High Performance Computing for Science and Engineering I

Tutorial 6: Cell-lists

Lucas Amoudruz
Computational Science & Engineering Laboratory

OUTLINE

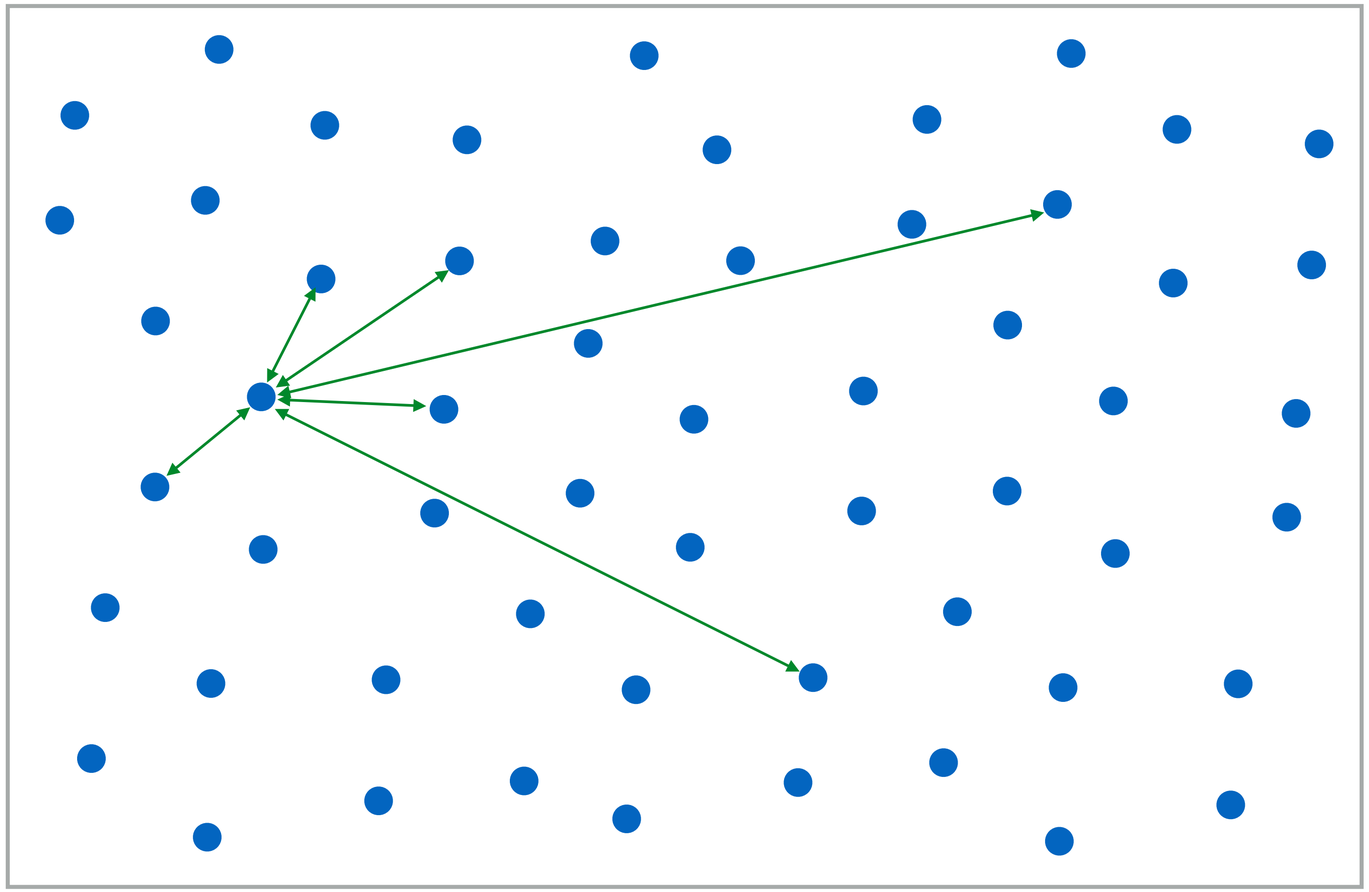
- The N-Body problem
- Lennard-Jones potential
- Cell-lists
- One possible implementation

N-Body problem

Consider N particles interacting with each other

Examples:

- Gravitation
- Electrodynamics
- Particle Strength Exchange
- Molecular Dynamics: Lennard-Jones
- Smooth Particle Hydrodynamics
- Dissipative Particle Dynamics

