

High End Simulation in Practice: Molecular Dynamics Part 1

Harald Köstler, Severin Strobl



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT



Part 1: Overview of Particle Techniques

Where can we use them?

Atomic Scale

(Micro)meter Scale

Process Scale

Planetary/Galactic Scale

Equation of Motion

Part 2: Details of Molecular Dynamics

Rough Outline of the Algorithm

Force Models

- Splitting the Force Term

- Atomic Forces

- Granular

Initial Conditions

Boundary Conditions

Integrator

- Purpose of the Integrator

- Choosing an Integration Scheme

- Velocity Verlet Integration

Final Remarks

Part I

Overview of Particle Techniques

Outline

Where can we use them?

Atomic Scale

(Micro)meter Scale

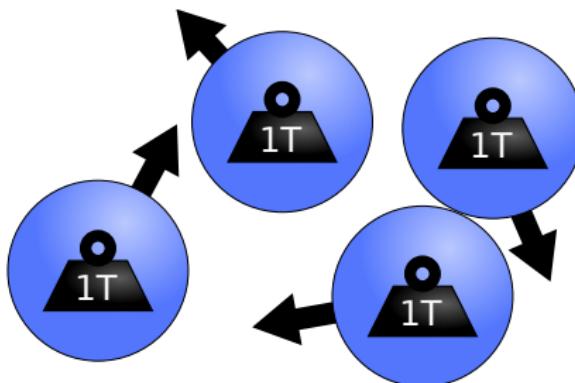
Process Scale

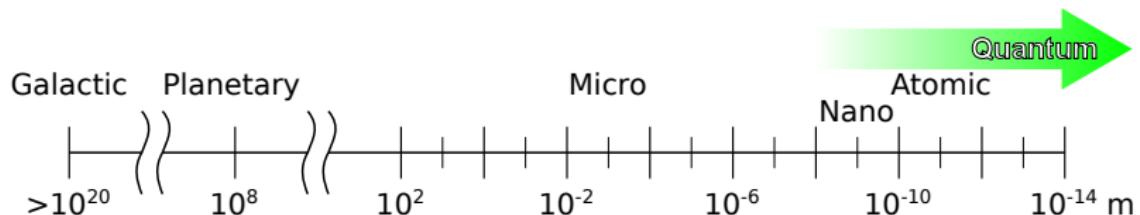
Planetary/Galactic Scale

Equation of Motion

Particle Method

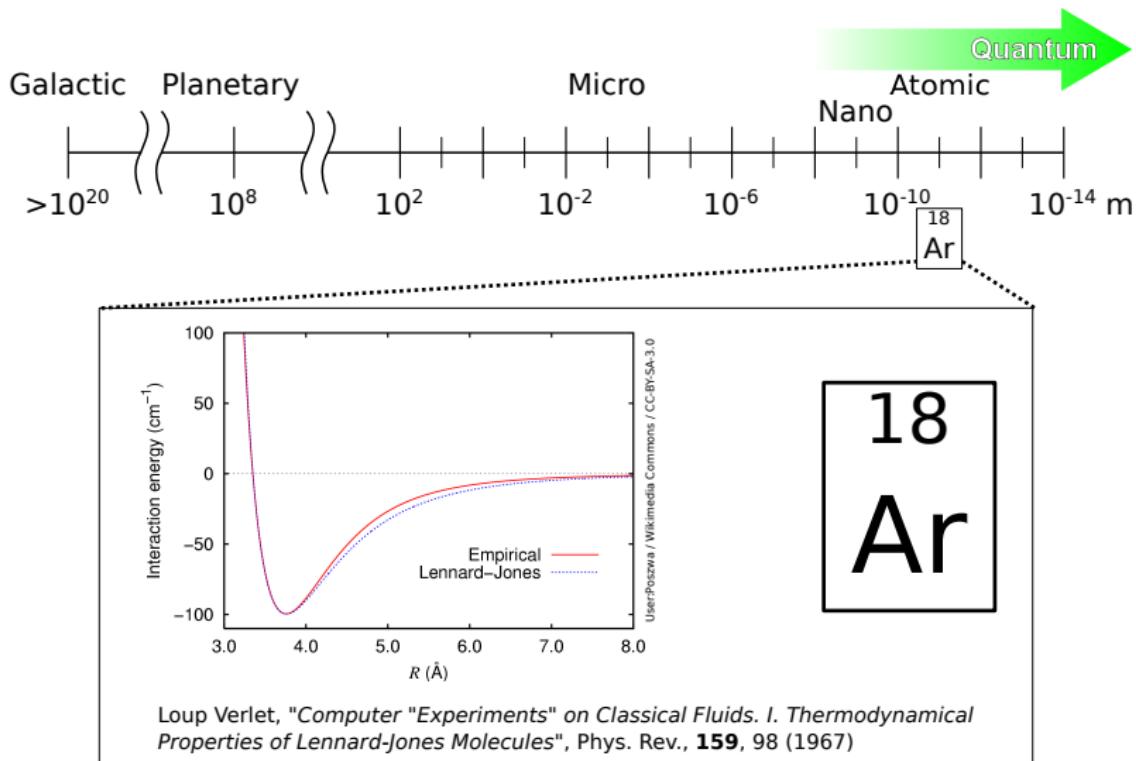
- ▶ Particle simulation concerns any system that can be modelled as a large collection of bodies/nodes/points/packets or simply particles.
- ▶ In its simplest definition, each particle represents some mass.





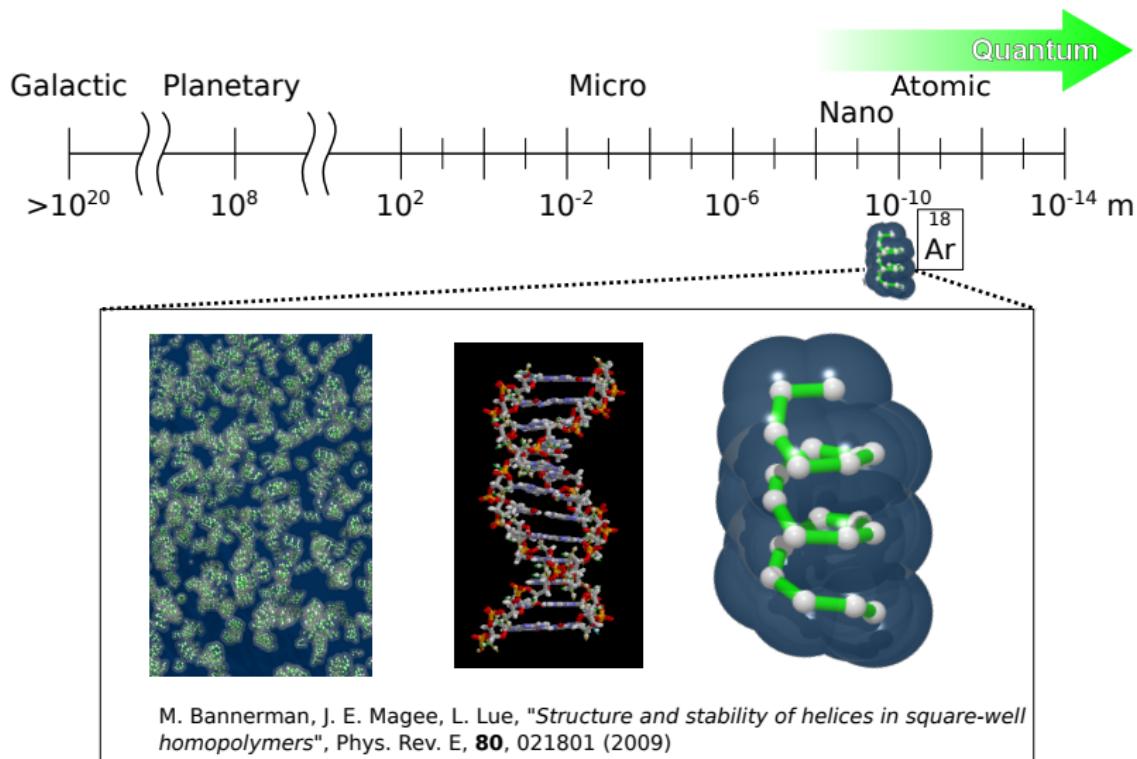
└ Where can we use them?

