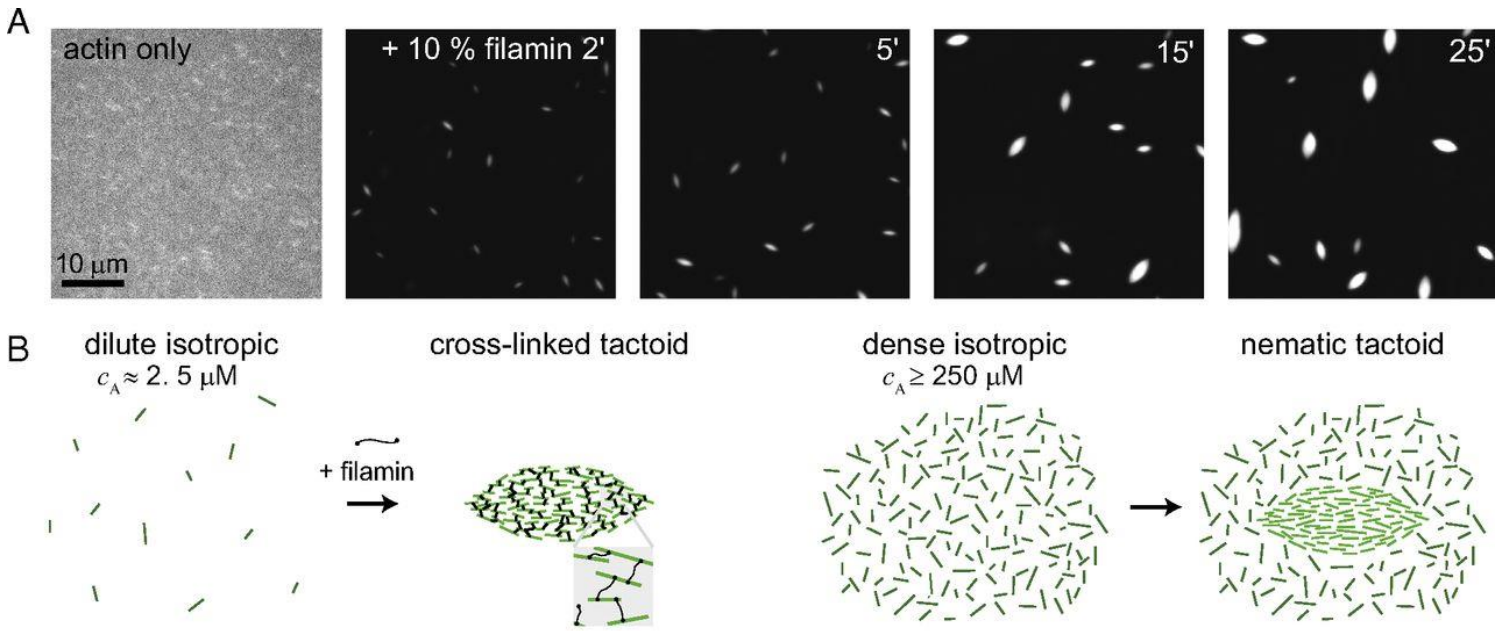
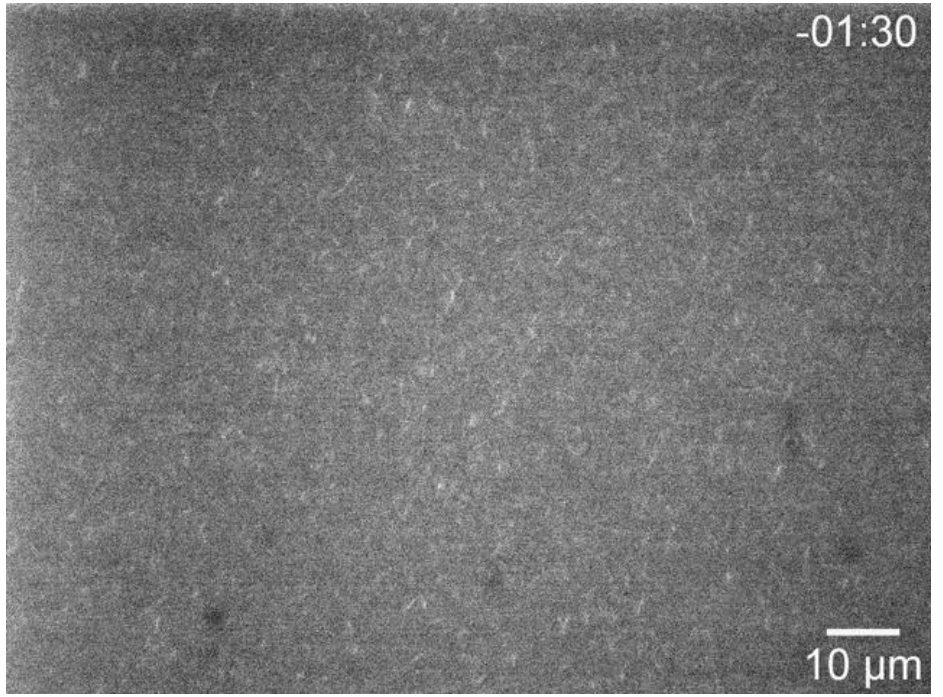
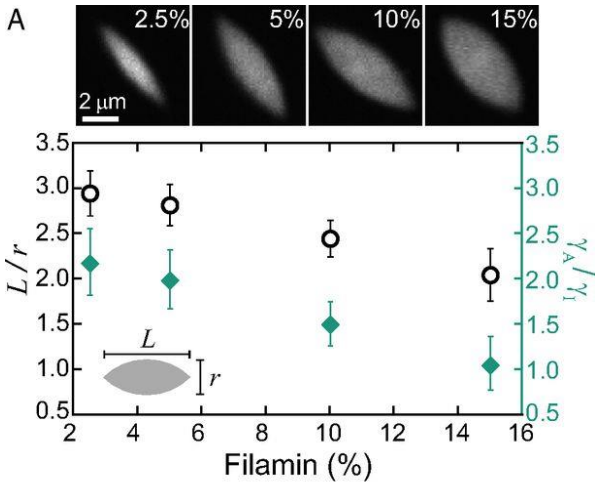
Liquid behavior of cross-linked actin bundles

