Simulations in Statistical Physics

Course for MSc physics students

Janos Török

Department of Theoretical Physics

October 1, 2013

Molecular dynamics

Program:

- Have an algorithm to calculate forces
- Get list of interacting particles
- Determine accelerations and velocities; step particles
- Set temperature

Temperature

Definition:

- Encyclopedia Britannica, Wikipedia:
  "A temperature is a numerical measure of hot or cold."
- Thermodynamics:
  Second law of thermodynamics & Carnot engine

$$\delta Q = T dS$$

$$\eta_{\text{max}} = \eta_{\text{Carnot}} = 1 - T_C / T_H$$

- Statistical physics:

$$\beta \equiv \frac{1}{k_B} \left(\frac{\partial S}{\partial E} \right)_{V,N} = \frac{1}{k_B T}$$

Definition of temperature

Temperature is a measure of the random submicroscopic motions and vibrations of the particle constituents of matter.

The average kinetic energy per particle degree of freedom is

$$\bar{E} = \frac{1}{2} k_B T$$

Molecular dynamics conserves only the *total* energy!

Task: Control kinetic energy!

Setting temperature

- Experiment
  - Environment
  - Mixing $\rightarrow$ uniform temperature
- Simulation
  - Control the kinetic energy (velocities)
  - Mixing $\rightarrow$ Maxwell-Boltzmann distribution

Nosé-Hoover thermostat

- Original Hamiltonian

$$H_0 = \sum_i \frac{\mathbf{p}_i^2}{2m_i} + U(\mathbf{q})$$

- Heatbath in the Hamiltonian:

$$H_n = \sum_i \frac{\mathbf{p}_i'^2}{2m_i} + U(\mathbf{q}') + \frac{p_s^2}{2Q} + gk_B T \log(s)$$

- Extra degree of freedom s.
- Q "mass" related to $s \rightarrow$ controls the speed of convergence
- $g = 3N$ the number degrees of freedom
- $\mathbf{p}'$ and $\mathbf{q}'$ are virtual coordinates

Nosé-Hoover thermostat

- Virtual coordinates, vs. original ones:

$$\mathbf{p} = \mathbf{p}'$$
$$\mathbf{q} = \mathbf{q}'/s$$
$$t = \int \frac{1}{s} dt'$$

- Solution of the new Hamiltonian:

$$\xi = \dot{s}/s = P_s/Q$$
$$\dot{\mathbf{q}} = \frac{\mathbf{p}}{m}$$
$$\dot{\mathbf{p}}_i = -\frac{\partial U}{\partial q_i} - \xi \mathbf{p}_i$$
$$\dot{\xi} = \frac{1}{Q}\left(\sum_i \frac{\dot{\mathbf{p}}_i}{m_i} - g k_B T\right)$$

