MAS212 Assignment #2: Investigating a dynamical system

Dr. Sam Dolan (s.dolan@sheffield.ac.uk)

In this assignment you will investigate the dynamics of a system governed by a pair of second-order differential equations,

$$\begin{aligned} \ddot{x} &= -x - 2xy, \\ \ddot{y} &= -y - x^2 + y^2. \end{aligned} \tag{1}$$

Here dots denote derivatives with respect to time, so $\ddot{x} = \frac{d^2x}{dt^2}$, $\ddot{y} = \frac{d^2y}{dt^2}$. You will use Python to calculate some trajectories $(x(t), y(t))$, and investigate some key properties of this system.

Imagine a ball rolling around in a basin of height $h = h(x, y)$. Let $\mathbf{x}(t) = [x(t), y(t)]$ be the two-dimensional vector describing the position of the ball at time t. The ball is accelerated downhill by the force of gravity. In a uniform gravitational field (with $g = 1$), and in absence of friction, Newton's second law implies that that the acceleration vector $\ddot{\mathbf{x}}$ is proportional to the gradient of the height function, $\frac{d^2\mathbf{x}}{dt^2} = -\boldsymbol{\nabla} h$. In coordinate form, this equation is

$$\ddot{x} = -\frac{\partial h}{\partial x}, \qquad \ddot{y} = -\frac{\partial h}{\partial y}. \tag{2}$$

The equations (1) arise from a particular choice of height function,

$$h(x, y) = \frac{1}{2}\left(x^2 + y^2\right) + x^2 y - \frac{1}{3}y^3, \tag{3}$$

corresponding to a landscape with a central basin between three mountains and three valleys.

The Submission: A completed assignment will comprise a `.pdf` of your report and a single code file (`.py` or `.ipynb`). The report should has sections/subsections corresponding to the numbered parts below. It should also include a conclusion paragraph. As well as addressing the parts of the brief, your report should also be a coherent document in its own right, which could be read by someone who has not seen this brief. The report should be no more that **six sides** including figures. If necessary, additional material can be included in an appendix. You can also add a bibliography, if you have done further reading (see 'Hints' below). The report should be accompanied by *one* Python script or Jupyter notebook. Note that the report will carry *significantly* more credit than the code. Please use LaTeX to write your report (or discuss with the lecturer if not possible).

The Deadline: The deadline for submission is found on the course website.
Files should be submitted at `https://somas-uploads.shef.ac.uk/mas212`.

Part 1: Introduction

This section should consist of text and mathematics only. No code is required for this section.

Begin your report with a statement of the problem you will be investigating, and state the differential equations (1).

Introduce the height function $h(x, y)$ in Eq. (3), and:

- Show that inserting the height function into the equations (2) leads to the equations (1).
- Show that the energy
$$E = \frac{1}{2}\left(\dot{x}^2 + \dot{y}^2\right) + h(x, y) \tag{4}$$
is a **constant of motion** along a trajectory (i.e. show that $\frac{dE}{dt} = 0$) (see footnote [1]).
- Find the coordinates of the **four** stationary points of $h(x, y)$ (see footnote [2]). Classify these stationary points.

By introducing a pair of new variables $p_x = \dot{x}$ and $p_y = \dot{y}$, show that Eqs. (1) may be written as a set of four first-order equations,

$$\begin{aligned} \dot{x} &= p_x \,, & \dot{p}_x &= ? \,, \\ \dot{y} &= p_y \,, & \dot{p}_y &= ? \,, \end{aligned} \tag{5}$$

You should fill in the right-hand sides.

Part 2: Bound trajectories

(a) A contour plot. Create a contour plot (`plt.contour()`) of the height function $h(x, y)$ similar to that shown in Fig. 1. Find the value of h on the 'special' contour that joins three stationary points (i.e. evaluate h at one of the stationary points on this contour). In the text, briefly describe the key features of your plot, such as (i) a symmetry property, (ii) the 'central basin', (iii) the value of h on the contour joining 3 stationary points, and (iv) the relative positions of three 'mountains' and 'valleys'.

(b) Example trajectories. Figure 2 shows three example trajectories in the 'central basin'. In each case I took an initial condition with $y = 0$ and $p_x = 0$, and I chose initial values of x and p_y such that the energy E defined in (4) is less than 1/6. By changing the initial values, one can produce a variety of interesting trajectories.

