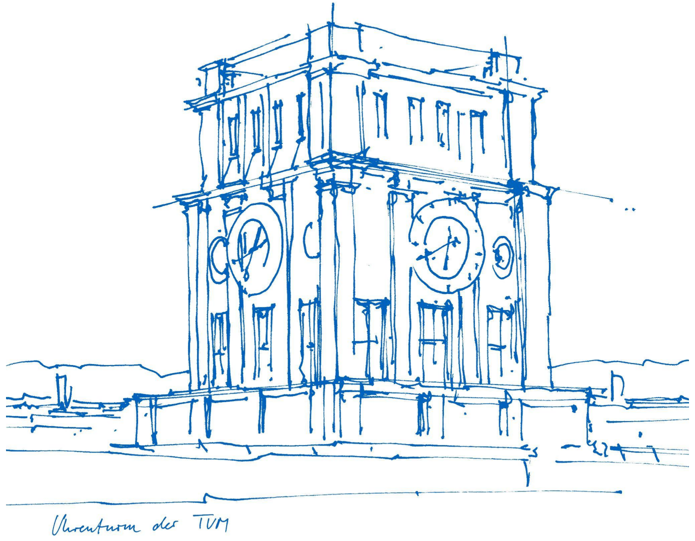
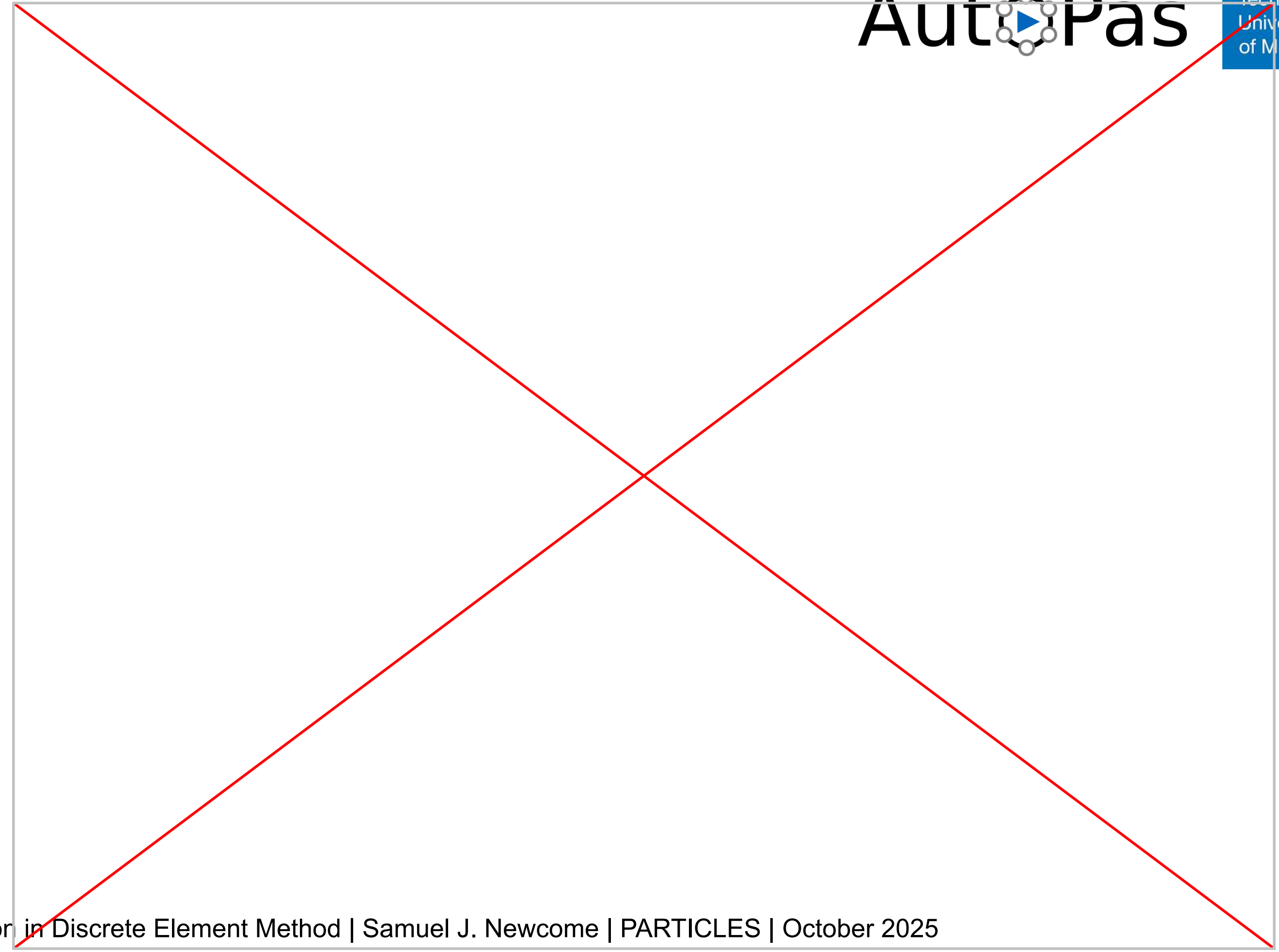
Algorithm Selection for Discrete Element Method Simulations

Samuel J Newcome, Manish K Mishra, Hans-Joachim Bungartz
Chair of Scientific Computing in Computer Science
School of Computation, Information and Technology
Technical University of Munich

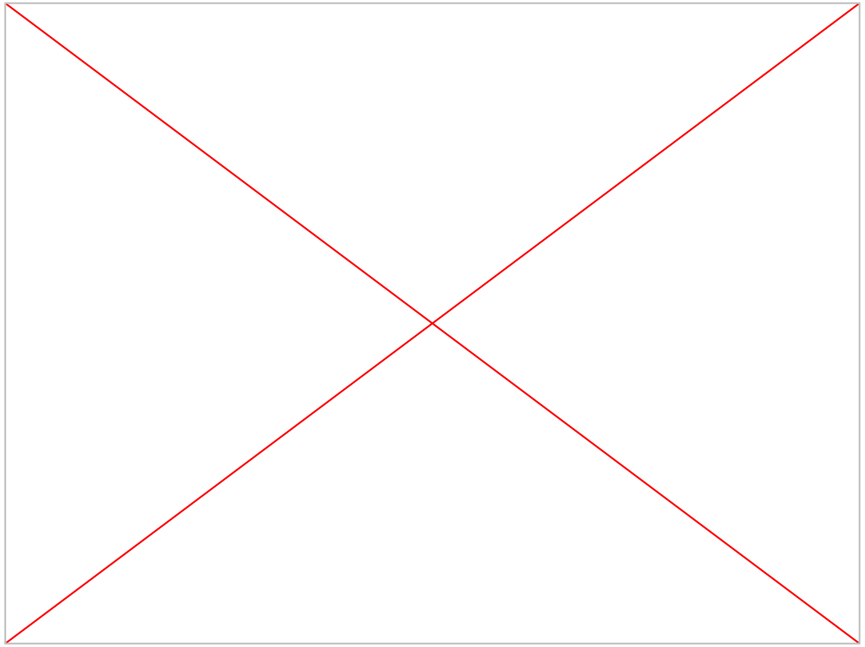
PARTICLES 2025
Barcelona, Spain
Tuesday 21st October 2025

