Homework 0: Memories!

0.1 Introduction

The problems for this assignment focus on computational tools from EGR 103 you will need to solve early problems in this class. You will not be turning it in for a grade. You can use either MATLAB or Python (or both!). The solution will be posted on the Sakai site. If you have never used Python or MATLAB (or even if you have), there will be guidance on Pratt Pundit for how to do these problems.

0.2 Problems

0.2.1 Linear fit

Most of the elements you will be learning about in this class have some kind of linear relationship between the voltage across the element and the current through it. You have been given files for ten experimental runs where a particular voltage was applied to an element and the current through it was measured. Write code that automates the process of

- Loading data from a file,
- Determining the line of best fit for the current as a function of the voltage, and
- Storing the slope and intercept of the fit into arrays called *slope* and `intercept`, respectively.

Once all data files are finished, print a table with results of all 10 runs to the screen using scientific notation for all the numbers and making sure the numbers line up under each other. Note that the slopes will generally be positive but the intercepts may be negative.

0.2.2 Nonlinear fit

Some of the elements you will be learning about have a nonlinear relationship between the voltage across the element and the current through it. You have been given a file (on Sakai in Resources) with voltages across a circuit with a resistor and an LED in series. The file has three columns. The first is total voltage across the resistor and LED; you won't need this one. The second is the voltage over the resistor; you will divide this by 1000 to get the LED current. The third is the voltage across the LED; you will use this as the LED voltage. Given a model of:

$$i = Ae^{Bv} + C$$

use nonlinear regression to determine values for A, B, and C. Good initial values for the three coefficients are 1e-15, 20, and 0, respectively. Make sure the values are displayed onscreen. Once you have calculated the values, make a plot of the current through the LED as a function of the voltage across the LED that has a curve for the original data as a solid black line and the model as a dashed green line. Label the axes, include a legend, and turn on the grid.

0.2.3 Solve a linear algebra problem

While there are several methods for coming up with systems of equations for a circuit, with purely resistive circuits these methods generally lead to a system of linear algebra equations to solve. Write code to numerically solve for the unknown values v_1, v_2, and v_3 assuming known values of:

$$v_s = 10\text{V} \qquad\qquad R_k = 1000\ \Omega$$

for all $k = 1..7$ and given the following equations:

$$\frac{v_a - v_s}{R_1} + \frac{v_a - v_b}{R_3} + \frac{v_a - v_c}{R_2} = 0$$
$$\frac{v_b - v_s}{R_4} + \frac{v_b - v_a}{R_3} + \frac{v_b - v_c}{R_5} + \frac{v_b}{R_6} = 0$$
$$\frac{v_c - v_a}{R_2} + \frac{v_c - v_b}{R_5} + \frac{v_c}{R_7} = 0$$

0.2.4 Solve a linear algebra problem with a sweep

Once you have symbolically solved for a system of equations, you may want to see what impact changing one or more of the values will have on the results. Given the following system of equations, assume that you want to know how the values of i_1 and i_2 change as R_4 changes:

$$-10 + 1000i_1 + 3000(i_1 - i_2) = 0$$
$$3000(i_2 - i_1) + 750i_2 + R_4 i_2 = 0$$

Write code that will solve i_1 and i_2 as a function of R_4 for 2000 linearly spaced values of R_4 between 0 and 5000. Make a figure with two subplots - the top one should be a graph of i_1 as a function of R_4 and the bottom one should be a graph of i_2 as a function of R_4. You do not need to label or title these plots nor do you need legends or grids.

0.2.5 Solve a single ODE

Note Some offerings of EGR 103 did not cover setting up and solving differential equations. Please take a look at the Pundit page (Python:Ordinary Differential Equations) for it!