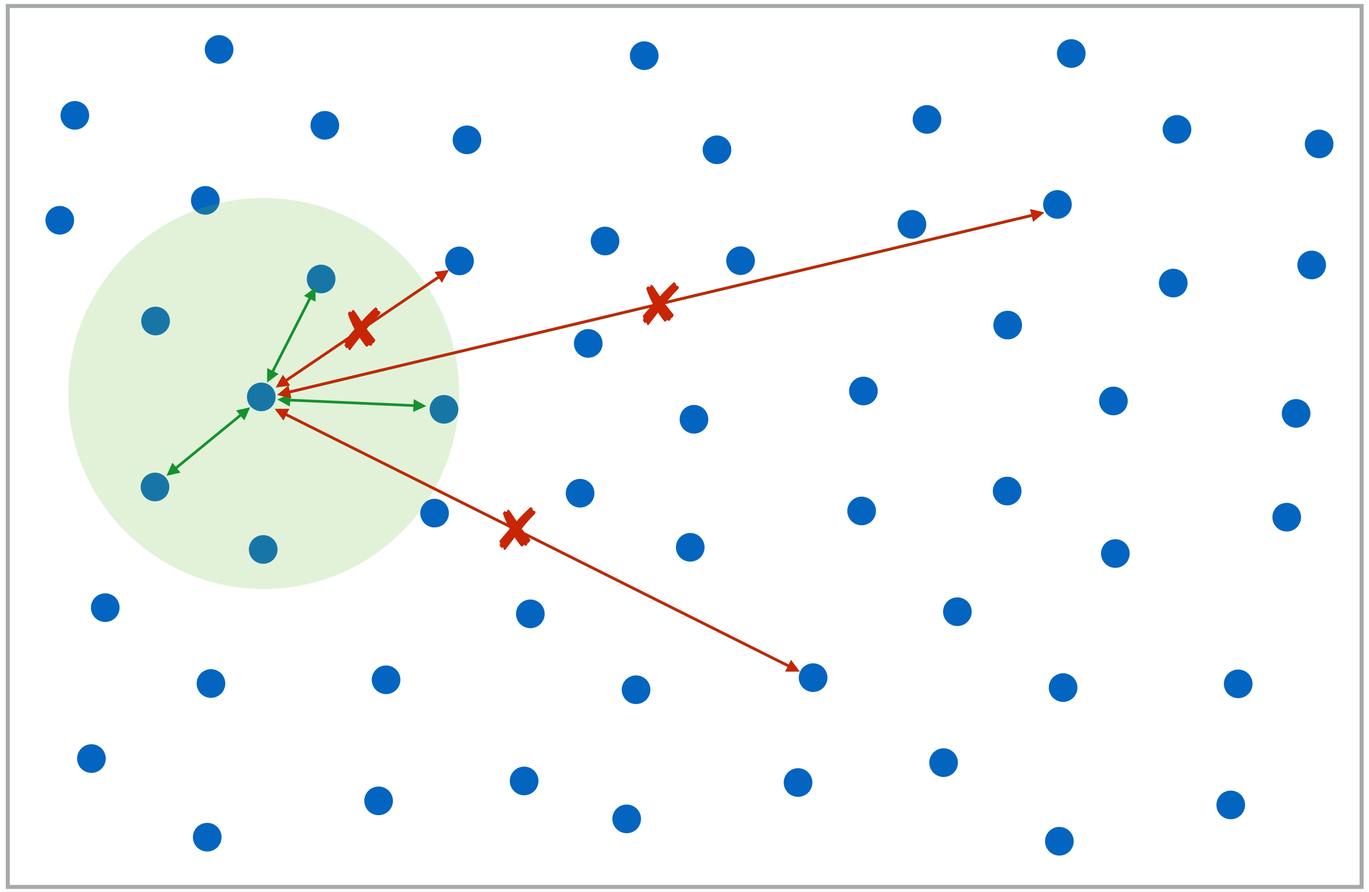


N-Body problem: complexity

Consider N particles interacting with each other

Complexity:
In principle, each particle interacts with all other particles
-> $O(N^2)$

Consider short range interactions
We can discard many interactions
-> $O(Nd)$
d is the number of neighbors per particle



Short range interaction: L-J potential

The Lennard-Jones potential describes simplified interaction between neutral atoms/molecules:

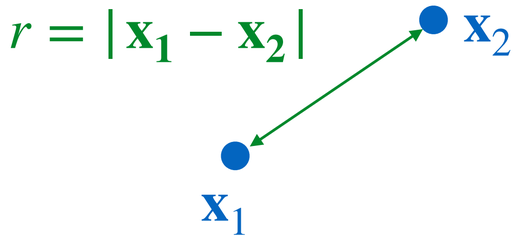
$$V(r) = 4\varepsilon\left(\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right)$$

