SPC Water Model

Semih Can Sancar, Puneeth Kouloorkar, Dibya Sahay, Rushabh Chouhan

Freie Universität Berlin

February 12, 2020

Table of contents

SPC Model

- **S**imple **P**oint **C**harge

SPC Model

- **S**imple **P**oint **C**harge
  - Only external forces

SPC Model

- **S**imple **P**oint **C**harge
    - Only external forces
- Flexible model: bond length and angle can vary

SPC Model

- **S**imple **P**oint **C**harge
  - Only external forces
- Flexible model: bond length and angle can vary
  - Consider internal forces as well

Hamiltonian system

$$H(p, q) = T(p) + V_{int}(r) + V_{ext}(r) \tag{1}$$

- Energy conserving system

Hamiltonian system

$$H(p,q) = T(p) + V_{int}(r) + V_{ext}(r) \tag{1}$$

- Energy conserving system
- Components: Temperature, internal energy and external energy

$$\frac{\partial p}{\partial t} = -\frac{\partial H}{\partial q}$$
$$\frac{\partial q}{\partial t} = \frac{\partial H}{\partial p}$$

Temperature

- Depended on the particles momenta

Temperature

- Depended on the particles momenta
- Therefore depended on the kinetic energy

$$E_{kin} = \frac{1}{2} m v^2 \tag{2}$$

$$T = \frac{2E_{kin}}{3k_b} \tag{3}$$

Internal energy

- Valence potential

$$U_{valence}(\theta) = k_{valence}(\theta - \theta_0)^2 \tag{4}$$

- Harmonic bond potential

$$U_{bond}(r) = k_{angular}(r - r_0)^2 \tag{5}$$

- Forces are calculated by deriving the energy terms

External Energy

Lennard-Jones Potential

- Repulsive and attractive Energy terms

$$E_{LJ} = \left(\frac{A}{r_{ij}^{12}} - \frac{B}{r_{ij}^{6}}\right) \tag{6}$$

$$A = 4\epsilon \cdot \sigma^{12}, B = 4\epsilon \cdot \sigma^{6} \tag{7}$$

Coulomb Energy

- Repulsive energy

$$U_{Coulomb} = \frac{1}{4\pi\epsilon_0}\frac{q_1 q_2}{r_{ij}} \tag{8}$$

Periodic Boundary Condition

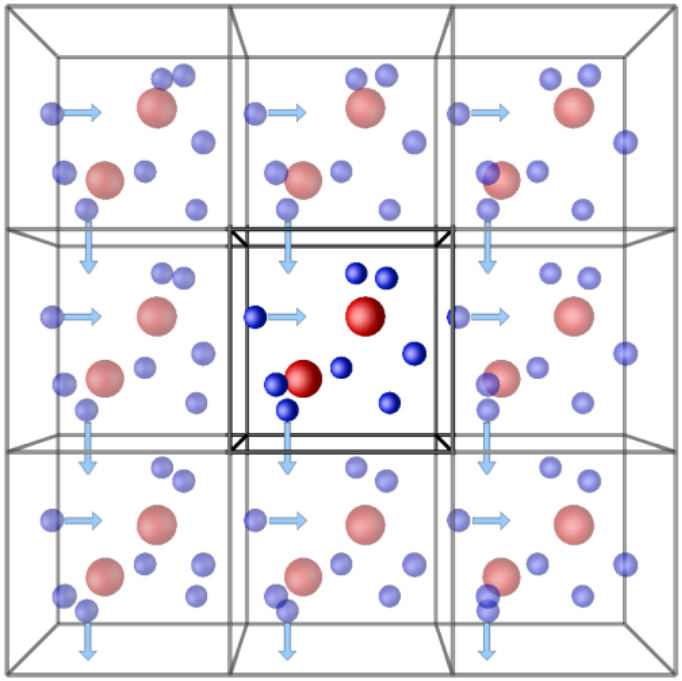

Figure: *http : //isaacs.sourceforge.net/phys/pbc.html*

Choice of numerical method

- Conservation of energy in Hamiltonian systems

Choice of numerical method

- Conservation of energy in Hamiltonian systems
  - Energy conservation through symplectic scheme

