Analytical Modelling of Human Population

Instructor: Dr. Sunil Pratap Singh

Group 3: Abee Nelson - 18007
Aiswarya Lakshmi - 18017
Derin Wilson -18082
Harsha Sudhakaran - 18100
Madhav Sharma K N -18131

Indian Institute of Science Education and Research, Bhopal

Acknowledgement

We would like to express our special gratitude to our professor Dr Sunil Pratap Singh, for providing us with this golden opportunity to do a wonderful project on Human Population Modeling and for his constant guidance and support throughout the project. We would also like to thank our family and friends for their constant support in these challenging times.

Abstract

In this project, we tried to learn about human population modelling and how different factors affect the population growth. We concentrated on exponential modelling of the human population and used it to predict the future population of three different countries taken entirely from different economical backgrounds. We have also used logistic growth model to model human population. To predict the future population we have made use of Euler and RK4 method. We selected India from developing, USA from developed and Madagascar from under developed category. Apart from all this we also used birth and death rate to model the population of the countries taken into consideration. We were able to successfully predict the population of the countries and conclude many things. We also studied the fertility growth rate of these three countries and also predicted the fertility growth rate for next 10 years using polynomial regression.

Contents

1 Introduction

1.1 What is a Population Model ?

A model is a simplification of a real system that is used to provide some insights into the system. Population models are often (although not always) formulated as a set of rules or assumptions, expressed as mathematical equations, that describe change in population size over time as a consequence of survival and reproduction, and they include external factors that affect these characteristics. A model simplifies a system, retaining essential components while eliminating parts that are not of interest.

Ecology has a rich history of using models to gain insights into populations, often borrowing both model structures and analysis methods from demographers and engineers. Much of the development of the models has been a consequence of mathematicians and physicists seeing simple analogies between their models and patterns in natural systems. Consequently, one major application of ecological modeling has been to emphasize the analysis of dynamics of often complex models to provide insights into theoretical aspects of ecology.

1.2 Why is it important ?

Population modelling help scientists in making better predictions about future population size and growth rates. This is very essential for answering questions in areas such as biodiversity conservation (how quickly a species population is declining or whether it is at the verge of extinction) and human population growth (how fast will the human population grow and how to sustain the required resources and biodiversity).

Population models are also used for determining maximum harvests for agriculturists, environmental conservation, to understand the spread of parasites, viruses, and diseases.Human Population modelling is very important because it helps country's policy makers to make efficient policies and decision for there people.

2 Different Types of Population Models

2.1 Exponential Growth Model

A biological population with plenty of food, space to grow, and no threat from predators, tends to grow at a rate that is proportional to the population - that is, in each unit of time, a certain percentage of the individuals produce new individuals. If reproduction takes place more or less continuously, then this growth rate is represented by

$$\frac{dN}{dt} = rN,$$

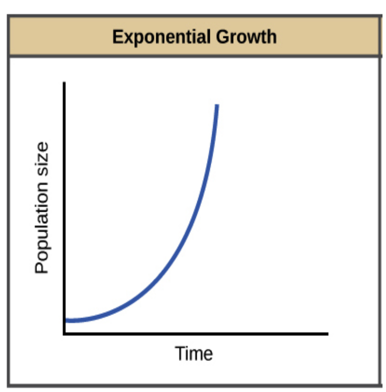
where N is the population as a function of time t, and r is the proportionality constant. We know that all solutions of this natural-growth equation have the form

$$N(t) = N_0 e^{rt},$$

where N_0 is the population at time $t = 0$. In short, unconstrained natural growth is exponential growth.

The best example of exponential growth is seen in bacteria. Bacteria are prokaryotes that reproduce by prokaryotic fission. This division takes about an hour for many bacterial species. If 1000 bacteria are placed in a large flask with an unlimited supply of nutrients, after an hour, there is one round of division and each organism divides, resulting in 2000 organisms - an increase of 1000. In another hour, each of the 2000 organisms will double, producing 4000, an increase of 2000 organisms. After the third hour, there should be 8000 bacteria in the flask, an increase of 4000 organisms.

Another example of exponential growth is that of human population.The world's population is presently growing at an exponential rate and hence the time required for the population to grow is becoming shorter.



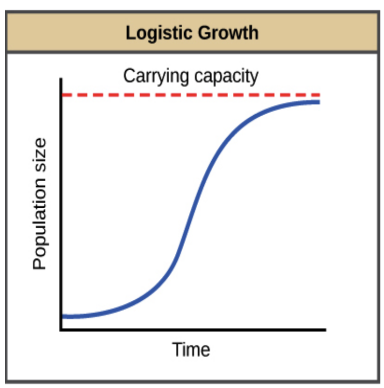
2.2 Logistic Growth Model

Exponential growth is possible only when infinite natural resources are available, this is not the case in the real world. In real world, there is limitation of resources and competition for existence among individuals. To model the reality of limited resources, ecologists developed the logistic growth model. Logistic model was first developed by Belgian mathematician Pierre Verhulst (1838) who suggested that the rate of population increase may be limited, i.e., it may depend on population density:

$$\frac{dN}{dt} = rN(1 - \frac{N}{K})$$

where the constant r defines the growth rate and K is the carrying capacity. Population growth rate declines with population, N, and reaches 0 when $N = K$. Parameter K is the upper limit of population growth and also called as carrying capacity. It is usually interpreted as the amount of resources expressed in the number of organisms that can be supported by these resources. If population numbers exceed K, then population growth rate becomes negative and population numbers decline.

Examples of organisms showing logistic growth include yeast, a microscopic fungus, that exhibits the classical S-shaped population size versus time curve when grown in a test tube. Its growth levels off as the population depletes the nutrients that are necessary for its growth.



3 Human Population Growth

World population has been growing since the end of the Black Death, around the year 1350. Improved agricultural productivity, sanitation and medical advancements that reduced mortality are the causes of such a growth. The world population is currently experiencing an exponential growth even though human reproduction is far below its biotic potential. A consequence of exponential human population growth is that the time it takes to add a particular number of humans to the earth is becoming shorter. 123 years were required to add 1 billion humans in 1930 but it took only 24 years to add 2 billion people between 1975 and 1999.

Global human population growth amounts to around 83 milllion annually or 1.1% per year. The global population has grown from 1 billion in 1800 to 7.9 billion in 2020. In accordance with present growth rate, the total population is estimated to be around 8.6 billion by mid 2030, 9.8 billion by mid 2050 and 11.2 billion by 2100. Humans are unique in their ability to alter their environment with the conscious purpose of increasing its carrying capacity. This ability is a major factor responsible for human population growth and a way of overcoming density-dependent growth regulation. Much of this ability is related to human intelligence, society, and communication. Other factors in human population growth are migration and public health. Public health, sanitation, and the use of antibiotics and vaccines have decreased the ability of infectious diseases to decrease human population growth.

3.1 Factors affecting Human Population Growth

3.1.1 Birth and Death Rate

Birth and death rates are crucial factors in determining population growth. Birth rate is typically indicated as the annual births per thousand total population. Similarly,the death rate is the annual deaths per thousand total population. The average global birth rate was 18.5 births per thousand people in 2016 and the death rate was 7.8 per thousand people.

3.1.2 Migration

Migration plays an important role in population growth. It changes not only the population size but also the composition of urban and rural population in terms of age and sex composition. The net migration rate is the difference between the number of immigrants (people coming into an area) and the number of emigrants (people leaving an area) throughout the year. The most important force behind European population change is international migration.

3.1.3 Fertility Rate

Fertility rate, average number of children born to women during their reproductive years. For the population in a given area to remain stable, an overall total fertility rate of 2.1 is needed, assuming no immigration or emigration occurs.

It is important to distinguish this from birth rates, which are defined as the number of live births per 1,000 women in the total population. The single most important factor in population growth is the total fertility rate (TFR). If, on average, women give birth to 2.1 children and these children survive to the age of 15, any given woman will have replaced herself and her partner upon death. A TFR of 2.1 is known as the replacement rate. Generally speaking, when the TFR is greater than 2.1, the population in a given area will increase, and when it is less than 2.1, the population in a given area will eventually decrease, though it may take some time because factors such as age structure, emigration, or immigration must be considered. More specifically, if there are numerous women of childbearing age and a relatively small number of older individuals within a given society, the death rate will be low, so even though the TFR is below the replacement rate, the population may remain stable or even increase slightly. This trend cannot last indefinitely but could persist for decades.