Once we start working with reactive elements (capacitors and inductors), the models will move from linear algebra to differential equations. Write code to calculate the values of y for 1000 times between 0 and 10 for the following differential equation:

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

with the following initial conditions and forcing functions:

(1) $y(0) = 0,\ f(t) = 1$

(2) $y(0) = 4,\ f(t) = 1$

(3) $y(0) = 0,\ f(t) = \sin(6t)$

(4) $y(0) = 4,\ f(t) = \sin(6t)$

For each of the four cases, your code should make a new figure that graphs $f(t)$ as a solid blue line and $y(t)$ as a solid red line. Each graph should have a legend and a title that looks like "Case N" where N is the case number.

0.3 Solutions

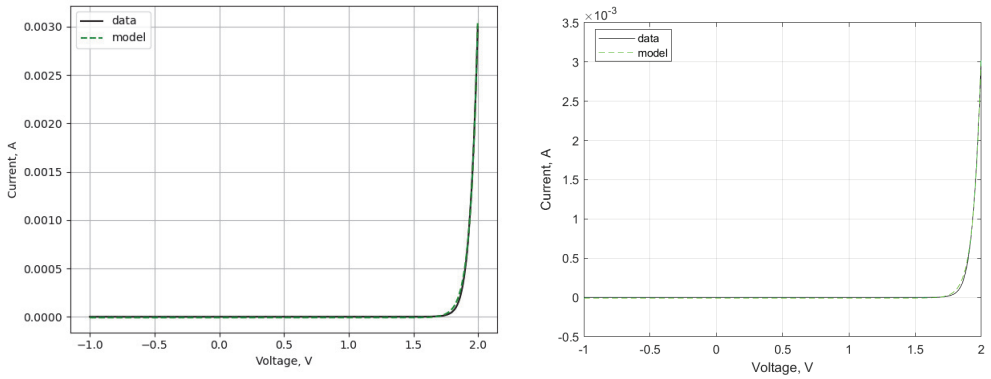
Example codes will eventually be located on Sakai; for now, here are the outputs from the programs:

0.3.1 Linear fit

```
     Slope   Intercept
 5.787e+00  1.440e-01
 4.640e+00  3.912e-02
 5.934e+00  1.174e-01
 5.234e+00  9.758e-02
 4.898e+00  1.168e-01
 5.923e+00  1.745e-01
 4.179e+00 -9.780e-02
 5.296e+00  8.575e-02
 5.206e+00  1.369e-01
 5.205e+00  4.678e-02
```