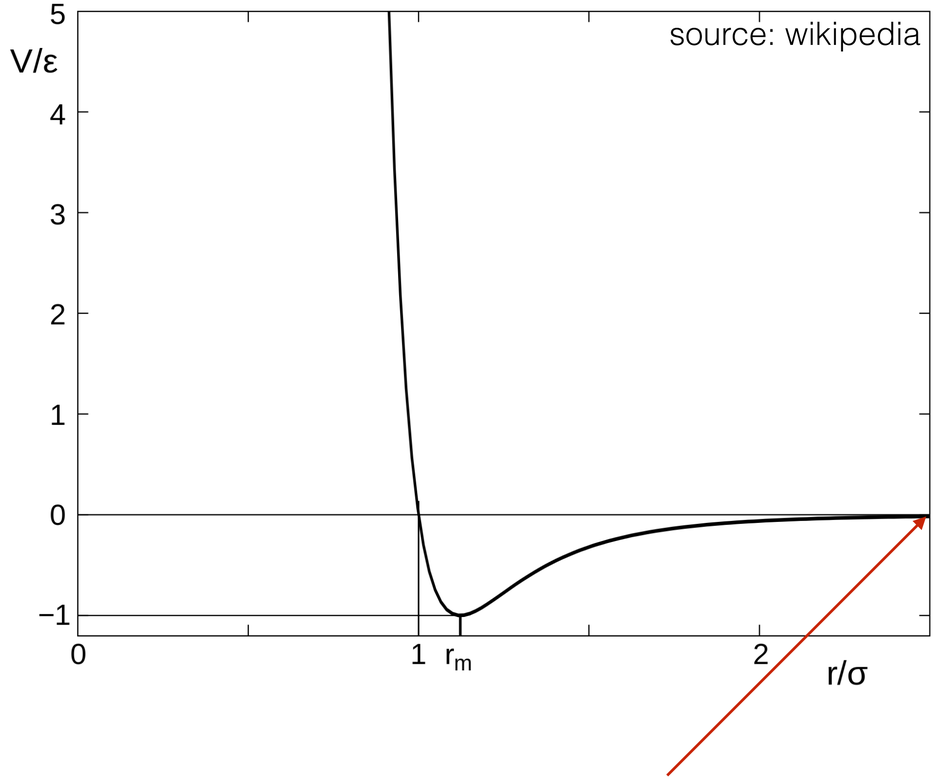
$$\mathbf{F}(\mathbf{r}) = -\nabla_{\mathbf{x}_1} V(r)$$

$$= \frac{\mathbf{r}}{r^2} 24\varepsilon\left(2\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right)$$

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{v}_i$$

$$\frac{d\mathbf{v}_i}{dt} = \frac{1}{m}\mathbf{F}_i = \frac{1}{m}\sum_{j\neq i}\mathbf{F}\left(\mathbf{r}_{ij}\right)$$





