Final Project - World Population
Nicolas Guerra

Explanation of Problem

In this report, I will be analyzing the world population over time. My goal is to find a fit to represent the data using the tools we have learned in this class. More specifically, I want to find a fit that is able to both interpolate and extrapolate the data. To find the best fit, I will experiment with different curves such as exponential, power-law, and linear. The data I will be using for this project comes from the following site:

https://www.worldometers.info/world-population/world-population-by-year/

Note that I will only be using the columns labeled "Year" and "World Population" which I simply copied and pasted into a CSV file.

Primary Matrix Decomposition

The primary matrix decomposition that is used in this project is QR decomposition. Since the data is not too big, speed was not the main priority of my algorithm but rather its stability was. As a result, I was in between QR and SVD. In the end, I chose QR decomposition because it is quite stable with the condition of the problem being ~K(A) (compared to Cholesky being ~K(A)^{2}) and it is a bit faster than SVD (although they are still both *$O(n^3)$*).

Finding Best Fit of World Population

Before fitting an exponential curve to the world population, I first linearized the problem in order to use the tools we learned in class.

$$\begin{aligned} population &= a * e^{b*year} \\ log(population) &= log(a * e^{b*year}) \\ log(population) &= log(a) + log(e^{b*year}) \\ log(population) &= log(a) + b * year \end{aligned}$$

The only unknowns in the final equation above is *a* and *b*. I will let *pop_tilda = log(population)* and *a_tilda = log(a)*, and then solve for *a_tilda* and *b*. To get *a*, just take the exponential of *a_tilda.* We now have:

$$pop\_tilda = a\_tilda + b * year$$

Putting this into vector form for each data point, one gets

$$V * c = pop\_tilda$$

where $c = [a\_tilda; b]$ and $V = [1 \quad year\_1; \ldots ; 1 \quad year\_n]$

We will find the least-squares solution to this using QR decomposition and back-substitution.

$$[Q, R] = qr(V)$$

$$c = backsolve(R, Q'*population)$$

After finding c, we can plot our data with the fitted exponential curve.