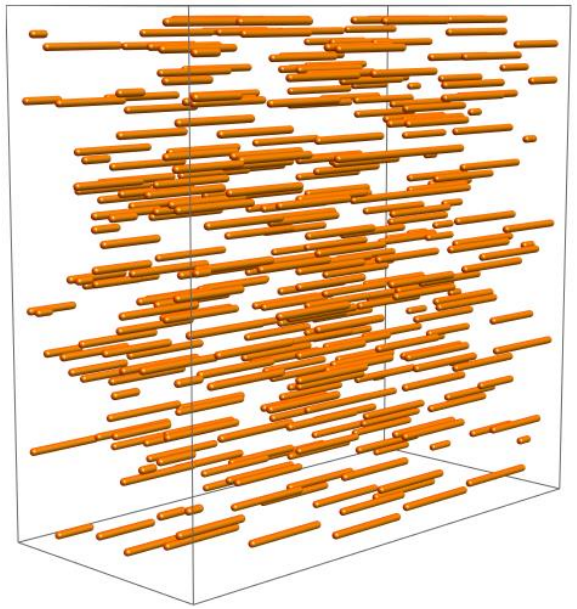


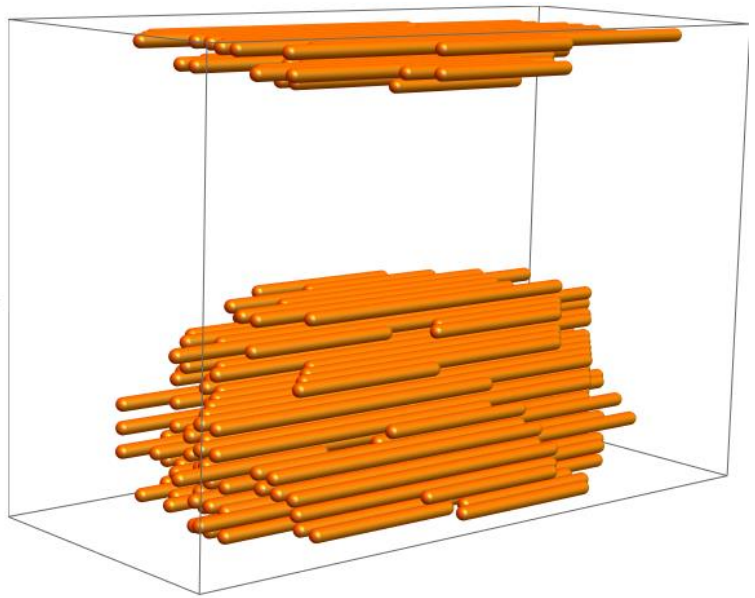


Weirich et.al PNAS 114(9) (**2017**)

How do the clusters turn into football shape?





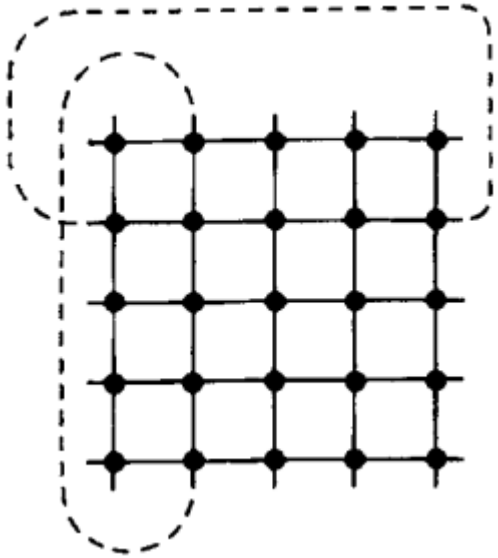


Weirich et.al PNAS 114(9) (**2017**)

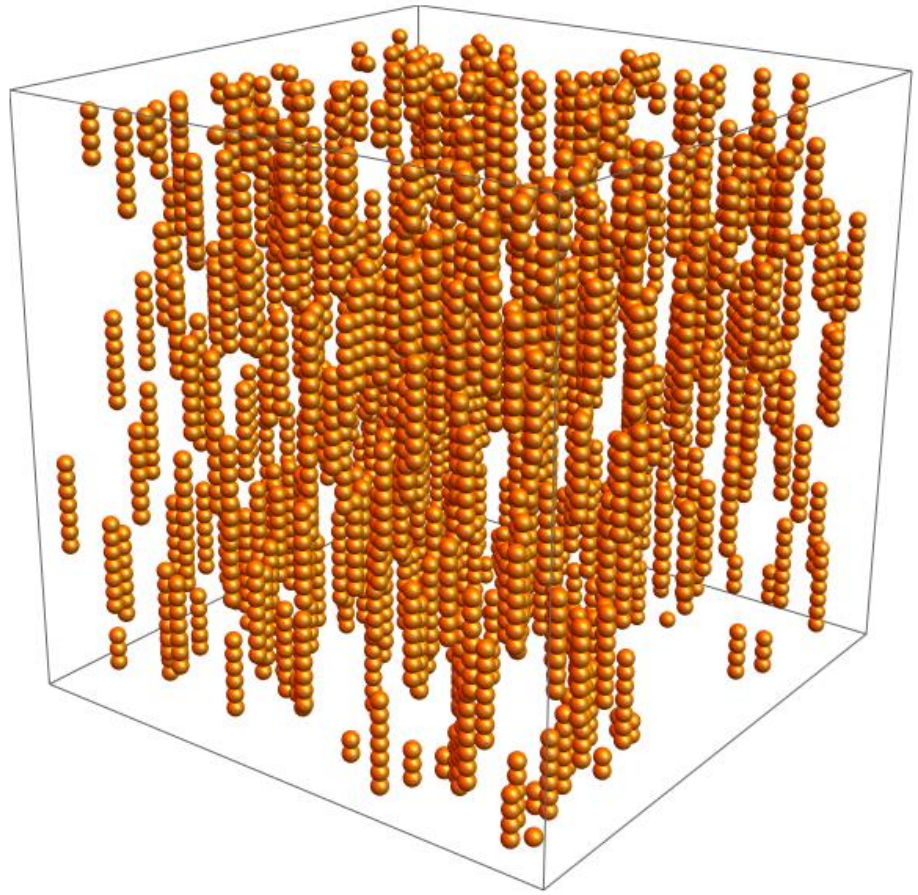
Metropolis Monte Carlo Algorithm

(*single proposed move*)

1) choose a random protein,
2) propose a new position,
3) calculate the energy difference ΔE, if the new position were to change,
4) generate a random number $0 < r < 1$,
5) if $r < \exp(-\Delta E)$, accept the new position, else repeat from step 2.

