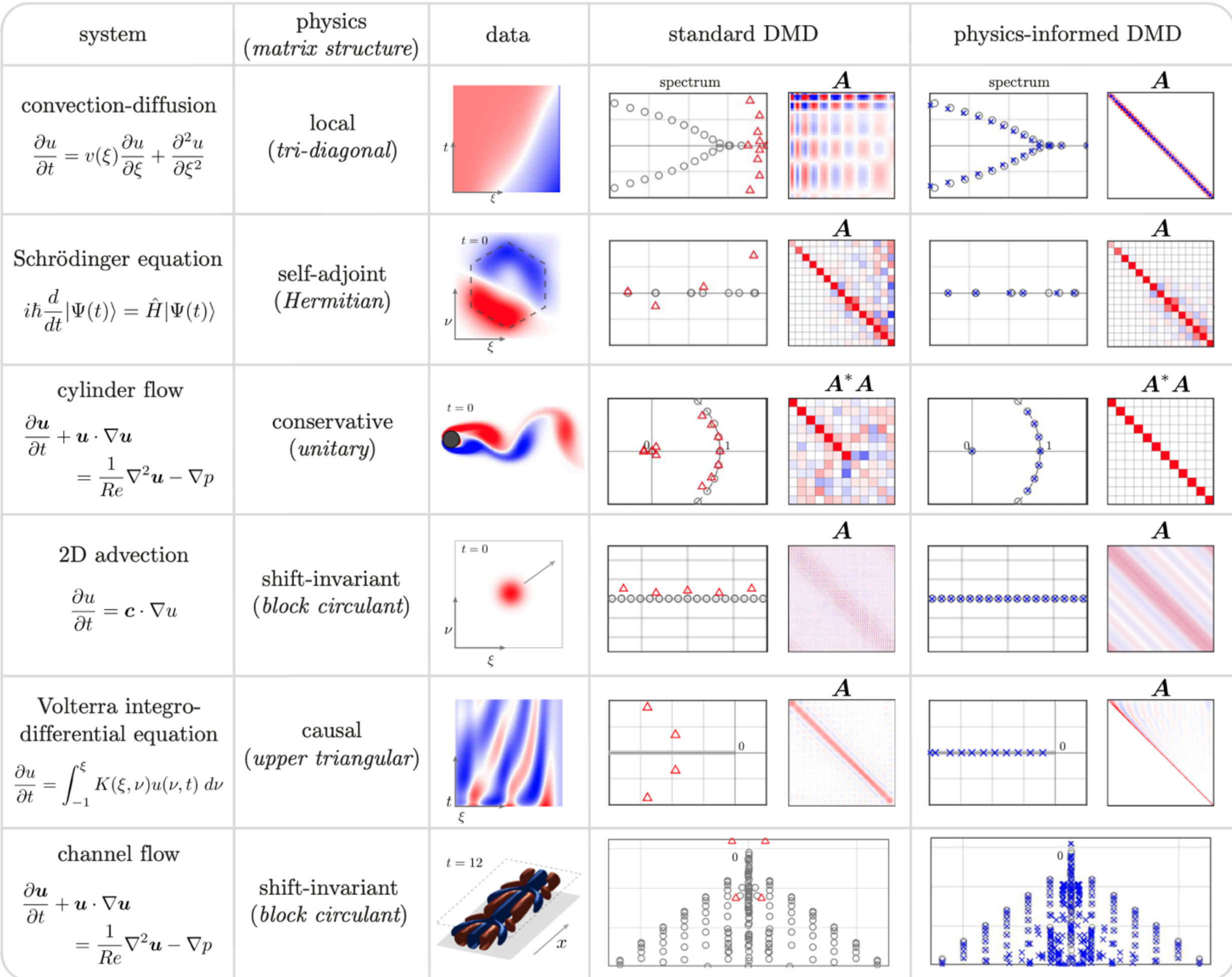
Stefano Pagani stefano.pagani@polimi.it
Roberto Piersanti roberto.piersanti@polimi.it
Paolo Joseph Baioni paolojoseph.baioni@polimi.it

Project 1: physics-informed dynamic mode decomposition

The goal of this project is to create grey box models, integrating physical principles such as symmetries, invariances, and conservation laws, into data-driven black box approaches. Students will first reproduce one of the test cases reported in [1] and then try to extend this framework to other data-driven surrogate models (like, e.g., neural networks).

[1] Baddoo, Peter J., et al. "Physics-informed dynamic mode decomposition." *Proceedings of the Royal Society A* 479.2271 (2023): 20220576.