Choice of numerical method

- Conservation of energy in Hamiltonian systems
  - Energy conservation through symplectic scheme
  - **Velocity Verlet/Leap frog Scheme**
    - Widely used method in molecular dynamics [1]

Update velocity at half timestep:

Update velocity at half timestep:

$$v_{1/2} = v_0 + \frac{\Delta t}{2} a_0 \tag{9}$$

Update position with $v(\frac{\Delta t}{2})$:

Update velocity at half timestep:

$$v_{1/2} = v_0 + \frac{\Delta t}{2} a_0 \tag{9}$$

Update position with $v(\frac{\Delta t}{2})$:

$$r_1 = r_0 + \Delta t v_{1/2} + \frac{(\Delta t)^2}{2} a_0 \tag{10}$$

Update acceleration:

Update velocity at half timestep:

$$v_{1/2} = v_0 + \frac{\Delta t}{2} a_0 \tag{9}$$

Update position with $v(\frac{\Delta t}{2})$:

$$r_1 = r_0 + \Delta t v_{1/2} + \frac{(\Delta t)^2}{2} a_0 \tag{10}$$

Update acceleration:

$$a_1 = \frac{F_{internal} + F_{external}}{mass} \tag{11}$$

Update velocity at full timestep:

Update velocity at half timestep:

$$v_{1/2} = v_0 + \frac{\Delta t}{2} a_0 \tag{9}$$

Update position with $v(\frac{\Delta t}{2})$:

$$r_1 = r_0 + \Delta t v_{1/2} + \frac{(\Delta t)^2}{2} a_0 \tag{10}$$

Update acceleration:

$$a_1 = \frac{F_{internal} + F_{external}}{mass} \tag{11}$$

Update velocity at full timestep:

$$v_1 = v_{1/2} + \frac{\Delta t}{2} a_1 \tag{12}$$

Software structure

- Split the tasks into three main code sections:

Software structure

- Split the tasks into three main code sections:
  - particle.py: Implementation of particle functions

Software structure

- Split the tasks into three main code sections:
  - particle.py: Implementation of particle functions
  - sim.py: Initializing the simulation

Software structure

- Split the tasks into three main code sections:
  - particle.py: Implementation of particle functions
  - sim.py: Initializing the simulation
  - main.py: Running the simulation

Initialize and Update file

- Align molecules on a grid in space

Initialize and Update file

- Align molecules on a grid in space
- Print the position each time step into the .xyz file

Initialize and Update file

- Align molecules on a grid in space
- Print the position each time step into the .xyz file
- Update that file for each timestep

Initialize and Update file

- Align molecules on a grid in space
- Print the position each time step into the .xyz file
- Update that file for each timestep
- Visualize the file with VMD (open source software)

