

# Simulations in Statistical Physics

## Discrete element methods part 2.

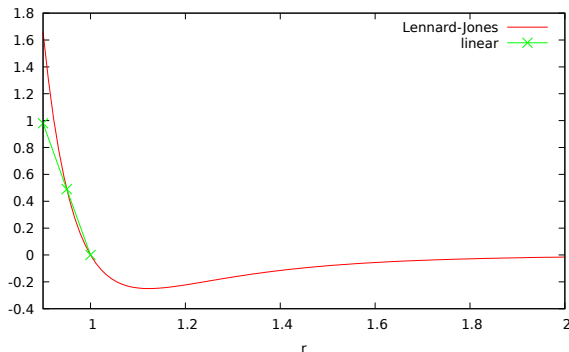
Janos Török

Department of Theoretical Physics

March 3, 2022

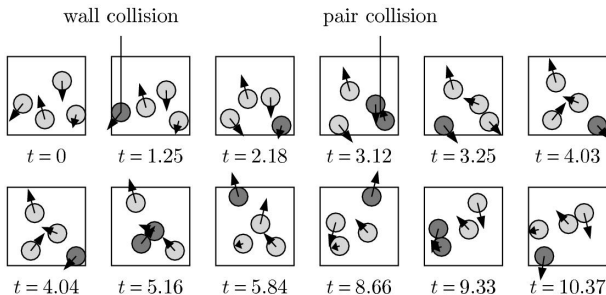
# Problems with molecular dynamics

- ▶ All particles are soft (force as function of overlap)
- ▶ Collision must be smooth ( $\sim 50$  timesteps), which sets  $dt$
- ▶ Problem with gases, free path is 100–1000 times the particle radius
- ▶ Timescale problems with temporal fluctuations



# Event driven dynamics

- ▶ Hard core interactions
- ▶ Interactions short in time compared to flight
- ▶ (MD needs  $\sim 20 - 50$  timesteps per collision, overlap of  $10^{-3}d$ )
- ▶ Integrable path  $\rightarrow$  do it



# Event driven algorithm

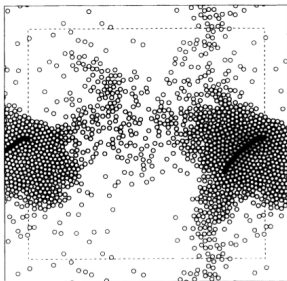
- ▶ **No gravity** (Can be also solved for gravity)
- ▶ Particles:  $r_i(t)$ ,  $v_i(t)$ ,  $\omega_i(t)$ ,  $R_i$
- ▶ Calculate collision time: Let  $d_{ij} = |r_i - r_j| - R_i - R_j$ , Then

$$\tau_{ij} = \frac{|d_{ij}|^2}{(v_i - v_j)d_{ij}}$$

- ▶ Order collision times, get the smallest  $\tau_c = \min_{ij}(\tau_{ij})$
- ▶ Go to time  $t + \tau_c$   $r_i(t + \tau_c)$
- ▶ Calculate velocities after collision  $v_i(t + \tau_c)$  (may be hard...)
- ▶ Restart loop
- ▶ Next time Calculate collision time only with  $i, j$
- ▶ Dynamic list, change only newly calculate collision times

# Inelastic collapse

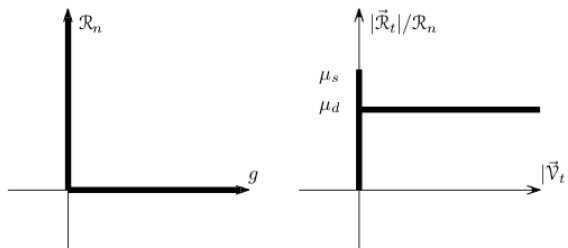
- ▶ Coefficient of restitution  $r = v_n(t_c+)/v_n(t_c-)$
- ▶ Energy is lost in an exponential way (Ping Pong)
- ▶ Infinite collisions in finite time
- ▶ Solution  $\rightarrow r = 1$  if collisions occur more frequently than a parameter  $t_{\text{cont}}$ , the contact duration
- ▶ Contact  $\rightarrow$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



MOVIE

# Contact dynamics

- ▶ Perfectly rigid particles
- ▶ Non-smooth dynamics
- ▶ Constraints



- ▶ Molecular dynamics
  - ▶ Normal force: overlap (smooth)
  - ▶ Shear force: history