[2] https://www.youtube.com/watch?v=lx-msllg1kU

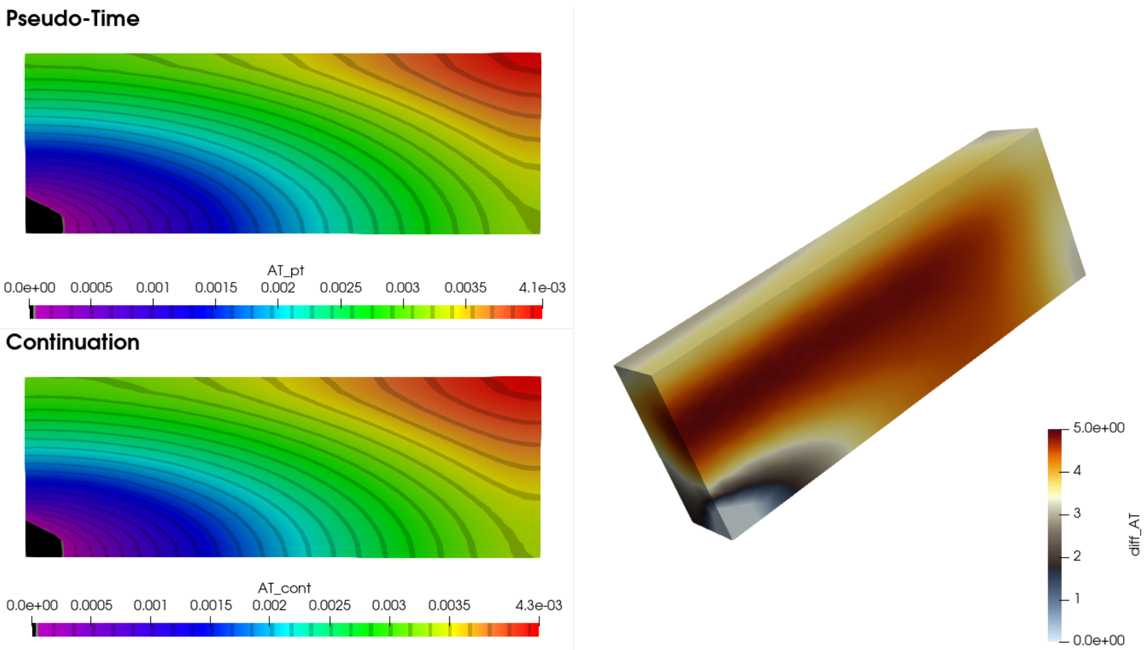


Project 2: Comparison between different numerical approaches for the Eikonal-Diffusion equation

The goal of this project is twofold: on the one hand, the students will produce a rigorous time and space convergence analysis for the standard numerical solution of the Eikonal-Diffusion equation, usign the the Pseudo-Time approach [1,2]; on the other hand, they will compare the standard approach with a new Continuation algorithm recently developed. The students will use an already implemented numerical solver based on lifex [2].

[1] P.C. Franzone et al. “Wavefront propagation in an activation model of the anisotropic cardiac tissue: asymptotic analysis and numerical simulations”. In: J Math Biol 28.2 (1990), pp. 121–176.

[2] P.C. Africa. lifeX: A flexible, high performance library for the numerical solution of complex finite element problems. SoftwareX, (2022), 101252.



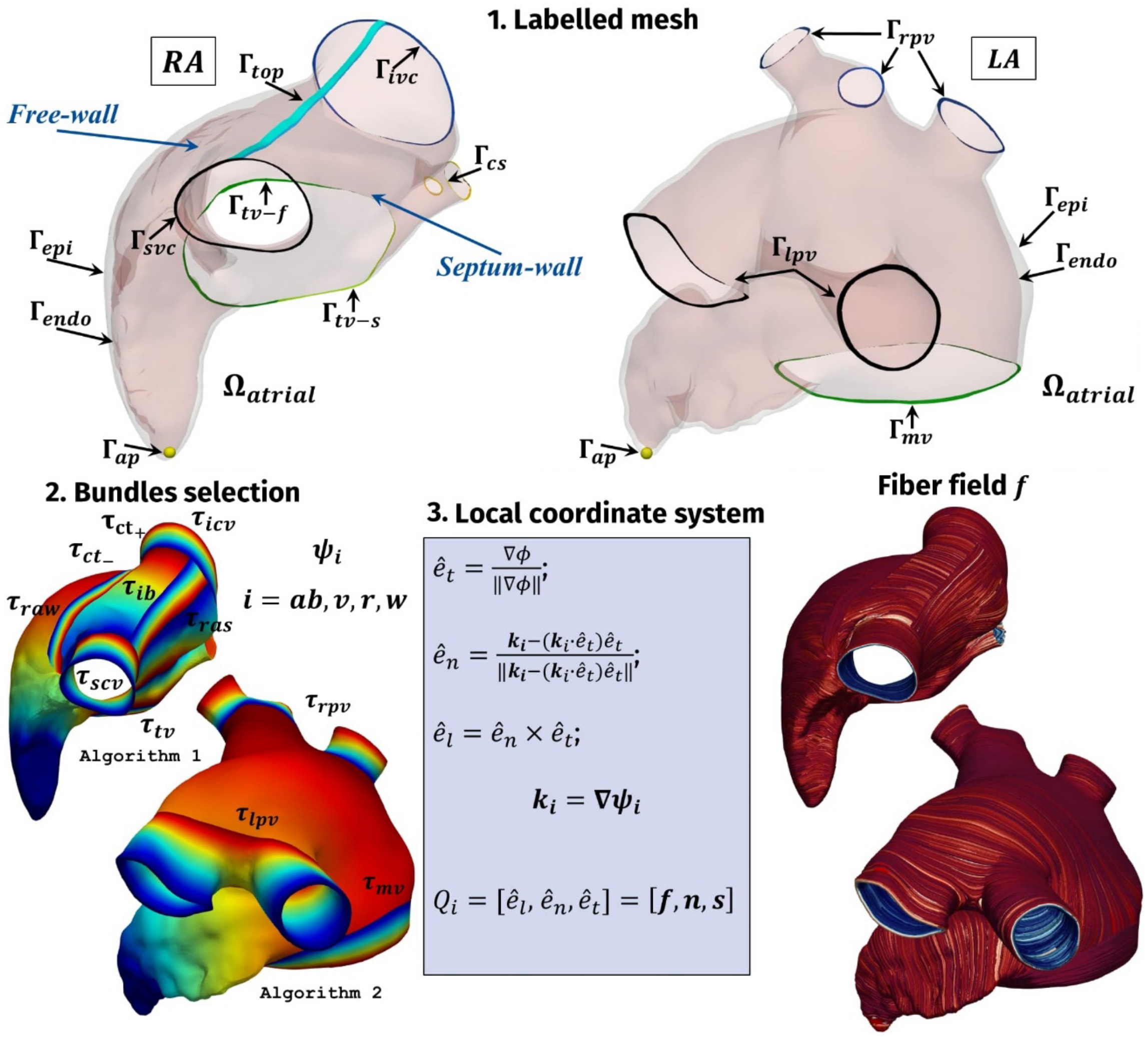
Project 3: Sensitivity analysis on the labelling procedure for generating atrial cardiac fibers using LDRBM

