potenergyCalculation

November 2, 2022

[1]:
```
import numpy as np
```

[2]:
```
address = "PROBLEM4.data"
```

[3]:
```
# This function uses given data to extract a list of particle coordinates
def readLocation(fileAddress, n):
    txt = open(fileAddress, "r")

    coordinates = []

    for x in txt:
        # print(x)
        if x[0] != "#":
            temp = x.split()
            location = np.array([float(temp[0]), float(temp[1]),␣
 ↪float(temp[2])])
            # print(velocity)
            coordinates.append(location)

    return np.vsplit(np.array(coordinates), n)
```

[4]:
```
def pot(r):
    eps = 0.997
    sig = 3.405
    return 4*eps*((sig/r)**12 - (sig/r)**6)
```

[5]:
```
s1, s2, s3, s4, s5 = readLocation(address, 5)
len(s1)
```

[5]:
```
365
```

[6]:
```
def minImgConv(s, i, j, L):
    dx, dy, dz = (s[j][0] - s[i][0]), (s[j][1] - s[i][1]), (s[j][2] - s[i][2])

    if dx > L/2: dx = dx - L
    if dx <= -L/2: dx = dx + L
    if dy > L/2: dy = dy - L
```

```
    if dy <= -L/2: dy = dy + L
    if dz > L/2: dz = dz - L
    if dz <= -L/2: z = dz + L

    return np.sqrt(dx**2 + dy**2 + dz**2)
```

[7]:
```
def totalPE(s, L):
    ans = 0
    for i in range(len(s) - 1):
        for j in range(i + 1, len(s)):
            r = minImgConv(s, i, j, L)
            ans = ans + pot(r)
    return ans
```

[8]:
```
print("The Total Potential Energy of state 1:", totalPE(s1, 26))
print("The Total Potential Energy of state 2:", totalPE(s2, 26))
print("The Total Potential Energy of state 3:", totalPE(s3, 26))
print("The Total Potential Energy of state 4:", totalPE(s4, 26))
print("The Total Potential Energy of state 5:", totalPE(s5, 26))
```

```
The Total Potential Energy of state 1: -2024.5630809347688
The Total Potential Energy of state 2: -2003.2702624102935
The Total Potential Energy of state 3: -2006.6601425125293
The Total Potential Energy of state 4: -2018.1662939385058
The Total Potential Energy of state 5: -2012.370615396599
```