MOLECULAR DYNAMICS

Sparsh Agarwal

2022AM11223

1. Initialization

- **Positions**: Initial positions are read from `liquid256.txt`, which specifies atom positions within a cubic cell of side length $L = 6.8$.
- **Velocities**:
  - Random velocities were generated in the range [-1, 1] for all components (x, y, z).
  - Adjusted to ensure zero net momentum:

$$\mathbf{v}_i = \mathbf{v}_i - \frac{1}{N}\sum_{j=1}^{N}\mathbf{v}_j$$

  - Scaled to match the target temperature $T = 300\,\mathrm{K}$:

$$\mathbf{v}_i \leftarrow \mathbf{v}_i \times \sqrt{\frac{3Nk_BT}{\sum_{j=1}^{N} m|\mathbf{v}_j|^2}}$$

2. Force Calculation

- **Lennard-Jones Potential**: Used to model the interaction between particles:

$$V(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^{6} \right]$$

  where:

  - r: Distance between particles.
  - ϵ: Depth of the potential well.
  - σ: Distance where $V(r) = 0$.

- **Force**: Derived as the negative gradient of the potential:

$$F(r) = -\frac{\mathrm{d}V(r)}{\mathrm{d}r} = 48\epsilon \left[\frac{\sigma^{12}}{r^{13}} - \frac{\sigma^{6}}{r^{7}} \right]$$

- **Cutoff Scheme**: Forces and potentials are truncated at $r_{\text{cut}} = 2.5\sigma$. The potential is shifted to ensure continuity:

```
def compute_potential_energy(positions):
    """Compute total potential energy of the system."""
    potential_energy = 0.0
    for i in range(N):
        for j in range(i + 1, N):
            # Periodic boundary conditions
            delta = positions[i] - positions[j]
            delta=apply_pbc(delta,system_size) # Nearest image convention
            r2 = np.dot(delta, delta)
            r_cut=2.5 * sigma
            if r2 < (r_cut)**2:  # Apply cutoff at r < 2.5*sigma
                r = np.sqrt(r2)
                #smooth functiona s discussed in class
                potential_energy += lj_potential(r)
                potential_energy-=lj_potential(r_cut)
                potential_energy-=(r-r_cut)*lj_force(r_cut)

    return potential_energy
```

$$F_{\text{modified}}(r) = F(r) - F(r_{\text{cut}})$$

Why This Formula Is Necessary:

- Without the linear correction term, the force would not transition smoothly to zero, potentially causing the system to behave unphysically.
- This modification ensures both the potential and its derivative are continuous at r_{cut}.

Solution part-3

Temperature Monitoring

The system's instantaneous temperature is derived from the kinetic energy using the formula:

$$T = \frac{2}{3Nk_B}K, \quad K = \frac{1}{2}m\sum_{i=1}^{N}|\vec{v}_i|^2$$

Part-4 and Part-5

3. Time Integration

- **Velocity Verlet Algorithm**: Used to integrate the equations of motion:

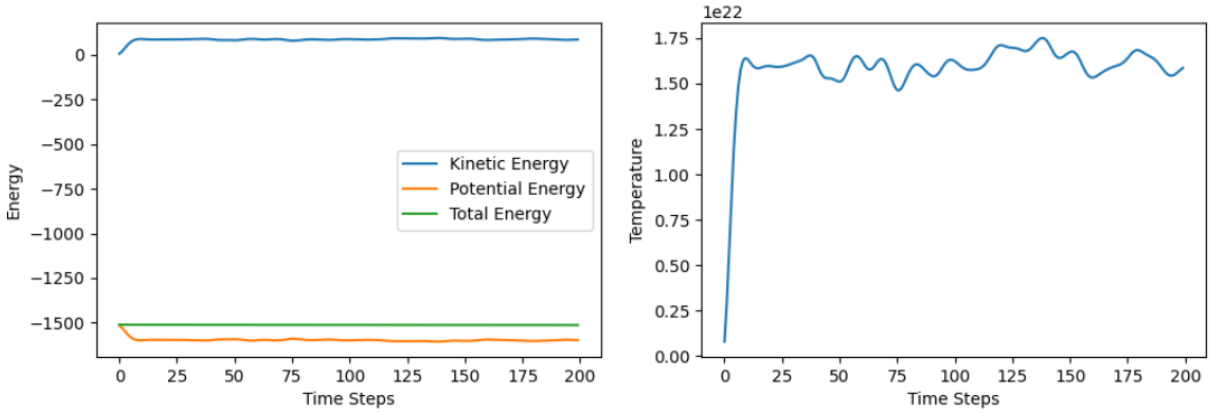
$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \mathbf{v}(t)\Delta t + \frac{\mathbf{F}(t)}{m}\frac{\Delta t^2}{2}$$

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t) + \frac{\mathbf{F}(t)+\mathbf{F}(t+\Delta t)}{2m}\Delta t$$

- **Periodic Boundary Conditions**: Applied to ensure particles leaving one side of the simulation box re-enter from the opposite side.

```python
def apply_pbc(delta, box_size):
    """Apply periodic boundary conditions to delta."""
    delta -= box_size * np.round(delta / box_size)
    return delta
```

