This is a version of HW3 that aims to make the problems more readable.

This is an individual assignment, so the answers you submit must be your own work. Before you upload the file to Canvas, execute the final version of the file to make sure that it runs properly and produces your intended answers. Insert your code to implement the functions as specified (without changing function names and/or arguments). Do not import from libraries other than those that are already imported into the template (such as numpy).

> Please DO NOT use this file as the basis for your homework submission.

> Add your code to `hw3_template.py` to create the file that you submit.

1. This question involves using the center-difference formula for the first derivative from Kutz' Table 4.2 to estimate the value of df/dx for the function f(x) = sin(x) at x = 1.0.

For those who do not have the text near at hand, the formula (with x as the independent variable and h as the spacing) is:

$$f'(x) = 1/(12*h)*(f(x-2h) - 8*f(x-h) + 8*f(x+h) - f(x+2h))$$

Your particular tasks are as follows:

- Compute the relative error in the derivative estimate for sample spacing values $h = 2^{-n} \text{ for } n \in \{0, 1, ..., 18\}$.
- Create a log-log plot of the error as a function of sample spacing using the `plot_errors()` function.
- Based on your plot, identify the order of truncation error and estimate the sample spacing at which the roundoff error becomes significant.

2. This question involves looking at the convergence properties of a well-known method for numerical integration or quadrature known as Simpson's 3/8 rule, described by Eq.(4.2.6c) in the text, for the function f(x) = sin(x) between 0 and 1.

Your particular tasks are as follows:

- Compute the relative error in the integration estimate for panel counts of $p = 4^n \text{ for } n \in \{0, 1, ..., 9\}$.
- Create a log-log plot of the error as a function of sample spacing using the `plot_errors()` function.
- Based on your plot, identify the order of truncation error and estimate the sample spacing at which the roundoff error becomes significant.

3. The models used by engineers often involve linear differential equations. A familiar example is the damped linear (or harmonic) oscillator:

$$\frac{d^2y}{dt^2} + c\frac{dy}{dt} + y = f(t)$$

or y'' + c*y' + y = f(t).

Linear equations are "friendly" because we can solve them analytically and they support superposition. However, the "real world" is not that friendly; most systems are actually nonlinear and we typically cannot write down analytic solutions. Instead, we often employ numerical methods to compute approximate solutions for more realistic systems modeled by nonlinear differential equations. The following problems deal with a classic nonlinear second-order ODE, the Van der Pol equation, which describes an oscillator with nonlinear damping:

$$\frac{d^2y}{dt^2} - \mu(1-y^2)\frac{dy}{dt} + y = 0$$

or y'' - mu*(1-y**2)*y' + y = 0

3a) The template includes an implementation of an Runge-Kutta ODE solver `rk_solve()` that calls a function `rk2_step()` that computes a single step for the second order RK method.

Your initial task is to implement the function `rk4_step` to compute a single step of the 4th order Runge-Kutta method.

Use `rk_solve` to call your `rk4_step` function to compute a numerical solution for y(t) on the interval t=[0,100] with stepsize h = 0.1, Van der Pol mu (a.k.a. epsilon) of 0.1, and initial conditions y(0)=0.25, y'(0)=0.

3b) What happens when you try this simulation with 500 steps with a Van der Pol parameter value (mu or epsilon) of 10.0? What is the steady-state behavior with 5000 steps?

3c) Compute a numerical solution for the system of problem 3b using the RK45 method in the library function `scipy.integrate.solve_ivp` that incorporates stepsize control. Follow the example for simulating the Lotka-Volterra system on the documentation page:

https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html

Compare and contrast the results of your RK4 and numpy's RK45 results.

Some info on the van der Pol oscillator:

The van der Pol equation or oscillator was formulated by Balthasar van der Pol about 100 years ago as a model for certain aspects of the behavior of electrical components with nonlinear characteristics. At that time, the relevant elements were triode vacuum tubes; today similar characteristics are provided by transistors.