The goal of this project is to investigate the influence of the preprocessing labelling procedure necessary to prescribe atrial cardiac fibers using Laplace-Dirichlet-Rule-Based Method (LDRBM) [1]. In the first part of this project, the students will first manually perform different labelling procedures for a couple of atrial geometries, using vmtk [2]. Then, they will prescribe atrial fibers, usign LDRBM, on the tagged meshes. Finally, they will compare the fiber architecture on the different labelled meshed. In the final part of this project, the students will try to produce an automatic tagging procedure exploiting the geodesic distances and some morphological feature of the atrial chamber. Concerning the LDRBM, the students will use an already implemented numerical solver based on lifex [3].

[1] R. Piersanti et al. Modeling cardiac muscle fibers in ventricular and atrial electrophysiology simulations. Comput Methods Appl Mech Eng. (2021); 373:113468.

[2] M. Fedele et al. Polygonal surface processing and mesh generation tools for the numerical simulation of the cardiac function, Int. J. Numer. Methods Biomed. Eng. 37 (4) (2021) e3435.

[3] P.C. Africa. lifex : A flexible, high performance library for the numerical solution of complex finite element problems. SoftwareX, (2022), 101252.


1. Labelled mesh
RA
LA
Free-wall
Septum-wall
Γ_{top}
Γ_{ivc}
Γ_{cs}
Γ_{tv-f}
Γ_{svc}
Γ_{epi}
Γ_{endo}
Γ_{tv-s}
Ω_{atrial}
Γ_{ap}
Γ_{rpv}
Γ_{lpv}
Γ_{mv}
2. Bundles selection
τ_{ct_+}
τ_{icv}
τ_{ct_-}
τ_{raw}
τ_{ib}
τ_{ras}
τ_{scv}
τ_{tv}
ψ_i
$i = ab, v, r, w$
Algorithm 1
τ_{rpv}
τ_{lpv}
τ_{mv}
Algorithm 2
3. Local coordinate system
$\hat{e}_t = \frac{\nabla\phi}{\|\nabla\phi\|}$;
$\hat{e}_n = \frac{k_i-(k_i\cdot\hat{e}_t)\hat{e}_t}{\|k_i-(k_i\cdot\hat{e}_t)\hat{e}_t\|}$;
$\hat{e}_l = \hat{e}_n \times \hat{e}_t$;
$k_i = \nabla\psi_i$
$Q_i = [\hat{e}_l, \hat{e}_n, \hat{e}_t] = [f, n, s]$
Fiber field f

Project 4: Impact of Mass Lumping technique on the cardiac electric signal propagation using the monodomain equation

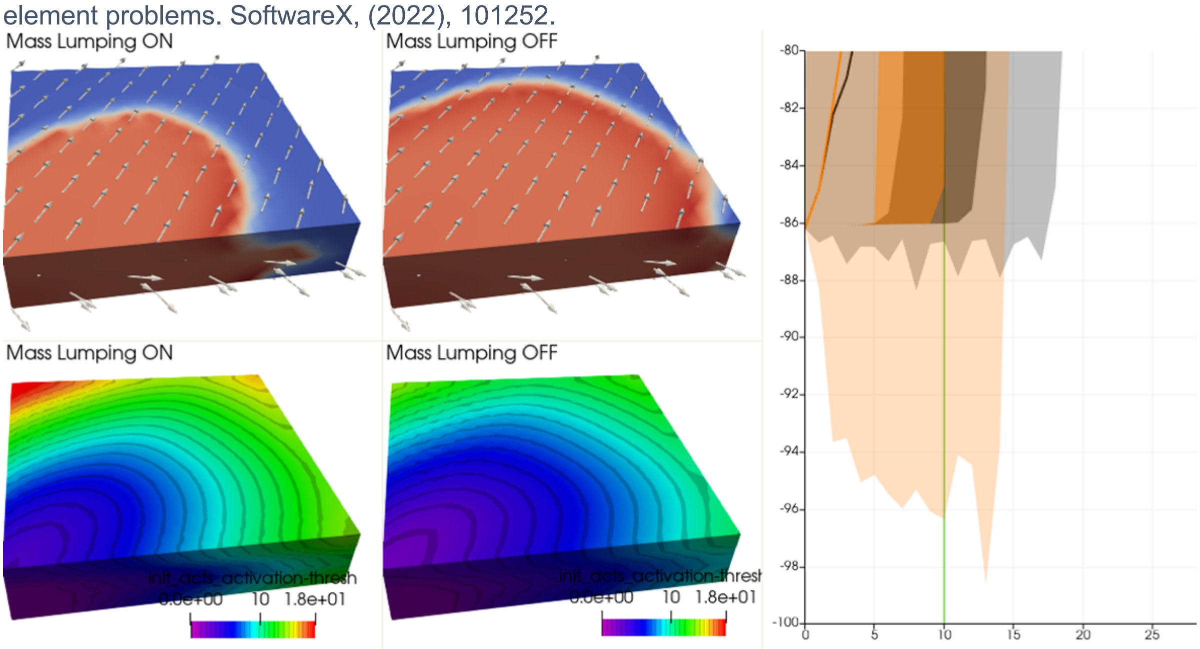
The goal of this project is to investigate the effect of different forms of mass lumping in several types of cardiac electrophysiology simulations, using the monodomain equation [1]. The students in the first part of the project will explore the effect of mass lumping, applied to the monodomain equation, on both the time derivate and the ionic current terms in a physiological benchmark simulation [2]. In the last part of this project, students will study the mass lumping effect on a pathological reentrant simulation [3]. The students will use an already implemented numerical solver based on lifex [4].

