PBCapplication

October 26, 2022

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

[2]:
```
address = "/Users/nishxntyxdxv/Desktop/sem7/Computational-Chemistry-Lab/Lab4/
 ↪PROBLEM3.data"
```

[3]:
```
# This function uses given data to extract a list of particle coordinates
def readLocation(fileAddress):
    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.array(coordinates)
```

[4]:
```
# Update function takes current state to next state
# Obeying Periodic Boundary Condition

def update(arr, l):
    # 'nudge' is the matrix that contains the changes
    # to be done in coordinates of particles to take
    # them to a new state. Norm is taken to ensure
    # 1 unit displacement in any random direction
    nudge = np.linalg.norm(np.random.rand(216, 3))
    arr = arr + nudge

    for i in arr:
        # Checking the Boundary condition
        if i[0] >= l/2: i[0] = i[0] - l
        if i[0] < -l/2: i[0] = i[0] + l
```

```
        if i[1] >= l/2: i[1] = i[1] - l
        if i[1] < -l/2: i[1] = i[1] + l
        if i[2] >= l/2: i[2] = i[2] - l
        if i[2] < -l/2: i[2] = i[2] + l

#     arr = np.array(arr)
    return arr
```

[5]:
```
# list of coordinates of all the particles in system
locs = readLocation(address)
```

[6]:
```
# length of the box
l = 18.64

# number of iterations
its = 10
```

[7]:
```
states = []
states.append(locs)

for i in range(1, its + 1):
    states.append(update(states[i - 1], l))
    #print("max coordinate of state {} is: {}".format(i, np.amax(states[i])))
```