✓ 0.0s



1. Energy Conservation and System Stability:

- The left plot shows three energy components over 200 time steps:
    - Kinetic energy (blue line) starts low and quickly rises to stabilize
    - Potential energy (orange line) starts high (negative) and quickly stabilizes
    - Total energy (green line) remains constant after initial equilibration, indicating good energy conservation
- The constant total energy suggests the Velocity Vervlet algorithm is working correctly, as it should be symplectic and conserve energy

2. Temperature Evolution (right plot):

- The temperature shows characteristic behavior of a molecular dynamics simulation:

    - Initial rapid rise from near-zero to approximately 1.5×10^{22} (in reduced units)
    - After equilibration (around step 25), the temperature fluctuates around a mean value
    - These fluctuations are normal and represent thermal fluctuations in the system
    - The magnitude of fluctuations appears reasonable for a stable simulation

3. System Behavior:

- The initial sharp changes in both plots represent the equilibration phase
- After about 25 time steps, the system reaches a quasi-equilibrium state where:
    - Energy components maintain stable average values
    - Temperature oscillates around a mean value
- The simulation appears well-behaved with no signs of numerical instabilities

4. Physical Interpretation:

- The negative potential energy indicates attractive interactions between particles dominate
- The kinetic energy stabilization suggests the velocity distribution has reached a Maxwell-Boltzmann-like distribution

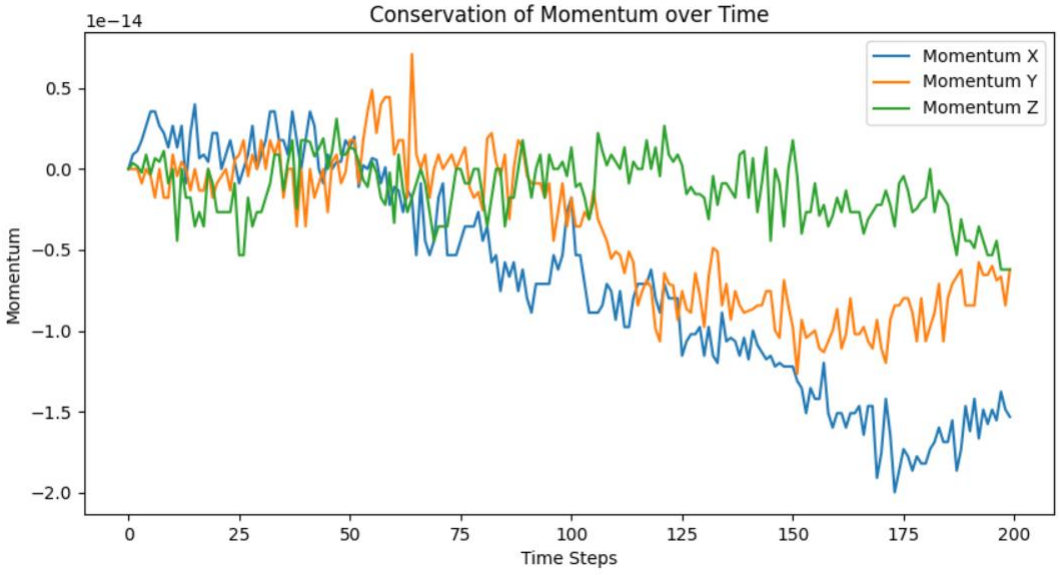
Temperature Equilibration

```python
def check_temperature_equilibration(temperatures, T_target, temp_diff_threshold=1.0, std_threshold=0.5):

    if len(temperatures) < 100:
        return False  # Not enough data points yet

    recent_temperatures = temperatures[-100:]
    mean_temp = np.mean(recent_temperatures)
    temp_std = np.std(recent_temperatures)

    # Check both mean deviation and fluctuation stability
    return (np.abs(mean_temp - T_target) < temp_diff_threshold and temp_std < std_threshold)
```

✓ 0.0s

Discussion

1. **Why This Method Is Effective**:
    - Combining temperature difference and fluctuation stability ensures robust equilibration detection.
    - The velocity-based temperature calculation links directly to the system's kinetic energy, ensuring accuracy.
2. **Interpretation of Results**:
    - The equilibration step indicates when the system reaches a steady-state temperature.
    - Deviations before equilibration may highlight transient effects or initialization artifacts.



Discussion on Momentum Plot

The graph depicts the conservation of momentum over time in the x-, y-, and z-directions. Ideally, the momentum in each direction should remain constant and close to zero in a well-equilibrated molecular dynamics simulation. However, the plot shows minor fluctuations around zero, with occasional deviations that are not exactly zero.

Reasons for Non-Zero Momentum

1. **Numerical Approximations:**
    - Molecular dynamics simulations rely on numerical integration methods (e.g., Verlet or Velocity-Verlet). These methods are not perfectly accurate and may introduce small errors in each time step, accumulating over time.
2. **Finite Precision of Calculations:**
    - Simulations are performed using floating-point arithmetic, which has finite precision. This can lead to rounding errors that manifest as small deviations in momentum.
3. **Initial Conditions:**
    - If the initial velocities are not perfectly assigned to satisfy zero total momentum (or the randomization process introduces bias), the system may exhibit slight residual momentum throughout the simulation.
4. **Thermostat Effects:**
    - If a thermostat is used to control the temperature, it may slightly perturb the system's velocities, indirectly impacting the momentum. This effect is usually minor but can lead to fluctuations.
5. **Boundary Effects:**
    - For systems with periodic boundary conditions, particles exiting one side and entering another can introduce minor inconsistencies in momentum conservation, especially near the boundaries.