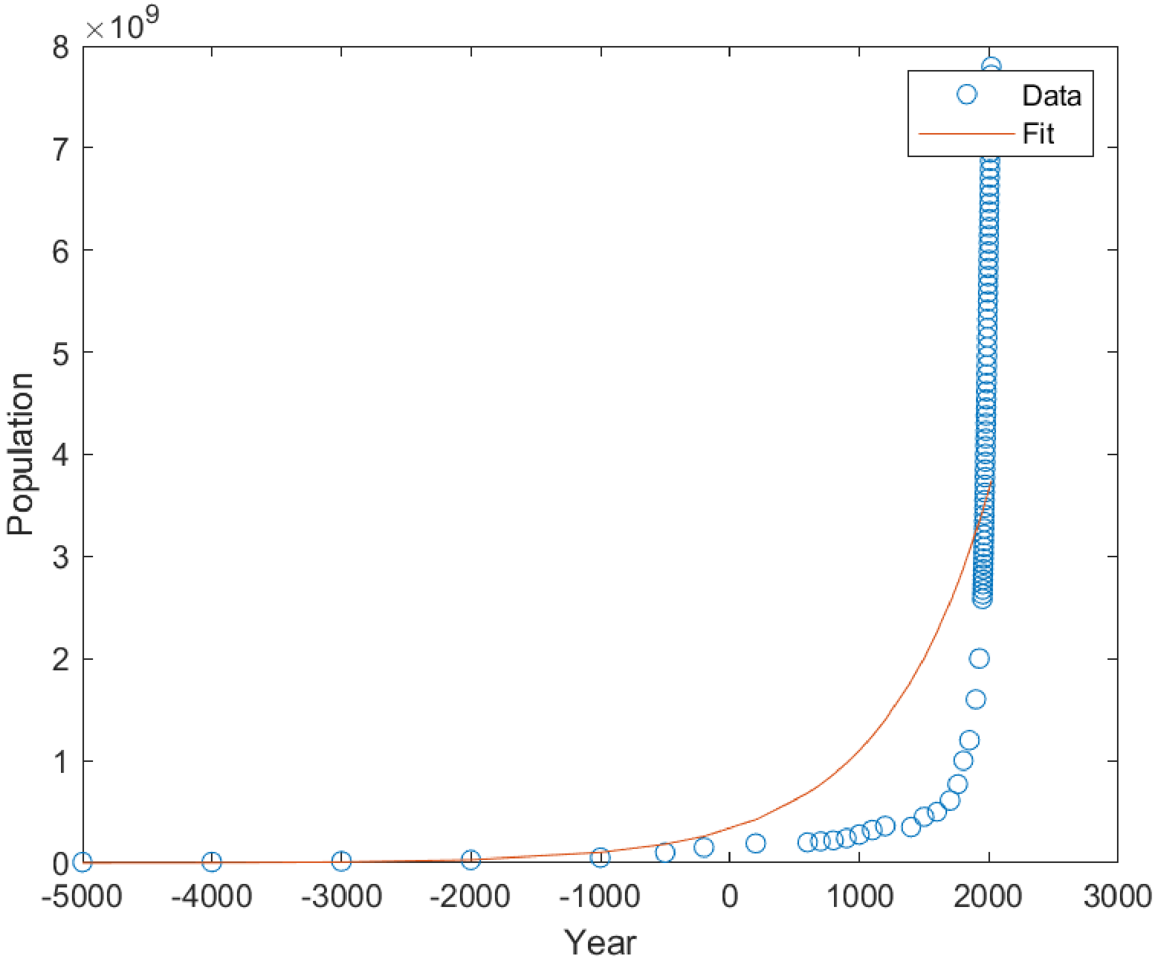

Figure 1: Exponential Fit with QR Decomposition (a = 3.3093e+08, b = 0.0012)

This fit does not look too good. Not being entirely satisfied with Figure 1, I now want to fit it to the power law *population* = $a*year^b$. To linearize the power law, it is very similar to the linearized version of the exponential curve.

$$population = a * year^b$$
$$log(population) = log(a * year^b)$$
$$log(population) = log(a) + b * log(year)$$

Using QR decomposition and back substitution, we can solve the least-squares problem again to get *a* and *b*. However, we need to first rescale both the year and the population data so that they are between (0, 1] before doing the linearizing process above. For the x-axis, the reason why we should have a lower bound of 0 is because we can't take the log of 0 and negative numbers. The reason why we should have an upper bound of 1 is because *a* from the power law would have to be extremely small in order for our fit to be suitable for the years 2000s. Having a max of 1, makes it computationally possible to find a more reasonable *a* value. The reason I am scaling the y-axis as well is so that both axes are in the same orders of magnitude. Doing this and finding the least square solution gets us the following graph.