[1] A. Quarteroni et al. Integrated heart—Coupling multiscale and multiphysics models for the simulation of the cardiac function, Comput. Methods Appl. Mech. Engrg. 314 (2017) 345–407.

[2] S.A. Niederer et al. Verification of cardiac tissue electrophysiology simulators using an n-version benchmark, Phil. Trans. R. Soc. A 369 (2011) 4331–4351.

[3] J. Jilberto et al. Semi-implicit non-conforming finite-element schemes for cardiac electrophysiology: a framework for mesh-coarsening heart simulations, Front. Physiol. 9 (2018) 1513.

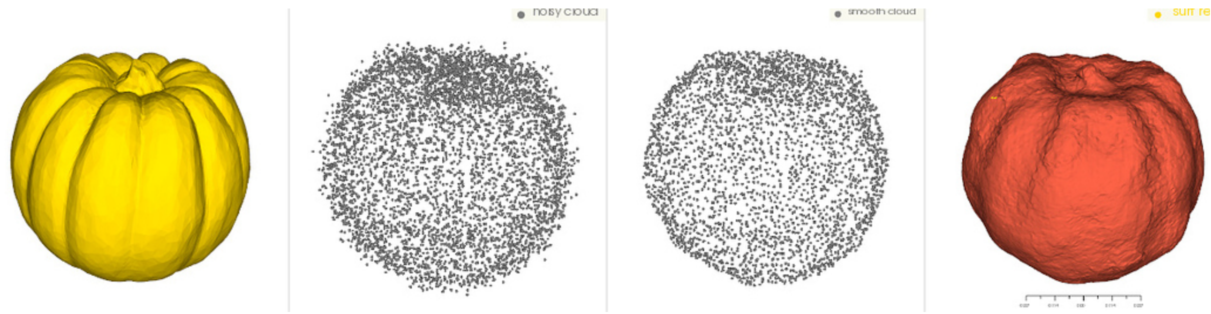
[4] P.C. Africa. lifex : A flexible, high performance library for the numerical solution of complex finite element problems. SoftwareX, (2022), 101252.



Project 5: Create a mesh surface from a point cloud data

The goal of this project is to build a procedure for constructuring a 3D mesh surface starting from point clouds given as an input. In the first part of this projct the students will explore the diffent possibilities among existing open source software at disposal from the net (e.g. MeshLab) [1]. Then in the second part, they will try to implement (in matlab or python) their own code to Construct surface mesh from 3-D point cloud [2,3].

[1] https://blogs.gre.ac.uk/designsupport/3d-realisation/laser-scanning/meshlab-point-cloud-to-mesh/
[2] https://www.mathworks.com/help/lidar/ref/pc2surfacemesh.html
[3] https://towardsdatascience.com/5-step-guide-to-generate-3d-meshes-from-point-clouds-with-python-36bad397d8ba

