AutoPas:
Dynamic Algorithm Selection in Molecular Dynamics for Optimal Time and Energy

Samuel J. Newcome[1], Fabio A. Gratl[1], Markus Muelhaeusser[1], Philipp Neumann[2], Hans-Joachim Bungartz[1]

[1]Chair of Scientific Computing in Computer Science, Technical University of Munich
[2]Chair of High Performance Computing, Helmut Schmidt University

Overview

Many algorithms exist for the effecient evaluation of short-range pairwise interactions in particle simulations, all with their strengths and weaknesses leading to different optimal algorithms in different scenarios.

AutoPas is a black-box particle container which a user can build their simulator on top of. It contains an extensive library of rank-level algorithms and parallelisations and can switch between these algorithms during the course of the simulation.
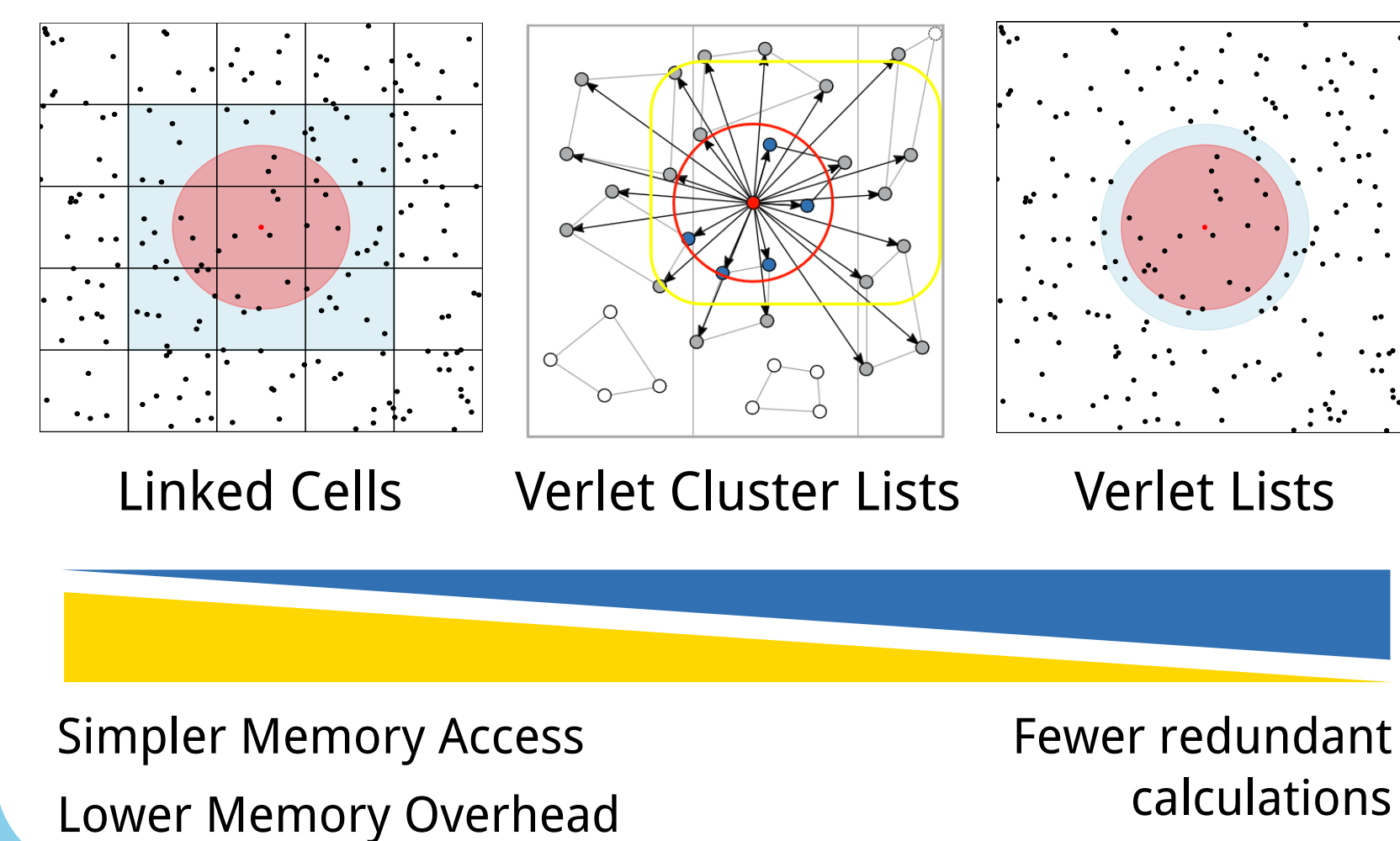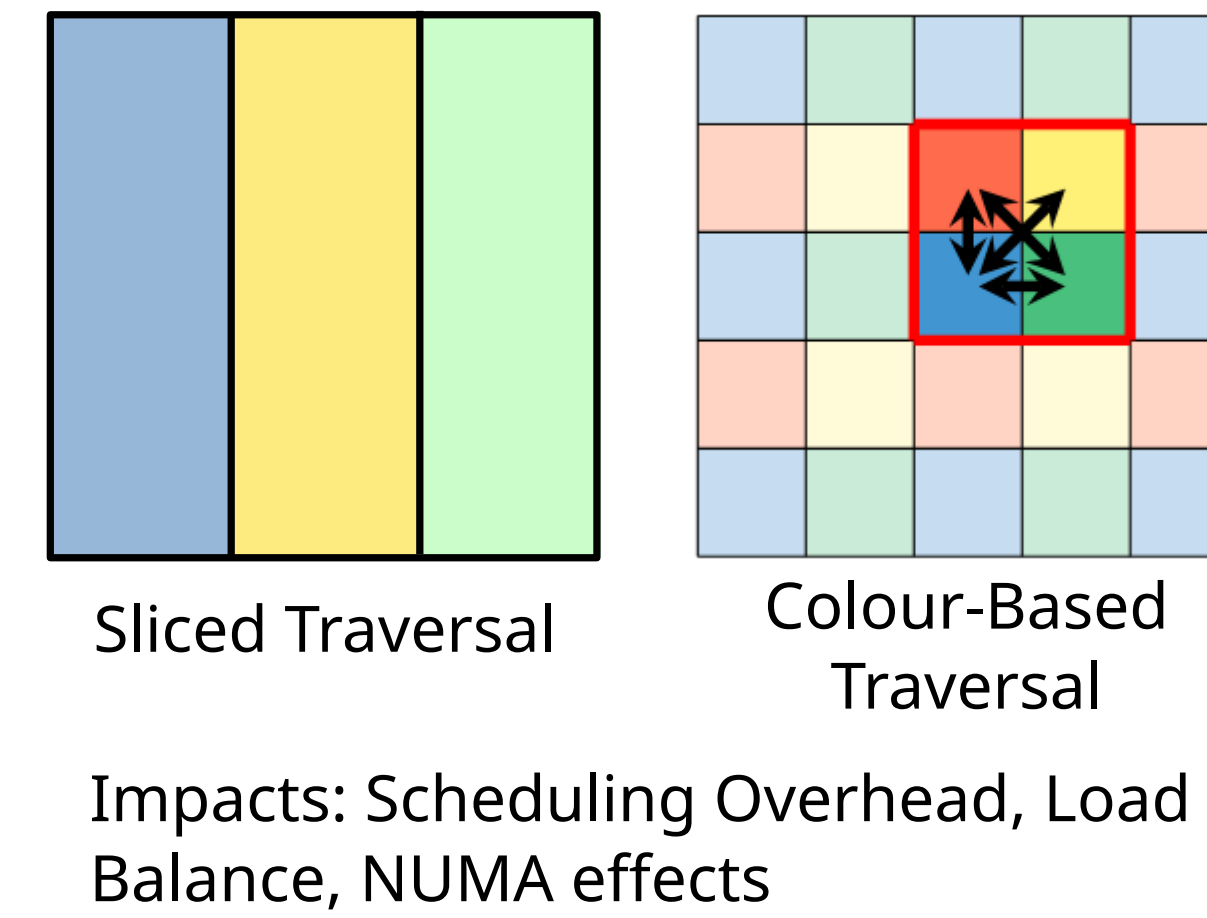
AutoPas selects the optimal algorithm by trailling them at intervals across the simulation. As some algorithms can perform far worse than other algorithms for some given scenario, a tuning strategy is used as a heuristic for which algorithms to eliminate.