Tracking fertility rates allows for more efficient and beneficial planning and resource allocation within a particular region. If a country experiences unusually high sustained fertility rates, it may need to build additional schools or expand access to afford child care. This occurred in the United States during the post-World War II baby boom era. During this period, the TFR peaked at about 3.8, roughly twice the average 21^{st} century rate in the United States. The unusually high number of children born during this period left communities unprepared. Conversely, sustained low fertility rates may signify a rapidly aging population, which may place an undue burden on the economy through increasing health care and social security costs.

4 Numerical Methods Used

4.1 Solutions to system of Linear equations

4.1.1 Gauss-Elimination

Gauss elimination, also known as row reduction is an algorithm for solving system of linear equations. It consists of a sequence of operations, namely forward elimination and back substitution performed on the corresponding matrix of coefficients. The approach is to solve a set of n equations as shown below:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + ... + a_{1n}x_n = b_1 \tag{1}$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + ... + a_{2n}x_n = b_2 \tag{2}$$
$$\vdots$$
$$a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + ... + a_{nn}x_n = b_n \tag{3}$$

Forward elimination of unknowns:
Here the set of n equations is reduced to an upper triangular system. The initial step is to eliminate the first unknown, x_1, from the second through nth equations. This is done by multiplying equation (1) by $\frac{a_{21}}{a_{11}}$.

$$a_{21}x_1 + \frac{a_{21}}{a_{11}}a_{12}x_2 + ... + \frac{a_{21}}{a_{11}}a_{1n}x_n = \frac{a_{21}}{a_{11}}b_n \tag{4}$$

Now this equation is subtracted from equation (2) to give

$$(a_{22} - \frac{a_{21}}{a_{11}}a_{12})x_2 + ... + (a_{2n} - \frac{a_{21}}{a_{11}}a_{1n})x_n = (b_2 - \frac{a_{21}}{a_{11}})b_n \tag{5}$$

or

$$a'_{22}x_2 + ... + a'_{2n}x_n = b'_2 \tag{6}$$

Repeating the procedure for the remaining equations results in the following modified system:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + ... + a_{1n}x_n = b_1 \tag{7}$$
$$a'_{22}x_2 + a'_{23}x_3 + ... + a'_{2n}x_n = b'_2 \tag{8}$$
$$a'_{32}x_2 + a'_{33}x_3 + ... + a'_{3n}x_n = b'_3 \tag{9}$$
$$\vdots$$
$$a'_{n2}x_2 + a'_{n3}x_3 + ... + a'_{nn}x_n = b'_n \tag{10}$$

Now to eliminate the second unknown, x_2, we multiply equation (8) by $\frac{a'_{32}}{a'_{22}}$ and subtract the result from equation (9). Performing a similar elimination for the remaining equations we get

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + ... + a_{1n}x_n = b_1 \tag{11}$$

$$a'_{22}x_2 + a'_{23}x_3 + ... + a'_{2n}x_n = b'_2 \tag{12}$$

$$a''_{33}x_3 + ... + a''_{3n}x_n = b''_3 \tag{13}$$

$$\vdots$$

$$a''_{n3}x_3 + ... + a''_{nn}x_n = b''_n \tag{14}$$

where the double prime indicates that the coefficients have been modified twice. The procedure can be continued using the remaining pivot equations. The final manipulation in the sequence is to use the $(n-1)^{th}$ equation to eliminate the x_{n-1} term from the n^{th} equation.

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + ... + a_{1n}x_n = b_1 \tag{15}$$

$$a'_{22}x_2 + a'_{23}x_3 + ... + a'_{2n}x_n = b'_2 \tag{16}$$

$$a''_{33}x_3 + ... + a''_{3n}x_n = b''_3 \tag{17}$$

$$\vdots$$

$$a^{n-1}_{nn}x_n = b^{n-1}_n \tag{18}$$

Back substitution:
Equation (18) can be solved for x_n as

$$x_n = \frac{b^{n-1}_n}{a^{n-1}_{nn}} \tag{19}$$

This result can be back substituted into the $(n-1)^{th}$ equation to solve for x_{n-1}. The procedure which is repeated to evaluate the remaining x_i's can be represented by the following formula:

$$x_i = \frac{b^{i-1}_i - \sum_{j=i+1}^{n} a^{i-1}_{ij}x_j}{a^{i-1}_{ii}} \tag{20}$$

4.2 Curve-fitting

Curve fitting is a technique used in numerical methods to find an appropriate mathematical model that expresses the relationship between a dependent variable and an independent variable and then estimating the values of its parameters with regression techniques.

4.2.1 Linear Regression-Least Squares Regression

This the most common technique used for curve fitting. In this an approximate function is derived that fits the shape or general trend of the data without necessarily matching the individual points. One way to do this is to derive a curve that minimizes the discrepancy between the data points and the curve. A technique for accomplishing this objective is called least squares regression. For a form of linear equation i.e.

$$y = a_0 + a_1 x \tag{21}$$

we use linear regression where we will minimise the squares of errors/discrepancies.

$$S_r = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2 \tag{22}$$

So by minimising S_r with respect to the variables a_0 and a_1 we will get the equation as follows:

$$\frac{\partial S_r}{\partial a_0} = -2\sum (y_i - a_0 - a_1 x_i) \qquad \frac{\partial S_r}{\partial a_1} = -2\sum ((y_i - a_0 - a_1 x_i)\, x_i) \tag{23}$$

Now equating both equations to zero we get :

$$-2\sum (y_i - a_0 - a_1 x_i) = 0 \tag{24}$$

$$-2\sum ((y_i - a_0 - a_1 x_i)\, x_i) = 0 \tag{25}$$

And now solving above two equations we get :

$$a_1 = \frac{n\sum(x_i y_i) - \sum x_i \sum y_i}{n\sum(x_i)^2 - (\sum x_i)^2} \tag{26}$$

and

$$a_0 = \bar{y} - a_1 \bar{x} \tag{27}$$

where and are means of x_i and y_i

Fitting An Exponential Function Using Linear Regression
Let ,

$$y = \alpha e^{\lambda x} \tag{28}$$

We can linearize this function by taking natural logarithm on both sides

$$ln(y) = ln(\alpha) + \lambda x \tag{29}$$

So by plotting the graph of ln(y) vs x we can obtain the slope of the graph, from which we can determine the value of λ and from intercept we can deduce the value of α.

4.2.2 Polynomial Regression - Least Squares Method

In polynomial regression we try fit a polynomial of appropriate degree to given the data. The general form of an n degree polynomial is

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + ... + a_m x^m$$

For this case the sum of the residual is

$$S_r = \sum_{i=0}^{n} \left(y_i - (a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + ... + a_m x^m)\right)^2$$

Similar to linear regression(previous section) we take derivative with respect to each of the coefficient and we get the following set of equations:

$$\frac{\partial S_r}{\partial a_0} = -2 \sum \left(y_i - (a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + ... + a_m x^m)\right) \tag{30}$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum x_i \left(y_i - (a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + ... + a_m x^m)\right) \tag{31}$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum x_i^2 \left(y_i - (a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + ... + a_m x^m)\right) \tag{32}$$

$$\frac{\partial S_r}{\partial a_3} = -2 \sum x_i^3 \left(y_i - (a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + ... + a_m x^m)\right) \tag{33}$$

$$\vdots$$

$$\frac{\partial S_r}{\partial a_n} = -2 \sum x_i^n \left(y_i - (a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + ... + a_m x^m)\right) \tag{34}$$

These equations are set to zero and rearranged to develop the following set of normal equations:

$$m a_0 + \left(\sum x_i\right) a_1 + \left(\sum x_i^2\right) a_2 + \left(\sum x_i^3\right) a_3 + \ldots + \left(\sum x_i^m\right) a_m = \sum y_i$$

$$\left(\sum x_i\right) a_0 + \left(\sum x_i^2\right) a_1 + \left(\sum x_i^3\right) a_2 + \left(\sum x_i^4\right) a_3 + \ldots + \left(\sum x_i^{m+1}\right) a_m = \sum x_i y_i$$

$$\left(\sum x_i^2\right) a_0 + \left(\sum x_i^3\right) a_1 + \left(\sum x_i^4\right) a_2 + \left(\sum x_i^5\right) a_3 + \ldots + \left(\sum x_i^{m+2}\right) a_m = \sum x_i^2 y_i$$

$$\vdots$$

$$\left(\sum x_i^m\right) a_0 + \left(\sum x_i^{m+1}\right) a_1 + \left(\sum x_i^{m+2}\right) a_2 + \left(\sum x_i^{m+3}\right) a_3 + \ldots + \left(\sum x_i^{m+m}\right) a_n = \sum x_i^m y_i$$

We can solve the above set of equation using Gauss elimination, and find out the values of the coefficients a_0,a_1,a_2...,a$_m$.