The equation can be stated in various forms (one equivalent version is attributed to Lord Rayleigh), but we will stick with something very close to van der Pol's original version:

$$\frac{d^2y}{dt^2} - \epsilon(1-y^2)\frac{dy}{dt} + y = 0$$

(Some people call the parameter μ, but I am used to using ϵ, so I will stick with that notation.) Considering a mechanical interpretation, this system looks much like a linear oscillator $y'' + \delta y + y = 0$, but the damping constant is replaced by a nonlinear function ($c \rightarrow \epsilon y^2 - 1$). We can think of this roughly as a system where large amplitude vibrations are heavily damped, while small amplitude oscillations have negative effective damping (i.e. there is some source that pumps energy in to make small oscillations grow).

Before resorting to numerical solutions (which is the point of this class, so we will not avoid it for long), let's consider what we can determine about this system analytically.

Start, as usual by converting to a first-order system:

$$\begin{aligned} y' &= v \\ v' &= -y + \epsilon(1-y^2)v \end{aligned}$$

1. The usual starting place is to seek equilibrium solutions where the rates vanish (equal zero).

The first rate equation gives $y' = 0 \implies v \equiv 0$

Plugging into the second rate equation gives: $v' = 0 \implies 0 = -y + \epsilon(1-y^2)\cancel{v} = 0 \implies y \equiv 0$

So there is an equilibirum solution at the origin of the yv-plane (or phase plane): $y(t) \equiv v(t) \equiv 0$

2. The next question is to determine the stability of the equilibrium solution. If you perturb the system slightly from the equilibrium, do solutions remain small and stay near the equilibrium (the stable case) or does a small perturbation cause solutions to grow large and leave the neighborhood of the equilibrium (the unstable case)?

The stability question is determined by the behavior for small displacements from the equilibrium where the linear terms dominate. Linearizing near the origin, the $y^2 v$ term becomes negligibly small and gives the linear system:

$$\begin{aligned} y' &= v \\ v' &= -y + \epsilon v - \cancel{\epsilon y^2 v} \end{aligned}$$

or equivalently:

$$y'' - \epsilon y' + y = 0$$

This system has negative damping, and solutions grow exponentially.

We now know that the equilibrium point that we found is unstable.

3. If small solutions grow, do they continue to grow to indefinitely large amplitudes? There are some tricks to show that the growth eventually stops (because the large damping during large displacement portions of the oscillation cancel out the energy inmput during the small displacement portions), but they are not of general usage.

Instead of pursuing that approach, let's try to determine what the actual steady-state behavior looks like for a small but non-zero parameter value; i.e. for $0 < \epsilon << 1$.

Answering this question brings us into the areas of perturbation theory and asymptotic expansions. These are big areas, so we will avoid getting into all the details and instead aim for a quick tour.

A basic perturbation approach is consider a system close to one that we knkow how to solve and to look for a solution in the form of an asymptotic expansion where each term is small compared to earlier terms for sufficiently small values of the parametter that measures how far we are from the known system.

Here we perturb off the undamped linear oscillator $y'' + y = 0$ and ϵ is our perturbation parameter. Thus we are looking for a steady-state solution of the form:

$$y(t;\epsilon) = y_0(t) + \epsilon y_1(t) + \epsilon^2 y_2(t) + \dots$$

A typical analysis approach is to (1) differentiate this formal solution, (2) substitute into the nonlinear equation, (3) collect powers of ϵ, and (4) solve term-by-term to desired accuracy.

1. The differentiated formal solutions are:

$$y(t;\epsilon) \to y_0(t) + \epsilon y_1(t) + \epsilon^2 y_2(t) + \dots$$

$$y'(t;\epsilon) \to y_0'(t) + \epsilon y_1'(t) + \epsilon^2 y_2'(t) + \dots$$

$$y''(t;\epsilon) \to y_0''(t) + \epsilon y_1''(t) + \epsilon^2 y_2''(t) + \dots$$

2. Sustituting in (just keeping linear terms in ϵ for simplicity) gives:

$$y_0''(t) + \epsilon y_1''(t) + \dots - \epsilon \left[1 - (y_0(t) + \epsilon y_1(t) + \dots)^2\right] (y_0'(t) + \epsilon y_1'(t) + \dots) + (y_0(t) + \epsilon y_1(t) + \dots) = 0$$

3. Collecting powers of ϵ gives:

$$\begin{aligned}
\epsilon^0 :& \quad y_0'' + y_0 = 0 \\
\epsilon^1 :& \quad y_1'' + y_1 = \epsilon(1 - y_0^2)y_0' \\
\epsilon^2 :& \quad y_2'' + y_2 = \epsilon\left[(1 - y_0^2)y_1' - 2y_0y_1y_0'\right] \\
& \quad \vdots \\
\epsilon_n :& \quad y_n'' + y_n = \dots
\end{aligned}$$

4. Note that at each power of ϵ, we need to solve the unperturbed linear oscillator with progressively more complicated forcing functions on the right-hand side (RHS).

