AMATH 301 – Autumn 2019
Homework 8

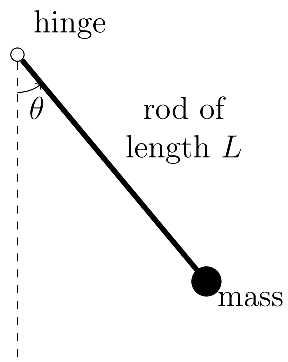
Due: 7:00pm, December 6, 2019.

Instructions for submitting:

- Scorelator problems are submitted as a MATLAB script (`.m` file). You should **NOT** upload your `.dat` files. The `.dat` files should be created by your script. You have 5 attempts on Scorelator.
- Writeup problems are submitted to Gradescope as a single `.pdf` file that contains text and plots. Put the problems in order and label each writeup problem. When you submit, **you must indicate which problem is which on Gradescope**. **All code you used for this part of the assignment should be included either at the end of the problem or at the end of your .pdf file.**

Scorelator problems

1. Consider a pendulum made by connecting a mass to a rod of length L to a hinge. This scenario is described by the following cartoon.



The equation for motion of this pendulum is given by the nonlinear pendulum equation with damping,

$$\ddot{\theta} = -\frac{g}{L}\sin(\theta) - \sigma\dot{\theta}. \tag{1}$$

Here $g = 9.8$ is the acceleration due to gravity, $L = 15$ is the length of the pendulum, and $\sigma = 0.15$ is a damping coefficient. Damping may come from friction at the hinge, friction between air and the pendulum, or other forces. The unknown that we must solve for is θ (theta), the angle of deflection of the pendulum from the vertical.

(a) By defining v (v for velocity) as the derivative of θ, $v = \dot{\theta}$, we can convert this single second-order ODE into a system of two first-order ODEs. Find this system of two first-order ODEs. Your system should look like

$$\dot{\theta} = \dots \tag{2}$$

$$\dot{v} = \dots . \tag{3}$$

(You do not need to submit anything here, but you will need to use it in what follows).

(b) We are now going to solve this system of differential equations in terms of θ and $v = \dot{\theta}$ over the time interval $0 \le t \le T$ for $T = 60$. In order to solve this system of differential equations we must define initial conditions (this is the initial setup of the system, then we let it go). We will use $\theta(t = 0) = \theta(0) = \pi/4$ and $v(t = 0) = v(0) = -0.75$. In other words, the pendulum starts with angle $\pi/4$ and we push it to the left with speed 0.75.

(c) Use the built-in MATLAB function `ode45` to solve the initial value problem for the nonlinear pendulum equation with damping. Make it so that `ode45` outputs the solutions at equally spaced times from $t = 0$ to $t = T = 60$ in step sizes of $\Delta t = 0.02$. Create a 3001×1 column vector containing the values of $\theta(t)$ at each time. Save the vector to `A1.dat`. Then create a column vector that contains the values of $v(t)$ and save it to `A2.dat`.

(d) Implement forward Euler,

$$\theta_{k+1} = \theta_k + \Delta t v_k, \tag{4}$$

$$v_{k+1} = v_k + \Delta t \left(-\frac{g}{L} \sin(\theta_k) - \sigma v_k \right), \tag{5}$$

with $\Delta t = 0.02$ to solve the initial value problem for the nonlinear pendulum. Make a 2×1 column vector which contains the absolute-value of the difference between the `ode45` solution found above ($\theta(t)$ and $v(t)$) and the forward Euler solution found here at the final time $T = 60$. Save this vector to `A3.dat`.

2. The nonlinear pendulum equation with damping is often too difficult to solve analytically, so people resort to a simpler model. When $|\theta|$ is small, one can use the small angle approximation for $\sin(\theta)$ which states that $\sin(\theta) \approx \theta$ and comes from the Taylor series for $\sin(\theta)$ about $\theta = 0$. This gives a linear second-order differential equation for the motion of the pendulum,

$$\ddot{\theta} = -\frac{g}{L}\theta - \sigma\dot{\theta}, \tag{6}$$

which is a significantly simpler model than (1).