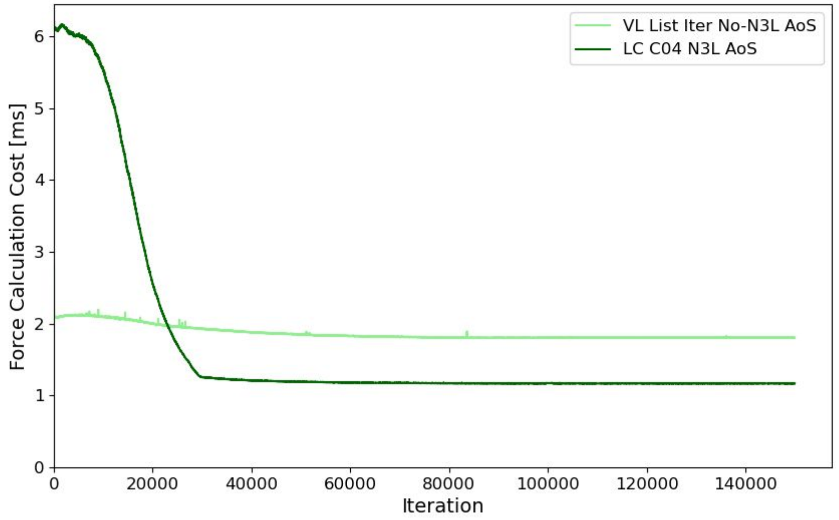




Raytracing courtesy of a ParaView tutorial from Louis Gombert

A Heated Sphere





=> Fastest algorithm can change during a simulation

Motivation

- Historically, we are a High Performance Computing for Molecular Dynamics group.
- But it was found that the fastest algorithm varies between scenarios or during a simulation.
- => So we developed a black-box particle container, AutoPas, that implements many algorithms and aims to automatically choose the best.
- AutoPas was designed for use with any kind of short-range particle simulation.
- But we still mostly only work with Molecular Dynamics.
- We want to change this!

Gratl et al., 2022, Comp. Phy. Comm.

Outline

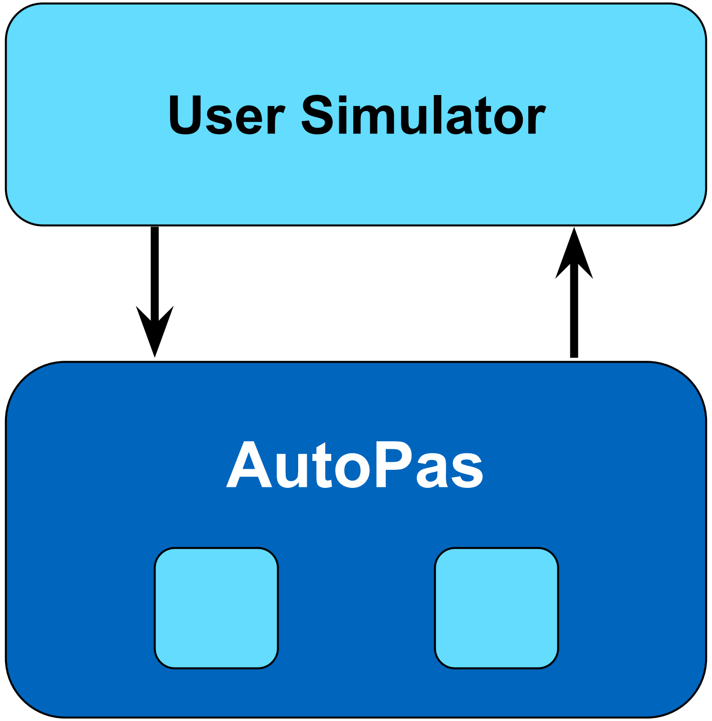
In this talk, I will

- Introduce AutoPas, some of its algorithms, and our algorithm selection.
- Show our preliminary work on using it with DEM.
- Show our preliminary performance results from our integration of DEM algorithms within AutoPas.

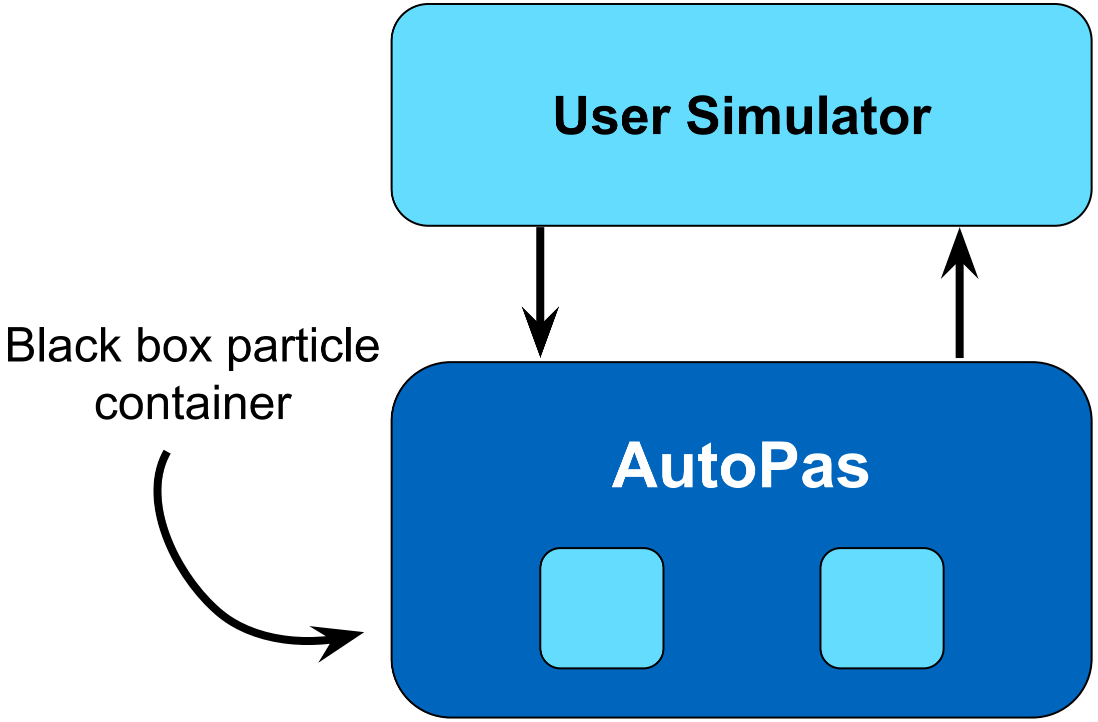
The goal of this talk is so that you can help guide our work in a way that is meaningful and beneficial to the DEM community.