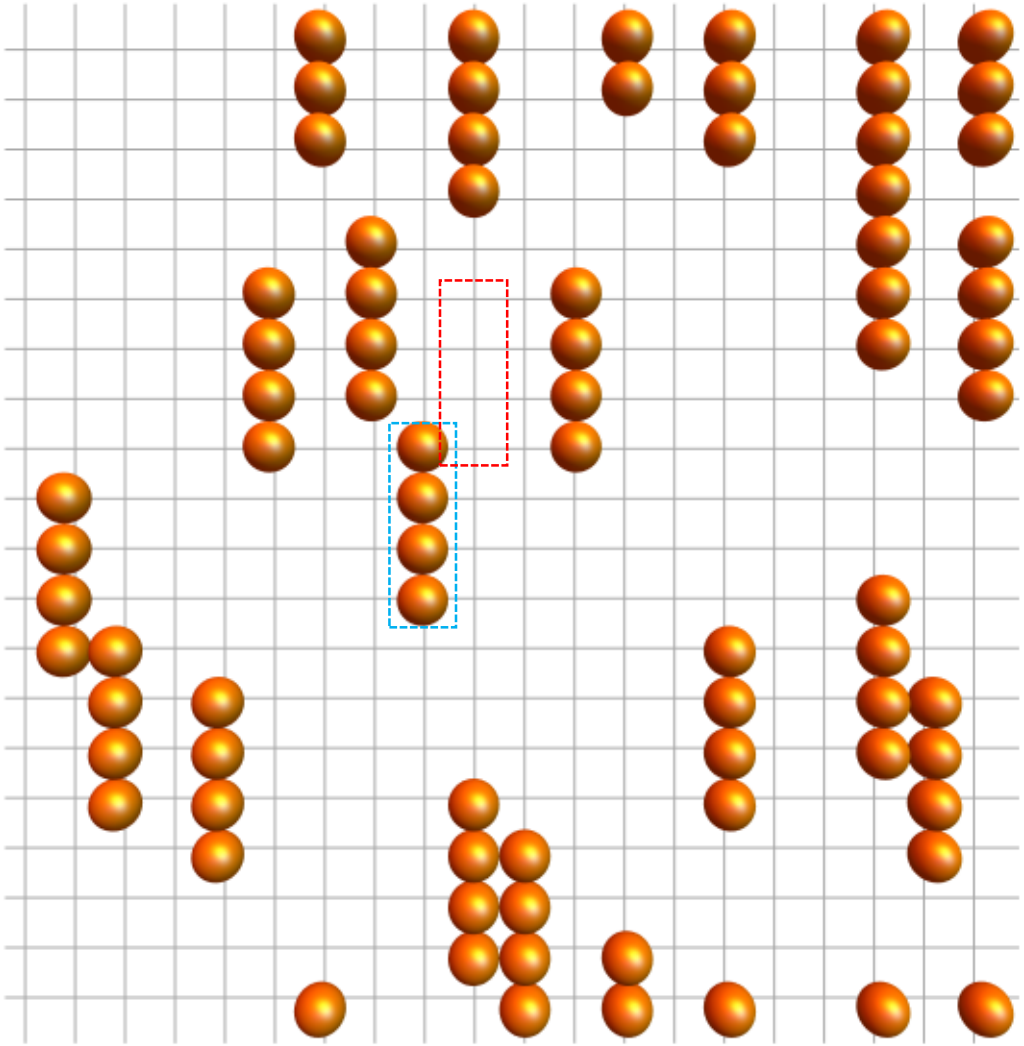


Periodic boundary condition



Initial state

Example 1

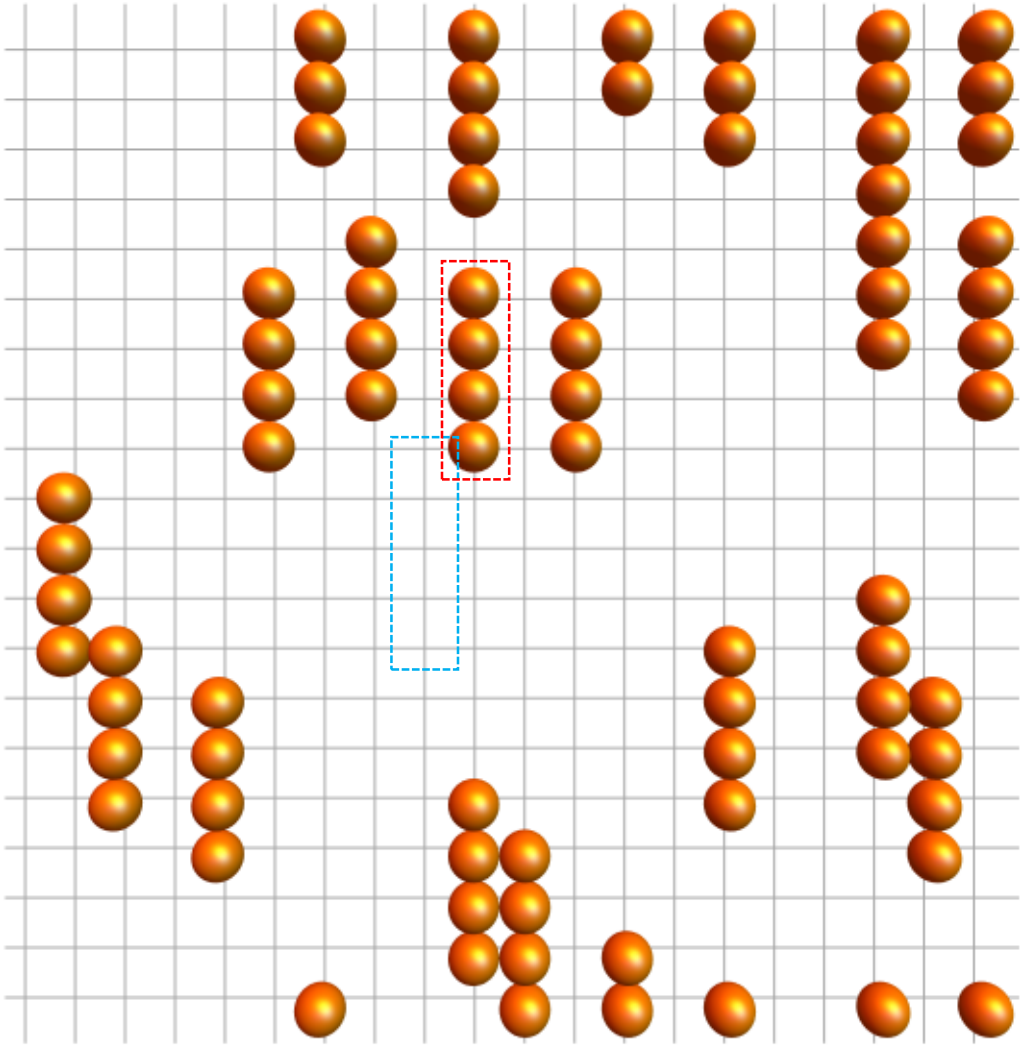


$E_1 = 0$, $E_2 = 0$

