Models and Numerical Methods in Mechanics and Physicochemistry

Lecture 1.
Molecular Dynamics I

Vladislav A. Yastrebov

MINES ParisTech, PSL Research University, Centre des Matériaux, CNRS UMR 7633,
Evry, France

@ Centre des Matériaux
February 1, 2016

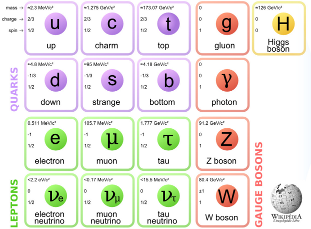
Outline

1. Matter constituents
2. Chemical bonds
3. Schrödinger equation
4. DFT and electronic structure calculations
5. From quantum mechanics to Newtonian mechanics
6. Long and short range potentials

1. Lennard-Jones potential
2. Cutoff radius
3. Force calculations
4. Linked cell method
5. Time integration
6. Algorithm
7. Boundary conditions
8. Initial conditions

Matter constituents

Standard Model



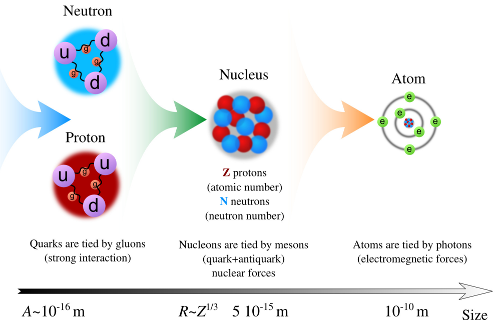
mass → charge → spin →	≈2.3 MeV/c^2 2/3 1/2 u up	≈1.275 GeV/c^2 2/3 1/2 c charm	≈173.07 GeV/c^2 2/3 1/2 t top	0 0 1 g gluon	≈126 GeV/c^2 0 0 H Higgs boson
QUARKS	≈4.8 MeV/c^2 -1/3 1/2 d down	≈95 MeV/c^2 -1/3 1/2 s strange	≈4.18 GeV/c^2 -1/3 1/2 b bottom	0 0 1 γ photon	
	0.511 MeV/c^2 -1 1/2 e electron	105.7 MeV/c^2 -1 1/2 μ muon	1.777 GeV/c^2 -1 1/2 τ tau	91.2 GeV/c^2 0 1 Z Z boson	GAUGE BOSONS
LEPTONS	<2.2 eV/c^2 0 1/2 ν_e electron neutrino	<0.17 MeV/c^2 0 1/2 ν_μ muon neutrino	<15.5 MeV/c^2 0 1/2 ν_τ tau neutrino	80.4 GeV/c^2 ±1 1 W W boson	

Table from WikipediA

QCD

Nuclear physics

Chemistry (QED)



Z protons
(atomic number)
N neutrons
(neutron number)

Quarks are tied by gluons
(strong interaction)

Nucleons are tied by mesons
(quark+antiquark)
nuclear forces

Atoms are tied by photons
(electromegnetic forces)

$A\sim10^{-16}$ m

$R\sim Z^{1/3}$ $5\ 10^{-15}$ m

10^{-10} m

Size

Chemical bonds

Nature of bonds :

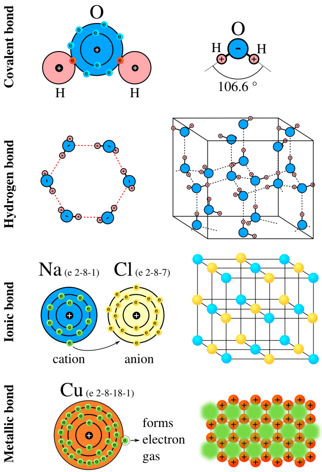
- Electrostatic force
- Electrons sharing mechanism

Strength of bonds :

- Strong (ionic, covalent)
- Weak (hydrogen, van der Waals)

Examples :

- Covalent (H_2O, H_2)
- Hydrogen (H_2O, DNA)
- Ionic (NaCl, NaF)
- Metallic (all metals)
- Van der Waals (dipole-dipole e.g. HCl-HCl, induced dipoles)



Linus Pauling "The Nature of the Chemical Bond"
`http://scarc.library.oregonstate.edu/coll/pauling/bond/index.html`

Assemblies

Lattices

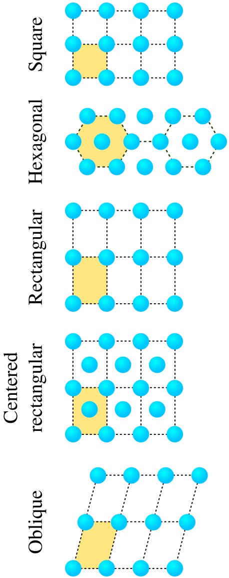
- In 2D : 5 Bravais lattices
- In 3D : 14 Bravais lattices

Molecules

- Diatomic gas (N_2, O_2)
- Ethanol (C_2H_50H)
- Macromolecules (rubber, DNA, polyethene, protein)

Amorphous

- Silica SiO_2
- Metallic glass



Complexity

- Non-relativistic Schrödinger equation for a single particle in an electric field

$$i\hbar\frac{\partial\Psi(\boldsymbol{r},t)}{\partial t} = \left[-\frac{\hbar^2}{2\mu}\nabla^2 + V(\boldsymbol{r},t)\right]\Psi(\boldsymbol{r},t),$$

  where Ψ is the wave function, V is particle's potential energy, μ is its reduced mass

- For n particles

$$i\hbar\frac{\partial\Psi(\boldsymbol{r}_1,\dots,\boldsymbol{r}_n,t)}{\partial t} = \left[-\frac{\hbar^2}{2}\left(\frac{\nabla_1^2}{\mu_1}+\dots+\frac{\nabla_n^2}{\mu_n}\right) + V(\boldsymbol{r}_1,\dots,\boldsymbol{r}_n,t)\right]\Psi(\boldsymbol{r}_1,\dots,\boldsymbol{r}_n,t),$$

- Time independent form :

$$E\Psi(\boldsymbol{r}_1,\dots,\boldsymbol{r}_n,t) = \left[-\frac{\hbar^2}{2}\left(\frac{\nabla_1^2}{\mu_1}+\dots+\frac{\nabla_n^2}{\mu_n}\right) + V(\boldsymbol{r}_1,\dots,\boldsymbol{r}_n,t)\right]\Psi(\boldsymbol{r}_1,\dots,\boldsymbol{r}_n,t),$$

- Modern chemistry can solve Schrödinger equation with up to 40-50 electrons (only !).

Density Functional Theory

- The DFT is the most successfull approach to compute the electronic structure of matter

  (Nicely presented in http://www.uam.es/personal_pdi/ciencias/jcuevas/Talks/JC-Cuevas-DFT.pdf)

- Applicable from nuclei to solids and fluids :
  molecular structures, vibrational frequencies, energies of atomization, ionization energies, electromagnetic properties, reaction paths, etc.

- Many-particle Schrödinger equation is reduced to minimization of an energy functional with respect to the non-universal functional V (system-dependent part of the total system energy)

- Nobel prize in Chemistry was attributed to Walter Kohn and John Pople for their developments in computational methods in quantum chemistry

  W Kohn, LJ Sham. Self-consistent equations including exchange and correlation effects, Phys Rev, 1965 (37 540 citations)

  RG Parr, W Yang. Density-functional theory of atoms and molecules. Oxford university press, 1989 (16 830 citations)

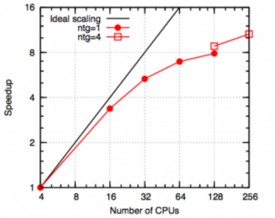
- Software : e.g. *QuantumEspresso* (www.quantum-espresso.org)

Density Functional Theory

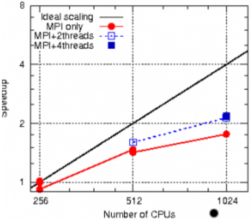
- The DFT is the most successfull approach to compute the electronic structure of matter

  (Nicely presented in http://www.uam.es/personal_pdi/ciencias/jcuevas/Talks/JC-Cuevas-DFT.pdf)

- Applicable from nuclei to solids and fluids :
- Nobel prize in Chemistry was attributed to Walter Kohn and John Pople for their developments in computational methods in quantum chemistry
- Software : e.g. *QuantumEspresso* (www.quantum-espresso.org)



128 Water molecules (1024 electrons) in a cubic box 13.35 Å side, MPI only.



Fragment of an Aβ-peptide in water containing 838 atoms and 2312 electrons in a 22.1×22.9×19.9 Å^3 cell : MPI+OpenMP

Simulating bonds

Straightforward in classical mechanics (Coulomb), non-trivial in quantum one (Schrödinger, DFT)

- **Ionic bonds**

Non-trivial (Schrödinger, DFT)

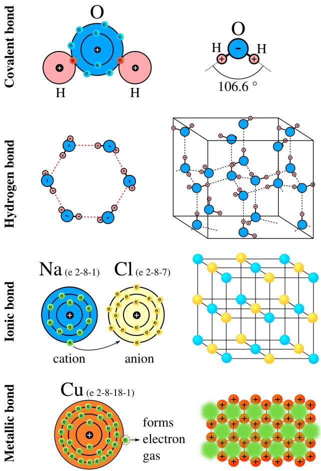
- **Covalent bonds**

If electronic structure of the molecule is well resolved (DFT) then feasible :

- **Hydrogen bonds**
- **Dipole-dipole**
- **Induced dipoles**

If electronic structure of the **lattice** *is well resolved (DFT) then feasible :*

- **Metallic bonds**



From Schrödinger equation to MD I

- Split wave function (nuclei $\boldsymbol{R}_i$, electrons $\boldsymbol{r}_j$)

$$\Psi(\boldsymbol{r}_i, \boldsymbol{R}_j, t) = \phi(\boldsymbol{r}_i, t)\chi(\boldsymbol{R}_j, t)$$

- Wave function for nuclei are generated to expectation values
- Restriction to the ground state ($\phi = \phi_0$ is retained, i.e. $\Delta E(\phi_0, \phi_1)$ is big)
- Simplified mixed quantum-mechanical and classical problem :

$$\begin{cases} M_k\ddot{\boldsymbol{R}}_k(t) = \boldsymbol{F}_{R_k} = -\nabla_{R_k} V_e(\boldsymbol{R}(t)) \\ i\hbar\frac{\partial\phi_0(\boldsymbol{r},t)}{\partial t} = \hat{\mathcal{H}}(\boldsymbol{R}(t), \boldsymbol{r})\phi_0(\boldsymbol{r}, t) \end{cases}$$

- Taylor expansion for V_e :

$$V_e(\boldsymbol{R}) \approx V_{\text{approx}}(\boldsymbol{R}) = \sum_i V_1(\boldsymbol{R}_i) + \sum_{ij} V_2(\boldsymbol{R}_i, \boldsymbol{R}_j) + \sum_{ijk} V_3(\boldsymbol{R}_i, \boldsymbol{R}_j, \boldsymbol{R}_k) + \dots,$$

where potentials V_n contain electronic degrees of freedom.

[1] Griebel M, Knapek S, Zumbusch G. “Numerical simulation in molecular dynamics”. Springer (2007).

