AM 2814G: Lab 3B

Aidan Sirbu, Katie Brown
asirbu@uwo.ca, kbrow327@uwo.ca

Western University — March 7, 2021

Abstract

The purpose of this lab was to experiment with several applications of the Least Squares Method for data-fitting and obtaining approximate solutions to inconsistent systems. Data taken from the Mauna Loa Observatory of monthly samples of atmospheric CO_2 concentration was fit to four different models using least-squares. Each model was a linear combination of various sine, cosine, linear, and quadratic terms. The RMSE of each was calculated to determine the most accurate model. Each model was then used to predict CO_2 concentrations at various points and compared with the real data. It was found that RMSE does not have a direct correlation on how well models can predict specific points, but gives an overall sense of how well the models fit the data. Furthermore, atmospheric CO_2 data from Alert, NWT, Canada was also fit to the aforementioned model with the lowest RMSE.

Secondly, the matrix equation Ax=b was produced, where A was a 6000 x 3000 matrix with 3000 rows, each repeated once, and b was a 6000 x 1 column vector, both populated with randomly generated integers. The matrix A was augmented with vector b. Due to the linear dependence of the rows in A and the independence of rows in b, application of Gaussian elimination to the augmented matrix yielded a reduced matrix with a pivot column as its augmented column. This indicated that the system is inconsistent. An approximate solution was then found using Matlab's backslash operator on both the original system $Ax = b$ and the normal equation $A^TAx = A^Tb$. Both produced the same solution (to 8-10 decimal places) and the same root-mean-squared errors (19.8423, identical to at least 4 decimal places), which was determined to be due to the fact that Matlab applies the least-square method when the backslash operator is applied.

Introduction and Background

The Least Squares Method is a valuable numerical analytical tool used for data-fitting. The principle is approximately solving an inconsistent system by determining the *closest* solution. Firstly, a system of equations is written in Matrix Form $Ax = b$, then transformed to the normal equation $A^TA\bar{x} = A^Tb$, where $\bar{x}$ is the approximate least squares solution. It is then straightforward to solve for $\bar{x}$ using methods such as Gaussian Elimination. If the system is consistent, $\bar{x}$ will be equal to x, the standard solution. The geometric interpretation of this problem is that *Ax* forms a plane, and if the vector *b* does not lie in that plane, the system is inconsistent. The least squares solution is thus chosen such that it minimizes the Euclidean distance between Ax and b. The perpendicular distance, $b - A\bar{x}$, is known as the residual, and is a vector containing the distances r_m between each corresponding point in b and $A\bar{x}$.

The residual can be expressed in several ways; the 2-norm is the Euclidean length (the norm) of the residual vector, the squared error (SE) is the sum of the squares of each r_m, and the root mean squared error (RMSE) is the square root of the average of each r_m^2. More precisely, these errors are each given by:

$$\|r\|_2 = \sqrt{r_1^2 + ... + r_m^2}$$

$$SE = r_1^2 + ... + r_m^2$$

$$RMSE = \frac{\|r\|_2}{\sqrt{m}} = \sqrt{\frac{SE}{m}} = \sqrt{\frac{r_1^2 + ... + r_m^2}{m}}.$$

The most common application of the Least Squares Method is constructing data fitting models. Given a set of data points, an ideal parameterized model is chosen. This can take many forms, including linear combinations of polynomial and sinusoidal functions, but the unknown coefficients must appear linearly. In some cases, the data can be linearized by applying the natural logarithm so that the coefficients will appear linearly. For some systems, such as periodic functions, it is necessary to use a linear combination of multiple sinusoidal and polynomial terms. Next, the data points are substituted into the model equation, such that each point produces an equation. This system of linear equations is represented in the form $Ax = b$, where x is the unknown parameters, the coefficients of the model. Finally, the least squares solution $\bar{x}$ will be determined though the above method, which provides the coefficient to complete the model.

The ultimate objective is to produce a model that minimizes the sum of the square of the residuals at each data point. This normal equation approach to the method of least squares is especially vulnerable to ill-conditioning as the high condition number, $cond(A^TA) \approx (cond(A))^2$, magnifies preexisting errors in input data [1]. To improve the conditioning of the model, it is often useful to test the addition or removal of terms and determine which produces the smallest residual error.

