HomeworkD-plots+nestedloops

June 12, 2022

Due on Saturday, June 25 at 5pm.

Please read the instructions *carefully*, and complete all the requested steps in a jupyter notebook. Include all the code you write as "Code" cells and any written responses as "Markdown/Raw NBConvert" cells. Name your notebook (up at the top) `homework#_{identikey}` (replacing `{identikey}` with your identikey and `#` with the appropriate letter for this week's homework).

This homework builds on HomeworkC. *(Have you looked at the HomeworkC solutions?)* It is a little more verbose, but the extra words are meant to help give guidance on the structure of code.

Be sure to use comments throughout your code to explain how it works, variables, etc. You don't need to include full docstrings for every function you write, but it *is* good practice!

1 N-body Dynamics (for N>2)

In HomeworkC, you wrote functions to calculate the gravitational force vector of one body acting on another. Your function accepted two masses and two position vectors, and returned the force vector $\vec{F}_{ij}$ representing the force on body i due to body j. In this homework, you will use that function to build a more complicated code.

Imagine a system of N bodies. The net force on each body is a combination of the gravitational influences of all other bodies; that is, the net force on body i due to all other bodies j is

$$\vec{F}_i = \sum_{j \neq i}^{N} \vec{F}_{ij}$$

The sum is over all values of $j \neq i$, since a body does not exert a force on itself. This net force $\vec{F}_i$ is also a vector quantity, and it is the vector sum of all of the forces.

2 Overview

In this assignment…

- You will calculate the force vectors for an arbitrary number of bodies. Instead of just one force vector for one body, you will construct a list that contains force vectors for all the bodies you are considering ($\vec{F}_i$ for all values of i).

  - Do not hard code your procedure based on the number of particles given in the example. Your program should be able to determine the number of particles based on the input list.

- You will calculate the net force vector $\vec{F}_i$ for each of those bodies. This force vector is the vector sum of the vector forces caused by all the other individual bodies, as shown above.
- You will make some plots to help visualize the particles and positions involved.

3 Procedure

1. Start a jupyter notebook. In the first cell, run the command "`%matplotlib inline`" to ensure that all your plots will appear inside your notebook. Also, use this cell to import tools we will need for using `numpy` arrays and for making plots with `matplotlib`.
2. In the second cell, define the masses and positions of our objects by copying and pasting the following block of code.

```
au = 1.496e+11    # in meters

masses = np.array([1.0e24, 25.0e24, 50.0e24, 30.0e24, 2.0e24])   # in kg

positions = [np.array([ 0.5,  2.6,  0.05])*au,
             np.array([ 0.8,  9.1,  0.10])*au,
             np.array([-4.1, -2.4,  0.80])*au,
             np.array([10.7,  3.7,  0.00])*au,
             np.array([-2.0, -1.9, -0.40])*au]
```

3. In a new code cell, write code to print a nicely formatted table that includes the index of each particle, its mass, and its x, y, z positions. Use a `for` loop to step through your `masses` and `positions` lists to make this table. It should look something like the following:

```
  particle |       mass |          x |          y |          z
       (#) |       (kg) |        (m) |        (m) |        (m)
-----------------------------------------------------------------
         0 |    1.0e+24 |    7.5e+10 |    3.9e+11 |    7.5e+09
         1 |    2.5e+25 |    1.2e+11 |    1.4e+12 |    1.5e+10
         2 |    5.0e+25 |   -6.1e+11 |   -3.6e+11 |    1.2e+11
         3 |    3.0e+25 |    1.6e+12 |    5.5e+11 |    0.0e+00
         4 |    2.0e+24 |   -3.0e+11 |   -2.8e+11 |   -6.0e+10
```

4. Copy and paste the functions you wrote in HomeworkC into one (or multiple) cell(s) in this notebook. For this homework, you will want to have access to a function that accepts four inputs (two masses and two position vectors), and returns one force vector as an array. If you want, you can use the functions defined in the solutions to HomeworkC (on Canvas). There, the function that calculates a force vector is called `forceVector` (it depends on the other functions, so they'll all need to be copied into this notebook). Make sure these functions work in your new notebook.

5. In a new code cell, make a plot of the positions of the particles in *units of AU*. Because we are (for now) restricted to two dimensions, plot only the x- and y-components of the the positions (even though you also have a z-component). Your plot should:
   - have labels (including accurate units) on the x and y axes
   - have appropriate x and y limits, to show all your particles
   - set the area of the symbol for each particle to be proportional to its mass (You can set the area of a point in `plt.scatter` by including `s=area` in the function call. Good values of area to put in are between 1 and 100; you will need to scale the mass values to get area values in this range.)
   - Note that you will need to extract the x and y values from your `positions` list of numpy arrays. You can either plot point-by-point using a `for` loop or you can extract the x and y values before plotting using a `for` loop.
6. In a new code cell, write a function that accepts a list of masses and a list of positions (just like those defined above). Using nested loops, the function should calculate the *net* force vectors on *each* of the particles. The function's output should be a list of force vectors. In this output list, there should be as many force vectors as there are particles, and each force vector should be a 3-element numpy array. Remember, to calculate the net force on each particle, you will need to add together the force vectors it feels from each other particle in the list (***but not itself!***)
7. In a new code cell, test your function by running it on `masses` and `positions` variables defined above. Write code to print the resulting list of force vectors in a nicely formatted table. It should look like this:

```
  particle |         Fx |         Fy |         Fz
       (#) |        (N) |        (N) |        (N)
-------------------------------------------------
         0 |   -1.3e+15 |   -6.8e+14 |    3.4e+14
         1 |    5.8e+15 |   -3.3e+16 |    1.1e+15
         2 |    6.9e+16 |    4.1e+16 |   -2.6e+16
         3 |   -3.3e+16 |    1.3e+15 |    1.0e+15
         4 |   -4.0e+16 |   -8.2e+15 |    2.4e+16
```

8. For an additional test of your code's ability to calculate net force vectors, make a plot that contains both the particles as you plotted them before *and* arrows specifying the directions of the force vector acting on each particle. You will likely want to reuse code that you wrote to make the first plot, and you have two options for how to make this new plot. You could:
   - modify your net force function above to generate the arrow plot while doing the force calculations, or…
   - write a block of code to generate this arrow plot from three input lists (one of masses, one of positions, one of force vectors), and run it after calculating your list of vectors.
     - See the `Quiver_Example.ipynb` file on Canvas for an example of using quivers.

4 Turn in your Assignment

Your final version of your assignment should run from top to bottom without errors. (You should comment out or delete code blocks that didn't work.) To create a clean version, rerun your notebook

using "Kernel|Restart & Run All." Be sure to save this final version (with output). To submit, click "File|Download As >|HTML (html)" inside jupyter notebook to convert your notebook into an HTML web page file. It should have a name like `homework#_{identikey}.html` (where # is the appropriate letter for this week's homework). Please upload this HTML file as your homework submission on Canvas.