Introducing AutoPas

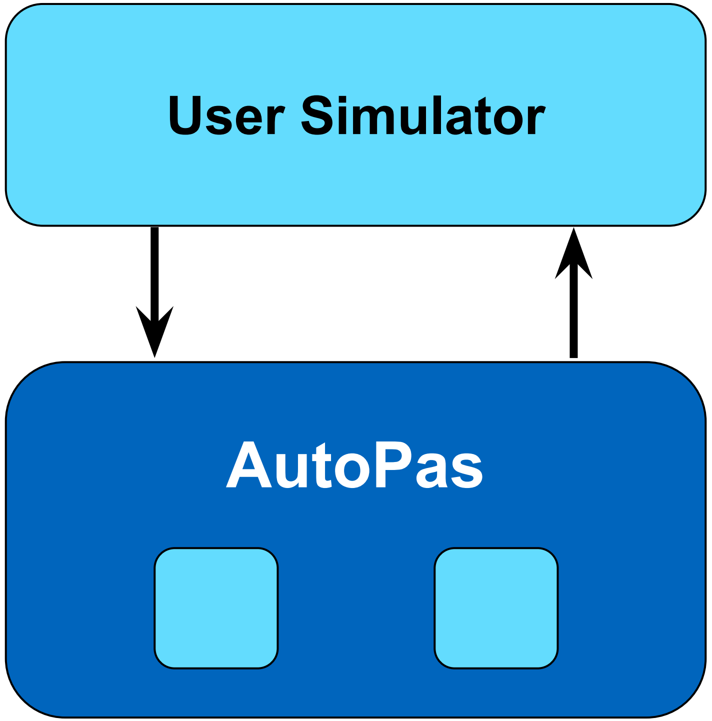
AutoPas



AutoPas



AutoPas



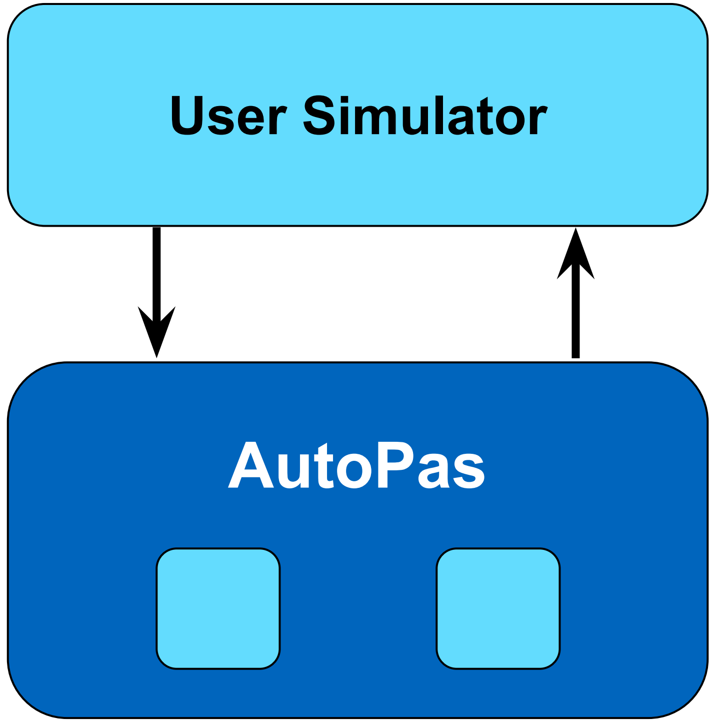
Written in C++ 20

Fully open source

Node-level

CPU-based, with GPU support WIP

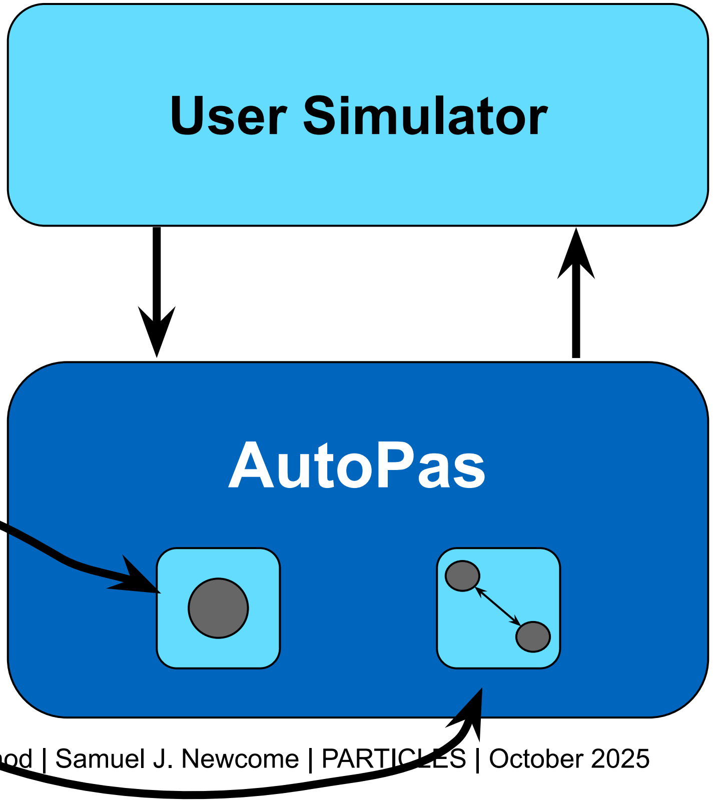
AutoPas



Black box particle container

Users build a short-range-interactions particle simulator on top of it
E.g
Molecular Dynamics, DEM, SPH

AutoPas

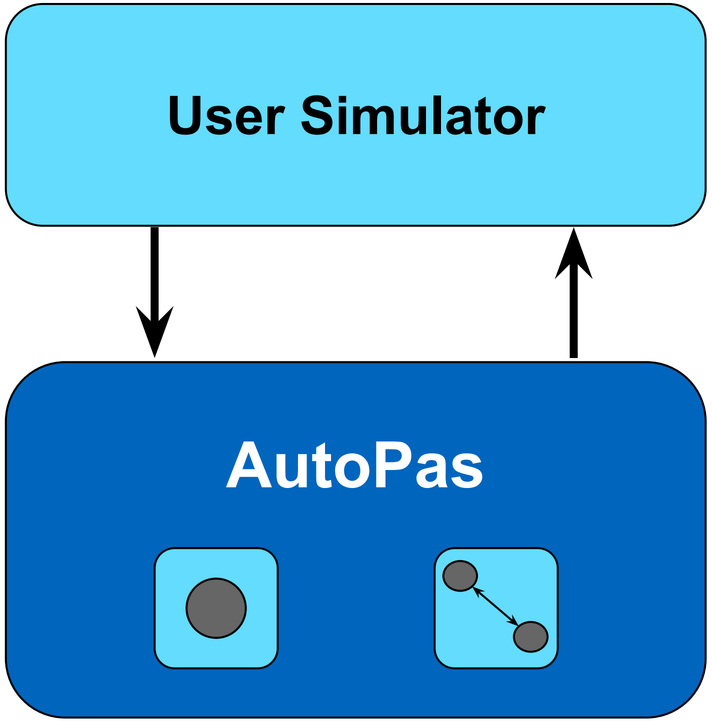


The container is templated with the user’s particle class and interaction model

AutoPas

The user simulator can then

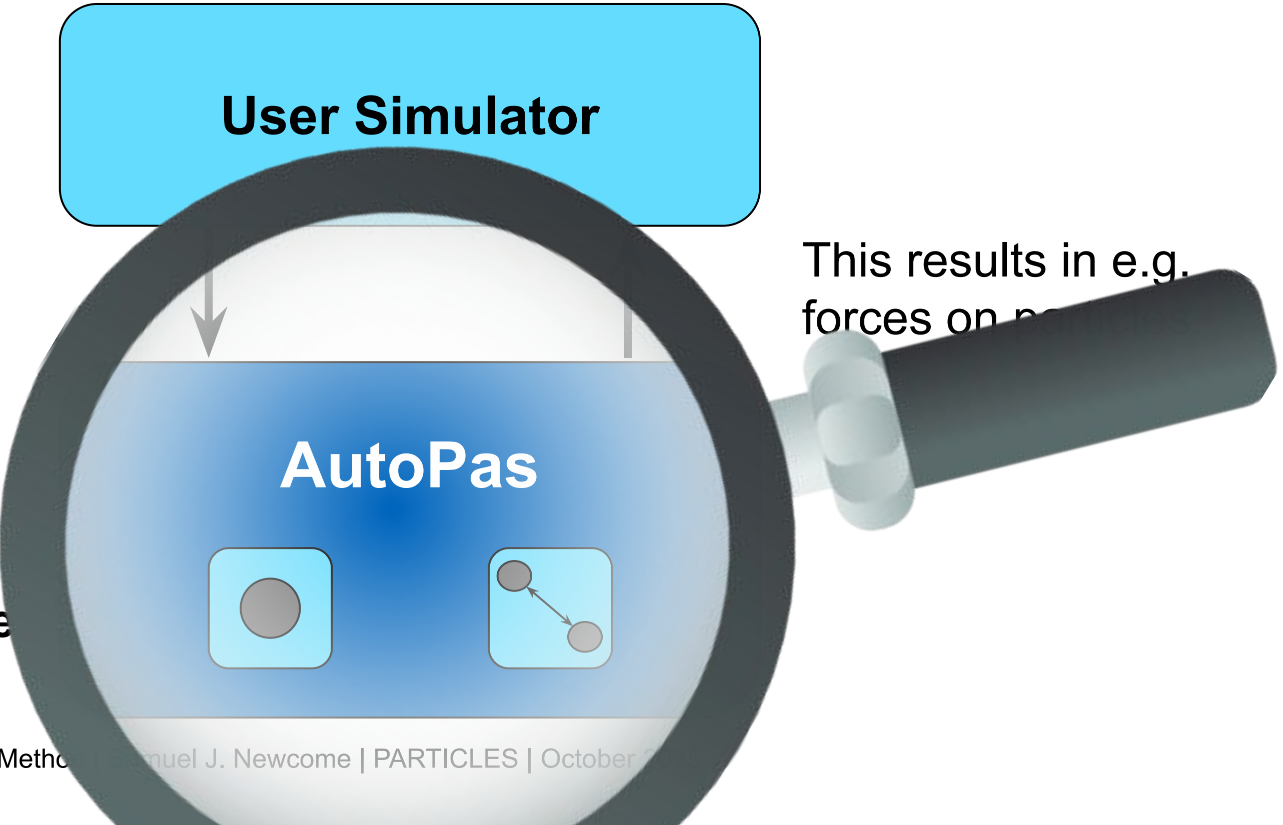
- Add particles
- Iterate over all particles (access / modification)
- Iterate over particles in a region
- **Tell AutoPas to compute all particle interactions**



This results in e.g. forces on particles.

