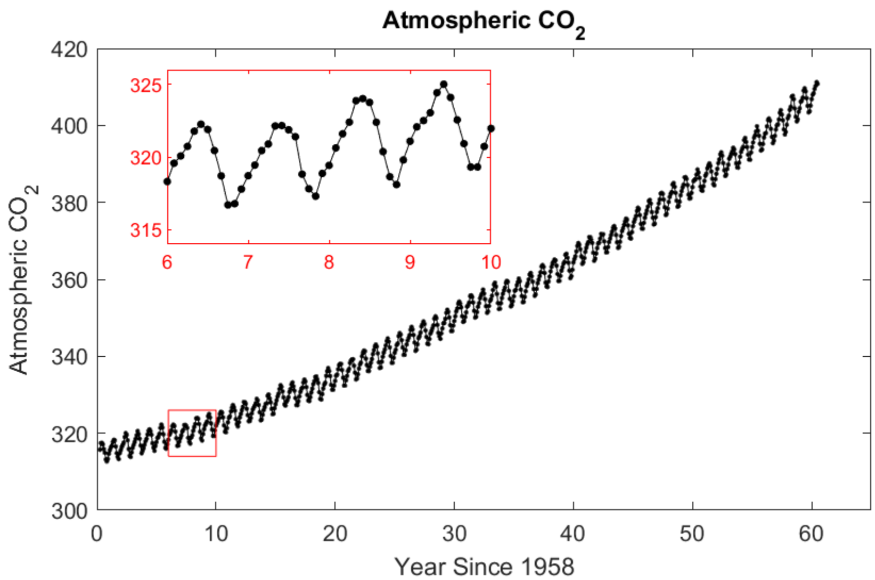
AMATH 301
Homework 5
Due: Friday, February 8, 2019

Atmospheric CO_2 Data

The amount of CO_2 in the atmosphere is regularly measured at the Mauna Loa observatory in Hawaii. Monthly averages since 1958 are publicly available and can be downloaded from the following web address: https://www.esrl.noaa.gov/gmd/ccgg/trends/data.html Shown below is a plot of the data. The data has an overall upward trend as well as seasonal oscillations.



This data has been converted into MATLAB variables and placed in the file `CO2_data.mat`. Download this file from Canvas and put it the same folder that you do your homework in. The file can then be loaded into the MATLAB workspace with the commands

```
clc
clear
load('CO2_data.mat');
```

This will load three variables into MATLAB:

1. `t` is a row vector of times corresponding to the beginning of each month from March 1958 through June 2018. The date is measured in years since 1958, so the first value is `t(1) = 3/12`.
2. `y` is a row vector of the CO_2 data values at each time.
3. `n` is the number of data points, `n = 724`.

This data file has already been uploaded to Scorelator, so you do not need to upload your own copy of the data file. <u>You do need to use the `load` command in order to load the data.</u>

Refresher on Least-squares Fitting

Recall that to fit the function $y = f(t; C_1, C_2, \ldots, C_m)$ to the data points (t_k, y_k) we construct the error function

$$\varepsilon = \frac{1}{2}\sum_{k=1}^{n}\left[f(t_k; C_1, C_2, \ldots, C_m) - y_k\right]^2.$$

We then take derivatives with respect to the parameters $C_1, C_2, \ldots, C_m$ and set them equal to zero. Solving the resulting system of equations yields the least-squares fit.

For a linear fit, $y = f(t; A, B) = At + B$, and the error equation becomes

$$\varepsilon = \frac{1}{2}\sum_{k=1}^{n}\left[(At_k + B) - y_k\right]^2,$$

and so, taking derivatives, we obtain

$$\frac{\partial\varepsilon}{\partial A} = \sum_{k=1}^{n}\left[(At_k + B) - y_k\right](t_k) \qquad\qquad \frac{\partial\varepsilon}{\partial B} = \sum_{k=1}^{n}\left[(At_k + B) - y_k\right].$$

Setting $\frac{\partial\varepsilon}{\partial A} = 0$ and $\frac{\partial\varepsilon}{\partial B} = 0$ we obtain a linear system

$$\begin{cases} \displaystyle\sum_{k=1}^{n}\left[At_k^2 + Bt_k - t_k y_k\right] = 0 \\ \displaystyle\sum_{k=1}^{n}\left[At_k + B - y_k\right] = 0 \end{cases} \quad\Longrightarrow\quad \begin{cases} \displaystyle A\sum_{k=1}^{n} t_k^2 + B\sum_{k=1}^{n} t_k = \sum_{k=1}^{n} t_k y_k \\ \displaystyle A\sum_{k=1}^{n} t_k + B\sum_{k=1}^{n} 1 = \sum_{k=1}^{n} y_k. \end{cases} \tag{1}$$