[10]:
```
print(states[10])
```

```
[[ 4.78833672  7.44583772 -5.82356228]
 [-2.93956328  1.33772772  8.39146772]
 [-9.20416128  1.85883772  5.58293772]
 [-5.50536328 -7.08096328  4.49203672]
 [ 8.19513872 -5.48036328 -5.63346228]
 [-2.09383528 -3.51771228  3.98403772]
 [-6.73036328  2.68393772 -8.10356128]
 [-5.57596228  0.46889772  8.94309772]
 [ 2.69106772 -1.86456328 -0.33268228]
 [ 8.39413672 -7.21276028 -7.68806128]
 [ 8.56263772 -2.03296028 -5.67236128]
 [ 8.72133772 -6.06040228 -1.18636128]
 [-0.08346128 -0.49968228  5.78553672]
 [ 5.83593772 -8.65770228 -0.04896328]
 [ 7.89023872  1.58253472  7.91175772]
 [ 7.14417772 -7.11642228 -3.65960228]
 [ 1.46273772 -8.58326128 -5.59060228]
 [-7.66687228  2.79663872  2.99963772]
 [ 4.59753772 -5.58066128  6.53423672]
 [ 4.44613672 -7.80989228 -2.63175628]
 [ 3.45043772  0.28443772 -5.46616128]
```

```
[-3.51518228 -0.86986128 -1.26116128]
[ 6.35983772  5.35870772 -9.11398228]
[ 5.18773672 -3.26056128 -0.33136328]
[-6.98896228 -3.91691228  7.21074872]
[ 5.60992772  1.54383772 -4.19996228]
[ 1.76510772 -5.74866328  2.93263672]
[ 7.99133772 -2.14435728  9.06523872]
[-8.71284228  3.27343772 -0.61816028]
[-6.82966228  7.23559772  0.97313672]
[-4.17596228 -2.12016328  1.00453672]
[-0.93337228  4.94543672 -5.13886228]
[-5.21376228  4.61562772  5.56634772]
[ 2.19873772  8.94493772  2.30882772]
[-4.00966228  8.46723872  1.32363872]
[-9.31487228  2.73053772 -3.31209228]
[-3.19975228 -7.71996228 -1.74856128]
[ 0.03403672 -8.06186228  3.33518772]
[-2.78445628  6.31963772  8.68829772]
[ 6.61683772 -9.24406128 -5.37866228]
[ 4.37873672 -6.08216128 -5.06166228]
[-7.23275228  1.14553872 -4.02186328]
[ 4.84614772 -1.72906528 -2.35536128]
[ 1.26512772  4.10113972 -3.76789228]
[-2.18156028 -8.27288228  1.54523672]
[ 3.43933672  3.63493772 -0.34386228]
[-5.20046228  1.12750772 -1.70657228]
[-0.40876328  2.70906772  8.66203872]
[-4.78746228 -6.59947228 -7.71486128]
[ 2.99073772  7.48091772 -8.03686128]
[ 3.48103772  7.40837872  6.08412772]
[-1.30750728 -2.60076328  9.28403972]
[ 7.27440772 -8.67286228  5.83053772]
[ 3.63673572  0.33386772  4.39673672]
[ 7.40391772  2.56183972  3.56493672]
[ 4.12803672  4.88003772  6.48493672]
[-3.75936528 -5.91696128  1.20193672]
[ 7.24197772  7.60067872  7.25473872]
[-8.97416228 -4.79926328  4.40853872]
[ 3.28550772  5.17954772  9.21663972]
[-1.83786228  3.63093672 -7.36346128]
[-5.99866128 -2.39036128 -1.99676128]
[ 1.89533772 -5.01246228 -4.63754228]
[ 1.86543672 -5.25784228  6.62953772]
[ 8.77283872 -4.50076328  1.29533672]
[-0.71776228 -9.10506028  8.70943772]
[ 7.10200672  8.04223772 -8.58666228]
[-5.45736328 -8.24276128 -0.24176228]
[-8.77266328 -5.16306228 -3.55916228]
```

```
[ 7.59240772  5.33103772 -0.29276228]
[ 7.07483672  4.88921772  6.85631772]
[ 4.70811772  2.49063872  8.12811772]
[ 4.07535772  6.60320772 -1.21633328]
[-0.20166428  6.98483672  8.75050772]
[ 7.10184772 -2.35436228  1.26553772]
[ 7.73013772 -6.25701228  3.17803972]
[ 7.74523572  0.48823572 -6.95315928]
[-8.80736528  0.08146772 -8.84594228]
[ 5.09873972  4.75813672 -6.61986128]
[ 8.87673672 -8.29679228  1.71293772]
[-4.02205628 -1.35813828 -7.67486128]
[ 3.40573872  6.60483772  1.64473772]
[ 5.81473772 -8.21836428 -7.67741228]
[ 2.81893772  3.18063772 -5.75836128]
[-1.37379528 -7.11149228 -8.08076428]
[-5.31426228 -2.32976528  8.74133872]
[-2.11046228  0.77658772 -7.75112228]
[ 4.34603772  8.67963872  8.69643872]
[-1.47886228 -2.51446128 -4.21826328]
[-8.09743228 -2.59811628 -4.08496028]
[ 9.16703672  1.31713672  1.65163772]
[-3.40640228  6.21363772  2.78863972]
[ 2.48643872  2.01853572  6.36304772]
[ 1.33353772  6.34653872 -1.71546128]
[ 6.05603772 -3.51795928 -6.33276128]
[ 0.78213472  6.71235772  6.28190772]
[ 3.35571772 -2.71256228  7.07114772]
[ 0.89403572  1.95793772  0.88263672]
[ 0.54013772  5.74753672 -7.57986428]
[-4.46219228 -5.66176228  7.34633672]
[ 0.15872772 -6.97186228 -4.08036028]
[ 7.55943972 -3.46078228  6.17463672]
[ 7.17845772  3.47704772 -5.11316328]
[-8.61436028 -3.23376228 -7.68212228]
[ 5.54953672  7.14513572  3.12892772]
[ 9.13395772 -1.40613428  2.59433772]
[-5.35846328 -2.59536128 -5.57521228]
[-2.88256228  8.83663872 -3.97624228]
[ 1.38937772 -5.66616228 -0.03716228]
[-6.16364228 -4.21845228 -7.34409228]
[-6.74546128  5.71263872 -2.89416528]