└ Atomic Scale



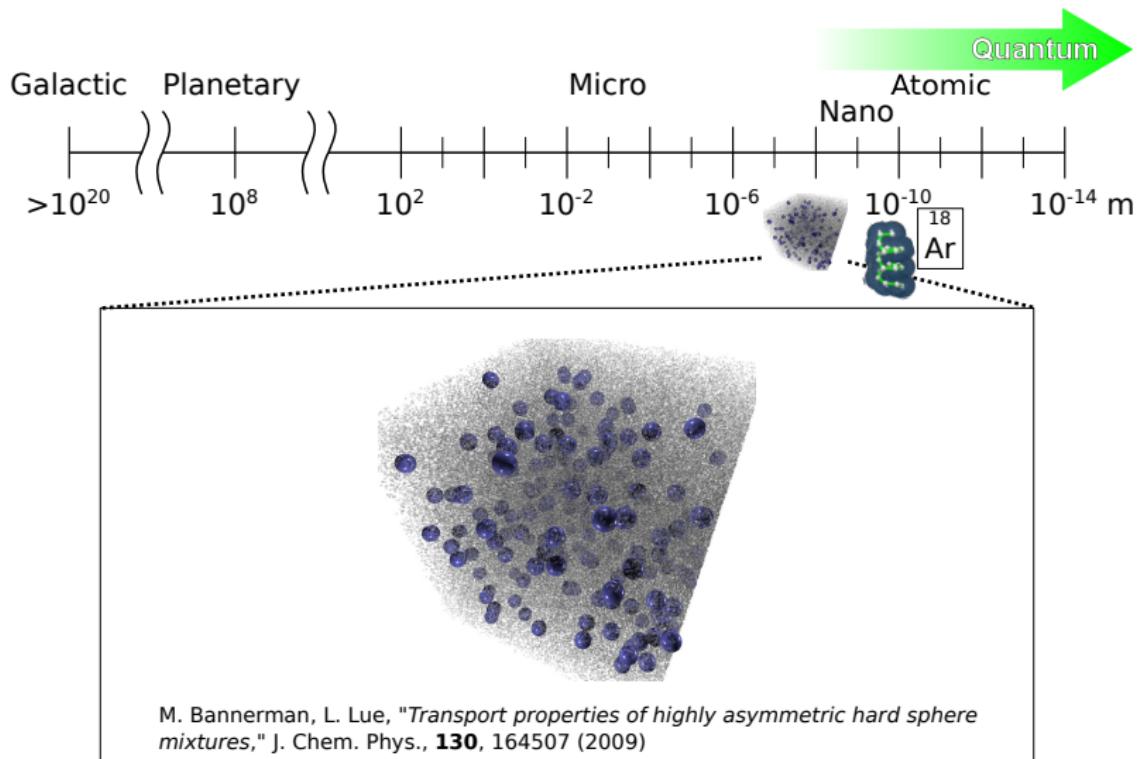
└ Where can we use them?

└ Atomic Scale



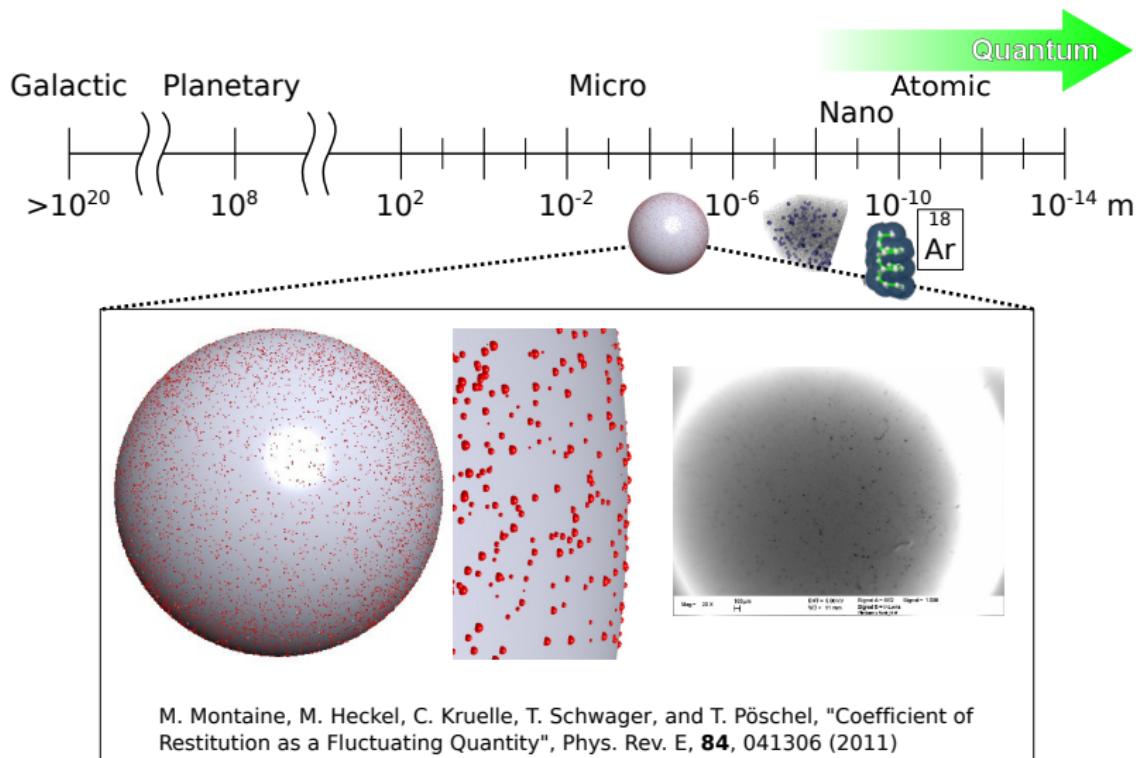
└ Where can we use them?

└ Atomic Scale



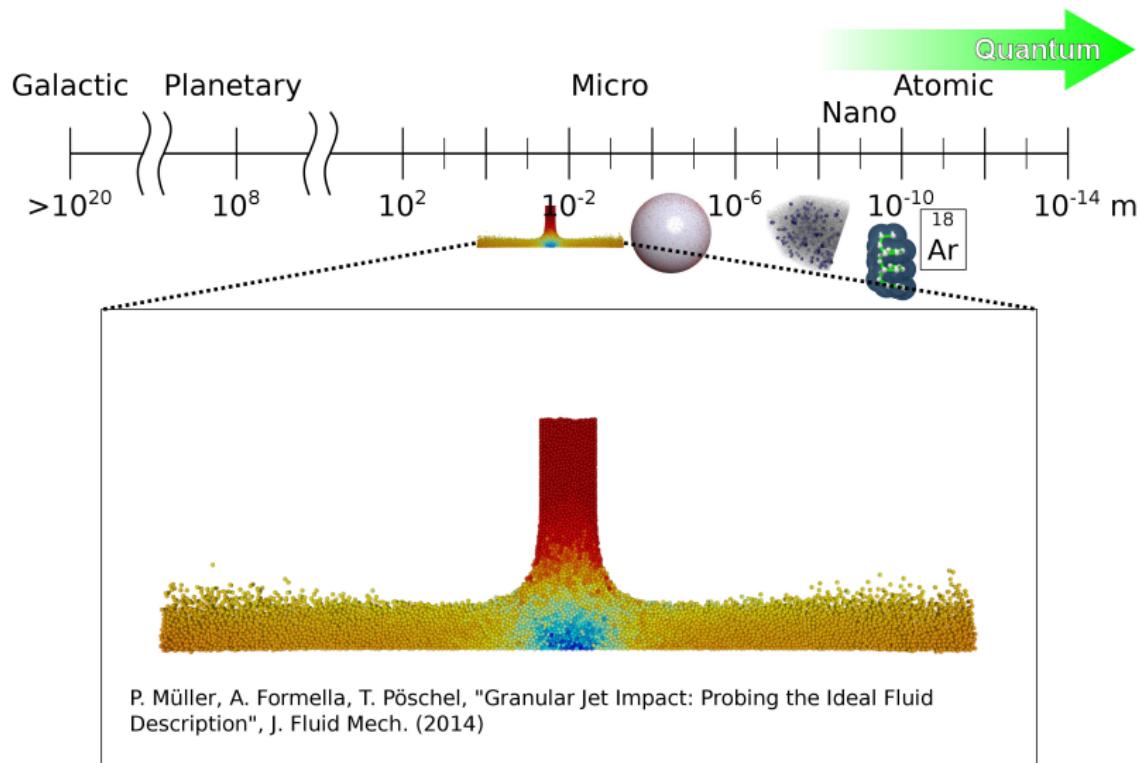
└ Where can we use them?

└ (Micro)meter Scale



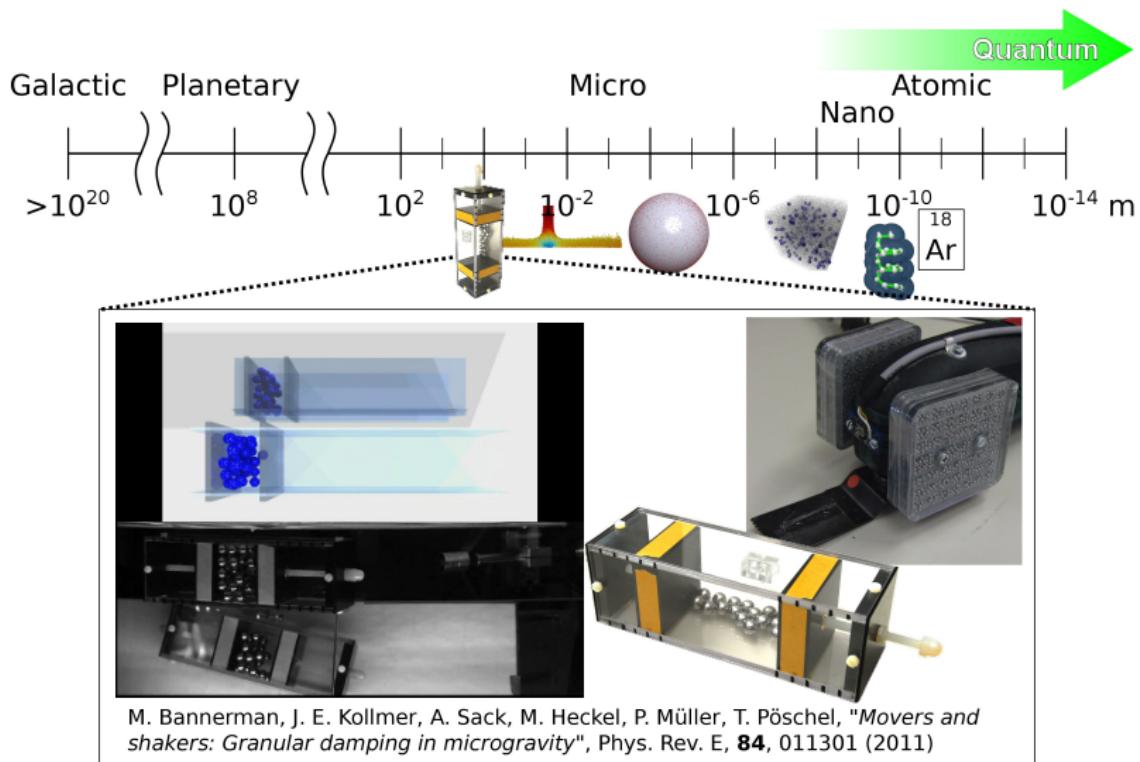
└ Where can we use them?