Analysis of atmospheric CO2 levels using least-squares data fitting models

Task 2 begins with the analysis of data from Mauna Loa Observatory in Hawaii, USA. The data analyzed is a list of 180 numbers which represent the concentration of atmospheric carbon dioxide in parts per million by volume (ppv) measured monthly from the aforementioned observatory between January 1996 and December 2010 [2]. The objective of the analysis was to fit the data to a model function using the method of Least Squares. The models which were expected to be tested were:

$$f_1(t) = c_1 + c_2t + c_3\cos 2\pi t + c_4 \sin 2\pi t \tag{1}$$
$$f_2(t) = c_1 + c_2t + c_3\cos 2\pi t + c_4 \sin 2\pi t + c_5 \cos 4\pi t \tag{2}$$
$$f_3(t) = c_1 + c_2t + c_3\cos 2\pi t + c_4 \sin 2\pi t + c_5 t^2 \tag{3}$$
$$f_4(t) = c_1 + c_2t + c_3\cos 2\pi t + c_4 \sin 2\pi t + c_5 \cos 4\pi t + c_6 t^2 \tag{4}$$

A Matlab file was coded for each model function. It can be observed that models (2) and (3) are simply model (1) with one extra term added after it while model (4) contains both of these extra terms. To remain concise, only the analysis and code which was used to obtain model (4) will be analyzed. However, the results of each shall be discussed afterwards. To begin the least squares fitting for model (4), the normal equations were built. To do this, the code begins by finding matrix A and vector b such that:

$$Ax = b \tag{5}$$

Vector b, written as CO_2 in the code, is a 180×1 column vector. Next, knowing that there are 180 months worth of data, a vector was produced to represent the time parameters. Since the models used are linear combinations of sine and cosine functions, the months were converted into fractions of years as to represent the yearly period of atmospheric CO_2 trends

```
month = [];
for i=1:180
   month = [month; i/12];
end
```

From a perspective of time complexity, this loop is not rigorously engineered since every iteration changes the size of the *month* vector. However, due to the fact that only 180 values were being added to the matrix, the inefficient time complexity of this can be ignored. Next, the coefficient matrix A was defined as:

```
A = zeros(180,6);
for i=1:180
```

```
    A(i,1) = 1;
    A(i,2) = month(i,:);
    A(i,3) = cos(2*pi*month(i,:));
    A(i,4) = sin(2*pi*month(i,:));
    A(i,5) = cos(4*pi*month(i,:));
    A(i,6) = month(i,:).^2;
end
```

The *for* loop shown above builds each column of the coefficient matrix separately. The first column contains all 1's as the first coefficient in model (4), c_1, is multiplied by coefficient 1. The second column will contain the column vector *month*, while the 3rd, 4th, 5th, and 6th columns containing the various terms in model (4) evaluated at the points contained by the *month* vector. For models (1-3), the code shown above was slightly modified by removing certain lines within the *for* loop as each line represents one of the terms in the model. The rest of the code shown from this point is identical for each model used. The normal equations for the system was then obtained as:

```
ATA = transpose(A)*A;
ATb = transpose(A)*CO2;
c = ATA\ATb;
```

where the column vector *c* is the approximate least squares solution to the inconsistent system. Models (1-4), along with their respective RMSE values, predictions for various months, and the absolute error concerning each models predictions for various months can be found in Table 1 below. As one can observe

Table 1: Models 1-4 obtained using method of least squares with respective RMSE values and predictions of CO_2 concentration at various dates with respective errors

Model	**RMSE**	**Model Predictions of CO_2 (ppv) / Absolute Model Error (ppv)**			
		May '04	Sept '04	May '05	Sept '05
$f_1(t) = 360.9977 + 1.9507t - 1.6669\cos(2\pi t) + 2.4359\sin(2\pi t)$	0.8015	380.07 / 0.55	275.03 / 1.57	382.03 / 0.42	377.85 / 0.85
$f_2(t) = 361.0121 + 1.9488t - 1.6668\cos(2\pi t) + 2.4353\sin(2\pi t) + 0.8595\cos(4\pi t)$	0.5225	380.51 / 0.12	374.77 / 0.71	382.45 / 0.01	376.72 / 0.01
$f_3(t) = 361.5409 + 1.7358t - 1.6683\cos(2\pi t) + 2.4355\sin(2\pi t) + 0.0142t^2$	0.7650	379.82 / 0.81	375.38 / 1.32	381.81 / 0.64	377.38 / 0.65
$f_4(t) = 361.5541 + 1.7344t - 1.6682\cos(2\pi t) + 2.4349\sin(2\pi t) + 0.8593\cos(4\pi t) + 0.0142t^2$	0.4649	380.25 / 0.38	374.52 / 0.47	382.24 / 0.21	376.52 / 0.21