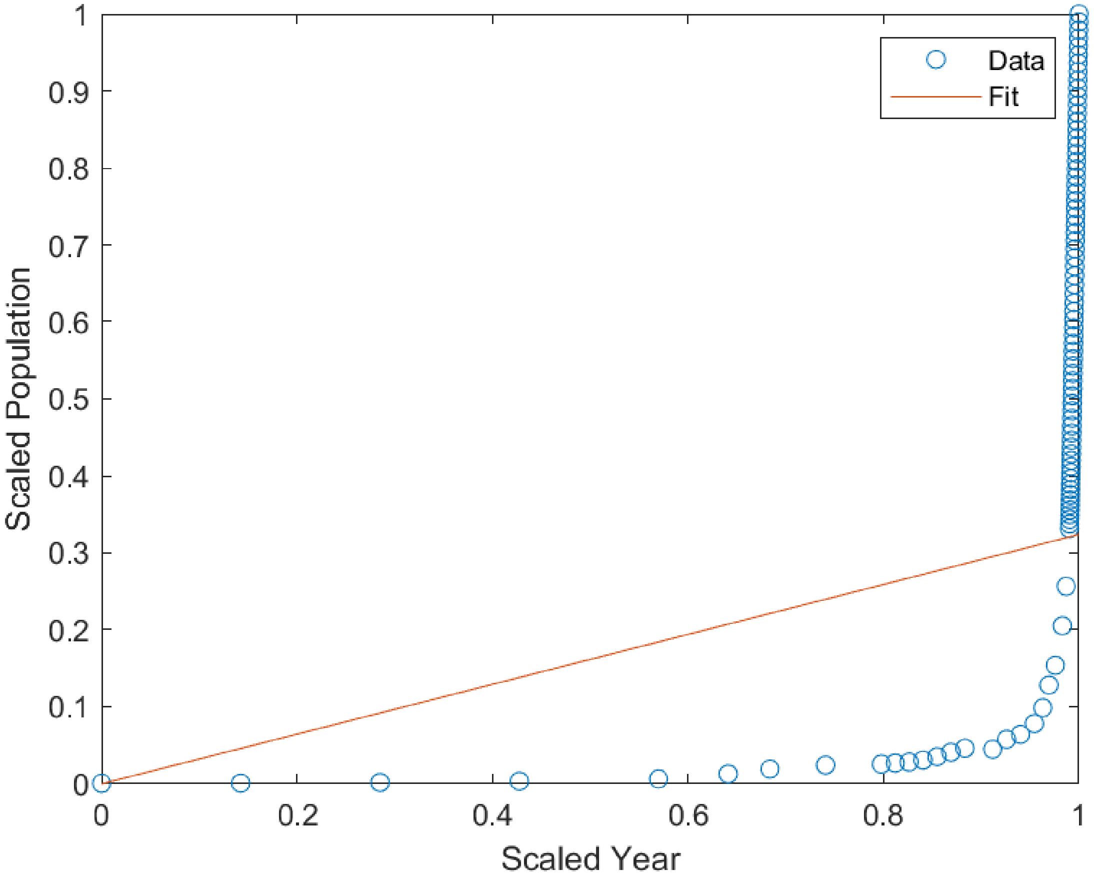

Figure 2: Scaled and Shifter Power Law Fit with QR Decomposition (a = 0.3238, b = 1.0025)

This also does not look too good. Once again, I am not satisfied with the results, so I resort to finding a linear fit after 1950 and an exponential fit before 1950. The reason I specifically say 1950 is because the website I took the data from says “From 1950 to current year: elaboration of data by United Nations, Department of Economic and Social Affairs, Population Division.” Basically, the data past 1950 comes from the same data source, and by eye, it looks linear. The “extraneous” data that didn’t come from the UN before 1950 seems to look exponential.

Using again QR decomposition and back substitution to find *m* and *b* from

$$population\_after\_1950 = m*year\_after\_1950 + b$$

and *a* and *d* from

$$population\_before\_1950 \ = a*e^{d*year\_before\_1950}$$

one gets the following plot.