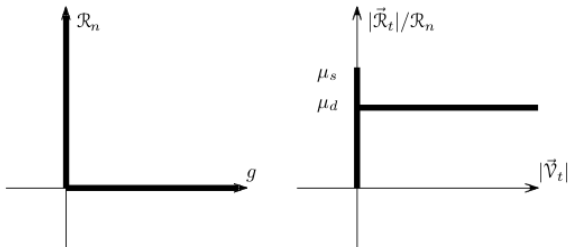
## Contact dynamics

- ▶ Implicit forces

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

$$x_i(t + \Delta t) = x_i(t) + v_i(t + \Delta t)\Delta t$$

such as constraints are fulfilled



# Contact dynamics

## ► Implicit forces

$$v_i(t + \Delta t) = v_i(t) + \frac{1}{m_i} F_i(t + \Delta t) \Delta t$$

$$x_i(t + \Delta t) = x_i(t) + v_i(t + \Delta t) \Delta t$$

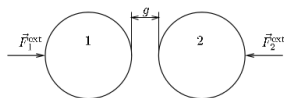
such as constraints are fulfilled

- if gap would be negative increase force
- if there would be a shear displacement increase shear force
- if shear force is larger than allowed restrict it to that value



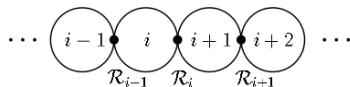
# Contact dynamics, force calculation

- Two particles with gap  $g$



$$\begin{aligned} &\text{if } \mathcal{V}_n^{\text{free}} \Delta t + g^{\text{pos}} > 0 \\ &\quad \text{then } \begin{cases} \vec{\mathcal{R}}^{\text{new}} = 0 \end{cases} && \text{(no contact)} \\ \\ &\quad \text{else } \begin{cases} \mathcal{R}_n^{\text{new}} = -\frac{1}{\Delta t} m_n \left( \frac{g^{\text{pos}}}{\Delta t} + \mathcal{V}_n^{\text{free}} \right) \\ \vec{\mathcal{R}}_t^{\text{new}} = -\frac{1}{\Delta t} m_t \vec{\mathcal{V}}_t^{\text{free}} \end{cases} && \text{(sticking contact)} \\ \\ &\text{if } \left| \vec{\mathcal{R}}_t^{\text{new}} \right| > \mu \mathcal{R}_n^{\text{new}} \\ &\quad \text{then } \begin{cases} \vec{\mathcal{R}}_t^{\text{new}} = \mu \mathcal{R}_n^{\text{new}} \frac{\vec{\mathcal{R}}_t^{\text{new}}}{\left| \vec{\mathcal{R}}_t^{\text{new}} \right|} \end{cases} && \text{(sliding contact)} \end{aligned}$$

# Particle chain



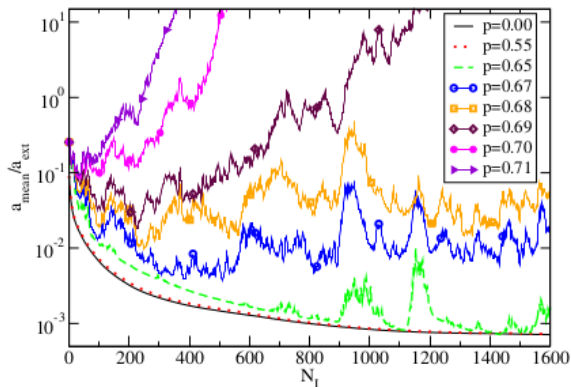
- One iteration step:

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

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

# Iterative solver

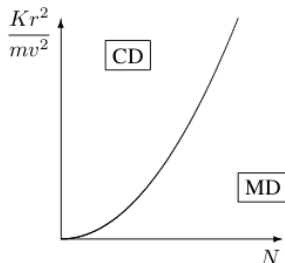
- Updates:
  - Parallel: calculate all contacts with old values then change to new at once → serious instabilities
  - Serial: update contacts one-by-one in random order



# Molecular versus Contact dynamics

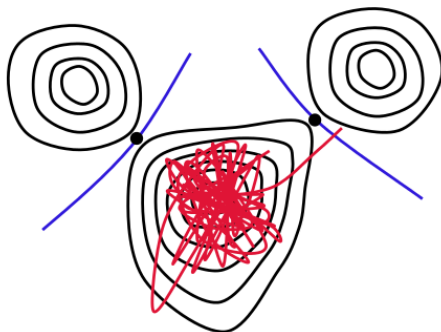
Limit

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



# Kinetic Monte Carlo

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

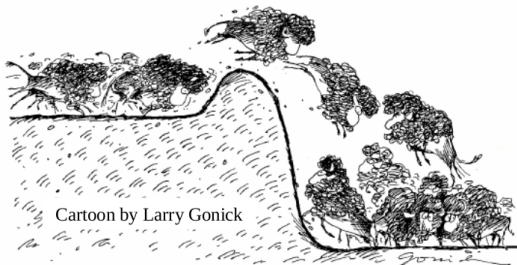


# Kinetic Monte Carlo

- ▶ 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

- ▶ All possible moves  $i$
- ▶ Rates for moves  $k_i$
- ▶ 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  $\sum_{j=1}^i k_j > uK > \sum_{j=1}^{i-1} k_j$
- ▶ 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

