forceCalculation

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 force(r):
    eps = 0.997
    sig = 3.405
    return 4*eps*((12)*(sig**12)/(r**13) - (6)*(sig**6)/(r**7))
```

[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 totalForce(s, L):
    forceMatrix = []
    for i in range(len(s)):
        ans = 0
        for j in range(len(s)):
            if j != i:
                r = minImgConv(s, i, j, L)
                ans = ans + force(r)
        forceMatrix.append(ans)
    return forceMatrix
```

[8]:
```
forceMatrixAbs = np.absolute(np.array(totalForce(s1, 26)))
```

[9]:
```
print(np.size(forceMatrixAbs))
print(forceMatrixAbs)
```

```
365
[ 4.33760362  1.43913765  3.95872046  0.59866702  1.24752281  5.19266492
 16.83823629  1.67406091  6.80498265  2.34239959  1.3983501   6.10429759
 10.49038153  2.11515375  3.71630818  4.80214031  1.80379195  4.58979101
  5.89567347  2.626029    0.39376276  0.63275136  0.33248927  0.81199048
 10.78915009 10.37198907  7.30217265  0.29635904  0.42108104  1.63484282
  0.05693747  6.97074436  2.10409206  0.02683397  2.80047276  4.21907532
  1.68275808 10.2904264   3.6549643  10.73699296  4.99139565  0.15615435
  0.1368776   9.9948214   1.93491325  3.30468605  0.88584535 10.98602178
  1.91377371 11.25543446 16.63160623  1.86970656  1.04132391  2.5205642
  0.08582736  2.49414702 17.03056256  5.43226249  0.92686099  5.16418695
  8.47840117  5.33004389  5.3822888   2.60530684  0.2415622   3.46173649
  5.78701668  4.80942926  7.94297566  8.7999865   7.24142804  5.33858789
 11.98171594  5.18224468  2.93617371  3.11616362  3.30154242  7.12512194
  5.08443891  3.9897689  11.91576178  2.6690888   4.63463129  7.24169486
  5.21515782 22.55258326  6.15278382  2.46880266  7.86467077 14.8050543
  0.07353797  2.24979245  2.8565054   4.54062384  0.14481014  5.20331457
  0.97629309  0.48494137 17.49989178 11.13861872  0.50011427 13.93587966
  0.13467326 10.32312642  3.81696372  6.64075672  5.80174588  2.91422577
  8.53635076 10.47572543  1.40442828  6.16848635 12.21466233  5.98392233
  0.65326354  2.05824077  0.54857865  5.04101689  2.39500294  2.53776565
 12.79988951  2.04387995  3.86499819  3.03961579  3.15950358  0.2729273
  3.35469483  3.51240119  3.03519789  1.66517868  1.72227733  0.75468554
  1.97744445 11.6560445   4.49747934  0.21523299  3.92311112  7.29110566
  3.05206815  0.05448392  7.94663487  4.02250013  4.86758942  7.25392373
  5.91116132  8.78008798  0.02829801  0.11405302 14.46422811  3.68186198
  4.18339721  2.87002603  6.46559468  1.79118993  7.43198282 15.08637486
 12.06101711  4.80266441  3.35013445  0.71972558  6.8631292   5.01300591
  5.42449194  0.70288571  2.54512315  4.67275057  6.67777096  1.51107231
  5.2168559   2.93108675  5.26613831  4.13600326  1.9524316   2.68646592
  5.78761068 11.3757511   0.55473925  4.42650179  9.75801735  1.55544714
  0.48984984  0.28825918  2.51813317  1.29331341  5.89670985  7.57805544
  1.33916269  9.10741431  9.00005197  1.72062007  0.36423057  6.90705961
  4.12519094  9.73487841  3.92624984  3.71944951  5.32039678 24.22066978
  6.46088792  9.20558808 16.46217007  6.59881097  1.29134783  5.039314
  5.31856638  2.97436731  5.72678326  7.89466463  4.39331567 18.61364686
  3.27632717 13.32019888  7.32844974  2.41572886  0.11107422  4.49436586
  3.85068482 16.60205675  2.77610783  2.96320941  0.04873411  3.43775061
  0.29831775  4.96095237  2.34253702  6.92844999  2.06994124  9.9297867
  5.86870933  4.56835244 12.44660712  1.47952975  2.94010065  2.15752989
 16.30759209  6.11587808  3.83888428  7.85478879  3.17907326 11.3933588
  2.563278    2.5227855   4.24746228  8.14108581  0.45370677  1.55414301
  5.05498634  1.15456377  1.15480622  1.68222243 17.48202245  1.80031083
  3.05695478  8.76375407  8.11325916  6.71727434  7.60842035  6.20260619
 13.83042966  7.47033519  5.63712944  4.79482061  4.54174431  0.38458658
  3.12932304 15.29010309  1.56047948  5.31301194  4.49823692  5.05687198
  8.17606893  1.0163701  11.52343111  0.93379091  4.0196988   4.0157677
  4.96886298  8.21925083  5.34169037  2.75551766 13.62321025  2.28220487
  0.87902187  7.97217046  4.83359609  5.07163382  9.02437012  2.66029054
  9.77223831  4.42740465  1.1822074   2.23196607  8.16557549  2.23129106
  2.37314679  3.52434305  3.8350423   8.08358966  4.71887086  1.28217741
  3.08655025  4.8056361   3.00183347  2.2455017   1.3293995   5.60240266
  2.50687122  8.37832049  7.86881993  2.45423495  3.17173531  6.42737894
  2.81583355  1.26856335  1.34972572  2.31364297  4.72435932 17.18752266
  1.92634447  4.36549769  0.27322205  8.69356766  4.74024247  4.27783638
  9.74726447  6.86322545  2.29207131 13.79904013  1.90672646  4.45287436
  2.36983061  5.54069196  8.08678901  2.40720273 15.77795517  0.1072967
  8.05195702  0.37469272  2.28303754 15.81396099  1.14614704 20.50066751
  4.85995854 11.83889602  4.89448651  2.76272768  6.50205668  0.08484528
  0.10827983  6.19248641  0.29880015  6.08920024  1.64834519  5.08635875
  5.33150406  2.04226506  7.46960486  3.57214224  5.50000325  2.19835815
 14.90219105  2.5886703   3.5068546   8.88395254  3.4604191 ]
```