The Journal of Open Source Software

ExaGOOP: an AMReX-based material point method solver

Sreejith N. A.[1], **Nicholas Deak**[1], **Yudong Li**[2], **Hariswaran Sitaraman**[1], **and Marc Day**[1]

1 Scalable Algorithms, Modeling and Simulation (SAMS) Group, National Renewable Energy Laboratory, USA **2** Catalytic Carbon Transformation and Scale-Up Center, National Renewable Energy Laboratory, USA

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 25 September 2025
Published: unpublished

Summary

ExaGOOP is a versatile continuum mechanics solver based on the material point method (MPM). Traditional numerical solvers for continuum physics typically employ finite difference,volume, or element methods. These approaches require the entire computational domain to be discretized using a computational grid, where the governing equations are solved in differential, integral or variational forms. The necessity of a computational grid with complex geometries makes it challenging to solve problems involving highly deformable and history-dependent materials and multiphase systems. In contrast, MPM, like many particle-based methods, is based on a Lagrangian formulation of the governing equations. Unlike conventional solvers that rely on grids, MPM stores the material's properties on a collection of particles (also called as material points). While MPM does require a background mesh, it is most often a uniform Cartesian grid, and used only as a temporary construct for calculating gradients and is reset after each time integration step. This approach effectively eliminates issues associated with grid element deformations. As a result, MPM methods are particularly well-suited for a wide range of continuum mechanics problems, especially those that involve significant material deformations.

ExaGOOP leverages the AMReX (Zhang et al., 2019) library, which has been widely utilized in adaptive Cartesian grid and particle based applications Deak et al. (2025). The AMReX library facilitates the generation of a block-structured, Cartesian background grid within ExaGOOP and efficient parallelization using distributed memory and performance portable shared memory paradigms. The material points related operations are managed by the particle classes provided by AMReX. Currently, the implementation supports a uniform grid without refinement, but the extension to adaptive grids is part of our future efforts.

The various steps in an MPM time update include particle-to-grid (P2G), nodal velocity update, grid-to-particle (G2P), and particle position update. ExaGOOP offers users the flexibility to select the spatial discretization scheme, allowing for the use of linear-hat, quadratic B-spline, or cubic B-spline shape functions for both the P2G and G2P operations. Currently, the nodal update is performed using explicit Euler time integration; however, implicit time stepping schemes are part of ongoing work and is present in beta testing branches. In addition to these options, ExaGOOP allows users to select various numerical input parameters, such as the particle-in-cell (PIC)-Fluid Implicit Particle (FIP) blending factor in the G2P step, and whether to use Update Stress Last (USL) or Modified Update Stress Last (MUSL) for stress calculations. The solver also supports CFL-based adaptive time-stepping. At present, ExaGOOP supports barotropic fluid and linear elastic solid constitutive models. However, adding new constitutive models is relatively straightforward for users, requiring only the development of the new constitutive model function without necessitating changes to other parts of the code. Complex, static wall boundaries are simulated using the level set method, while moving boundaries can be simulated with fictitious rigid material points.

ExaGOOP has undergone extensive validation and verification using 1D, 2D, and 3D test cases, all of which are available in the GitHub repository. Preprocessing scripts in the repository enable users to generate initial material point distribution with the desired number of material points per cell for either for user-defined simple geometries or based on user-provided images of complex bodies. Users can specify the constitutive model for each material point, facilitating multi-body and multi-phase simulations with ease.

ExaGOOP was developed and is actively maintained in C++ and utilizes parallelization subroutines from the AMReX library. It employs an MPI+X approach, where Message Passing Interface (MPI) is used to distribute Cartesian grid patches and co-located particles across different distributed memory ranks. Each grid can be further divided into logical tiles, which can be distributed among threads using shared-memory OpenMP on multi-core Central-processing-units (CPU) based machines or among Graphics-processing-units (GPU) threads on NVIDIA/AMD/Intel based GPU-accelerated systems.

Statement of Need

There are numerous MPM solvers available online, such as Karamelo (Vaucorbeil et al., 2021), Matter (Blatny & Gaume, 2025), GEOS-MPM (Kumar et al., 2019), and Taichi-MPM (Hu et al., 2018), but what truly distinguishes ExaGOOP is its performance portability. This allows ExaGOOP to excel on CPU, GPU, and hybrid architectures, making it exceptionally versatile. The advanced memory management, powerful parallel processing capabilities, and robust embedded boundary support offered by the use of AMReX render ExaGOOP as a performance-portable MPM solver. The remarkable exascale performance demonstrated by AMReX in various other solvers highlights ExaGOOP's extraordinary potential to efficiently manage billions of particles on GPU-accelerated and heterogeneous computing systems.

ExaGOOP is intended for students, researchers, and engineers interested in simulating multi-material dynamics involving severe deformations. Originally developed as a tool for studying membrane compaction in high-pressure reverse osmosis applications (**?**), ExaGOOP is now being used for simulating continumm mechanics in a variety of other applications such as in lithium-ion battery manufacturing and biomass feedstock flows.

Acknowledgements

The development of this software was supported by the National Alliance for Water Innovation (NAWI), funded by the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy (EERE), Advanced Manufacturing Office, under Funding Opportunity Announcement Number DE-FOA-0001905. All of the research was performed using computational resources sponsored by the Department of Energy's Office of Energy Efficiency and Renewable Energy and located at the National Renewable Energy Laboratory. This work was authored in part by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

References

Blatny, L., & Gaume, J. (2025). Matter (v1): An open-source MPM solver for granular matter. *EGUsphere [Preprint]*. https://doi.org/10.5194/egusphere-2025-1157

Deak, N., Sitaraman, H., Lu, Y., Saha, N., Klinger, J., & Xia, Y. (2025). A high-performance discrete-element framework for simulating flow and jamming of moisture bearing biomass feedstocks. *Powder Technology*, *452*, 120548.

Esclapez, L., Day, M., Bell, J., Felden, A., Gilet, C., Grout, R., Henry de Frahan, M., Motheau, E., Nonaka, A., Owen, L., Perry, B., Rood, J., Wimer, N., & Zhang, W. (2023). PeleLMeX: an AMR Low Mach Number Reactive Flow Simulation Code without level sub-cycling. *Journal of Open Source Software*, *8*(90), 5450. https://doi.org/10.21105/joss.05450

Henry de Frahan, M. T., Esclapez, L., Rood, J., Wimer, N. T., Mullowney, P., Perry, B. A., Owen, L., Sitaraman, H., Yellapantula, S., Hassanaly, M., Rahimi, M. J., Martin, M. J., Doronina, O. A., A., S. N., Rieth, M., Ge, W., Sankaran, R., Almgren, A. S., Zhang, W., … Chen, J. H. (2024). The pele simulation suite for reacting flows at exascale. *Proceedings of the 2024 SIAM Conference on Parallel Processing for Scientific Computing*, 13–25. https://doi.org/10.1137/1.9781611977967.2

Hu, Y., Fang, Y., Ge, Z., Qu, Z., Zhu, Y., Pradhana, A., & Jiang, C. (2018). A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Transactions on Graphics (SIGGRAPH)*, *37*(4), 1–14. https://doi.org/10.1145/3197517.3201309

Kumar, K., Salmond, J., Kularathna, S., Wilkes, C., Tjung, E., Biscontin, G., & Soga, K. (2019). Scalable and modular material point method for large-scale simulations. *arXiv Preprint*. https://arxiv.org/abs/1909.13380

N. A., S., Sitaraman, H., Deak, N., & Day, M. (2024). *Spectral analysis of regular material point method and its application to study high pressure reverse osmosis membrane compaction and embossing*. National Renewable Energy Laboratory (NREL), Golden, CO (United States). https://www.osti.gov/biblio/2447832

Sitaraman, H., Yellapantula, S., Henry de Frahan, M. T., Perry, B., Rood, J., Grout, R., & Day, M. (2021). Adaptive mesh based combustion simulations of direct fuel injection effects in a supersonic cavity flame-holder. *Combustion and Flame*, *232*, 111531. https://doi.org/https://doi.org/10.1016/j.combustflame.2021.111531

Vaucorbeil, A. de, Nguyen, V. P., & Nguyen-Thanh, C. (2021). Karamelo: An open source parallel c++ package for the material point method. *Computational Particle Mechanics*, *8*(4), 767–789. https://doi.org/10.1007/s40571-020-00369-8

Zhang, W., Almgren, A., Beckner, V., Bell, J., Blaschke, J., Chan, C., Day, M., Friesen, B., Gott, K., Graves, D., & others. (2019). AMReX: A framework for block-structured adaptive mesh refinement. *The Journal of Open Source Software*, *4*(37), 1370.