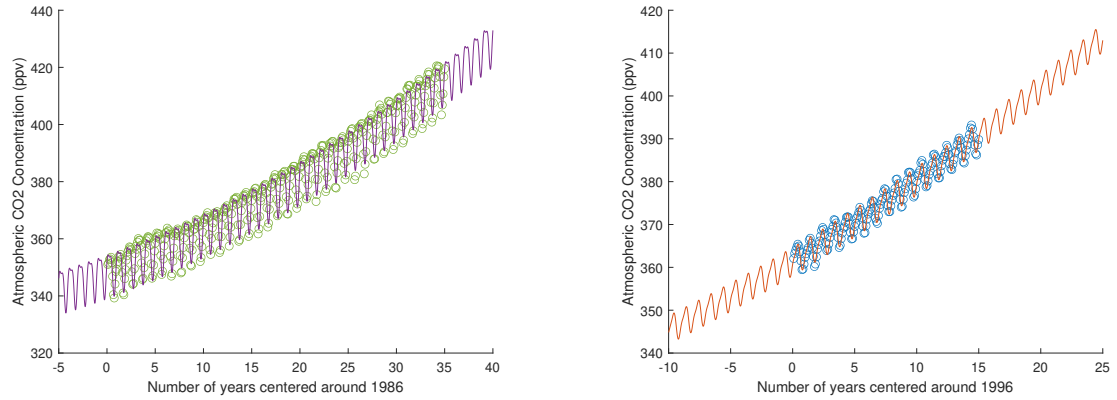
from the RMSE values of models (2) and (3), adding the $\cos(4\pi t)$ term decreases the error more than the t^2 term. This is also evident when the analysis is focused on the predicting power of models (2) and (3) of CO_2 levels for May and September in 2004 and 2005. By comparing RMSE values, it is evident that model (2) predicts the CO_2 concentration with greater accuracy than model (3). Furthermore, model (4) has the lowest RMSE value and therefore fits the data very well. Upon further analysis of model (4), certain behaviours of CO_2 concentration can be deduced. First, its constant term 361.5541 is merely the y-intercept and since the data is centered around 1996, its y-intercept is its prediction of CO_2 levels in that year. Furthermore, the CO_2 levels follow yearly sine trends due to the fact that the function is a linear combination of two cosine functions and one sine function. The second term $1.7344t$ shows that the CO_2 levels have been rising linearly from 1996 to 2010 with a rate approximately equal to 1.7344 ppv per year. The term $0.0142t^2$ implies some sort of parabolic trend to the data. However, the coefficient also indicates that while there may be some parabolic trend, it is compressed to the point in which the graph will not display it within the time interval in which the analysis occurs.

To continue, the same, but less rigorous analysis was performed on atmospheric CO_2 taken from a station in Alert, NWT, Canada from January 1986 to December 2020 [2]. The code used to perform a least-squares model fit will be omitted as it is identical to that shown in Listing 1 in the code appendix

with the only exception being the use of 420 values of CO_2 which is accounted for in the various *for* loops. The model in which the data was fit to was model (4) from the previous task $f(t) = c_1 + c_2t + c_3 \cos 2\pi t + c_4 \sin 2\pi t + c_5 \cos 4\pi t + c_6 t^2$. This model was specifically chosen since upon analysis of the previous task, it was found to fit the data from the Mauna Loa station very well. In addition to this, upon fitting Alert's data to models (1-4) used on Mauna Loa, it was found that model (4) produced the smallest RMSE. The resulting model fit with least squares was:

$$y = 348.6987 + 1.2389t + 0.6070 \cos(2\pi t) + 7.1601 \sin(2\pi t) + 2.5665 \cos(4\pi t) + 0.0197t^2 \quad (6)$$

The RMSE for this model was 1.6655. While this RMSE is clearly larger than the RMSE for model (4) for the Mauna Loa station, it is still small enough to produce an arguably good fit. The graphs comparing the data from the Moa Launa station with model (4) and that of the data from the Alert station are shown below in Figure 1.