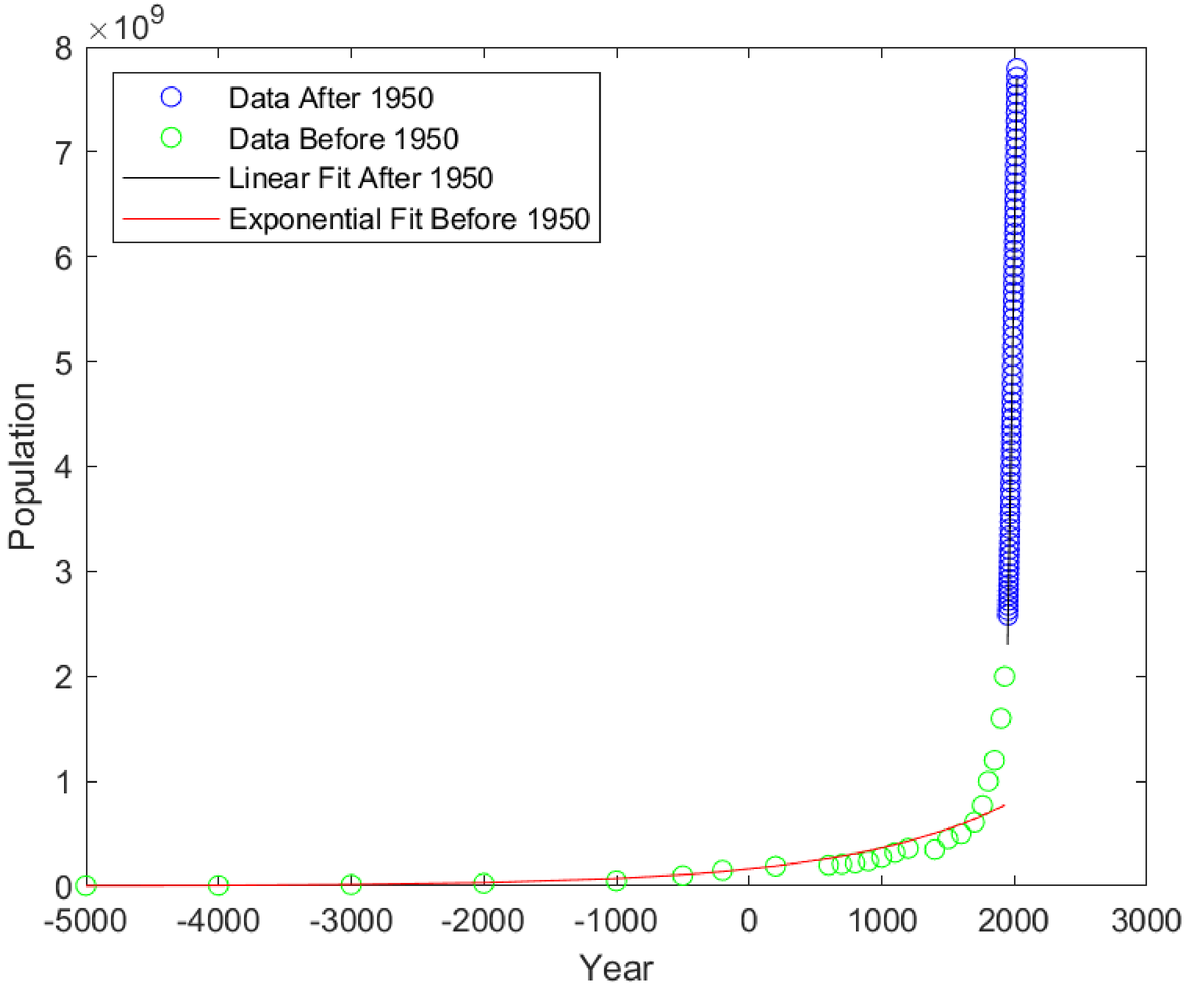

Figure 3: Exponential and Linear Fit with QR Decomposition (m = 7.8243e+07, b = -1.5035e+11; a = 1.6291e+08, d = 8.0707e-04)

Although this fit is not very sophisticated, it uses topics we have learned in class, looks better than Figures 1 and 2, and in the end, gets the job of representing the data done. Looking only at dates after 1950, if we left out the last three datapoints when making the linear fit, how well would the fit be able to predict the last three points? Below is a table that shows how well the linear fit does.

Table 1: Prediction of Last Three Datapoints

Year	**Actual Population**	**Predicted Population**	**Residual (Actual-Predicted)**
2018	7.6311e+09	7.5281e+09	1.0304e+08
2019	7.7135e+09	7.6059e+09	1.0755e+08
2020	7.7948e+09	7.6838e+09	1.1102e+08

Considering the estimated and actual population values are on the same (quite large) orders of magnitude, I find the residuals not to be too bad when it comes to finding an approximate value of the population for the future. Seeing that Figure 3 provides the best representation of world population for extrapolation and interpolation, the following are the equations for the fitted curves.

(1) Before 1950:

$$population = (1.6291e+08) * e^{(8.0707e\text{-}04) * year}$$

(2) After 1950:

$$population = (7.8243e+07) * year - (1.5035e+11)$$

Discussion of Results

Using only what we have learned in class, the exponential fit of the population using the linear least squares approach does not represent the world population well as one can see in Figure 1. It seems that perhaps the smaller population values influence the linear regression more than the larger population values, which would require a more sophisticated approach. Looking at the power-law fit in Figure 2, this also does not look like it represents world population well since the world population does not grow linearly as it shows. For both the exponential and power-law fit, the fits that are being found using the linear least squares approach are not very good qualitatively speaking. It seems the best fit, although not the prettiest, is Figure 3. The combination of exponential and linear fit seem to represent population well. However, the one gap in between the two fits is a big drawback from this approach.

This project provides insight on the world population because now one is able to interpolate what the population is for any given year, of course, keeping in mind that there is a grey area immediately before the year 1950. If you would like to interpolate the population before 1950, use the exponential fit found in Figure 3 and equation (1). If you would like to interpolate the population after 1950, use the linear fit found in Figure 3 and equation (2). Another aspect this project provides insight on is extrapolation. One can can extrapolate what the population will be like in the future using the linear fit found in Figure 3 and equation (2). Looking at Table 1, the error does not turn out to be too bad and can give the interested party a good approximation of what the population will be like in the future.

Using only the tools from this class, I was able to linearize both exponential and power law fits, find the least squares solution to linear, power law, and exponential fits, reason why a QR decomposition approach would be best for this project, find a fit that best represents the data, and explain the drawbacks and insights the chosen fit provides.

