Gravitational N-body modelling: validation test

The validation test for this project involves coding a two-body problem in two dimensions using *odeint*. To complete this test, use the following step-by-step guide to reproduce the plots displayed on the next page. *These plots must form the first figure in your project report*.

1) Write down the eight coupled ODEs you will need to solve on a piece of paper. *Only once you have done this*, start to write a two-body code that simulates two bodies with equal mass orbiting around each other. Use the following initial conditions.

$$x_1 = -0.5\ \mathrm{AU},\ y_1 = 0,\ v_{x1} = 0,\ v_{y1} = -15\ \mathrm{km\ s^{-1}}$$
$$x_2 = 0.5\ \mathrm{AU},\ y_2 = 0,\ v_{x2} = 0,\ v_{y2} = 15\ \mathrm{km\ s^{-1}}$$

where $1\ \mathrm{AU} = 1.496 \times 10^{11}\ \mathrm{m}$. Assume that $\mathrm{M}_1 = \mathrm{M}_2 = M_\odot$, where the mass of the Sun is $M_\odot = 1.989 \times 10^{30}\ \mathrm{kg}$.

2) Solve the set of eight coupled ODEs using the scipy *odeint* package. This is imported into your python script using the command

from scipy.integrate import odeint

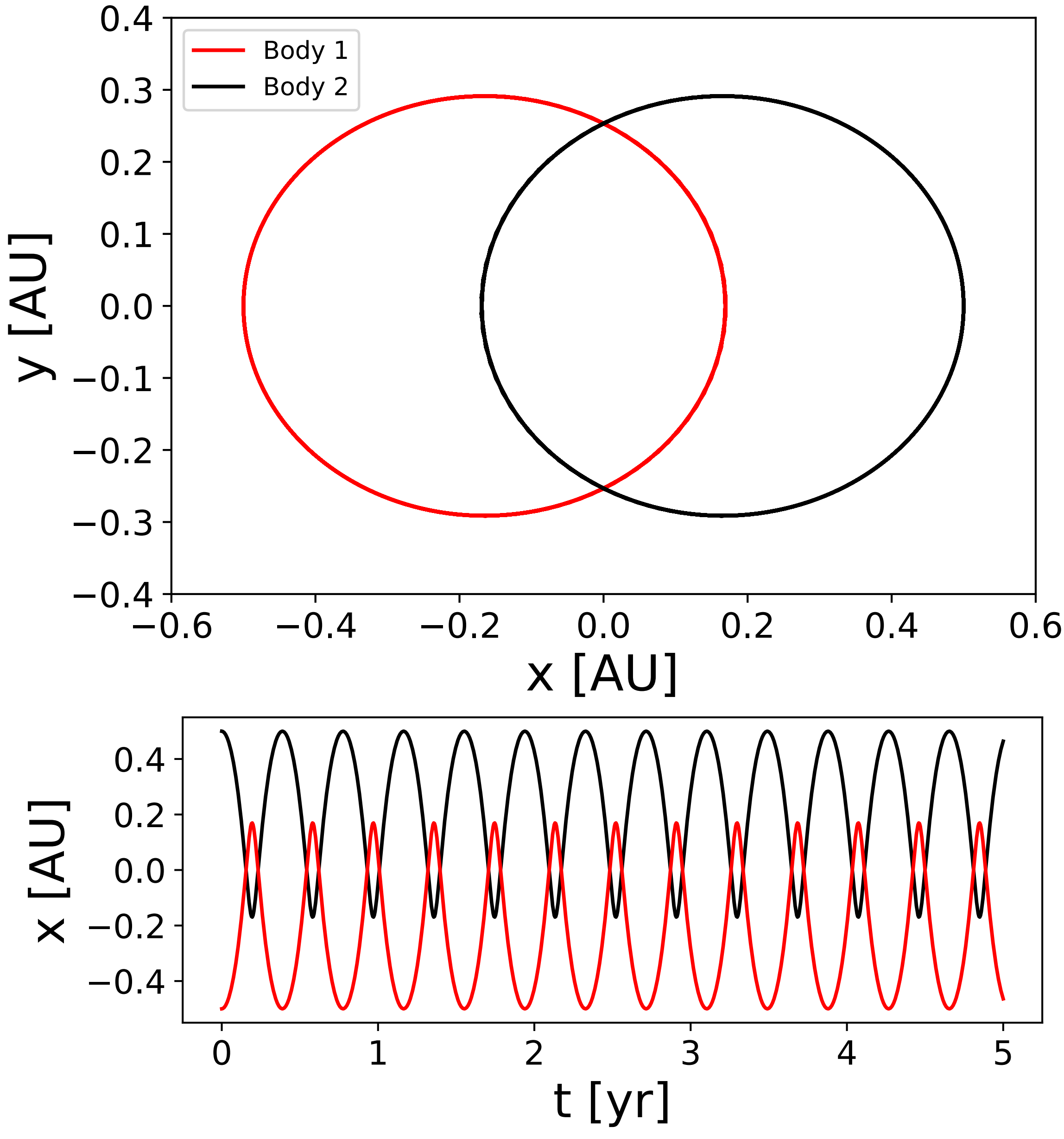
Detailed guidance on how to call and use the *odeint* package can be found here:
https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.odeint.html

3) Plot orbits (i.e. y against x in units of AU) over a period of $t = 5$ years. In addition, plot x_1 and x_2 against time. Your results should match the figures shown on the next page. *Save these plots and add them as the first figure in your draft project report*.

4) As a final step, determine if your numerical solution is accurate by verifying the total energy of the system is conserved. To do this, plot the relative change in the total energy of the system against time, $\Delta E = (E(t) - E(0))/E(0)$, where $E(t)$ and $E(0)$ are the total energy (i.e. kinetic and potential, $E = K + U$) of the system at time t and $t = 0$, respectively. For the default tolerance settings in *odeint*, you should find $|\Delta E| < 10^{-5}$.

You may have to adjust the *odeint* integration tolerances (given by the inputs *atol* and *rtol*) to ensure a well converged solution, particularly for any subsequent, more complicated models that use multiple bodies and/or have regular close encounters (e.g. Burrau's problem). Try a numerical experiment by changing the integration tolerances. How does this change the accuracy of your solution? Is energy still conserved?

Note that a common error is to use the virial theorem here, rather than calculate the total energy for the system. *Do not do this!* The virial theorem will *not hold in general* for all N-body systems.


Body 1
Body 2
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
y [AU]
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
x [AU]
0.4
0.2
0.0
−0.2
−0.4
x [AU]
0
1
2
3
4
5
t [yr]