(a) Alert Station, model: $y = 348.6987 + 1.2389t + 0.6070 \cos(2\pi t) + 7.1601 \sin(2\pi t) + 2.5665 \cos(4\pi t) + 0.0197t^2$

(b) Mauna Loa Station, model: $y = 361.5541 + 1.7344t - 1.6682 \cos(2\pi t) + 2.4349 \sin(2\pi t) + 0.8593 \cos(4\pi t) + 0.0142t^2$

Figure 1: Comparison of atmospheric CO_2 levels in Alert vs Mauna Loa

Due to the nature of the data obtained from each station, Figure 1(a) is centered around the year 1986 while Figure 1(b) is centered around 1996. Furthermore, the data obtained from the Alert station stretches to 2020 while the Mauna Loa station ends at 2010. Looking at the model fit to the Mauna Loa data, it can be seen that by 2020 (represented by 25 on the x-axis of Figure 1(b) and 35 on the x-axis of Figure 1(a)) it predicts that atmospheric CO_2 concentration should be 415ppv. The Alert station recorded 415ppv of CO_2 concentration in 2020. This suggests that the model fit to the Mauna Loa station is well suited.

Comparing least squares and backslash operator methods

The objective of the next task was to experiment with the Least Squares Method in Matlab. Firstly, a 6000 x 3000 matrix A was constructed. The first half of the rows were filled with randomly generated integers, and the second half repeated the first. A column vector b was also generated with 6000 random integer elements.

```
randomMatrix = randi(100,3000,3000);
A = [randomMatrix; randomMatrix];
b = randi(100,6000,1);
```

This will generate the same matrices A and b every time the code is run since Matlab's random number generator is initialized to the same state each time the application is used [4].

Consider the augmented matrix [A|b] representing the system Ax=b. When performing Gaussian Elimination (GE), two linearly dependent rows in A will produce a zero row. If the corresponding row in

the augmented matrix is not also linearly dependent (as the b values do not scale by the same factor), the result will be a zero row that is set to equal a non-zero number; the system is inconsistent. Each of the rows in A are repeated once, so half of the rows will equal zero once GE has been performed. However, each of the 6000 elements in b were randomly generated, so GE will likely not cause any of these values to equal zero. It will thus produce 3000 equations resembling $0x_1 + ... + 0x_{3000} = b_n$, $b_n \in \mathbb{R}$. This evidently has no solutions; the system is inconsistent.

Next, the system was solved using both the backslash operator, and the Least Squares Method:

```
x = A\b;
ATA = transpose(A)*A;
ATb = transpose(A)*b;
c = ATA\ATb;
```

where x and c are the approximate solutions produced by applying Matlab's backslash operator on the systems $Ax = b$ and $A^TAx = A^Tb$, respectively. Interestingly, the solutions were close to equal, with differences in elements in the range of 10^{-8} to 10^{-10}. This similarity is because Matlab actually performs a least-square fit to solve all non-square matrix equations when the backslash operator is applied [3]. The slight differences between the two solutions are most likely due to the backslash operator using QR factorization method to perform a least-square fit rather than using the normal equations [3].

Finally the RMSE of the regular and Least Squares solutions were calculated by taking the residual at each point and applying the RMSE formula:

```
% RMSE from regular solution
rR = b - A*x;
sR = 0;
for i=1:6000
    sR = sR + rR(i).^2;
end
RMSEreg = (sR/6000)^(0.5);

% RMSE from Least Squares solution
rLS = b - A*c;
sLS = 0;
for i=1:6000
    sLS = sLS + rLS(i).^2;
end
RMSEls = (sLS/6000)^(0.5);
```

Unsurprisingly, as both methods produce the same solution up to approximately 8 decimal places, their RMSE's, accurate to only 4 decimal places are obtained to be equal. Since the generated matrices are the same for each execution, the errors remained constant at 19.8423.