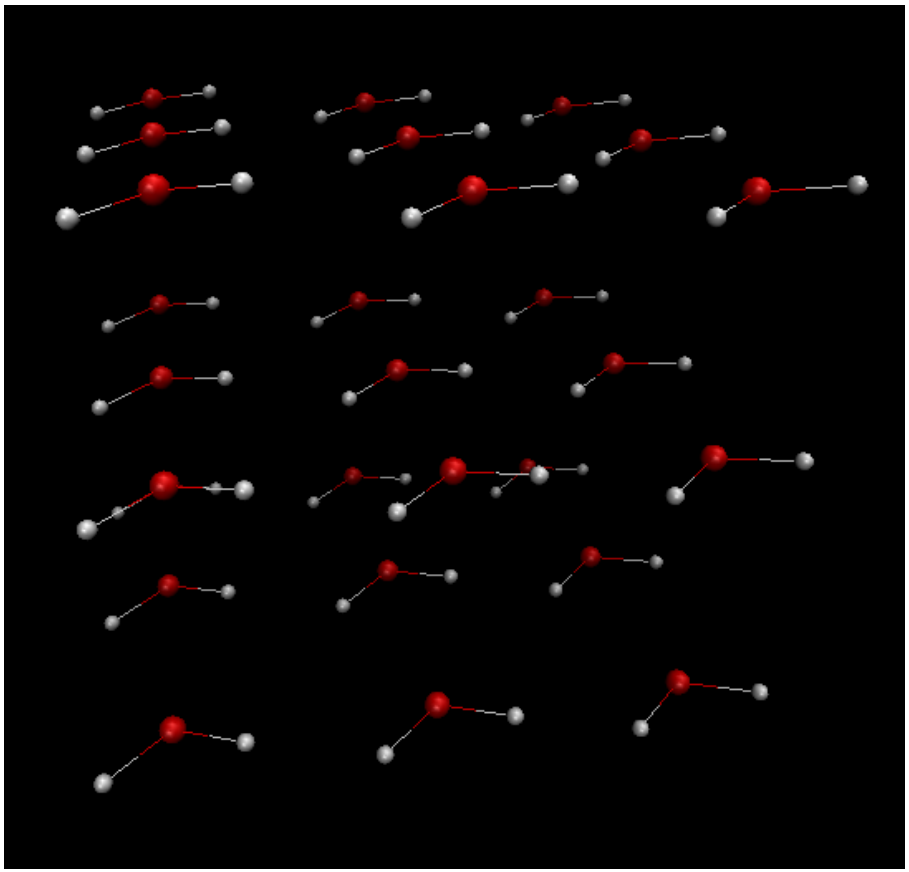

Figure: Initial state: 81 particles, 27 molecules

Listing 1: Update velocities

```
def update_velocities(self):
        for i in range(n_particles):
                for j in range(DIMS):
                        vel[i][j]=0.5*deltaT*acc[i][j] + vel[i][j]
```

Listing 2: Update positions

```
def update_positions(self):
        for i in range(n_particles):
                for j in range(DIMS):
                        r[i][j]=r[i][j]+deltaT*velo[i][j]+((deltaT**2)/2)*acc[i][j]
```

Listing 3: Update acceleration

```
def update_acceleration(self):
        for i in range(n_particles):
                for j in range(DIMS):
                        acc[i][j] = forces[i][j]/mass[i]
```

Internal energy calculations

- Internal energy updated for each timestep

Internal energy calculations

- Internal energy updated for each timestep
- Calculation of bond length (r_{OH}) and angle (Θ)

Internal energy calculations

- Internal energy updated for each timestep
- Calculation of bond length (r_{OH}) and angle (Θ)
- Internal forces are calculated [2]

Internal Energy implementation

Listing 4: Angular potential

```
harmonic_theta = k_theta*(theta - thetaOH)**2
```

Listing 5: Bond length potential

```
harmonic_r = k_bond*((bond1 - bondOH)**2 + (bond2 - bondOH)**2)
```

Derivation of internal forces

$$f_{bond} = \frac{\partial U(r)}{\partial r} = 2k_{valence}(r - r_0) \tag{13}$$

$$f_{angular} = \frac{\partial U(\theta)}{\partial r} = \frac{\partial U(\theta)}{\partial \theta}\frac{\partial U(\theta)}{\partial r} = 2k_{angular}(\theta - \theta_0)\frac{1}{|r_{OH}|} \tag{14}$$

Used Parameters

parameter	unit	value
$k_{valence}$	$\frac{kcal}{mol\,Å}$	1054.2
k_θ	$\frac{kcal}{mol\ radian}$	75.9
r_0	Å	1
θ_0	°	109.47

Table: Implemented parameters [3]

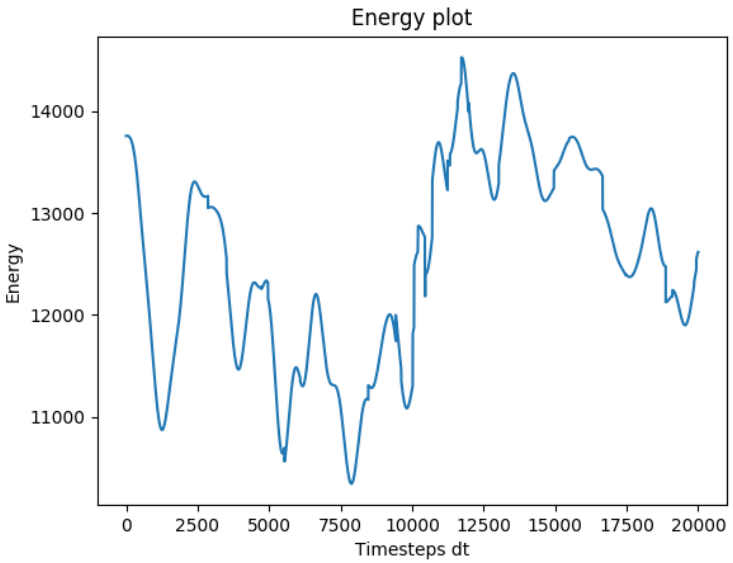

Figure: Conserves angular energy in time

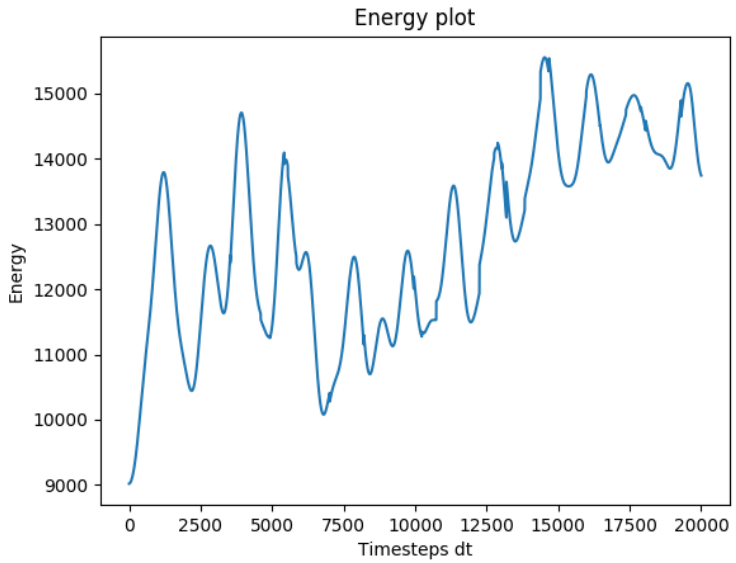

Figure: Conserves valence energy in time

External energy calculations

- Define cutoff radius: $r_{cut} = 2.5\sigma$

External energy calculations

- Define cutoff radius: $r_{cut} = 2.5\sigma$
- Calculation of Lennard-Jones Potential within r_{cut}

External energy calculations

- Define cutoff radius: $r_{cut} = 2.5\sigma$
- Calculation of Lennard-Jones Potential within r_{cut}
    - Derivation and calculation of LJ force [2]

External energy calculations

- Define cutoff radius: $r_{cut} = 2.5\sigma$
- Calculation of Lennard-Jones Potential within r_{cut}
    - Derivation and calculation of LJ force [2]
- Calculation of Coulomb energy

External energy calculations

- Define cutoff radius: $r_{cut} = 2.5\sigma$
- Calculation of Lennard-Jones Potential within r_{cut}
    - Derivation and calculation of LJ force [2]
- Calculation of Coulomb energy
    - Derivation and calculation of Coulomb force

Derivation of external forces

$$f_{LJ} = \frac{\partial U(r)}{\partial r} = 4\epsilon(-12\frac{A}{r_{ij}^{13}} + 6\frac{B}{r_{ij}^{7}}) \tag{15}$$

$$f_{Coulomb} = \frac{\partial U_{coulomb}}{\partial r} = -\frac{1}{4\pi\epsilon_0}\frac{q_1 q_2}{r_{ij}^2} = -C\frac{q_1 q_2}{r_{ij}^2} \tag{16}$$

Used Parameters

parameter	unit	value
ϵ	$\frac{kcal}{mol}$	0.1553537
σ	Å	3.165492
C	$\frac{kcal Å}{mol\ C^2}$	332.0636
r_{cutoff}	Å	2.5σ
q_O	C	-0.82
q_H	C	0.41

Table: Implemented parameters [3]

Energy and temperature plot

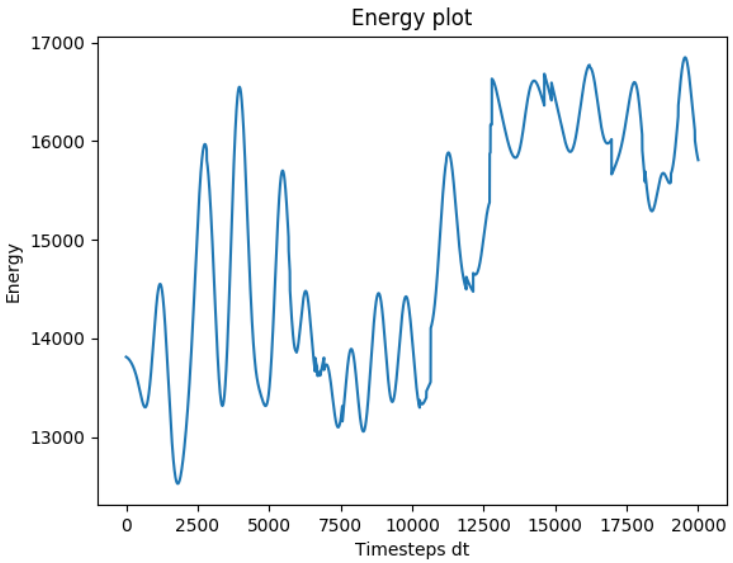

Figure: Plot of total energy

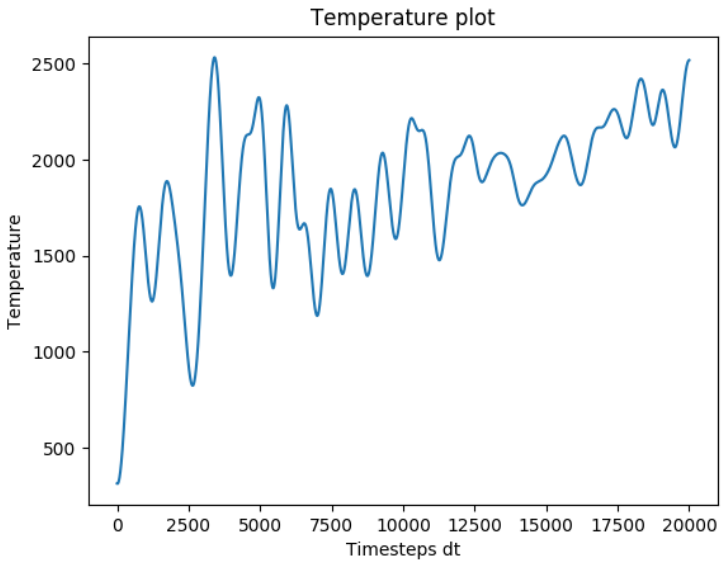

Figure: Temperature plot

Division of tasks

- Divided between internal and external forces
  - Internal forces: Michelle and Can
  - External forces: Puneeth, Dibya and Rushabh

Division of tasks

- Divided between internal and external forces
  - Internal forces: Michelle and Can
  - External forces: Puneeth, Dibya and Rushabh
- Reorganised tasks:
  - Can and Puneeth: bug fixing
  - Dibya and Rushabh: implement Ewald Summation

Difficulties

- Where to start from?

Difficulties

- Where to start from?
  - flexible SPC, Lennard-Jones and initialization

Difficulties

- Where to start from?
    - flexible SPC, Lennard-Jones and initialization
- Michelle withdrew from the program
    - reorder the tasks

Difficulties

- Where to start from?
    - flexible SPC, Lennard-Jones and initialization
- Michelle withdrew from the program
    - reorder the tasks
- Could not implement Ewald Summation

[1] Hiroshi, W. Stability of velocity-Verlet- and Liouville-operator-derived algorithms to integrate non-Hamiltonian systems. Journal of Chemical Physics, 2018

[2] Monasse, B. Determination of Forces from a Potential in Molecular Dynamics, https://www.researchgate.net/publication/259578531 [01/01/2020], 2014

[3] Wu, Y., Tepper, H. Flexible simple point-charge water model with improved liquid-state properties. Journal of Chemical Physics, 2006