AutoPas

The user simulator can then
- Add particles
- Iterate over all particles (access / modification)
- Iterate over particles in a region
- **Tell AutoPas to compute all particle interactions**



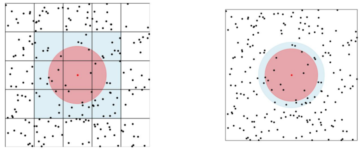
This results in e.g. forces on p

AutoPas

AutoPas

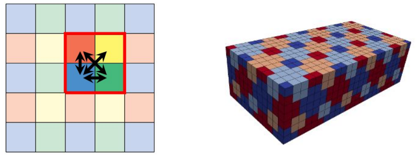
Algorithm Library

Particle Container



How we store particles
&
Neighbour Identification

Shared Memory Parallelisation



Other

- Data Layout & Vectorisation
- Cell Size

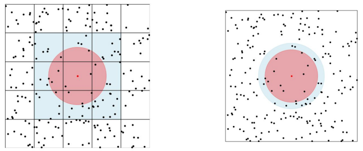
Newcome et al., 2023, J. Comp. App. Math.
Tchipev, 2020, Doct. Diss.

AutoPas

AutoPas

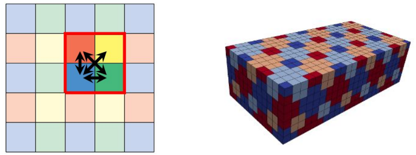
Algorithm Library

Particle Container



How we store particles
&
Neighbour Identification

Shared Memory
Parallelisation

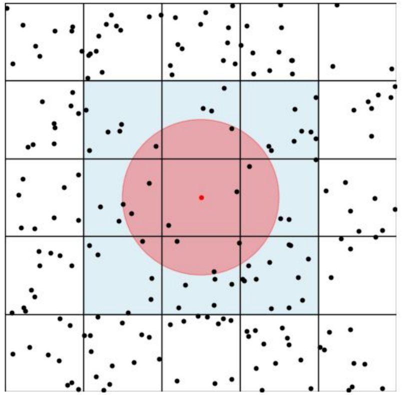


Other

- Data Layout & Vectorisation
- Cell Size

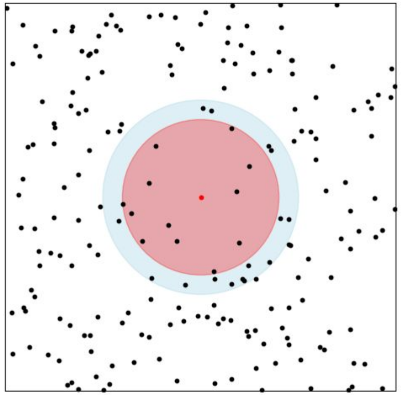
Newcome et al., 2023, J. Comp. App. Math.
Tchipev, 2020, Doct. Diss.

Particle Containers



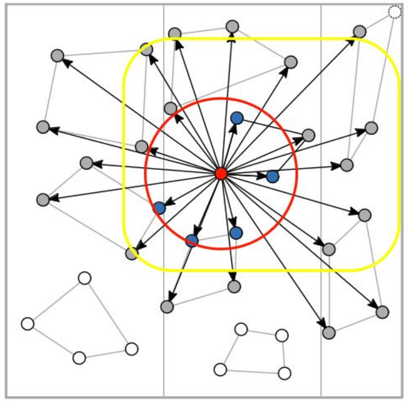
Linked Cells

+ Vectorises Well
+ Low Memory Overhead
- Many Redundant Calculations



Verlet Lists

+ Very Few Redundant Calculations
- High Rebuild Cost
- Meh Vectorisability



Verlet Cluster Lists

+ Few Redundant Calculations
+ Good Vectorisability
- High Rebuild Cost

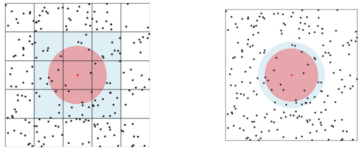
Newcome et al., 2023, J. Comp. App. Math.
Gratl et al., 2022, Comp. Phy. Comm.

AutoPas

AutoPas

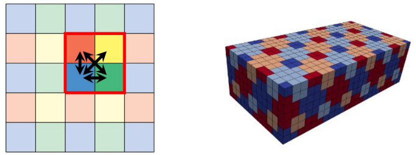
Algorithm Library

Particle Container



How we store particles
&
Neighbour Identification

Shared Memory Parallelisation

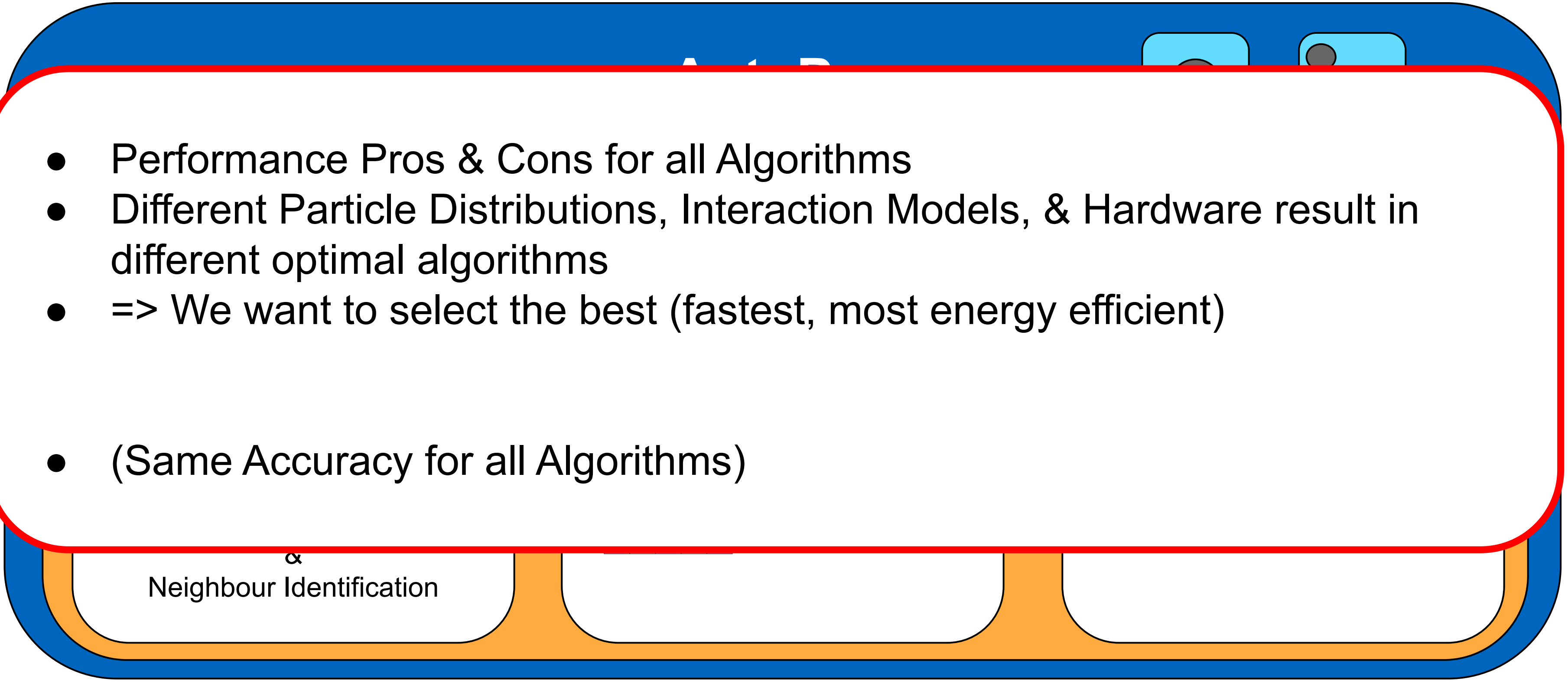


Other

- Data Layout & Vectorisation
- Cell Size

Newcome et al., 2023, J. Comp. App. Math.
Tchipev, 2020, Doct. Diss.