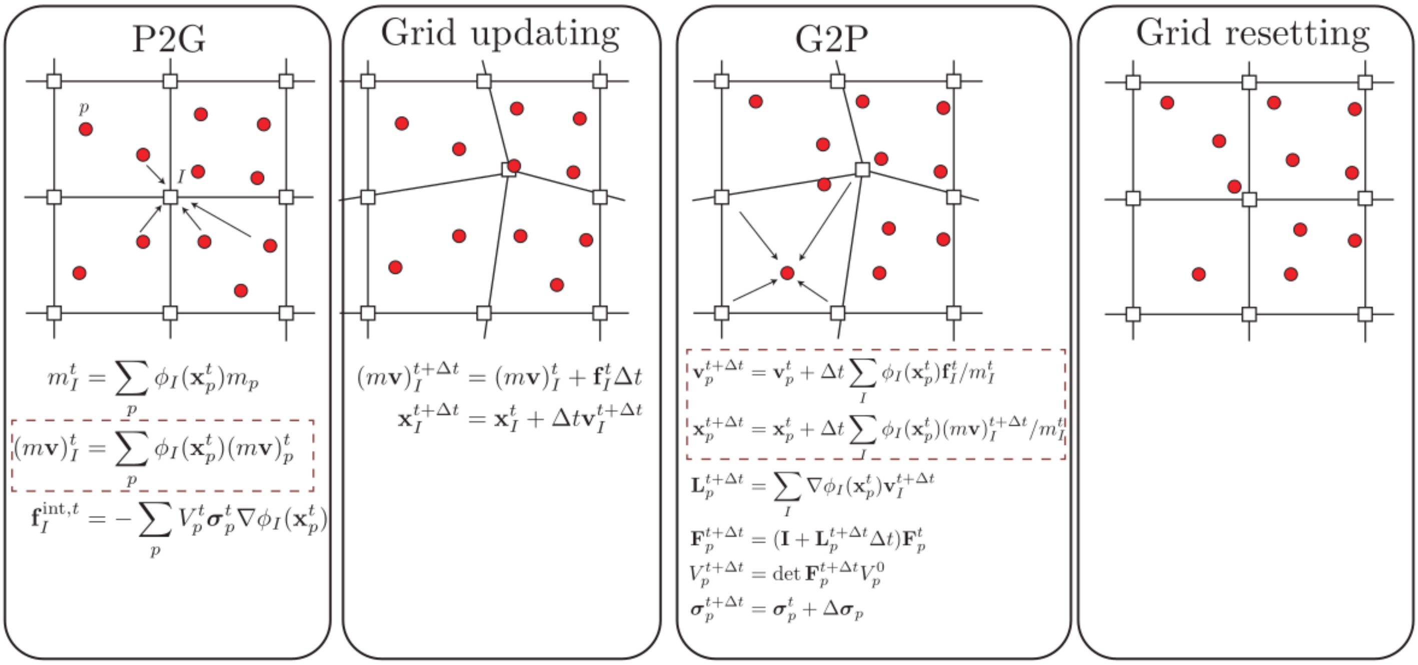


Project 6: Quadgrid extension to 3D

paolojoseph.baioni@polimi.it

Quadgrid [1] is a C++ open-source library which provides classes that can serve as starting point for the implementation of hybrid mesh–particles numerical methods.
The library is thought for 2D test-cases; the goal of this project is to extend it to address 3D problems and to test its functionalities for a later use in 3D particles dynamics simulations.

[1] https://github.com/carlodefalco/quadgrid
[2] M Griebel, G Zumbusch, S Knapek, Numerical Simulation in Molecular Dynamics, 2007, https://doi.org/10.1007/978-3-540-68095-6

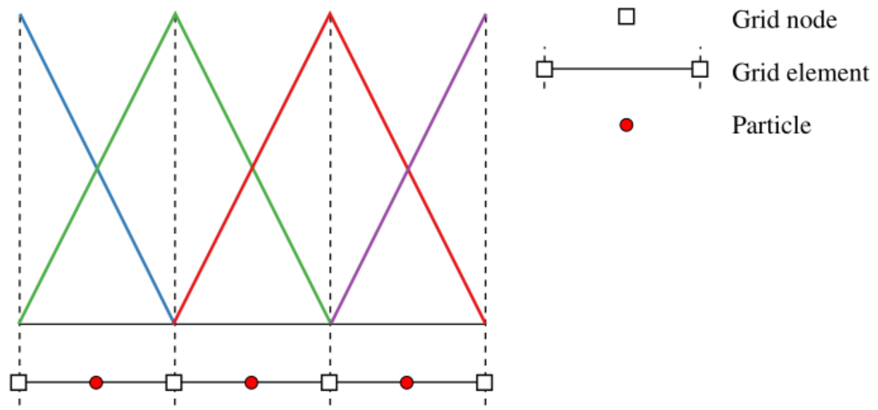


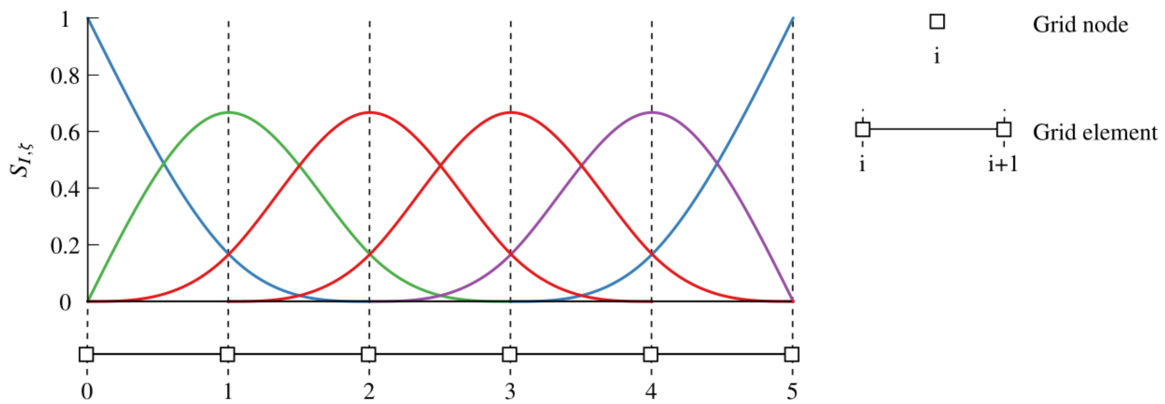
Project 7: B-Splines based kernels for Particles – Grid mappings

paolojoseph.baioni@polimi.it

Quadgrid [1] is a C++ open-source library which provides classes that can serve as starting point for the implementation of hybrid mesh–particles numerical methods.
The library core operations consist in information mappings between mesh and particles. Currently they rely on Lagrangian linear hat functions; the goal of this project is to extend such methods to higher order, B-Spline based basis functions, and to test the effectiveness of the implementation against literature test case(s), comparing its performance and accuracy w.r.t. the linear case.