Write Python code to calculate trajectories by numerically solving the ODE system in first-order form (5) with the function `scipy.integrate.solve_ivp()`. Create three labelled plots, showing three trajectories. The trajectories you choose should be somewhat different to those shown in Fig. 2, and also qualitatively-different from one another. Include these plots as a figure in your report, and describe the trajectories in the text of your report. (*Bonus mark:* plot the contour $h(x, y) = E$ along with the trajectory; see Fig. 3 for an example.)

[1] Apply the chain rule and use Eq. (2). Note that $\frac{d}{dt}(\dot{x}^2) = 2\dot{x}\ddot{x}$, for example.

[2] A stationary point is a point (x, y) at which $\frac{\partial h}{\partial x} = 0$ and $\frac{\partial h}{\partial y} = 0$.

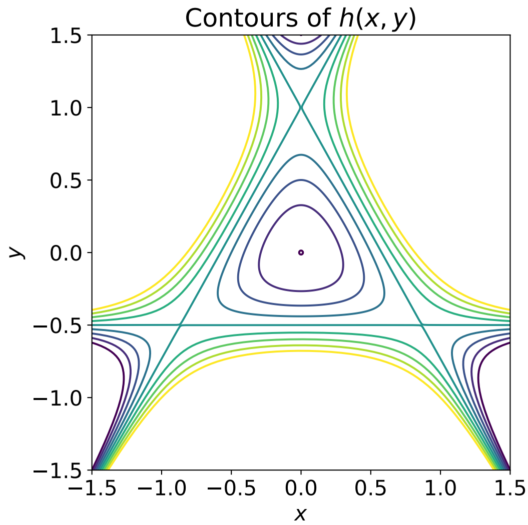

Figure 1: A contour plot of the height function $h(x, y)$ defined in Eq. (3).

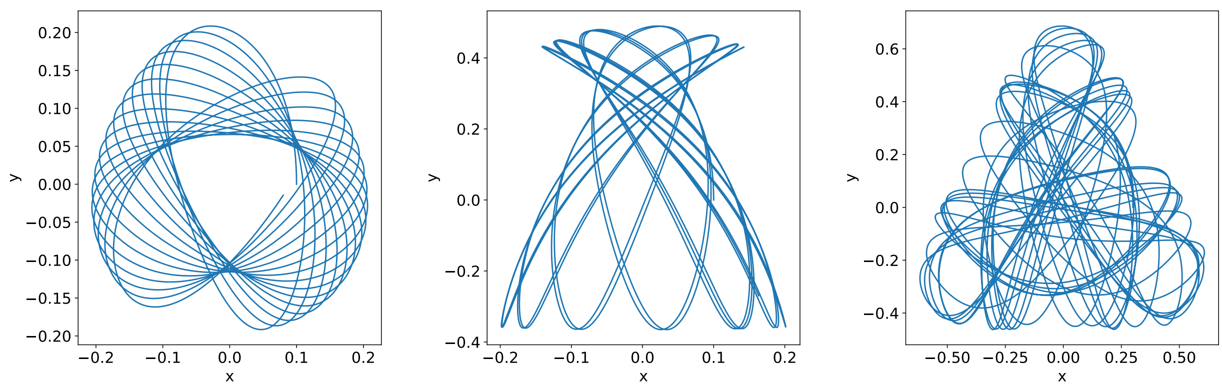

Figure 2: Three example trajectories. In each case, the initial condition was $x(0) = x_0$, $y(0) = 0$, $p_x(0) = 0$ and $p_y(0) = v_0$, where x_0 and v_0 are positive constants.

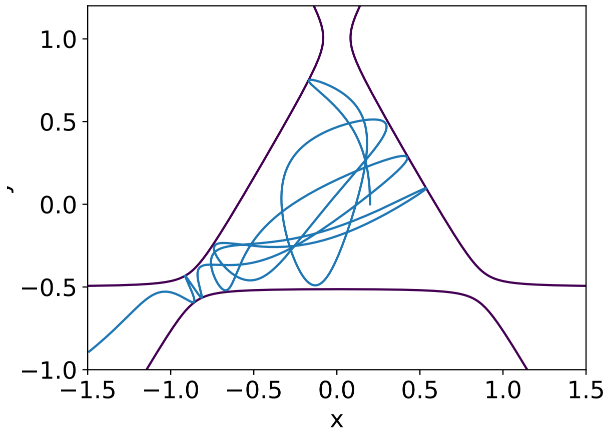

Figure 3: An example of a trajectory with energy $E > 1/6$ that escapes from the central basin via the lower left exit and rolls down the hill.