potential and forces almost zero at

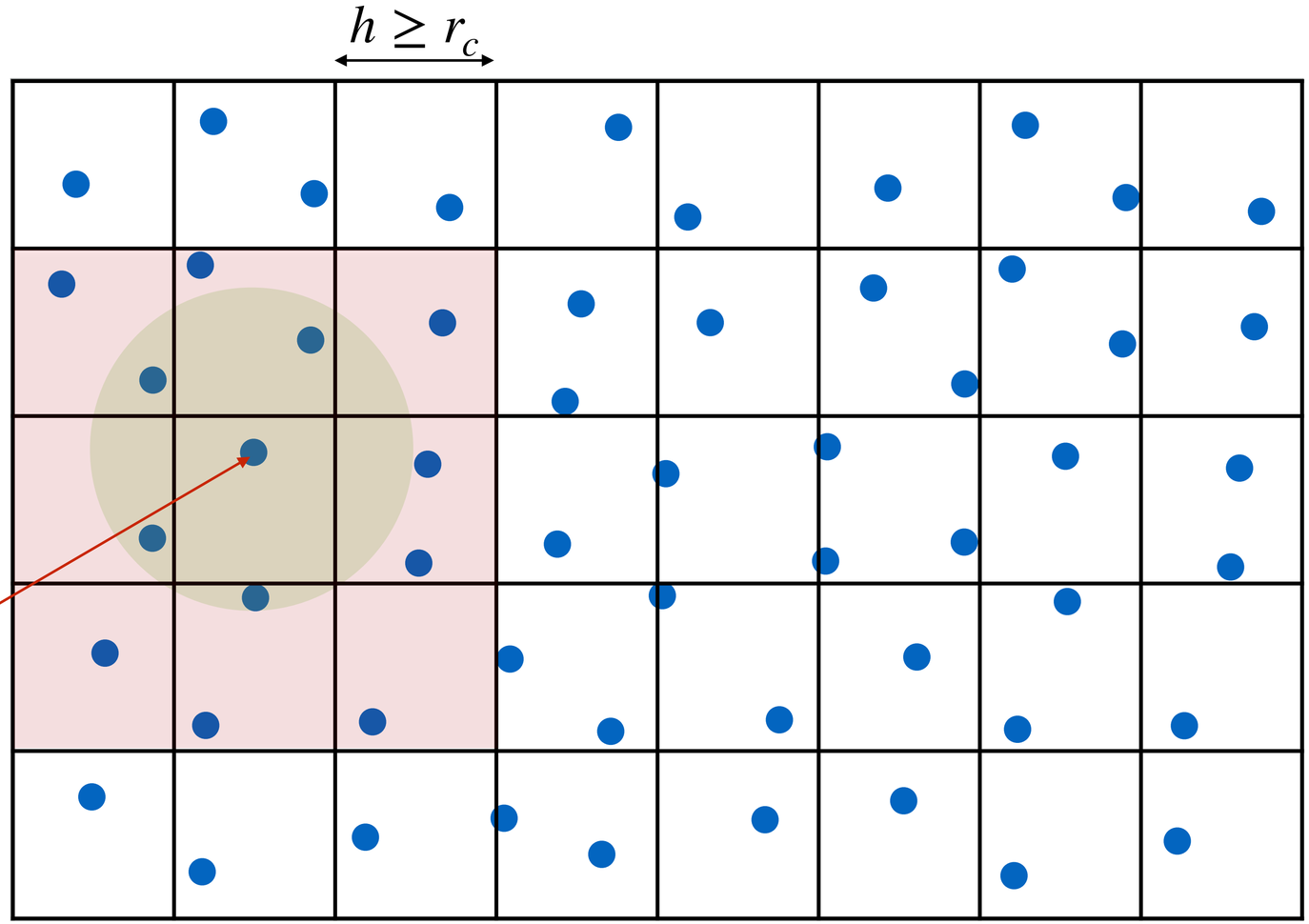
$r_c = 2.5\sigma$

N-Body problem with short range: cell-lists

Consider N particles interacting with each other within cutoff radius

How can I find my neighbors?

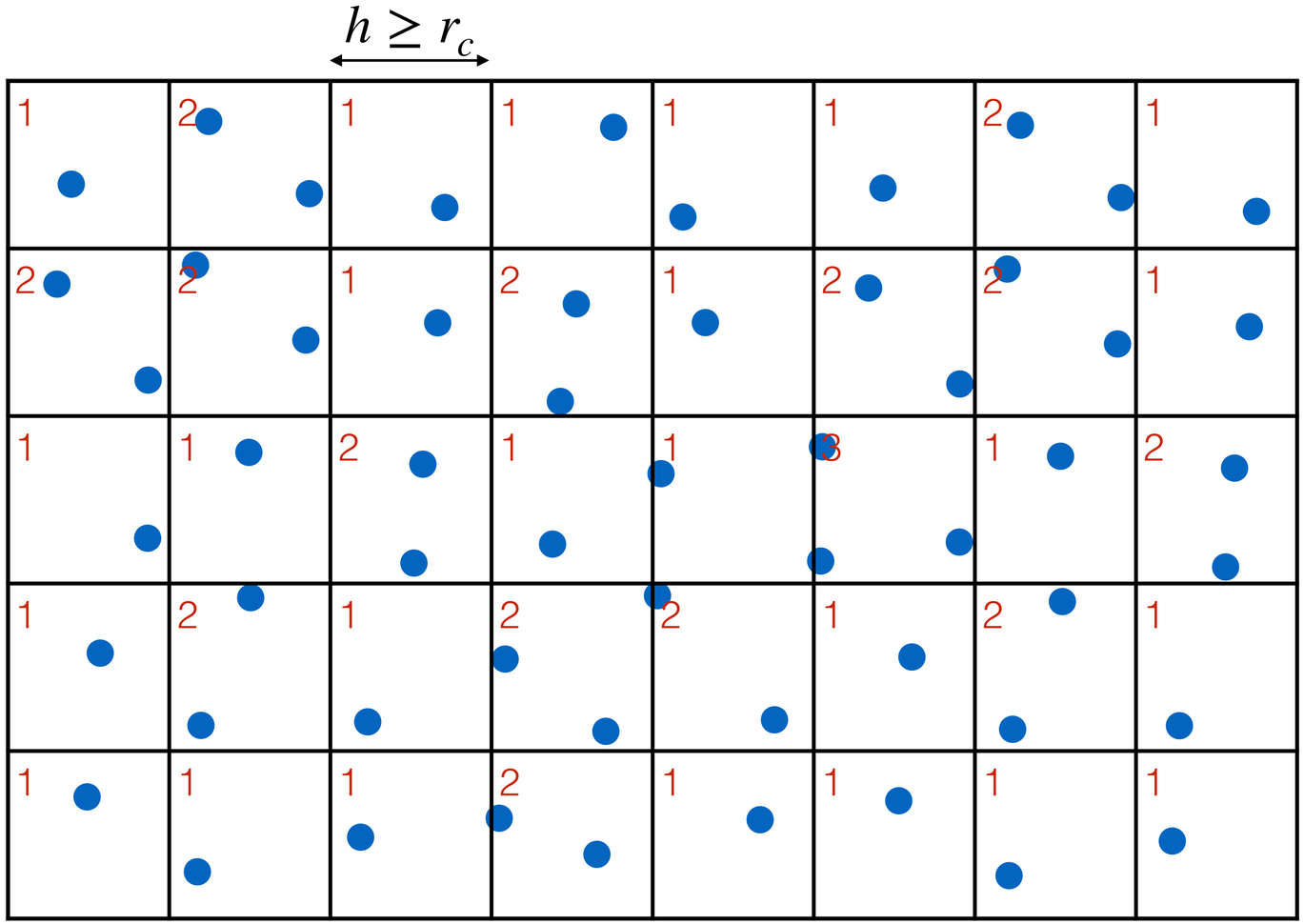
We need a map from space to particles: the cell-lists