Different Scenarios lead to Different Optimal Algorithms

A Homogeneous Simulation
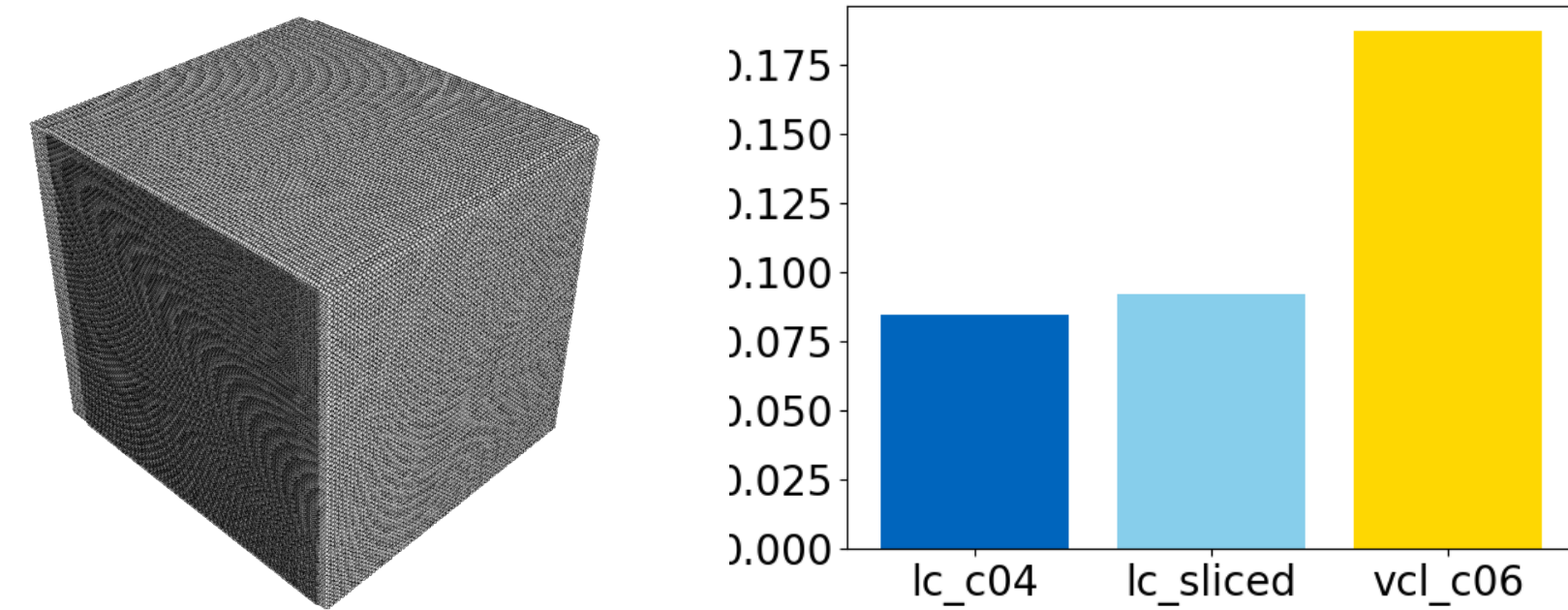