# Kinetic Monte Carlo

- ▶ Rates
  - ▶ Physics
  - ▶ Molecular dynamics
- ▶ Must include all rates!
- ▶ Used for:
  - ▶ Surface diffusion
  - ▶ Surface growth
  - ▶ Sintering
  - ▶ 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

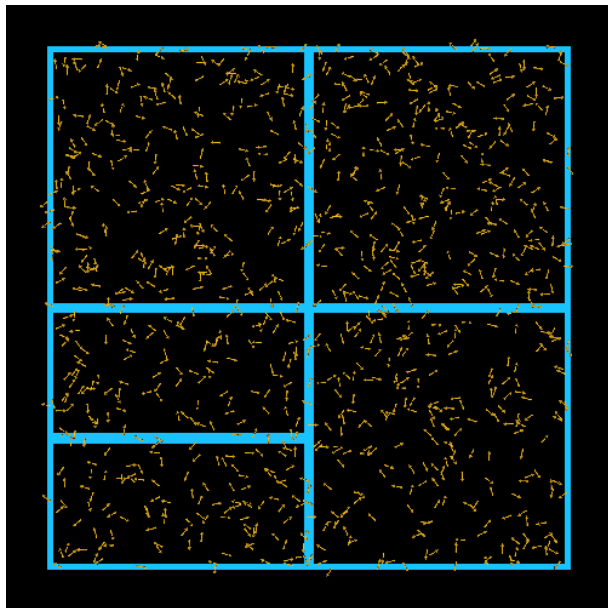
- ▶ Why?
  - ▶ The speed of one core processor is limited
  - ▶ Larger system sizes
  - ▶ Multi-core processors
  - ▶ On multi-core system inter-processor data change is fast
- ▶ Why not?
  - ▶ Computing power is lost
  - ▶ **Much more code development**
  - ▶ Very often ensemble average is needed
 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

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

# Parallelization: How?

- ▶ Code asks for more instances (e.g. run a loop in parallel)
  - ▶ Fork, multi-threading
  - ▶ Used in desktop applications
  - ▶ Punished on clusters
  - ▶ Shared memory
- ▶ Operating system (or even multiple machines) launches the code multiple times which can communicate
  - ▶ Now de facto standard: MPI (Message passing interface)
  - ▶ Communication is standardized, environment can be inhomogeneous
- ▶ GPU:
  - ▶ High number of cores
  - ▶ Non-standard processors
  - ▶ Non-standard libraries
  - ▶ Limited memory

## Parallelization (Bird flocking model)



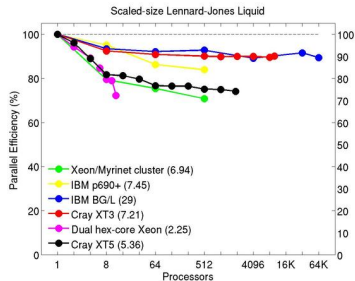
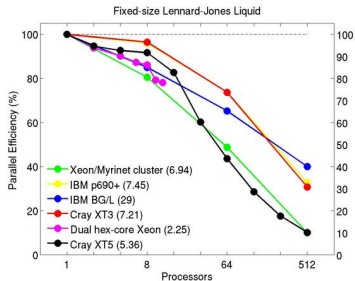
# Parallelization

Extra steps needed:

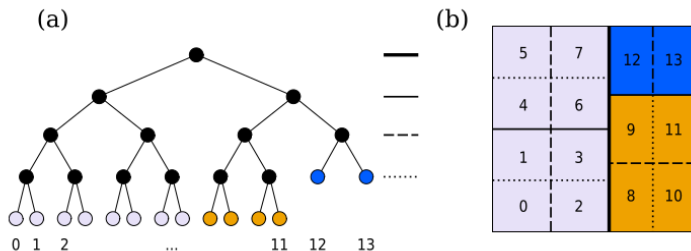
- ▶ Molecular dynamics
  - ▶ Short range interactions: Box must be duplicated, Verlet in parallel
  - ▶ Long range: Parallel fast Fourier transformation
- ▶ Contact dynamics
  - ▶ Short range interactions: Box must be duplicated, Iteration in parallel
- ▶ Event Driven Dynamics
  - ▶ List must be global,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

# Efficiency of parallelization

- ▶ Large systems are needed
- ▶ Boundary must be minimal
- ▶ System size can be increased simulation time not really



# Efficiency of parallelization



- ▶ Calculate time spent in a branch
- ▶ Calculate  $\sigma_T = \sqrt{\langle T^2 \rangle - \langle T \rangle^2} / \langle T \rangle$
- ▶ Move line if necessary ( $\sigma_T > \sigma_T^*$ )
- ▶ Lower in tree (up in Fig), larger the mass of the border
- ▶ Only rarely, data transfer is expensive