6 — PHY 494: Homework assignment (20 points total)

Due Saturday, Mar 4, 2016, 5pm.

Submission is to your **private GitHub repository**.

Enter the repository and run the script `scripts/update.sh` (replace *YourGitHubUsername* with your GitHub username):

```
cd assignments-2016-YourGitHubUsername
bash ./scripts/update.sh
```

It should create three subdirectories[1] `assignment_06/Submission`, `assignment_06/Grade`, and `assignment_06/Work` and also pull in the PDF of the assignment and an additional file.

To submit your assignment, commit and push a PDF, text file or Jupyter notebook inside the `assignment_06/Submission` directory and **name it `hw06.ipynb` (or `hw06.pdf`)**. *Commit any other additional files exactly as required in the problems.*

Failure to adhere to the following requirements may lead to homework being returned ungraded with 0 points for the problem.

- Homeworks must be legible and intelligible (write complete English sentences when you are asked to explain or describe).
- If you are required to submit code, it must run without errors under Python 3 with the anaconda distribution.
- If you are required to submit data then the data files must be formatted exactly as required to be machine parseable.

Collaboration: Up to three students may form a team and solve the homework together. Each student may submit the same solution to their own repo. **Add a text file COLLABORATION.txt** to the repository in which you

- list the names of all team members (last name, first name), and
- provide a brief statement ($<$ half a page) as to who in the team contributed what to the solution

If you solved the problem on your own, just put your own name as the single name in COLLABORATION.txt and state in the filw "I solved and completed the homework by myself.".

[1] If the script fails, file an issue in the Issue Tracker for PHY494-assignments-skeleton and just create the directories manually.

planet	mass (M_{sol})	distance (AU)	orbital period (yr)	angular position (1690) θ
Sun	1	—	—	—
Uranus	4.366244×10^{-5}	19.1914	84.0110	205.640°
Neptune	5.151389×10^{-5}	30.0611	164.7901	288.380°

Table 1: Parameters for Uranus and Neptune

6.1 Discovery of Neptune (3-body problem) (20 points)

Historically, the planet Neptune was discovered because of its (small) observable influence on the orbit of Uranus. Neptune attracts Uranus and this leads to small changes in Uranus' orbital velocity. In this problem you are studying a simplified situation only containing the Sun, Uranus and Neptune. (For more background, see *Computational Physics* 9.7.)

Assume that to a first approximation that the orbits of Uranus and Neptune are circular (so that you can easily calculate the initial angular velocity)[2] and co-planar. The initial position is given by an angle θ relative to the x-axis and the distance r from the sun, as listed in Table 1.

Calculate the variation in the angular velocity of Uranus $\Delta\omega_U(t)$ due to the influence of Neptune: Compare the instantaneous angular velocity[3] $\omega_U(t)$ to the one in the absence of Neptune, ω_U^0:

$$\Delta\omega_U(t) = \omega_U(t) - \omega_U^0 \tag{1}$$

by plotting $\Delta\omega_U(t)$ over t and briefly comment on your result.

- Use astronomical units: AU for length, year for time, mass in terms of solar masses (i.e. sun's mass $M = 1$), gravitational constant in AU: $G = 4\pi^2$
- Use the *Velocity Verlet* algorithm with a time step of 0.1 years to integrate the equations of motions of Uranus and Neptune.
- The sun can be considered stationary ($M \gg m_U \approx m_N$).
- Analyze 160 years (almost one complete orbit of Neptune).

[2] In your simulation, do not enforce circular orbits, though. With the given parameters, orbits should turn out to be nearly circular.

[3] The angular velocity ω for circular motion is

$$v = \frac{2\pi r}{T} = \omega r$$
$$\omega = \frac{2\pi}{T} = \frac{v}{r}$$

where T is the period and v the speed (velocity is normal to the radial vector $\mathbf{r}$, r is the distance from the center).

- Skeleton code is provided in `outerplanets.py`; you can use it but you don't have to. However, you must write your code so that
    1. it can be imported as `import outerplanets`
    2. it contains a function `integrate_orbits(dt=0.1, t_max=320, coupled=True)` to do the integration; the function has to return a tuple `time, r, v` (see `outerplanets.py` for details).

  A test case `test_outerplanets.py` is provided; you can run

  ```
  nosetests -v test_outerplanets
  ```

  and all tests should pass if you have done everything correctly.

- Also plot the orbits of Uranus and Neptune.

- The file `outerplanets.py` must be a separate file. Provide the two figures as individual PDFs[4] (use `ax.figure.savefig("filename.pdf")`). Other descriptions and code can be submitted a notebook or in other suitable forms.

To calculate $\Delta\omega$ in Eq. 1 you need to run your integration

1. *without* the interaction between Neptune and Urane (`coupled=False`) and
2. *with* the gravitational interaction.

6.1.1 Angular velocity

You can calculate the instantaneous angular velocity as[5]

$$\omega(t) = \frac{v(t)}{r(t)}, \quad r = |\mathbf{r}|, v = |\mathbf{v}|$$

where position $\mathbf{r}(t)$ and velocity $\mathbf{v}(t)$ are determined from the integrator.

6.1.2 Interactions

This is a *three-body* problem. You must compute the forces for Uranus and for Neptune, using the three interactions

1. Uranus — Sun
2. Neptune — Sun
3. Uranus – Neptune

[4] If PDF does not work for whatever reason, use the JPG format instead of PDF.
[5] The code contains the function `omega()` to perform the calculation.

Newton's force law for gravitation is

$$\mathbf{F} = -\frac{GmM}{r^2}\hat{\mathbf{r}} \tag{2}$$

$$\hat{\mathbf{r}} = \frac{1}{\sqrt{x^2+y^2}}\begin{pmatrix} x \\ y \end{pmatrix} \tag{3}$$

Remember that Newton's second law states

$$\mathbf{F}_{12} = -\mathbf{F}_{21}. \tag{4}$$