$\Delta E = 0$ ($e^0 = 1$)

$\rightarrow$ **accept** the move

Example 1

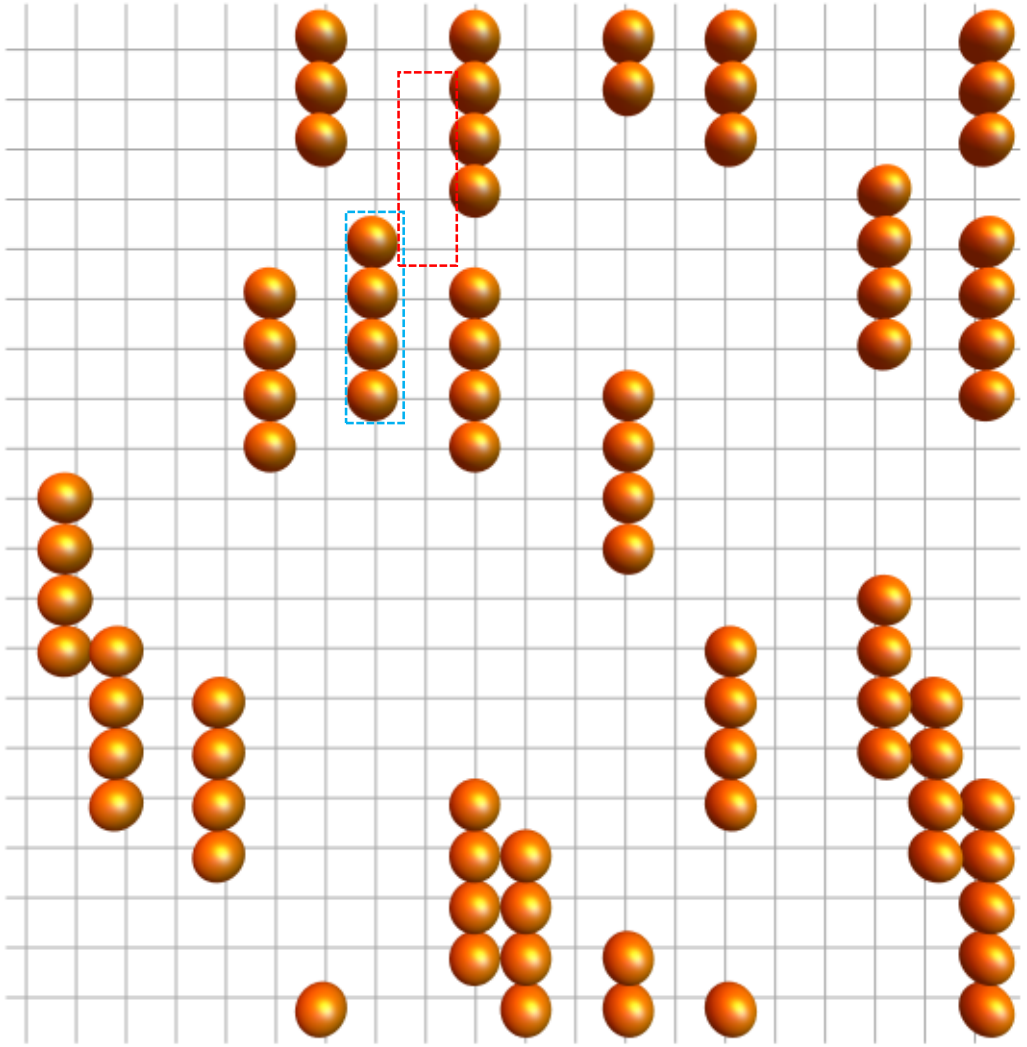


$E_1 = 0, E_2 = 0$

$\Delta E = 0$ ($e^0 = 1$)