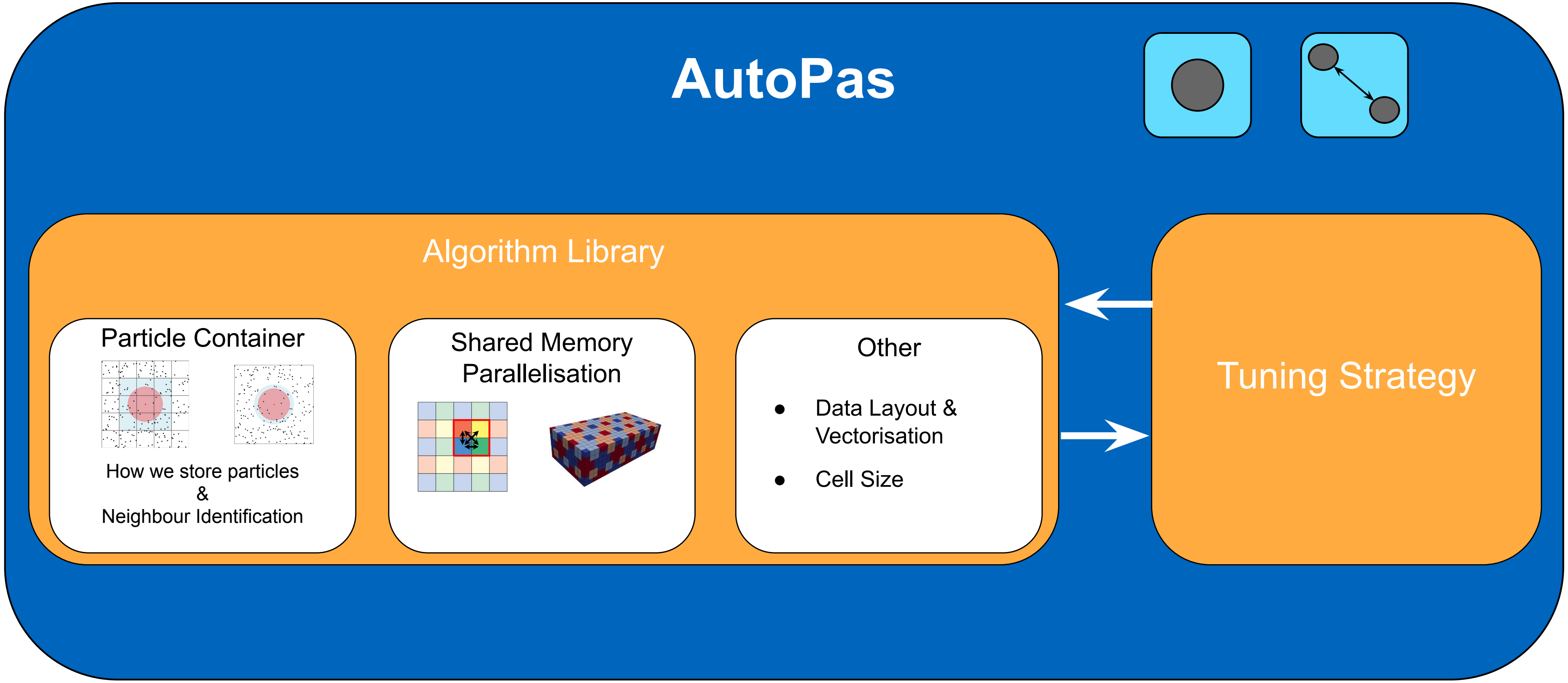
AutoPas



- Performance Pros & Cons for all Algorithms
- Different Particle Distributions, Interaction Models, & Hardware result in different optimal algorithms
- => We want to select the best (fastest, most energy efficient)
- (Same Accuracy for all Algorithms)

Newcome et al., 2023, J. Comp. App. Math.
Tchipev, 2020, Doct. Diss.

AutoPas



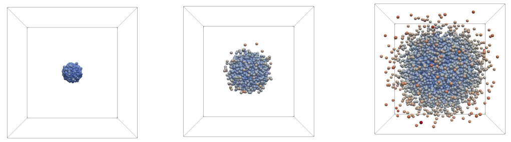
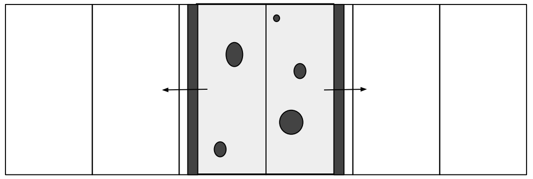
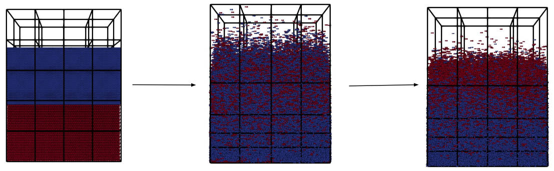
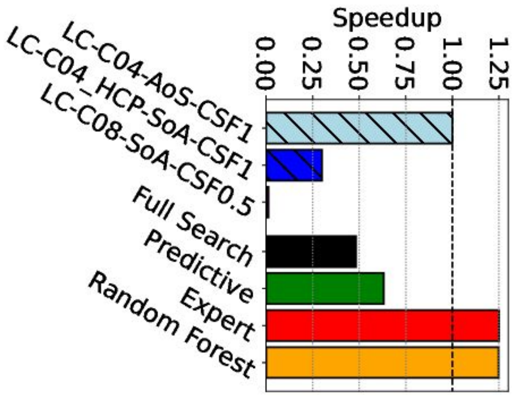
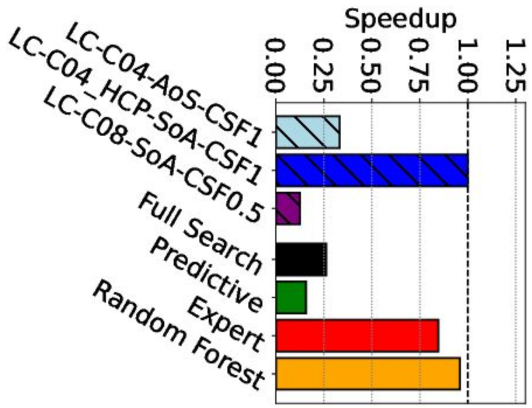
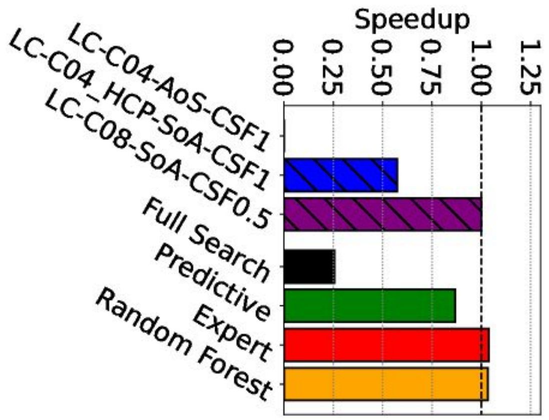
Newcome et al., 2023, J. Comp. App. Math.
Tchipev, 2020, Doct. Diss.