The standard error for this method is

$$St_{error} = \sqrt{\frac{S_r}{n-(m+1)}} \tag{35}$$

here n is the number of data points and m is the degree of the polynomial to be fitted.

4.3 Solving Differential Equations

4.3.1 Euler's Method

The Euler Method (also called point-slope method) is a first order numerical procedure for solving ordinary differential equations with a given initial value. It is the most basic explicit method for numerical integration of ordinary differential equations and the simplest Runge-Kutta Method.
It is a first-order method, which means the local error (the error per step) is proportional to the square of step size and the global error (error at a given time) is proportional to step size.

$$\frac{dy}{dx} = f(x,y) \tag{36}$$

$$y_{i+1} = y_i + f(x_i, y_i) \cdot h \tag{37}$$

where $f(x_i, y_i)$ is the slope at x_i and h is the step size.

4.3.2 RK-4 Method

In numerical analysis, the Runge–Kutta methods are a family of implicit and explicit iterative methods which are used for finding the approximate solutions of ordinary differential equations. The most widely used member from the Runge-Kutta family is 'RK4' or 'Classic Runge-Kutta Method'.
The general form of RK-4 method is given by

$$y(x+h) \simeq y(x) + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{38}$$

where

$$k_1 = f(x,y)h \tag{39}$$

$$k_2 = f(x + h/2, y + k_1/2)h \tag{40}$$

$$k_3 = f(x + h/2, y + k_2/2)h \tag{41}$$

$$k_4 = f(x + h, y + k_3)h \tag{42}$$

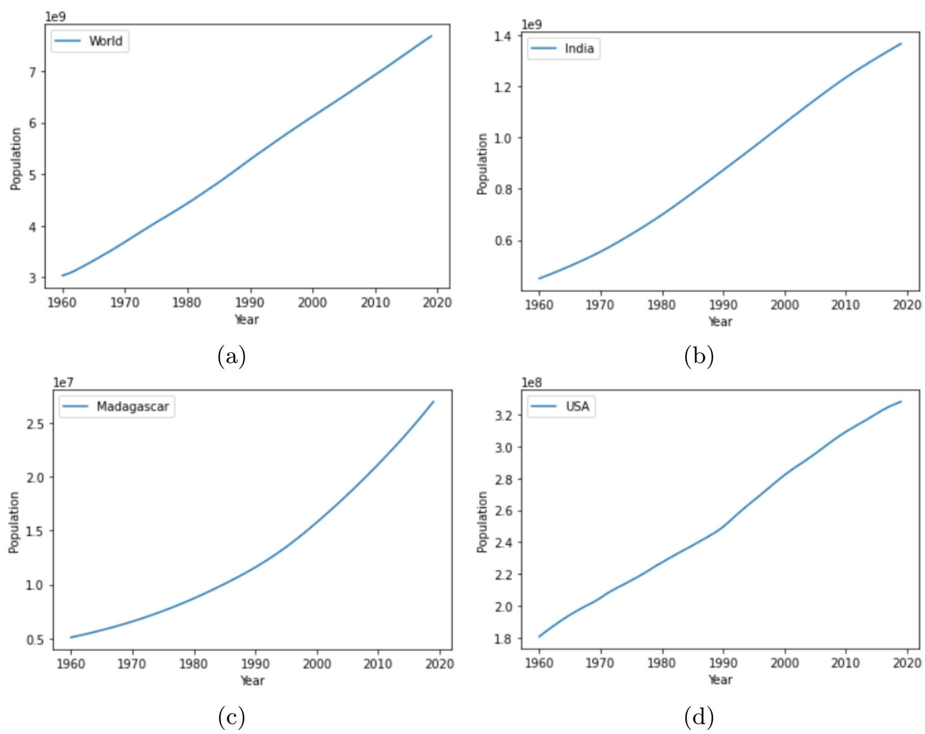
where h is the step-size and k_1 is the slope at the beginning of the interval, k_2 is the slope at the midpoint of the interval(using y and k_1), k_3 is again the slope at the midpoint(using y andk_2), k_4 is the slope at the end of the interval (using y and k_3). The RK-4 method is basically a fourth-order method, meaning that the local truncation error is on the order of $O(h^5)$, while the total accumulated error is of the order of $O(h^4)$.

5 Case Study

5.1 Analysing population of different countries based on Development

For this purpose, we took 3 different countries each from Developed, Developing and Under-Developed category. India represented Developing category,United States Of America represented Developed category and Madagascar (an island nation) represented under developed category. We took the population data from the World Bank and used different curve fitting methods to find an appropriate mathematical model. We also plotted the population data to get the general trend.

5.1.1 Population Graphs of different countries



(a) (b) (c) (d)

Figure 1: Population versus year graph for (a) World, (b) India, (c) Madagascar and (d) USA

From the above graphs, we can infer that all the graph follows an exponential increase especially that of Madagascar, and so in the next section we are going to try to fit an exponential curve to the given data and observe the trend.

5.1.2 Exponential Curve Fitting

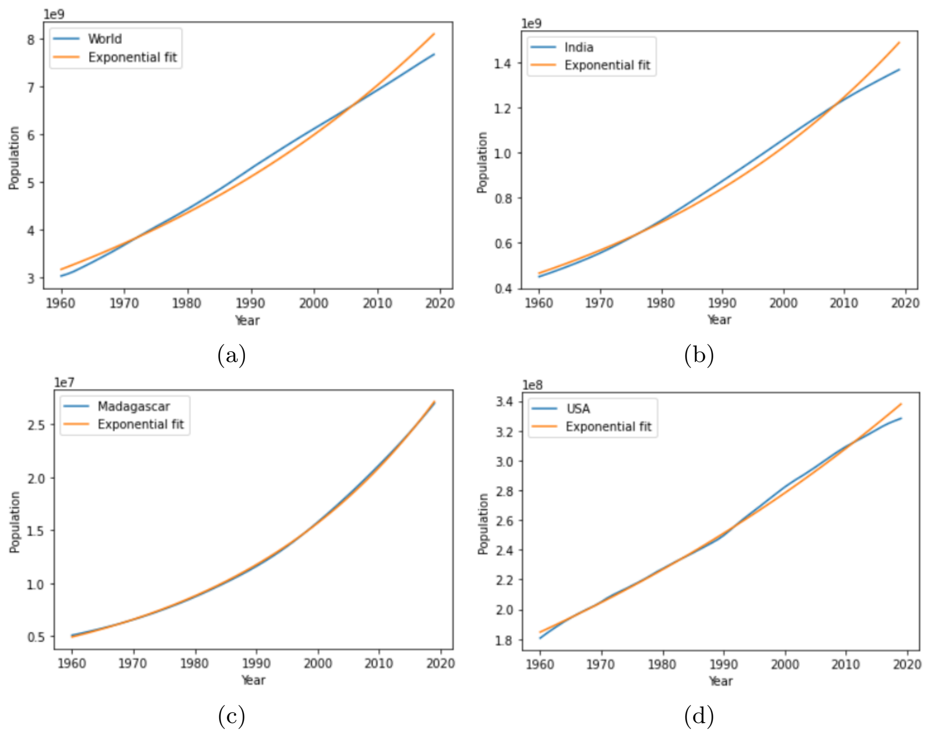


Figure 2: Population versus year graph for (a) World, (b) India, (c) Madagascar and (d) USA

In this section we fitted an exponential growth model of the form

$$\frac{dp}{dt} = r \cdot p$$

$$\implies p(t) = p_0 e^{r \cdot t}$$

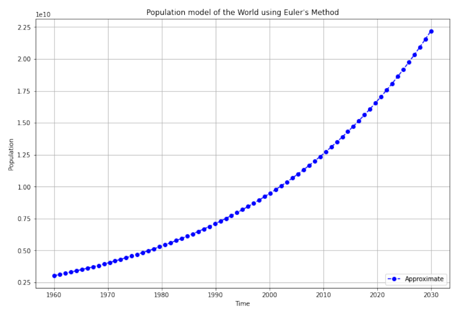
By curve fitting we can find out the value of r for each country and then use it in Euler's and RK4 method to predict the future trend.

