How to code a simulation with LAMMPS

Ella Carlander and Michael Bolish
Bucknell University

Motivation and Background: *Computer Simulations*

- No “impurities”
- Well-defined model
- Molecular Dynamics (MD)
  - Atoms/Particles
  - Potential → Forces → Predict motion

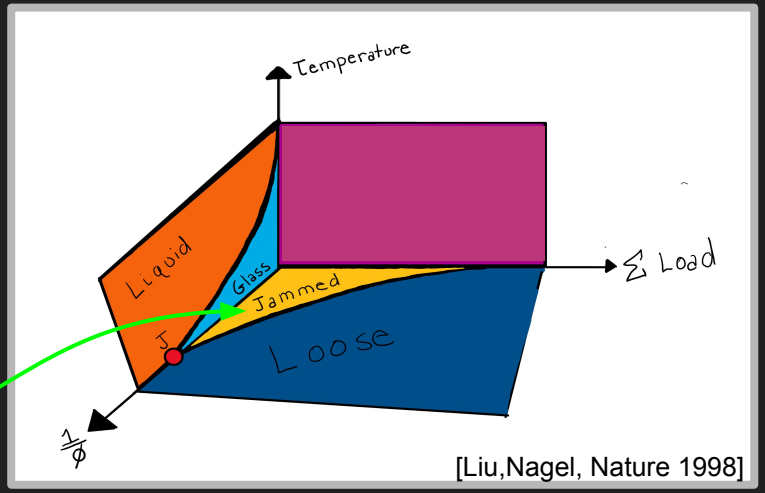
- *Address questions not accessible by theory or experiments*

Motivation and Background: *Granular Media*

Granular Media: collection of distinct macroscopic particles

Lacks theoretical foundation

This plane is where granular media exists



[Liu,Nagel, Nature 1998]

What is LAMMPS?

“**L**arge-scale **A**tomic/**M**olecular **M**assively **P**arallel **S**imulator”

- Classical Molecular Dynamics code
  - Open-source
- Provides Potential
  - Solid-state, soft-matter, coarse-grained
  - Atomic, meso, continuum

For Molecular Dynamics (MD) Simulations: *LAMMPS is very powerful*

Particle interactions in Python:

```
# AA Interactions
  for i in range(Na-1):
    xi=x[i]
    yi=y[i]
    for j in range(i+1,Na):
      xij=xi-x[j]
      yij=yi-y[j]
      #minimum image convention
      if xij > Ldiv2: xij -= L
      elif xij < - Ldiv2: xij  += L
      if yij > Ldiv2: yij -= L
      elif yij < - Ldiv2: yij  += L

      rijto2 = xij*xij + yij*yij
      rij = rijto2**0.5
      if(rij < rcutAA):
        Vtot += (1.0 - rij/rcutAA)**2
```

The same interactions in LAMMPS:

```
pair_style  dpd $T ${rmax} 11223
pair_coeff  1 1  ${Aaa} ${gamma} ${rcutAA}
```

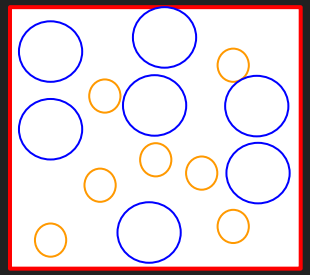
To achieve the same simulation:
Python Script → 525 lines
LAMMPS Script → 50 lines

Outline

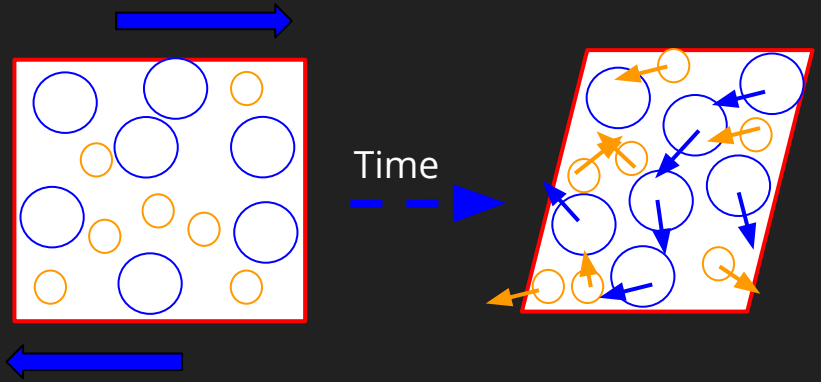
- General Plan
  - Setting up system
  - Running a Simulation
  - Collecting Results
  - More complex features
- Our System
- Preliminary Results

System

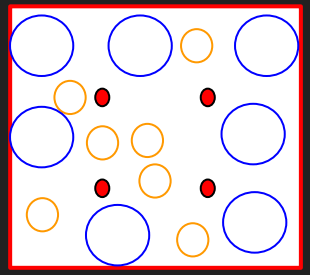
2-d box



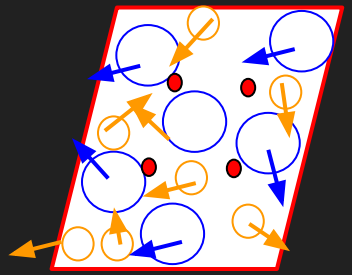
2-d box, **two types of particles**



2-d sheared box over time



2-d, **fixed pin box**, two types of particles



2-d, pinned, **sheared box**, two types of particles

Setting Up A System

- Simulation Region
  - Dimensions/Shape
  - Temperature
  - Units
- Atom Parameters
- Initializing Atoms to System
- Potential/Forces
- Periodic Boundary Conditions