[2] Marx D, Hutter J. “Ab initio molecular dynamics : Theory and implementation” in *Modern methods and algorithms of quantum chemistry* (2000)

From Schrödinger equation to MD II

- Global potential energy hyper-surface

$$V_{\text{approx}}(\boldsymbol{R}) = \sum_i V_1(\boldsymbol{R}_i) + \sum_{ij} V_2(\boldsymbol{R}_i, \boldsymbol{R}_j) + \sum_{ijk} V_3(\boldsymbol{R}_i, \boldsymbol{R}_j, \boldsymbol{R}_k) + \dots$$

- Newton's equation of motion in classical Molecular Dynamics

$$M_k \ddot{\boldsymbol{R}}_k(t) = -\nabla_{R_k} V_{\text{approx}}(\boldsymbol{R}(t))$$

- Potential $V_{\text{approx}}(\boldsymbol{R})$ is often approximated as a 1D pair potential

$$V_{\text{approx}}(\boldsymbol{R}) \approx \sum_{ij} V_2(r_{ij}), \quad r_{ij} = |\boldsymbol{R}_i - \boldsymbol{R}_j|$$

- Justification and error estimation are problematic
- Quantum effects and thus chemical reactions are excluded by construction

MD from the family of Particle Methods

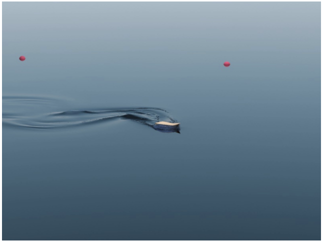
Particle methods

- **SPH**
  Smooth-particle hydrodynamics (fluids, solids)
- **DEM**
  Discrete element method (granular matter)
- **LBM**
  Lattice Boltzmann Method (fluids)
- Multi-body gravity methods (space scale systems)

Common algorithms

- Search and detection
- Data structure
- Parallelization

Coupled SPH and particle level-set
Losasso, Talton, Kwatra, Fedkiw. IEEE TVCG (2008).



Coupling grid+particle
Zheng, Zhu, Kim, Fedkiw, J. Comp. Phys. (2015)

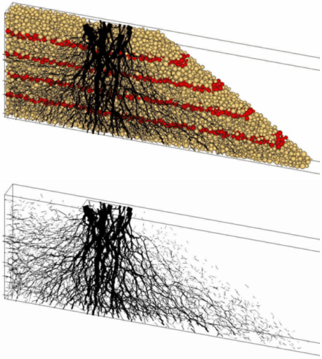
MD from the family of Particle Methods

Particle methods

- **SPH**
  Smooth-particle hydrodynamics (fluids, solids)
- **DEM**
  Discrete element method (granular matter)
- **LBM**
  Lattice Boltzmann Method (fluids)
- Multi-body gravity methods (space scale systems)

Common algorithms

- Search and detection
- Data structure
- Parallelization



DEM simulation
Fabio Gabrieli (University of Padova)
`geotechlab.wordpress.com`

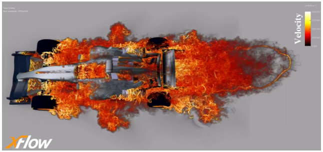
MD from the family of Particle Methods

Particle methods

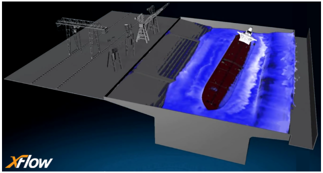
- **SPH**
  Smooth-particle hydrodynamics (fluids, solids)
- **DEM**
  Discrete element method (granular matter)
- **LBM**
  Lattice Boltzmann Method (fluids)
- Multi-body gravity methods (space scale systems)

Common algorithms

- Search and detection
- Data structure
- Parallelization



Formula 1 simulation



Ship launch simulation

BLM simulations
`www.xflowcfd.com`

Examples of simple pair potentials

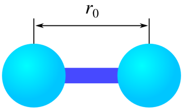
- **Short-range potentials (possible cut-off, fast)**
  - Lennard-Jones potential $U(r_{ij}) = \alpha\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^n - \left(\frac{\sigma}{r_{ij}}\right)^m\right], \quad m < n$
  - Morse potential $U(r_{ij}) = \alpha\left[1 - \exp(-\beta(r_{ij} - r_0)\right]^2$
  - Van der Waals potential $U(r_{ij}) = -\alpha\varepsilon\left(\frac{\sigma}{r_{ij}}\right)^6$
- **Long-range potentials (cut-off prohibited, slow)**
  - Gravitational potential $U(r_{ij}) = -G\frac{m_1 m_2}{r_{ij}}$
  - Electrostatic (Coulomb) potential : $U(r_{ij}) = -\frac{1}{4\pi\varepsilon_0}\frac{q_1 q_2}{r_{ij}}$
  - Elastic (harmonic) potential : $U(r_{ij}) = \frac{k}{2}(r_{ij} - r_0)^2$
- **Regularization**

$$\frac{1}{r_{ij}} \sim \frac{1}{\sqrt{r_{ij}^2 + \varepsilon^2}}$$

Examples of simple molecular models

- Covalent bonds approximated by harmonic potential

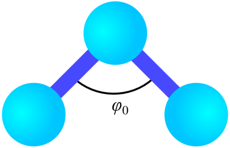
$$V_l(\boldsymbol{r}_1,\boldsymbol{r}_2) = \tfrac{k_l}{2}(|\boldsymbol{r}_1-\boldsymbol{r}_2| - r_0)^2$$



- In-plane angular potential

$$V_a(\boldsymbol{r}_1,\boldsymbol{r}_2,\boldsymbol{r}_3) = \tfrac{k_a}{2}(1-\cos(\phi-\phi_0))^2$$

$$V_a(\boldsymbol{r}_1,\boldsymbol{r}_2,\boldsymbol{r}_3) \approx \tfrac{k_a}{2}(\phi-\phi_0)^2$$

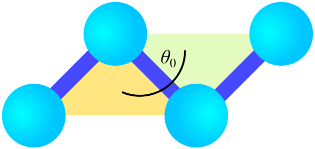


- Torsional potential

$$V_t(\boldsymbol{r}_1,\boldsymbol{r}_2,\boldsymbol{r}_3,\boldsymbol{r}_4) \approx \tfrac{k_t}{2}(\theta-\theta_0)^2$$

- Intra-molecular potential

$$V_m(\boldsymbol{r}_1,\boldsymbol{r}_2,\boldsymbol{r}_3,\ldots,\boldsymbol{r}_n) =$$
$$\tfrac{1}{2}\left(\textstyle\sum_{i=1}^{n-1} V_l(r^i) + \sum_{i=1}^{n-2} V_a(\phi^i) + \sum_{i=1}^{n-3} V_t(\theta^i)\right)$$



- **Total potential is complemented by an interaction potential with other molecules**

$$V(\boldsymbol{r}) = \sum_{\text{molecules}} V_m(\boldsymbol{r}_1,\boldsymbol{r}_2,\ldots,\boldsymbol{r}_n) + V_i(\boldsymbol{r})$$

Many-body Hamiltonian system

General algorithm

- Potential :

  $U(r_{ij})$

- System pair potential :

  $$V(\boldsymbol{r}) = \sum_{\forall i,j:\, i<j} U(|\boldsymbol{r}_i - \boldsymbol{r}_j|)$$

- Compute force $\boldsymbol{f}_0$ on particle $\boldsymbol{r}_0$:

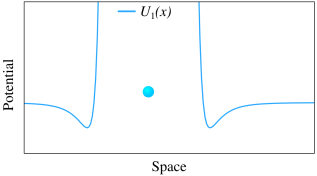
  $\boldsymbol{F}_0 = -\nabla_{\boldsymbol{r}_0} V(\boldsymbol{r}) = -\sum_{j\neq 0} \nabla_{\boldsymbol{r}_0} U(|\boldsymbol{r}_0 - \boldsymbol{r}_j|)$

- 2^{nd} Newton's law :

  $\ddot{\boldsymbol{r}}_0 = \frac{1}{m_0}\boldsymbol{F}_0$

- Integrate in time :

  $\boldsymbol{r}_0(t) \to \boldsymbol{r}_0(t+\Delta t)$



Properties :

- Energy conservation

  $$E = \underbrace{\frac{1}{2}\sum_i m_i \dot{\boldsymbol{r}}_i^2}_{Kinetic} + \underbrace{V(\boldsymbol{r})}_{Potential}$$

Many-body Hamiltonian system

General algorithm

- Potential :

  $U(r_{ij})$

- System pair potential :

  $$V(\boldsymbol{r}) = \sum_{\forall i,j:\, i<j} U(|\boldsymbol{r}_i - \boldsymbol{r}_j|)$$

- Compute force $\boldsymbol{f}_0$ on particle $\boldsymbol{r}_0$:

  $\boldsymbol{F}_0 = -\nabla_{\boldsymbol{r}_0} V(\boldsymbol{r}) = -\sum_{j\neq 0} \nabla_{\boldsymbol{r}_0} U(|\boldsymbol{r}_0 - \boldsymbol{r}_j|)$

- 2^{nd} Newton's law :

  $\ddot{\boldsymbol{r}}_0 = \frac{1}{m_0}\boldsymbol{F}_0$

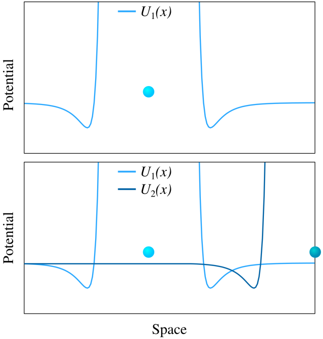
- Integrate in time :

  $\boldsymbol{r}_0(t) \to \boldsymbol{r}_0(t+\Delta t)$

Properties :

- Energy conservation

  $$E = \underbrace{\frac{1}{2}\sum_i m_i \dot{\boldsymbol{r}}_i^2}_{Kinetic} + \underbrace{V(\boldsymbol{r})}_{Potential}$$



Many-body Hamiltonian system

General algorithm

- Potential :

  $U(r_{ij})$

- System pair potential :

  $$V(\boldsymbol{r}) = \sum_{\forall i,j:\, i<j} U(|\boldsymbol{r}_i - \boldsymbol{r}_j|)$$

- Compute force f_0 on particle $\boldsymbol{r}_0$:

  $\boldsymbol{F}_0 = -\nabla_{\boldsymbol{r}_0} V(\boldsymbol{r}) = -\sum_{j\neq 0} \nabla_{\boldsymbol{r}_0} U(|\boldsymbol{r}_0 - \boldsymbol{r}_j|)$

- 2^{nd} Newton's law :

  $\ddot{\boldsymbol{r}}_0 = \frac{1}{m_0}\boldsymbol{F}_0$

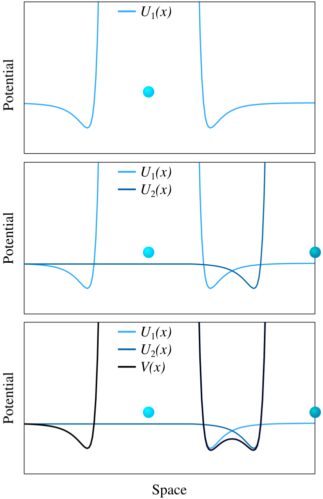
- Integrate in time :

  $\boldsymbol{r}_0(t) \to \boldsymbol{r}_0(t+\Delta t)$

Properties :

- Energy conservation

  $$E = \underbrace{\frac{1}{2}\sum_i m_i \dot{\boldsymbol{r}}_i^2}_{Kinetic} + \underbrace{V(\boldsymbol{r})}_{Potential}$$



Many-body Hamiltonian system

General algorithm

- Potential :

  $U(r_{ij})$

- System pair potential :

  $$V(\boldsymbol{r}) = \sum_{\forall i,j:\, i<j} U(|\boldsymbol{r}_i - \boldsymbol{r}_j|)$$

- Compute force $\boldsymbol{f}_0$ on particle $\boldsymbol{r}_0$:

  $$\boldsymbol{F}_0 = -\nabla_{\boldsymbol{r}_0} V(\boldsymbol{r}) = -\sum_{j\neq 0} \nabla_{\boldsymbol{r}_0} U(|\boldsymbol{r}_0 - \boldsymbol{r}_j|)$$

- 2^{nd} Newton's law :

  $$\ddot{\boldsymbol{r}}_0 = \frac{1}{m_0}\boldsymbol{F}_0$$

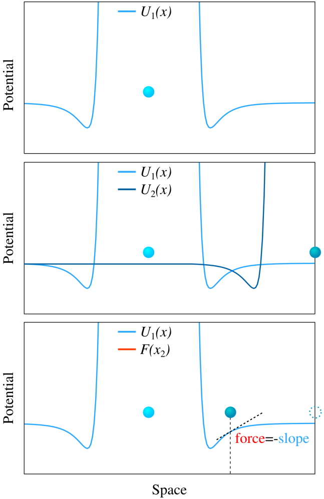
- Integrate in time :

  $$\boldsymbol{r}_0(t) \to \boldsymbol{r}_0(t+\Delta t)$$

Properties :