In matrix form, we have $\mathbf{M}\vec{x} = \vec{b}$.

$$\mathbf{M} = \begin{bmatrix} \sum_{k=1}^{n} t_k^2 & \sum_{k=1}^{n} t_k \\ \sum_{k=1}^{n} t_k & n \end{bmatrix} \qquad \vec{x} = \begin{bmatrix} A \\ B \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} \sum_{k=1}^{n} t_k y_k \\ \sum_{k=1}^{n} y_k \end{bmatrix}.$$

This simple two variable linear system can easily be solved with the backslash command. The hard part is constructing the matrix $\mathbf{M}$ and vector $\vec{b}$. These sums can be constructed in a loop, or alternatively with vector operations, like in the following example:

```
>> t = [1 2 3 4]
t =
     1     2     3     4
>> y = [4.3 3.7 2.8 3.1]
y =
    4.3000    3.7000    2.8000    3.1000
>> t.*y
ans =
    4.3000    7.4000    8.4000   12.4000
>> sum(t.*y)
ans =
   32.5000
```

The key tricks here are that we are using component-wise operations (`.*`) and the `sum` function to sum up the elements of a vector. We first create the terms of the vector (`t.*y`) and then sum them up.

Linear fit

We will begin with a very simple linear fit. The function to fit to the data is $y(t) = At + B$, where A and B are the unknown fit parameters.

Problem 1 (Scorelator).

Learning Goal: Use `sum` and vector operations to compute sums

Use MATLAB to construct the matrix and vector for a linear fit to the CO_2 data. Save $\mathbf{M}$ to the file **A1.dat** and $\vec{b}$ to **A2.dat**. You may find it convenient to name variables `M_fitlinear, b_fitlinear` to avoid confusion with similarly-named variables. As a check, the first entry of $\mathbf{M}$ should be approximately 887611.4.

Next solve for the coefficient vector $\vec{x}$ that contains the parameters A and B of the linear fit. Save the vector as `coeffs_fitlinear`, and output it to the file **A3.dat**.

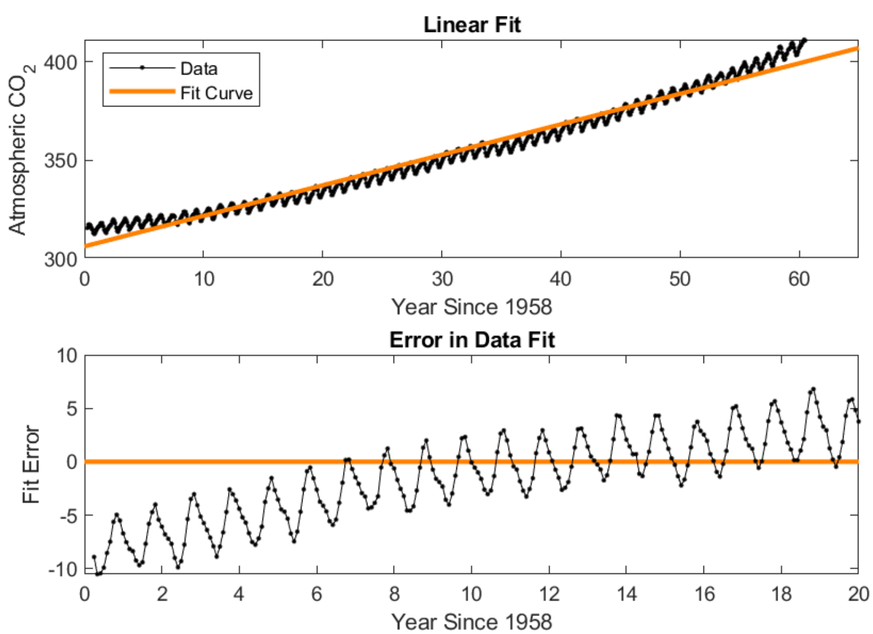
Problem 2 (Writeup).

Learning Goal: Produce a figure showing data, a fit trend, and the error of the fit trend.

Produce a figure with two plots: the first showing the data and the linear fit you obtained from Problem 1, and the second the error in the fit: the difference between the fit prediction value $y(t_k)$ and the data point, y_k, at each time t_k for which we have data. To do this, you can define a function based on the coefficients you found:

```
A = coeffs_fitlinear(1);
B = coeffs_fitlinear(2);
y_fit_linear = @(t) A*t + B;
```

Match the title, axis labels, and legend entries as below. Plot the linear fit from $t = 0$ to $t = 65$ years; the year $t = 65$ corresponds to 2023, so the plot shows a projection into the near future. Plot the fit curve with the custom RGB color `[1 .5 0]` with line width 2. You can look up how to specify a custom RGB (red green blue) color in the `plot` documentation; this particular color is a shade of orange. Print your figure to file as `co2_fit_linear.png`.



