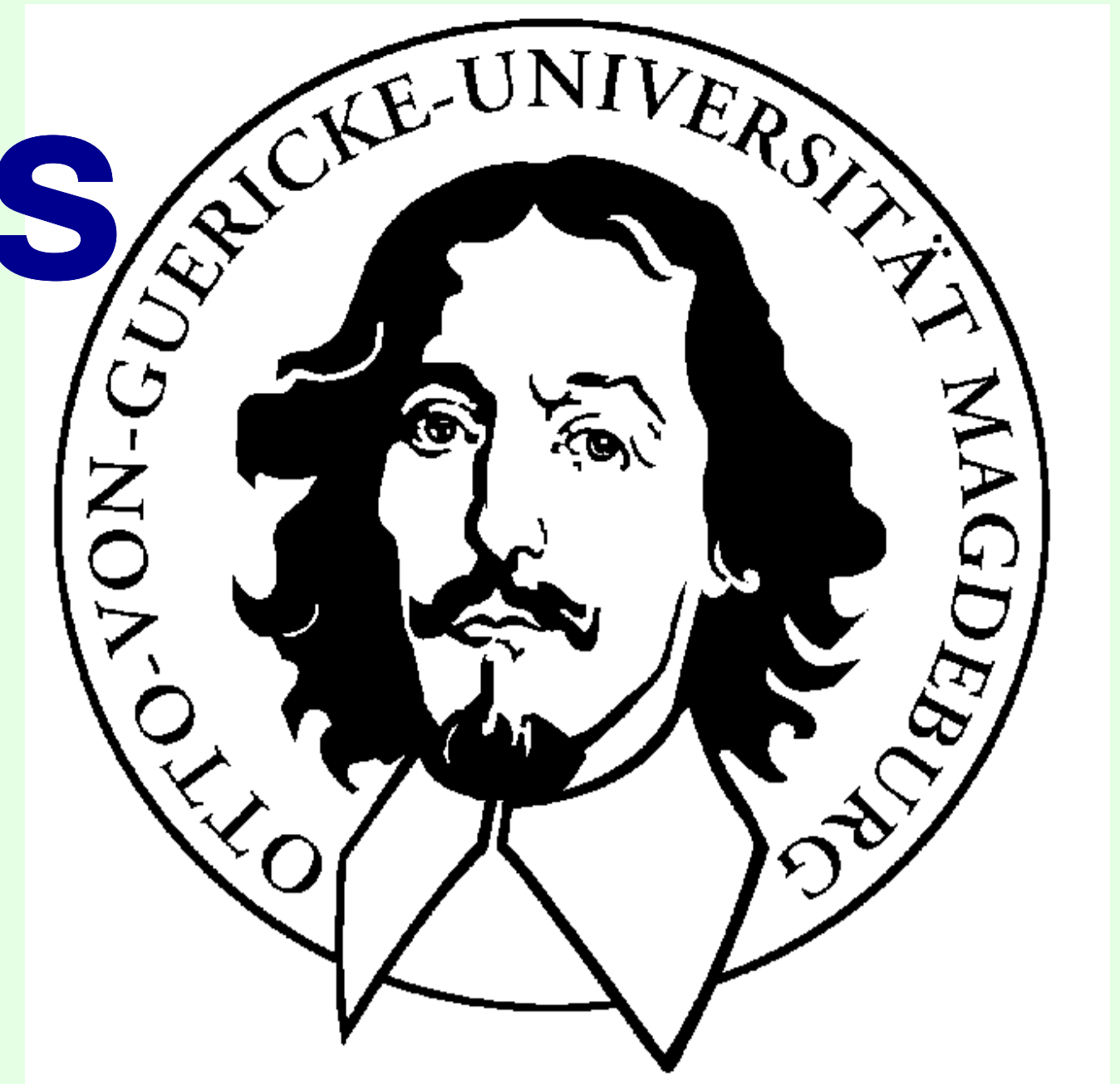


Shared Memory Parallelization for Molecular Dynamics Simulations of Non-Spherical Granular Materials

A. Schinner, K. Kassner

alexander.schinner@physik.uni-magdeburg.de, klaus.kassner@physik.uni-magdeburg.de

Fakultät für Naturwissenschaft, Computerorientierte Theoretische Physik



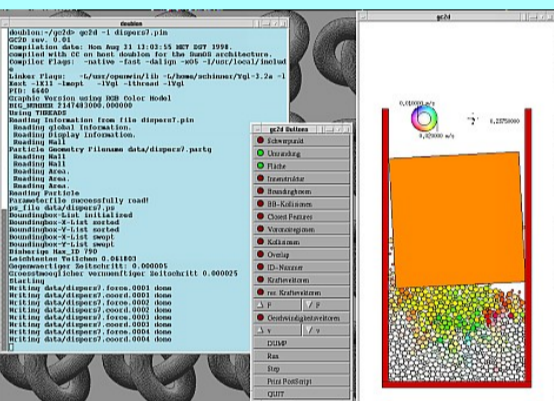
Introduction

Granular matter

- material properties
- structures

→ interesting physical phenomena


- "dip under the heap"
- size segregation
- stratification



gc2d-simulation: non-spherical particles fast algorithms

Parallelization Method

- shared memory workstations
- using threads
- try to reduce communication



SUN Ultra2
2 UltraSparc-I processors
shared memory

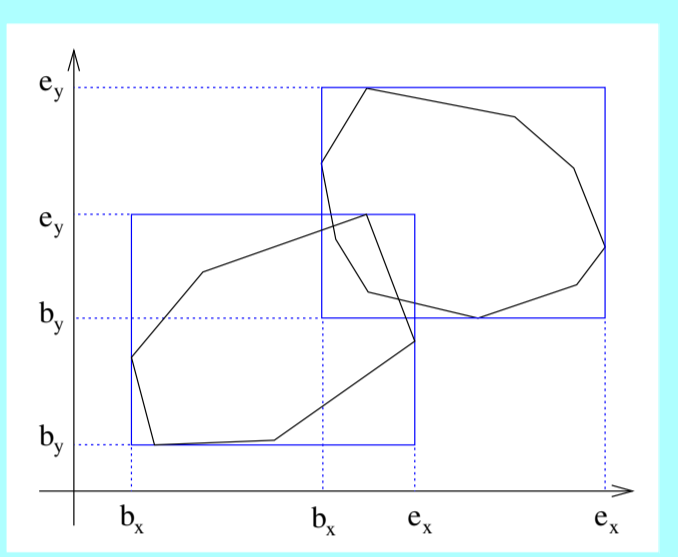
Substructure of the Algorithm for two threads

Thread #1		Thread #2	
Predictor	Synchronization	Predictor	} 1 time step
Bounding-Box X-Axes	semaphore	Bounding-Box Y-Axes	
	Synchronization		
Closest-Feature Algorithm	Synchronization	Closest-Feature Algorithm	
	Synchronization		
Force Calculation	semaphore	Force Calculation	
	Synchronization		
Corrector	Synchronization	Corrector	

- split the work into independent parts
- the subthreads shall not communicate
- the threads shall not write data used by other threads
- the threads have to be synchronized after each intermediate step

Step #1 Bounding-Box Algorithm

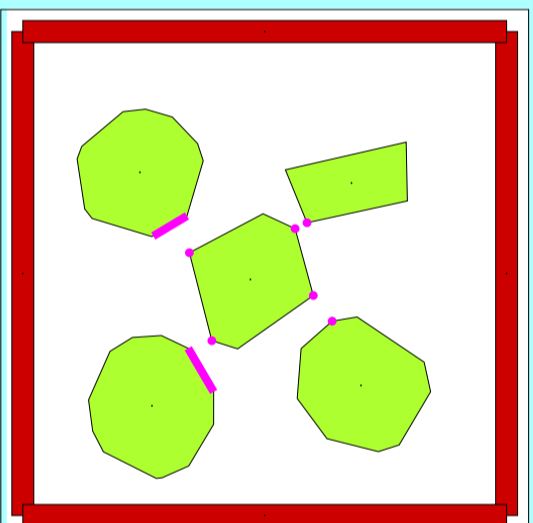
- Update the list of collisions by sorting both axes
- Each axis can be sorted by (at least) one thread
- Changes in the list of collisions can be made by both threads → **semaphore**



Result: a list of possible particle-particle collisions

Step #2 Closest-Feature Algorithm

- calculate distance of particles
- keep track of the closest features
- list of possible particle-particle collisions is distributed to the threads
- needed information will not be changed in this step → **no semaphore**
- results are not shared across the threads → **no semaphore**



Result: a list of particle-particle collisions

Step #3 Force Calculation

- calculate area of overlap for each collision
- calculate force and torque from overlap area
- list of particle-particle collisions is distributed to the threads.
- needed information is either local or will not be changed in this step → **no semaphore**
- contributions to the force for a certain particle may be calculated by different threads → **semaphore**

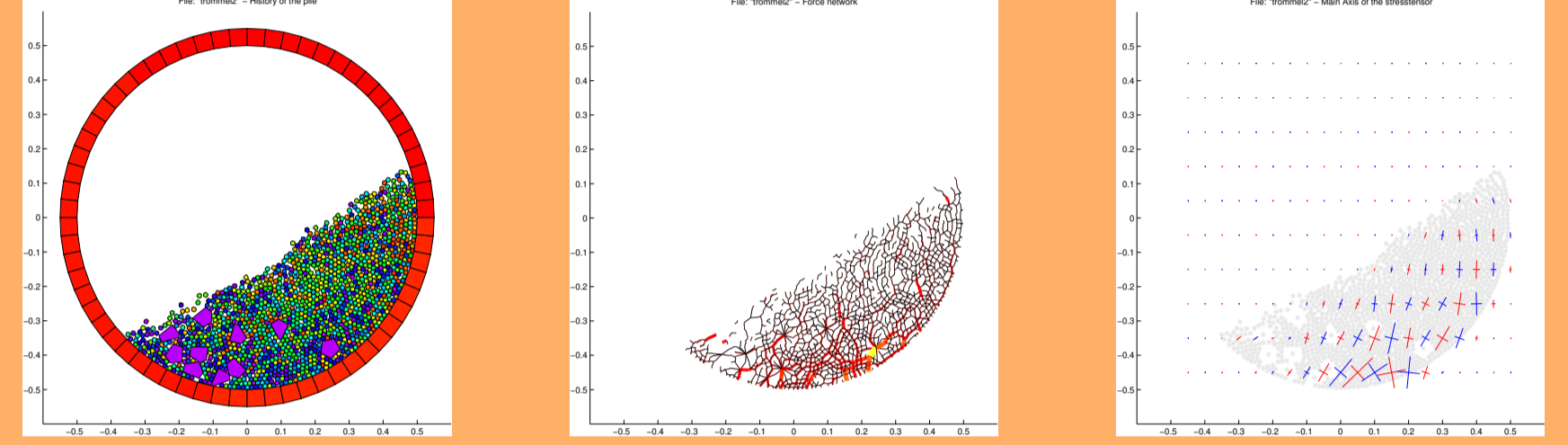
Result: forces and torques for each particle

Step #4 Differential Equation Solver

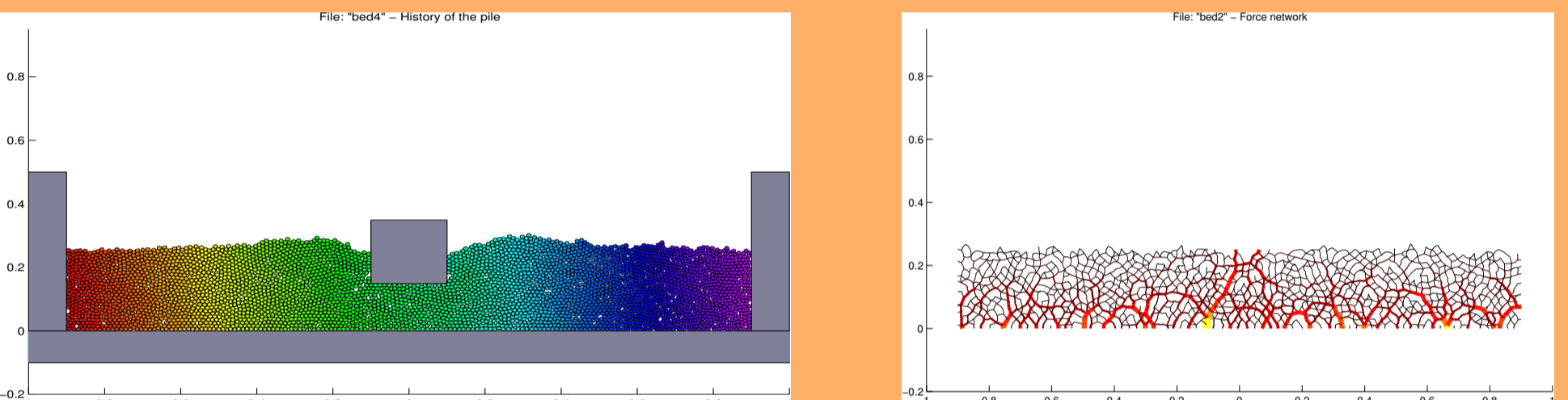
- divided into two parts
- predictor is calculated before Step #1
- corrector is calculated after Step #3
- list of particles is split among the threads
- needed information is local in this step → **no semaphore**
- result of one thread is not affected by another thread → **no semaphore**

Result: new position and velocity

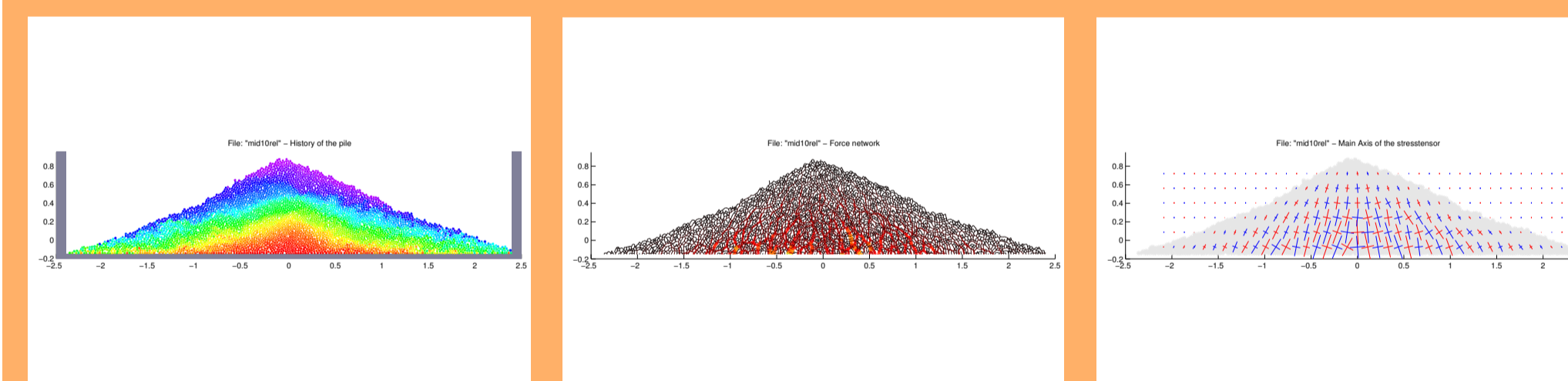
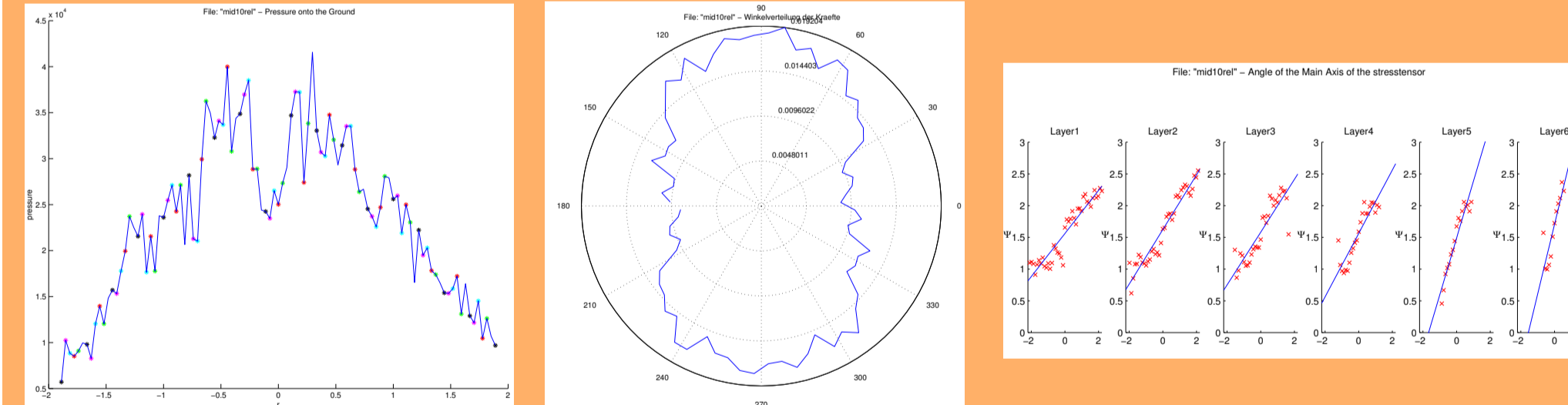
Inside of a Rotating Drum



A Big Particle Pressing onto a Bed of Granular Matter



Sandpile - Build from a Point Source

Threads - Basic Idea

A thread of control, or more simply a thread, is an independent sequence of execution of program code inside a UNIX process.