A Heterogeneous Simulation
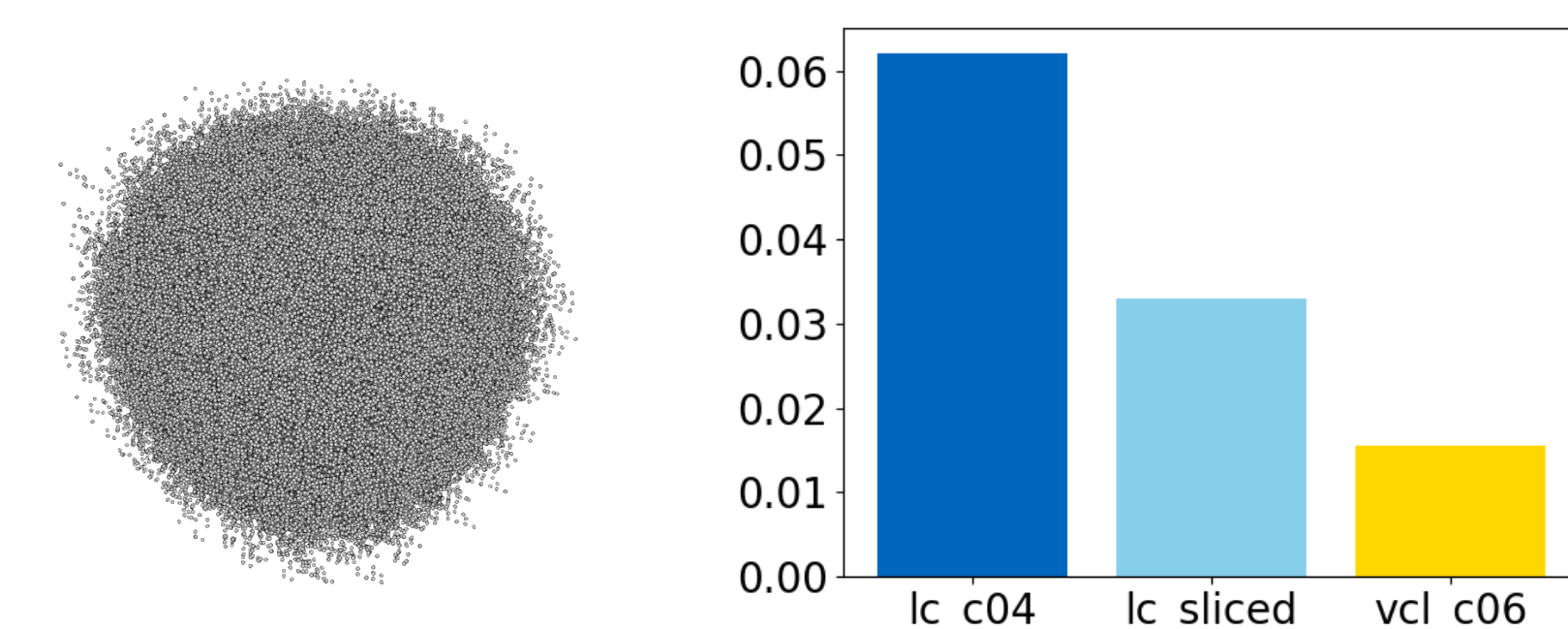
Rules-Based Tuning