Another interesting phenomena was that when the input range for randomly generated integers was increased or decreased by a factor of 10, the RMSE's changed accordingly. For example, when the *randi* method range was increased from 100 to 1000, the RMSE values increased to 198. It was determined that this was because each element of the generated matrix was simply increased by a factor of 10 as well (ie. $a_{1,1}$ changed from 85 to 850). It then follows that the RMSE values should be expected to increase by a factor of ten:

$$RMSE = \sqrt{\frac{(10 \cdot r_1)^2 + ... + (10 \cdot r_m)^2}{m}} = \sqrt{\frac{100(r_1^2 + ... + r_m^2)}{m}} = 10 \cdot RMSE_0.$$

Summary of Results

The lab began by analyzing the atmospheric CO_2 levels using least-squares data fitting models. The data set of 180 sample points of monthly atmospheric CO_2 were obtained and the trend was fit to four different models. Each model contained a linear combination of the terms $\{1, t, \cos(2\pi t), \sin(2\pi t)\}$. The difference between the models lied in either maintaining the above linear combination or adding one or both of the terms $\cos(4\pi t)$ or t^2. The least-squares fit was performed using normal equations fitting the data to the aforementioned models. Through analysis of the various RMSE's of the models it was found that the least accurate model was model (1) with $RMSE = 0.8015$. This was as expected since it was the simplest model, merely a linear combination of $\{1, t, \cos(2\pi t), \sin(2\pi t)\}$ with no extra terms added. Furthermore, the most accurate model was model (4) with $RMSE = 0.4649$ which was again unsurprising as it contains the most terms, each of which accounting for various quirks in the trend such as a possible parabolic trend that would otherwise be non-apparent by simply looking at the graph over the given time interval.

Despite of model (4)'s lowest RMSE, it was model (2) which best predicted the CO_2 levels in May and September of 2004 and 2005. This is however most likely due to coincidence as the points taken from those dates better fit on model (2) while the rest of the points were still, on average, a better fit to model (4) than to model (2). Furthermore, upon performing the same analysis to atmospheric CO_2 data obtained from Alert, NWT, Canada, the model of best fit was found to be (4) which gave $RMSE = 1.6655$

The second section of this lab involved experimenting with the Least Squares Method in Matlab and comparing it to the backslash operator method. A 6000 x 3000 matrix A was generated in which the second half of the rows were copied from the first half, which were filled with randomly generated integers. A column vector b was produced with 6000 independent random integers, and the equation Ax=b was produced. As each pair of rows in A were linearly dependent yet the corresponding rows in b were not, the system was shown to be inconsistent.

The system was then solved approximately using Matlab's backslash operator on both the original equation $Ax = b$ and the normal equation $A^TAx = A^Tb$. It was found that both methods produced approximately the same solution, as Matlab actually applies a least-square fit to solve all non-square matrix equations when the backslash operator is applied. As the approximate solutions were equal to within 8-10 decimal places, the root-mean-squared errors (which were rounded to 4 decimal places) were identical as well. When the matrices generated with random integers between 0 and 100, the RMSE's were both equal to 19.8423.

References

[1] Sauer, T. (2019). Numerical analysis (3rd ed., p. 23). Hoboken, NJ: Pearson.

[2] C. D. Keeling, S. C. Piper, R. B. Bacastow, M. Wahlen, T. P. Whorf, M. Heimann, and H. A. Meijer, Exchanges of atmospheric CO2 and 13CO2 with the terrestrial biosphere and oceans from 1978 to 2000. I. Global aspects, SIO Reference Series, No. 01-06, Scripps Institution of Oceanography, San Diego, 88 pages, 2001.