Neighbors particles are in the 3x3 cells around myself

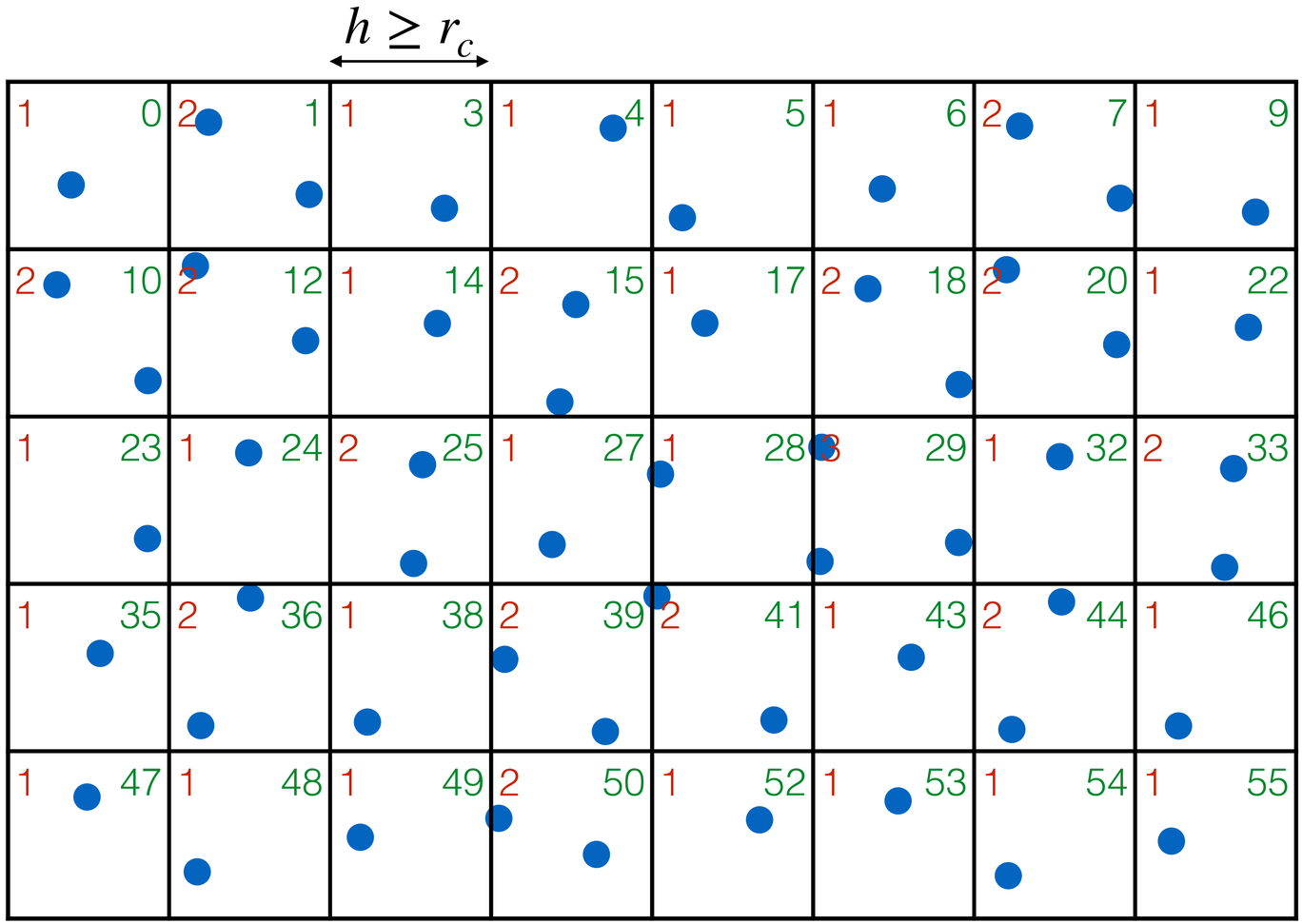
Building the cell-lists

1. Count number of particles in each cell



Building the cell-lists

1. Count number of particles in each cell
2. Compute the starts of each cells (prefix sum)

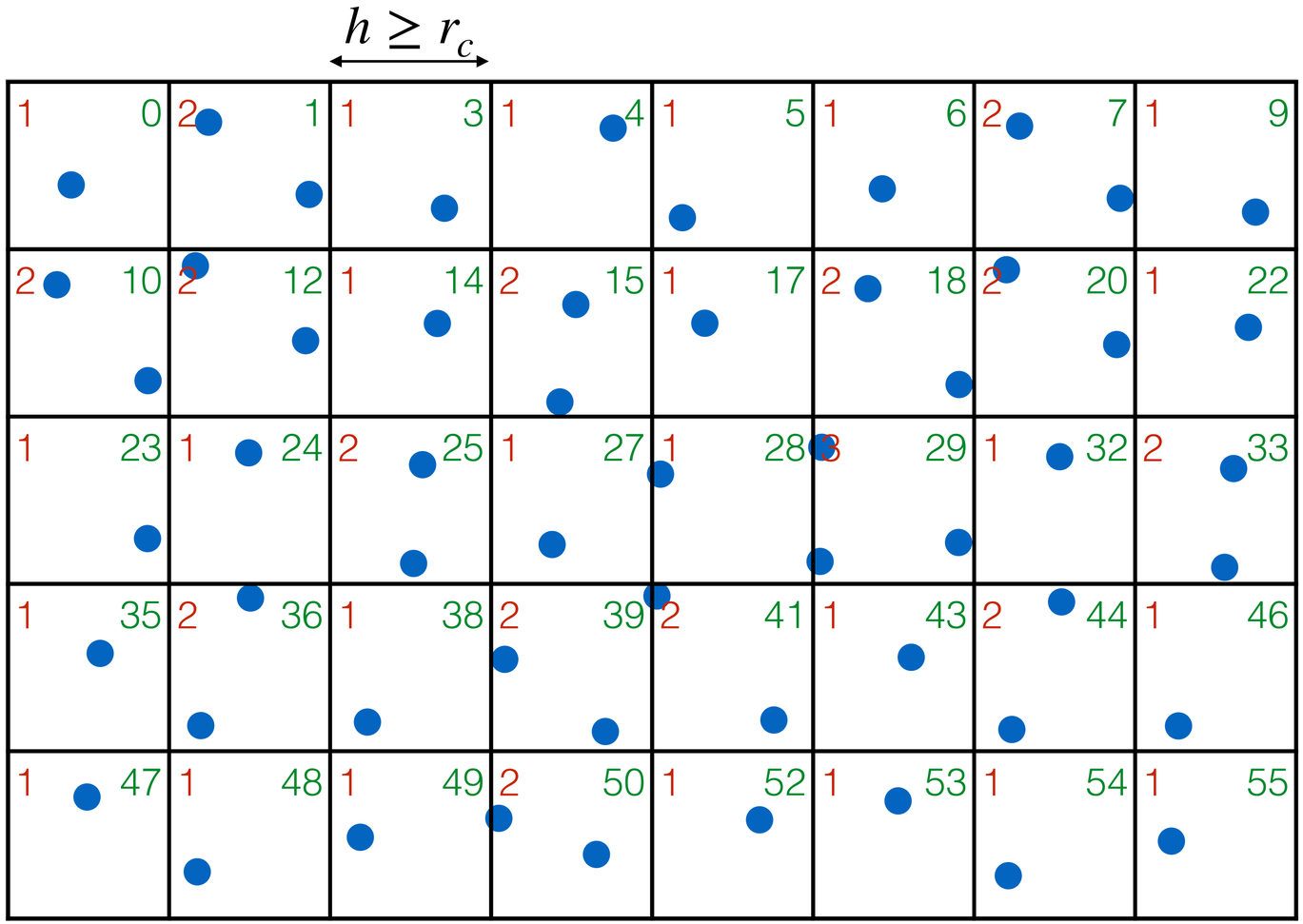


Building the cell-lists