$\rightarrow$ **accept** the move

Example 2



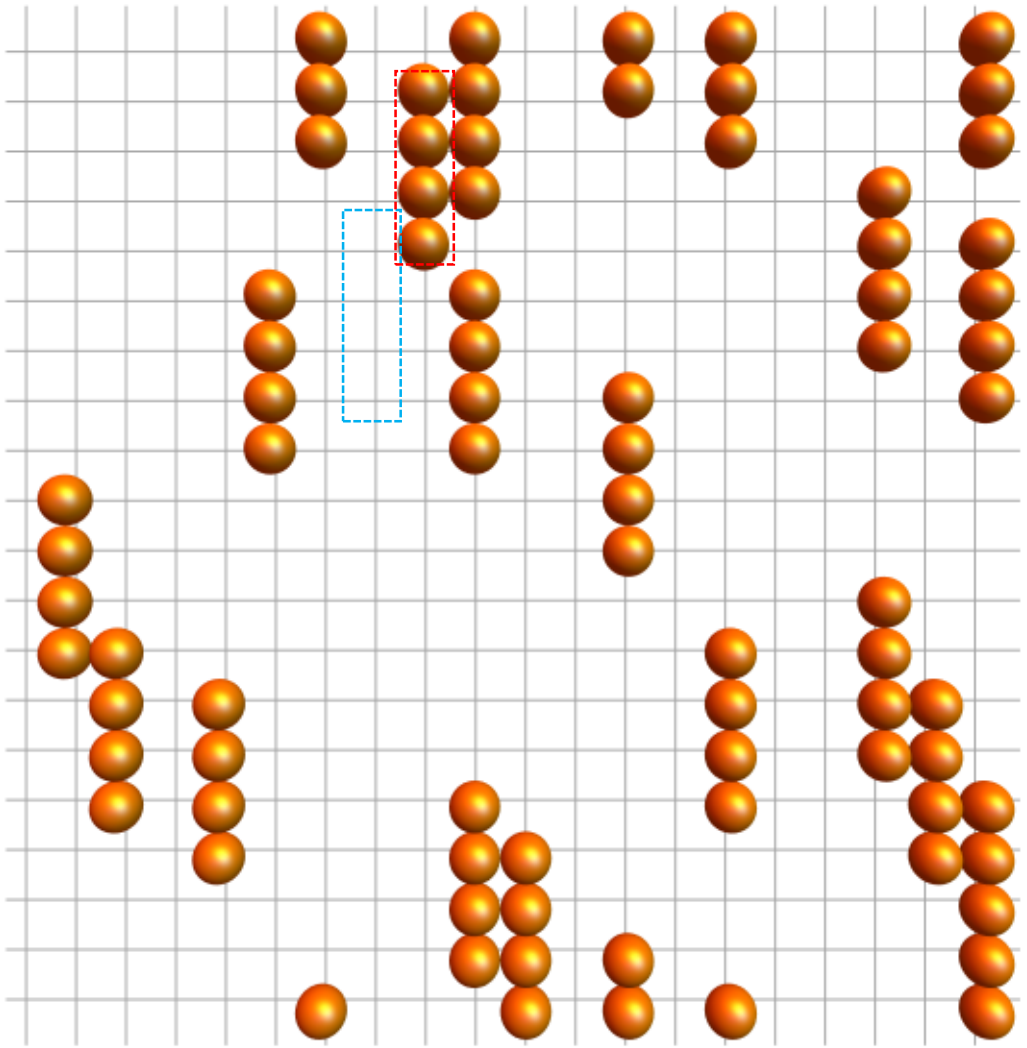
$E_1 = 0$, $E_2 = 3\varepsilon$

$\Delta E = 3\varepsilon \quad (\varepsilon = -0.6)$

$\exp(-3\varepsilon) > 1$

$\rightarrow$ **accept** the move

Example 2



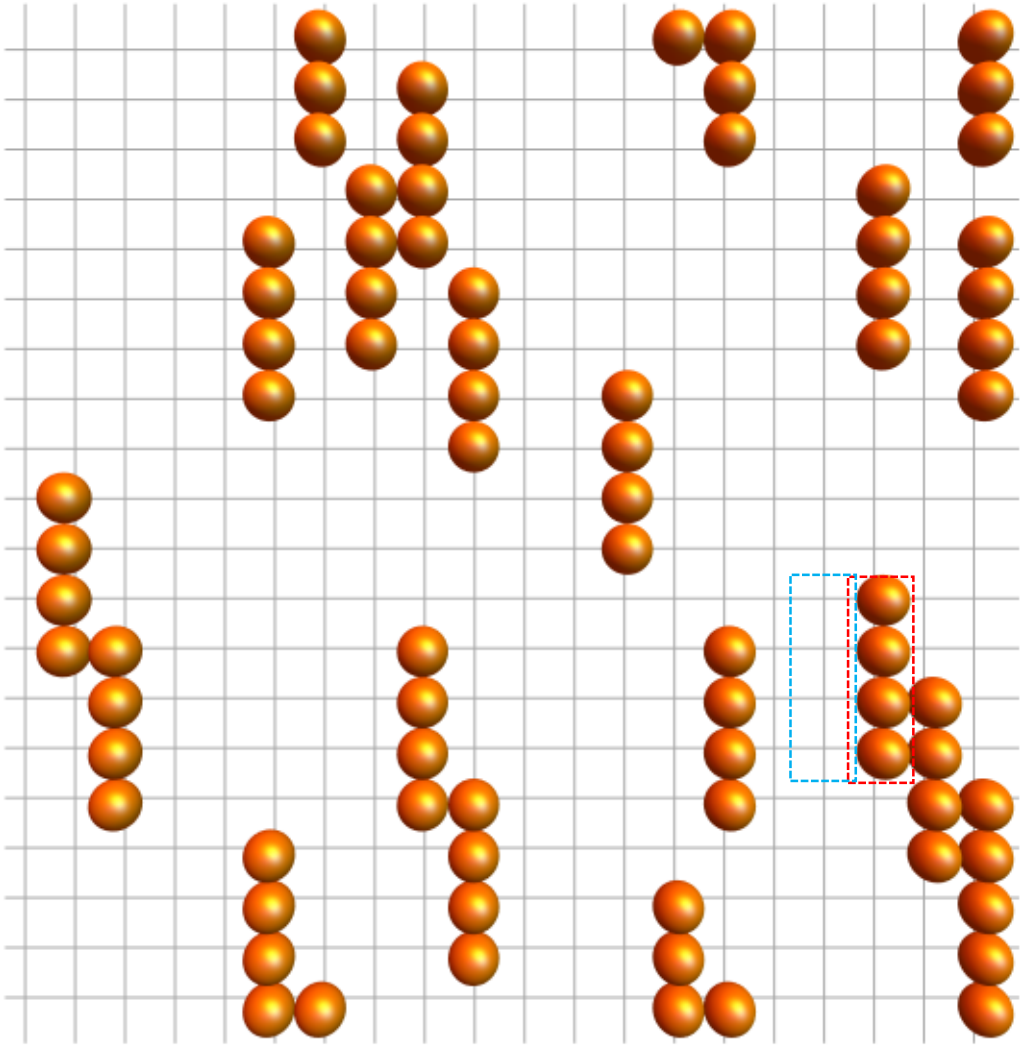