Quadratic fit

We will next perform a quadratic fit: $y = f(t) = At^2 + Bt + C$. For this fit function, the error equation becomes

$$\varepsilon = \frac{1}{2}\sum_{k=1}^{n}\left[At_k^2 + Bt_k + C - y_k\right]^2.$$

Taking partial derivatives with respect to A, B, C and setting them equal to zero yields the following system of equations,

$$\begin{cases} \sum_{k=1}^{n}\left[At_k^4 + Bt_k^3 + Ct_k^2 - t_k^2 y_k\right] = 0 \\ \sum_{k=1}^{n}\left[At_k^3 + Bt_k^2 + Ct_k - t_k y_k\right] = 0 \\ \sum_{k=1}^{n}\left[At_k^2 + Bt_k + C - y_k\right] = 0 \end{cases} \tag{2}$$

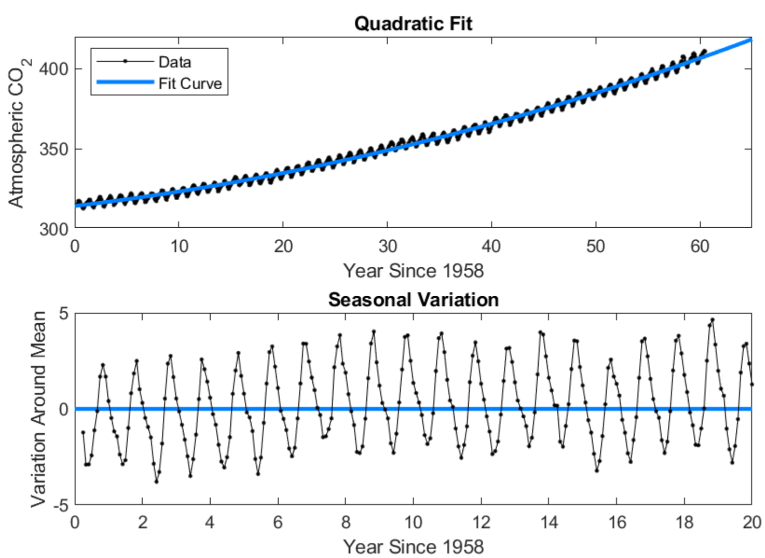
Problem 3 (Scorelator).

Learning Goal: Convert a system of linear equations to matrix-vector form.

Put the system (2) into matrix-vector form and and save the matrix and vector as `M_fitquadratic`, `b_fitquadratic`. Save this matrix and vector to the files **A4.dat** and **A5.dat**, respectively. Solve the system for the coefficient vector `coeffs_fitquadratic` and save this vector to **A6.dat**.

Problem 4 (Writeup).

Produce a figure showing the quadratic fit, similar to the linear fit. Use `subplot` to create upper and lower plots. Plot the data (small black dots with solid line) and the fit function (line width 2, color `[0 .5 1]`, which is a light blue) in the upper axes. In a lower axes, plot the fit error at each data point, along with a light-blue line with line width 2 along the x-axis. Here the quadratic fit captures the overall trend well, but misses the seasonal variation. By subtracting the data values from the prediction values, we can isolate the seasonal variation of the CO2 levels. Set the x-limit to $[0, 20]$ so that the variation over a limited number of years can be more clearly seen. Save your figure as `co2_fit_quad.png`.



Quadratic+Sinusoidal fit

We will next add an oscillatory term to our fit function to account for the seasonal variation of the data,

$$y = f(x) = At^2 + Bt + C + D\sin(2\pi t).$$

The error function is

$$\varepsilon = \frac{1}{2}\sum_{k=1}^{n}\left[At_k^2 + Bt_k + C + D\sin(2\pi t_k) - y_k\right]^2,$$

and the system of equations to solve is

$$\begin{cases} \displaystyle\sum_{k=1}^{n}\left[At_k^4 + Bt_k^3 + Ct_k^2 + D\sin(2\pi t_k)t_k^2 - t_k^2 y_k\right] = 0 \\ \displaystyle\sum_{k=1}^{n}\left[At_k^3 + Bt_k^2 + Ct_k + D\sin(2\pi t_k)t_k - t_k y_k\right] = 0 \\ \displaystyle\sum_{k=1}^{n}\left[At_k^2 + Bt_k + C + D\sin(2\pi t_k) - y_k\right] = 0 \\ \displaystyle\sum_{k=1}^{n}\Big[At_k^2\sin(2\pi t_k) + Bt_k\sin(2\pi t_k) + C\sin(2\pi t_k) + D\sin(2\pi t_k)^2 - y_k\sin(2\pi t_k)\Big] = 0. \end{cases}$$

Remember that A, B, C, D are the variables. $t_k^2, t_k, \sin(2\pi t_k)$ are coefficients of those variables.