Molecular dynamics

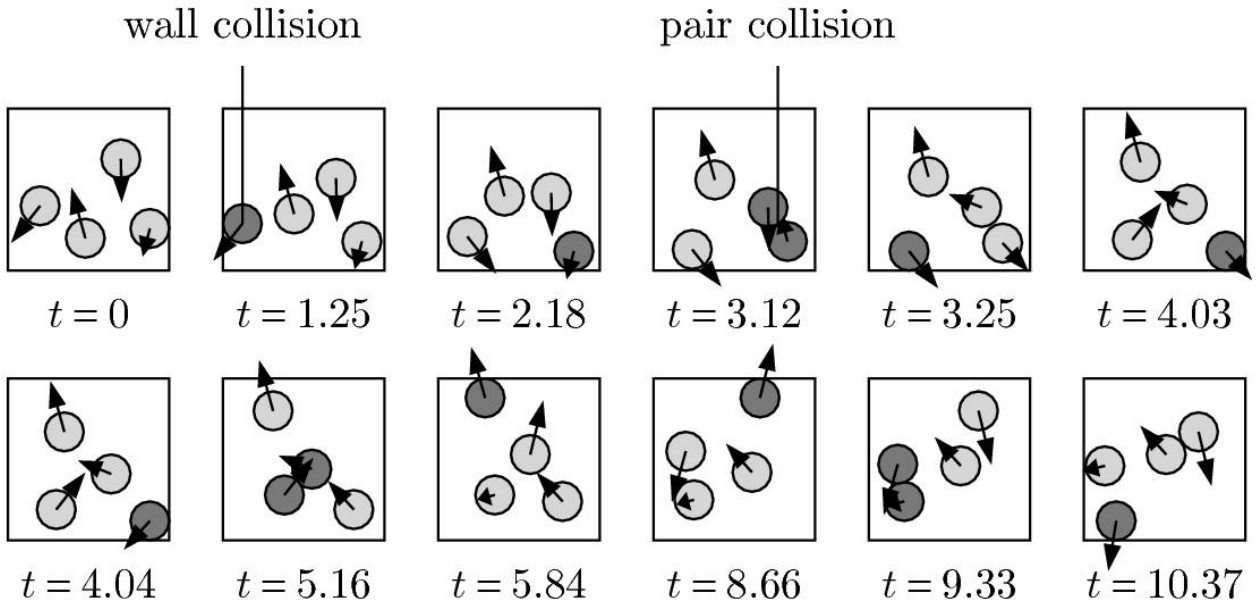
- Create sample
  - Crystal
  - Random deposition
  - Distorted crystal
- Temperate sample
- Make test
- Collect data
  - Data size: e.g. $N = 10^4$, $t = 10^6$ small simulation:
  - 1 hour on 1 core PC
  - 3 doubles/atom $\rightarrow$ 24 bytes/atom/timesteps
  - Result 2.4 10^{11} bytes = 240 Gigabytes

Alternatives

- Event Driven Dynamics
- Contact Dynamics
- Kinetic Monte Carlo

Event driven dynamics

- Hard core interactions
- Interactions short in time compared to flight
- (MD needs ~ 20 timesteps per collision, overlap of $10^{-3}d$)
- Integrable path $\rightarrow$ do it



Event driven algorithm

- Particles: $\mathbf{x}_i(t)$, $\mathbf{v}_i(t)$, $\boldsymbol{\omega}_i(t)$, $\mathbf{r}_i$
- Calculate collision time: Let $\mathbf{d}_{ij} = \mathbf{x}_i - \mathbf{x}_j$, Then

$$\tau_{ij} = \frac{|\mathbf{d}_{ij}|^2}{(\mathbf{v}_i - \mathbf{v}_j)\mathbf{d}_{ij}}$$

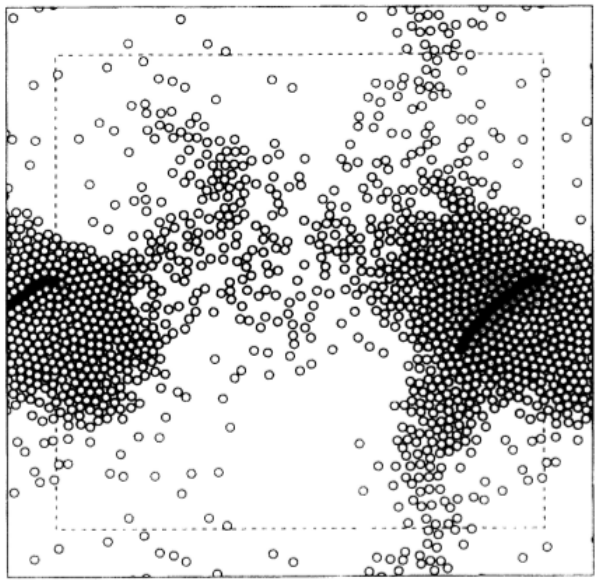
- Order collision times, get the smallest $\tau_c = \min_{ij}(\tau_{ij})$
- Go to time $t + \tau_c$: $\mathbf{x}_i(t + \tau_c)$
- Calculate velocities after collision $\mathbf{v}_i(t + \tau_c)$ (may be hard...)
- Restart loop