$r_{World} = 0.015906861837129742$
$r_{India} = 0.01965563394203113$
$r_{Madagascar} = 0.02886290941170391$
$r_{USA} = 0.010234409905400798$

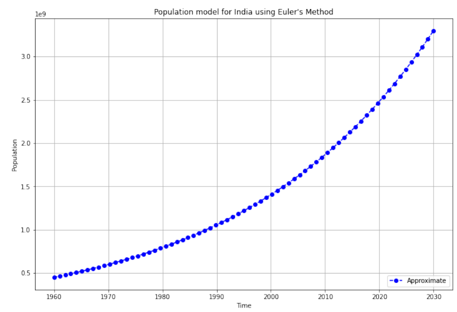
5.2 Predicting population trend for next 10 years

5.2.1 Euler Method

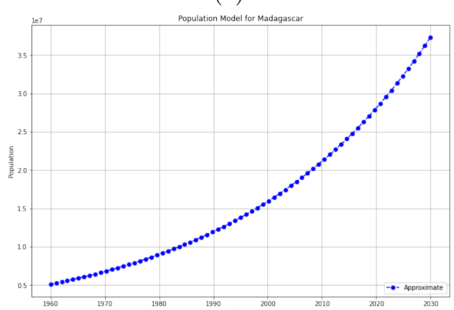
In this section we used Euler's algorithm to solve the differential equation and predict the next 10 years population trend for the three countries and world.



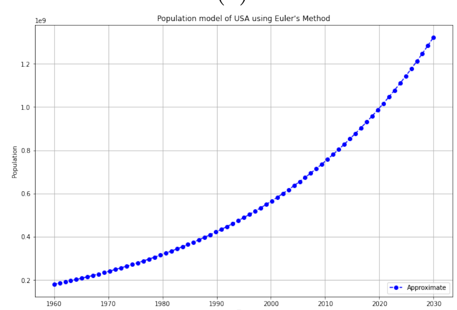
(a)



(b)



(c)

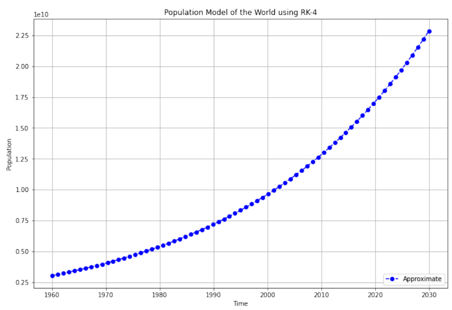


(d)

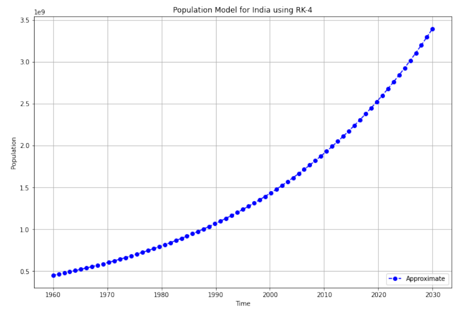
Figure 3: Population versus year graph using Euler Method for (a) World, (b) India, (c) Madagascar and (d) USA

5.2.2 RK4 Method

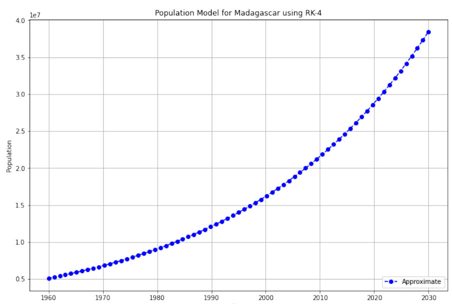
Under this section we used Runge-Kutta's algorithm to solve the exponential population growth model's differential equation and predict next 10 years population trend for the three countries and the world.



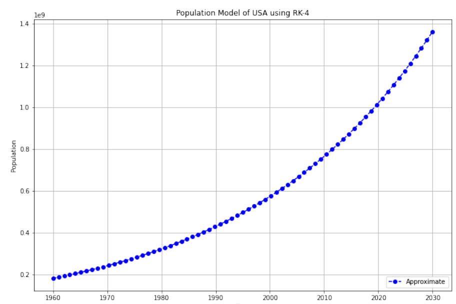
(a)



(b)



(c)



(d)

Figure 4: Population versus year graph using RK4 Method for (a) World, (b) India, (c) Madagascar and (d) USA

5.2.3 Observations

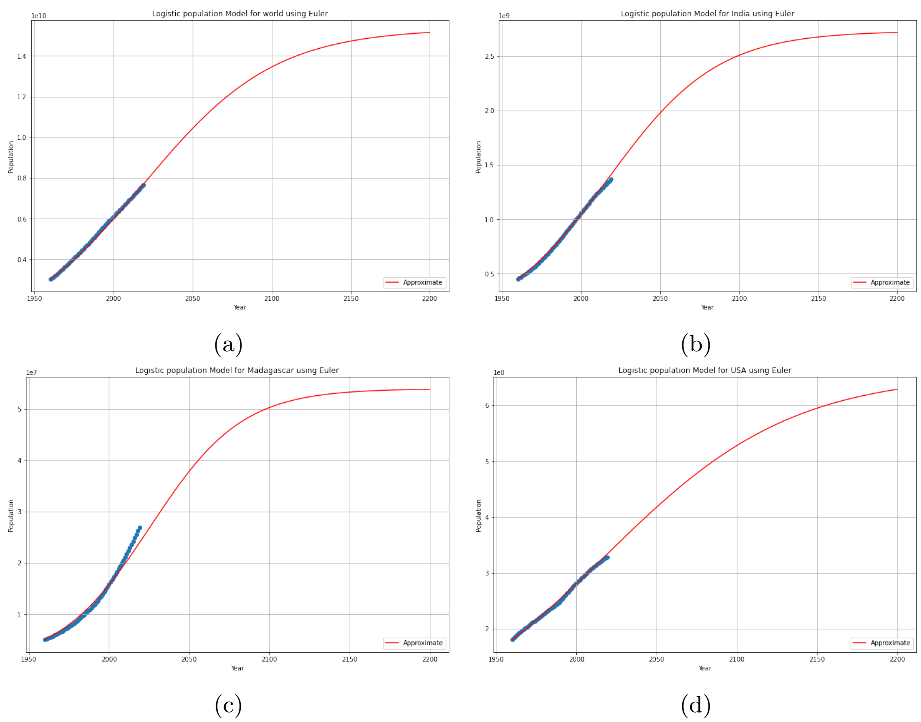
Using both the RK4 and Euler method to plot differential equation, our exponential model of population predicts that population of India will touch approximately 3.5 billion, Madagascar will touch 36 million, USA will touch 1.4 billion and whole world population will touch 27.5 billion. But the prediction has not taken into factors like fertility rate, mortality rate etc. of a country. Hence there is huge margin of error in this prediction. In the next section we will see fertility rate of different countries and try to make a better qualitative prediction.

5.3 Logistic Growth Model

To logistically model population of a country we need the carrying capacity (K) and population growth rate (r) of a country. For each country and for world we have assumed the carrying capacity K to be double the maximum population of the country. The r value for each country has been decided by trial and error method, also by using prior knowledge of population growth rate from the previous section.

5.3.1 Using Euler

Under this section we are going to use Euler's algorithm to solve the logistic growth differential equation and obtain the parameters r and K of the equation for observing the population trend for over 200 years and to find the flattening of the curve.

