

Homework 5.

Amath 301
Beginning Scientific Computing

© Ryan Creedon, University of Washington

Due: 5/2/23 at 11:59pm to Gradescope

Directions:

For the **written exercises** of the homework, please do the following:

- Write your name, student number, and section at the top of your written exercises. A half point will be deducted if these are not included.
- Complete all written exercises to the best of your ability and as neatly as possible.
- For those who are writing their answers by hand, please scan your work and save it as a pdf file.
- You are encouraged to type your written homework using L^AT_EX. Doing so will earn you a half bonus point. Check out my L^AT_EX beginner document and overleaf.com if you are new to L^AT_EX.

For the **coding exercises** of the homework, please do the following:

- Create a MATLAB script titled `hw5.m` and divide your script into sections according to the various coding exercises.
- Make sure your final answers in each coding exercise are assigned to the correct variable names.
- Publish your script as a pdf file. You can do this by typing `publish('hw5.m','pdf')` in the Command Window.
- Attach your published code to the end of your written exercises. Submit your entire homework assignment (written + coding exercises) to Gradescope.

Pro-Tips:

- You can check many of your answers to the written exercises of this homework by having MATLAB, Wolfram Alpha, or Chat GPT do the calculations for you.
- Teamwork makes the dream work, but please make sure you write up your solutions.
- Don't wait until the last minute. ☺

Written Exercises

(7 points total)

Exercise 1. (Component Skill 5.1)

Solve the following least-squares problem:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}.$$

Show how you arrived at the normal equation. Once there, you can just report the solution of the linear system.

Exercise 2. (Component Skill 5.2)

The following data represents the temperature T (in $^{\circ}\text{C}$) in Seattle t hours after sunrise:

t	T
0	0
1	0
2	2

What is the equation of the line that fits these data points and minimizes the least-squares error? Show how you arrived at the normal equation. Once there, you can just report the solution of the linear system.

Exercise 3. (Component Skill 5.3)

Suppose we sample $y = \sin^2(x)$ at $x = 0$, $x = \pi/4$, and $x = \pi/2$. We obtain the following data:

x	y
0	0
$\pi/4$	$1/2$
$\pi/2$	1

- a. Fit the data points above to the curve

$$y = a + b \cos(2x).$$

Determine a and b that minimize the least-squares error. Show how you arrived at the normal equation. Once there, you can just report the solution of the linear system.

- b. The values of a and b you found in (a) should not surprise you. Give a one sentence explanation for why you should not be surprised.

Coding Exercises

(13 points total)

Exercise 1. (Component Skill 5.3)

The file `salmon_data.mat` contains the annual count of Chinook salmon taken at Bonneville on the Columbia river from the years 1938 to 2020. In this problem, we will fit these data with various polynomial curves to predict the salmon population in 2021.

- a. Download the file `salmon_data.mat` from the Homework 5 page on Canvas. Make sure this file is in the same folder as your `hw5.m` script. In the Exercise 1 section of your `hw5.m` script, type

```
load('salmon_data.mat');
```

This will automatically create two variables. The first variable is called `salmon`. This is an 83×1 column vector of the annual salmon count taken at Bonneville. The second variable is called `year`. This is an 83×1 column vector of the corresponding years for which the salmon count was recorded.

Treating `salmon` as the y -variable and `year` as the x -variable, use the MATLAB built-in function `polyfit` to find the coefficients of the first-degree polynomial that best fits the salmon data in the least-squares sense. Assign the row vector of coefficients of this first-degree polynomial to the variable `A1`. **You may get MATLAB warnings about the polynomial being badly conditioned.** Normally, we'd heed these warnings, but let's ignore them for the purposes of this exercise.

Answer:

`A1 =`

```
1.0e+06 *  
  
0.004188150266572   -7.865418196822168
```

- b. Use the MATLAB built-in function `polyfit` to find the coefficients of the third-degree polynomial that best fits the salmon data in the least-squares sense. Assign the row vector of coefficients of this third-degree polynomial to the variable `A2`.

Answer:

A2 =

```
1.0e+10 *
```

```
Columns 1 through 2
```

```
0.000000000144628 -0.000000843920337
```

```
Columns 3 through 4
```

```
0.001641227241351 -1.063748293304195
```

- c. Use the MATLAB built-in function `polyfit` to find the coefficients of the fifth-degree polynomial that best fits the salmon data in the least-squares sense. Assign the row vector of coefficients of this fifth-degree polynomial to the variable **A3**.

Answer:

A3 =

```
1.0e+14 *
```

```
Columns 1 through 2
```

```
-0.0000000000000000 0.000000000001338
```

```
Columns 3 through 4
```

```
-0.000000005285986 0.000010443108507
```

```
Columns 5 through 6
```

```
-0.010315260022162 4.075371689671109
```

- d. Assuming (a)-(c) are correct, you now have three polynomials that approximate the count of Chinook salmon as a function of the year: a linear polynomial, a cubic polynomial, and a quintic polynomial. Call these polynomials p_1 , p_3 , and p_5 , respectively. Let's test the predictive power of these polynomials.