Random Forest Tuning Strategy

- Train a Random Forest that predicts the optimal algorithm depending on cheap-to-calculate features.
  - Mean #particles per cell
  - Std. Dev. #particles per cell
  - …
- How to generate the data?
  - Need a large, representative dataset on performance data for different scenarios.
  - Real data requires running lots of real experiments.
  - **=> “Fake” it**
  - Trial algorithms on “fake” particle distributions (e.g. randomly distributed)
  - Scenarios are physically nonsense, but computationally representative

AutoPas Technical University of Munich TUM

Algorithm Selection Results

Heating Sphere	Exploding Liquid	"Rayleigh-Taylor Instability"-Inspired / Particle Mixing
		
		

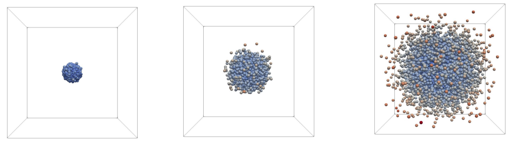
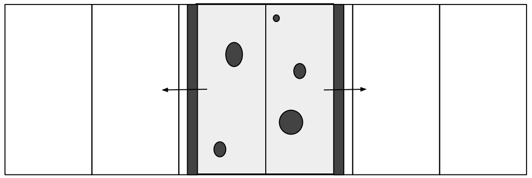
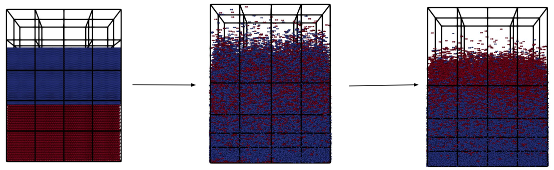
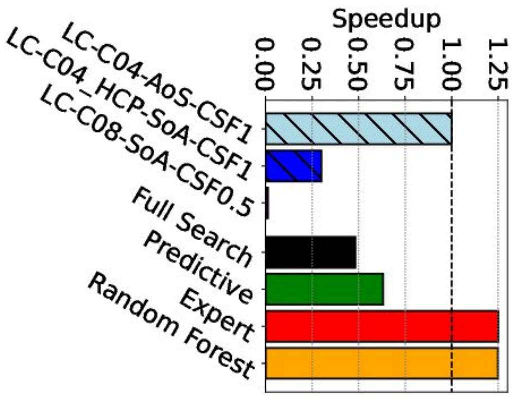
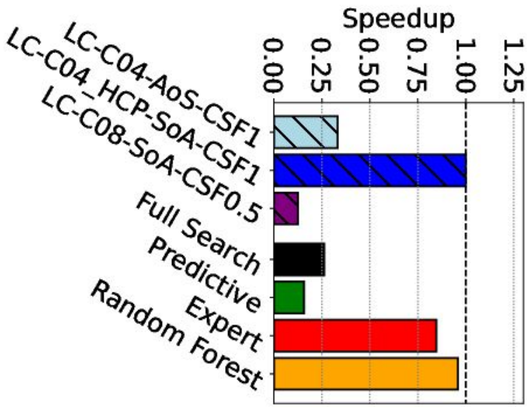
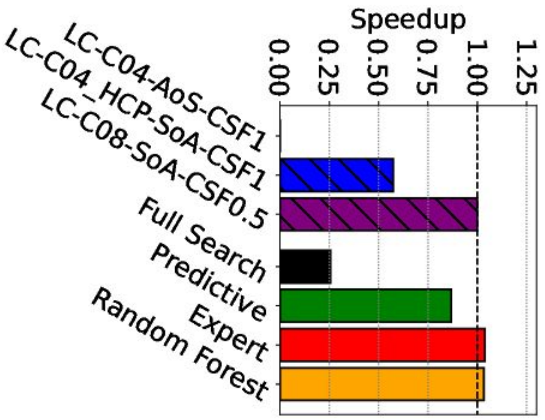
Newcome et al., 2025, ICCS

AutoPas

Technical University of Munich

Algorit

Speedup is relative to optimal single algorithm.
This is not known is advance!
Sub-1.0 (but close to 1.0) speedup is therefore a good result.
>1.0 speedup shows benefit of changing algorithm during simulation

Heat		-Taylor Instability"-Inspired / Particle Mixing
		
		

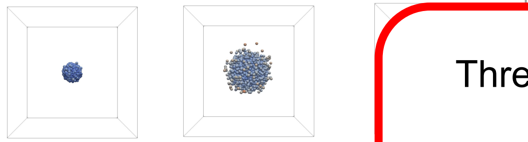
Newcome et al., 2025, ICCS

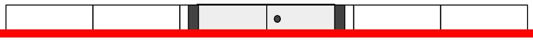
Algorit

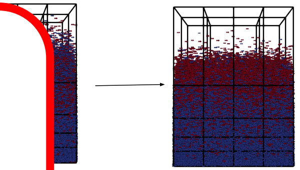
Heat

-Taylor Instability”-Inspired / Particle Mixing

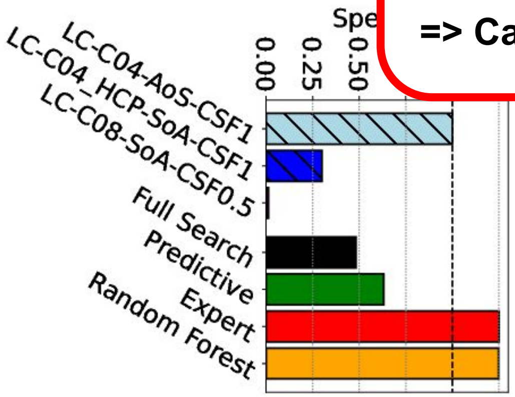
Speedup is relative to optimal single algorithm.
This is not known is advance!
Sub-1.0 (but close to 1.0) speedup is therefore a good result.
>1.0 speedup shows benefit of changing algorithm during simulation

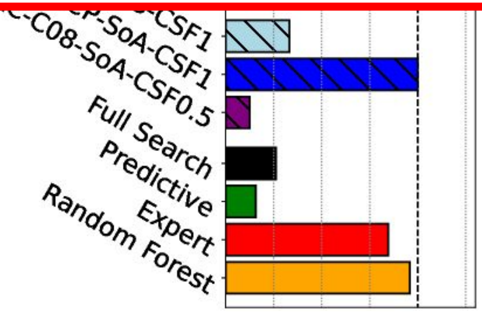


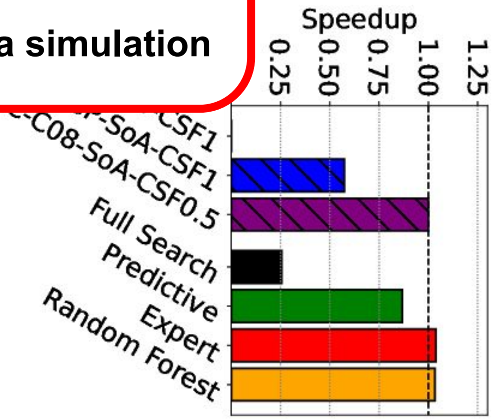




Three different simulations -> Three different optimal single algorithms
Random Forest tuning makes good selections (provided good data)
=> Can adapt to different simulations & during a simulation