0.3.2 Nonlinear fit

$A \approx 1e-18$ $B \approx 18$ $C \approx 7.5e-6$

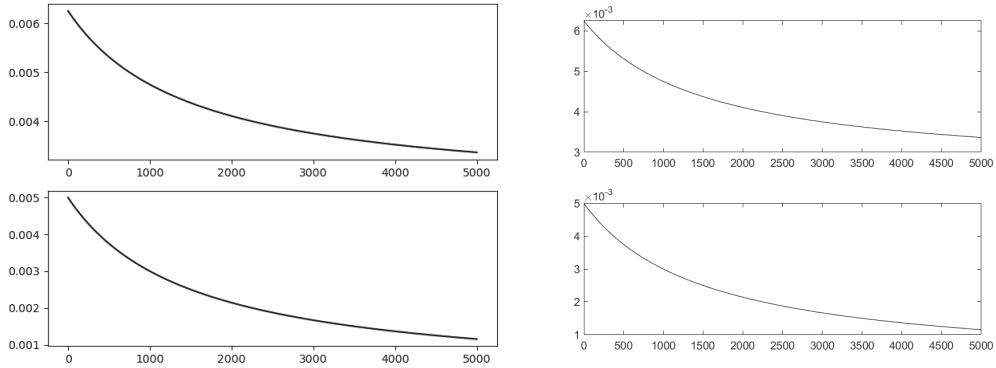


Python and MATLAB versions of graph

0.3.3 Linear Algebra

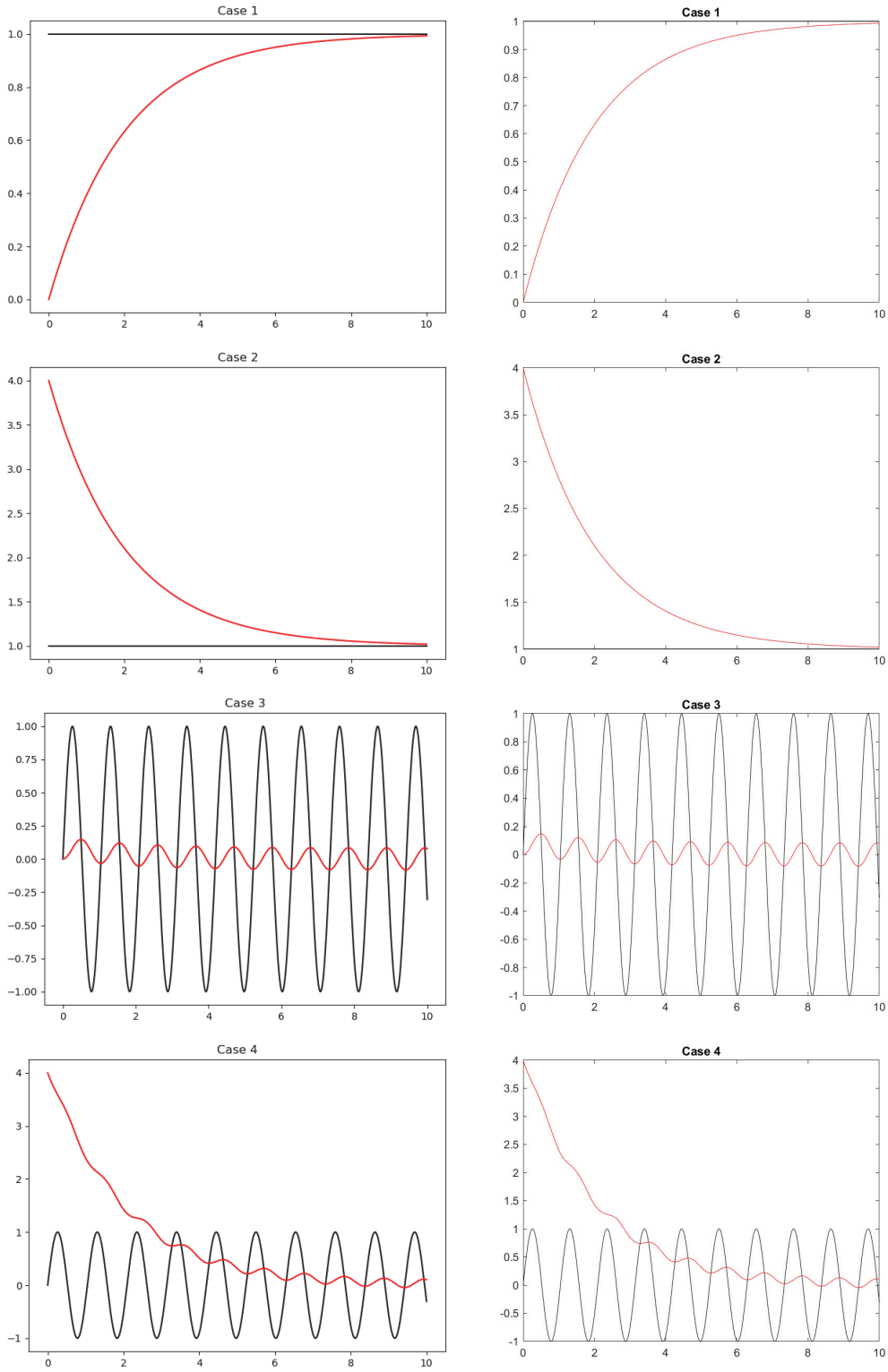
$v_a = 6.25$ $v_b = 5.0$ $v_c = 3.75$

0.3.4 Linear Algebra Sweep



Python and MATLAB versions of graph

0.3.5 ODE



Python and MATLAB versions of graphs