Part 3: Escape trajectories

If the energy E is less than the critical value $E_c = 1/6$, then the trajectory is confined to the central basin, and cannot escape. If the energy exceeds the critical value, $E > E_c$, then it possible (but not necessarily guaranteed) that the trajectory may escape through one of the three exits. Even so, it may spend a long time in the central basin before doing so. Predicting which exit the trajectory will escape from is not straightforward.

(a) An escape trajectory. Find an example of an escape trajectory, and show a plot in a figure in your report. See Fig. 3 for one such example. *Warning: You may have problems with numerical divergences for escape trajectories when using* `solve_ivp()`*. Please see the hint on the last page.*

(b) *(Challenging)* Choose **only one** of the two tasks below:

1. An eternal orbit? In the case $E > E_c$, it is possible that there may exist one or more trajectories that remain with the central basin forever. By changing the initial conditions, search for long-lived trajectories. In the text, describe how you conducted the search. If you find one or more examples of a long-lived orbit, show examples of the trajectories in a figure, and describe their properties in the text. For example, are the orbits periodic?

2. Exit by initial conditions. Figure 4 shows the fate of 50×50 trajectories that start with $y(0) = 0, p_y(0) = 0$ and $x(0) = x_0$ and $p_y(0) = v_0$, with x_0 and v_0 in the ranges 0 to 0.8. The colour indicates whether the trajectory was still in the central basin after $t = 100$ (black), or escaped via the top, bottom-left or bottom-right corners. Write code to compute similar plot(s). You may wish to consider other choices of initial condition. Describe your plot(s) in the report. Are the boundaries of the escape basins regular? Present evidence to support your answer.

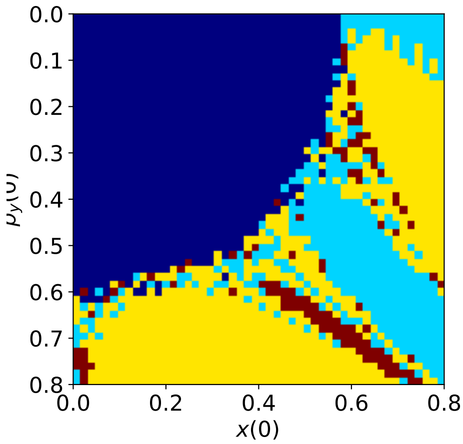

Figure 4: Showing the fate of a trajectory with initial conditions $x(0) = x_0$, $p_y(0) = v_0$ and $y(0) = 0$, $p_x(0) = 0$ after $t = 100$. *Key:* Dark blue = remains in the central basin. Light blue / yellow / red: escaped via exit 1, 2, 3.

Conclusion

Add a short conclusion to your report. This should be three sentences or so, summarising the most important findings of your work.

Guidance:

This assignment will count for $\sim 35\%$ of your module mark, and thus it should be a substantial piece of work. For this assignment, **fair means** include: asking the lecturer for advice; reading online materials; using and adapting short code snippets from lectures; etc. Please avoid verbatim copying, even in the introduction, and please cite all sources used. **Unfair means** include (but are not limited to): sharing or distributing files; copying-and-pasting from work that is not your own; posting your work online; passing off other's work as your own, etc.

Note that all submissions will be checked for excessive similarities. Where there is circumstantial evidence of unfair means, I reserve the right to award zero marks for this assignment.

Hints:

- You may find it helpful to do some background reading on the Henon-Heiles system.
- The default accuracy of `solve_ivp()` is insufficient for finding trajectories accurately. You should use the `atol` and `rtol` optional arguments to control (and improve) the accuracy.
- For Part 3, it is necessary to stop the numerical integration once the trajectory leaves the central basin. One way to tackle this is by using the 'events' optional argument in the `solve_ivp` function (`https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html`). Another way by writing your own code using e.g. the midpoint method, so that you can terminate the loop once soon as $x^2 + y^2 > r^2$ (with $r = 1.3$, say).