(a) Using the same definition for v as above, $v = \dot{\theta}$, find a system of two first-order ODEs for the linear pendulum model. Your system should look like

$$\dot{\theta} = \dots \tag{7}$$

$$\dot{v} = \dots . \tag{8}$$

(You do not need to turn in anything here, but you will need to use this later).

(b) The system of ODEs you find should be linear and therefore can be written in matrix form,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}, \tag{9}$$

where

$$\mathbf{x} = \begin{pmatrix} \theta \\ v \end{pmatrix} \quad \text{and} \quad \dot{\mathbf{x}} = \begin{pmatrix} \dot{\theta} \\ \dot{v} \end{pmatrix}.$$

You should find $\mathbf{A}$ yourself from the system of linear equations you wrote down in (a). We are going to solve this system of differential equations in terms of θ and $v = \dot{\theta}$ over the time interval $0 \leq t \leq T$ for $T = 60$. We will use the same initial conditions as before, $\theta(t = 0) = \theta(0) = \pi/4$ and $v(t = 0) = v(0) = -0.75$.

(c) The forward Euler formula for a system of linear differential equations is

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \mathbf{A}\mathbf{x}_k.$$

Implement the forward Euler method with $\Delta t = 0.02$ to solve the initial value problem for the pendulum. Make a 2×1 column vector with the forward Euler solution ($\theta(t)$ and $v(t)$) at the final time $T = 60$. Save this vector to `A4.dat`.

(d) The backward Euler formula for a system of linear differential equations is

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \mathbf{A}\mathbf{x}_{k+1}.$$

Using some matrix algebra, we can write this as the following linear system

$$(\mathbf{I} - \Delta t \mathbf{A})\mathbf{x}_{k+1} = \mathbf{x}_k,$$

where $\mathbf{I}$ is the 2×2 identity matrix. At each time step of backward Euler, we must solve this linear system with the same matrix $\mathbf{I} - \Delta t\mathbf{A}$ but a different right-hand side vector $\mathbf{x}_k$. This is a good candidate for LU decomposition.

(i) Perform an LU decomposition on the matrix $\mathbf{I} - \Delta t\mathbf{A}$ to obtain matrices $\mathbf{L}$, $\mathbf{U}$, and $\mathbf{P}$. Save the matrix $\mathbf{LUP}$ in `A5.dat`.

(ii) Implement the backward Euler method with $\Delta t = 0.02$ to solve the initial value problem for the pendulum. At each time step, you should use the LU decomposition to solve the linear system. Make a 2×1 column vector with the backward Euler solution ($\theta(t)$ and $v(t)$) at the final time $T = 60$. Save this vector to `A6.dat`.

(e) The forward Euler formula may be rewritten as

$$\mathbf{x}_{k+1} = (\mathbf{I} + \Delta t \mathbf{A})\, \mathbf{x}_k,$$

where $\mathbf{I}$ is the 2×2 identity matrix. Forward Euler is stable for this problem if $|\lambda| < 1$ for all eigenvalues of $\mathbf{I} + \Delta t\mathbf{A}$. Find the eigenvalue of $\mathbf{I} + \Delta t\mathbf{A}$ that is largest in magnitude (or absolute value) when $\Delta t = 0.02$.

(f) Backward Euler will be stable for the linear pendulum problem if $|\lambda| < 1$ for all eigenvalues of $(\mathbf{I} - \Delta t\mathbf{A})^{-1}$. Find the eigenvalue of $(\mathbf{I} - \Delta t\mathbf{A})^{-1}$ that is largest in magnitude (or absolute value) when $\Delta t = 0.02$. You should use the `inv` command to compute the matrix inverse here.

(g) Create a 1×2 row vector that has the magnitude of the largest eigenvalue from part (c) as the first component and the magnitude of the largest eigenvalue from part (d) as the second component. Save this vector to `A7.dat`.

(h) Recall that forward Euler is obtained by using a forward difference to approximate the derivative and backward Euler is obtained by using a backward difference to approximate the derivative. The Leapfrog method is based on using a central difference to approximate the derivative:

$$\frac{\mathbf{x}_{k+1} - \mathbf{x}_{k-1}}{2\Delta t} = f(\mathbf{x}_k) \qquad \Rightarrow \qquad \mathbf{x}_{k+1} = \mathbf{x}_{k-1} + 2\Delta t f(\mathbf{x}_k).$$

This method is explicit like forward Euler because the formula we end up with only depends on past solution values. A major difference between this method and the Euler methods is that it is a *multistep* method – the iteration actually uses two past solution values: $\mathbf{x}_{k-1}$ and $\mathbf{x}_k$. This means that each time we generate a new iterate $\mathbf{x}_{k+1}$, we need to use the 'current' solution value $\mathbf{x}_k$ as well as the one before it, $\mathbf{x}_{k-1}$.

Since this method is a multistep method, we cannot start the iteration with just the initial values because the formula to compute $\mathbf{x}_1$ is

$$\mathbf{x}_1 = \mathbf{x}_{-1} + 2\Delta t f(\mathbf{x}_0),$$

which is a problem since $\mathbf{x}_{-1}$ isn't defined. To get around this, we perform a single step of Forward Euler to calculate $\mathbf{x}_1$, and then switch over to Leapfrog to calculate $\mathbf{x}_2$ and the remaining iterates.

Implement the Leapfrog method with $\Delta t = 0.02$ to solve the initial value problem for the linear pendulum. Make a 2×1 column vector with the Leapfrog solution ($\theta(t)$ and $v(t)$) at the final time $T = 60$. Save this vector to `A8.dat`.

(i) Use the MATLAB function `ode45` to solve the initial value problem for the linear pendulum. Make a 2×1 column vector with the `ode45` solution ($\theta(t)$ and $v(t)$) at the final time $T = 60$. Save this vector to `A9.dat`.

3. A researcher is trying to solve a system of linear differential equations using forward Euler for $0 \leq t \leq 100$. They start by using $\Delta t = 0.1$. Before solving the system they notice that the magnitude of the largest eigenvalue of the matrix $\mathbf{I} + \Delta t\mathbf{A}$ is equal to 1.01 but all of the other eigenvalues are less than 1 in magnitude. Which of the following options is the best way for the researcher to proceed? (Note that this is not necessary the best way for them to proceed, but the best of these options). Save your answer in `A10.dat`.

   A Implement forward Euler and accept the solution. Only one of the eigenvalues has magnitude greater than 1, the others are less than one so there is nothing to worry about.

B Implement forward Euler but be wary of the solution. Since the largest magnitude eigenvalue is not very much bigger than one the solution found should be pretty accurate.

C Increase Δt and check the eigenvalues again. Since forward Euler is unstable and the magnitude of the largest eigenvalue is greater than 1, we cannot trust the solution we have found. We need to increase Δt to try to make forward Euler stable.

D Decrease Δt and check the eigenvalues again. Since forward Euler is unstable and the magnitude of the largest eigenvalue is greater than 1, we cannot trust the solution we have found. We need to decrease Δt to tro to make forward Euler stable.

Gradescope problems

1. This problem mirrors Scorelator problem 1, the nonlinear damped pendulum equation. Here you will construct a phase portrait with θ on the horizontal axis and v on the vertical axis. You should turn in the plot created in (a)-(e), the explanations in (f), and the code you used for this problem.

   (a) Use `meshgrid` to generate a grid of points between $-3\pi \leq \theta \leq 3\pi$ and $-3 \leq v \leq 3$ with 25 equally spaced points in both directions.

   (b) Use the `quiver` function to draw a grid of arrows with components $(\dot{\theta}, \dot{v})$, where $\dot{\theta}$ and $\dot{v}$ are given by the ODEs of the nonlinear damped pendulum system found in Scorelator Problem 1 (a).

   (c) Include labels for the axes. Unlike the activity, you do not need to draw axes. (To make the θ symbol, type `\theta`).

   (d) We now want to include trajectories in the phase portrait.

      i. Use `ode45` to solve the system with the following initial conditions.
         - $\theta(0) = \pi,\ v(0) = 0.1$
         - $\theta(0) = \pi,\ v(0) = -0.1$
         - $\theta(0) = 2\pi,\ v(0) = -3$
         - $\theta(0) = -2\pi,\ v(0) = 3$

         Add these solution trajectories to the phase portrait. These solution trajectories should flow with the arrows from `quiver`.

   (e) Set the axes to display from $-3\pi \leq \theta \leq 3\pi$ and $-3 \leq v \leq 3$. You do not need to add a legend, because it is self-explanatory which trajectory belongs to which initial condition.

   (f) Explain what is happening physically.

      i. As time increases, the solutions with the different initial conditions are all going to the same physical situation. Describe that physical scenario in terms of the pendulum. **Hint:** How are $\theta = 0, \ -2\pi$, and 2π related physically?

ii. Compare the two solutions with initial conditions $(\theta(0), v(0)) = (\pi, 0.1)$ and $(\theta(0), v(0)) = (\pi, -0.1)$. Describe what is happening to these two solutions. Your description should include a physical description of the initial condition and what happens immediately after $t = 0$.

iii. Describe what is happening physically to the solution with the initial conditions $(\theta, v) = (2\pi, -3)$ and $(\theta, v) = (-2\pi, 3)$.

2. This problem mirrors Scorelator problem 2, the linear damped pendulum equation. Here you will construct a phase portrait with θ on the horizontal axis and v on the vertical axis. You should turn in the plot created in (a)-(d), the explanations in (e) and (f), and the code you used for this problem.

   (a) Use `meshgrid` to generate a grid of points between $-\pi/3 \leq \theta \leq \pi/3$ and $-\pi/3 \leq v \leq \pi/3$ with 25 equally spaced points in both directions.

   (b) Use the `quiver` function to draw a grid of arrows with components $(\dot{\theta}, \dot{v})$, where $\dot{\theta}$ and $\dot{v}$ are given by the ODEs of the linear damped pendulum system found in Scorelator Problem 2 (a).

   (c) Include labels for the axes. Unlike the activity, you do not need to draw axes. (To make the θ symbol, type `\theta`).

   (d) We now want to include trajectories in the phase portrait.

      i. Use `ode45` to solve the initial value problem with the same initial conditions as used in the Scorelator part of the homework. Plot this trajectory with yellow dots (`'y:'`) and linewidth 2.

      ii. Implement the forward Euler method with $\Delta t = 0.02$ to solve the initial value problem for the linear pendulum. Create vectors that contain the forward Euler approximations of $\theta(t)$ and $v(t)$ for all time steps from $t = 0$ to $t = T = 60$. Use these vectors to add a trajectory to the phase portrait. Plot this trajectory in cyan and with linewidth 2.

      iii. Repeat part (b) for both the backward Euler and Leapfrog methods. Use magenta for the backward Euler method and red for the Leapfrog method.

      iv. Include a legend so that the methods can be identified. In order to include the trajectories in your legend and not the quiver arrows, you should look at the documentation for the `legend` command in the section "Included Subset of Graphics Objects in Legend."

   (e) Assume that the `ode45` solution is the 'exact' solution. Which method is the most accurate here?

   (f) What happens to the error as time increases? Does it increase or decrease? How do you see this from the phase-portrait plot?