- Energy conservation

  $$E = \underbrace{\frac{1}{2}\sum_i m_i \dot{\boldsymbol{r}}_i^2}_{Kinetic} + \underbrace{V(\boldsymbol{r})}_{Potential}$$



Many-body Hamiltonian system

General algorithm

- Potential :

  $U(r_{ij})$

- System pair potential :

  $$V(\boldsymbol{r}) = \sum_{\forall i,j:\, i<j} U(|\boldsymbol{r}_i - \boldsymbol{r}_j|)$$

- Compute force f_0 on particle r_0 :

  $$\boldsymbol{F}_0 = -\nabla_{\boldsymbol{r}_0} V(\boldsymbol{r}) = -\sum_{j\neq 0} \nabla_{\boldsymbol{r}_0} U(|\boldsymbol{r}_0 - \boldsymbol{r}_j|)$$

- 2^{nd} Newton's law :

  $$\ddot{\boldsymbol{r}}_0 = \frac{1}{m_0}\boldsymbol{F}_0$$

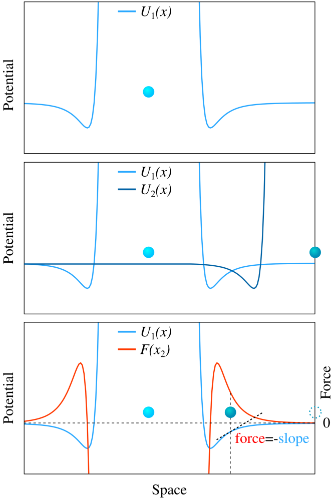
- Integrate in time :

  $$\boldsymbol{r}_0(t) \to \boldsymbol{r}_0(t+\Delta t)$$

Properties :

- Energy conservation

  $$E = \underbrace{\frac{1}{2}\sum_i m_i \dot{\boldsymbol{r}}_i^2}_{Kinetic} + \underbrace{V(\boldsymbol{r})}_{Potential}$$



Many-body Hamiltonian system

General algorithm

- Potential :

  $U(r_{ij})$

- System pair potential :

  $$V(\boldsymbol{r}) = \sum_{\forall i,j:\, i<j} U(|\boldsymbol{r}_i - \boldsymbol{r}_j|)$$

- Compute force f_0 on particle $\boldsymbol{r}_0$:

  $$\boldsymbol{F}_0 = -\nabla_{\boldsymbol{r}_0} V(\boldsymbol{r}) = -\sum_{j\neq 0} \nabla_{\boldsymbol{r}_0} U(|\boldsymbol{r}_0 - \boldsymbol{r}_j|)$$

- 2^{nd} Newton's law :

  $$\ddot{\boldsymbol{r}}_0 = \frac{1}{m_0}\boldsymbol{F}_0$$

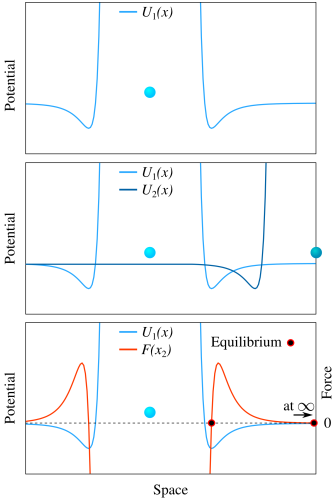
- Integrate in time :

  $$\boldsymbol{r}_0(t) \to \boldsymbol{r}_0(t+\Delta t)$$

Properties :

- Energy conservation

  $$E = \underbrace{\frac{1}{2}\sum_i m_i \dot{\boldsymbol{r}}_i^2}_{Kinetic} + \underbrace{V(\boldsymbol{r})}_{Potential}$$



Example : Lennard-Jones potential

Example :

- Lennard-Jones 6-12 (LJ 6-12) :

$$U(r_{ij}) = 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]$$

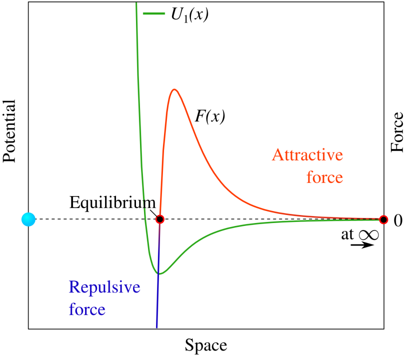
- Force :

$$\boldsymbol{F}_i(\boldsymbol{r}_i, \boldsymbol{r}_j) = -\nabla U(r_{ij}) =$$

$$= 24\varepsilon\left(\frac{\sigma}{r_{ij}}\right)^{6}\left[1 - 2\left(\frac{\sigma}{r_{ij}}\right)^{6}\right]\frac{\boldsymbol{r}_j - \boldsymbol{r}_i}{r_{ij}^2}$$

- Equilibrium :
  - At $T = 0$ K : $r_{ij}^e = 2^{1/6}\sigma$
  - At $T > 0$: $r_{ij}^e(T) > 2^{1/6}\sigma$
- Stable lattice : hcp (or fcc (111))
- Parameters :

σ - length units ~ lattice spacing
ε - energy units ~ bonding energy.



Example : Lennard-Jones potential

Example :

- Lennard-Jones 6-12 (LJ 6-12) :

$$U(r_{ij}) = 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]$$

- Force :

$$\boldsymbol{F}_i(\boldsymbol{r}_i, \boldsymbol{r}_j) = -\nabla U(r_{ij}) =$$

$$= 24\varepsilon\left(\frac{\sigma}{r_{ij}}\right)^{6}\left[1 - 2\left(\frac{\sigma}{r_{ij}}\right)^{6}\right]\frac{\boldsymbol{r}_j - \boldsymbol{r}_i}{r_{ij}^2}$$

- Equilibrium :

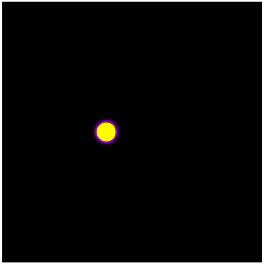
  • At $T = 0$ K : $r_{ij}^e = 2^{1/6}\sigma$

  • At $T > 0$: $r_{ij}^e(T) > 2^{1/6}\sigma$

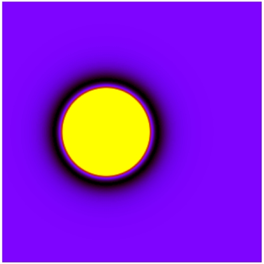
- Stable lattice : hcp (or fcc (111))
- Parameters :

  σ - length units ~ lattice spacing
  ε - energy units ~ bonding energy.

Particles



Potential



Example : Lennard-Jones potential

Example :

- Lennard-Jones 6-12 (LJ 6-12) :

$$U(r_{ij}) = 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]$$

- Force :

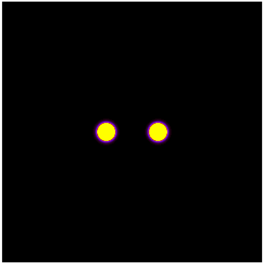
$$\boldsymbol{F}_i(\boldsymbol{r}_i,\boldsymbol{r}_j) = -\nabla U(r_{ij}) =$$

$$= 24\varepsilon\left(\frac{\sigma}{r_{ij}}\right)^{6}\left[1-2\left(\frac{\sigma}{r_{ij}}\right)^{6}\right]\frac{\boldsymbol{r}_j-\boldsymbol{r}_i}{r_{ij}^2}$$

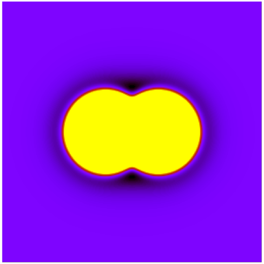
- Equilibrium :
  - At $T = 0$ K : $r_{ij}^e = 2^{1/6}\sigma$
  - At $T > 0$: $r_{ij}^e(T) > 2^{1/6}\sigma$
- Stable lattice : hcp (or fcc (111))
- Parameters :

  σ - length units ~ lattice spacing
  ε - energy units ~ bonding energy.

Particles



Potential



Example : Lennard-Jones potential

Example :

- Lennard-Jones 6-12 (LJ 6-12) :

$$U(r_{ij}) = 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]$$

- Force :

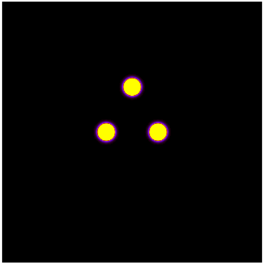
$$\boldsymbol{F}_i(\boldsymbol{r}_i, \boldsymbol{r}_j) = -\nabla U(r_{ij}) =$$

$$= 24\varepsilon\left(\frac{\sigma}{r_{ij}}\right)^{6}\left[1 - 2\left(\frac{\sigma}{r_{ij}}\right)^{6}\right]\frac{\boldsymbol{r}_j - \boldsymbol{r}_i}{r_{ij}^2}$$

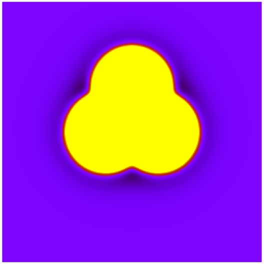
- Equilibrium :
  - At $T = 0$ K : $r_{ij}^e = 2^{1/6}\sigma$
  - At $T > 0$: $r_{ij}^e(T) > 2^{1/6}\sigma$
- Stable lattice : hcp (or fcc (111))
- Parameters :

  σ - length units ~ lattice spacing
  ε - energy units ~ bonding energy.

Particles



Potential



Example : Lennard-Jones potential

Example :

- Lennard-Jones 6-12 (LJ 6-12) :

$$U(r_{ij}) = 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]$$

- Force :

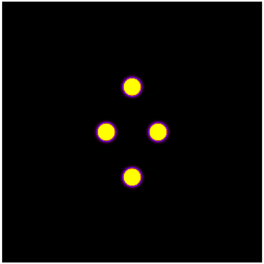
$$\boldsymbol{F}_i(\boldsymbol{r}_i,\boldsymbol{r}_j) = -\nabla U(r_{ij}) =$$

$$= 24\varepsilon\left(\frac{\sigma}{r_{ij}}\right)^{6}\left[1 - 2\left(\frac{\sigma}{r_{ij}}\right)^{6}\right]\frac{\boldsymbol{r}_j - \boldsymbol{r}_i}{r_{ij}^2}$$

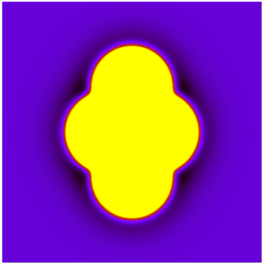
- Equilibrium :
  - At $T = 0$ K : $r^e_{ij} = 2^{1/6}\sigma$
  - At $T > 0$: $r^e_{ij}(T) > 2^{1/6}\sigma$
- Stable lattice : hcp (or fcc (111))
- Parameters :

  σ - length units ~ lattice spacing
  ε - energy units ~ bonding energy.

Particles



Potential



Example : Lennard-Jones potential

Example :

- Lennard-Jones 6-12 (LJ 6-12) :

$$U(r_{ij}) = 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]$$

- Force :

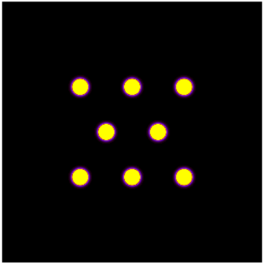
$$\boldsymbol{F}_i(\boldsymbol{r}_i,\boldsymbol{r}_j) = -\nabla U(r_{ij}) =$$

$$= 24\varepsilon\left(\frac{\sigma}{r_{ij}}\right)^{6}\left[1 - 2\left(\frac{\sigma}{r_{ij}}\right)^{6}\right]\frac{\boldsymbol{r}_j - \boldsymbol{r}_i}{r_{ij}^2}$$

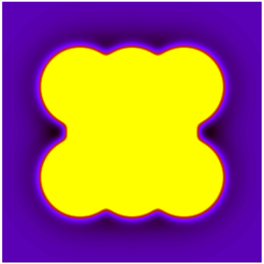
- Equilibrium :
  - • At $T = 0$ K : $r_{ij}^e = 2^{1/6}\sigma$
  - • At $T > 0$: $r_{ij}^e(T) > 2^{1/6}\sigma$
- Stable lattice : hcp (or fcc (111))
- Parameters :

  σ - length units ~ lattice spacing
  ε - energy units ~ bonding energy.

Particles



Potential



Example : Lennard-Jones potential

Example :

- Lennard-Jones 6-12 (LJ 6-12) :

$$U(r_{ij}) = 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]$$

- Force :

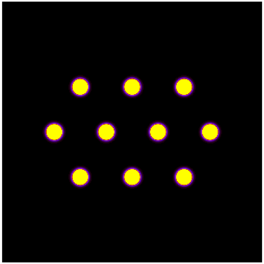
$$\boldsymbol{F}_i(\boldsymbol{r}_i,\boldsymbol{r}_j) = -\nabla U(r_{ij}) =$$

$$= 24\varepsilon\left(\frac{\sigma}{r_{ij}}\right)^{6}\left[1 - 2\left(\frac{\sigma}{r_{ij}}\right)^{6}\right]\frac{\boldsymbol{r}_j - \boldsymbol{r}_i}{r_{ij}^2}$$

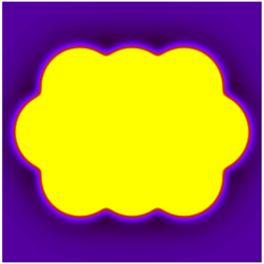
- Equilibrium :
  - At $T = 0$ K : $r_{ij}^e = 2^{1/6}\sigma$
  - At $T > 0$: $r_{ij}^e(T) > 2^{1/6}\sigma$
- Stable lattice : hcp (or fcc (111))
- Parameters :

  σ - length units ~ lattice spacing
  ε - energy units ~ bonding energy.

Particles



Potential



Example : Lennard-Jones potential

Example :

- Lennard-Jones 6-12 (LJ 6-12) :

$$U(r_{ij}) = 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]$$

- Force :

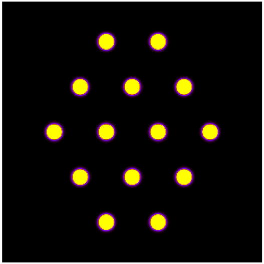
$$\boldsymbol{F}_i(\boldsymbol{r}_i,\boldsymbol{r}_j) = -\nabla U(r_{ij}) =$$

$$= 24\varepsilon\left(\frac{\sigma}{r_{ij}}\right)^{6}\left[1 - 2\left(\frac{\sigma}{r_{ij}}\right)^{6}\right]\frac{\boldsymbol{r}_j - \boldsymbol{r}_i}{r_{ij}^2}$$

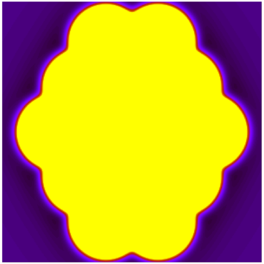
- Equilibrium :
  - At $T = 0$ K : $r_{ij}^e = 2^{1/6}\sigma$
  - At $T > 0$: $r_{ij}^e(T) > 2^{1/6}\sigma$
- Stable lattice : hcp (or fcc (111))
- Parameters :

  σ - length units ~ lattice spacing
  ε - energy units ~ bonding energy.

Particles



Potential

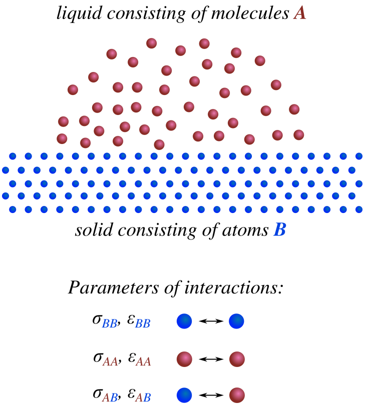


Mixing rule

- Consider a system containing 2 different atoms (molecules) : A, B
- We know ε_{AA}, σ_{AA} and ε_{BB}, σ_{BB}
- To compute energy and forces between atoms A and B we need σ_{AB} and ε_{AB}
- The classical mixing rule by Lorentz-Berthelot[1] :

$$\sigma_{AB} = \frac{1}{2}(\sigma_{AA} + \sigma_{BB})$$
$$\varepsilon_{AB} = \sqrt{\varepsilon_{AA}\varepsilon_{BB}}$$

- From algorithmic point of view one needs to check atom types
- For a liquid drop on surface, values of σ_{AB} and ε_{AB} can be obtained from the macroscopic value of the contact angle


*liquid consisting of molecules **A***
*solid consisting of atoms **B***
Parameters of interactions:
σ_{BB}, ε_{BB}
σ_{AA}, ε_{AA}
σ_{AB}, ε_{AB}

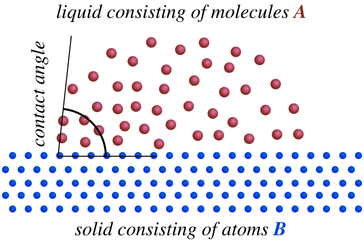
Mixing rule

- Consider a system containing 2 different atoms (molecules) : A, B
- We know ε_{AA}, σ_{AA} and ε_{BB}, σ_{BB}
- To compute energy and forces between atoms A and B we need σ_{AB} and ε_{AB}
- The classical mixing rule by Lorentz-Berthelot[1] :

$$\sigma_{AB} = \frac{1}{2}(\sigma_{AA} + \sigma_{BB})$$

$$\varepsilon_{AB} = \sqrt{\varepsilon_{AA}\varepsilon_{BB}}$$

- From algorithmic point of view one needs to check atom types
- For a liquid drop on surface, values of σ_{AB} and ε_{AB} can be obtained from the macroscopic value of the contact angle



Parameters of interactions:

$\sigma_{BB}, \varepsilon_{BB}$ ● ↔ ●

$\sigma_{AA}, \varepsilon_{AA}$ ● ↔ ●

$\sigma_{AB}, \varepsilon_{AB}$ ● ↔ ●

Short-range potentials and a cutoff

- Short-range potential

  $$V \sim \frac{1}{r_{ij}^{\alpha}}, \quad \alpha < \text{dim}$$

- System pair potential :

  $$V(\boldsymbol{r}) = \sum_{\forall i,j:\, i<j} U(|\boldsymbol{r}_i - \boldsymbol{r}_j|)$$

- Complexity of the force evaluation : $O(N^2)$
- First simplification, for two particles :

  $$\boldsymbol{F}_{ij} = -\boldsymbol{F}_{ji}$$

- Critical simplification : **cutoff radius** r_{cut} :

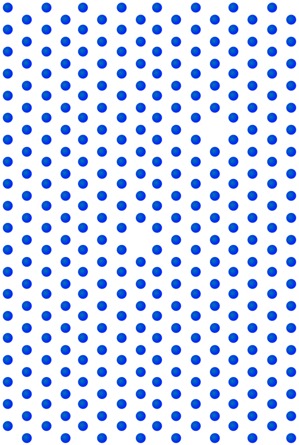
  $$U(r_{ij}) = \begin{cases} 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right], & \text{if } r_{ij} \le r_{\text{cut}} \\ 0, & \text{if } r_{ij} > r_{\text{cut}} \end{cases}$$

- Cutoff value : $r_{\text{cut}} > 2.5\sigma$
- Attention : truncated potential is discontinuous, additional errors are introduced.

Algorithm : linked-cell method

- Create a spatial grid $d \geq r_{\text{cut}}$
- Every cell contains a list of particles and a list of neighbouring cells
- Forces are evaluated in the cell and with respect to the neighbouring cells
- 3^{rd} Newton's law is used

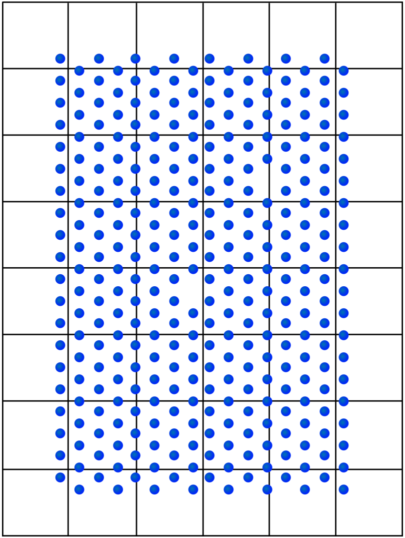
  Instead of checking 8 neighbouring cells, we check only 4.



Algorithm : linked-cell method

- Create a spatial grid $d \geq r_{\mathrm{cut}}$
- Every cell contains a list of particles and a list of neighbouring cells
- Forces are evaluated in the cell and with respect to the neighbouring cells
- 3^{rd} Newton's law is used

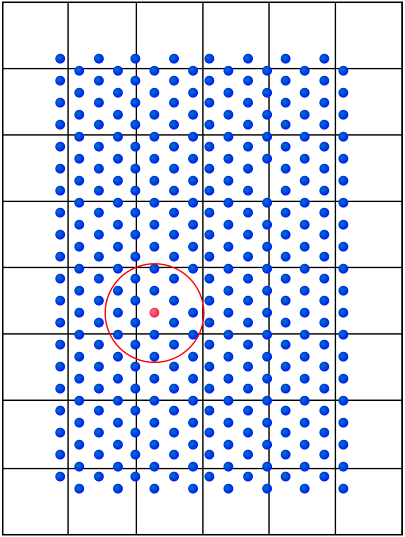
  Instead of checking 8 neighbouring cells, we check only 4.



Algorithm : linked-cell method

- Create a spatial grid $d \geq r_{\text{cut}}$
- Every cell contains a list of particles and a list of neighbouring cells
- Forces are evaluated in the cell and with respect to the neighbouring cells
- 3^{rd} Newton's law is used

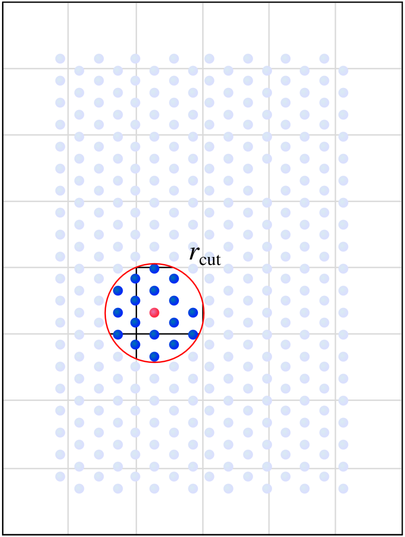
  Instead of checking 8 neighbouring cells, we check only 4.



Algorithm : linked-cell method

- Create a spatial grid $d \geq r_{\text{cut}}$
- Every cell contains a list of particles and a list of neighbouring cells
- Forces are evaluated in the cell and with respect to the neighbouring cells
- 3^{rd} Newton's law is used

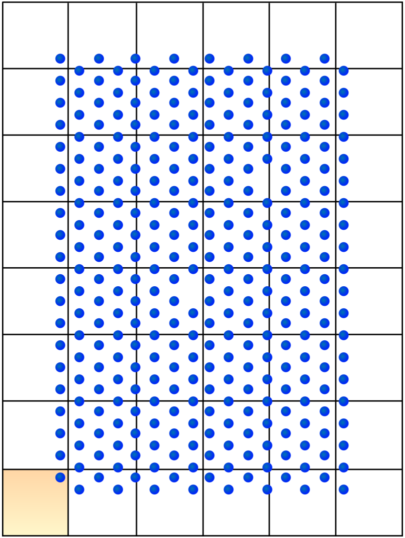
  Instead of checking 8 neighbouring cells, we check only 4.



Algorithm : linked-cell method

- Create a spatial grid $d \geq r_{\text{cut}}$
- Every cell contains a list of particles and a list of neighbouring cells
- Forces are evaluated in the cell and with respect to the neighbouring cells
- 3^{rd} Newton's law is used

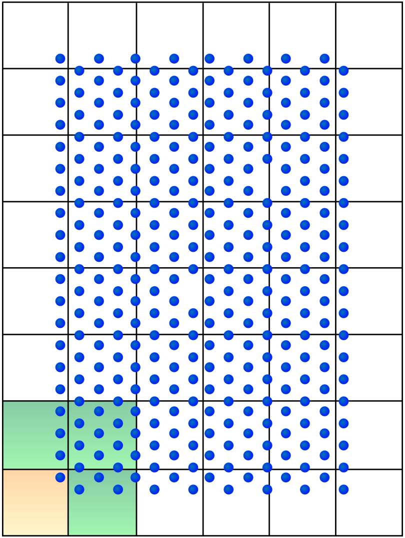
  Instead of checking 8 neighbouring cells, we check only 4.



Algorithm : linked-cell method

- Create a spatial grid $d \geq r_{\text{cut}}$
- Every cell contains a list of particles and a list of neighbouring cells
- Forces are evaluated in the cell and with respect to the neighbouring cells
- 3^{rd} Newton's law is used

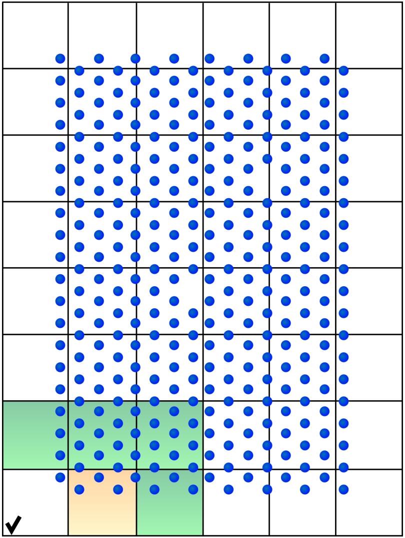
  Instead of checking 8 neighbouring cells, we check only 4.



Algorithm : linked-cell method

- Create a spatial grid $d \geq r_{\text{cut}}$
- Every cell contains a list of particles and a list of neighbouring cells
- Forces are evaluated in the cell and with respect to the neighbouring cells
- 3^{rd} Newton's law is used

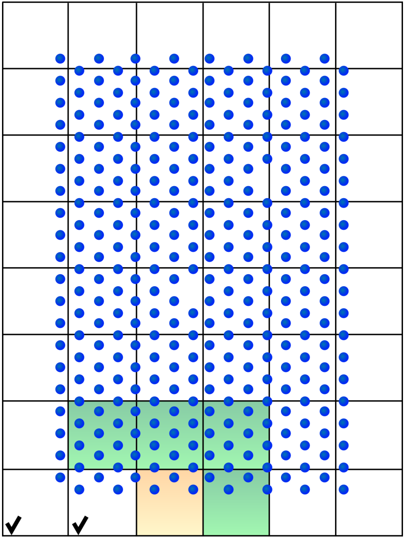
  Instead of checking 8 neighbouring cells, we check only 4.



Algorithm : linked-cell method

- Create a spatial grid $d \geq r_{\text{cut}}$
- Every cell contains a list of particles and a list of neighbouring cells
- Forces are evaluated in the cell and with respect to the neighbouring cells
- 3^{rd} Newton's law is used

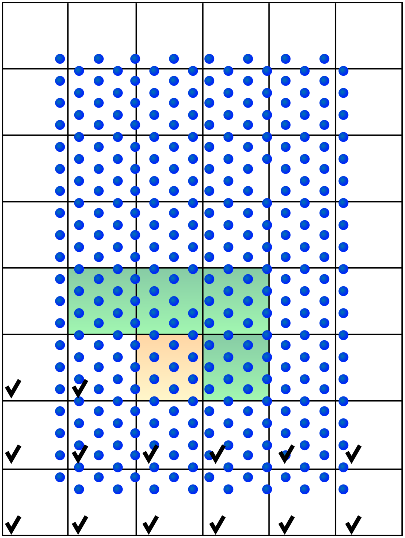
  Instead of checking 8 neighbouring cells, we check only 4.



Algorithm : linked-cell method

- Create a spatial grid $d \geq r_{\text{cut}}$
- Every cell contains a list of particles and a list of neighbouring cells
- Forces are evaluated in the cell and with respect to the neighbouring cells
- 3rd Newton's law is used

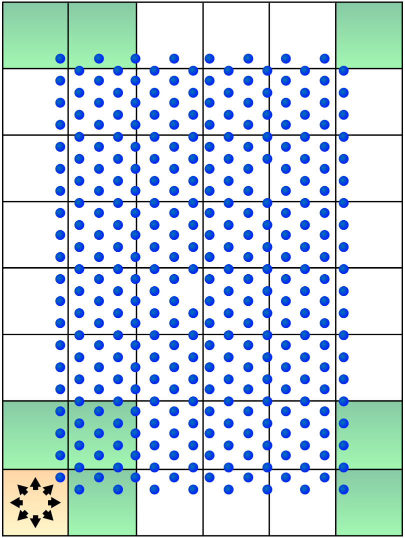
  Instead of checking 8 neighbouring cells, we check only 4.



Algorithm : linked-cell method

- Create a spatial grid $d \geq r_{\text{cut}}$
- Every cell contains a list of particles and a list of neighbouring cells
- Forces are evaluated in the cell and with respect to the neighbouring cells
- 3^{rd} Newton's law is used

  Instead of checking 8 neighbouring cells, we check only 4.
- Case of periodic BC



Time integration : explicit Euler

- Initial value problem

$$\begin{cases} M\ddot{\boldsymbol{X}}(t) = \boldsymbol{F}(\boldsymbol{X}) \\ \boldsymbol{X}(0) = \boldsymbol{X}_0, \quad \dot{\boldsymbol{X}}(0) = \dot{\boldsymbol{X}}_0 \end{cases}$$

- Straight forward approach (**explicit Euler**)

Compute : $\boldsymbol{f}_i(\boldsymbol{x}(t))$

$$m\ddot{\boldsymbol{x}}_i = \boldsymbol{f}_i$$

$$m\frac{\dot{\boldsymbol{x}}_i(t+\Delta t)-\dot{\boldsymbol{x}}_i(t)}{\Delta t} = \boldsymbol{f}_i$$

Compute :

$$\boxed{\dot{\boldsymbol{x}}_i(t+\Delta t) = \dot{\boldsymbol{x}}_i(t) + \frac{\Delta t}{m_i}\boldsymbol{f}_i}$$

$$\dot{\boldsymbol{x}}_i(t+\Delta t) = \frac{\boldsymbol{x}_i(t+\Delta t)-\boldsymbol{x}_i(t)}{\Delta t}$$

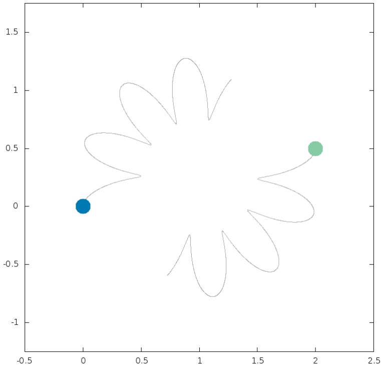
Compute :

$$\boxed{\boldsymbol{x}_i(t+\Delta t) = \boldsymbol{x}_i(t) + \dot{\boldsymbol{x}}_i(t)\Delta t}$$

- Let's see how fast it diverges

Example : $\sigma = 1, \varepsilon = 1, m = 1, v_0 = 0.2$

$\Delta t = 0.00001$



Time integration : explicit Euler

- Initial value problem

$$\begin{cases} M\ddot{X}(t) = F(X) \\ X(0) = X_0, \quad \dot{X}(0) = \dot{X}_0 \end{cases}$$

- Straight forward approach (**explicit Euler**)

Compute : $f_i(x(t))$

$$m\ddot{x}_i = f_i$$

$$m\frac{\dot{x}_i(t+\Delta t) - \dot{x}_i(t)}{\Delta t} = f_i$$

Compute :

$$\boxed{\dot{x}_i(t+\Delta t) = \dot{x}_i(t) + \frac{\Delta t}{m_i} f_i}$$

$$\dot{x}_i(t+\Delta t) = \frac{x_i(t+\Delta t) - x_i(t)}{\Delta t}$$

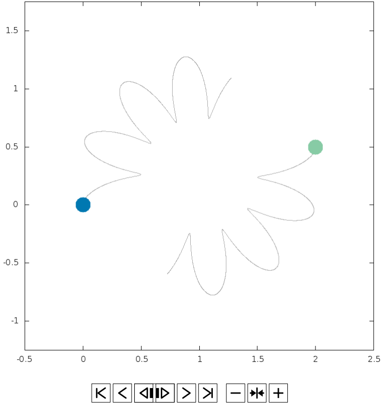
Compute :

$$\boxed{x_i(t+\Delta t) = x_i(t) + \dot{x}_i(t)\Delta t}$$

- Let's see how fast it diverges

Example : $\sigma = 1, \varepsilon = 1, m = 1, v_0 = 0.2$

$\Delta t = 0.00005$



Time integration : explicit Euler

- Initial value problem

$$\begin{cases} M\ddot{\boldsymbol{X}}(t) = \boldsymbol{F}(\boldsymbol{X}) \\ \boldsymbol{X}(0) = \boldsymbol{X}_0, \quad \dot{\boldsymbol{X}}(0) = \dot{\boldsymbol{X}}_0 \end{cases}$$

- Straight forward approach (**explicit Euler**)

Compute : $\boldsymbol{f}_i(\boldsymbol{x}(t))$

$$m\ddot{\boldsymbol{x}}_i = \boldsymbol{f}_i$$

$$m\frac{\dot{\boldsymbol{x}}_i(t+\Delta t)-\dot{\boldsymbol{x}}_i(t)}{\Delta t} = \boldsymbol{f}_i$$

Compute :

$$\boxed{\dot{\boldsymbol{x}}_i(t+\Delta t) = \dot{\boldsymbol{x}}_i(t) + \frac{\Delta t}{m_i}\boldsymbol{f}_i}$$

$$\dot{\boldsymbol{x}}_i(t+\Delta t) = \frac{\boldsymbol{x}_i(t+\Delta t)-\boldsymbol{x}_i(t)}{\Delta t}$$

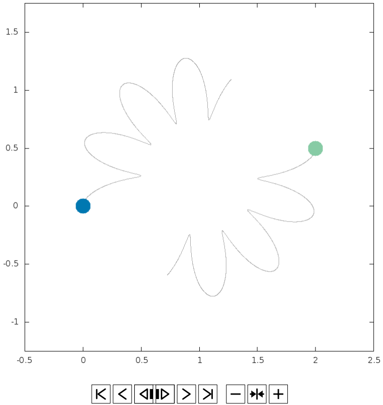
Compute :

$$\boxed{\boldsymbol{x}_i(t+\Delta t) = \boldsymbol{x}_i(t) + \dot{\boldsymbol{x}}_i(t)\Delta t}$$

- Let's see how fast it diverges

Example : $\sigma = 1, \varepsilon = 1, m = 1, v_0 = 0.2$

$\Delta t = 0.00010$



Time integration : semi-implicit Euler

- Initial value problem

$$\begin{cases} M\ddot{\boldsymbol{X}}(t) = \boldsymbol{F}(\boldsymbol{X}) \\ \boldsymbol{X}(0) = \boldsymbol{X}_0, \quad \dot{\boldsymbol{X}}(0) = \dot{\boldsymbol{X}}_0 \end{cases}$$

- **A better approach** (**semi-implicit Euler**)

Compute : $\boldsymbol{f}_i(x(t))$

$$m\ddot{x}_i = \boldsymbol{f}_i$$

$$m\frac{\dot{x}_i(t+\Delta t)-\dot{x}_i(t)}{\Delta t} = \boldsymbol{f}_i$$

Compute :

$$\boxed{\dot{x}_i(t+\Delta t) = \dot{x}_i(t) + \frac{\Delta t}{m_i}\boldsymbol{f}_i}$$

$$\dot{x}_i(t+\Delta t) = \frac{x_i(t+\Delta t)-x_i(t)}{\Delta t}$$

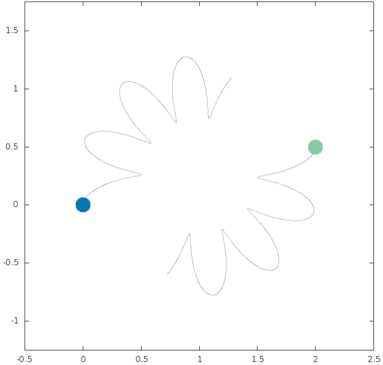
Compute :

$$\boxed{x_i(t+\Delta t) = x_i(t) + \dot{x}_i(t+\Delta t)\Delta t}$$

- Symplectic integrator ! In average it preserves the energy.

Example : $\sigma = 1, \varepsilon = 1, m = 1, v_0 = 0.2$

$\Delta t = 0.01$



Time integration : semi-implicit Euler

- Initial value problem

$$\begin{cases} M\ddot{\boldsymbol{X}}(t) = \boldsymbol{F}(\boldsymbol{X}) \\ \boldsymbol{X}(0) = \boldsymbol{X}_0, \quad \dot{\boldsymbol{X}}(0) = \dot{\boldsymbol{X}}_0 \end{cases}$$

- **A better approach** (**semi-implicit Euler**)

  Compute : $\boldsymbol{f}_i(\boldsymbol{x}(t))$

$$m\ddot{\boldsymbol{x}}_i = \boldsymbol{f}_i$$

$$m\frac{\dot{\boldsymbol{x}}_i(t+\Delta t) - \dot{\boldsymbol{x}}_i(t)}{\Delta t} = \boldsymbol{f}_i$$

  Compute :

$$\boxed{\dot{\boldsymbol{x}}_i(t+\Delta t) = \dot{\boldsymbol{x}}_i(t) + \frac{\Delta t}{m_i}\boldsymbol{f}_i}$$

$$\dot{\boldsymbol{x}}_i(t+\Delta t) = \frac{\boldsymbol{x}_i(t+\Delta t) - \boldsymbol{x}_i(t)}{\Delta t}$$

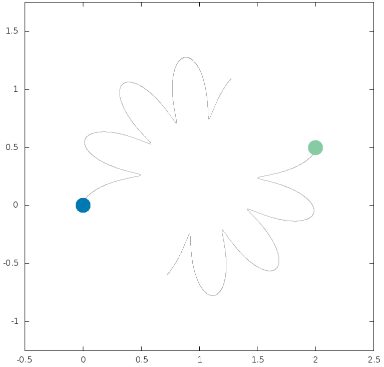
  Compute :

$$\boxed{\boldsymbol{x}_i(t+\Delta t) = \boldsymbol{x}_i(t) + \dot{\boldsymbol{x}}_i(t+\Delta t)\Delta t}$$

- Symplectic integrator ! In average it preserves the energy.

Example : $\sigma = 1, \varepsilon = 1, m = 1, v_0 = 0.2$

$\Delta t = 0.02$



Time integration : semi-implicit Euler

- Initial value problem

$$\begin{cases} M\ddot{X}(t) = F(X) \\ X(0) = X_0, \quad \dot{X}(0) = \dot{X}_0 \end{cases}$$

- **A better approach**
  (**semi-implicit Euler**)

  Compute : $f_i(x(t))$

$$m\ddot{x}_i = f_i$$

$$m\frac{\dot{x}_i(t+\Delta t) - \dot{x}_i(t)}{\Delta t} = f_i$$

  Compute :

$$\boxed{\dot{x}_i(t+\Delta t) = \dot{x}_i(t) + \frac{\Delta t}{m_i} f_i}$$

$$\dot{x}_i(t+\Delta t) = \frac{x_i(t+\Delta t) - x_i(t)}{\Delta t}$$

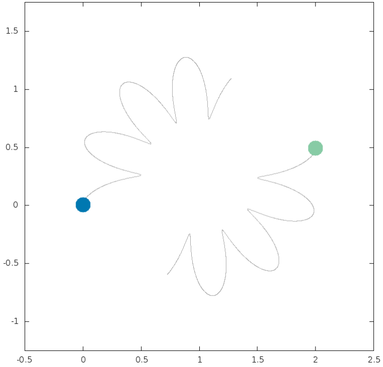
  Compute :

$$\boxed{x_i(t+\Delta t) = x_i(t) + \dot{x}_i(t+\Delta t)\Delta t}$$

- Symplectic integrator !
  In average it preserves the energy.

Example : $\sigma = 1, \varepsilon = 1, m = 1, v_0 = 0.2$

$\Delta t = 0.05$



Time integration : semi-implicit Euler

- Initial value problem

$$\begin{cases} M\ddot{X}(t) = F(X) \\ X(0) = X_0, \quad \dot{X}(0) = \dot{X}_0 \end{cases}$$

- **A better approach**

  (**semi-implicit Euler**)

  Compute : $f_i(x(t))$

$$m\ddot{x}_i = f_i$$

$$m\frac{\dot{x}_i(t+\Delta t)-\dot{x}_i(t)}{\Delta t} = f_i$$

  Compute :

$$\boxed{\dot{x}_i(t+\Delta t) = \dot{x}_i(t) + \frac{\Delta t}{m_i} f_i}$$

$$\dot{x}_i(t+\Delta t) = \frac{x_i(t+\Delta t)-x_i(t)}{\Delta t}$$

  Compute :

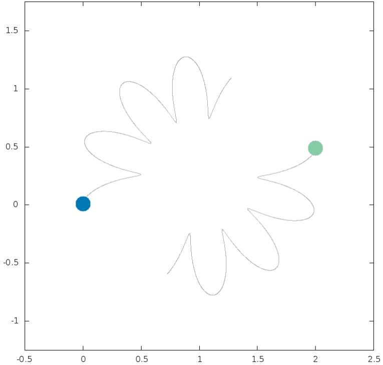
$$\boxed{x_i(t+\Delta t) = x_i(t) + \dot{x}_i(t+\Delta t)\Delta t}$$

- Symplectic integrator !

  In average it preserves the energy.

Example : $\sigma = 1, \varepsilon = 1, m = 1, v_0 = 0.2$

$\Delta t = 0.10$



Time integration : Verlet method

- Initial value problem

$$\begin{cases} M\ddot{\boldsymbol{X}}(t) = \boldsymbol{F}(\boldsymbol{X}) \\ \boldsymbol{X}(0) = \boldsymbol{X}_0, \quad \dot{\boldsymbol{X}}(0) = \dot{\boldsymbol{X}}_0 \end{cases}$$

- Velocity-Verlet method[1]

Compute : $\boxed{\boldsymbol{x}_i(t+\Delta t) = \boldsymbol{x}_i(t) + \left[\dot{\boldsymbol{x}}_i(t) + \frac{\Delta t}{2m_i}\boldsymbol{f}_i(t)\right]\Delta t}$

Store $\boldsymbol{f}_i(t)$

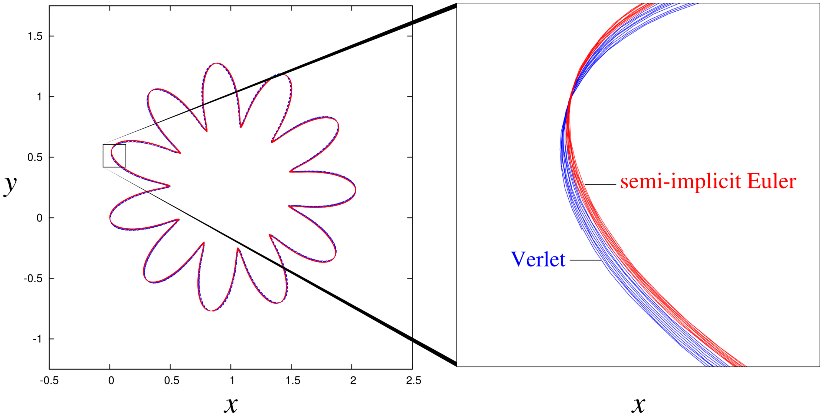
Compute : $\boldsymbol{f}_i(t+\Delta t) = \boldsymbol{f}_i(\boldsymbol{x}(t+\Delta t))$

Compute : $\boxed{\dot{\boldsymbol{x}}_i(t+\Delta t) = \dot{\boldsymbol{x}}_i(t) + \frac{\Delta t}{2m_i}\left[\boldsymbol{f}_i(t) + \boldsymbol{f}_i(t+\Delta t)\right]}$

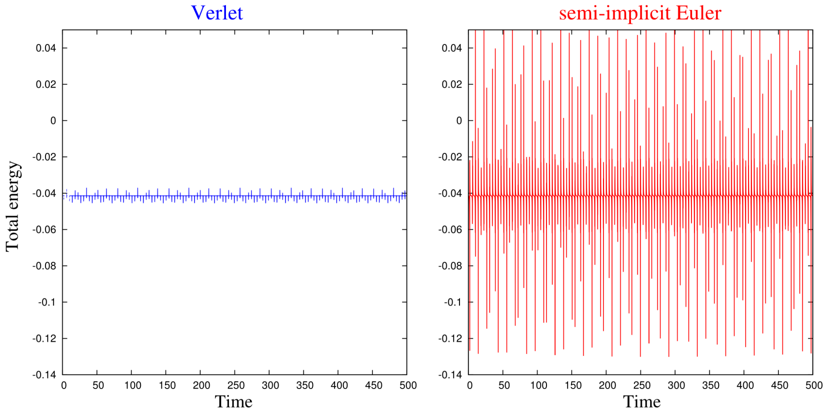
- Requires additional storage for $\boldsymbol{f}_i(t)$.
- Symplectic integrator !
  In average it preserves the energy.

[1] Verlet L. "Computer Experiments on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules". Phys Rev (1967)

Comparison Verlet vs Euler



Comparison Verlet vs Euler



- *Explicit Euler* method is of no use
- Both *Velocity-Verlet* method and *semi-implicit Euler* methods are simplectic, i.e. in average they preserve the system energy
- *Velocity-Verlet* has better energy preserving properties

General algorithm

- **Initialize :**
  1. distribute particles $x_i(0)$ for $i \in [0, N]$
  2. assign initial velocity field $\dot{x}_i(0)$
  3. assign boundary conditions
  4. evaluate forces on particles $f_i(x(0))$
- **Integrate in time (velocity Verlet method) :**
  1. $t \to t + \Delta t$
  2. update boundary conditions
  3. compute new positions

     $$x_i(t + \Delta t) = x_i(t) + \left[\dot{x}_i(t) + \frac{\Delta t}{2m_i} f_i(t)\right] \Delta t$$
  4. store forces $f_i(t)$
  5. evaluate new forces (using, e.g., linked-cell method) $f_i(x(t + \Delta t))$
  6. compute new velocities

     $$\dot{x}_i(t + \Delta t) = \dot{x}_i(t) + \frac{\Delta t}{2m_i}\left[f_i(t) + f_i(t + \Delta t)\right]$$
  7. if needed store data and compute energies.