Next time:

- Calculate collision time only with i, j
- Dynamic list, change only newly calculate collision times

Inelastic collapse

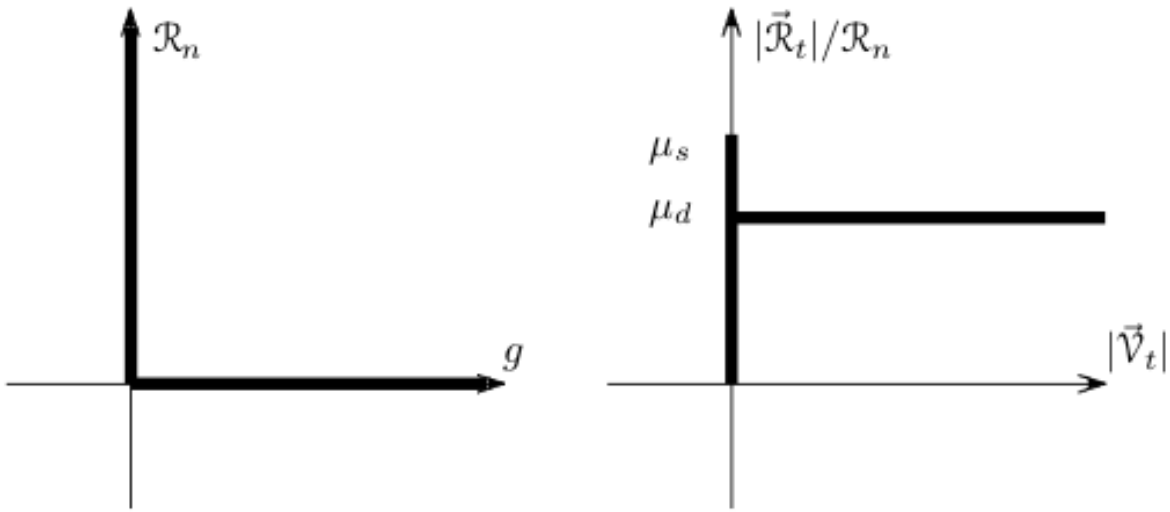
- Coefficient of restitution $r = v_n(t_c+)/v_n(t_c-)$
- Energy is lost in an exponential way (Ping Pong)
- Infinite collisions in finite time
- Solution $\rightarrow$ $r = 1$ if collisions occur more frequently than a parameter t_{cont}, the contact duration
- Contact $\rightarrow$ small vibration :-(well...)



Contact dynamics

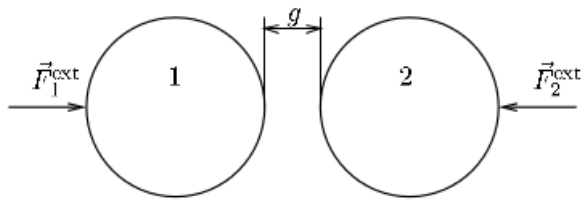
- Perfectly rigid particles
- Constraints
- Implicit forces

$$\mathbf{v}_i(t+\Delta t) = \mathbf{v}_i(t) + \frac{1}{m_i}\mathbf{F}_i(t+\Delta t)\Delta t$$
$$\mathbf{x}_i(t+\Delta t) = \mathbf{x}_i(t) + \mathbf{v}_i(t+\Delta t)\Delta t$$



Contact dynamics, force calculation

- Two particles with gap g



$$\text{if} \quad {\mathcal{V}^{\text{free}}}_n \Delta t + g^{\text{pos}} > 0$$

$$\text{then} \quad \left\{ \vec{\mathcal{R}}^{\text{new}} = 0 \right. \qquad \text{(no contact)}$$

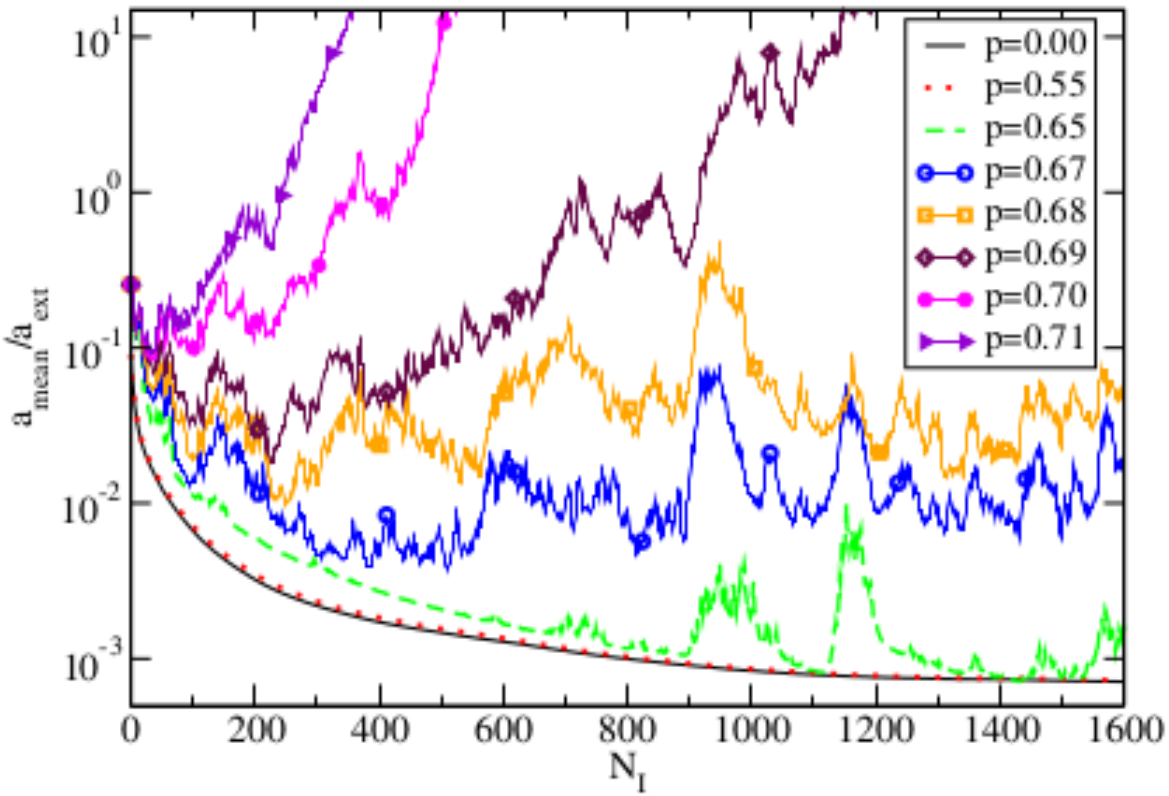
$$\text{else} \quad \left\{ \begin{array}{l} \mathcal{R}_n^{\text{new}} = -\dfrac{1}{\Delta t} m_n \left(\dfrac{g^{\text{pos}}}{\Delta t} + \mathcal{V}_n^{\text{free}} \right) \\ \vec{\mathcal{R}}_t^{\text{new}} = -\dfrac{1}{\Delta t} m_t \vec{\mathcal{V}}_t^{\text{free}} \end{array} \right. \qquad \text{(sticking contact)}$$

$$\text{if} \quad \left| \vec{\mathcal{R}}_t^{\text{new}} \right| > \mu \mathcal{R}_n^{\text{new}}$$

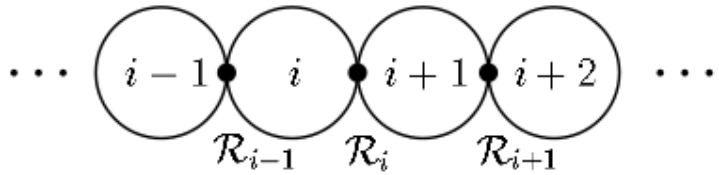
$$\text{then} \quad \left\{ \vec{\mathcal{R}}_t^{\text{new}} = \mu \mathcal{R}_n^{\text{new}} \frac{\vec{\mathcal{R}}_t^{\text{new}}}{\left| \vec{\mathcal{R}}_t^{\text{new}} \right|} \right. \qquad \text{(sliding contact)}$$

Iterative solver

- Updates:
  - Parallel: calculate all contacts with old values then change to new at once $\rightarrow$ serious instabilities
  - Serial: update contacts one-by-one in random order



Particle chain



- One iteration step:

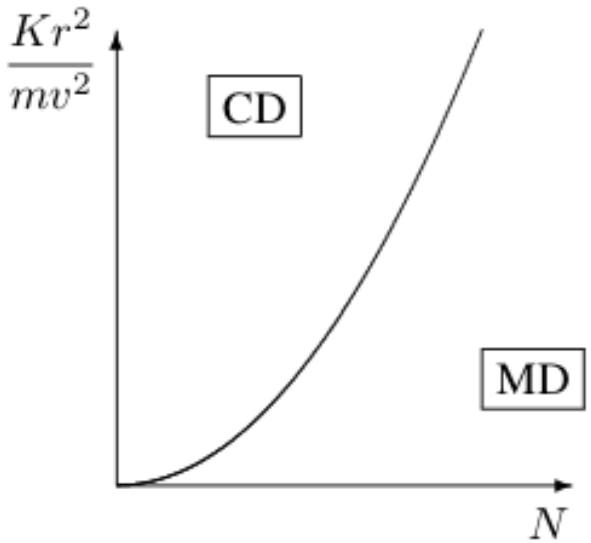
$$\mathcal{R}_i^{\text{new}} = \frac{1}{2}\left(\mathcal{R}_{i-1}^{\text{new}} + \mathcal{R}_{i+1}^{\text{new}}\right),$$

- Discretized one-dimensional diffusion equation
- Model of rigid particles $\rightarrow$ elastic
- Elasticity depends on the number of iterations

Molecular versus Contact dynamics

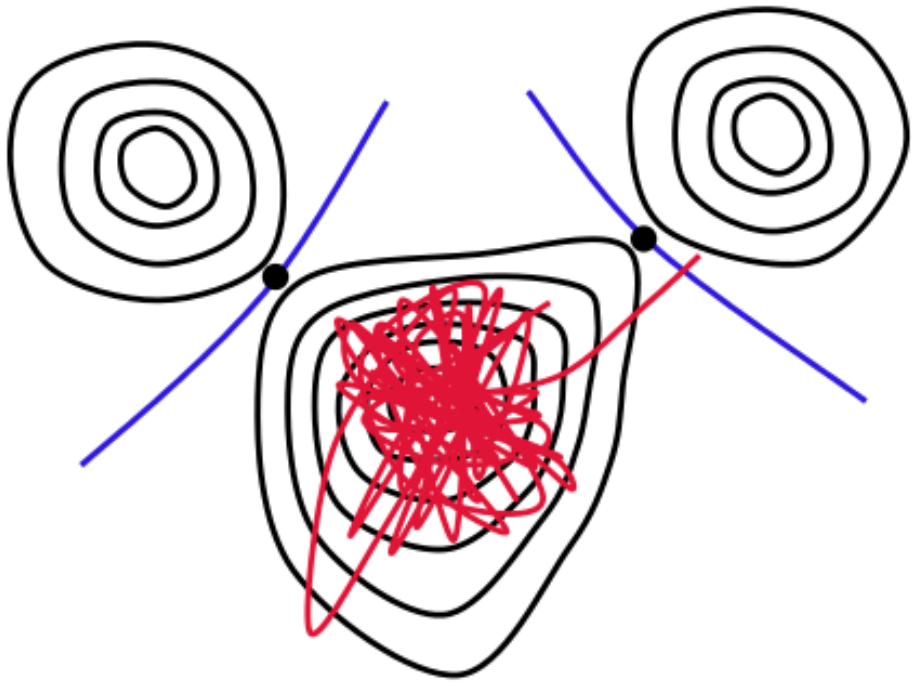
Limit

$$\frac{Kr^2}{mv^2} = N^{4/d}$$



Kinetic Monte Carlo

- Particle sits in a potential well for ages . . .
- What to do?



Kinetic Monte Carlo

- Long lasting steady state positions
- Slow thermally activated processes
- Infrequent-event system

Solution:

- Consider only jumps between neighboring energy wells
- Probability of jump $P \sim \exp(-\beta E_b)$
- Rate of jump $i \rightarrow j$, $k_{ij} = E_b$.

Kinetic Monte Carlo

- Get all possible rates k_i
- Calculate the cumulative function $K = \sum_i k_i$
- Get a uniform random number u (between 0 and 1)
- Execute the event i for which $\sum_{j=1}^{i} k_j \geq Ku > \sum_{j=1}^{i-1} k_j$
- Get new uniform random number u' (between 0 and 1)
- Update time to $t = t + \Delta t$, $\Delta t = -\log(u')/k_i$
- Recalculate rates, which have changed
- Restart loop

Monte Carlo

Why Monte Carlo? $\rightarrow$ Random numbers play big role!

Kinetic Monte Carlo

- Rates
  - Physics
  - Molecular dynamics
- Must include all rates!
- Used for:
  - Surface diffusion
  - Surface growth
  - Syntering
  - Domain evolution

Example....

Methods

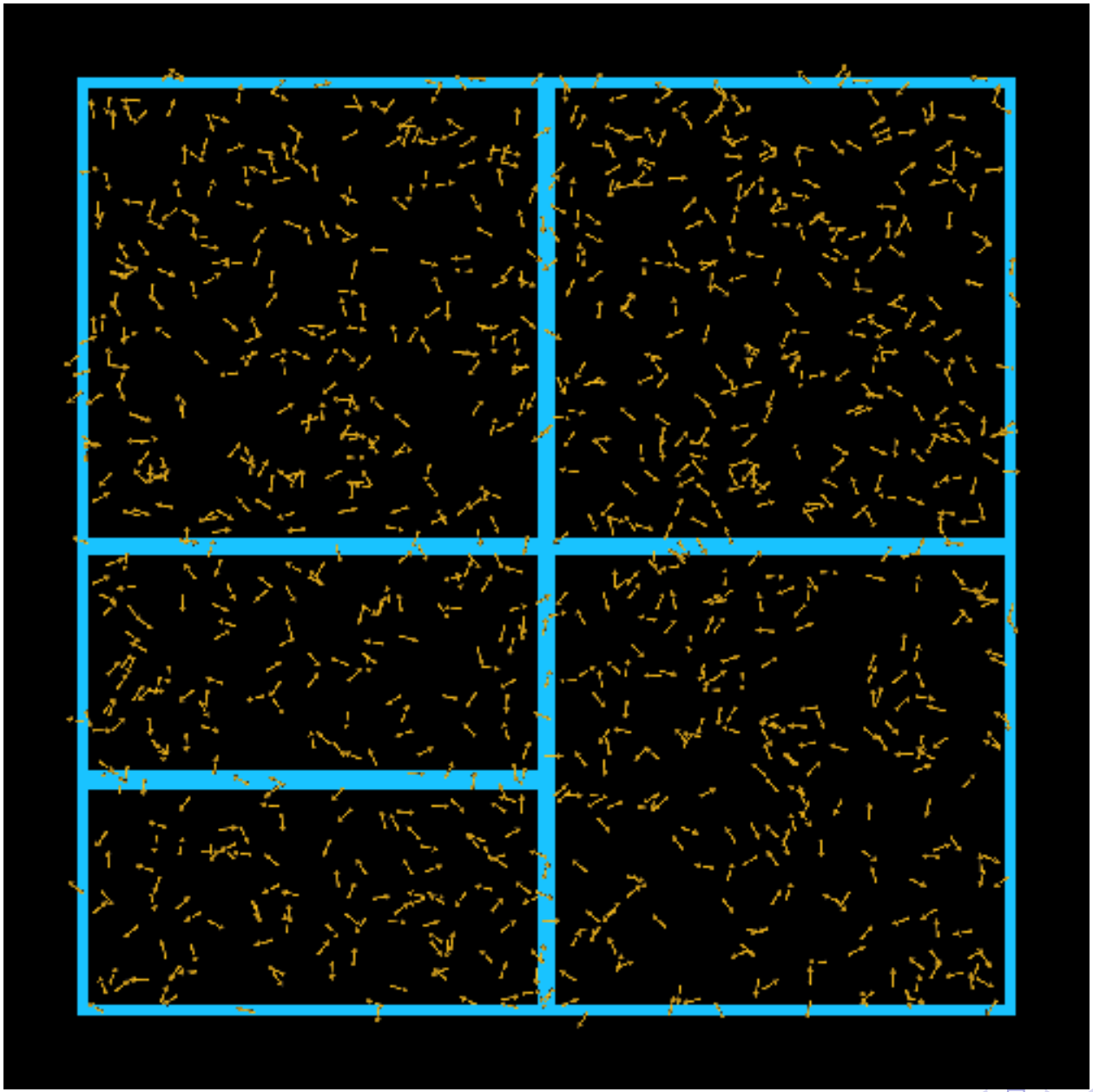
- Molecular Dynamics
  - General
- Event Driven Dynamics
  - Hard objects, at low density
- Contact Dynamics
  - Rigid particles
- Kinetic Monte Carlo
  - Infrequent events, bonded particles