1. Count number of particles in each cell
2. Compute the starts of each cells (prefix sum)
3. Reorder the particles in memory

The cell lists are just 2 arrays: starts and counts

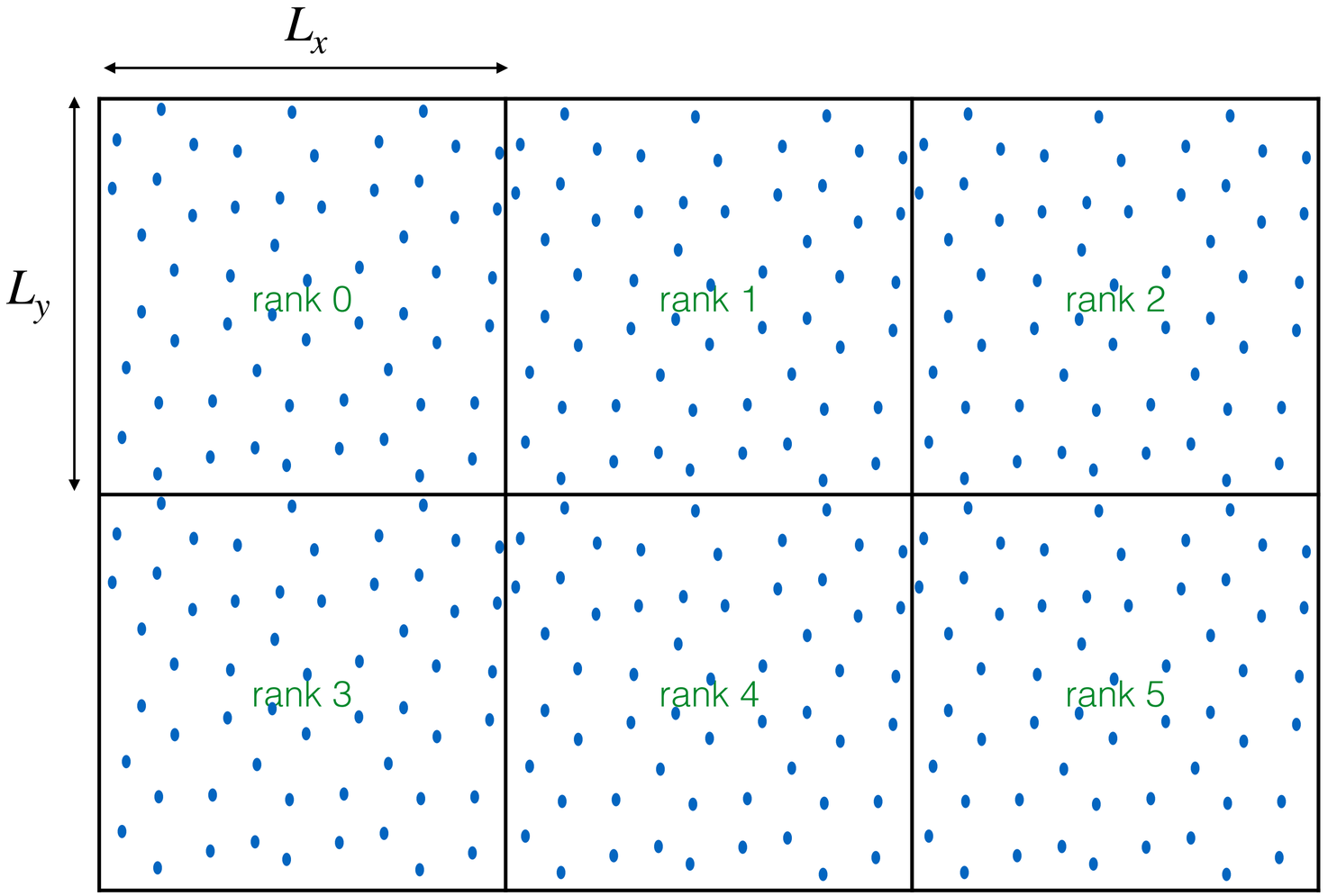
Particles in cell with id **`cid`** have indices between **`starts[cid]`** and **`starts[cid]+counts[cid]`**