[1] https://github.com/carlodefalco/quadgrid
[2] L. Piegl, W. Tiller, The NURBS Book, https://doi.org/10.1007/978-3-642-97385-7
[3] C. de Falco, A. Reali, R. Vázquez, GeoPDEs: A research tool for Isogeometric Analysis of PDEs, Advances in Engineering Software, Volume 42, Issue 12, 2011, Pages 1020-1034, ISSN 0965-9978, https://doi.org/10.1016/j.advengsoft.2011.06.010
[4] A de Vaucorbeil, VP Nguyen, S Sinaie, JY Wu. Material point method after 25 years: Theory, implementation, and applications, 2020.



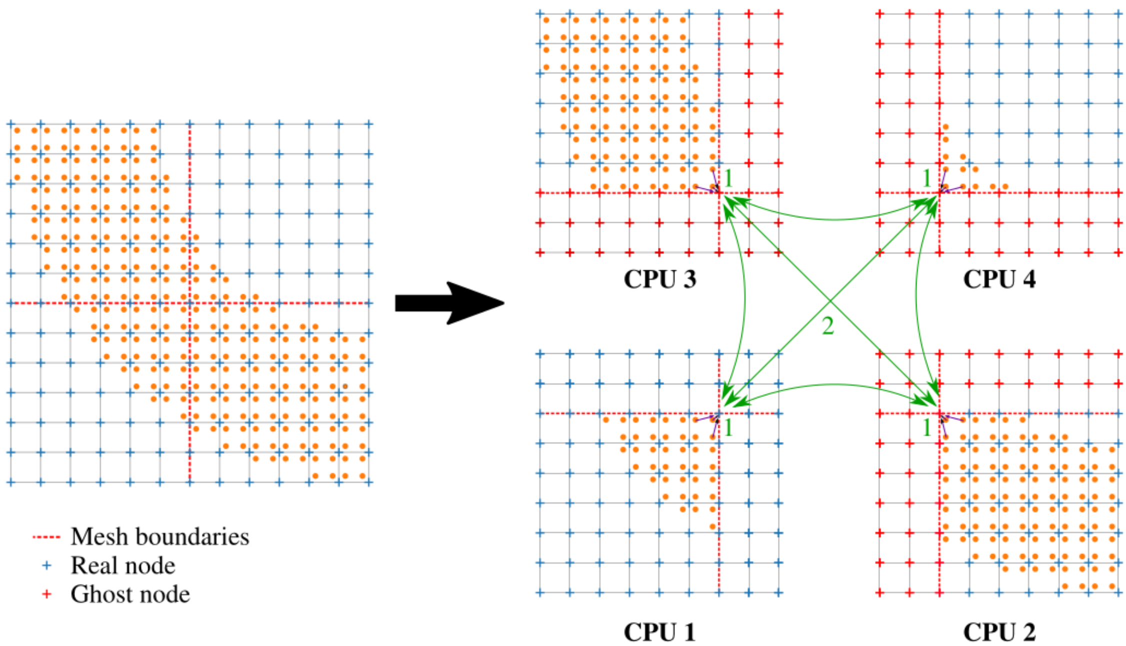


Project 8: Mesh- Particles Domain decomposition and MPI parallelization

paolojoseph.baioni@polimi.it

Quadgrid [1] is a C++ open-source library which provides classes that can serve as starting point for the implementation of hybrid mesh–particles numerical methods.
Its MPI [2] parallelization is predisposed but not realized yet; the goal of this project is to implement it and to test its effectiveness and its scalability against simple test cases, such as classical particles Newtonian dynamics [3].

[1] https://github.com/carlodefalco/quadgrid
[2] github.com/VictorEijkhout/TheArtofHPC_pdfs/blob/main/vol2/EijkhoutParallelProgramming.pdf
[3] M Griebel, G Zumbusch, S Knapek, Numerical Simulation in Molecular Dynamics, 2007, https://doi.org/10.1007/978-3-540-68095-6

