WWU
MÜNSTER

Fearless simulation of many-particle dynamics

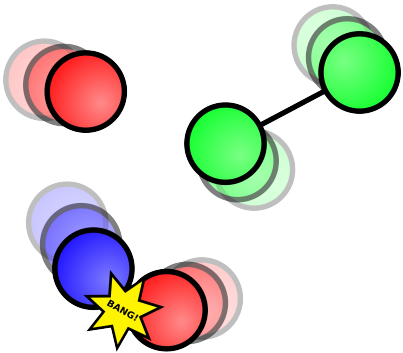
Scientific Computing in Rust 2023

Julian Jeggle*
Raphael Wittkowski

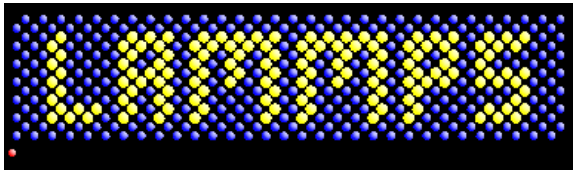
2023-03-20

Many-particle dynamics

- Particle dynamics: **time evolution** of systems of **discrete, interacting entities**
- “Particles”: atoms, molecules, colloids, bacteria, sand, animals, asteroids, ...
- Mathematical: systems of **coupled differential equations** (ODEs/SDEs)
- Challenge: **extremely high number of DoF** for realistic systems



Prior art (molecular dynamics)



FAST. FLEXIBLE. FREE.
GROMACS

Problems with classical MD software

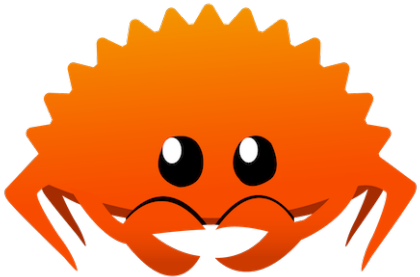
- Only composition of pre-defined force-fields, integrators, particle types
  $\Rightarrow$ **Lack of generality**
- Implementing new primitives requires C/C++/FORTRAN
  $\Rightarrow$ **Brittle extensibility**
- Parallelism usually based on message passing (i.e. MPI)
  $\Rightarrow$ Still the right model for **modern hardware**?

“Modern MD”

- Popular examples: OpenMM, OpenFPM, FDPS, HOOMD-blue
- There is still room for improvement!
- Design goals:
  - **Maximize generality**
  - **Abstract** away all the “technicalities” of implementing MD
    - But also: **escape hatch** if you know what you are doing
  - **Safe shared-memory** parallelism

“Modern MD”

- Popular examples: OpenMM, OpenFPM, FDPS, HOOMD-blue
- There is still room for improvement!
- Design goals:
  - **Maximize generality**
  - **Abstract** away all the “technicalities” of implementing MD
    - But also: **escape hatch** if you know what you are doing
  - **Safe shared-memory** parallelism



What even is a particle?

- **Particle** = bundle of physical quantities
  - Mandatory: **position**
- **Particle system** = set of particles that might interact
- **Particle dynamics** = rules for transforming one particle system (state) into another

What even is a particle?

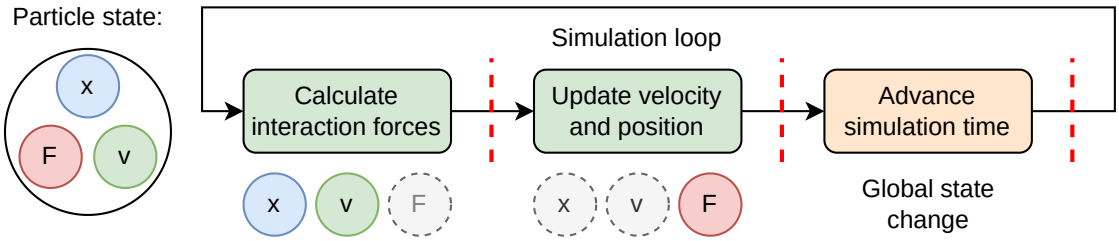
- **Particle** = bundle of physical quantities
  - Mandatory: **position**
- **Particle system** = set of particles that might interact
- **Particle dynamics** = rules for transforming one particle system (state) into another
- Way too generic for static analysis ($\Rightarrow$ **safe** parallelism)!

Fearlessly parallel particle dynamics

- In principle: parallel time evolution **for each particle**
- Rules of borrowing:
  - Either an **arbitrary amount** of **read-only** references …
  - or a **single mutable** reference
- For particles: mutable quantities must be “invisible” to other particles
- Divide simulation into **stages** separated by **synchronization barriers**
- Example: Ideal gas (all quantities invisible)

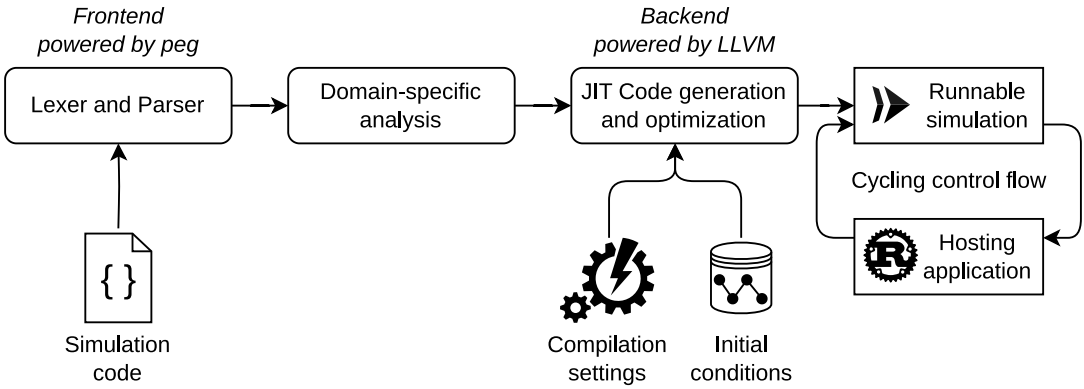
Fearlessly parallel particle dynamics

- Another example: Force-like interaction (Euler integration)



Implementation (in Rust 🦀)

Fearlessly **I**ntegrating **P**article **S**imulator (**FIPS**)



Thank you for your attention!

- Preprint for particle model: J. Jeggle, R. Wittkowski, Generic framework for data-race-free many-particle simulations on shared memory hardware (2023), arXiv:2302.14170
- FIPS: github.com/jjegg01/fips (10.5281/zenodo.7689625)
- fips-md.zulipchat.com
- Funding by DFG as part of SFB 1459 - Intelligent Matter

CRC 1459 INTELLIGENT MATTER

www.uni-muenster.de/SFB1459