An expert can write rules for excluding algorithms depending on the scenario:

Example: If less than 6 particles per cell, then exclude Structure-of-Arrays

Energy Cost Vs Time

Work by Vincent Fisher

Algorithm Library

Particle Containers

Linked Cells   Verlet Cluster Lists   Verlet Lists

Simpler Memory Access
Lower Memory Overhead   →   Fewer redundant calculations

Shared-Memory Parallel Traversals

Sliced Traversal   Colour-Based Traversal

Impacts: Scheduling Overhead, Load Balance, NUMA effects

Data Structures

* Array-of-Structures
* Structure-of-Arrays

Tunable Parameters

* Cell Sizes
* Verlet Skin Sizes
* OpenMP Chunk Sizes (ToDo)

Optimisation Goal

* Time
* Energy Consumption

Energy Consumption is available only on Intel processors using Intel's RAPL interface.

AutoPas

Tuning Strategy

Heuritic for eliminating poorly performing algorithmic configurations.

* Expert Knowledge Rules Based Tuning
* Predictive Tuning
* Descion Trees Based Tuning (ToDo)

User Interface

User Simulator

ls1-MarDyn MD Sim.

Fork of LAMMPS
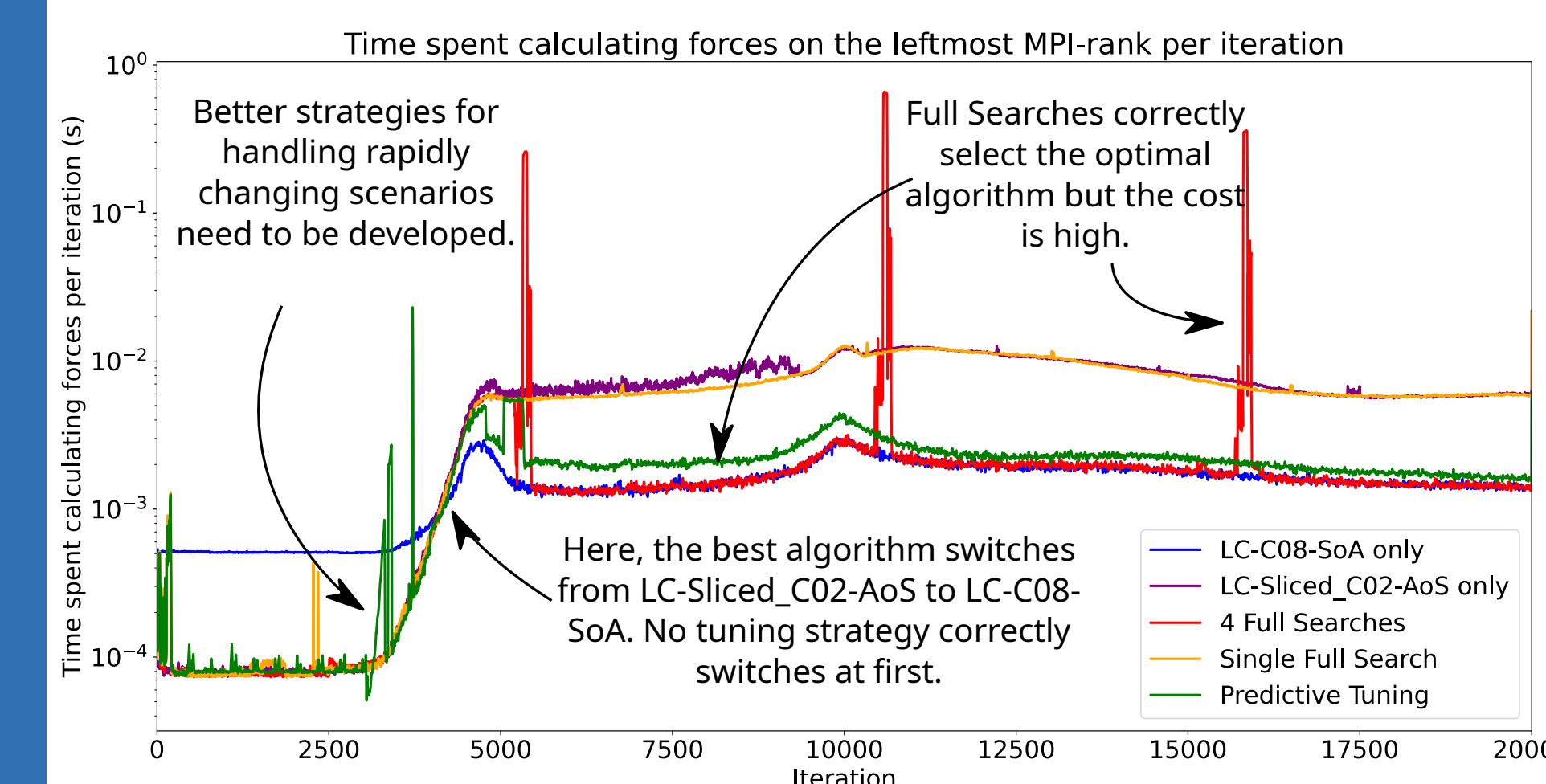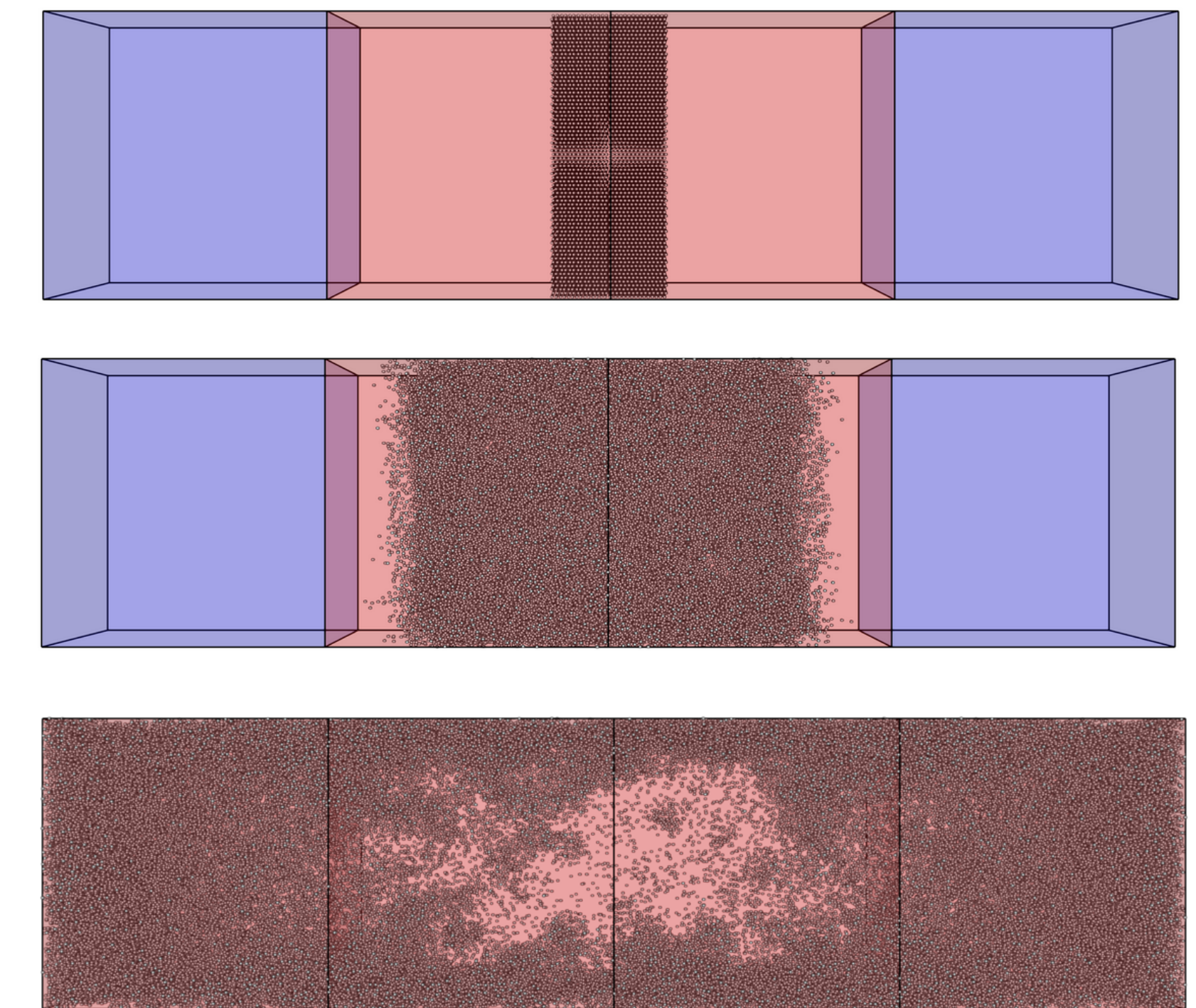
Space Debris Sim. European Space Agency

Your Simulator?

An Exploding Liquid

A densely packed liquid explods outwards resulting in a sparse inhomogeneously populated domain.

Time spent calculating forces on the leftmost MPI-rank per iteration

Better strategies for handling rapidly changing scenarios need to be developed.

Full Searches correctly select the optimal algorithm but the cost is high.

Here, the best algorithm switches from LC-Sliced_C02-AoS to LC-C08-SoA. No tuning strategy correctly switches at first.

- LC-C08-SoA only
- LC-Sliced_C02-AoS only
- 4 Full Searches
- Single Full Search
- Predictive Tuning

GitHub

Reference Paper