Parallelization

- Why?
  - The speed of one core processor is limited
  - Larger system sizes
  - Multi-core processors
  - On multi-core system inter-processor data change is fast
- Why not?
  - Computing power is lost
  - Much more code development
  - Very often ensemble average is needed
  - Inter-computer communication is terribly slow

RAM $\rightarrow$ $\sim$15GB/s, Ethernet 125MB/s, Infiniband $\sim$1GB/s

Parallelization (Bird flocking model)

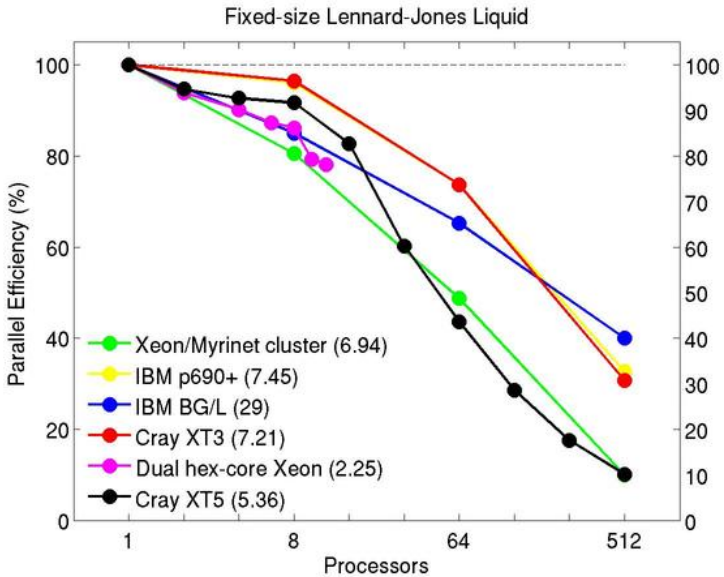


Parallelization

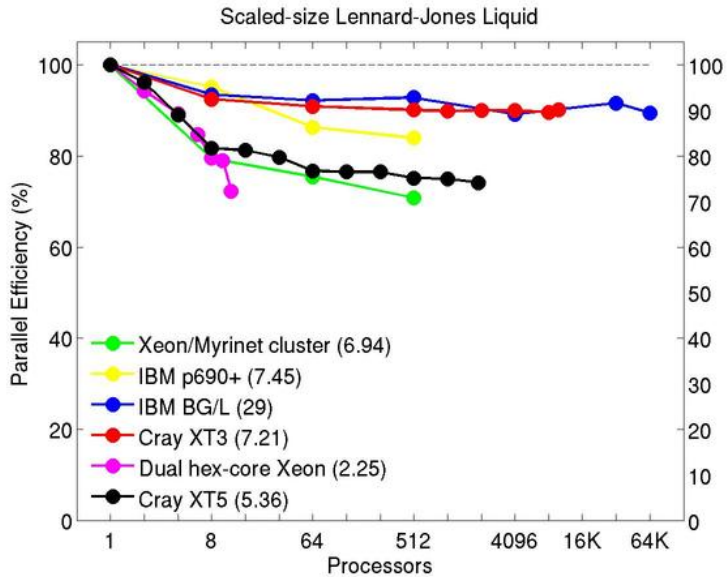
- Molecular dynamics
  - Short range interactions: Box must be duplicated, Verlet in parallel
  - Long range: Parallel fast Fourier transformation
- Contact dynamics
  - Short range interactions: Box must be duplicated, Iteration in parallel
- Event Driven Dynamics
  - List must be global, no way!
- Kinetic Monte Carlo
  - List must be global, no way!

Efficiency of parallelization

- Large systems are needed
- Boundary must be minimal

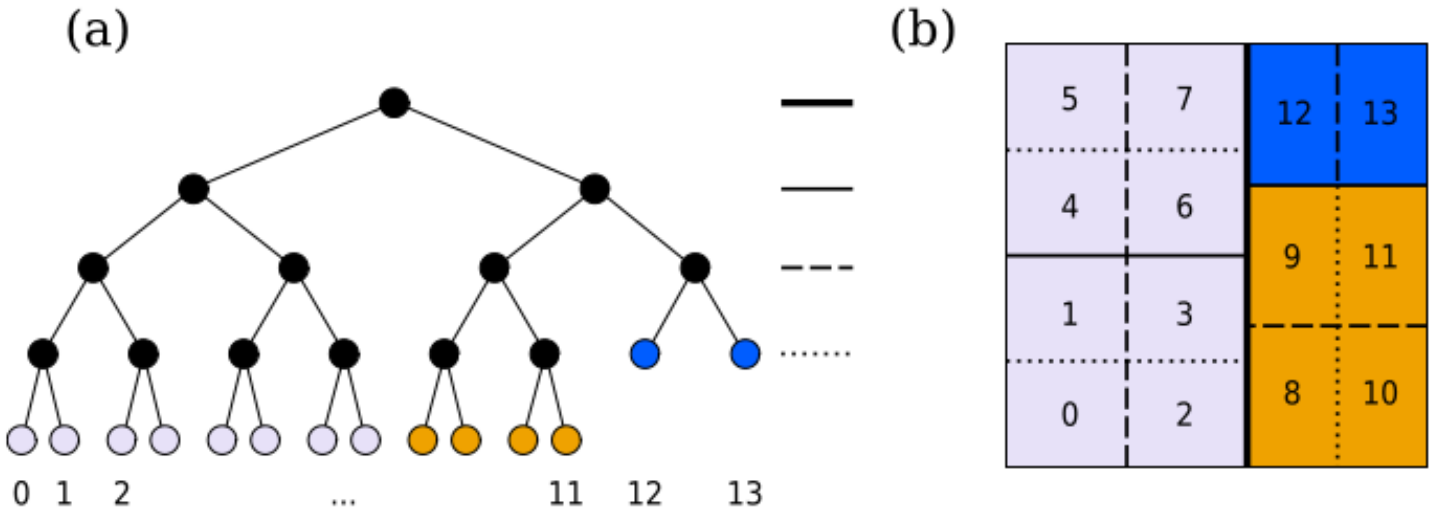


$N = 32000$ atoms



$N = 32000$ atoms/processor

Domain decomposition



- Calculate time spent in a branch
- Move line if necessary
- Lower in tree (up in Fig), larger the mass
- Only rarely, data transfer is expensive