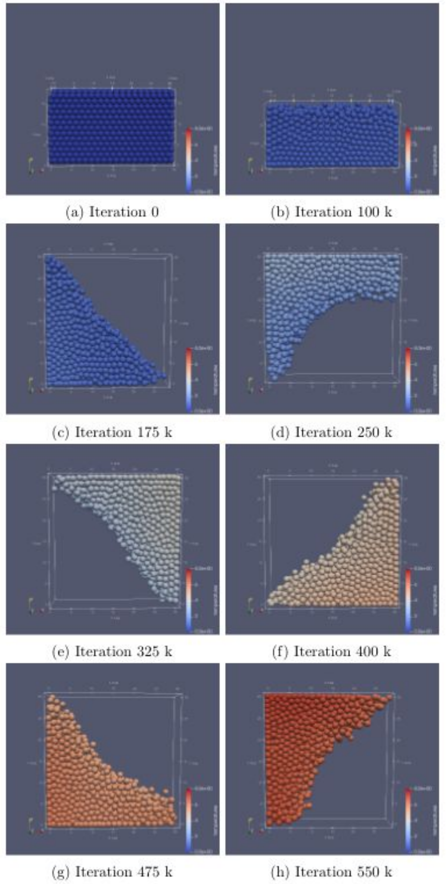




AutoPas & DEM

AutoPas & DEM

- Example DEM Simulator Created
  - Linear Spring Contact Model with Dampening
  - Sliding Frictional Forces + Torques
  - Rolling Resistance Torque
  - Torision Resistance Torque
  - Background Friction
  - Multi-spherical Particles
- **=> We still find optimal algorithm varies between experiments**
- Still missing:
  - Tangential Spring
  - Vectorisation

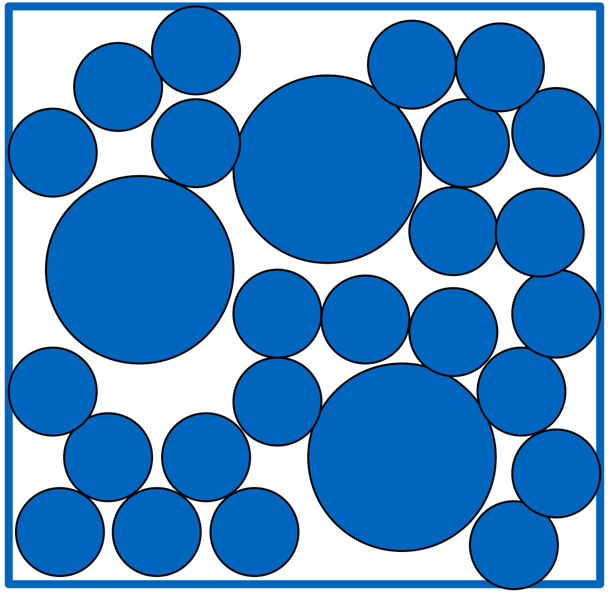


Kim, 2025, BSc. Thesis

AutoPas & Hierarchical Grids

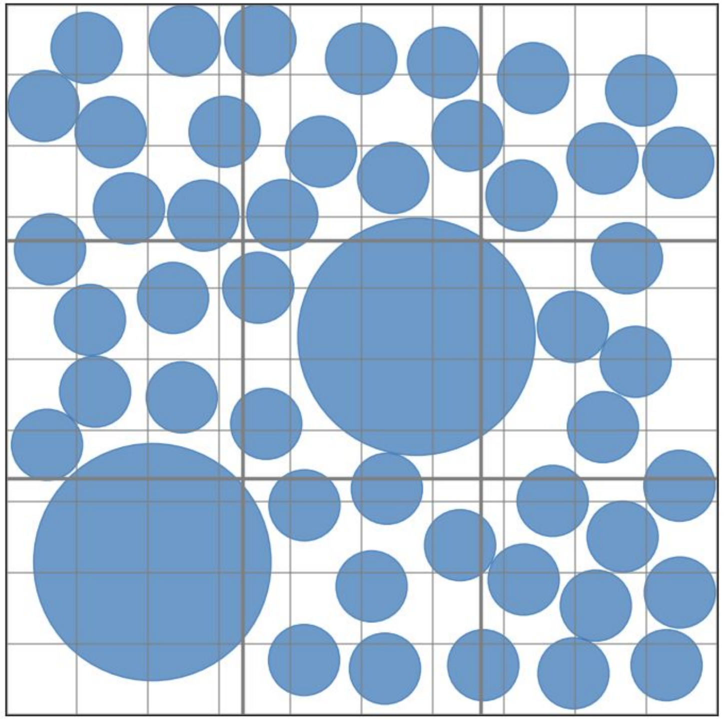
AutoPas & Hierarchical Grids

- Can have different sized particles in DEM
- All existing particle containers assume same size
- => A lot of redundant calculations involved with the smaller particles



AutoPas & Hierarchical Grids

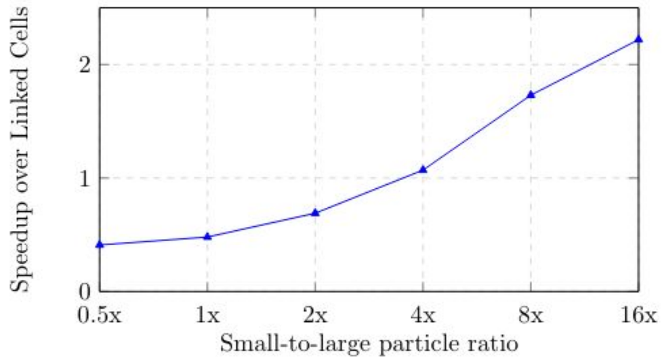
- Can have different sized particles in DEM
- All existing particle containers assume same size
- => A lot of redundant calculations involved with the smaller particles
- We implemented the Hierarchical Grid method of V. Ogarko & S. Luding (& OpenMP parallelisation)



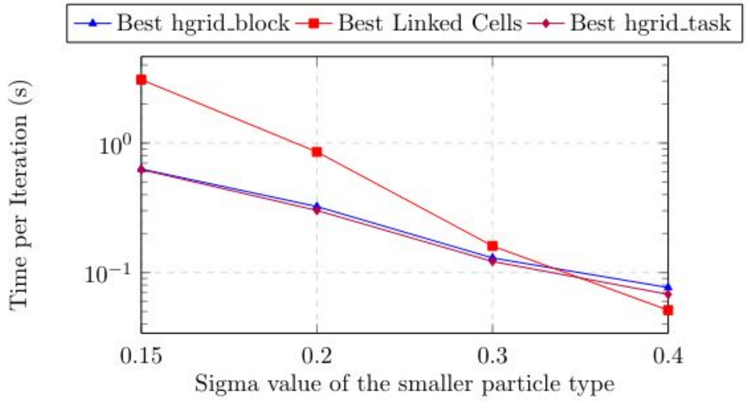
AutoPas & Hierarchical Grids: Preliminary Results

We performed some initial MD experiments with LJ potential and a scalable cutoff (=> recreates different sized particles)

Mixing different ratios of full-sized and half-sized particles



Keeping ratio of at 2x but changing size of small particle

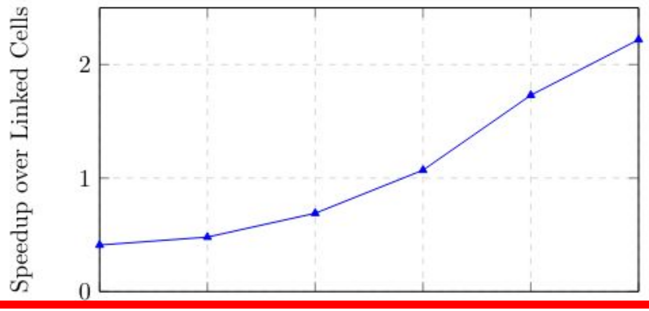


Iyidogan, 2025, MSc. Thesis

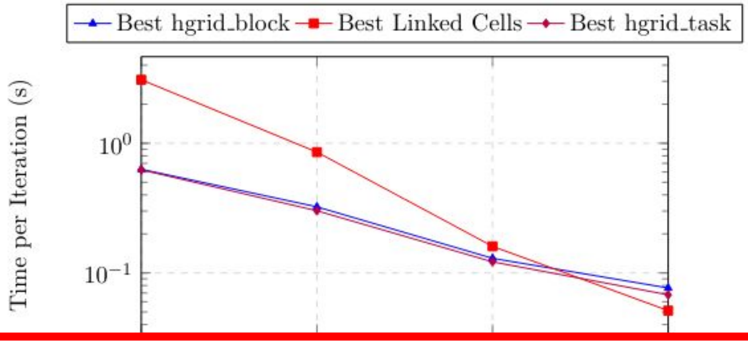
AutoPas & Hierarchical Grids: Preliminary Results

We performed some initial MD experiments with LJ potential and a scalable cutoff (=> recreates different sized particles)

Mixing different ratios of full-sized and half-sized particles



Keeping ratio of at 2x but changing size of small particle



Hierarchical Grids outperform other AutoPas methods in the scenarios it was designed for.

