Class exercise

Simulation and Modeling (CSCI 3010U)

Faisal Qureshi

Many particle systems

Part 1: Setting up initial positions

Complete the `random_pos()` method in the code provided below. This methods sets the initial positions of the particles at random locations such that the minimum distance between any two particles is `min_sep`.

```
import numpy as np
from numpy import random
from scipy.integrate import ode
from matplotlib import pyplot as plt
from matplotlib import animation


def rect_pos(nx, ny, Lx, Ly):
    '''
    Create 4 x n_particles state vector
    '''
    n_particles = nx * ny
    state = np.zeros([4, n_particles])
    dx = Lx / float(nx)
    dy = Ly / float(ny)
    for ix in range(nx):
        for iy in range(ny):
            i = ix + iy * nx
            state[0,i] = dx * (ix + 0.5)
            state[1,i] = dy * (iy + 0.5)
    return state

# TO DO
def random_pos(nx, ny, Lx, Ly, min_sep):
    n_particles = nx * ny
    state = np.zeros([4, n_particles])
    return state

nx = 10
ny = 10
Lx = 20
Ly = 15

state = rect_pos(nx, ny, Lx, Ly)
```

```
print state

plt.figure(1)
plt.axes(xlim=(0,Lx), ylim=(0,Ly))
plt.title('Initial positions')
plt.xlabel('x')
plt.ylabel('y')
plt.plot(state[0,:], state[1,:], '.')
plt.show()
```

Part 2: Periodic boundary conditions

Compute the distance between three particles at locations $(2, 2)$, $(8, 8)$ and $(8, 6)$ in a box whose width (x) and height (y) are 10. Assume periodic boundary conditions in both x and y.

Due back before the end of the class.