$E_1 = 0$, $E_2 = 3\varepsilon$

$\Delta E = 3\varepsilon \quad (\varepsilon = -0.6)$

$\exp(-3\varepsilon) > 1$

$\rightarrow$ **accept** the move

Example 3



$E_1 = 2\varepsilon$, $E_2 = 0$

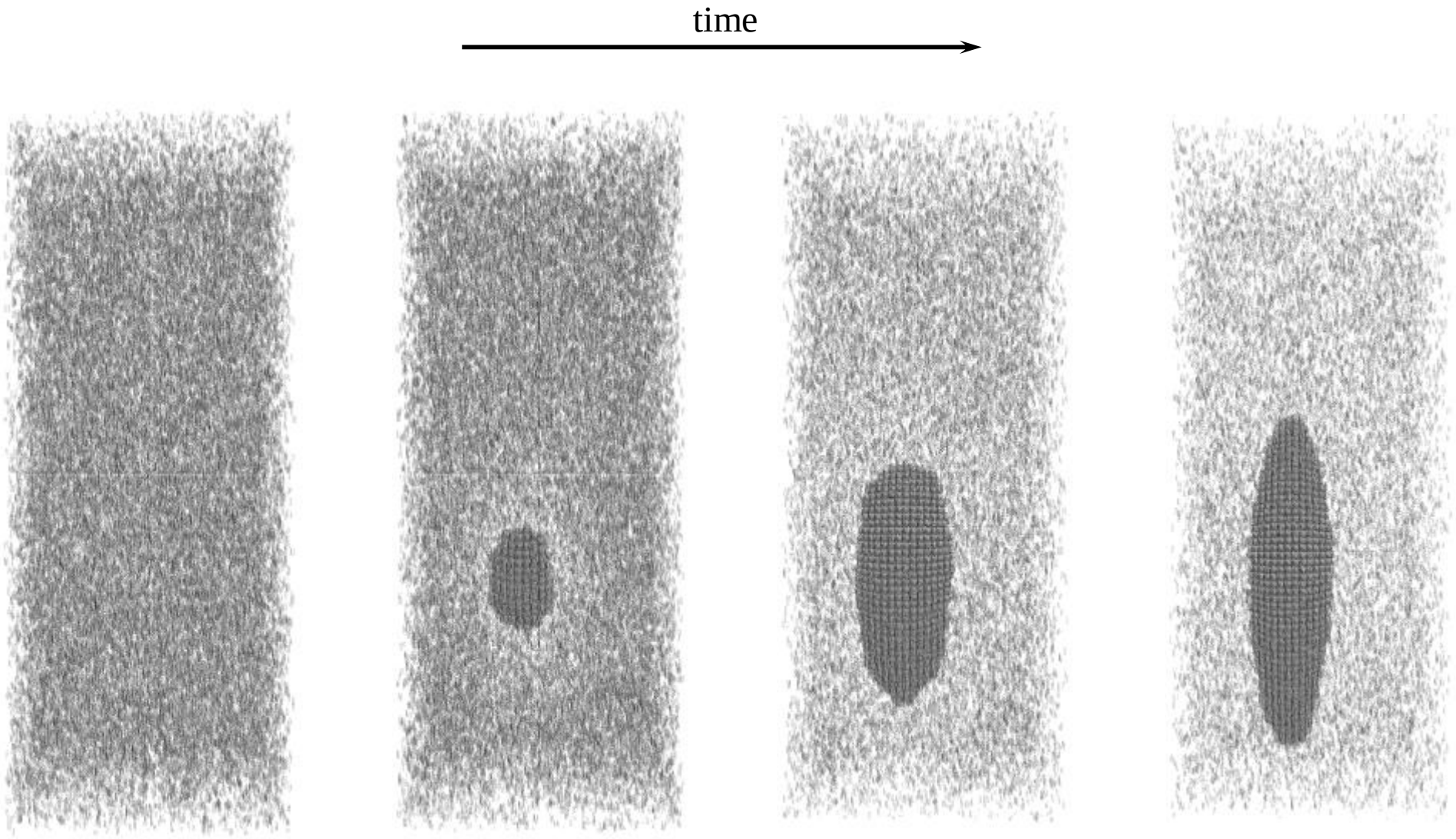
$\Delta E = -2\varepsilon \quad (\varepsilon = -0.6)$

$r = 0.478532$

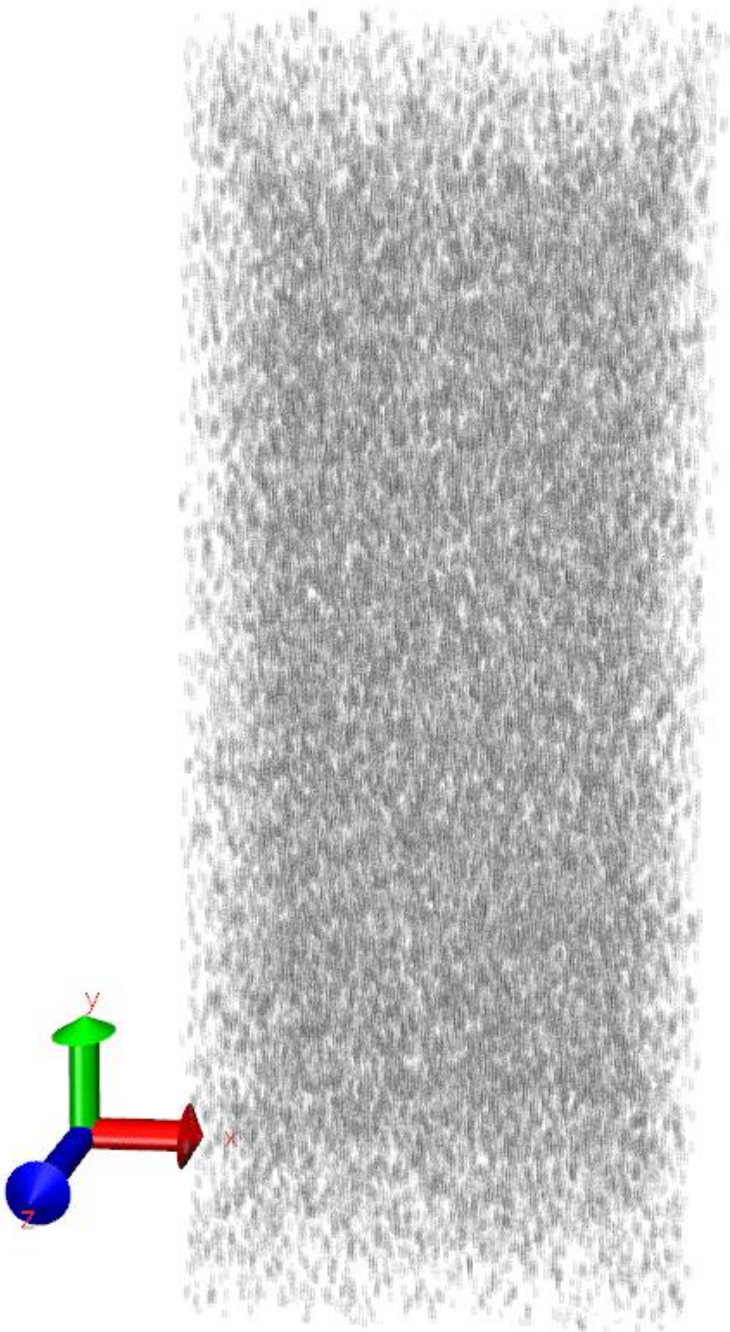
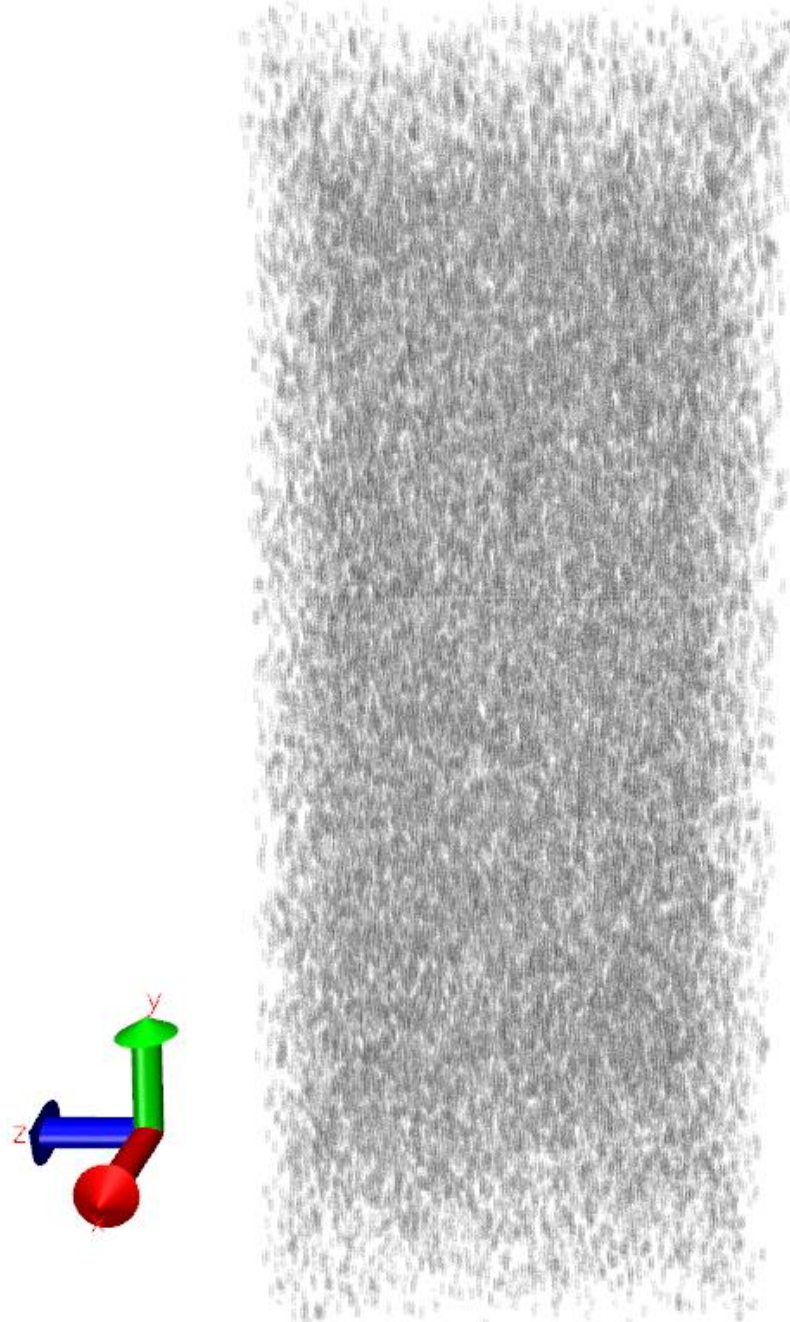
$r > \exp(2\varepsilon) = 0.301194$

$\rightarrow$ **reject** the move

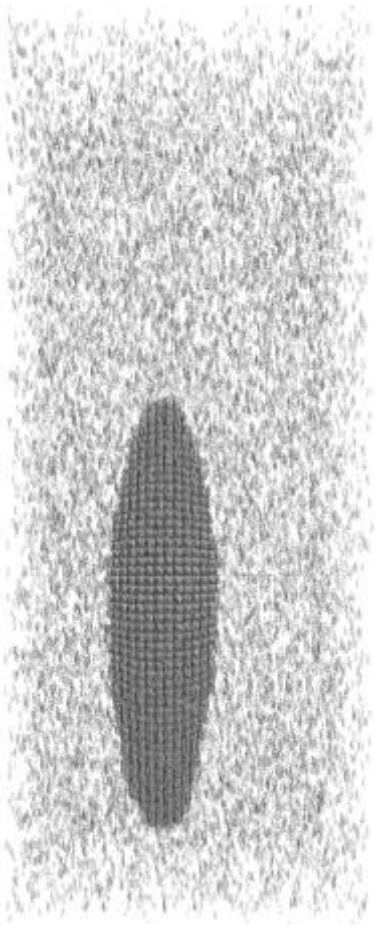
Cluster shape as a function of time



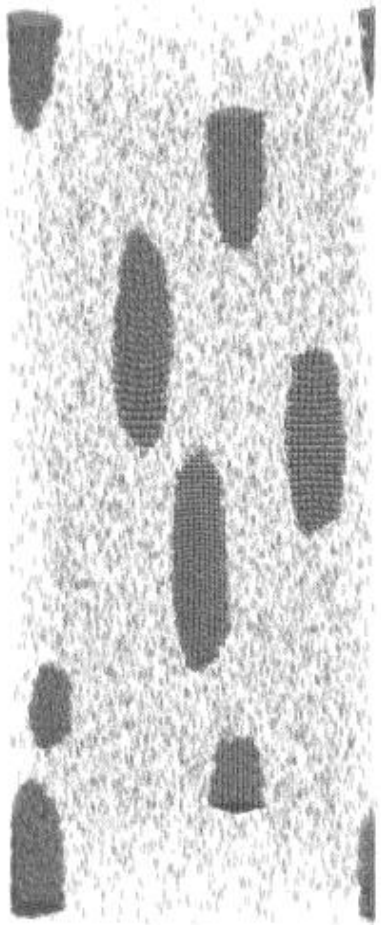
Box size 200x500x200


y
x
z

y
z

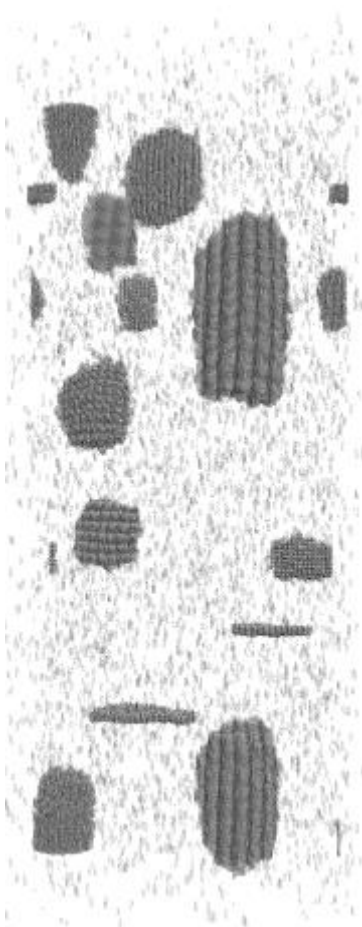
bond energy



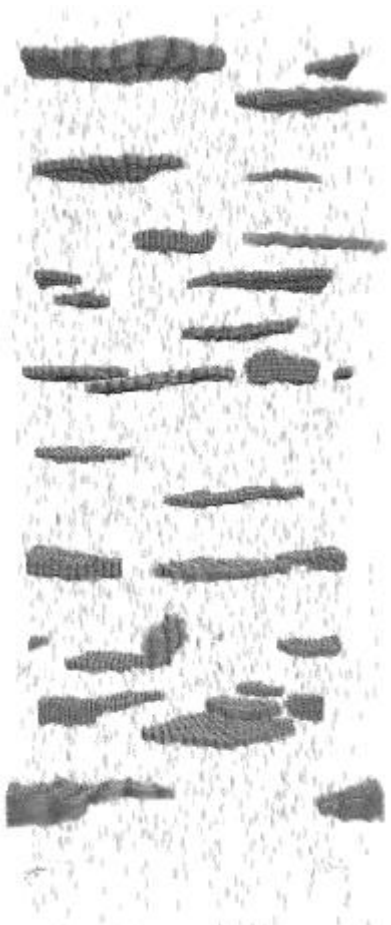
0.6



0.7



0.8



0.9

parameters

Bond energy, total number of proteins, protein length

```
icpc -std=c++11 -O3 MMC3D.cpp -o MMC3D

./MMC3D --bondEn 0.6 --iterations 3000000000 --split 1000000000 --blocks
20250 --length 5 --runId 1
```

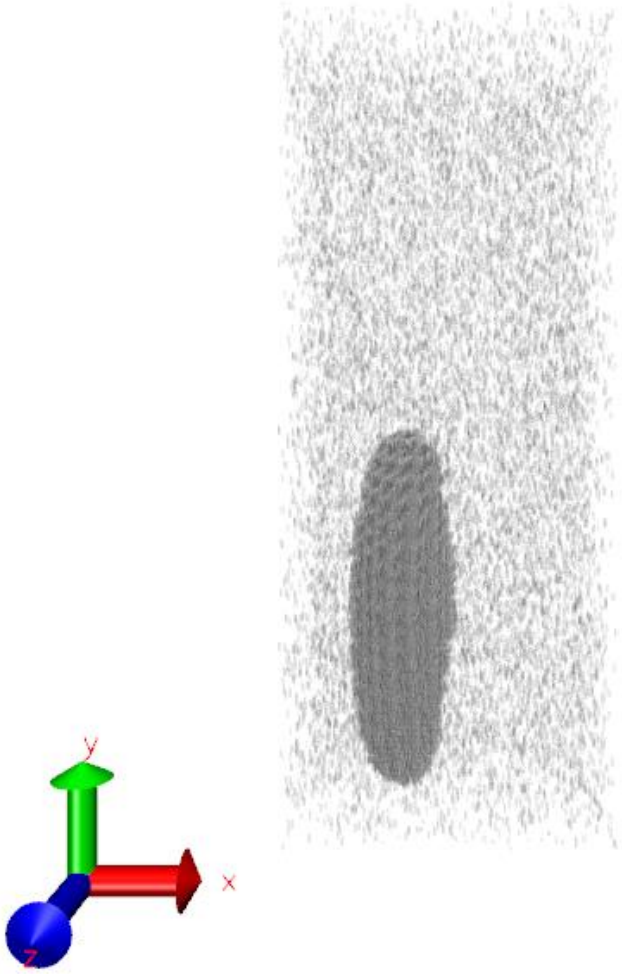
`--bondEn:` interaction energy
`--iterations:` total number of accepted moves
`--splits:` frequency of saving files
`--blocks:` total blocks in simulation box
`--length:` total units in a protein
`--runId:` number for submitting jobs in array.

output

3D array stores all proteins labeled from 1 to N
2D array stores coordinates in 3D array

1 x_1 y_1 z_1
2 x_2 y_2 z_2
3 x_3 y_3 z_3 …

Trajectory file in format .xyz for visualization



Ideas to parallelize the code?

Single proposed move:
- Use multithreading to divide the tasks?

Multiple proposed moves?
- The system may be decomposed into different parts and each processor may be assigned to work on a different part of the system.

Parallel Metropolis-Hasting algorithm?