Appendix - Code

```
% Read in data
data = readmatrix("population.csv");
year = data(:,1);
pop = data(:,2);

%% Exponential Fit
% We want to fit to pop=ae^(bx) where a and b are unknown.
% To do so, we can make the equation linear and get
% pop_tilda = a_tilda + bx where a_tilda=log(a) and pop_tilda=log(pop).

V = [ones(size(year)) year];% Data Matrix
pop_tilda = log(pop);

% To find a and b, we'll solve Vc=pop_tilda with QR decomp
% which automatically finds the least square solution.
% Note: c = [a_tilda, b]

[Q,R]=qr(V,0);
c = backsolve(R,Q'*pop_tilda);

% Get Parameters
a = exp(c(1));
b = c(2);

% Plot
figure(1)
plot(year,pop,'o')
hold on
plot(year, a.*exp(b.*year))
xlabel('Year')
ylabel('Population')
```

```
legend('Data','Fit')
title('Exponential Fit with QR Decomposition')
hold off

%% Power-Law Fit
% We need to scale both axes to (0 1] and
% fit to power law pop = a*year^b
% pop_scaled = a*year_adj_scaled^b
% pop/max(pop) = a*( (year+abs(min(year)))/max( (year+abs(min(year))) ) )^b
pop_scaled = pop/max(pop);

year_adj = year + abs(min(year)) + 1; %added +1 to not have any zeros
year_adj_scaled = year_adj/max(year_adj);


% Make power law linear
% pop_scaled_tilda = a_tilda + b*year_adj_scaled_tilda
% where pop_scaled_tilda = log(pop_scaled)
% a_tilda = log(a)
% year_tilda = log(year_adj_scaled)
pop_scaled_tilda = log(pop_scaled);
year_adj_scaled_tilda = log(year_adj_scaled);

V = [ones(size(year_adj_scaled_tilda)) year_adj_scaled_tilda];

% lets solve Vc=pop_tilda where c = [a_tilda b] with QR Decomp.
% which finds the least square solution

[Q,R]=qr(V,0);
c = backsolve(R,Q'*pop_scaled_tilda);

% Get parameters
a = exp(c(1));
b = c(2);

figure(2)
plot(year_adj_scaled,pop_scaled,'o')
hold on
plot(year_adj_scaled, a.*year_adj_scaled.^b)
xlabel('Scaled Year')
ylabel('Scaled Population')
legend('Data','Fit')
title('Scaled and Shifted Power Law Fit with QR Decomposition')
hold off

%% Piecewise Fit - Exponential and Linear
% I will now do linear fit after 1950 and
% an exponential fit before 1950.

% Fit After 1950
year_after = data(1:end-24,1);
pop_after = data(1:end-24,2);
```

```
% We'd like to fit the data to pop = m*year+b where m and b are unknown.
% Thus the data matrix is the following:
V = [ones(size(year_after)) year_after];

% We will solve V*a=population where a = [b;m] using QR Decomposition
[Q,R]=qr(V,0);
c = backsolve(R,Q'*pop_after);

% Get Parameters
b = c(1);
m = c(2);

% Fit Before 1950
year_before = data(end-23:end,1);
pop_before = data(end-23:end,2);

% We want to fit to pop=ae^(dx) where a and d are unknown.
% To do so, we can make the equation linear and get
% pop_tilda = a_tilda + dx where a_tilda=log(a) and pop_tilda=log(pop).

V = [ones(size(year_before)) year_before];% Data Matrix
pop_tilda = log(pop_before);

% To find a and d, we'll solve Vc=pop_tilda with QR decomp
% which automatically finds the least square solution.
% Note: c = [a_tilda, d]

[Q,R]=qr(V,0);
c = backsolve(R,Q'*pop_tilda);

% Get Parameters
a = exp(c(1));
d = c(2);

% Plot
figure(3)
plot(year_after,pop_after,'o','Color','blue')
hold on
plot(year_before,pop_before,'o','Color','green')

plot(year_after,m*year_after+b,'Color','black')
plot(year_before, a.*exp(d.*year_before),'Color','red')

xlabel('Year')
ylabel('Population')
legend({'Data After 1950','Data Before 1950','Linear Fit After 1950','Exponential Fit Before
1950'},'Location','northwest')
title('Exponential and Linear Fit with QR Decomposition')
hold off
```