└ (Micro)meter Scale



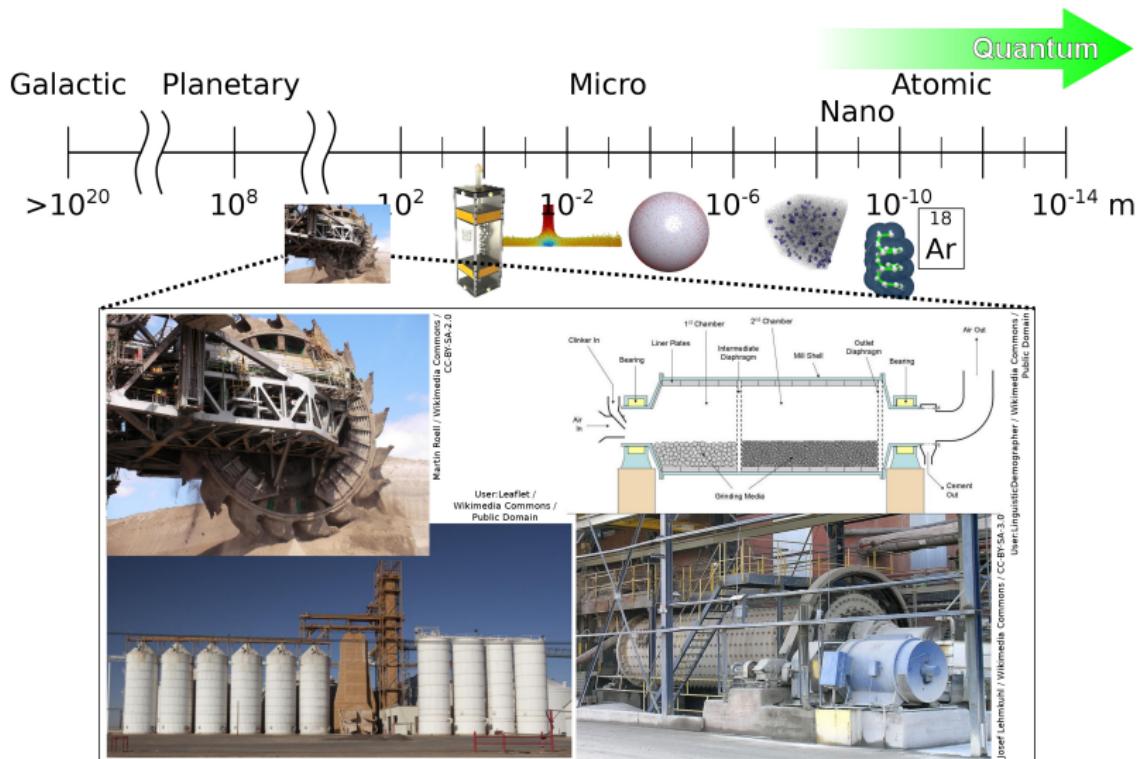
└ Where can we use them?

└ (Micro)meter Scale



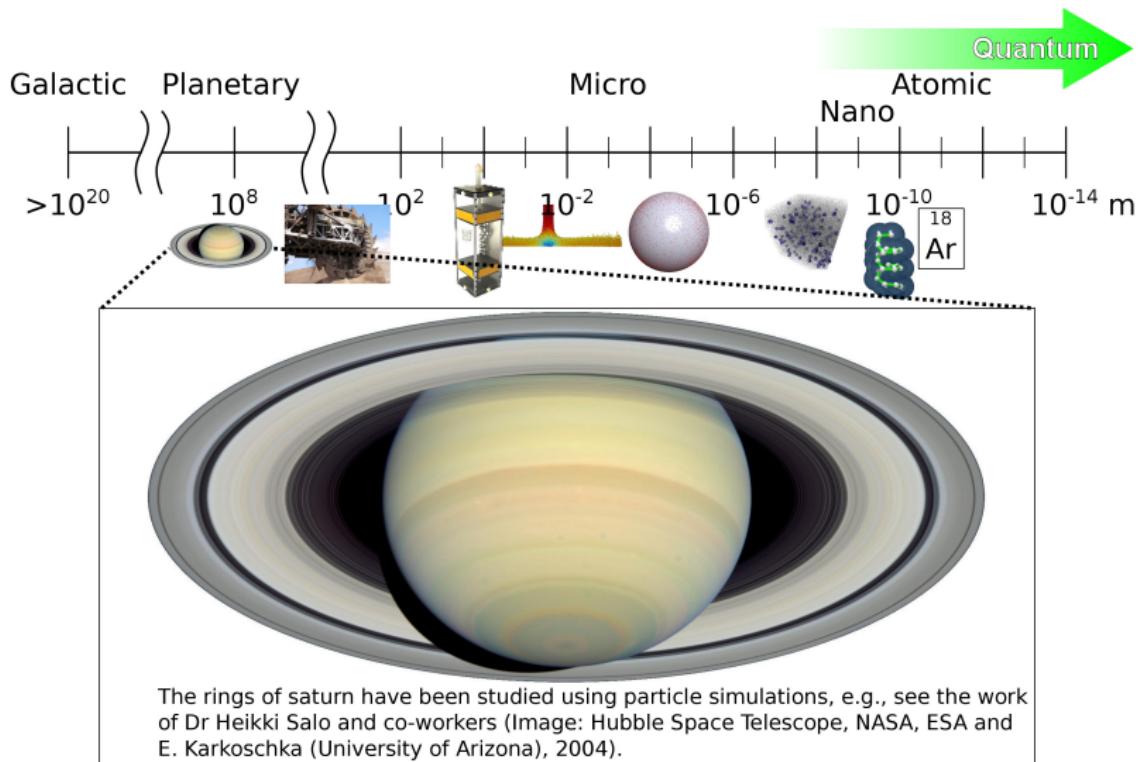
└ Where can we use them?

└ Process Scale



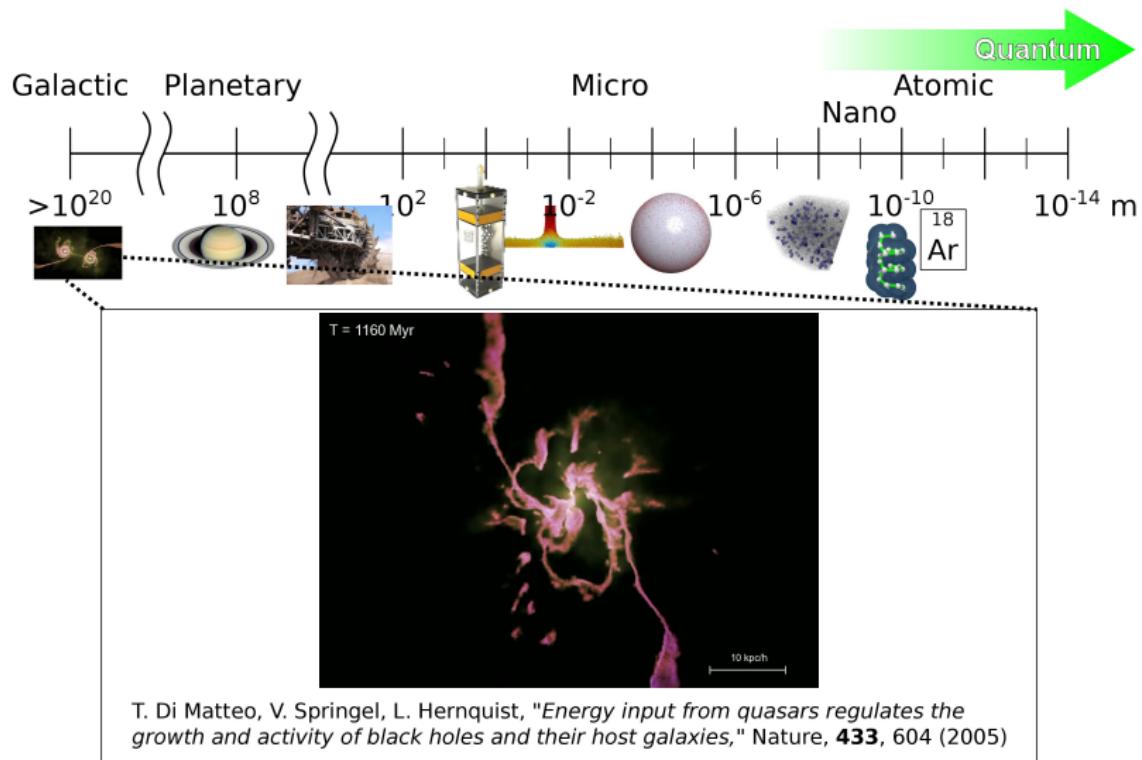
└ Where can we use them?