Let's start with the 0^{th}-order solution which is straightforward. Solutions are $\sin(t)$ and $\cos(t)$; to simplify things, let's choose initial velocity to be zero to eliminate the $\sin(t)$ term. We then have:

$$y_0(t) = A\cos(t) \quad \text{with derivative} \quad y'(t) = -A\sin(t)$$

We can now substitute in to evaluate the RHS of the ϵ^1 equation:

$$\begin{aligned}
y_1'' + y_1 &= \epsilon(1 - A^2\cos^2(t))(-A\sin(t)) \\
&= \epsilon A(-\sin(t) + A^2\cos^2(t)\sin(t)) \\
&= \epsilon A\left[-\sin(t) + A^2\frac{1}{4}(\sin(t) + \sin(3t))\right] \\
&= \epsilon A\left[(-1 + \frac{A^2}{4})\sin(t)\right] + \epsilon A\sin(3t)
\end{aligned}$$

The solution will have a homogeneous piece (with $\sin(t), \cos(t)$) and a particular solution associated with each forcing term. The solution driven by $\sin(3t)$ is again proportional to $\sin(3t)$. However, the $\sin(t)$ term provides resonance forcing and produces a term that grows like $t\cos(t)$ and (after a sufficiently long time) that term would become larger than the y_0 term and would violate the asymptotic expansion assumption. Thus, we "suppress secular terms" and require that the coefficient of the resonant forcing term must vanish:

$$\implies -1 + \frac{A^2}{4} = 0 \implies |A| = 2$$

We have now completed the first term in the asymptotic expansion:

$$y(t;\epsilon) = y_0(t) + \dots = 2\cos(t) + \dots$$

We now have reason to expect asymptotic behavior to be a "limit cycle" oscillation which is sinusoidal with amplitude 2. (Hopefully, you see that in your numerical solution with $\epsilon = 0.1$)

5. Now we move beyond the basics and ask what happens when ϵ becomes large...

Rescale time:

$$t \to \epsilon\tau, \quad \frac{d}{dt} \to \frac{d\tau}{dt}\frac{d}{dt} = \frac{1}{\epsilon}\frac{d}{d\tau}$$

Each transformed time derivative produces a factor of $1/\epsilon$:

$$\frac{1}{\epsilon^2}\frac{d^2y}{d\tau^2} - \frac{\cancel{\epsilon}}{\cancel{\epsilon}}(1 - y^2)\frac{dy}{d\tau} + y = 0$$

For large ϵ, treat $\nu = \frac{1}{\epsilon^2}$ as the small parameter. To leading order, ignore the second derivative term and integrate:

$$d\tau = \frac{dy}{y} - y\,dy \implies \tau + C = ln(y) - \frac{1}{2}y^2$$

During the "slow phase", solutions follow this curve (which looks much like a parabola with the opening in the negative τ direction). When the slow decay reaches $y \approx 1$, the solution "falls off the curve" and jumps quickly across to a similar curve (but translated horizontally and reflected vertically).

We are skipping some details, but I hope this is enough to motivate the difference between the harmonic limit cycle oscillation for $\epsilon << 1$ and the relaxation oscillations for $\epsilon >> 1$.

While this sort of topic tends to provide an opportunity for symbolic computing, this behavior with multiple time scales is also enough to justify the application of more sophisticated numerical methods (e.g. predictor-corrector or RK45 with automatic stepsize adjustment).

6. Finally, think about what would happen when you couple 2 such oscillators together. Consider, for example, linking them with a lilnear spring. What do you thin will happen? What might the steady-state behavior look like?