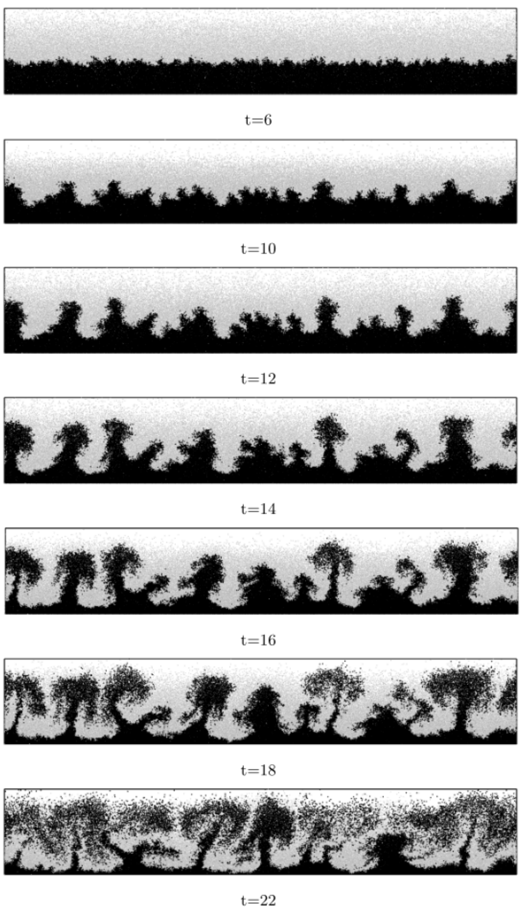


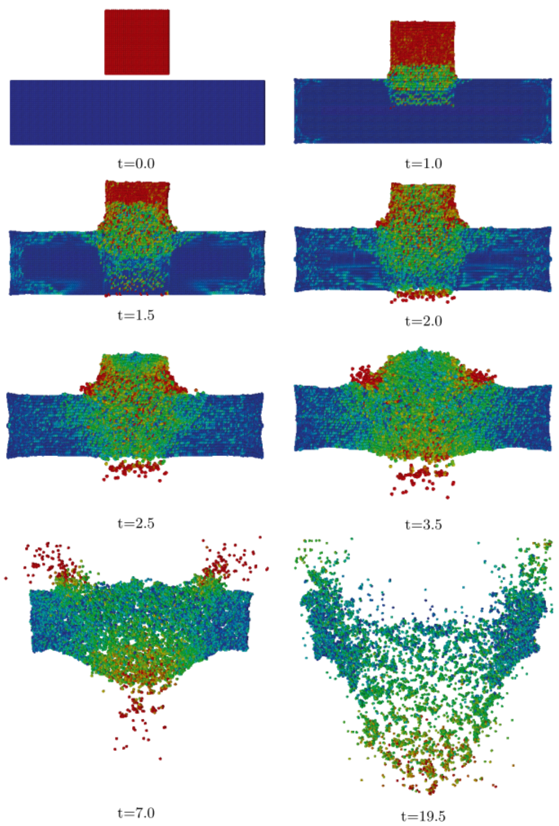
Project 9: Efficient 2D Simulations via Computational Molecular Dynamics techniques

paolojoseph.baioni@polimi.it

Quadgrid [1] is a C++ open-source library which provides classes that can serve as starting point for the implementation of hybrid mesh–particles numerical methods.
The library already contains some simple usage examples; the aim of this project is to realize a more complex test case relying on Computational Molecular Dynamics techniques, such as the “Collision of Two Bodies” or the “Rayleigh-Taylor Instability” problem at § 3.6 of [2], eventually adding useful features to the library. A C language implementation of such examples can be found in the book, and similar Matlab/Octave codes can be provided.

[1] https://github.com/carlodefalco/quadgrid
[2] M Griebel, G Zumbusch, S Knapek, Numerical Simulation in Molecular Dynamics, 2007, https://doi.org/10.1007/978-3-540-68095-6



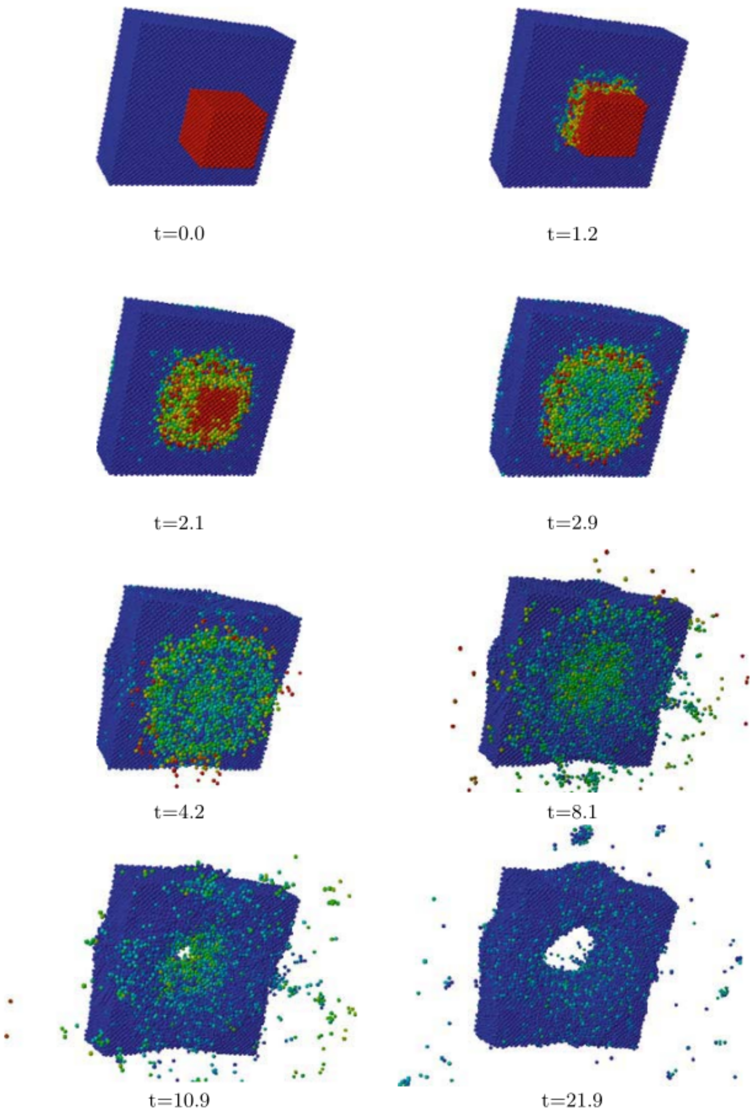


Project 10: Collision of Two Bodies 3D simulation via Computational Molecular Dynamics techniques

paolojoseph.baioni@polimi.it

The aim of this project is to realize a 3D particles dynamics simulation relying on numerical methods adopted, among others, in Computational Molecular Dynamics. A C language 2D version of such test-case can be found in § 3.6 of [1], a similar Matlab/Octave example can be provided, and some snapshot of a 3D simulation, together with some code fragments, can be found in §4.5. The project can be implemented in any programming language.