└ Planetary/Galactic Scale



└ Where can we use them?

└ Planetary/Galactic Scale



Outline

Where can we use them?

Atomic Scale

(Micro)meter Scale

Process Scale

Planetary/Galactic Scale

Equation of Motion

- ▶ Molecular dynamics is used at all these different scales!
- ▶ But why is it so general?

- ▶ Particle simulations are simply the task of finding a solution to Newton's equation of motion, applied to each body/particle/node i :

$$\mathbf{F}_i = m_i \mathbf{a}_i$$

where \mathbf{F}_i is the force acting on particle i , m_i is its mass and \mathbf{a}_i its acceleration.

- ▶ This is why particle simulation techniques are so general, the underlying equation is so general!

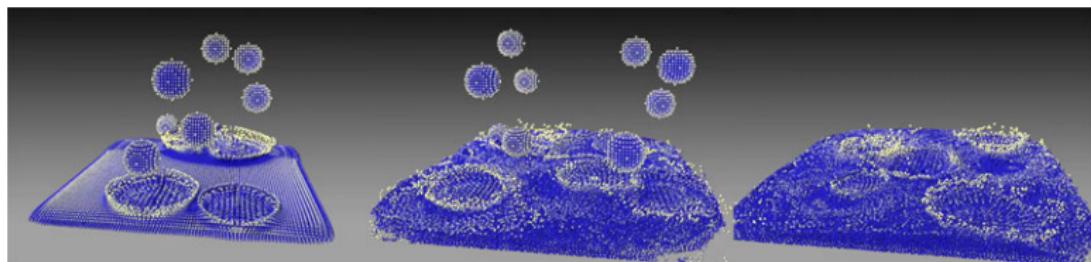


Figure : Real-time SPH simulation, Harada et al., Computer Graphics International (2007)

- ▶ This is why even hydrodynamics can be solved.
- ▶ If our particles are “packets” of fluid, we just use the Navier-Stokes equation:

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \nabla \cdot \mathbf{T} + \mathbf{F}_{body}$$

which is another form of Newton's equation of motion:

$$m\mathbf{a} = \mathbf{F} = \mathbf{F}_{pressure} + \mathbf{F}_{viscous} + \mathbf{F}_{body}$$

Part II

Details of Molecular Dynamics

Outline

Rough Outline of the Algorithm

Force Models

Splitting the Force Term

Atomic Forces

Granular

Initial Conditions

Boundary Conditions

Integrator

Purpose of the Integrator

Choosing an Integration Scheme

Velocity Verlet Integration

Final Remarks

- ▶ We have Newton's equation of motion to solve for every particle.
- ▶ But how do we solve the $3N$ equations of motion?

$$\frac{\mathbf{F}_i}{m_i} = \mathbf{a}_i = \dot{\mathbf{v}}_i = \ddot{\mathbf{x}}_i$$

$$\frac{\mathbf{F}_i}{m_i} = \mathbf{a}_i = \dot{\mathbf{v}}_i = \ddot{\mathbf{x}}_i$$

- ▶ We first need some **model** for the forces in the system:
 - ▶ These should be simple expressions of the particle position \mathbf{x}_i and velocity \mathbf{v}_i :

$$\mathbf{F}_i \equiv \mathbf{F}_i(\mathbf{x}_i, \mathbf{v}_i)$$

- ▶ Probably the simplest force to implement is the constant gravity e.g. on earth (gravity between galaxies is harder!):

$$\mathbf{F}_i^{gravity} \equiv m_i \mathbf{g}$$

- ▶ More on this in a bit...

$$\frac{\mathbf{F}_i}{m_i} = \mathbf{a}_i = \dot{\mathbf{v}}_i = \ddot{\mathbf{x}}_i$$

- ▶ We then need some initial conditions: The positions and velocities of all particles at the time $t = 0$.
- ▶ What's a valid starting point for a simulation of the universe? A big bang?
- ▶ What about atomic systems? We can't even see those, let alone take a snapshot to replicate in our simulation.
- ▶ More on this later...

$$\frac{\mathbf{F}_i}{m_i} = \mathbf{a}_i = \dot{\mathbf{v}}_i = \ddot{\mathbf{x}}_i$$

- ▶ We also need boundary conditions...
- ▶ If we want to simulate the whole universe, are there boundary conditions?
- ▶ What about the tiny atomic systems? How do we get rid of the effect of boundaries?
- ▶ More later...

$$\frac{\mathbf{F}_i}{m_i} = \mathbf{a}_i = \dot{\mathbf{v}}_i = \ddot{\mathbf{x}}_i$$

- ▶ Finally, once we have an **expression for \mathbf{F}_i** , the **initial conditions**, and the **boundary conditions**, we need only one more thing...
...an integrator!
- ▶ If we numerically integrate \mathbf{F}_i/m_i twice in time, we will get the velocity and position of all particles at all later times!
- ▶ That's it! Lets look at all these topics in detail...

Outline

Rough Outline of the Algorithm

Force Models

Splitting the Force Term

Atomic Forces

Granular

Initial Conditions

Boundary Conditions

Integrator

Purpose of the Integrator

Choosing an Integration Scheme

Velocity Verlet Integration

Final Remarks

The Force Term

- ▶ Our major challenge is the selection of an appropriate model for the force term \mathbf{F}_i .
- ▶ The force model decides whether we're simulating an atom, or a galaxy, or a grain of sand.

The Force Term

- ▶ We have two parts of the force to define:

$$\mathbf{F}_i = \mathbf{F}_{body}(\mathbf{x}_i, \mathbf{v}_i) + \sum_{j \neq i} \mathbf{F}_2(\mathbf{x}_i, \mathbf{v}_i, \mathbf{x}_j, \mathbf{v}_j) + \dots$$

- ▶ \mathbf{F}_{body} is the body force acting on a particle arising from an external field (e.g., simple gravitational force: $\mathbf{F}_{body} = \mathbf{F}_i^{gravity} = m_i \mathbf{g}$).
- ▶ \mathbf{F}_2 are the forces arising between pairs of particles (we'll ignore 3...N particle interactions!).
- ▶ The condition on the sum ($j \neq i$) excludes self interactions.

The Force Term

- ▶ We need to define the two-particle force term:

$$\mathbf{F}_2(\mathbf{x}_i, \mathbf{v}_i, \mathbf{x}_j, \mathbf{v}_j)$$

- ▶ To ensure that momentum is conserved, our simulation has to obey Newton's third law (each force has an equal and opposite reaction force).

$$\mathbf{F}_2(\mathbf{x}_i, \mathbf{v}_i, \mathbf{x}_j, \mathbf{v}_j) = -\mathbf{F}_2(\mathbf{x}_j, \mathbf{v}_j, \mathbf{x}_i, \mathbf{v}_i)$$

The Force Term

- ▶ A common way of obtaining a conservative force law is to just define the energy *potential* between two particles, $V_2(\mathbf{x}_i, \mathbf{x}_j)$.
- ▶ The force is just the spatial derivative of this energy potential.

$$\mathbf{F}_2(\mathbf{x}_i, \mathbf{x}_j) = -\nabla_{\mathbf{x}_i} V_2(\mathbf{x}_i, \mathbf{x}_j)$$

- ▶ We can define our forces by defining the energy between two particles (i.e., like the energy of a spring).
- ▶ This is usually used in molecular dynamics (atoms), as the interactions **conserve** energy.

Atomic Forces

- ▶ There is a classic example of an atomic potential, called the **Lennard-Jones** potential.
- ▶ It is attractive at long ranges and repulsive at short distances and can be used for simple systems like an Argon gas.
- ▶ The Lennard-Jones potential energy is given by

$$V_2^{LJ}(\mathbf{x}_i, \mathbf{x}_j) = V_2^{LJ}(x_{ij}) = 4 \varepsilon \left[\left(\frac{\sigma}{x_{ij}} \right)^{12} - \left(\frac{\sigma}{x_{ij}} \right)^6 \right],$$

where the value $x_{ij} = |\mathbf{x}_{ij}| = |\mathbf{x}_i - \mathbf{x}_j|$ is the length of the vector separating the centers of the two atoms.