A simple 2D LJ simulation with MPI

1. periodic position
2. Exchange with 8 neighbors
3. each sub-domain has a local frame of reference

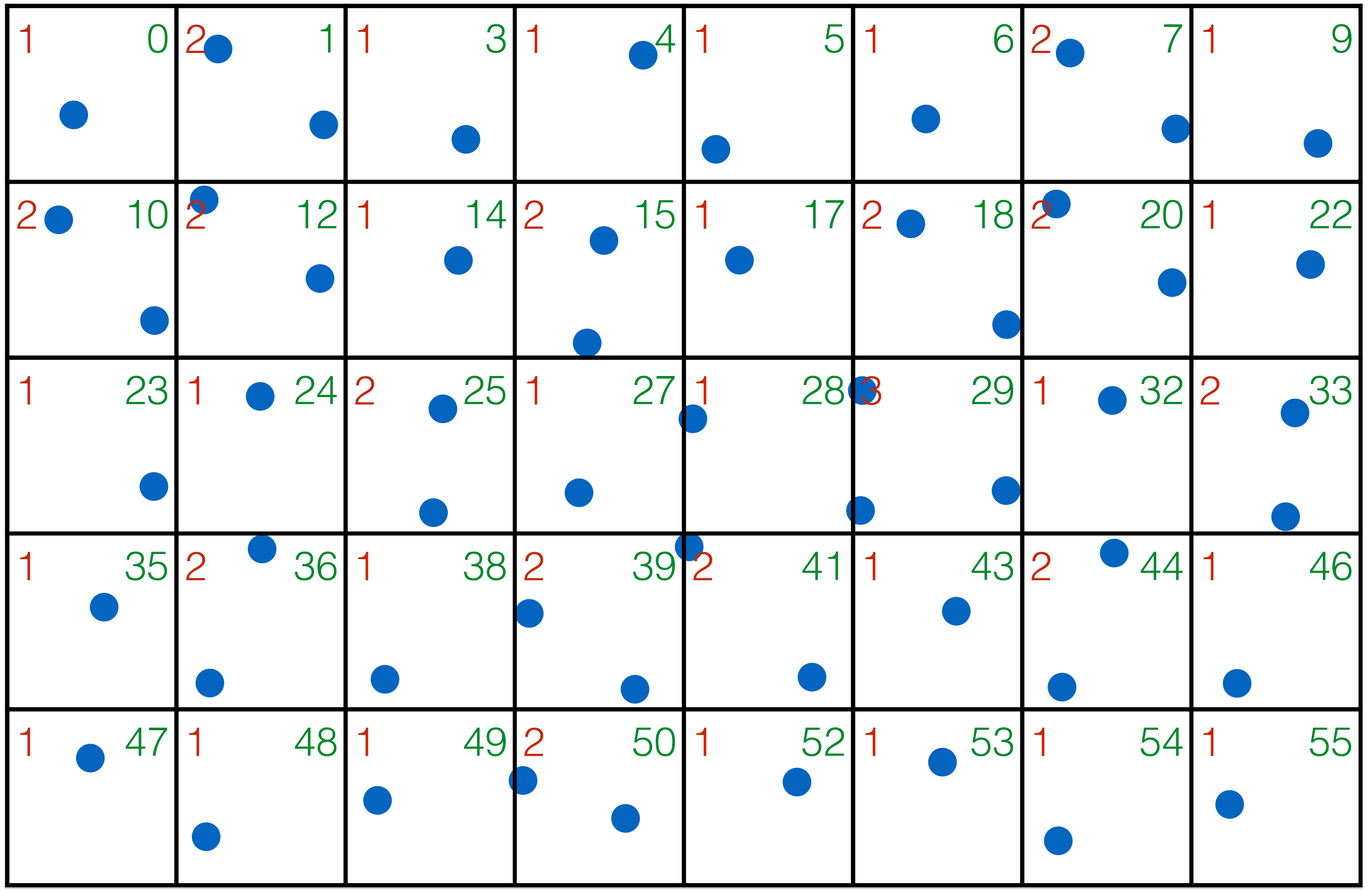
$$\left[-\frac{L_x}{2}, \frac{L_x}{2}\right] \times \left[-\frac{L_y}{2}, \frac{L_y}{2}\right]$$



