The Journal of Open Source Software

ExaDEM: a HPC application based on exaNBody targeting scalable DEM simulations with complex particle shapes

Raphaël Prat [1], **Thierry Carrard** [2,3], **Lhassan Amarsid** [1], **Vincent Richefeu** [4], **Carlo-elia Doncecchi** [1], **Paul Lafourcade** [2,3], **Guillaume Latu** [1], **and Jean-Mathieu Vanson** [1]

1 CEA, DES, IRESNE, DEC, Cadarache F 13108 St-Paul-Lez-Durance **2** CEA, DAM, DIF, F-91297 Arpajon, France **3** Université Paris-Saclay, LMCE, F-91680 Bruyères-le-Châtel, France **4** 3SR, CNRS, University of Grenoble Alpes, Grenoble, 38400, France

DOI: 10.21105/joss.07484

Software

- Review
- Repository
- Archive

Editor: Jed Brown

Reviewers:

- @vsangelidakis
- @slamont1

Submitted: 27 June 2024
Published: 28 February 2025

Summary

ExaDEM is a Discrete Element Method (DEM) code developed using the exaNBody framework (Carrard et al., 2023) at the French atomic commission (CEA). This software provides DEM modeling capabilities to enable the modeling of mechanical interactions occuring at contact between spheres and polyhedra particles while offering performance optimizations on modern HPC platforms. A notable aspect of ExaDEM stands in its hybrid parallelization approach, combining the use of MPI and Threads (OpenMP). Additionally, ExaDEM offers portability to clusters of GPUs, using the CUDA programming model and MPI for DEM simulations with spherical particles. Developed in C++17, ExaDEM is designed to offer HPC capabilities to the scientific community.

Statement of need

A comprehesive description of the behavior of granular media remains a significant challenge for the scientific community. DEM simulations enhance our knowledge by examining phenomena that are otherwise unreachable or too costly to study through experiments. To accurately reproduce these phenomena originated at particle scale, we must simulate a representative number of particles, ranging from thousands to billions. While a single processor can easily handle simulations involving thousands of particles, simulating millions requires HPC resources. These simulations are constrained either by memory footprint or excessively long execution time. The DEM method is inherently parallel, with various techniques avaiable, such as spatial domain decomposition (Plimpton, 1995), thread parallelization over cells and vectorization. In this paper, we introduce the code ExaDEM, designed for large-scale DEM simulation on HPC platforms. This code leverages numerous features of the exaNBody framework, including data structure management, MPI+X parallelization and add-on modules (optimized IO, Paraview export).

DEM Background

The DEM method, employed to study granular media, falls within the scope of so-called N-body methods. It involves numerically replicating the kinetics and interactions of a collection of rigid particles over time. Time is divided into discrete steps, where at each step n, Newton's equation **f**=m**a** is solved to determine the translation and rotation acceleration for each particle.

Subsequently, the corresponding velocity and moment are computed, leading to the calculation of the new position at step n+1. The force **f** is evaluated from particle interactions, including contact interactions (normal and tangential force) and external forces such as gravity. A commonly employed numerical scheme is the Velocity Verlet, while the Hooke law is often used to model contact interactions. The versatility of the DEM method allows for simulating rigid bodies with various shapes, ranging from spherical to polyhedral particles. Note that in ExaDEM, complex particle shapes are handled using a sphero-polyhedral approach, facilitating simplified contact detection and representing intricate shapes (Alonso-Marroquín, 2008). For clarity, the term "polyhedron" is used to refer to a sphero-polyhedral particle.

Efficient neighbor particle detection is a critical aspect of DEM simulation codes. Typically, this is achieved by combining the linked cells method (Ciccotti et al., 1987) and Verlet lists (Verlet, 1967), which optimize neighbor searches using a cartesian grid of cells (with a complexity of N, N is the number of particles) while limiting the refresh rate of neighbor lists.

Several DEM software packages have emerged in recent years, offering HPC capabilities. Examples include LIGGGHTS (Kloss et al., 2012), which is based on LAMMPS (Thompson et al., 2022) data structures for Molecular Dynamics simulations with spherical particles (MPI), and Blaze-DEM (Govender et al., 2018) which employs spheres and polyhedra on GPU via CUDA. ExaDEM aims to establish itself as a software solution that combines MPI parallelization, OpenMP thread parallelization for both polyhedral and spherical particles, and CUDA GPU parallelization for spherical particles. Similar to LIGGGHTS with LAMMPS, ExaDEM benefits from various HPC features developed in the Molecular Dynamics code ExaSTAMP (Cieren et al., 2014) that have been mutualized in the exaNBody framework such as AMR data structures (Prat et al., 2018) with load balancing (Prat et al., 2020), generation of compact particle configurations (Josien & Prat, 2024), and In-situ analysis (Dirand et al., 2018). ExaDEM aims to incorporate the physics of interest from the Rockable (Richefeu et al., 2025; Richefeu & Villard, 2016) (open source), originally created at CNRS for polyhedra on a HPC framework. To the best of our knowledge, this is a non-exhaustive list of other well-known DEM codes: EDEM (not open source), Rocky DEM (not open source), MercuryDPM (Weinhart et al., 2020) (open source), and Yade (Smilauer et al., 2023) (open source).

Implementation Details

ExaDEM leverages the exaNBody data structures (grid, cells, fields) as well as key parallel algorithms (domain decomposition, particles migration, numerical schemes) while proposing DEM-specific mechanism. ExaDEM achieves a MPI parallelization where the simulation domain is decomposed into subdomains using spatial domain decomposition and the Recursive Coordinate Bisection (RCB) partitioning method to evenly distribute the workload among MPI processes. Particle information stored within these cells, and a subdomain corresponds to a grid of cells. The use of cells aims to apply the state-of-the-art linked cells method to expedite neighbor searches with a complexity of O(N), where N represents the number of particles. Additionally, the Verlet lists method maintains larger neighbor lists over timesteps as long as a particle has a displacement shorter than half of the Verlet radius. Regarding the data layout, it is decomposed into two levels. The first level consists of cells (SOA), each composed of fields (Array) containing particle data. The second level is associated with the grid of cells (AOSOA), corresponding to a subdomain. The DEM grid includes the following fields: type, position, velocity, acceleration, radius, angular velocity, orientation. The AOSAO data structure facilitates data movement between MPI processes while maintaining good data locality, ensuring that particles in the same cell or neighboring cells interact with a good memory hit ratio. Moreover, the use of SOA storage (cell layout) enhances the use of SIMD instructions.

Regarding the intra-MPI parallelization, we distinguish two main differences based on the type of particle, i.e. sphere or polyhedron:

- For spherical particles, OpenMP parallelization involves iterating over cells. In GPU paral-

lelization, a block of GPU threads is assigned to a cell, with each GPU thread processing a particle.

- For polyhedral particles, another parallel level is chosen for thread parallelization, the interaction. Unlike spherical particles, polyhedra can have multiple contacts of different types (vertex-vertex, vertex-edge, vertex-face, edge-edge). Hence, it is more effective to consider interactions rather than cells to achieved thread-parallelization. However, this strategy introduces synchronizations, such as the usage of mutexes. The GPU parallelization of polyhedra is still an upcoming development.

In conclusion, the design of ExaDEM, guided by the exaNBody framework, facilitates the addition or removal of individual operators or features without disrupting other functionalities, provided theses operators are independents. For instance, removing the `gravity_force` from the ExaDEM repository is feasible, while ensuring the preservation of the `contact_neighbor` operator (building neighbor lists for every particle), which is crucial for running the `hooke_force` operator. Significant efforts have been made to minimize interactions between operators, simplifying the process of adding or removing new modules/operators.

Main Features

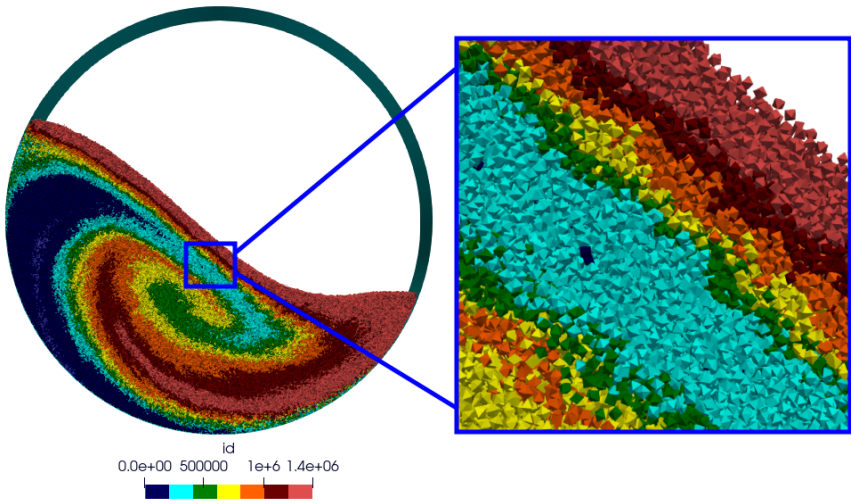

Figure 1: Simulation of 700 thousand octahedra in a rotating drum running on 128 MPI processes with 8 OpenMP threads per MPI process (processor: AMD EPYC Milan 7763).

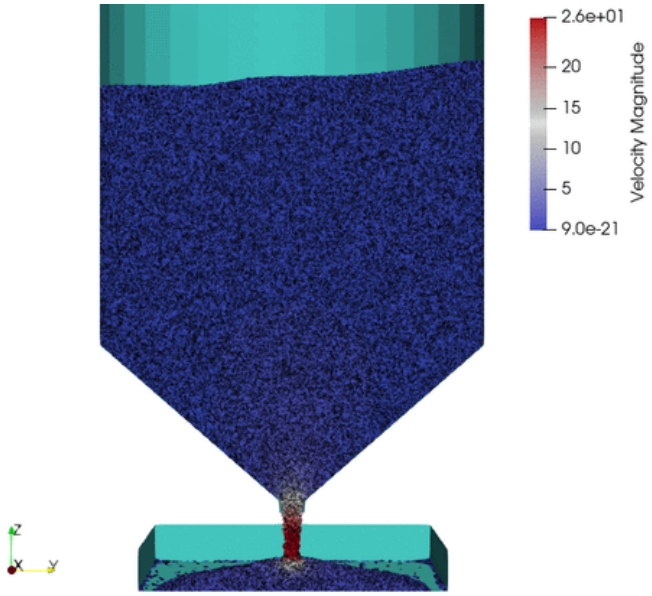

Figure 2: Simulation of 20 million spherical particles falling into a funnel. This simulation runs on 512 MPI processes with 8 OpenMP threads per MPI process (processor: AMD EPYC Milan 7763).

ExaDEM aims to to meet scientific expectations, particularly for nuclear fuel simulations involving scenarios such as rotating drums (see Figure 1) or compaction processes. To achieve this goal, ExaDEM provides the following key features:

- Handling of different particle types: spherical and polyhedral,
- Hybrid parallelisation MPI + X,
  - X = OpenMP or CUDA for spherical particles,
  - X = OpenMP for polyhedral particles,
  - The Recursive Coordinate Bissection method is used for the load balancing,
- I/O support for check and restart files (MPI-IO files),
- Paraview output files containing fields,
- Drivers: rigid wall, rotating drum or mesh of polyhedron surface for complex geometries such as funnel (see Figure 2),
- Time integration scheme: Velocity Verlet,
- Contact detection: Linked-cell method and Verlet Lists,
- Force fields: contact force (Hooke law), cohesive force, and gravity.

All these functionalities are likely to evolve to accomodate new development needs, such as the addition of particle fragmentation. It is worth noting that most of these functionalities have been rigorously tested over 500 million spheres or 10 million polyhedra over ten thousand cores with hybrid MPI + OpenMP programming on AMD EPYC Milan 7763 processors.

Acknowledgements

This work was performed using HPC resources from CCRT funded by the CEA/DEs simulation program. ExaDEM is part of the PLEIADES platform which has been developped in collaboration with the French nuclear industry - mainly CEA, EDF, and Framatome - for simulation of fuel elements.

References

Alonso-Marroquín, F. (2008). Spheropolygons: A new method to simulate conservative and dissipative interactions between 2D complex-shaped rigid bodies. *Europhysics Letters*, *83*(1), 14001. https://doi.org/10.1209/0295-5075/83/14001

Carrard, T., Prat, R., Latu, G., Babilotte, K., Lafourcade, P., Amarsid, L., & Soulard, L. (2023). ExaNBody: A HPC framework for n-body applications. *European Conference on Parallel Processing*, 342–354. https://doi.org/10.1007/978-3-031-50684-0_27

Ciccotti, G., Frenkel, D., & Mc Donald, I. R. (1987). *Simulation of liquids and solids*. Elsevier. ISBN: 978-0-44487-062-9

Cieren, E., Colombet, L., Pitoiset, S., & Namyst, R. (2014). ExaStamp: A parallel framework for molecular dynamics on heterogeneous clusters. *Euro-Par 2014: Parallel Processing Workshops: Euro-Par 2014 International Workshops, Porto, Portugal, August 25-26, 2014, Revised Selected Papers, Part II 20*, 121–132. https://doi.org/10.1007/978-3-319-14313-2_11

Dirand, E., Colombet, L., & Raffin, B. (2018). Tins: A task-based dynamic helper core strategy for in situ analytics. *Supercomputing Frontiers: 4th Asian Conference, SCFA 2018, Singapore, March 26-29, 2018, Proceedings 4*, 159–178. https://doi.org/10.1007/978-3-319-69953-0_10

Govender, N., Wilke, D. N., Pizette, P., & Abriak, N.-E. (2018). A study of shape non-uniformity and poly-dispersity in hopper discharge of spherical and polyhedral particle systems using the Blaze-DEM GPU code. *Applied Mathematics and Computation*, *319*, 318–336. https://doi.org/10.1016/j.amc.2017.03.037

Josien, M., & Prat, R. (2024). Parallel and bias-free RSA algorithm for maximal Poisson-sphere sampling. *Computer Physics Communications*, *305*, 109354. https://doi.org/10.1016/j.cpc.2024.109354

Kloss, C., Goniva, C., Hager, A., Amberger, S., & Pirker, S. (2012). Models, algorithms and validation for opensource DEM and CFD–DEM. *Progress in Computational Fluid Dynamics, an International Journal*, *12*(2-3), 140–152. https://doi.org/10.1504/PCFD.2012.047457

Plimpton, S. (1995). Fast parallel algorithms for short-range molecular dynamics. *Journal of Computational Physics*, *117*(1), 1–19. https://doi.org/10.2172/10176421

Prat, R., Carrard, T., Soulard, L., Durand, O., Namyst, R., & Colombet, L. (2020). AMR-based molecular dynamics for non-uniform, highly dynamic particle simulations. *Computer Physics Communications*, *253*, 107177. https://doi.org/10.1016/j.cpc.2020.107177

Prat, R., Colombet, L., & Namyst, R. (2018). Combining task-based parallelism and adaptive mesh refinement techniques in molecular dynamics simulations. *Proceedings of the 47th International Conference on Parallel Processing*, 1–10. https://doi.org/10.1145/3225058.3225085

Richefeu, V., Combe, G., Villard, P., Delenne, J.-Y., Amarsid, L., Nezamabadi, S., Radjai, F., Vanson, J.-M., Prat, R., & Mutabaruka, P. (2025). *Rockable*. https://hal.univ-grenoble-alpes.fr/hal-04933604

Richefeu, V., & Villard, P. (2016). *Modeling gravity hazards from rockfalls to landslides* (V. Richefeu & P. Villard, Eds.). Elsevier. https://doi.org/10.1016/c2015-0-01294-1

Smilauer, V., Angelidakis, V., Catalano, E., Caulk, R., Chareyre, B., Chevremont, W., Dorofeenko, S., Duriez, J., Dyck, N., Elias, J., & others. (2023). Yade documentation. *arXiv Preprint arXiv:2301.00611*. https://doi.org/10.48550/arXiv.2301.00611

Thompson, A. P., Aktulga, H. M., Berger, R., Bolintineanu, D. S., Brown, W. M., Crozier,

P. S., In't Veld, P. J., Kohlmeyer, A., Moore, S. G., Nguyen, T. D., & others. (2022). LAMMPS-a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Computer Physics Communications*, *271*, 108171. https://doi.org/10.1016/j.cpc.2021.108171

Verlet, L. (1967). Computer" experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Physical Review*, *159*(1), 98. https://doi.org/10.1103/physrev.159.98

Weinhart, T., Orefice, L., Post, M., Schrojenstein Lantman, M. P. van, Denissen, I. F., Tunuguntla, D. R., Tsang, J., Cheng, H., Shaheen, M. Y., Shi, H., & others. (2020). Fast, flexible particle simulations—an introduction to MercuryDPM. *Computer Physics Communications*, *249*, 107129. https://doi.org/10.1016/j.cpc.2019.107129