FK8028: Project 1

Author: Andreas Evensen

Date: January 23, 2024

1 Theory

2 Method

3 Result

4 Conclusion

5 Appendix

A

```python
import numpy as np
from dataclasses import dataclass
from typing import List

@dataclass
class Atom:
    radius: float
    position: np.ndarray

def Lennard_Jones(pos: List[np.ndarray]):
    """
        Calculate the Lennard-Jones potential of a system of N atoms, given their position.
        the type Atom is a dataclass with two attributes: radius and position.
    """
    potential: float = 0
    force: np.ndarray = np.zeros(3)
    sigma: float = 1
    epsilon: float = 1

    for i in range(len(pos)):
        for j in range(len(pos)):
            if i == j:
                continue
            delta = (sigma / np.linalg.norm(pos[i]- pos[j])) ** 6
            potential += epsilon * (delta ** 2 - delta)
            force += 4 * epsilon * (12 * delta ** 2 - 6 * delta) * (pos[i] - pos[j]) / np.linalg.norm(pos[i] - pos[j])**2
            print(force)

    return potential, force




def Force_x() -> None:
```

```
    """
        Computes the force in the x direction for each atom in the system.
        The force is given by grad(V) = -dV/dx * e_x  -dV/dy * e_y  -dV/dz * e_z,
        however only the x component is computed here.
    """
    Atom1 = np.array([-1,0,0])
    Atom2 = np.array([7,1,0])

    potential, force = Lennard_Jones([Atom1, Atom2])

    print(force)









if __name__ == '__main__':
    Force_x()
```