Coding time: build cell lists

Fill the `TODOs` in `skeleton/celllists.cpp`

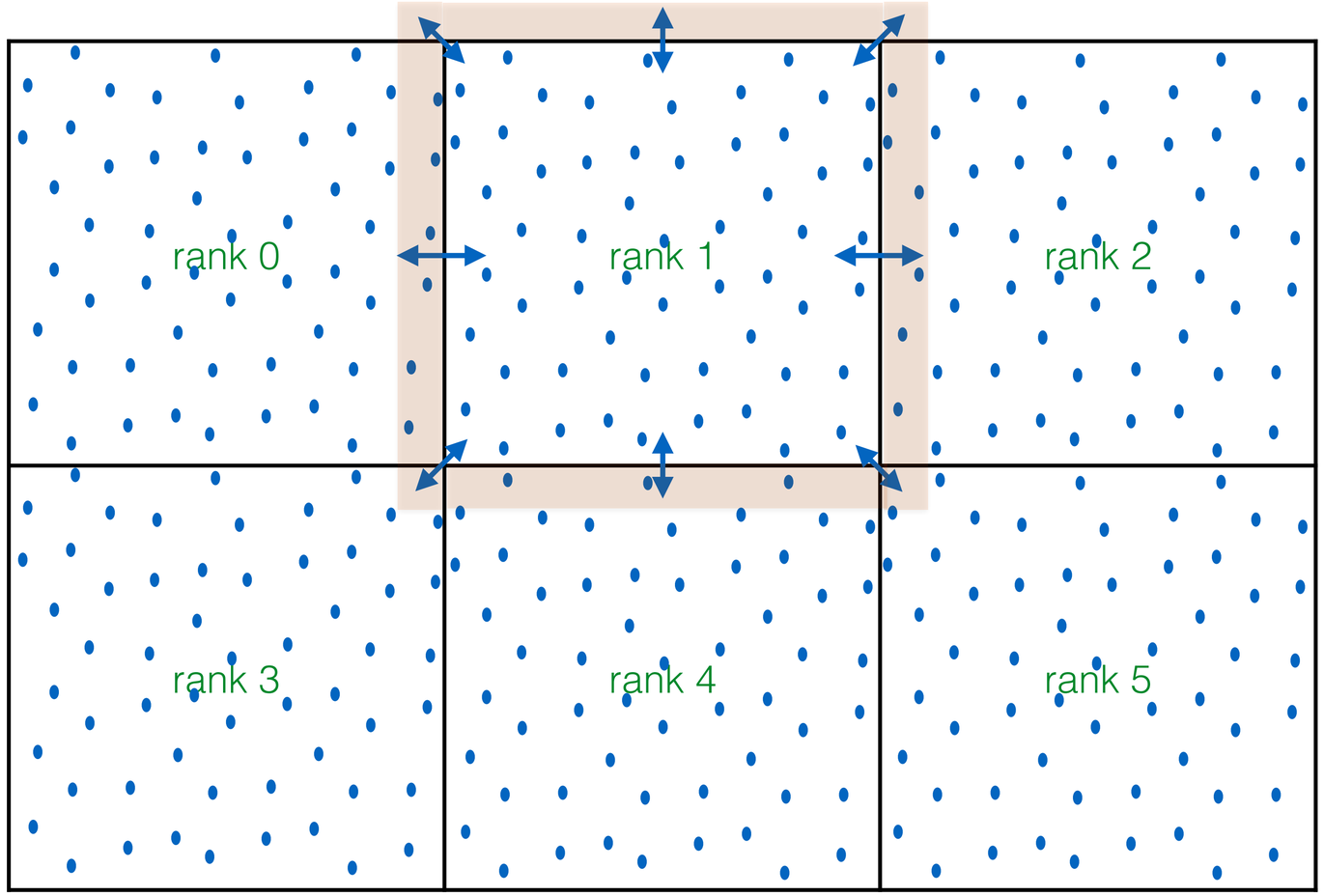
1. Count number of particles in each cell
2. Compute the starts of each cells (prefix sum)
3. Reorder the particles in memory



A simple 2D LJ simulation with MPI

One time step looks like:

1. redistribute the particles among ranks
2. build the cell lists
3. exchange ghost particles
4. compute interactions
5. update velocities and positions



Coding time: compute ghost interactions

Fill the `TODOs` in `skeleton/interactions.h`

1. loop over ghost particles (`srcParticles`)
2. find cell id
3. loop over (existing) neighbor cells
4. loop over each particle in those cells (`dstParticles`)
5. compute interaction and add force to those particles (`dstForces`)