[3] Mrdivide, /. (n.d.). Retrieved March 07, 2021, from https://www.mathworks.com/help/matlab/ref/mrdivide.html

[4] Why does my Compiled rand function give the same values every time I run my MATLAB-generated Standalone application? (n.d.). Retrieved March 08, 2021, from https://www.mathworks.com/matlabcentral/answers/104306-why-does-my-compiled-rand-function-give-the-same-values-every-time-i-run-my-matlab-generated-standal

Code Appendix

```
% Model used for least squares:
% f(t)=c1+c2*t+c3*cos(2pi*t)+c4*sin(2pi*t)+c5*cos(4pi*t)+c6*t^2

% month represents a column matrix with number of months from
% Jan 1996 to Dec 2010
month = [];
for i=1:180
   month = [month; i/12];
end

% CO2Matrix represents a column matrix with the CO2 levels in ppv by month from Jan 1996 to Dec
    2010
CO2=[362.05; ... 389.83];

% Define coefficient matrix
A = zeros(180,6);
for i=1:180
    A(i,1) = 1;
    A(i,2) = month(i,:);
    A(i,3) = cos(2*pi*month(i,:));
    A(i,4) = sin(2*pi*month(i,:));
    A(i,5) = cos(4*pi*month(i,:));
    A(i,6) = month(i,:).^2;
end

% Normal equations
ATA = transpose(A)*A;
ATb = transpose(A)*CO2;
c = ATA\ATb;

% RMSE
r = CO2-A*c;
SE = 0;
for i=1:180
    SE = SE + r(i)^2;
end
RMSE = (SE/180)^(0.5);

% Plotting
scatter(month, CO2);
hold on
leastSquaresFunc =
    @(t)361.5541+1.7344*t-1.6682*cos(2*pi*t)+2.4349*sin(2*pi*t)+0.8593*cos(4*pi*t)+0.0142*t.^2;
fplot(leastSquaresFunc);
xlim([-10 25]);
```

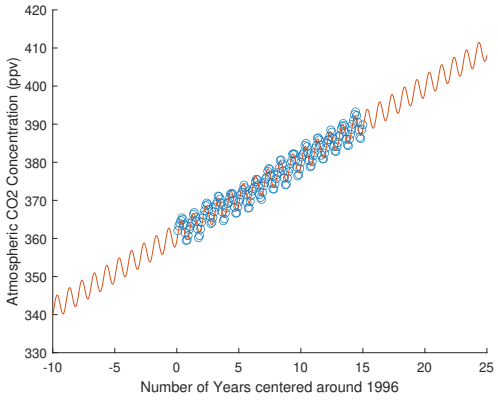
Listing 1: Method of Least Squares analysis to obtain Model 4 for atmospheric CO2 concentration in Mauna Loa, Hawaii. The CO2 vector has been shortened for the sake of being concise, in reality it is a 180×1 column vector. This same code was used with slight alterations to the second *for* loop to produce models 1-3.

```
% Create 3000x3000 matrix of random numbers
rng('default')
randomMatrix = randi(100,3000,3000);

% Concatenate randomMatrix for 6000x3000 matrix A where first 3000 rows are
% random and second 3000 rows repeat the first 3000
A = [randomMatrix; randomMatrix];
```

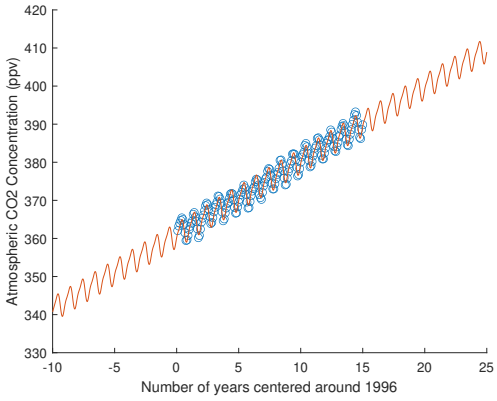
```
% Create 6000x1 random vector b
b = randi(100,6000,1);

% Solve system
x = A\b;

% Normal equations
ATA = transpose(A)*A;
ATb = transpose(A)*b;
% Least square solution vector c
c = ATA\ATb;

% RMSE from regular solution
rR = b - A*x;
sR = 0;
for i=1:6000
    sR = sR + rR(i).^2;
end
RMSEreg = (sR/6000)^(0.5);

% RMSE from Least Squares solution
rLS = b - A*c;
sLS = 0;
for i=1:6000
    sLS = sLS + rLS(i).^2;
end
RMSEls = (sLS/6000)^(0.5);
```