Acknowledgements

Recent work on DEM and Hierarchical Grids comes from Joon Kim and Atacan Iyidogan

Video was made prettier with guidance from Louis Gombert

Work was financially supported by, as well as computational resources (HSUper) from, the project hpc.bw, which is funded by dtec.bw - Digitalisation and Technology Research Centre of the Bundeswehr. dtec.bw is funded by the European Union — NextGenerationEU.

We thank the Leibniz Rechnen Zentrum for other computational resources (CoolMUC)

Big thanks to Markus Mühlhäußer & Jonas Schumacher, and all other developers of AutoPas.

dtec.bw
Zentrum für Digitalisierungs- und Technologieforschung der Bundeswehr

Funded by the European Union
NextGenerationEU

Summary & Outlook

- Introduced the algorithm selection particle simulation library, AutoPas, now with a basic DEM testbed.
- Showed preliminary results of Hierarchical Grid container in AutoPas

Outlook:

- Further experimentation with Hierarchical Grids.
- Let AutoPas tune HGrid metrics (cell sizes), parallelisation schemes on different levels.
- Expand data-driven algorithm selection with particle size statistics.

Get the slides

samuel.newcome@tum.de

References

[1] Newcome, S.J., Gratl, F.A., Lerchner, M., Pazar, A., Mishra, M.K., Bungartz, HJ. (2025). Algorithm Selection in Short-Range Molecular Dynamics Simulations. In: Lees, M.H., *et al.* Computational Science – ICCS 2025. ICCS 2025. Lecture Notes in Computer Science, vol 15906. Springer, Cham. https://doi.org/10.1007/978-3-031-97635-3_35

[2] Gratl, F. A., Seckler, S., Bungartz, H. J., & Neumann, P. (2022). N ways to simulate short-range particle systems: Automated algorithm selection with the node-level library AutoPas. *Computer Physics Communications*, *273*, 108262. https://doi.org/10.1016/j.cpc.2021.108262

[3] Tchipev, N. P. (2020). Algorithmic and implementational optimizations of molecular dynamics simulations for process engineering, Doctoral Thesis, TU Munich. https://mediatum.ub.tum.de/doc/1524715

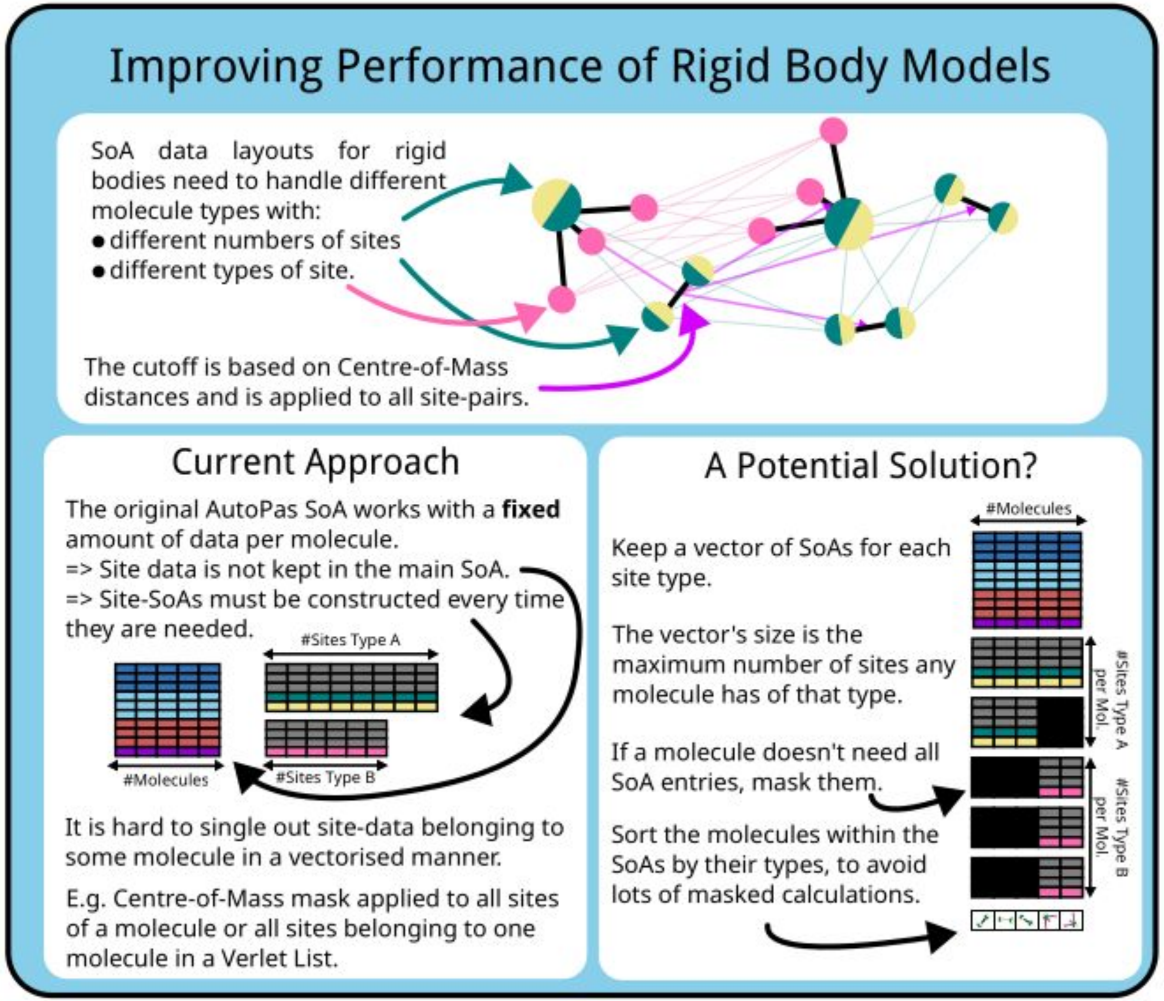
[4] Kim, J. (2025). Exploring the Discrete Element Method: Simulation of Granular Particles using AutoPas, Bachelor's Thesis, TU Munich. https://mediatum.ub.tum.de/node?id=1773224

[5] Iyidogan, A. (2025). Parallelization and Implementation of Hierarchical Grid Method for Contact Detection in AutoPas. https://mediatum.ub.tum.de/node?id=1782419

Backup Slides

S

Technical University of Munich



Improving Performance of Rigid Body Models

SoA data layouts for rigid bodies need to handle different molecule types with:

- different numbers of sites
- different types of site.

The cutoff is based on Centre-of-Mass distances and is applied to all site-pairs.

Current Approach

The original AutoPas SoA works with a **fixed** amount of data per molecule.
=> Site data is not kept in the main SoA.
=> Site-SoAs must be constructed every time they are needed.

It is hard to single out site-data belonging to some molecule in a vectorised manner.

E.g. Centre-of-Mass mask applied to all sites of a molecule or all sites belonging to one molecule in a Verlet List.

A Potential Solution?

Keep a vector of SoAs for each site type.

The vector's size is the maximum number of sites any molecule has of that type.

If a molecule doesn't need all SoA entries, mask them.

Sort the molecules within the SoAs by their types, to avoid lots of masked calculations.