Problem 5 (Scorelator).

Learning Goal: Convert a system of linear equations to matrix-vector form.

Put this system into matrix-vector form and and save the matrix and vector as `M_fitquadsinu`, `b_fitquadsinu`. Save this matrix and vector to the files **A7.dat** and **A8.dat**, respectively. Solve the system for the coefficient vector `coeffs_fitquadsinu` and save this vector to **A9.dat**.

Problem 6 (Writeup).

Learning Goal: Judge whether results are correct and reasonable based on visual inference.

Produce a figure like the one in Problem 4 for the quadratic fit, but with the following change. Since the quadratic fit only captured the general trend, we called the error of the fit the 'seasonal variation'. In the quadratic+sinusoidal fit, we have added a term to account for the seasonal variation, and so now the difference between the data points and the fits is simply error. Update the title of the error plot to reflect that this is simply the error, and not 'seasonal variation'.

Use the custom color `[0 .8 0]` (forest green).

Save your figure as `co2_fit_quadsinu.png`.

Exponential fit

The fits we have performed so far have done progressively better jobs of capturing first the general trend, then the shape of the general trend, and lastly some of the finer detail (oscillations due to seasonal variation) in the data. However, there are some problems with these fits (what are they?).

We will next use `fminsearch` to fit general nonlinear models to the data. With these models, the fit parameters cannot be written in terms of a linear system, so we cannot simply solve a linear system to find them. To begin we will fit an exponential function to the data to capture the overall rate of growth,

$$y = e^{A(t-B)} + C.$$

`fminsearch` must be provided with two things: the error function to minimize, and an initial guess for the parameters. In this case, the error function is

$$\varepsilon(A,B,C) = \frac{1}{2}\sum_{k=1}^{n}\left[\left(e^{A(t_k-B)} + C\right) - y_k\right]^2 \tag{3}$$

The parameters of this error function are A, B, C. The data points t_k, y_k are not unknowns; they are known data values. As A, B, C vary, the error varies, and some 'best' values of A, B, C achieve minimal error.

We also need to supply `fminsearch` with an initial guess. The better the initial guess, the more sure you can be that `fminsearch` will perform well.

Problem 7 (Scorelator).

Learning Goal: Write a function to calculate the error of a fit curve to data points.

Write a function to evaluate the fit error. Use `fminsearch` to fit the exponential fit function to the data, with an initial guess for the parameters of $A = 0.03, B = -100, C = 300$. Save the vector of parameters `p_bestfit` produced by `fminsearch` to the file **A10.dat**. Evaluate the fit at `ts = [-58; 0; 62]` to estimate the average CO_2 levels during the years 1900, 1958, and 2020, respectively. Save these values in a column vector and output to file as **A11.dat**.

Problem 8 (Writeup).

Learning Goal: Judge whether results are correct and reasonable based on visual inference.

Produce a figure for the exponential fit like the one you produced for the quadratic fit (without oscillations). Use the color `[.75 0 1]` (indigo) for the fit. Save your figure as `co2_fit_exp.png`.

Exponential+Sinusoidal fit

Finally, we will do a fit to an exponential function plus a sinusoidal term,

$$y = e^{A(t-B)} + C + D\sin(E(t-F)).$$

Problem 9 (Scorelator).

Learning Goal: Perform data fitting with a sophisticated fit function.

Modify your code from the previous problem to do this fit. Good guesses for the new parameters are $A = 0.03, B = -100, C = 300, D = 3, E = 2\pi, F = 0$. Save the vector of parameters output by `fminsearch` to the file **A12.dat**.

Use the fit to estimate the levels of CO_2 in the atmosphere in the year 2020 at the beginning of each month: `t = 2020-1958 + (0:11).'/12;`. Save these values in a column vector and output to file as **A13.dat**.

Problem 10 (Writeup).

Learning Goal: Judge whether results are correct and reasonable based on visual inference.

Produce a figure for the exponential + sinusoidal fit, similar to the quadratic + sinusoidal fit. Use the custom color `[1 0 .25]`. Save your figure as `co2_fit_expsinu.png`.