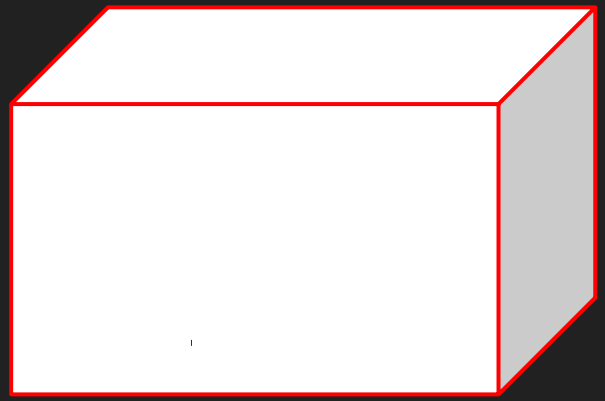
Our Simulation

General Simulation

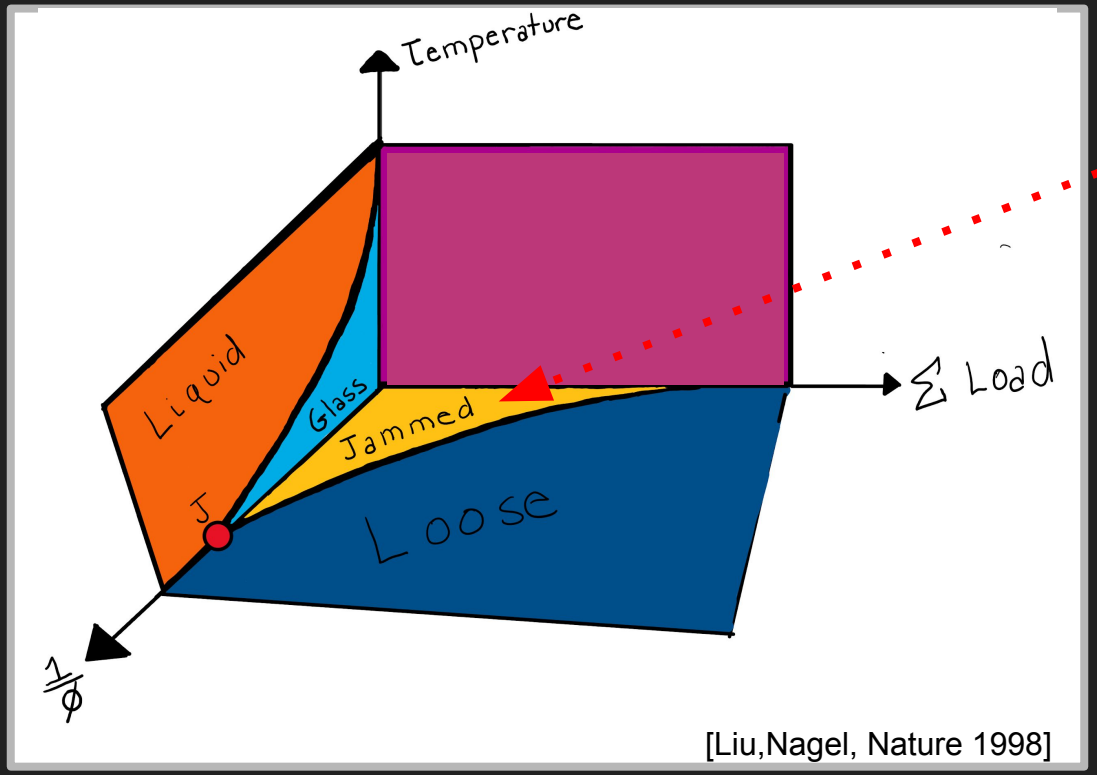
Setting up your Simulation Region: *Dimension/Shape*

2D Box Our System

3D Box General System



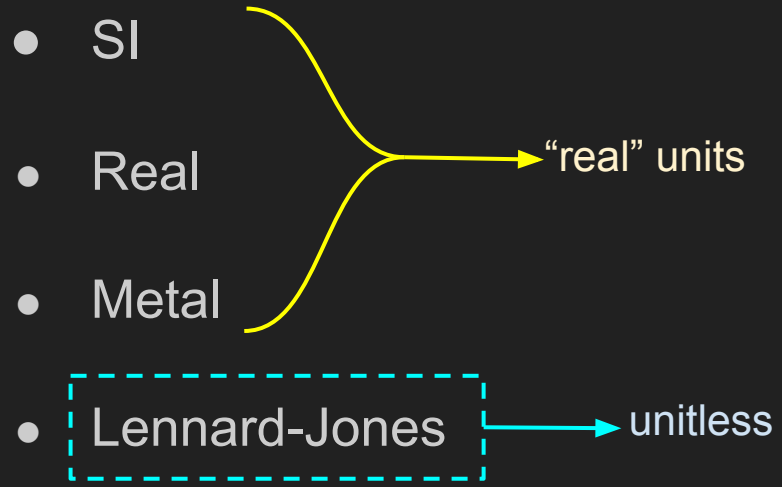
Setting up your Simulation Region: *Temperature*



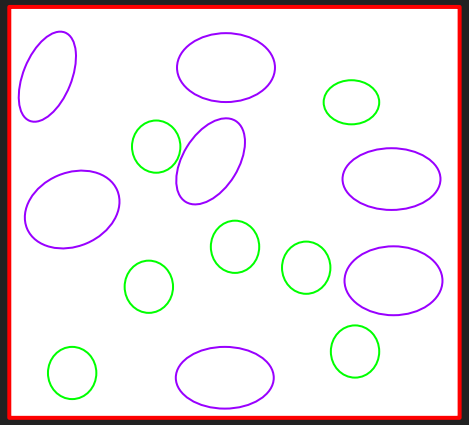
[Liu,Nagel, Nature 1998]

Focused on low temperatures for jamming transition

Setting up your Simulation Region: *Defining Units*

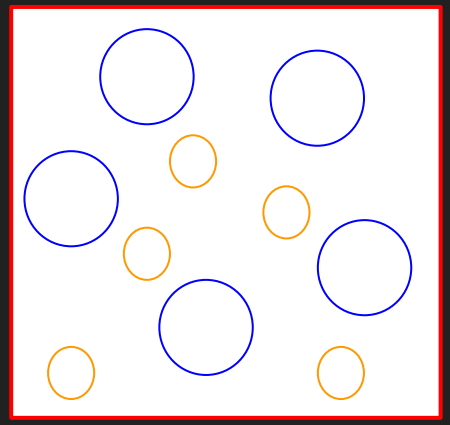


Atom Parameters: *Type, Mass, N, and Size*



General System

Style = ?
Mass = ?
N = ?
Size = ?



Our System

Style = Atomic
Type A and B
Mass = 1.0
N = 1000 (50:50)
Size = 1:1.4

Box Length

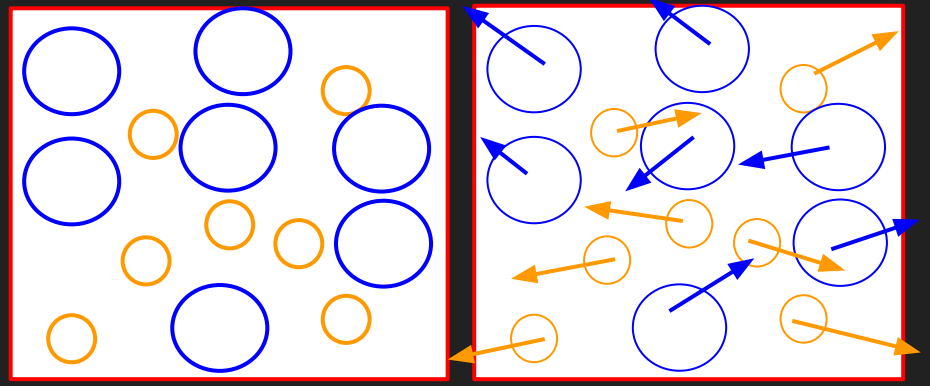
“Packing fraction”

Telling the computer what you plan to put into your simulation region

Initializing Atoms into System

From Scratch: Our System

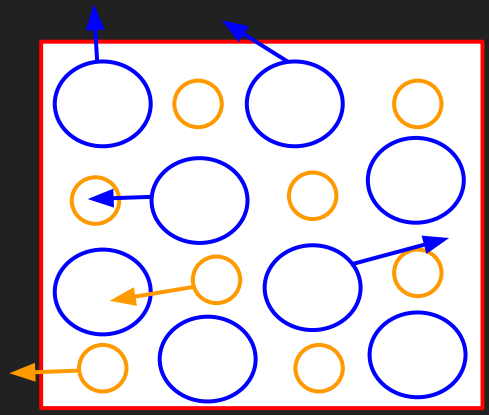
Generate Random Position and Velocity for each Particle



- Lammps command
  1. Random Positions
  2. Random Velocities

From File

Initialize Specific Position and Velocity for each Particle



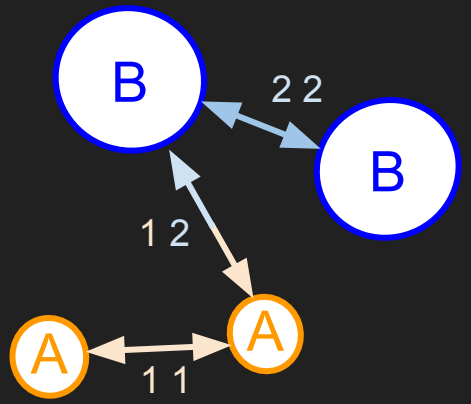
- Data File read in
  1. Random positions and Velocities at same time

Particle Interactions

particles *in* simulation → tell computer how to interact

Goal: Use model to govern forces between particles

Simulation Technique: Pair Coefficients



pair_style lets us choose the desired model for the potential

```
pair_style  dpd $T ${rmax} 11223
pair_coeff  1 1  ${Aaa} ${gamma} ${rcutAA}
```

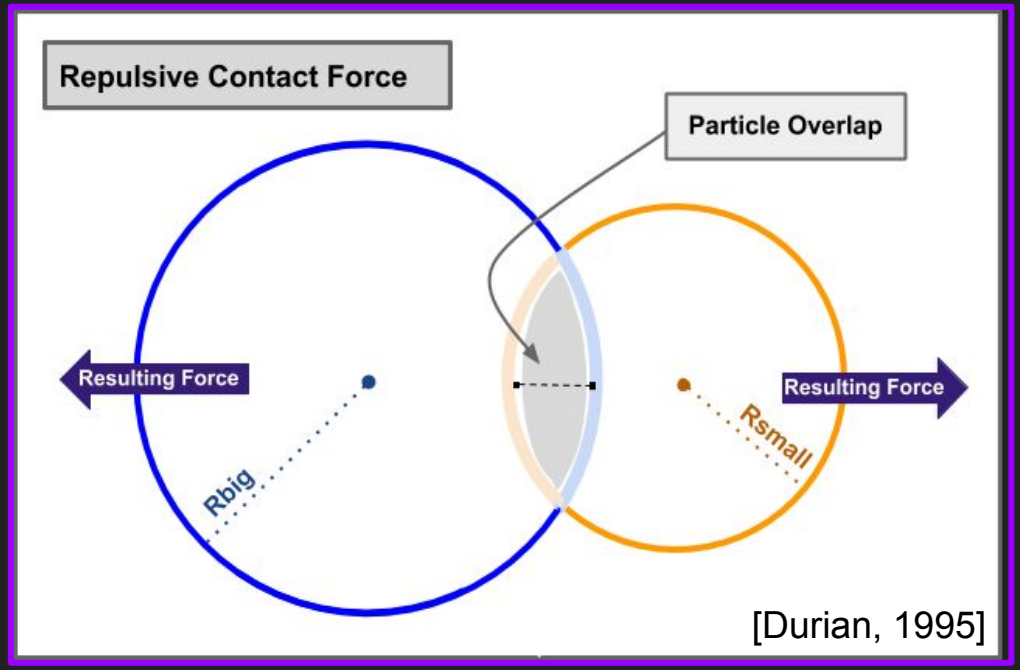
Define necessary parameters (A, gamma, rcut) with *pair_coeff*
Necessary for each possible combination of particle types

Particle Interactions: *Our Simulation*

Purely Repulsive Harmonic Potential

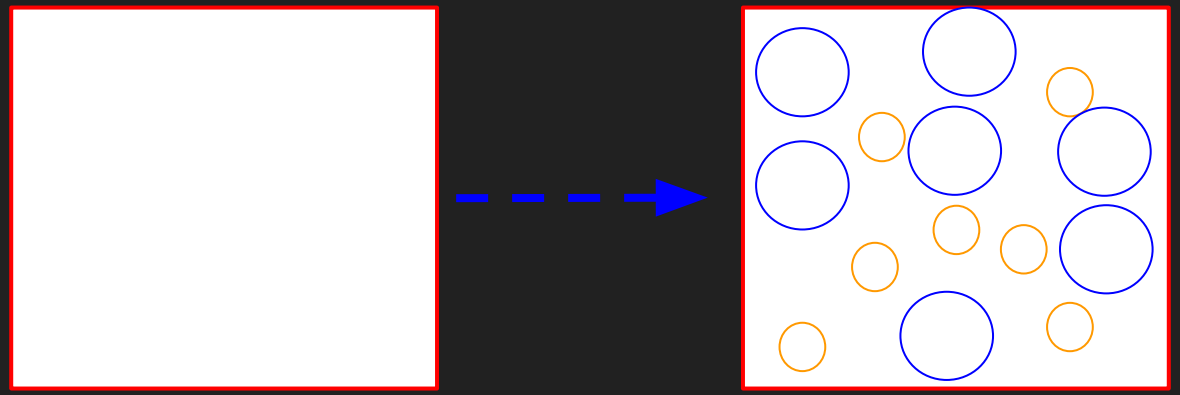
$$\vec{f} = (F^C + F^D + F^R)\hat{r_{ij}} \qquad r < r_c$$
$$F^C = Aw(r)$$
$$F^D = -\gamma w^2(r)(\hat{r_{ij}} \bullet \vec{v_{ij}})$$
$$F^R = \sigma w(r)\alpha(\Delta t)^{-1/2}$$
$$w(r) = 1 - r/r_c$$

```
pair_style  dpd $T ${rmax} 11223
pair_coeff  1 1  ${Aaa} ${gamma} ${rcutAA}
```



Forces only arise when the particles *overlap*.

System: Mini Summary

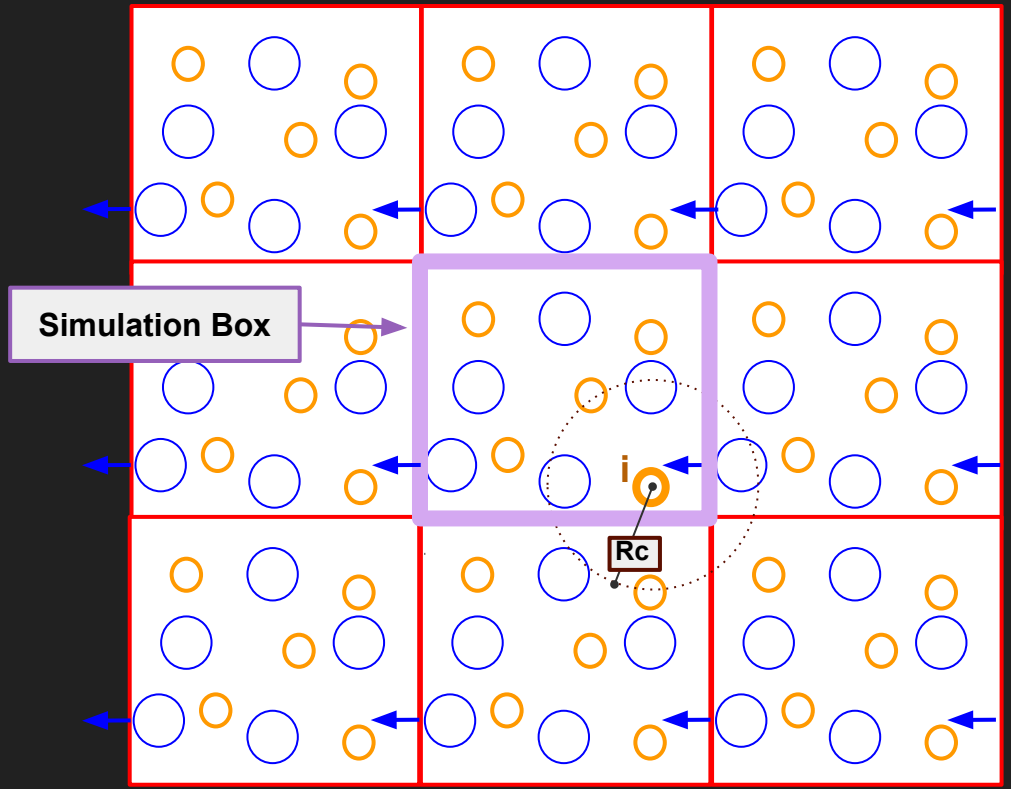


2-d box

2-d box, **two types of particles, repulsive contact forces**

Periodic Boundary Conditions

- Surrounded by images of itself
  - Many-particle simulation
  - Desired physics
- Rc=Ri+Rj
  - “Minimum image convention”
  - Particle’s neighbors
- Atoms cross boundary, re-enter other side



Periodic Boundary Conditions: *Short Video*

Running a Simulation

- Numerical Integration
- Run
- Minimization

Numerical Integration

General System

Initialization:
Set for each particle *i*
Position $r_i(t_O)$
Velocity $v_i(t_O)$

Loop over Time Steps

Time Step:
Update for each particle *i*
Position $r_i(t_O) \rightarrow r_i(t+\Delta t)$
Velocity $v_i(t_O) \rightarrow v_i(t+\Delta t)$

NVE

Initialization:
For all *i* set $r_i(t_O)$, $v_i(t_O)$

For all *i* set $a_i(t_O)$ (use min image)

Loop over Time Steps

Time Step: Our System

Update Positions

For all i:

$$r_i(t+\Delta t) \rightarrow r_i(t) + v_i(t)\Delta t + 0.5a_i(t)(\Delta t)^2$$

Apply Periodic Boundary Conditions

Update Velocities

For all *i*:

$$v_i^{mid} \rightarrow v_i(t) + 0.5a_i(t)(\Delta t)$$

For all *i*:

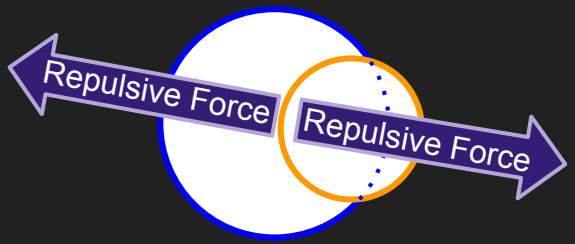
Determine $a_i(t+\Delta t)$ (use min image)

For all *i*:

$$v_i(t+\Delta t) = v_i^{mid} + 0.5a_i(t+\Delta t)\,\Delta t$$

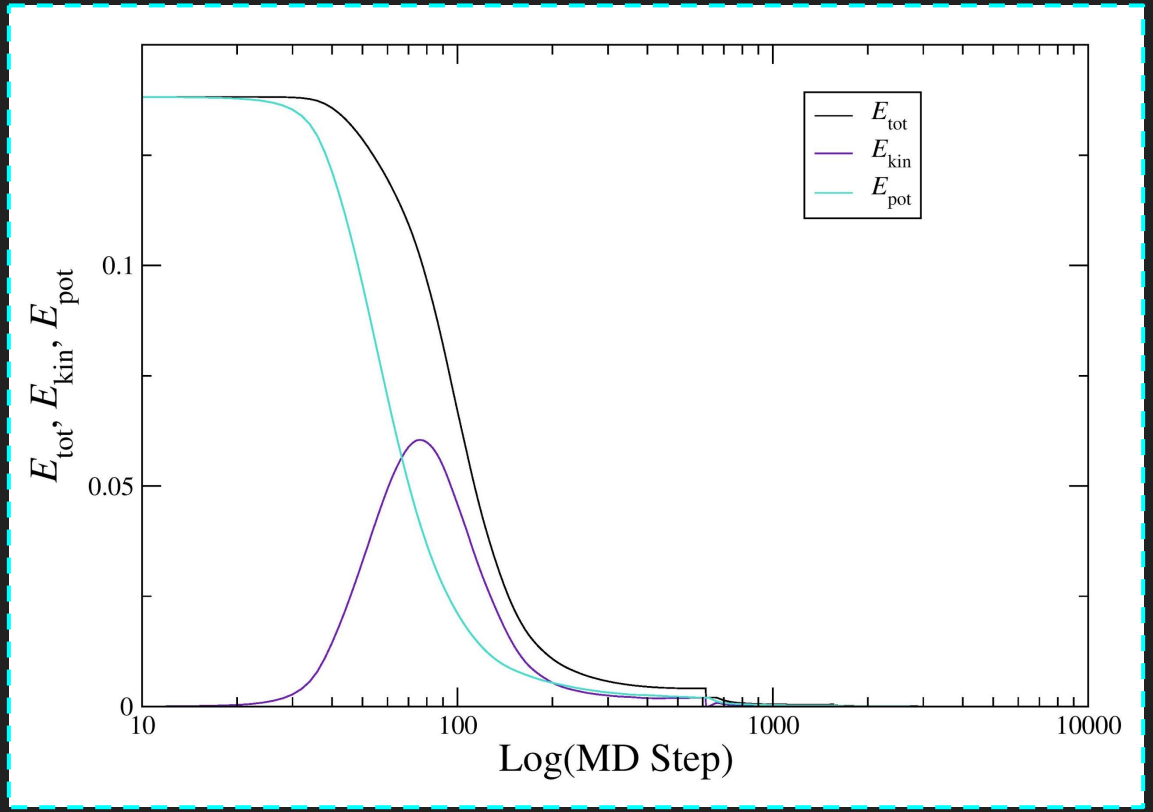
Minimization

- Random positions → too much overlap



- Minimize total potential energy

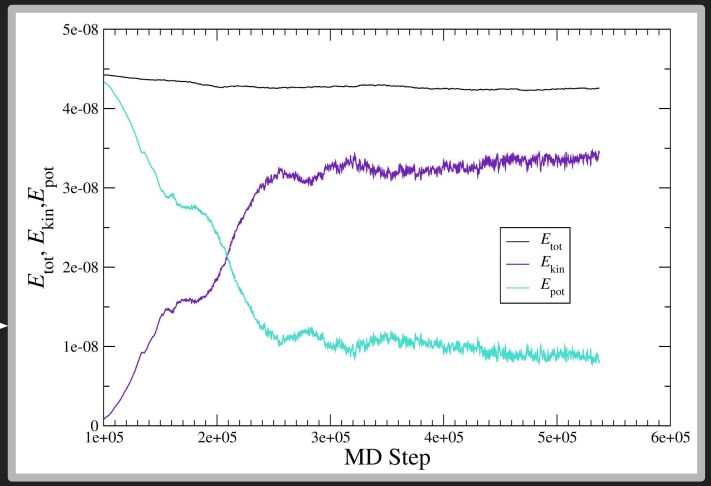
Energy vs. Log(MD Step)



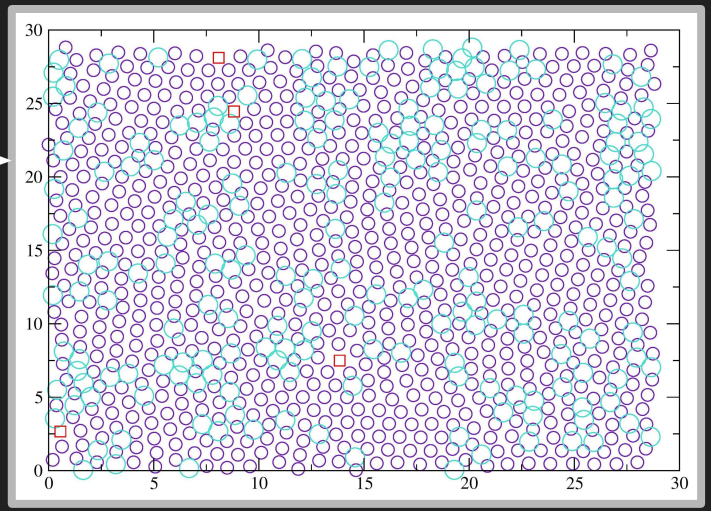
Collecting Results

Two Approaches:

1. Over time
   a. Whole system
   b. One particle

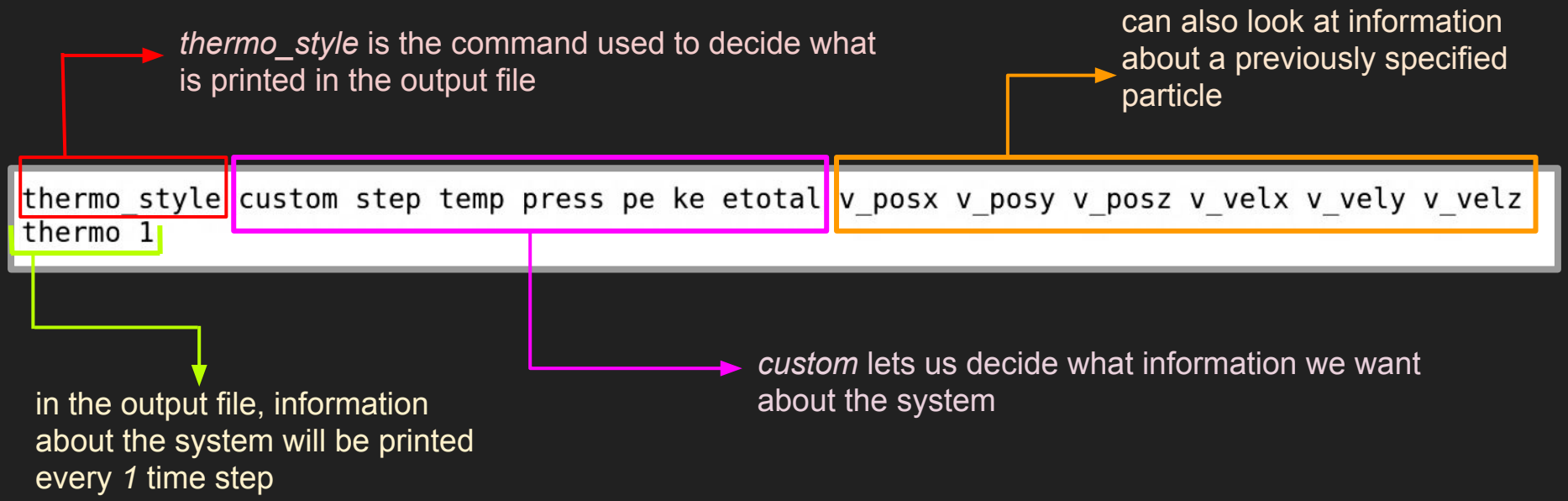


2. Snapshot (moment in time)
   a. All particles



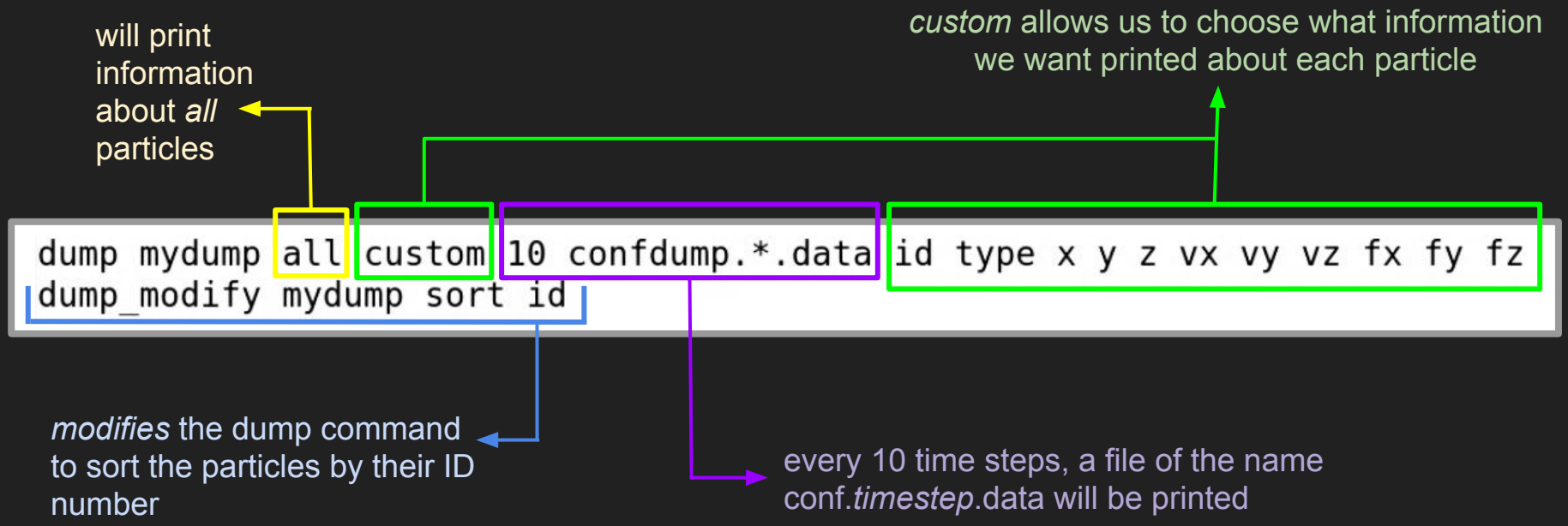
Collecting Results: *Over Time*

- Overview of the simulation results, automatically printed
- Provides information about the system *over time*



Collecting Results: *Snapshot*

- Provides a snapshot of individual particles at *one time-step*

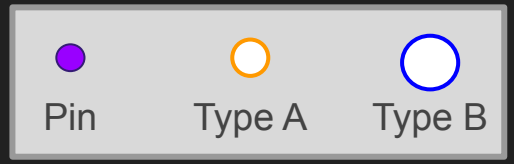


More Complex Features

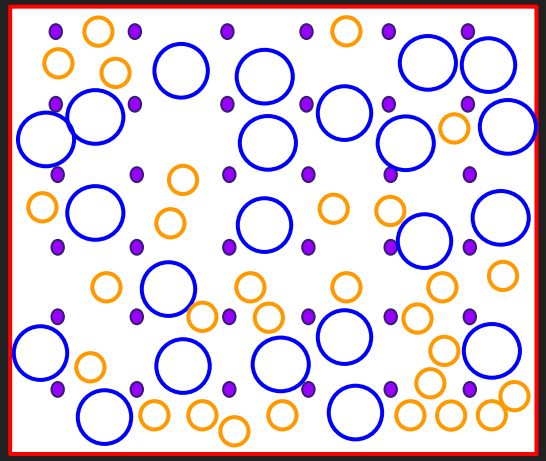
Two examples of further complexities:

1. Pins → fixing certain particles
2. Shearing → box deformation
    a. Positions
    b. Velocities

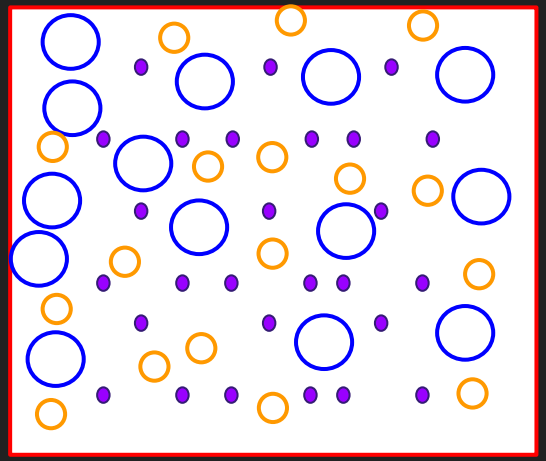
More Complex Features: *Pins*



Placed on Square Lattice

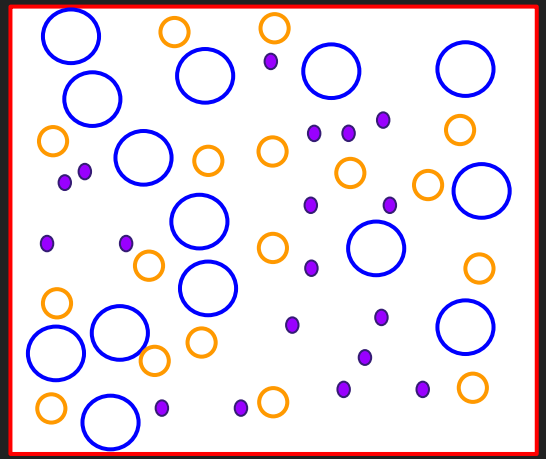


Placed on Triangular Lattice



command creates a specified type of lattice (square, triangle, hexagonal, custom) and places the pins on it

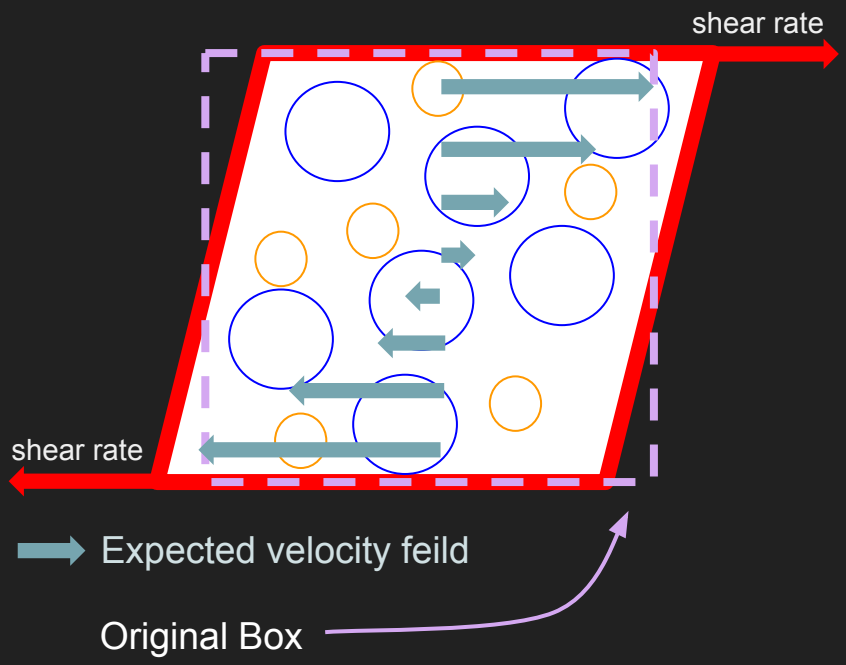
Randomly Placed



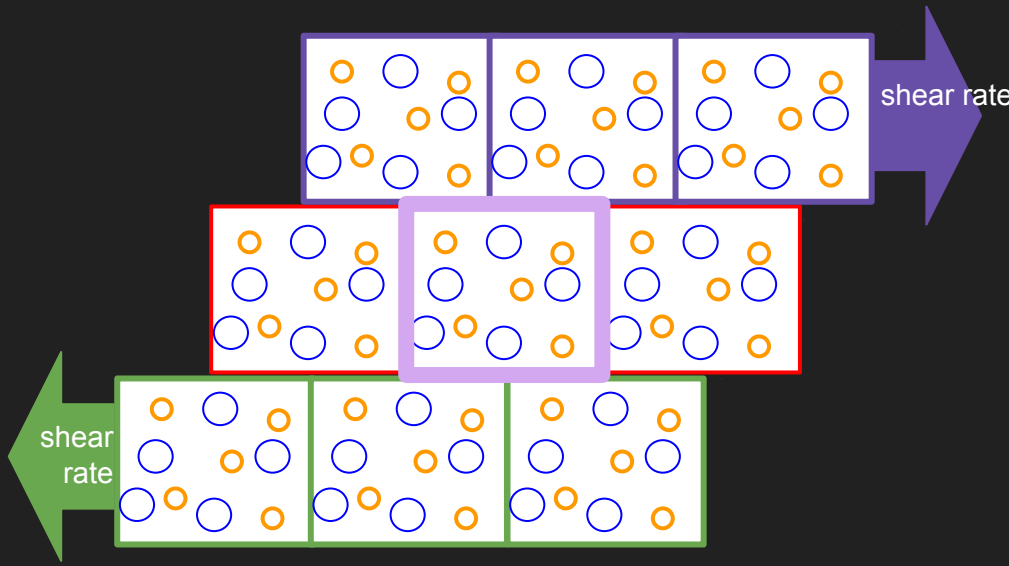
command randomly places particles

More Complex Features: *Shear*

Desired Box Deformation:



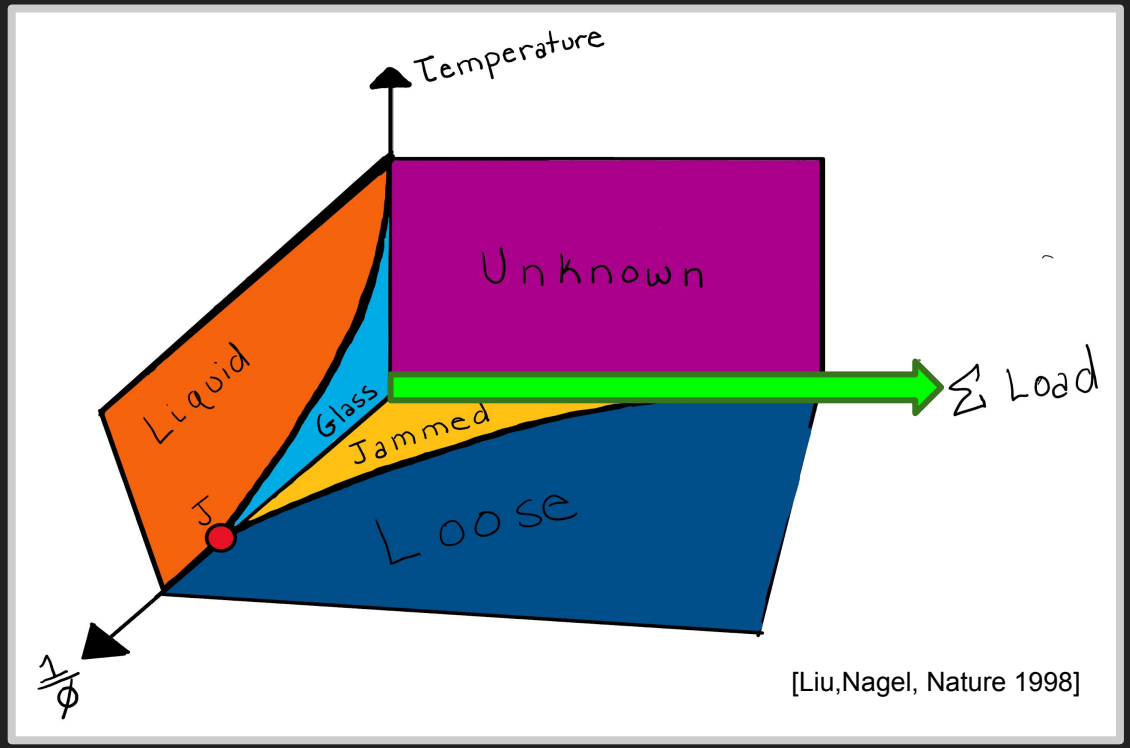
Simulation Technique:



“Lees-Edwards Boundary Conditions”

More Complex Features: *Shear*

Why Shear?

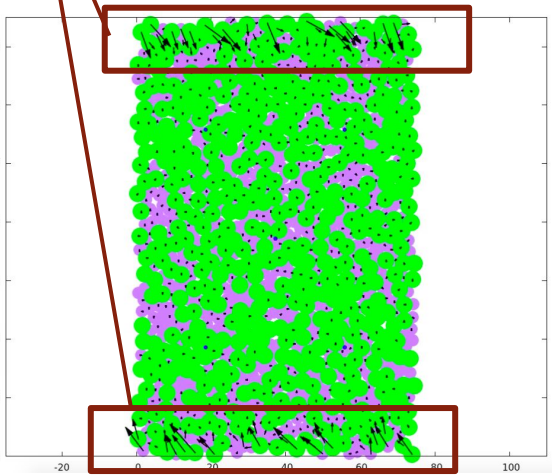


[Liu,Nagel, Nature 1998]

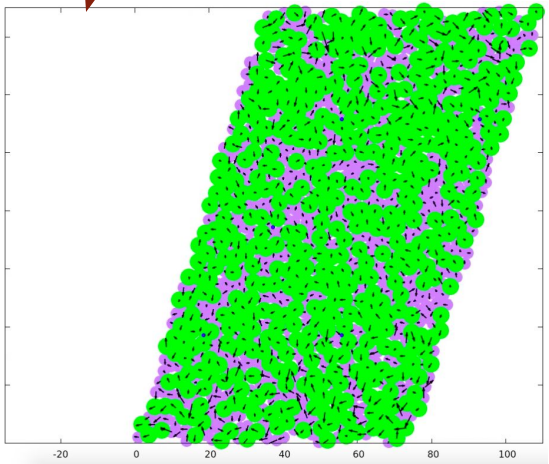
Remapped velocities of particles at boundaries

Visibly Sheared

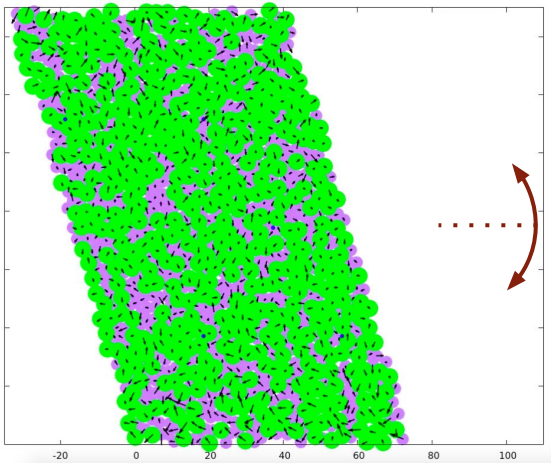
Beginning of Simulation



During Simulation



Later in Simulation

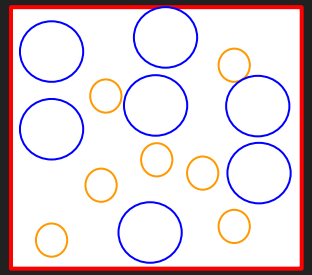


Simulation box gets flipped

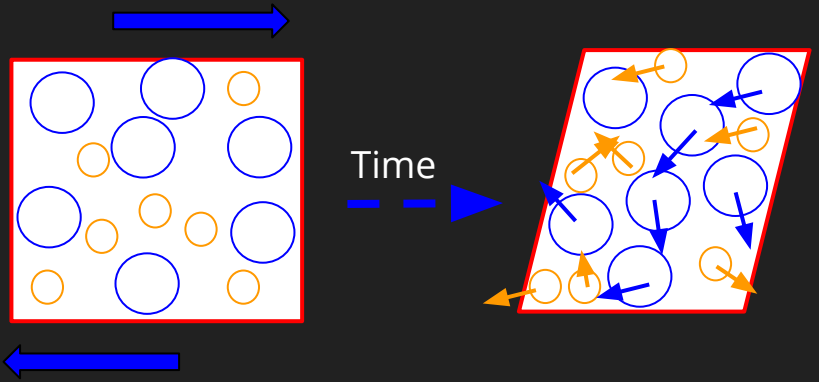
More Complex Features: *Shear Continued*

System Recap

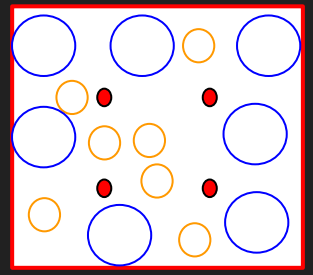
2-d box



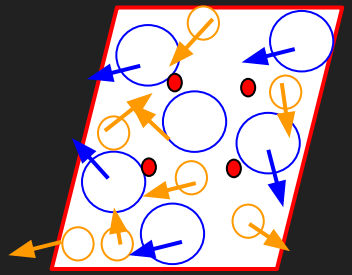
2-d box, **two types of particles**



2-d sheared box over time



2-d, **fixed pin box**, two types of particles



2-d, pinned, **sheared box**, two types of particles

Conclusion

- Computer simulations are extremely useful research tools
- For granular media, MD simulations are particularly useful
- LAMMPS is a highly optimized tool for building MD simulations

To summarize → Steps for building a simulation with LAMMPS:

1. Decide what physics you want to observe
2. Consider what type of system will allow you to observe that physics
3. Choose parameters
4. Run the simulation
5. Collect and analyze results

Acknowledgements and Questions

- Katharina Vollmayr-Lee, Amy Graves, Cacey Bester, Brian Utter
- Supported by NSF Grant DMR-1905737

Bucknell UNIVERSITY

SWARTHMORE COLLEGE