[-3.73326128  5.35783672 -1.80804028]
[ 0.14452772  7.16245772  0.43988772]
[ 1.57913672 -7.92915228  7.46378872]
[ 6.09803672  5.16103572 -2.75675928]
[ 0.54563572 -3.27623228  0.84527772]
[-5.38196328 -7.91854228  8.68673872]
```

```
[ 3.04073772  2.38618772 -2.66992428]
[-1.15502728 -5.97231228 -1.88706428]
[-2.03565928 -3.80226528  1.17791772]
[ 1.71763672  0.45583772 -1.25727928]
[ 4.35123672 -5.91256128 -8.01866128]
[-8.83669228  8.50633772  7.57493872]
[ 4.18298772  3.80077772  3.44663672]
[ 6.62471772 -0.73338228  6.35893772]
[-9.29056128 -1.24966228  5.42933872]
[-5.77186228  5.60003672 -5.76169228]
[-8.12686128  4.74166772  9.19842772]
[-7.50376228  7.33152772  5.03403672]
[-6.68312228  0.02303672  6.00565772]
[-4.71056128 -8.07121228 -5.38706228]
[-3.12016128  1.46953672  3.56883872]
[ 2.37798772 -7.99646328 -7.99156128]
[-5.55130228  1.95533672  4.52053672]
[-2.20656228  5.14933772  0.55543572]
[-0.24646128 -3.59709228 -1.98829928]
[ 6.19983872 -4.94396228  8.73146772]
[ 1.44873872  4.41103572  2.19413772]
[-3.81407228  1.43273772 -4.19496128]
[-3.57286428  3.83153772  7.61513772]
[-9.29071228  4.98445772  3.59361772]
[-3.40526228 -4.21066228 -9.10186228]
[-6.72175928 -7.11346028 -2.36616128]
[ 7.01528772  8.07053872 -1.78206228]
[ 5.29333672 -2.35446328 -9.30106328]
[ 8.61313772  8.59952772  3.79797772]
[-6.57299228 -0.78953228  0.70252772]
[-8.36846128  8.00521772 -3.00056228]
[-4.78796228  2.95700772 -6.31076228]
[-6.33432228 -4.66516328  3.96514772]
[ 2.11612772 -2.59944428 -3.32386428]
[ 6.48663772  0.39123772 -1.45746228]
[-1.61826128 -8.24586128  5.36327772]
[ 5.74923672  1.87953672  5.52753672]
[-1.45776328  0.25573772 -4.15786128]
[-8.82576128  2.69376772 -6.37916128]
[-9.16227228  5.60693572 -6.65956328]
[-8.58756228  7.82624772 -8.38390228]
[ 0.63174772  0.90413672 -7.30146228]
[-3.36566128  6.09583272 -4.24716128]
[-6.50946128  6.24843672  7.51513772]
[ 1.90853872  0.57436772  8.89835772]
[ 5.64833772  2.29193672  0.25143872]
[ 1.01133772 -0.63363228 -4.90655228]
[ 7.36750872  2.70469772 -8.44636128]
```

```
[ 0.90943972  4.03820772  5.17673672]
[ 5.65391772 -4.15336228 -3.58766128]
[ 3.07373872  8.83273772 -3.90826428]
[-0.90896228  9.21303872 -0.76106228]
[-5.24418228  2.62693672  0.88343772]
[-5.49656328  9.19408772  6.08953672]
[-0.90645928  0.04643672 -1.50251628]
[-5.89330228  8.12514772 -8.99006128]
[-5.54326228 -2.39266228  5.37413672]
[-2.54096228 -4.34776328 -6.17246228]
[ 5.42793772 -0.96755228 -6.86872228]
[ 6.88183772  4.91805772  2.20046772]
[ 2.87783772 -7.93026128 -0.31876128]
[-8.28516228 -6.70336128  8.99153972]
[-6.40341228  4.82863772 -0.37086128]
[ 0.45339772 -1.22806328  3.08553872]
[ 0.01943772 -5.10406228 -6.54816028]
[ 8.15873772  7.18483772 -4.58856228]
[ 1.27273672 -8.81916228 -2.29018728]
[ 8.93873872 -6.61186228  6.75418672]
[-7.67236128 -6.11616528 -6.06220228]
[ 1.43973772  3.47093672 -8.17136028]
[ 4.04403672 -9.28916128  4.35193772]
[ 0.50128772 -3.79926328  4.91473672]
[-2.70706328 -1.27786228  7.32585772]
[ 1.55470772 -2.10586328  8.89076772]
[ 4.87075772 -1.06476228  1.96239772]
[ 3.32565772  1.57623672  2.08953672]
[-7.13261228 -8.69846128 -4.43306328]
[-1.65286228  5.94443672  4.99383772]
[-6.87316328 -0.38389228 -6.78362228]
[-3.64976528 -1.07786128  3.78123872]
[ 4.23063872 -2.35466328  4.48089772]
[-8.44516128  4.61658772  6.11229772]
[-7.12176228 -3.61916328  1.54962772]
[-4.44532228 -5.24056228 -4.27596028]
[-3.43136228  7.54573872  6.32324772]
[ 5.29574772 -7.03316228  4.39743772]
[ 4.42520772 -5.32276528  2.38413772]
[ 9.27500772  8.53073872 -0.54256228]
[-3.11066228 -6.16120228  5.05393572]
[-6.12807228  5.63603572  3.33663772]
[-2.67489828  2.36183772  1.18903772]
[ 5.62933772 -5.82955928 -0.02796128]
[-6.65978228 -7.15915228  1.83673672]
[-1.39626628  9.01373972 -6.75221228]
[ 2.11303872 -4.39682228 -8.19126328]
[-3.58303228  7.29646772 -7.36652228]
```

```
[-1.23756028  0.13873872  1.32140772]
[-0.92936328  1.94984072  5.03696772]
[-4.68915928 -4.76856228 -1.63956228]]
```

[9]:
```
# Saving the 2D array in a text file

file = open("out.txt", "w+")
file.write("# The coordinates of particles after 10 iterations: \n")
for x in states[10]:
    file.write(str(x[0])[:8] + "       " + str(x[1])[:8] + "       " +␣
 ↪str(x[2])[:8] + "\n")
```