General algorithm

- **Initialize :**
  1. **distribute particles** $x_i(0)$ **for** $i \in [0, N]$
  2. assign initial velocity field $\dot{x}_i(0)$
  3. **assign boundary conditions**
  4. evaluate forces on particles $f_i(x(0))$
- **Integrate in time (velocity Verlet method) :**
  1. $t \to t + \Delta t$
  2. update boundary conditions
  3. compute new positions

     $x_i(t + \Delta t) = x_i(t) + \left[\dot{x}_i(t) + \frac{\Delta t}{2m_i} f_i(t)\right] \Delta t$
  4. store forces $f_i(t)$
  5. evaluate new forces (using, e.g., linked-cell method) $f_i(x(t + \Delta t))$
  6. compute new velocities

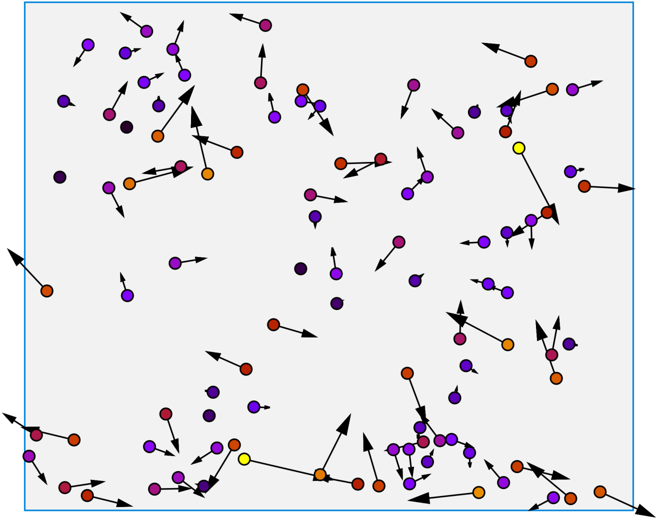
     $\dot{x}_i(t + \Delta t) = \dot{x}_i(t) + \frac{\Delta t}{2m_i} \left[f_i(t) + f_i(t + \Delta t)\right]$
  7. if needed store data and compute energies.

Boundary conditions I

Animation `pbc.gif`

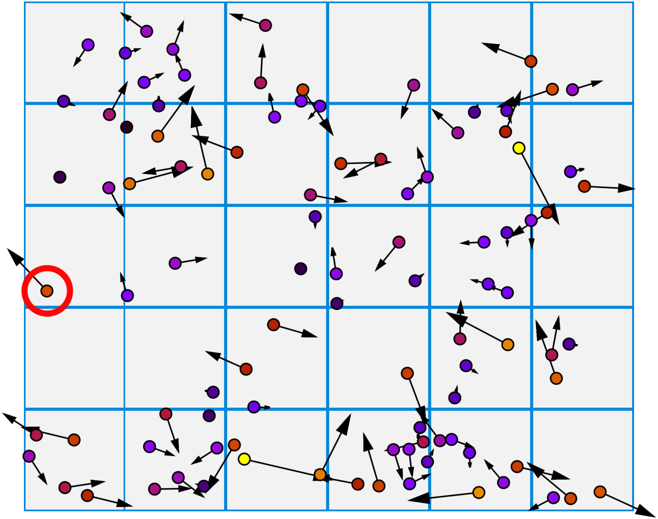
Boundary conditions I

- Periodic boundary conditions



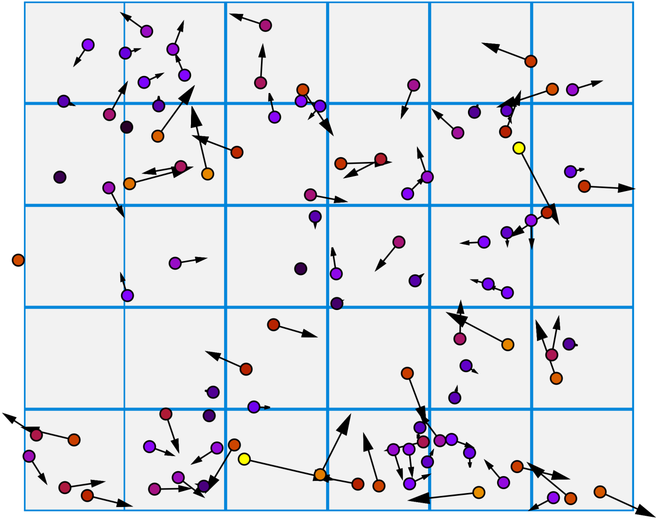
Boundary conditions I

- Periodic boundary conditions



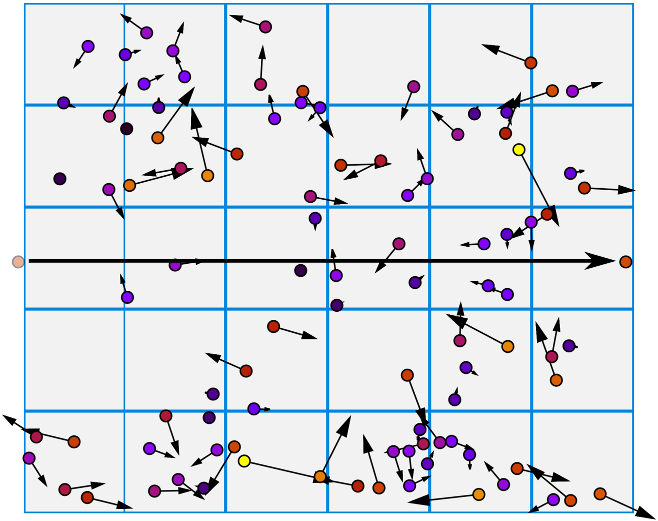
Boundary conditions I

- Periodic boundary conditions



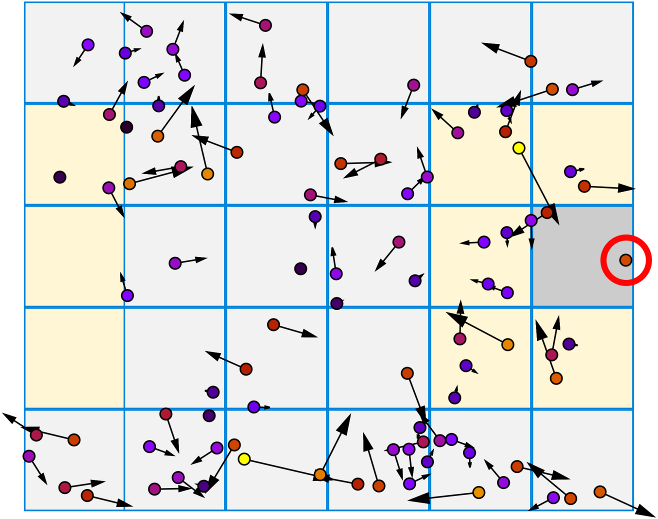
Boundary conditions I

- Periodic boundary conditions



Boundary conditions I

- Periodic boundary conditions

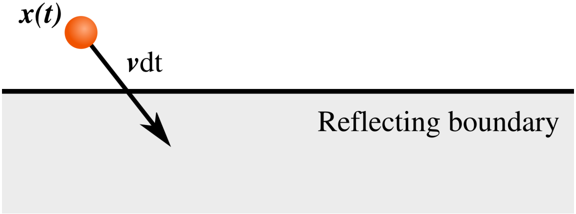


Boundary conditions II

Animation `rbc.gif`

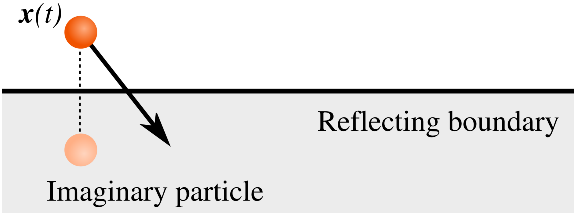
Boundary conditions II

- Reflecting boundary conditions



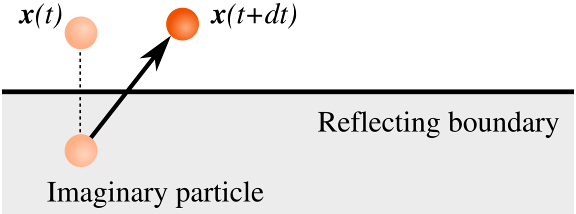
Boundary conditions II

- Reflecting boundary conditions



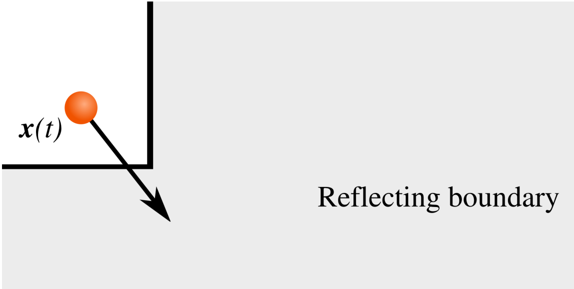
Boundary conditions II

- Reflecting boundary conditions



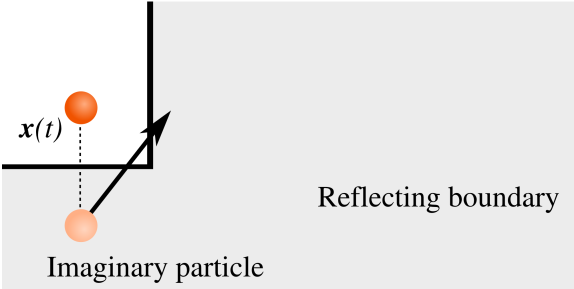
Boundary conditions II

- Reflecting boundary conditions



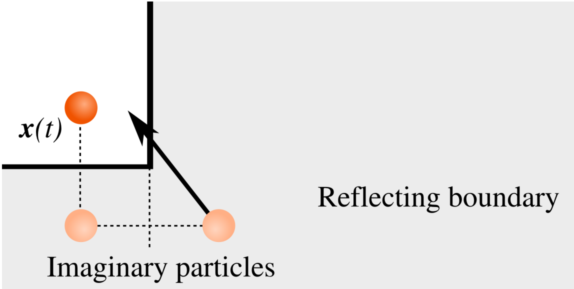
Boundary conditions II

- Reflecting boundary conditions



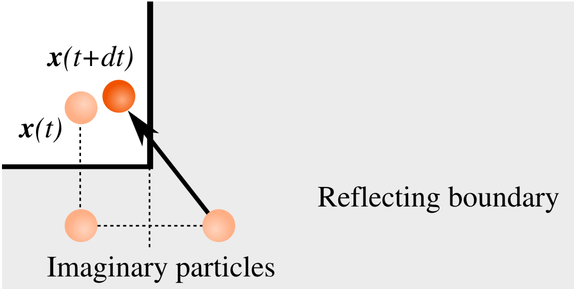
Boundary conditions II

- Reflecting boundary conditions



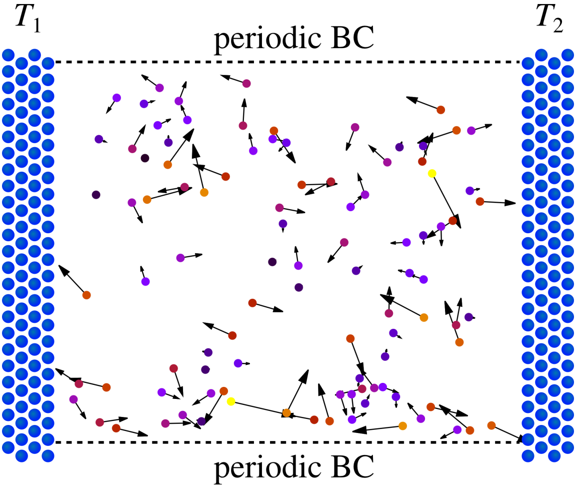
Boundary conditions II

- Reflecting boundary conditions



Boundary conditions II

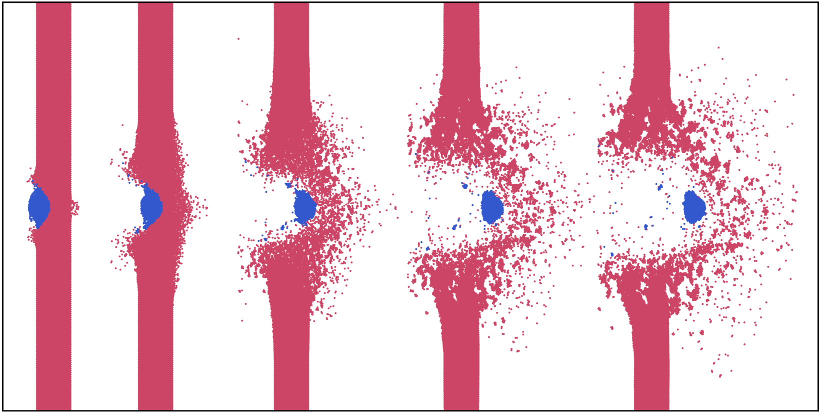
- Reflecting boundary conditions (different approach)



