Homework 1: ODEs and dynamical systems

Kevin J. Kircher, Purdue ME 597

Directions:

- Students may work individually or in groups, but each student must upload their own solutions to Gradescope by **11:59 PM ET on Thursday, January 25**.
- Use any outside resources you want, but **cite your sources**. (If you really want to learn the material, I recommend seriously attempting the problems yourself before looking for outside help.)
- The TA will grade each problem quickly on a three-tier scale:
  - Zero points for a solution that's mostly unreadable or missing.
  - One point for a serious attempt that's not easy to read or is substantially incorrect.
  - Two points for a solution that's clearly readable and nearly or completely correct.

Problems:

1. (Refer to 'Linear ordinary differential equations' lecture slides.) Consider the linear scalar ODE IVP

$$x(1) = \frac{1}{2}, \ \frac{\mathrm{d}x(t)}{\mathrm{d}t} = -\frac{2x(t)}{t} + t - 1 + \frac{1}{t}.$$

   (a) Write down the definitions of t^{init}, $x(t^{\text{init}})$, $a(t)$, and $b(t)$ for this problem.

   (b) Find $g(t)$.

   (c) Calculate $\int_{t^{\text{init}}}^{t} g(\tau)b(\tau)\mathrm{d}\tau$.

   (d) Write down the solution $x(t)$.

2. (Refer to 'Linear ordinary differential equations' lecture slides.) Show that the solution to the linear vector ODE IVP

$$x(t^{\text{init}}) = x^{\text{init}} \in \mathbf{R}^n, \ \frac{\mathrm{d}x(t)}{\mathrm{d}t} = Ax(t) + b(t)$$

   is

$$x(t) = e^{(t-t^{\text{init}})A}x^{\text{init}} + e^{tA}\int_{t^{\text{init}}}^{t} e^{-\tau A}b(\tau)\mathrm{d}\tau.$$

   Hints:

   - Follow the steps from the scalar ODE IVP proof on slides 16–19.
   - Use properties of the matrix exponential from slide 31.
   - Use the product rule: for $G : \mathbf{R} \to \mathbf{R}^{n\times n}$ and $x : \mathbf{R} \to \mathbf{R}^n$,

$$\frac{\mathrm{d}}{\mathrm{d}t}(G(t)x(t)) = G(t)\frac{\mathrm{d}x(t)}{\mathrm{d}t} + \frac{\mathrm{d}G(t)}{\mathrm{d}t}x(t).$$

3. (Refer to 'Linear dynamical systems' lecture slides.) Download the Matlab files in the Github repository simple-climate-model. Fill in the missing code from the functions `nonlinearClimateSim` and `linearizedClimateSim`. Given the inputs in the `simpleClimateModel` script, these functions should return the true state trajectory (the solution to the nonlinear ODE) and the approximate state trajectory (the solution to the linearized ODE), respectively. Show the missing lines of code here. Show the graphs here that `simpleClimateModel` draws in figures 2 and 3.

4. (Extra credit, graded 0 or 1.) Modify the script `simpleClimateModel`, find some behavior that's interesting to you, and report what you learned. Include a graph or two.