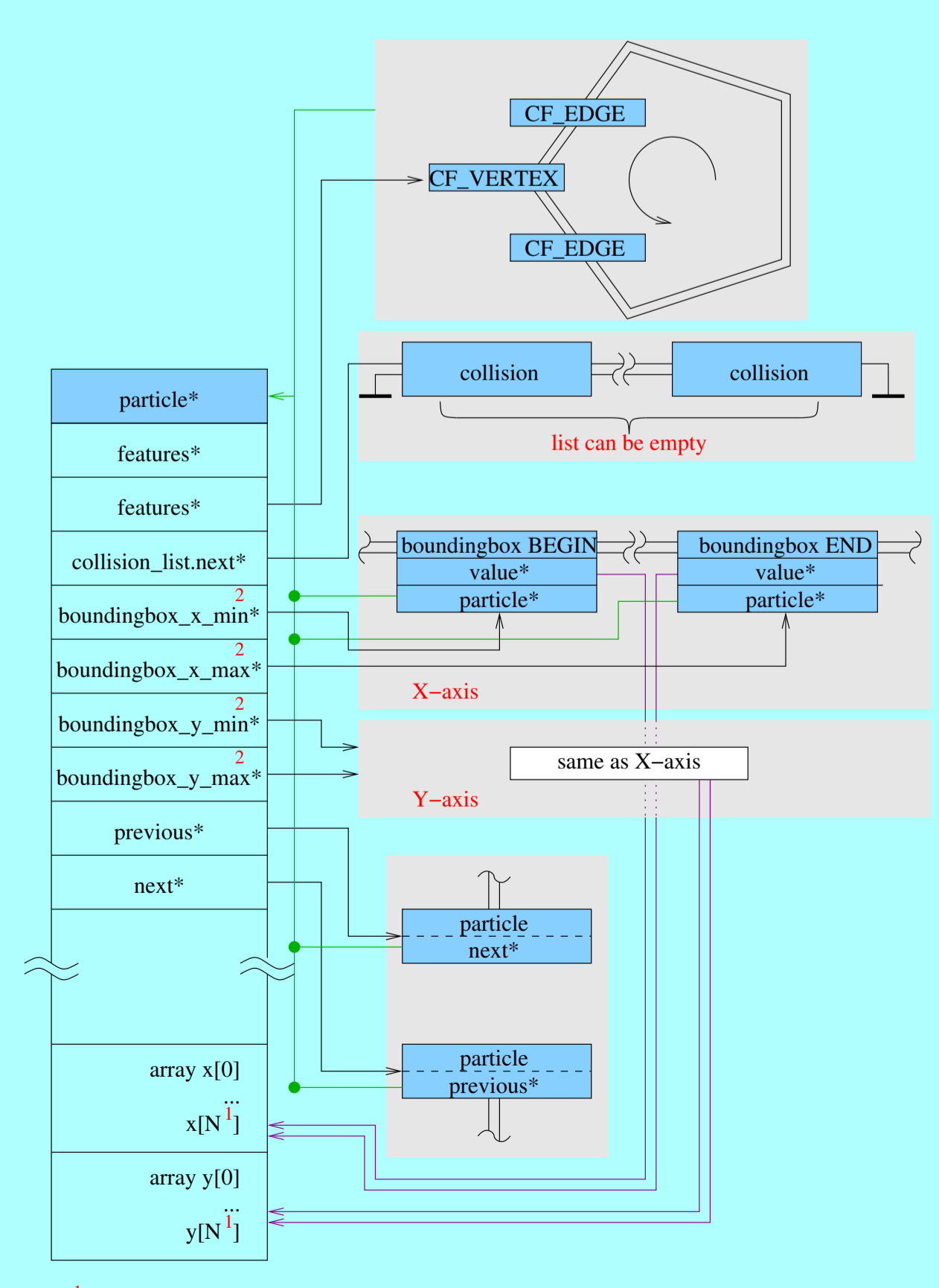
Serial Program

```
for i=0:4
  y(i)=x(i)*k
end
```

Multithreaded Program

Thread #1	Thread #2
for i=0:4 y(i)=x(i)*k end	for i=5:9 y(i)=x(i)*k end


Scheme of the Data Structure



¹ N=MAX_NUM_CORNER
² boundingbox_* are type of SENTINEL

Conclusion

- parallel (multithreaded) simulation of granular matter
- no communication, only synchronization
- semaphores only for rarely used variables
- symmetric algorithm



"But is that true symmetric multiprocessing?"