The number of Chinook salmon in 2021 was 489,523. For each of your three polynomials, calculate the relative error between what your polynomial predicts is the salmon count in 2021 and what the true salmon count is. In other words, calculate

$$\epsilon_1 = \frac{|p_1(2021) - 489523|}{489523}, \quad \epsilon_2 = \frac{|p_3(2021) - 489523|}{489523}, \quad \text{and} \quad \epsilon_3 = \frac{|p_5(2021) - 489523|}{489523}.$$

Use the MATLAB built-in function `polyval` to evaluate your polynomials. Assign the variable **A4** to the 3×1 row vector $[\epsilon_1, \epsilon_2, \epsilon_3]$.

Answer:

A4 =

Columns 1 through 2

0.223300012295425 0.672206101221254

Column 3

0.177800251469287

Exercise 2. (Component Skill 5.3)

The amount of CO_2 in the atmosphere is measured at the Mauna Loa observatory in Hawaii in parts per million (ppm). The file `C02_data.mat` contains the monthly average CO_2 concentrations from March 1958 to December 2021. A plot of the data is shown below. Clearly, there is an overall increasing trend in CO_2 concentrations as well as small seasonal oscillations. In this exercise, we fit the data to an exponential curve to capture the overall increasing trend in CO_2 concentrations.

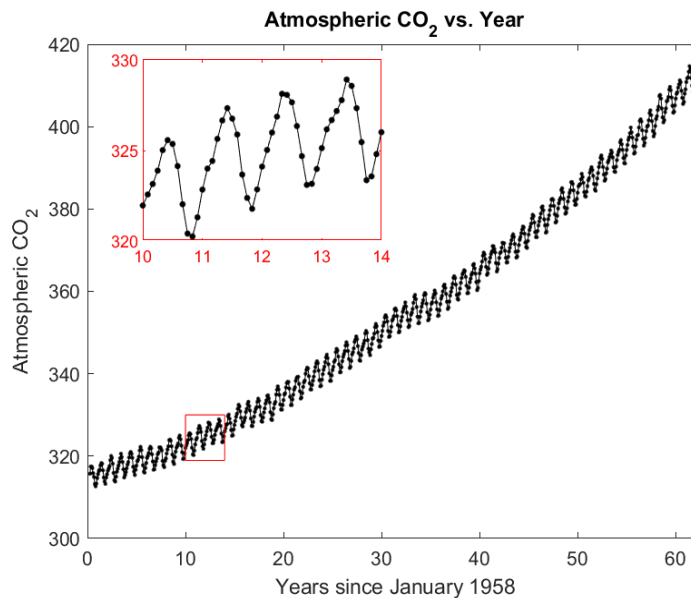


Figure 1: Monthly average concentration of atmospheric CO_2 in ppm as a function of time

- Download the file `C02_data.mat` from the Homework 5 page on Canvas. Make sure this file is in the same folder as your `hw5.m` script. In the Exercise 2 section of your `hw5.m` script, type

```
load('C02_data.mat');
```

This will automatically create two variables. The first variable is called `C02`. This is a 753×1 column vector of monthly averaged CO₂ measurements (in parts per million). The second variable is called `year`. This is a 753×1 column vector of time in years since January 1958. These times correspond to when the CO₂ measurements were taken.

Treating `C02` as the y -variable and `year` as the x -variable, suppose we fit the data with an exponential curve of the form

$$y = ae^{rx} + b.$$

Create a function handle that takes in a 3×1 column vector whose first component is a , second component is r , and third component is b and returns the least-squares error of the exponential fit to the data. Use the MATLAB built-in function `fminsearch` to find the values of a , r , and b that minimize the least-squares error. Use the initial guess `[30;0.03;300]`. Assign the variable `A5` to the column vector `[a;r;b]`.

Answer:

`A5 =`

```
1.0e+02 *  
  
0.564547944545247  
0.000163374706853  
2.569875853592275
```

- b. Repeat (a) but using the L^1 error instead of the least-squares error. Assign the variable `A6` to the column vector `[a;r;b]`.