- Rigid walls of immobile atoms (only repulsive or combined action)
- Or walls of moving atoms at certain temperature

Boundary conditions III

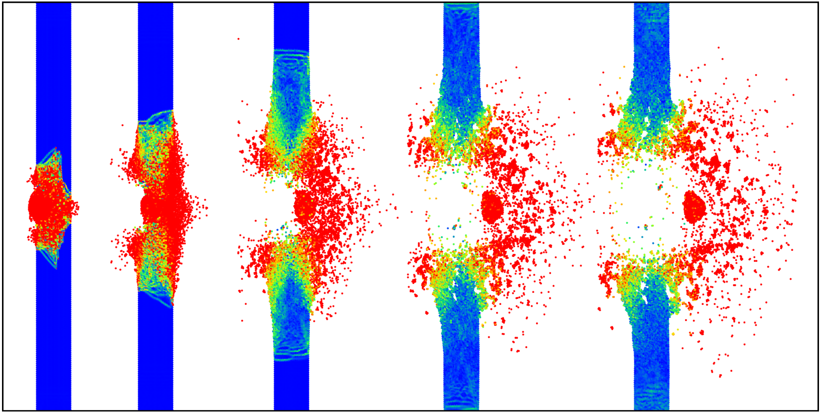
- Initial velocity (initial value problem) : impact, penetration



- Volumetric forces : gravity (additional force $F_i \mathrel{+}= m_i g$)

Boundary conditions III

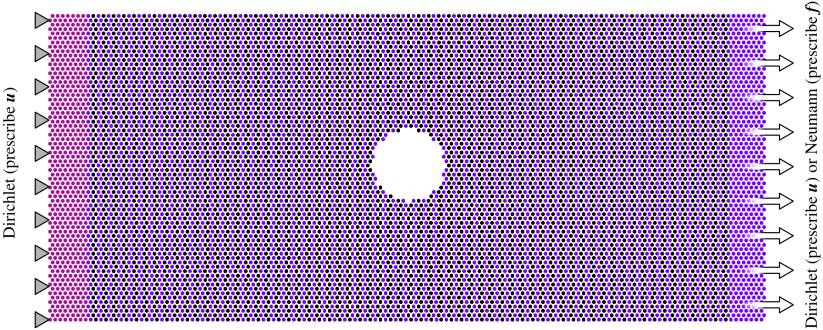
- Initial velocity (initial value problem) : impact, penetration



- Volumetric forces : gravity (additional force $F_i \mathrel{+}= m_i g$)

Boundary conditions IV

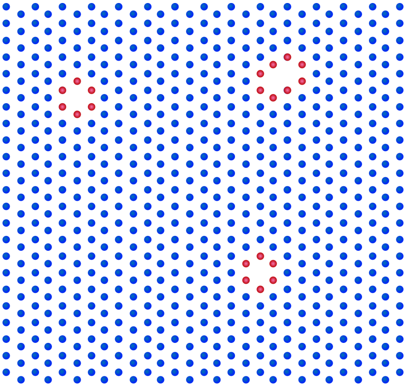
- Mechanical boundary conditions : Dirichlet and Neumann


Dirichlet (prescribe $\boldsymbol{u}$)
Dirichlet (prescribe $\boldsymbol{u}$) or Neumann (prescribe $\boldsymbol{f}$)

Initial configuration

Simple configurations :

- Gas/liquid
- Perfect crystal
- Crystal with vacancy defects
- Bi-crystals
- Stacking faults and dislocations
- Simple geometries
- Long molecules

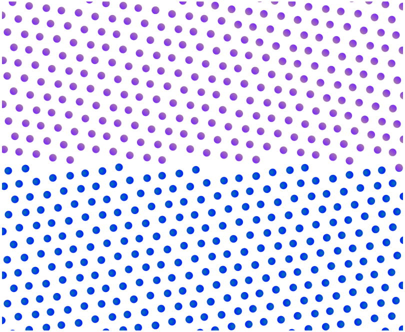


Crystal with vacancy defects (easy to control)

Initial configuration

Simple configurations :

- Gas/liquid
- Perfect crystal
- Crystal with vacancy defects
- Bi-crystals[1]
- Stacking faults and dislocations
- Simple geometries
- Long molecules



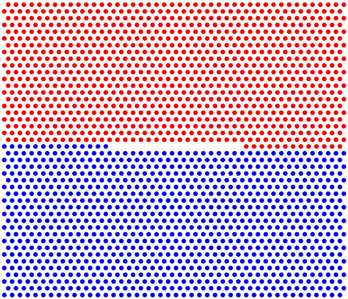
Bi-Crystal (grain boundary)

[1] Coffman & Sethna. Grain boundary energies and cohesive strength as a function of geometry. Phys Rev B 77 (2008)

Initial configuration

Simple configurations :

- Gas/liquid
- Perfect crystal
- Crystal with vacancy defects
- Bi-crystals
- Stacking faults and dislocations
- Simple geometries
- Long molecules

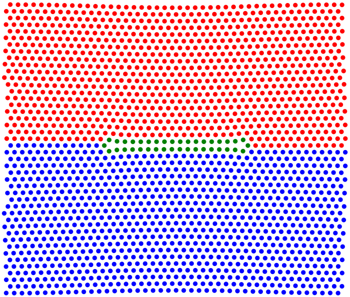


Remove several atoms

Initial configuration

Simple configurations :

- Gas/liquid
- Perfect crystal
- Crystal with vacancy defects
- Bi-crystals
- Stacking faults and dislocations
- Simple geometries
- Long molecules

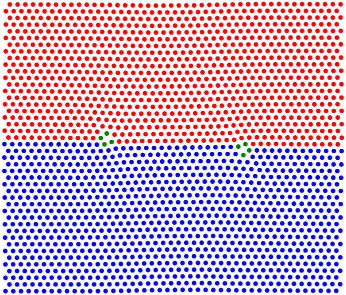


Stacking fault with partial dislocations

Initial configuration

Simple configurations :

- Gas/liquid
- Perfect crystal
- Crystal with vacancy defects
- Bi-crystals
- Stacking faults and dislocations
- Simple geometries
- Long molecules

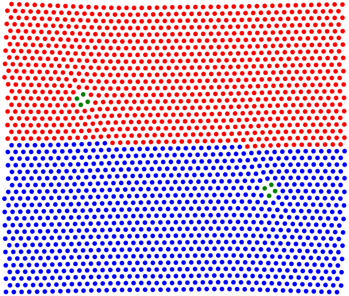


Healing stacking fault forms two perfect edge dislocations

Initial configuration

Simple configurations :

- Gas/liquid
- Perfect crystal
- Crystal with vacancy defects
- Bi-crystals
- Stacking faults and dislocations
- Simple geometries
- Long molecules

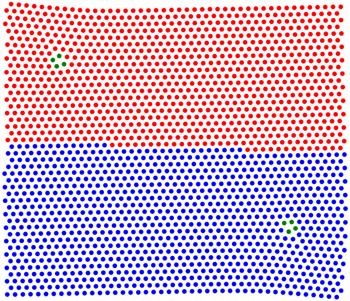


Dislocations glide

Initial configuration

Simple configurations :

- Gas/liquid
- Perfect crystal
- Crystal with vacancy defects
- Bi-crystals
- Stacking faults and dislocations
- Simple geometries
- Long molecules

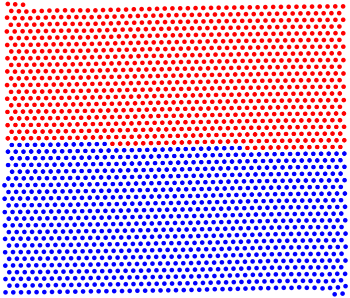


Dislocations glide

Initial configuration

Simple configurations :

- Gas/liquid
- Perfect crystal
- Crystal with vacancy defects
- Bi-crystals
- Stacking faults and dislocations
- Simple geometries
- Long molecules

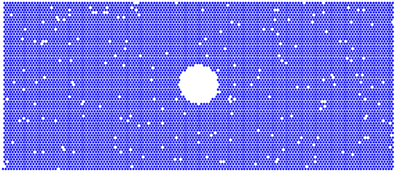


Dislocations form steps on the surface

Initial configuration

Simple configurations :

- Gas/liquid
- Perfect crystal
- Crystal with vacancy defects
- Bi-crystals
- Stacking faults and dislocations
- Simple geometries
- Long molecules

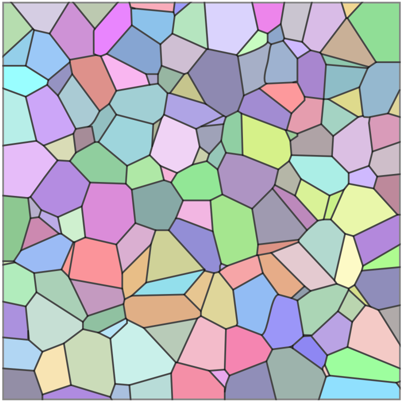


Layer with a circular hole

Initial configuration

Physically based configurations :

- Amorphous solid
  rapidly solidified from a liquid
- Voronoi-based polycrystal
- Polycrystalline solid
  porosity and grain size are controlled by the cooling rate
- Hight-temperature corrosion
  heat up and cool down initial configuration

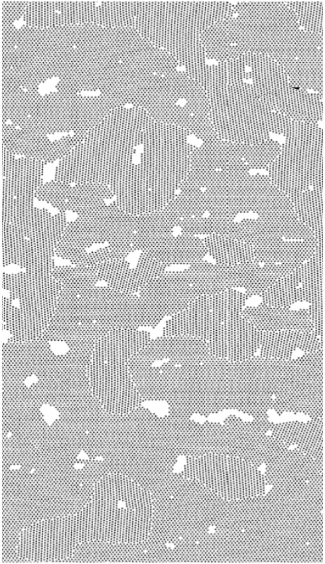


Voronoi tesselation as a basis for construction of a nano-grained material
(*adapted from Wikipedia*)

Initial configuration

Physically based configurations :

- Amorphous solid
  rapidly solidified from a liquid
- Voronoi-based polycrystal
- Polycrystalline solid
  porosity and grain size are controlled by the cooling rate
- Hight-temperature corrosion
  heat up and cool down initial configuration



Porous polycrystal obtained from liquid state by relatively fast cooling

General algorithm

- **Initialize :**
  1. distribute particles $x_i(0)$ for $i \in [0, N]$
  2. assign initial velocity field $\dot{x}_i(0)$
  3. assign boundary conditions
  4. evaluate forces on particles $f_i(x(0))$
- **Integrate in time (velocity Verlet method) :**
  1. $t \to t + \Delta t$
  2. update boundary conditions
  3. compute new positions

     $$x_i(t + \Delta t) = x_i(t) + \left[\dot{x}_i(t) + \frac{\Delta t}{2m_i} f_i(t)\right] \Delta t$$
  4. store forces $f_i(t)$
  5. evaluate new forces (using, e.g., linked-cell method) $f_i(x(t + \Delta t))$
  6. compute new velocities

     $$\dot{x}_i(t + \Delta t) = \dot{x}_i(t) + \frac{\Delta t}{2m_i} \left[f_i(t) + f_i(t + \Delta t)\right]$$
  7. if needed store data and compute energies.

End of part I

End of part I