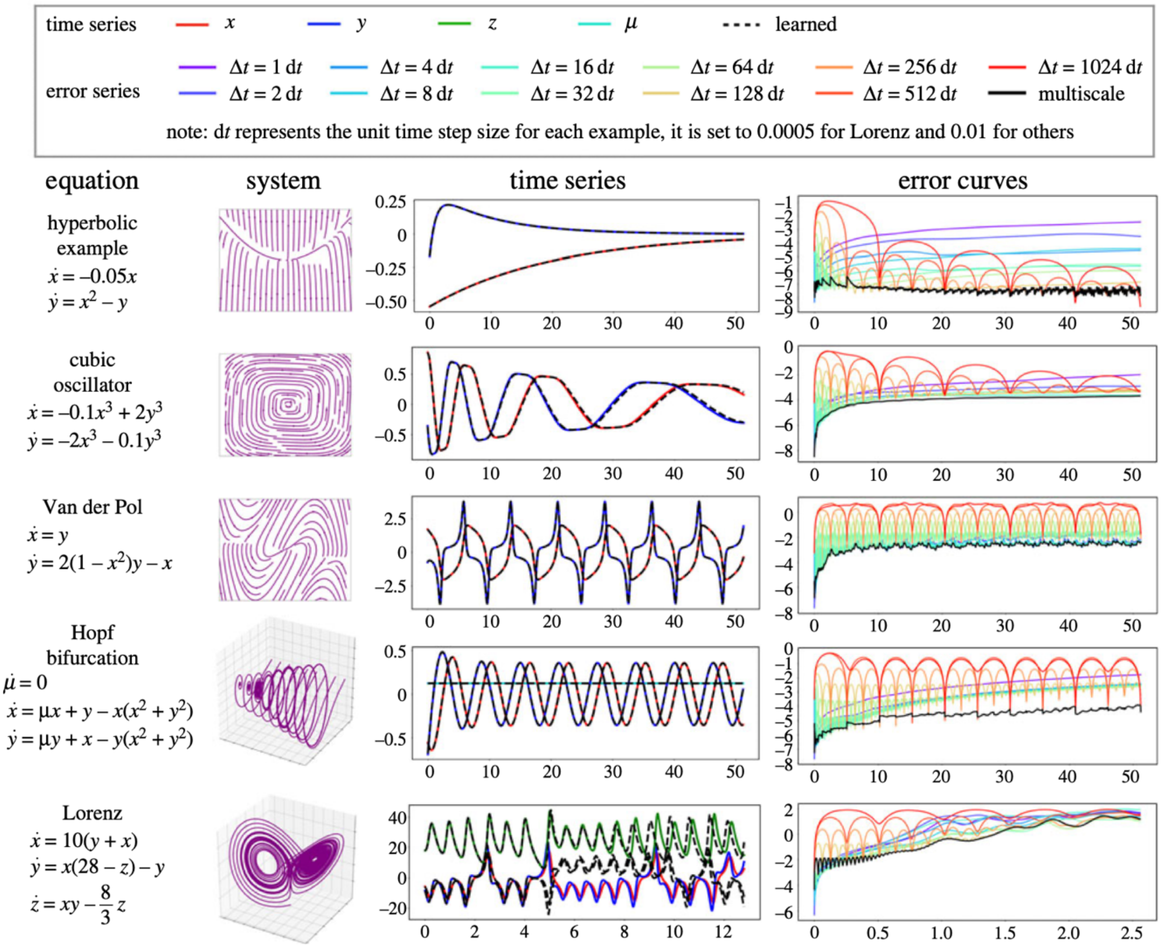
[1] M Griebel, G Zumbusch, S Knapek, Numerical Simulation in Molecular Dynamics, 2007, https://doi.org/10.1007/978-3-540-68095-6



Project 11: Hierarchical time-steppers for differential equations

The goal of this project is to investigate the effectivity of hierarchical time-steppers to capture the different dynamics present in complex multi-scale systems.
Students will first reproduce one of the test cases reported in [1] and then try to extend this framework coupled PDE-ODE systems like the one seen in guided project 1.

[1] Liu, Yuying, J. Nathan Kutz, and Steven L. Brunton. "Hierarchical deep learning of multiscale differential equation time-steppers." *Philosophical Transactions of the Royal Society A* 380.2229 (2022): 20210200.



Project 12: Multi-resolution convolutional auto-encoders

The goal of this project is to investigate the multi-resolution convolutional autoencoders for dimensionality reduction in complex multi-scale systems.
Students will first reproduce one of the test cases reported in [1] and then try to extend this framework coupled PDE-ODE systems like the one seen in guided project 1.

[1] Liu, Yuying, et al. "Multiresolution convolutional autoencoders." *Journal of Computational Physics* 474 (2023): 111801.