Answer:

`A6 =`

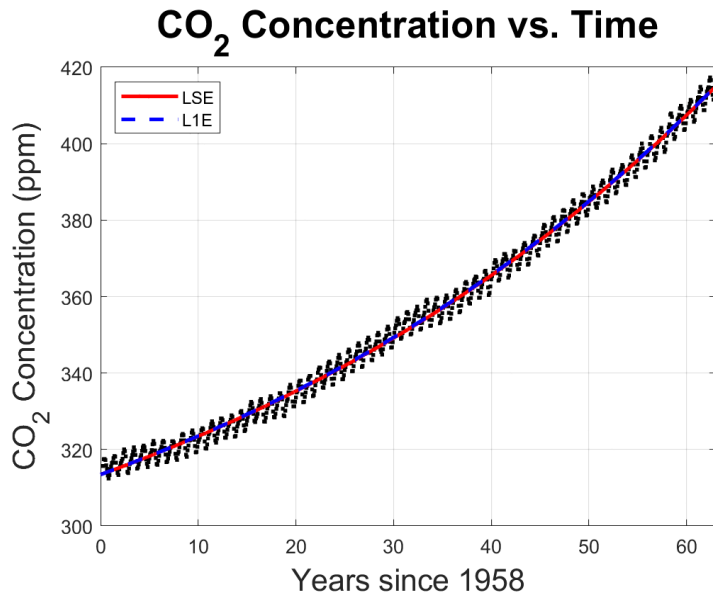
```
1.0e+02 *  
  
0.572803039886350  
0.000161952557068  
2.561796495315599
```

- c. Let's plot our results from (a) and (b). Do the following **in order**:

- Plot the CO₂ data in black. Use `.'` for the `'Marker'` option and `'none'` for the `'LineStyle'` option. Set the domain to $0 \leq x \leq 63$. Set the range to $300 \leq y \leq 420$.
- Superimpose your exponential curve from (a) to the plot of the CO₂ data. Be sure to sample this curve with enough points to appear continuous. Plot this exponential curve in red. Set the `'LineWidth'` option to 2.

- Superimpose your exponential curve from (b) to the plot of the CO₂ data and your exponential curve from (a). Be sure to sample this curve with enough points to appear continuous. Plot this exponential curve in blue. Set the 'LineWidth' option to 2. Set the 'LineStyle' option to '--'.
- Add a legend that distinguishes your two exponential curves. Label your exponential curve from (a) 'LSE' for "least-squares error." Label your exponential curve from (b) 'L1E' for "L¹ error." Move the legend to the northwest corner of the plot.
- Create a x -axis label 'Years since 1958'. Create a y -axis label 'CO₂ Concentration (ppm)'. Set the font size of both labels to 16. Create a title 'CO₂ Concentration vs. Time'. Set the font size of the title to 20.
- Turn on the grid and box.

Answer:



Exercise 3. (Component Skill 5.3)

Hanging from one of Sally's cat towers is a spring with a small toy connected at its end. When the toy is vertically displaced, it oscillates up and down until eventually settling back to equilibrium because of friction and air resistance. Sally is very interested in the behavior of her oscillating toy and has taken careful measurements of its displacement as a function of time. These measurements can be found in the file `spring_data.mat`.

- a. Download the file `spring_data.mat` from the Homework 5 page on Canvas. Make sure his file is in the same folder as your `hw5.m` script. In the Exercise 3 section of your `hw5.m` script, type

```
load('spring_data.mat');
```

This will automatically create two variables. The first variable is called `deltaz`. This is a 51×1 column vector of vertical displacements in centimeters. The second variable is called `time`. This is a 51×1 column vector of time in seconds. These times correspond to the measured displacements.

Sally has a suspicion that her toy's maximum displacement below its equilibrium position happens at time $t = \pi/2$ seconds. She'd like to obtain the value of this displacement to know where to strike during play time. Unfortunately for Sally, she has no measurements of displacement at exactly $t = \pi/2$ seconds. Her closest measurements are at $t = 1.4$ seconds and $t = 1.6$ seconds. Thus, she's going to need to interpolate.

Using the MATLAB built-in function `interp1`, help Sally estimate the value of the toy displacement at $t = \pi/2$ seconds using a cubic spline interpolation. Assign the value of this displacement to the variable `A7`.

Answer:

`A7 =`

```
-0.730388432785160
```

- b. Help Sally create a plot of her data and cubic spline interpolation. Do the following **in order**:
- Plot the cubic spline interpolation of the displacement data in blue. Use the MATLAB built-in function `interp1` with sample points `t = 0:0.01:10` to construct this cubic spline interpolation. Adjust the `'LineWidth'` option to 2. Set the domain to $0 \leq x \leq 10$. Set the range to $-1 \leq y \leq 1$.
 - Superimpose the displacement data to your cubic spline plot. Plot the displacement data in black. Use `','` for the `'Marker'` option and `'none'` for the `'LineStyle'` option. Adjust the `'MarkerSize'` option to 15.
 - Create a x -axis label `'Time (s)'`. Create a y -axis label `'Vertical Displacement (cm)'`. Set the font size of both labels to 16. Create a title `'Vertical Displacement vs. Time'`. Set the font size of the title to 20.

- Turn on the grid and box.

Answer:

