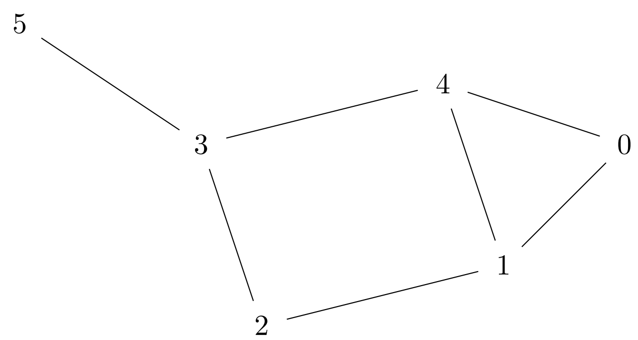
CS 111: Sample Problems for Midterm 2

See Exam e02 on the course GitHub page for midterm rules, syllabus, and more sample problems.



1. Consider the graph above.

1a. What is its Laplacian matrix?

1b. What is its adjacency matrix?

2. This is a simpler version of NCM exercise 7.23. It concerns a "double pendulum" which consists of two segments, the top segment hinged to a fixed point and the bottom segment hinged to the top one. The variables $u(t)$ and $v(t)$ are the angles that the two segments make with the vertical. The equations that describe the motion of the pendulum are given as a system of linear equations, that is, by the product of a matrix and a vector:

$$\begin{pmatrix} 2 & \cos(u-v) \\ \cos(u-v) & 1 \end{pmatrix} \begin{pmatrix} \ddot{u} \\ \ddot{v} \end{pmatrix} = \begin{pmatrix} -g\sin u - \dot{v}^2 \sin(u-v) \\ -g\sin v - \dot{u}^2 \sin(u-v) \end{pmatrix},$$

where $g = 9.81$ is the acceleration of gravity. The initial conditions are $u(0) = 1$, $v(0) = 0$, $\dot{u}(0) = 1$, and $\dot{v}(0) = 0$.

Write python code that uses `integrate.solve_ivp` to solve the equations for $0 \le t \le 100$, and then plots $u(t)$ versus $v(t)$. You need to:

- Decide how many elements the unknown vector y must have, and what they represent.
- Write a python function `ydot = f(t,y)` that computes the derivatives of the elements of the vector y. Hint: In this part, you can compute the elements of y that correspond to $\ddot{u}$ and $\ddot{v}$ by forming the 2-by-2 matrix in the equation above and then using `npla.solve()` to solve the matrix-vector equation.
- Write one line of python code that calls `integrate.solve_ivp`, which will have `f` as an argument.
- Write one line of python code that makes the correct plot from the output of `integrate.solve_ivp`.

Explain clearly in English what y represents, and what your python code does.

3. Consider the following ODE:

$$\dot{y}(t) = -y(t)/2, \qquad y(0) = 1.$$

This equation is simple enough that it has an exact solution in closed form, namely $y(t) = e^{-t/2}$. (You don't have to prove this.) Therefore, for example, we know $y(1) = e^{-1/2} = 0.6065....$ Let's see whether forward or backward Euler does a better job of approximating this exact answer.

3a. Using pencil and paper, take one step of the forward Euler algorithm, beginning at $t = 0$ and using step size $h = 1$, to get an estimate of $y(1)$. Show your work, and say clearly what the numerical value of your estimate is. (This is a much bigger step than you'd want to use in practice, but it makes the arithmetic easy.)

3a. Using pencil and paper, take one step of the backward Euler algorithm, beginning at $t = 0$ and using step size $h = 1$, to get an estimate of $y(1)$. Show your work, and say clearly what the numerical value of your estimate is.

4. The exam will may have a problem something like problem 4 on homework 5 (matrix condition numbers), except that we won't ask you to convert between decimal and hexadecimal.

5. The exam will surely have a problem like problem 5 on homework 5 (the loops in floating-point), except that we will give credit for any answer within 10% or so of correct, and we won't ask you to convert between decimal and hexadecimal.

6. The exam will surely have a problem like problem 2 on homework 7 (standard form of ODEs, possibly with multiple derivatives and multiple variables).