FEEDBACK ON MAS2008 ASSIGNMENT 2 (INVESTIGATING A DYNAMICAL SYSTEM)

Most people did a good job, and average marks were again high.
Marks (out of a total of 100) were assigned as follows:

- There were 6 marks each of the first five tasks (conservation of energy, the complex form of the equations, finding the stationary points, factorisation of $V - 1/6$, the curve where $V = 0$).
- There was another 6 marks for explaining why the system is invariant under rotation by $2\pi/3$.
- There were 8 marks for `draw_contours()`: roughly half for correctness of the code and half for the quality of the resulting diagram.
- Similarly, there were 8 marks for `show_trajectory()`: roughly half for correctness of the code and half for the quality of the resulting diagram.
- There were 3 marks for defining the `escape()` function, 6 marks for `solve()`, 3 for `escape_time()`
- There were 8 marks for finding interesting examples of trajectories showing the full range of possibilities.
- There were 8 marks for finding and plotting trajectories with long escape times.
- There were 20 marks for presentation. This covered organization into sections, general quality of explanations, use of LaTeX, spelling, grammar and so on.

Conservation of energy

A few people effectively said that $dE/dt = 0$ because t does not appear explicitly in the formula for E. A few more people tried to calculate dE/dt using `sympy` but set things up in such a way that `sympy` made the equivalent mistake. Apart from that, most people answered this correctly.

Complex form of the equations

Most people did this correctly, but with varying degrees of clarity. When asked to prove that $A = B$, some people wrote $A = B$ as if they already new that it was true, and then manipulated both sides until they were obviously equal. Although this style of argument does not cause too much trouble in logically simple cases like the ones considered here, it is not valid in general and can lead to error in cases where the logical complexity is even a little bit higher. Some marks were deducted for this sort of thing.

Finding the stationary points

Again, this was usually done correctly but there were some deductions for lack of clarity.

Factorisation of $V - 1/6$

Some people did this by hand, with a range of slightly different methods. Where people lost marks, it was usually for working backwards from the desired answer without distinguishing properly between things that are true by definition and things that are supposed to be proved. Some people used `sympy` with varying degrees of success.

The last part of the task was to describe the points (x, y) where $V(x, y) = 1/6$. For this to happen, at least one of the equations $x = -1/2$, $y = (x-1)/\sqrt{3}$ or $y = (1-x)/\sqrt{3}$ must hold. Thus, the relevant set is the union of three straight lines, which are clearly visible in the contour plot. Some people said that two of the above three equations must hold, which forces (x, y) to be one of the stationary points. That is not correct: it is enough for any one of the three equations to hold.

The curve where $V = 0$

Most people did this correctly, either by hand or using `sympy`.

Rotational invariance

In the complex plane, rotation through $2\pi/3$ corresponds to multiplication by $\omega = e^{2\pi i/3}$. This has $|\omega| = 1$ and $\omega^3 = 1$ so $|\omega z| = |z|$ and $\mathrm{Re}((\omega z)^3) = \mathrm{Re}(\omega^3 z^3) = \mathrm{Re}(z^3)$. From this we get

$$V(\omega z) = \tfrac{1}{2}|\omega z|^2 - \tfrac{1}{3}\mathrm{Re}((\omega z)^3) = \tfrac{1}{2}|z|^2 - \tfrac{1}{3}\mathrm{Re}(z^3) = V(z).$$

Unfortunately, most people used other methods that require far more work than this.

Drawing the contours

In some cases marks were deducted for missing docstrings, failure to provide default values for R and n, failure to set the aspect ratio so that the circles are circular, or failure to adjust the picture correctly when R is not equal to 2.

Escape detection

The brief says that the `escape()` function should be positive inside the circle of radius two, and negative outside. Quite a few people lost a mark by doing it the other way around.

Solving the differential equations

Marks were deducted for various (mostly minor) failures to comply with the details of the specification, including missing docstrings.

To get reliable results, especially when searching for long trajectories, it is necessary to ask for enhanced accuracy by supplying the arguments `atol` and `rtol` to `solve_ivp()`. Very few people did this, but I did not impose any penalties, bbecause this issue was not mentioned in the brief.

Plotting the trajectories

Issues were similar to those for drawing the contours.

Examples of trajectories

Typically 4 or 5 marks were awarded for providing a selection of examples of the most obvious types. More exotic trajectories were required for the full 8 marks.

Escape time

Suppose you call `solve_ivp()` specifying a time interval of length 10 and 1000 time steps of size 0.01. Suppose that the solution escapes at step k, close to time $0.01k$. You might think that `sol.t[-1]` and `sol.t_events[0][0]` would be equal to $0.01k$, but that is not correct. You can only expect that $0.01k$ will be within 0.01 of the actual escape time, but `solve_ivp()` sets `sol.t[-1]` and `sol.t_events[0][0]` to a slightly different time which approximates the actual escape time to much higher accuracy. There was a one mark penalty for approaches that are equivalent to returning $0.01k$.

Long trajectories

The brief specified that you should include plots of the longest trajectories that you found. Quite a few people did not do that, and so lost four marks or so.

Presentation

Some LaTeX-specific points:

- Chains of equations should be laid out using `\begin{align*} ... \end{align*}`.
- If you want brackets around a fraction or other vertically-extended expression, you should use `\left( ... \right)` to produce taller brackets that will enclose the whole expression.
- If you have an equation that you want to call Equation A, you should tag it by including `\tag{A}` in the equation.

Using sympy

Various parts of the brief can be done using `sympy` as follows.

```
import sympy

x, y, p, q, s = sympy.symbols('x y p q s', real=True)
V = x**2/2 + y**2/2 + x*y**2 - x**3/3
T = p**2/2 + q**2/2
x_dot = p
y_dot = q
p_dot = -x + x**2 - y**2
q_dot = -y - 2*x*y
V_dot = V.diff(x)*x_dot + V.diff(y)*y_dot
T_dot = T.diff(p)*p_dot + T.diff(q)*q_dot
E_dot = V_dot + T_dot
z = x + y * sympy.I
z_dot = p + q * sympy.I
z_ddot = p_dot + q_dot * sympy.I
half = sympy.S('1/2')
sixth = sympy.S('1/6')
root3 = sympy.sqrt(3)
err0 = sympy.simplify(E_dot)
err1 = sympy.simplify(z_ddot - (- z + z.conjugate() ** 2))
stationary_points = sympy.solvers.solve([p_dot, q_dot],[x, y])
err2 = V - (abs(z) ** 2 / 2 - sympy.re(z ** 3)/3)
err3 = T - (abs(z_dot) ** 2 / 2)
err4 = sympy.simplify(V - sixth - (x + half)*(y - (x-1)/root3)*(y + (x-1)/root3))
m = (3 + s**2)/(2*root3*(1-s**2))
err5 = sympy.simplify(V.subs({x : m * root3, y : m * s}))
print(f"Checking conservation of energy: {err0}")
print(f"Checking d^2z/dt^2             : {err1}")
print(f"Checking V in terms of z       : {err2}")
print(f"Checking T in terms of z       : {err3}")
print(f"Checking formula for V - 1/6   : {err4}")
print(f"Checking curve where V=0       : {err5}")
print(f"Stationary points: {stationary_points}")
```