Listing 2: Exploring least-squares solving in Matlab using a 6000×3000 matrix

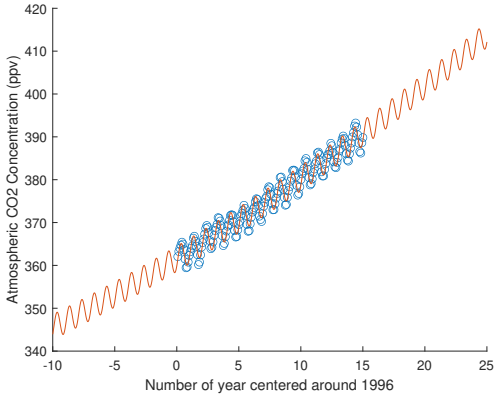
Plot Appendix



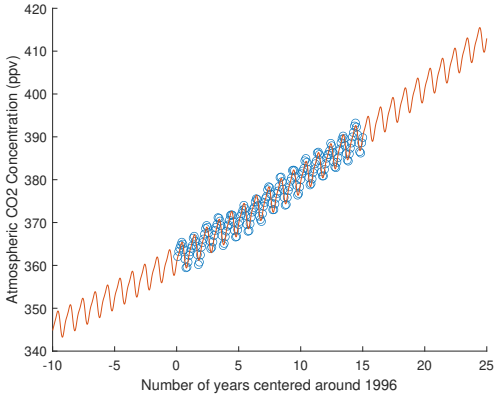
(a) Model 1: $y = 360.9977 + 1.9507t - 1.6669\cos(2\pi t) + 2.4359\sin(2\pi t)$



(b) Model 2: $y = 361.0121 + 1.9488t - 1.6668\cos(2\pi t) + 2.4353\sin(2\pi t) + 0.8595\cos(4\pi t)$



(c) Model 3: $y = 361.5409 + 1.7358t - 1.6683\cos(2\pi t) + 2.4355\sin(2\pi t) + 0.0142t^2$



(d) Model 4: $y = 361.5541 + 1.7344t - 1.6682\cos(2\pi t) + 2.4349\sin(2\pi t) + 0.8593\cos(4\pi t) + 0.0142t^2$

Figure 2: Models (1-4) of a curve of best fit for Atmospheric CO2 between the years of 1986 and 2021 measured from Mauna Loa Observatory, Hawaii. The blue points represent the actual monthly data while the orange curves represent the model functions.

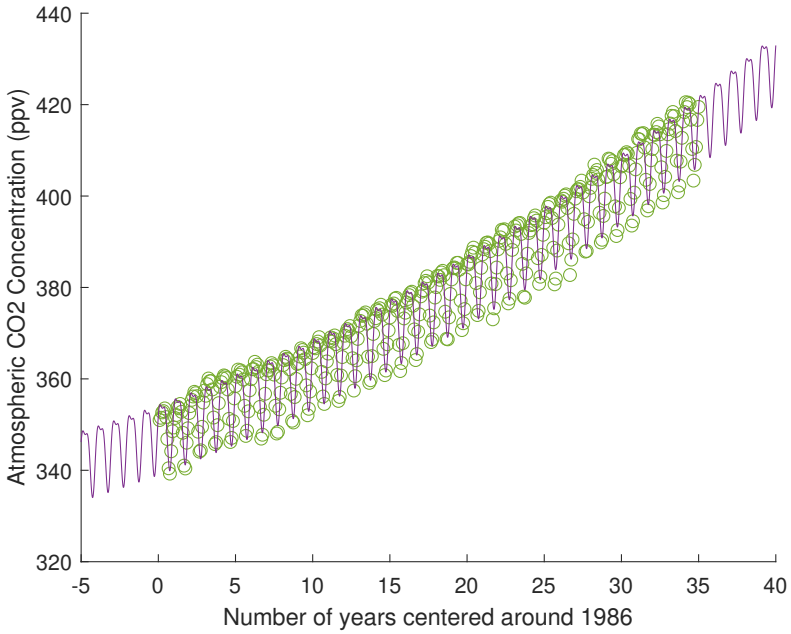


Figure 3: Curve of best fit for atmospheric CO2 between the years of 1981 and 2021 measured from Alert, NWT, Canada. The green points represent the actual monthly data while the purple curve represents the model function $y = 348.6987 + 1.2389t + 0.6070\cos(2\pi t) + 7.1601\sin(2\pi t) + 2.5665\cos(4\pi t) + 0.0197t^2$