Atomic Forces

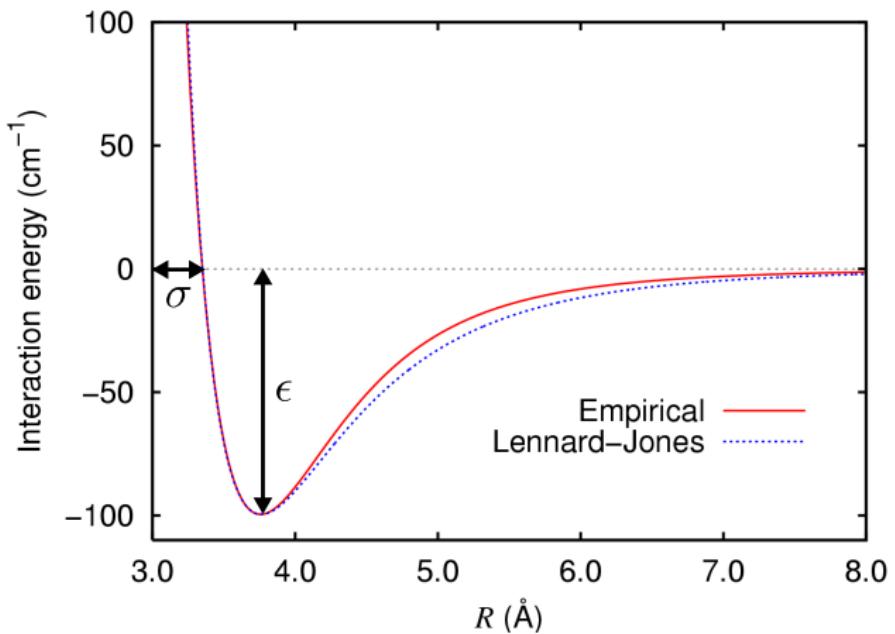


Figure : Comparing a Lennard-Jones potential for an Argon dimer with empirical results. (Source: User:Poszwa / Wikimedia Commons / CC-BY-SA-3.0)

Atomic Forces

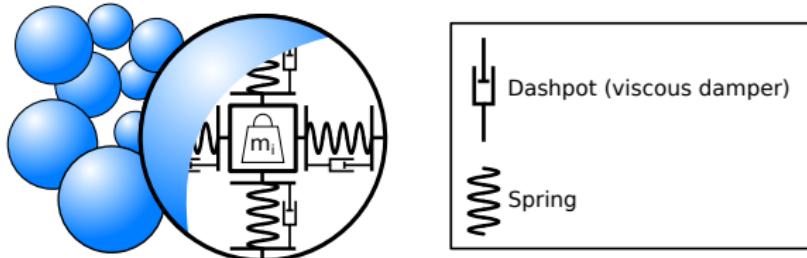
- ▶ The force is then given by

$$\begin{aligned}
 \mathbf{F}_2^{LJ}(\mathbf{x}_i, \mathbf{x}_j) &= -\nabla_{x_{ij}} V_2^{LJ}(x_{ij}) \frac{\mathbf{x}_{ij}}{x_{ij}} \\
 &= -4 \varepsilon \frac{\partial}{\partial x_{ij}} \left[\left(\frac{\sigma}{x_{ij}} \right)^{12} - \left(\frac{\sigma}{x_{ij}} \right)^6 \right] \frac{\mathbf{x}_{ij}}{x_{ij}} \\
 &= -4 \varepsilon \left[-12 \left(\frac{\sigma}{x_{ij}} \right)^{11} \frac{\sigma}{x_{ij}^2} + 6 \left(\frac{\sigma}{x_{ij}} \right)^5 \frac{\sigma}{x_{ij}^2} \right] \frac{\mathbf{x}_{ij}}{x_{ij}} \\
 &= 24 \varepsilon \left(\frac{\sigma}{x_{ij}} \right)^6 \left[2 \left(\frac{\sigma}{x_{ij}} \right)^6 - 1 \right] \frac{\mathbf{x}_{ij}}{x_{ij}^2}
 \end{aligned}$$

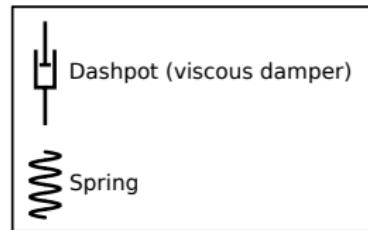
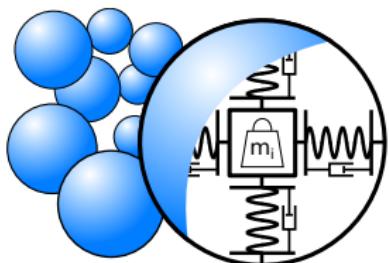
- ▶ This is our example of an atomic potential which you'll have to implement in the exercises.

Granular Forces

- ▶ Lets take another system...
- ▶ The particles may represent dust, sand, ball bearings, or ice crystals in Saturn's rings.
- ▶ A popular and simple force law for these systems is the Spring-Dashpot model.



Granular Forces



- ▶ The resultant expression for the force is quite simple:

$$\mathbf{F}_2^{SD}(\mathbf{x}_i, \mathbf{v}_i, \mathbf{x}_j, \mathbf{v}_j) = K \boldsymbol{\xi} + \gamma \dot{\boldsymbol{\xi}},$$

where K is the spring constant, γ is the dissipation constant, $\boldsymbol{\xi} = \hat{\mathbf{x}}_{ij} (\sigma - x_{ij}) \Theta(\sigma - x_{ij})$ is the overlap of the spheres with $\hat{\mathbf{x}}_{ij}$ being the unit vector $\frac{\mathbf{x}_{ij}}{x_{ij}}$, and σ is the sphere diameter.

- ▶ $\Theta(x)$ is the Heaviside step function:

$$\Theta(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$

Summing Up

- ▶ We have our example forces:

$$\mathbf{F}_2^{LJ}(\mathbf{x}_i, \mathbf{x}_j) = 24 \varepsilon \left(\frac{\sigma}{x_{ij}} \right)^6 \left[2 \left(\frac{\sigma}{x_{ij}} \right)^6 - 1 \right] \frac{\mathbf{x}_{ij}}{x_{ij}^2}$$

$$\mathbf{F}_2^{SD}(\mathbf{x}_i, \mathbf{v}_i, \mathbf{x}_j, \mathbf{v}_j) = K \boldsymbol{\xi} + \gamma \dot{\boldsymbol{\xi}}$$

- ▶ We can now calculate the force on any particle i , by summing up the one and two particle forces:

$$\mathbf{F}_i = \mathbf{F}_{body}(\mathbf{x}_i, \mathbf{v}_i) + \sum_{j \neq i} \mathbf{F}_2(\mathbf{x}_i, \mathbf{v}_i, \mathbf{x}_j, \mathbf{v}_j) + \dots$$

- ▶ But before we can integrate this, we need to know the initial and boundary conditions!

Outline

Rough Outline of the Algorithm

Force Models

- Splitting the Force Term

- Atomic Forces

- Granular

Initial Conditions

Boundary Conditions

Integrator

- Purpose of the Integrator

- Choosing an Integration Scheme

- Velocity Verlet Integration

Final Remarks

Initial Conditions

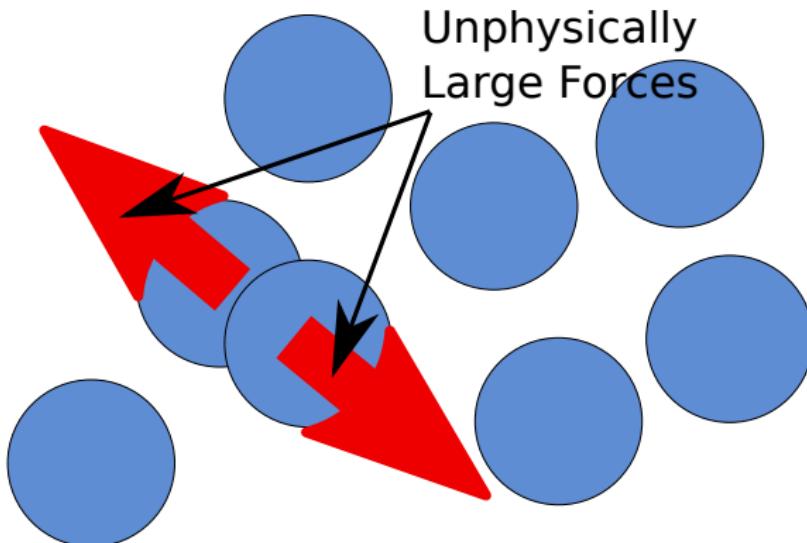
- ▶ When initializing your simulation, you must be careful.
- ▶ We should start the simulation from a realistic starting point.
- ▶ But what is a realistic starting point? Do we have any idea?
- ▶ A big assumption in molecular systems (which is usually correct) is that the system is **ergodic**.
- ▶ This means that wherever you start from, after a long enough time, you eventually end up with the same results.

Initial Conditions

- ▶ So, the simulations should be:
 - ▶ **Independent of the starting conditions** wherever you start, you will pass arbitrarily close to all possible configurations (the system is ergodic).
 - ▶ **Repeated with a wide range of valid initial conditions** to ensure a good sampling.

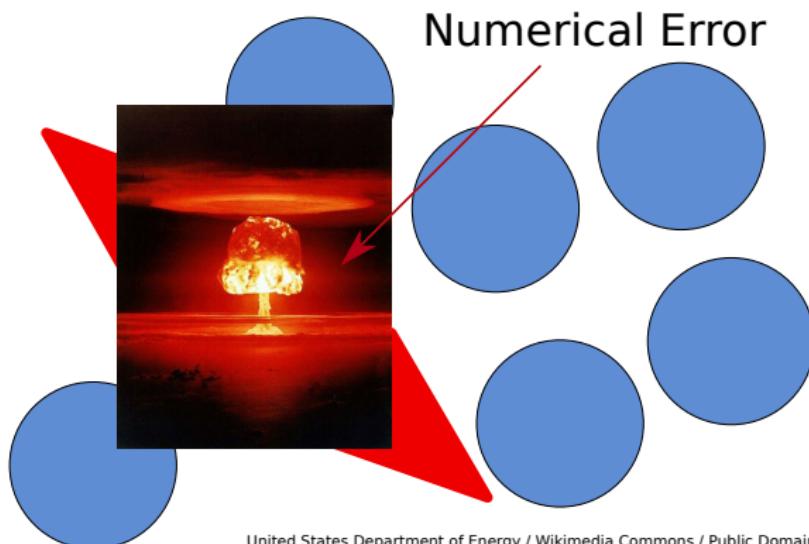
Initial Conditions

- ▶ But we still have to be careful!
- ▶ For example, if we started a simulation of steel ball bearings and accidentally started with two particles significantly overlapping, the model can predict unphysical forces.



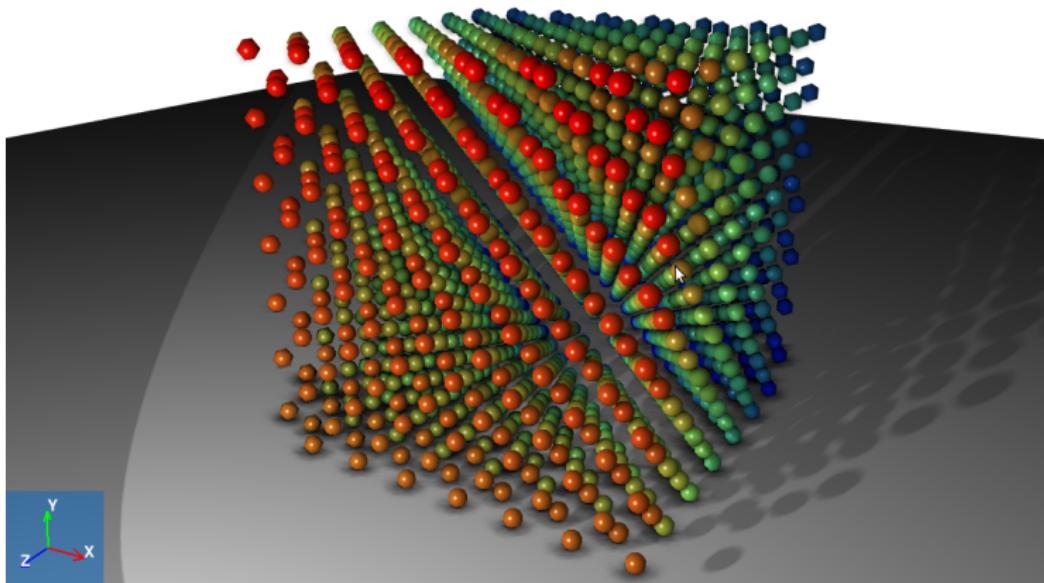
Initial Conditions

- ▶ But we still have to be careful!
- ▶ For example, if we started a simulation of steel ball bearings and accidentally started with two particles significantly overlapping, the model can predict unphysical forces.



Initial Conditions

- ▶ Typically, we arrange the particles on a regular grid, where we can ensure a minimum separation:



Initial Conditions

- ▶ For molecular simulations, the initial conditions usually assume the system is in thermodynamic equilibrium.
- ▶ The velocity components are then sampled from a scaled Gaussian/normal distribution with mean $\mu = 0$ and variance $\sigma^2 = \frac{k_B T}{m_i}$:

$$f(v_{i,d}) = \sqrt{\frac{m_i}{2\pi k_B T}} e^{-\frac{m_i v_{i,d}^2}{2k_B T}} \quad \text{for } d = 1 \dots 3$$

where $k_B T$ is the initial temperature of the fluid.

- ▶ This distribution is used for each component of the velocity of all particles in the system.

Outline

Rough Outline of the Algorithm

Force Models

- Splitting the Force Term

- Atomic Forces

- Granular

Initial Conditions

Boundary Conditions

Integrator

- Purpose of the Integrator

- Choosing an Integration Scheme

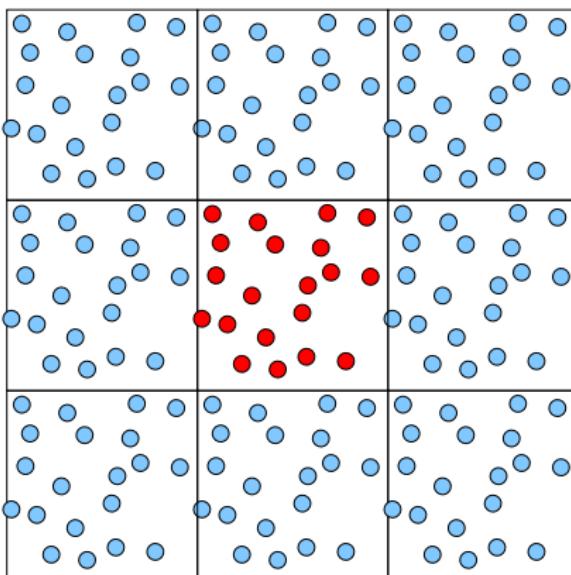
- Velocity Verlet Integration

Final Remarks

Boundary Conditions

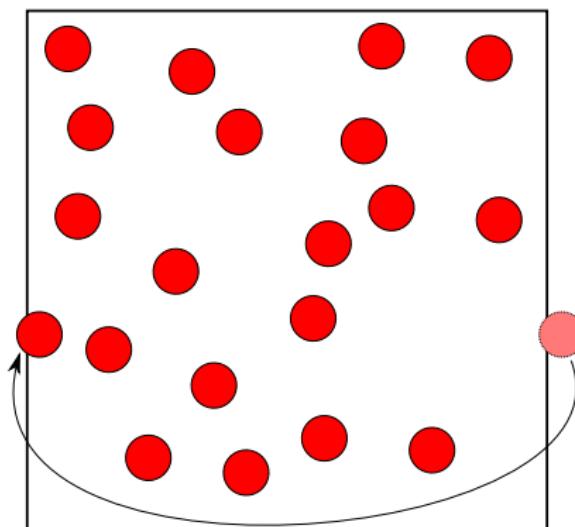
- ▶ Some N-body systems are unbounded and exist in an infinite space (e.g, galactic simulations, isolated polymers).
- ▶ Others systems are completely bounded by walls (like the granular dampeners).
- ▶ But in molecular dynamics, usually you wish to simulate some element of a fluid, without wall effects and representative of a much larger bulk fluid.
- ▶ Here, periodic boundary conditions (think of the game asteroids) are typically used.

Periodic Boundary Conditions



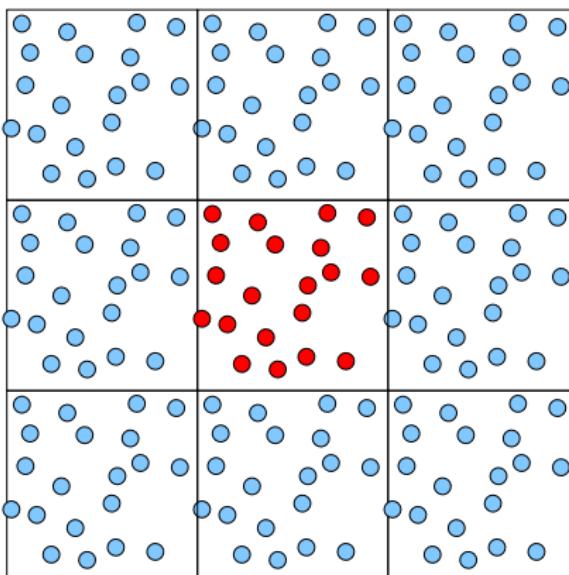
- ▶ The “primary image” of the simulation is surrounded with an infinite number of periodic images.

Periodic Boundary Conditions



- ▶ If a particle leaves one side of the simulation, it re-enters on the other side.

Periodic Boundary Conditions



- ▶ These periodic boundary conditions eliminate wall effects and allows small simulations to be representative of bulk fluids.

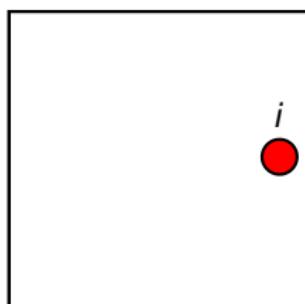
Periodic Boundary Conditions

- ▶ For interactions with a finite range periodic boundary conditions are quite easy to implement.
- ▶ The size of the simulation domain L_{sys} has to be at least twice the maximum interaction range.
- ▶ Any time the distance between two particles has to be calculated, the **minimum image criterion** can be used.
- ▶ The nearest periodic image can be calculated using:

$$\mathbf{x}_{ij}^{PBC} = \mathbf{x}_{ij} - L_{sys} \operatorname{int} \left(\frac{\mathbf{x}_{ij}}{L_{sys}} \right)$$

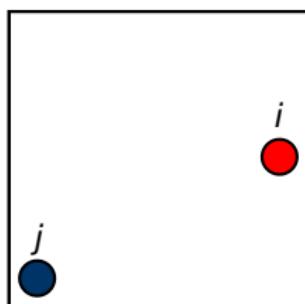
Periodic Boundary Conditions

$$\mathbf{x}_{ij}^{PBC} = \mathbf{x}_{ij} - L_{sys} \operatorname{int} \left(\frac{\mathbf{x}_{ij}}{L_{sys}} \right)$$



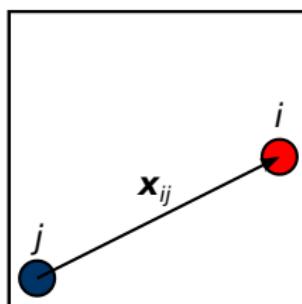
Periodic Boundary Conditions

$$\mathbf{x}_{ij}^{PBC} = \mathbf{x}_{ij} - L_{sys} \operatorname{int} \left(\frac{\mathbf{x}_{ij}}{L_{sys}} \right)$$



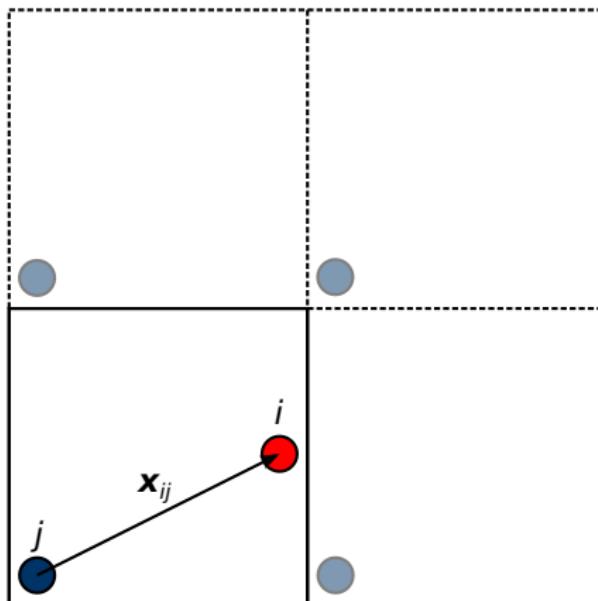
Periodic Boundary Conditions

$$\mathbf{x}_{ij}^{PBC} = \mathbf{x}_{ij} - L_{sys} \operatorname{int} \left(\frac{\mathbf{x}_{ij}}{L_{sys}} \right)$$



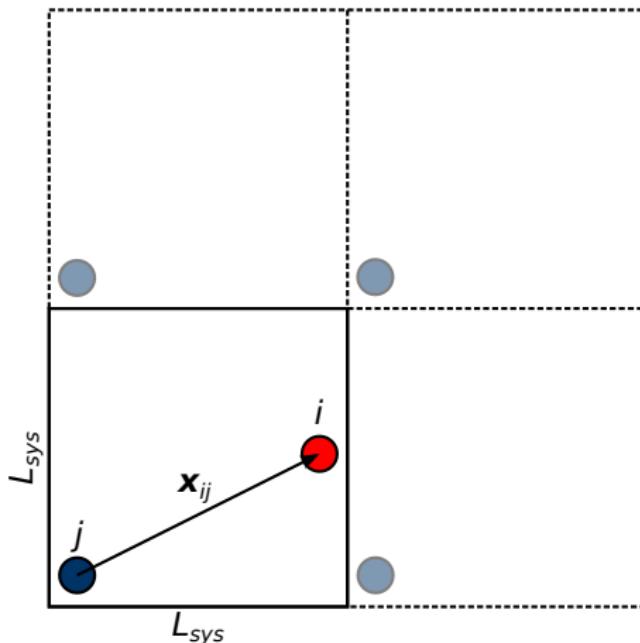
Periodic Boundary Conditions

$$\mathbf{x}_{ij}^{PBC} = \mathbf{x}_{ij} - L_{sys} \text{ int} \left(\frac{\mathbf{x}_{ij}}{L_{sys}} \right)$$



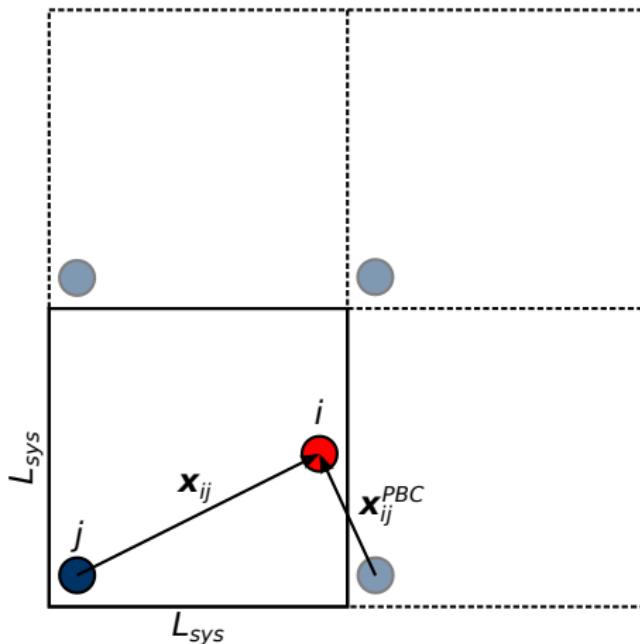
Periodic Boundary Conditions

$$\mathbf{x}_{ij}^{PBC} = \mathbf{x}_{ij} - L_{sys} \text{ int} \left(\frac{\mathbf{x}_{ij}}{L_{sys}} \right)$$



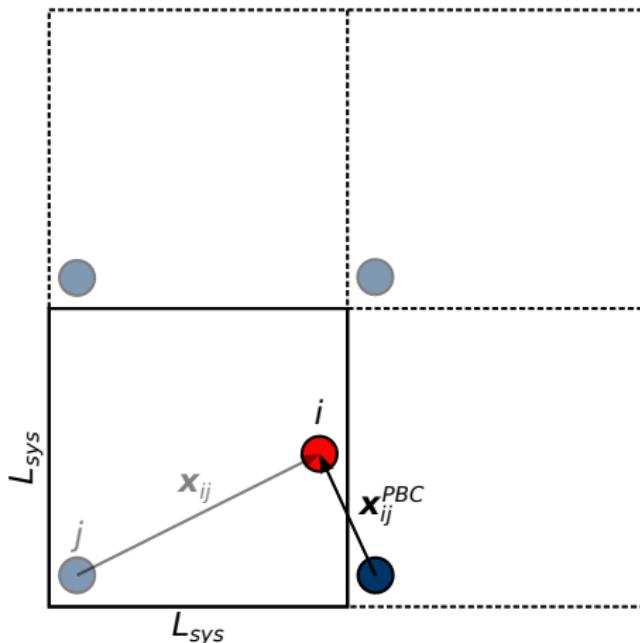
Periodic Boundary Conditions

$$\mathbf{x}_{ij}^{PBC} = \mathbf{x}_{ij} - L_{sys} \text{ int} \left(\frac{\mathbf{x}_{ij}}{L_{sys}} \right)$$



Periodic Boundary Conditions

$$\mathbf{x}_{ij}^{PBC} = \mathbf{x}_{ij} - L_{sys} \text{ int} \left(\frac{\mathbf{x}_{ij}}{L_{sys}} \right)$$





Outline

Splitting the Force Term

Atomic Forces

Granular

Integrator

Purpose of the Integrator

Choosing an Integration Scheme

Velocity Verlet Integration

Use for Numerical Integration

- ▶ We use numerical integration to integrate $\mathbf{F}_i = m_i \mathbf{a}_i$ as it is almost always too difficult to solve analytically.
- ▶ We need an algorithm to calculate the positions, velocities and accelerations at the next time step:

$$\mathbf{x}_i(t) \rightarrow \mathbf{x}_i(t + \Delta t)$$

$$\mathbf{v}_i(t) \rightarrow \mathbf{v}_i(t + \Delta t)$$

$$\mathbf{a}_i(t) \rightarrow \mathbf{a}_i(t + \Delta t)$$

where Δt is our time step (how much time passes in a single integration step).

Integration Schemes

- ▶ Thanks to the definition of the forces, we can exactly define the accelerations at the current time step.

$$\mathbf{a}_i(t) = m_i^{-1} \mathbf{F}_i(\mathbf{x}_i(t), \mathbf{v}_i(t))$$

- ▶ So we only need an integration scheme for the positions and velocities, given that we know the accelerations.

$$\mathbf{x}_i(t) \rightarrow \mathbf{x}_i(t + \Delta t)$$

$$\mathbf{v}_i(t) \rightarrow \mathbf{v}_i(t + \Delta t)$$

Integration Schemes

- ▶ The simplest integration method is to just take a truncated Taylor series (explicit Euler), like so

$$\begin{aligned}\mathbf{x}_i(t + \Delta t) &= \mathbf{x}_i(t) + \Delta t \mathbf{v}_i(t) \\ \mathbf{v}_i(t + \Delta t) &= \mathbf{v}_i(t) + \Delta t \mathbf{a}_i(t)\end{aligned}$$

- ▶ This works, but is typically not used in Molecular Dynamics due to numeric problems.
- ▶ Can we choose a better method? How do we choose a better method?

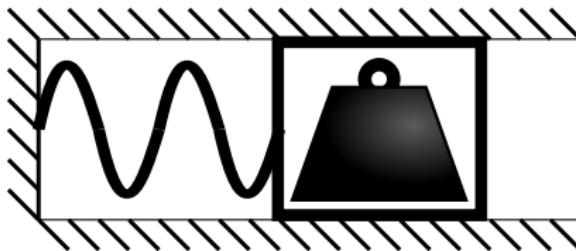
Choosing an Integrator

Choosing a numerical integration scheme is difficult for several reasons:

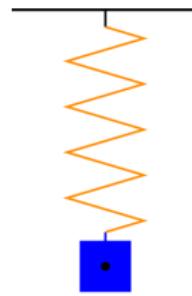
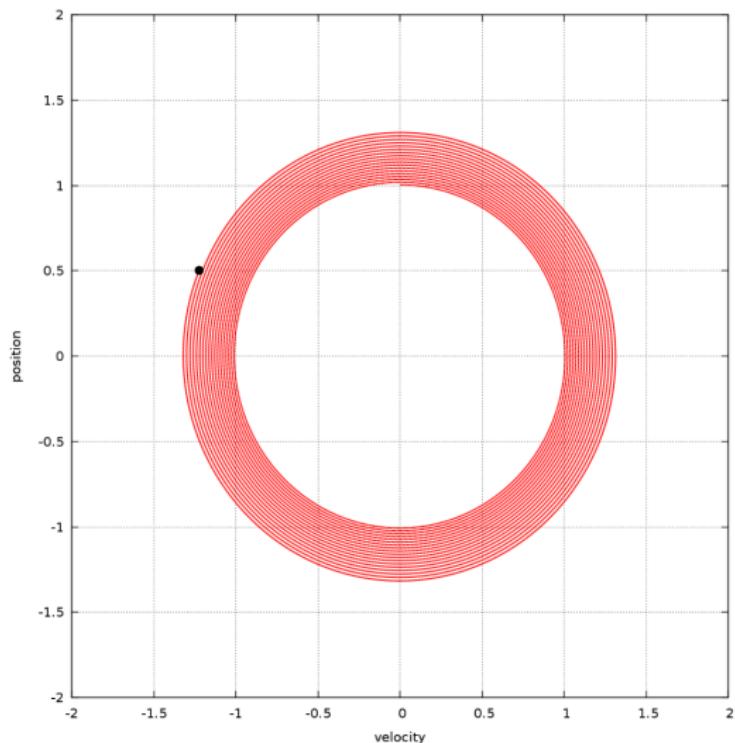
- ▶ There are many techniques and each is solvable at varying orders of approximation (e.g., Runge-Kutta).
- ▶ There is a trade-off between memory and order of approximation (higher orders require more stored derivatives).
- ▶ There is a trade-off between order of approximation and speed (higher orders can be more stable at larger time steps, but can also be more computationally expensive).

Choosing an Integrator

- ▶ A good choice is a **symplectic integrator**, as these are good at conserving the total energy of the system (the Hamiltonian).
- ▶ This can be easily illustrated using a simple harmonic oscillator:



Choosing an Integrator



Velocity Verlet/Leapfrog Algorithm

- ▶ A common symplectic integrator in Molecular Dynamics is the Velocity Verlet algorithm.

$$\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \mathbf{v}_i(t)\Delta t + \frac{\mathbf{a}_i(t)\Delta t^2}{2}$$
$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \frac{\mathbf{a}_i(t) + \mathbf{a}_i(t + \Delta t)}{2}\Delta t$$

- ▶ An undesirable part of the algorithm is that both the old and new values of the acceleration are required in the velocity integration.
- ▶ But this is easily fixed...

Velocity Verlet/Leapfrog Algorithm

- ▶ The algorithm can be rewritten in a form which uses less memory and does not require the old value of the accelerations when integrating the velocity.
 - ▶ First, the position is integrated and a half step is performed in the velocity.

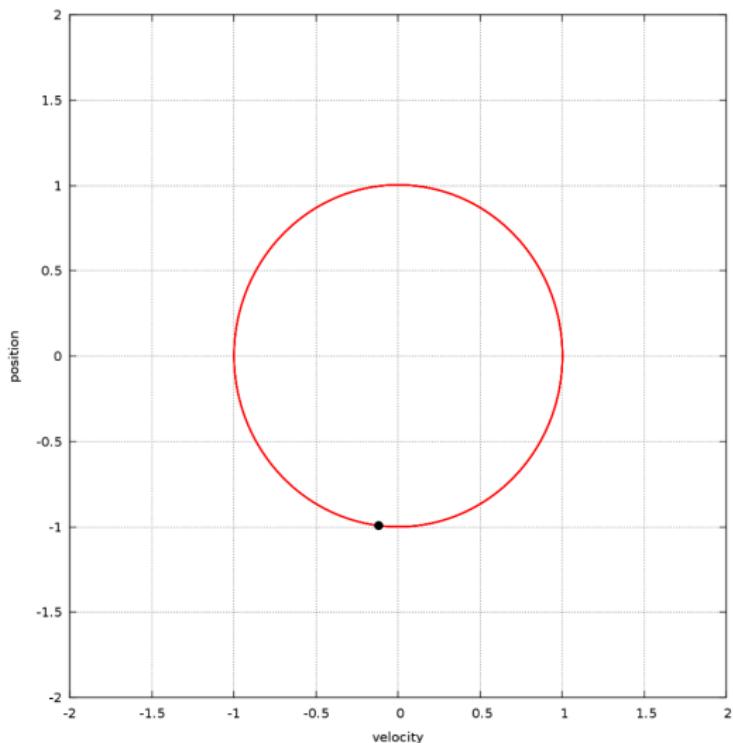
$$\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \mathbf{v}_i(t)\Delta t + \frac{\mathbf{a}_i(t)\Delta t^2}{2}$$

$$\mathbf{v}_i(t + \frac{1}{2}\Delta t) = \mathbf{v}_i(t) + \frac{\mathbf{a}_i(t)}{2}\Delta t$$

- ▶ The forces are then recalculated to generate $\mathbf{a}_i(t + \Delta t)$.
- ▶ Finally, the velocity integration is completed.

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t + \frac{1}{2}\Delta t) + \frac{\mathbf{a}_i(t + \Delta t)}{2}\Delta t$$

Choosing an Integrator



```
integrator: Velocity Verlet
kinetic energy: 0.00708702
potential energy: 0.492913
total energy: 0.5
```

- ▶ The Velocity Verlet algorithm is useful in molecular dynamics as the forces are typically only a function of the positions.
- ▶ But what about the spring-dashpot system?

$$\mathbf{F}_2^{SD}(\mathbf{x}_i, \mathbf{v}_i, \mathbf{x}_j, \mathbf{v}_j) = K \boldsymbol{\xi} + \gamma \dot{\boldsymbol{\xi}}$$

where $\dot{\boldsymbol{\xi}} = -\hat{\mathbf{x}}_{ij} (\hat{\mathbf{x}}_{ij} \cdot (\mathbf{v}_i - \mathbf{v}_j)) \Theta(\sigma - |\mathbf{x}_{ij}|)$

- ▶ If the forces are functions of the velocity then we can no longer re-calculate the forces at the half time step and we must resort to another integrator (e.g. special Runge-Kutta schemes).

Outline

Rough Outline of the Algorithm

Force Models

- Splitting the Force Term

- Atomic Forces

- Granular

Initial Conditions

Boundary Conditions

Integrator

- Purpose of the Integrator

- Choosing an Integration Scheme

- Velocity Verlet Integration

Final Remarks

Summary

A simple molecular dynamics algorithm can be broken down to:

1. Take N particles, on a regular grid with Gaussian distributed velocities.
2. First integration step.

$$\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \mathbf{v}_i(t)\Delta t + \frac{\mathbf{a}_i(t)\Delta t^2}{2}$$

$$\mathbf{v}_i(t + \frac{1}{2}\Delta t) = \mathbf{v}_i(t) + \frac{\mathbf{a}_i(t)}{2}\Delta t$$

3. Calculate forces acting on the particles.

$$\mathbf{a}_i(t + \Delta t) = \frac{\mathbf{F}_i}{m_i} = \frac{1}{m_i} \left[\mathbf{F}_{body}(\mathbf{x}_i, \mathbf{v}_i) + \sum_{j \neq i} \mathbf{F}_2(\mathbf{x}_i, \mathbf{v}_i, \mathbf{x}_j, \mathbf{v}_j) \right]$$

4. Finish integration to the next time step.

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t + \frac{1}{2}\Delta t) + \frac{\mathbf{a}_i(t + \Delta t)}{2}\Delta t$$

5. If we've simulated long enough, stop, otherwise goto 2.

Summary

- ▶ To make sure you've implemented the integration right you should check the total energy:

$$U = \sum_i \left(\frac{1}{2} m_i \mathbf{v}_i^2 + \sum_{j=i+1}^N V_2(\mathbf{x}_i, \mathbf{x}_j) \right)$$

- ▶ It should stay **approximately constant!**
- ▶ Make sure that your simulation does not spiral out of control!

Outlook

- ▶ Next time we will cover some optimization strategies you can apply to your MD code to increase the performance.
- ▶ In the second part of the next lecture, Smoothed Particle Hydrodynamics (SPH) will be introduced, a way to do particle-based hydrodynamics.