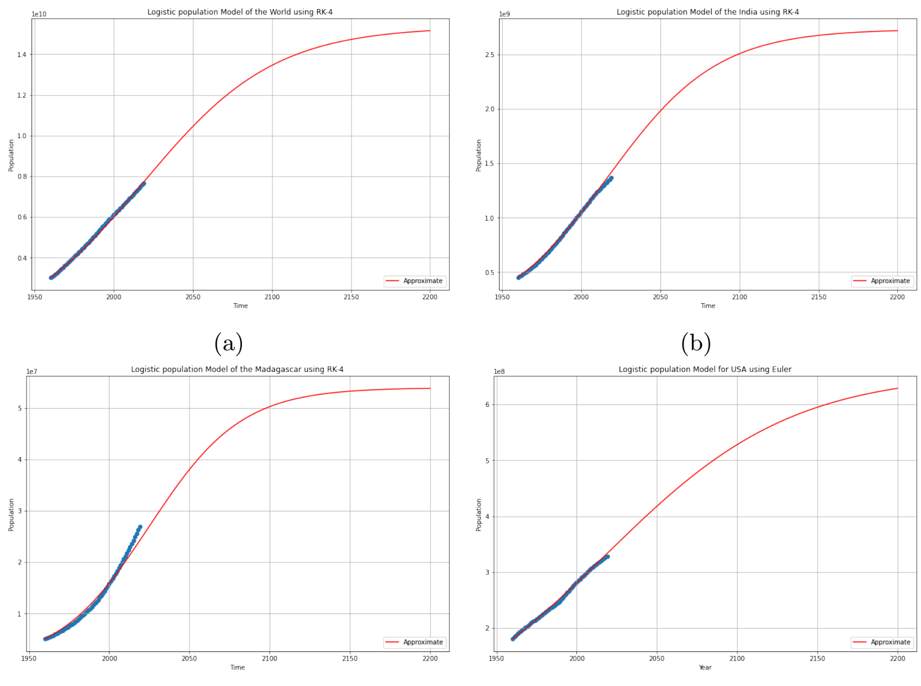


(a) (b) (c) (d)

Figure 5: Population growth via logistic model versus year graph using Euler Method for (a) World, (b) India, (c) Madagascar and (d) USA

5.3.2 Using RK4

Under this section we are going to use Runge–Kutta's algorithm to solve the logistic growth differential equation and obtain the parameters r and K of the equation for observing the population trend for over 200 years and to find the flattening of the curve.



(a) (b) (c) (d)

Figure 6: Population growth via logistic model versus year graph using RK4 Method for (a) World, (b) India, (c) Madagascar and (d) USA

5.3.3 Observations

As expected from Logistic growth model we got an S shaped graph. For most of the countries, the graph flattens after the year 2200 because the population reaches the carrying capacity (K) of the country.

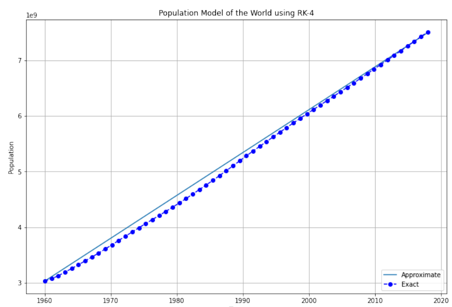
r_{World} =0.024
r_{India} =0.02879
$r_{Madagascar}$ = 0.03478
r_{USA} = 0.017

5.4 Modelling the population growth using Birth and Death Rate

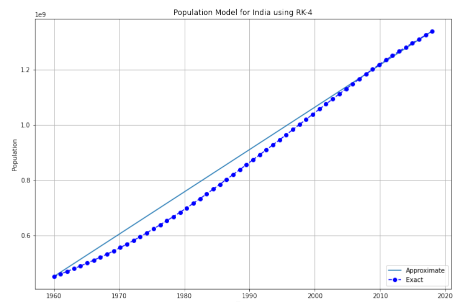
In this section, we take the average birth rate and death rate of the countries from 1960 to 2018 to model the population growth of the country using the formula

$$\frac{dp}{dt} = B - D$$

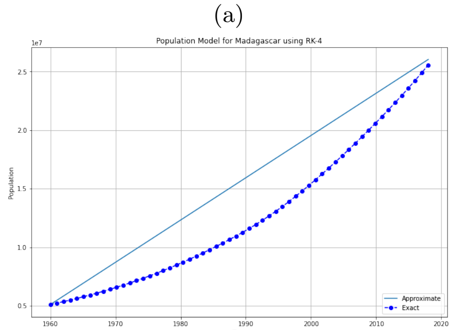
where B = average birth rate and D = average death rate
We use RK4 method to plot this differential equation once and calculate B and D for each country.



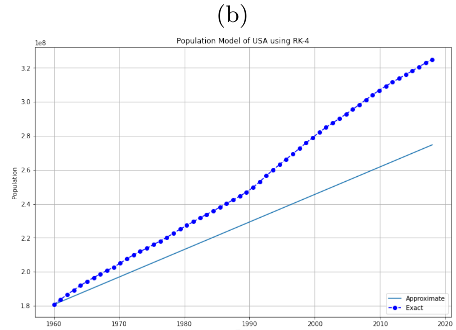
(a)



(b)



(c)

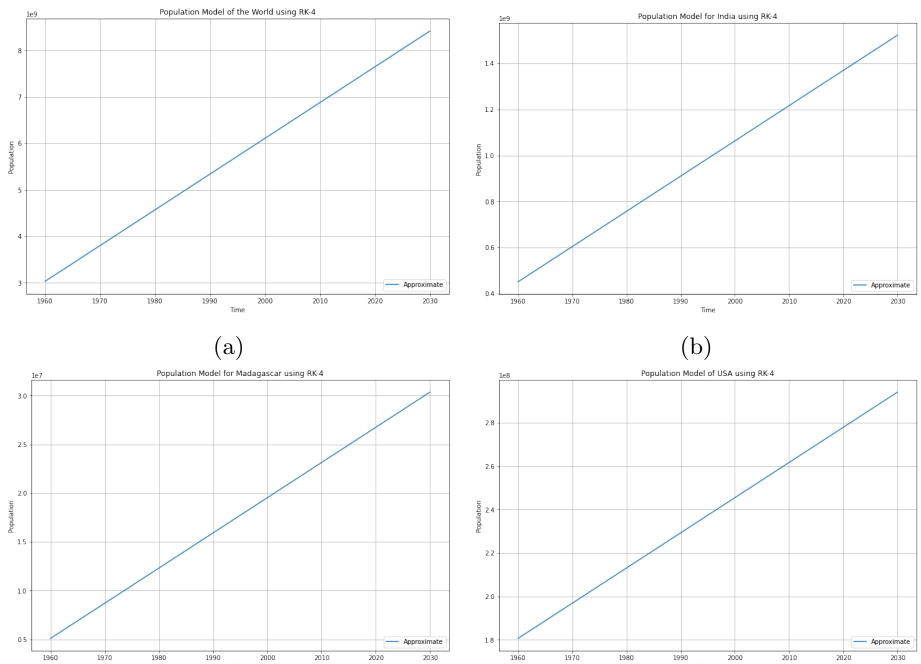


(d)

Figure 7: Population versus year graph using RK4 method for (a) World, (b) India, (c) Madagascar and (d) USA

So from the above figure it is clear that this model predicts the overall world population and India's population more accurately but miserably fails for USA and there is a significant error in the case of Madagascar. This huge error can be accounted if we take into consideration other factors like immigration, emigration etc, because USA is country that has a huge number of immigrations every year. Not taking into account the immigration factor is why this model fails miserably to predict USA's growth model. This also explains the reason why the model predicts the world population growth with very little error.

5.4.1 Predicting population trend for next 10 years using the above model



(a) (b)

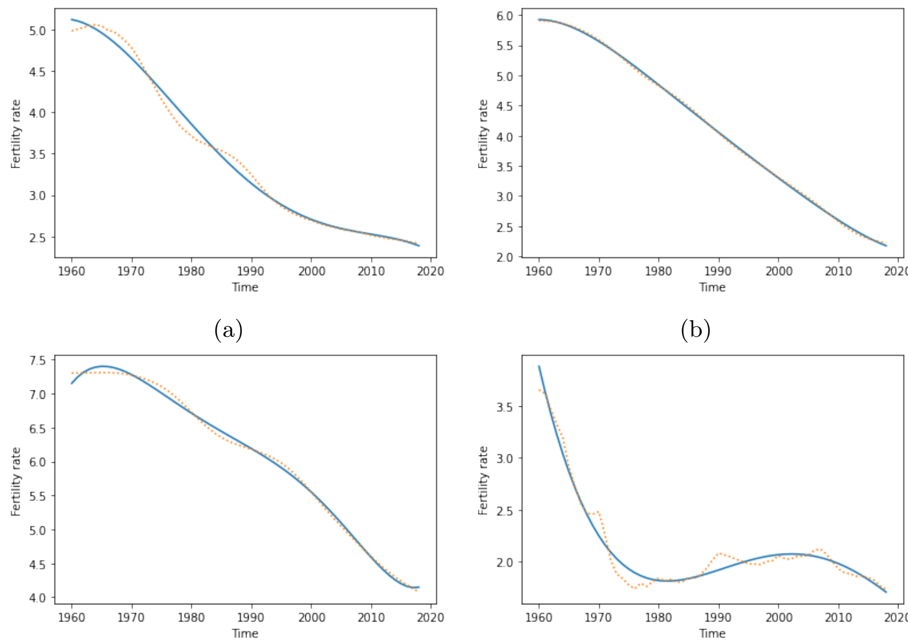
(c) (d)

Figure 8: Population versus year graph using RK4 method for (a) World, (b) India, (c) Madagascar and (d) USA

According to this model the world population will reach more than 8 billion by the end of 2030. At the same time India will reach a population of 1.5 billion approx, USA will reach 280 million and Madagascar will reach 30 million. From these numbers we can clearly say that by 2030 around 19% of the world population will be Indians.

5.5 Fertility growth rate

In this section we try to fit polynomial function (solid line) over the fertility growth rate data of a country obtained from the World Bank. We use polynomial regression technique for this purpose.



(a) (b) (c) (d)

Figure 9: Fertility versus year graph using polynomial regression for (a) World, (b) India, (c) Madagascar and (d) USA

So from the above graph it is clear that that India's fertility rate over the years have come down to 2 children per women in 2019 from 6 children per women in 1960. Similarly for USA it has come down from 4 to 1(approx) and for Madagascar it has come down from 7.5 to 4 children per women, which is still quiet high. Overall the world population fertility rate has come down from 5 to 2.5, which is a good sign. We can see all the countries are approaching the replacement rate of 2.1. As you can see from the graph, the fertility rate of USA has gone down from the replacement rate and hence the population is expected to decline. On the other hand India it is going to reach the replacement rate very soon and currently it is at 2.5, therefore India's population will grow at a more stable rate and will not see drastic increase. Whereas in the case of Madagascar the fertility rate is pretty high and hence it will see drastic increase in its population in the coming years.
So, from the graph we can conclude that Under Developed Countries have high

fertility growth rate, whereas Developed and Developing countries have comparatively low fertility growth rate.

5.5.1 Predicting next 10 year fertility growth

Using polynomial regression in the above section we found the polynomial that approximately mimics the fertility growth of different countries, using the same polynomial obtained in the previous section we are going to predict the fertility growth of different countries for next 10 years in this section.

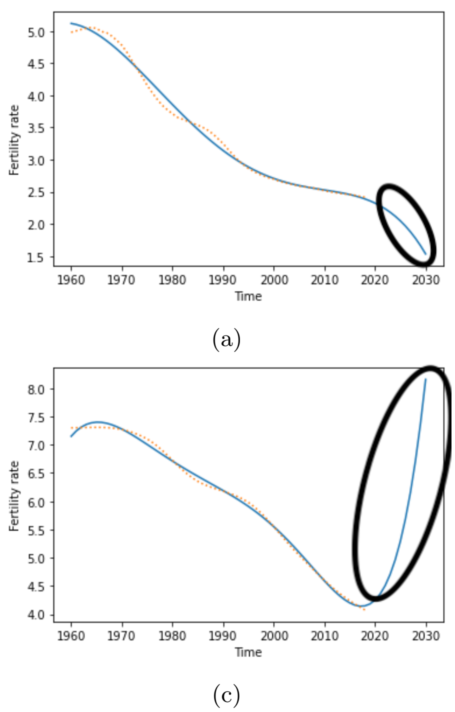


Figure 10: Fertility versus year graph using polynomial regression for (a) World, (b) India, (c) Madagascar and (d) USA

Now in the above graphs the black encircled region is the predicted fertility growth of different countries for next 10 years. We can see that for Madagascar there is an abrupt increase in fertility growth which is a worst case scenario, and hence our model is not accurate. But for the rest of the 2 countries and even for the world it does not show this kind of abrupt behaviour and hence we can take those predictions into account. So, the polynomial model for India predicts that by the next 10 years the fertility growth will experience a slight increase, whereas for USA it will further fall down from 1.5 to 1 (approx). Overall, the world's fertility growth is going to fall down and reach the mark of 1.5 by the end 2030.

6 Conclusion

So we can conclude following things :

- Population of developing countries like India is going to grow with a fertility rate of 2 children's per women and the total population will reach 1.5 billion by 2030.
- Population of developed countries like USA is going to grow with a fertility rate of 1 child per women and the total population will reach 280 million by 2030.
- Population of under developed countries like Madagascar will grow with a fertility rate of 4 children's per women and is going to reach 36 million by 2030.
- The total world population is going to grow with fertility rate of 2.5 children's per women and the total world population will reach 8 billion people by 2030.
- We can also interpret from above figures that under developing countries like Madagascar are going to experience exponential population growth, which is problem that needs to be addressed urgently. The people of this countries must be educated about their rising population.
- On the other hand developed and developing countries are doing good they brought down their fertility rate significantly and are managing their population growth efficiently.
- Also, according to logistic growth model it may flatten after the year 2200. That is population may stabilise by 2200 and it will not grow beyond that. This is true for all the three countries and world population.
- Fertility growth prediction also agree with the above observations because for every single country fertility growth is decreasing except Madagascar which we believe is an error associated with model.

7 Future Work

In this project we modeled population model with simple exponential growth model, logistic growth model and birth and death rate model. In the future we would like to work more on this project and come up with a mathematical model that takes into account different factors like economic status of a country, health facility, educational status etc. before predicting the population. We also wish to a make model that predicts the population more accurately with very less error because this will help countries to devise their future policies, that benefits all the people. We would like to inculcate machine learning techniques to predict more accurate mathematical models. We would also like to learn non-linear regression techniques so that we can fit the logistic curve more accurately to the data and find the parameters involved with more precision.

8 Codes

8.1 Curve fitting for Population (Exponential model)

```
import numpy as np
import math
import pandas as pd
import matplotlib.pyplot as plt

from google.colab import files
uploaded = files.upload()

data=pd.read_csv("population data - Sheet1.csv",header=None)

data

d=pd.DataFrame(data)

x_=pd.to_numeric(d.iloc[0,1:])

y_=pd.to_numeric(d.iloc[2,1:])

plt.plot(x_, y_, label = "line 1")

"""Curve fitting-Exponential"""

def estimate_coef(x, y):
    n = np.size(x)

    m_x, m_y = np.mean(x), np.mean(y)

    SS_xy = n*np.sum(y*x) - n*n*m_y*m_x
    SS_xx = n*np.sum(x*x) -n*n*m_x*m_x

    a_1 = SS_xy / SS_xx
    a_0 = m_y - a_1*m_x

    return(a_0, a_1)

def plot_regression_line(x, y, a):
    plt.scatter(x, y, color = "g",
                marker = "o", s = 30)

    y_pred = a[0] + a[1]*x

    plt.plot(x, y_pred, color = "r")

    plt.xlabel('x')
    plt.ylabel('y')

    plt.show()

x=x_
y=np.log(y_)
a = estimate_coef(x, y)
print(a)
plot_regression_line(x, y, a)
```

```

N0=math.exp(a[0])
l=a[1]

y=N0*np.exp(l*x)
plt.plot(x_, y_)
plt.plot(x_, y)
```

8.2 Euler's Method and RK4 for Population Model (Exponential model)

```
"""## Euler's Method"""

def euler(f,a,b,n,No):
  h = (b-a)/(n-1)
  ts = a + np.arange(n)*h
  ns = np.zeros(n)
  n = No
  for j,t in enumerate(ts):
    ns[j] = n
    n += h*f(t, n)
  return ts, ns

"""## RK-4 Method"""

def rk4(f,a,b,n,No):
  h = (b-a)/(n-1)
  ts = a + np.arange(n)*h
  ns = np.zeros(n)
  n = No
  for j,t in enumerate(ts):
    ns[j] = n
    k0 = h*f(t, n)
    k1 = h*f(t+h/2, n+k0/2)
    k2 = h*f(t+h/2, n+k1/2)
    k3 = h*f(t+h, n+k2)
    n += (k0 + 2*k1 + 2*k2 + k3)/6
  return ts, ns

"""## Population Model (Exponential Model)
### $$dN/dt=rN$$ N= population as a function of time, r=
    proportionality constant
### $$
N(t) = N_0 e^{rt},
$$ No= population at time t=0

### 1. Madagascar
r=0.02886290941170391, No=5099373
"""

def f(t,n):
  r=0.02886290941170391
  return r*n   #we are defining the differential equation here

"""### Euler's Method"""

```

```python
t, n= euler(f, 1960, 2030, 69,5099373 )

plt.figure(figsize = (12, 8))
plt.plot(t, n, 'bo--', label='Approximate')
plt.title('Population Model for Madagascar')
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()

"""### RK-4 Method"""

t1, n1= rk4(f, 1960, 2030, 69,5099373 )

plt.figure(figsize = (12, 8))
plt.plot(t1, n1, 'bo--', label='Approximate')
plt.title('Population Model for Madagascar using RK-4')
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()

"""## India
r=0.01965563394203113, No=450547679
"""

def f(t,n):
  r=0.01965563394203113
  return r*n

"""### Euler's Method"""

t, n= euler(f, 1960, 2030, 69, 450547679 )

plt.figure(figsize = (12, 8))
plt.plot(t, n, 'bo--', label='Approximate')
plt.title("Population model for India using Euler's Method")
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()

"""### RK-4 Method"""

t1, n1= rk4(f, 1960, 2030, 69, 450547679 )

plt.figure(figsize = (12, 8))
plt.plot(t1, n1, 'bo--', label='Approximate')
plt.title('Population Model for India using RK-4')
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()
```

```python

"""## USA
r=0.010234409905400798, No=180671000
"""

def f(t,n):
  r=0.010234409905400798
  return r*n

"""### Euler's Method"""

t, n= euler(f, 1960, 2030, 69, 180671000 )

plt.figure(figsize = (12, 8))
plt.plot(t, n, 'bo--', label='Approximate')
plt.title("Population model of USA using Euler's Method")
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()

"""### RK-4 Method"""

t1, n1= rk4(f, 1960, 2030, 69, 180671000 )

plt.figure(figsize = (12, 8))
plt.plot(t1, n1, 'bo--', label='Approximate')
plt.title('Population Model of USA using RK-4')
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()

"""## World
r=0.015906861837129742, No=3031437768
"""

def f(t,n):
  r=0.015906861837129742
  return r*n

"""### Euler's Method"""

t, n= euler(f, 1960, 2030, 69, 3031437768)

plt.figure(figsize = (12, 8))
plt.plot(t, n, 'bo--', label='Approximate')
plt.title("Population model of the World using Euler's Method")
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()

"""### RK-4 Method"""
```

```python

t1, n1= rk4(f, 1960, 2030, 69, 3031437768)

plt.figure(figsize = (12, 8))
plt.plot(t1, n1, 'bo--', label='Approximate')
plt.title('Population Model of the World using RK-4')
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()

c=0.0035429
d=-3.61298*10**-11

def p(t):
  return p

def f(t,p):
  c=0.0035429
  d=-3.61298*10**-11
  k= (p*c*(c-d*p))/c
  return k

t1, p1=rk4(f, 1960, 2019, 59, 450547679)

plt.figure(figsize = (12, 8))
plt.plot(t1, p1, 'bo--', label='Approximate')
plt.title("Population model of the World using Euler's Method")
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()
```

8.3 Logistic Growth Model

```python
data=pd.read_csv("population data - Sheet1.csv",header=None)

data

d=pd.DataFrame(data)

x_=pd.to_numeric(d.iloc[0,1:])

y_=pd.to_numeric(d.iloc[4,1:])

def euler(f,a,b,n,No):
  h = (b-a)/(n-1)
  ts = a + np.arange(n)*h
  ns = np.zeros(n)
  n = No
  for j,t in enumerate(ts):
    ns[j] = n
    n += h*f(t, n)
  return ts, ns

```

```
def rk4(f,a,b,n,No):
  h = (b-a)/(n-1)
  ts = a + np.arange(n)*h
  ns = np.zeros(n)
  n = No
  for j,t in enumerate(ts):
    ns[j] = n
    k0 = h*f(t, n)
    k1 = h*f(t+h/2, n+k0/2)
    k2 = h*f(t+h/2, n+k1/2)
    k3 = h*f(t+h, n+k2)
    n += (k0 + 2*k1 + 2*k2 + k3)/6
  return ts, ns

"""World"""

x_=pd.to_numeric(d.iloc[0,1:])
y_=pd.to_numeric(d.iloc[4,1:])

def f(t,n):
  r=0.024
  k=2*max(y_)
  return r*n*(1-(float)(n/k))  #we are defining the differential equation here

t1,n1 = euler(f,1960,2200,240,y_[1])

plt.figure(figsize = (12, 8))
plt.scatter(x_, y_)
plt.plot(t1,n1,'r',label = 'Approximate')
plt.title('Logistic population Model for world using Euler')
plt.xlabel('Year')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()

t1, n1= rk4(f, 1960, 2200, 240,y_[1])

plt.figure(figsize = (12, 8))
plt.plot(t1, n1,'r',label='Approximate')
plt.scatter(x_,y_)
plt.title('Logistic population Model of the World using RK-4')
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()

"""India"""

x_=pd.to_numeric(d.iloc[0,1:])
y_=pd.to_numeric(d.iloc[1,1:])

def f(t,n):
  r=0.02879
  k=2*max(y_)
```

```python
  return r*n*(1-(float)(n/k))  #we are defining the differential
    equation here

t1,n1 = euler(f,1960,2200,240,y_[1])

plt.figure(figsize = (12, 8))
plt.scatter(x_, y_)
plt.plot(t1,n1,'r',label = 'Approximate')
plt.title('Logistic population Model for India using Euler')
plt.xlabel('Year')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()

t1, n1= rk4(f, 1960, 2200, 240,y_[1])

plt.figure(figsize = (12, 8))
plt.plot(t1, n1,'r',label='Approximate')
plt.scatter(x_,y_)
plt.title('Logistic population Model of the India using RK-4')
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()

"""Madagascar"""

x_=pd.to_numeric(d.iloc[0,1:])
y_=pd.to_numeric(d.iloc[3,1:])

def f(t,n):
  r=0.03478
  k=2*max(y_)
  return r*n*(1-(float)(n/k))  #we are defining the differential
    equation here

t1,n1 = euler(f,1960,2200,240,y_[1])

plt.figure(figsize = (12, 8))
plt.scatter(x_, y_)
plt.plot(t1,n1,'r',label = 'Approximate')
plt.title('Logistic population Model for Madagascar using Euler')
plt.xlabel('Year')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()

t1, n1= rk4(f, 1960, 2200, 240,y_[1])

plt.figure(figsize = (12, 8))
plt.plot(t1, n1,'r', label='Approximate')
plt.scatter(x_,y_)
plt.title('Logistic population Model of the Madagascar using RK-4')
plt.xlabel('Time')
```

```python
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()

"""USA"""

x_=pd.to_numeric(d.iloc[0,1:])
y_=pd.to_numeric(d.iloc[2,1:])

def f(t,n):
  r=0.017
  k=2*max(y_)
  return r*n*(1-(float)(n/k))  #we are defining the differential
    equation here

t1,n1 = euler(f,1960,2200,240,y_[1])

plt.figure(figsize = (12, 8))
plt.scatter(x_, y_)
plt.plot(t1,n1,'r',label = 'Approximate')
plt.title('Logistic population Model for USA using Euler')
plt.xlabel('Year')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()

t1, n1= rk4(f, 1960, 2200, 240,y_[1])

plt.figure(figsize = (12, 8))
plt.plot(t1, n1,'r', label='Approximate')
plt.scatter(x_,y_)
plt.title('Logistic population Model of the USA using RK-4')
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()
```

8.4 Population Modelling using Birth Rate and Death Rate

```python
data_p=pd.read_excel("population data.xlsx",header=None)
data_p

data_b=pd.read_excel("Birth_Rate.xlsx",header=None)
data_b

data_d=pd.read_excel("Death_Rate.xlsx",header=None)
data_d

z=4
d1_d=data_d.iloc[z]
d1_p=data_p.iloc[z]
d1_b=data_b.iloc[z]
```

```
sum = 0

for i in range(1,np.size(d1_d)):
   sum = sum + ((d1_d[i]*d1_p[i])/1000.0)
d_avg = sum/np.size(d1_d)
d_avg

sum = 0

for i in range(1,np.size(d1_b)):
   sum = sum + ((d1_b[i]*d1_p[i])/1000.0)
b_avg = sum/np.size(d1_b)
b_avg

def rk4(f,a,b,n,No):
  h = (b-a)/(n-1)
  ts = a + np.arange(n)*h
  ns = np.zeros(n)
  n = No
  for j,t in enumerate(ts):
    ns[j] = n
    k0 = h*f(t, n)
    k1 = h*f(t+h/2, n+k0/2)
    k2 = h*f(t+h/2, n+k1/2)
    k3 = h*f(t+h, n+k2)
    n += (k0 + 2*k1 + 2*k2 + k3)/6
  return ts, ns

def f(t,n):
  return b_avg-d_avg

"""India"""

t1, n1= rk4(f, 1960, 2018, 58, 450547679 )

plt.figure(figsize = (12, 8))
plt.plot(t1, n1, linestyle='solid', label='Approximate')
plt.plot(t1,d1_p[1:59], 'bo--', label='Exact')
plt.title('Population Model for India using RK-4')
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()

"""USA"""

t1, n1= rk4(f, 1960, 2018, 58, 180671000 )

plt.figure(figsize = (12, 8))
plt.plot(t1, n1, linestyle='solid', label='Approximate')
plt.plot(t1,d1_p[1:59], 'bo--', label='Exact')
plt.title('Population Model of USA using RK-4')
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
```

```
plt.legend(loc='lower right')
plt.show()

"""Madagascar

"""

t1, n1= rk4(f, 1960, 2018, 58,5099373 )

plt.figure(figsize = (12, 8))
plt.plot(t1, n1, linestyle='solid', label='Approximate')
plt.plot(t1,d1_p[1:59], 'bo--', label='Exact')
plt.title('Population Model for Madagascar using RK-4')
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()

"""World"""

t1, n1= rk4(f, 1960, 2018, 58, 3031437768)

plt.figure(figsize = (12, 8))
plt.plot(t1, n1, linestyle='solid', label='Approximate')
plt.plot(t1,d1_p[1:59], 'bo--', label='Exact')
plt.title('Population Model of the World using RK-4')
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()
```

8.5 Prediction using Birth Rate and Death Rate

```
"""
India
"""

t1, n1= rk4(f, 1960, 2030, 69, 450547679 )

plt.figure(figsize = (12, 8))
plt.plot(t1, n1, label='Approximate')
plt.title('Population Model for India using RK-4')
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()

"""USA

"""

t1, n1= rk4(f, 1960, 2030, 69, 180671000 )

plt.figure(figsize = (12, 8))
```

```
plt.plot(t1, n1, 'bo--', label='Approximate')
plt.title('Population Model of USA using RK-4')
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()

"""Madagascar

"""

t1, n1= rk4(f, 1960, 2030, 69,5099373 )

plt.figure(figsize = (12, 8))
plt.plot(t1, n1, 'bo--', label='Approximate')
plt.title('Population Model for Madagascar using RK-4')
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()

"""World"""

t1, n1= rk4(f, 1960, 2030, 69, 3031437768)

plt.figure(figsize = (12, 8))
plt.plot(t1, n1, 'bo--', label='Approximate')
plt.title('Population Model of the World using RK-4')
plt.xlabel('Time')
plt.ylabel('Population')
plt.grid()
plt.legend(loc='lower right')
plt.show()
```

8.6 Curve fitting for Fertility Rate (Exponential model)

```
d1=pd.read_csv("Fertility_rate _data - Sheet1.csv",header=None)

d1

d1=pd.DataFrame(d1)

y_=pd.to_numeric(d1.iloc[3,1:])
x_=pd.to_numeric(d1.iloc[0,1:])
plt.plot(x_, y_, label = "line 1")

def estimate_coef(x, y):
    n = np.size(x)

    m_x, m_y = np.mean(x), np.mean(y)

    SS_xy = n*np.sum(y*x) - n*n*m_y*m_x
    SS_xx = n*np.sum(x*x) -n*n*m_x*m_x

    a_1 = SS_xy / SS_xx
```

```
    a_0 = m_y - a_1*m_x

    return(a_0, a_1)

def plot_regression_line(x, y, a):
    plt.scatter(x, y, color = "g",
                marker = "o", s = 30)

    y_pred = a[0] + a[1]*x

    plt.plot(x, y_pred, color = "r")

    plt.xlabel('x')
    plt.ylabel('y')

    plt.show()

x=x_
y=np.log(y_)
a = estimate_coef(x, y)
print(a)
plot_regression_line(x, y, a)

N0=math.exp(a[0])
l=a[1]

y=N0*np.exp(l*x)
plt.plot(x_, y_)
plt.plot(x_, y)
```

8.7 Polynomial Regression for Fertility Rate

```
df=pd.read_excel("Fertility_rate _data.xlsx",header=None)
print(df)

df1=df.iloc[4]
df1

fertility=df1.to_numpy()
fertility

year=np.zeros((59))
i=0
for i in range(59):
    year[i]=1960+i
print(year)

P=np.zeros((59))
for i in range(59):
    P[i]=fertility[i+1]
print(P)

Errortable=np.zeros((30))
equation=np.zeros(([30,30]))
for n in range(1,30):
    AUGM=np.zeros(([n+1,n+1]))
    AUGM2=np.zeros(n+1)
```

```
    i=0
    j=0
    for i in range (n+1):
        for j in range (n+1):
            sum=0
            for k in range(np.size(P)):
                sum= sum+((year[k])**(j+i))

            AUGM[j][i]= sum

    for h in range(n+1):
        sum=0
        for k in range(np.size(P)):
            sum=sum+(P[k]*(year[k]**h))

        AUGM2[h]=sum
    m = n+1
    x=[0]*m
    for k in range(0,m-1):
        for i in range(k+1,m):
            factor = AUGM[i][k]/AUGM[k][k]
            for j in range(k+1,m):
                AUGM[i][j]=AUGM[i][j]-factor*AUGM[k][j]
            AUGM2[i]=AUGM2[i]-factor*AUGM2[k]

    x[m-1]=AUGM2[m-1]/AUGM[m-1][m-1]
    for i in range(n-1,-1,-1):
        sum = AUGM2[i]
        for j in range(i+1,m):
            sum = sum - AUGM[i][j]*x[j]
        x[i] = sum/AUGM[i][i]

    Sr=0
    for q in range(59):
        sum2=0
        for e in range(m):
            sum2=sum2+ (x[e]*(year[q]**e))

        Sr=Sr+((P[q]-(sum2))**2)
    Sterror= (Sr/(60-(n+1)))**(1/2)
    print(Sterror)
    Errortable[n]=Sterror
    for i in range(m):
        equation[n][i]=x[i]

print(AUGM[0][0],AUGM[1][0])

dummy =Errortable[1]
degree=1
for i in range(2,30):
    if Errortable[i]<=dummy:
        dummy=Errortable[i]
        degree=i
print("degree=",degree,",standard error=", dummy,)

plt.plot(year,P,linestyle='dotted')

```

```
equation[degree]

y=0
for i in range(degree+1):
    y=y+(equation[degree][i]*(year**i))

plt.plot(year,y,linestyle='solid')
plt.plot(year,P,linestyle='dotted')

for i in range(1,30):
    v=0
    for j in range(30):
        v=v+equation[i][j]*(year**j)

    plt.plot(year,v)
```

8.8 Predicting Fertility Rate with Polynomial Regression

```
y=0
yearnew=np.zeros((71))
for i in range(71):
    yearnew[i]=1960+i
print(yearnew)
for i in range(degree+1):
    y=y+(equation[degree][i]*(yearnew**i))

plt.plot(yearnew,y,linestyle='solid')
plt.plot(year,P,linestyle='dotted')
plt.xlabel('Time')
plt.ylabel('Fertility rate')

for i in range(1,30):
    v=0
    for j in range(30):
        v=v+equation[i][j]*(year**j)

    plt.plot(year,v)
```

9 References

- https://data.worldbank.org/indicator/SP.POP.TOTL
- Numerical Methods for Engineers by Steven C. Chapra and Raymond P. Canale.
- moak, Natalie. ”Fertility rate”. Encyclopedia Britannica, 12 May. 2016, https://www.britannica.com/topic/fertility-rate. Accessed 16 April 2021.
- Ecology and the environment-lumen biology https://courses.lumenlearning.com/biology2xmaster/chapter/human-population-growth/
- Vandermeer, J. (2010) How Populations Grow: The Exponential and Logistic Equations. Nature https://www.nature.com/scitable/knowledge/library/how-populations-grow-the-exponential-and-logistic-13240157/
- Population ecology-Khan academy https://www.khanacademy.org/science/ap-biology/ecology-ap/population-ecology-ap/a/exponential-logistic-growth
- Population growth and Regulation-Concepts of Biology https://opentextbc.ca/conceptsofbiologyopenstax/chapter/population-growth-and-regulation/
- https://en.wikipedia.org/wiki/Populationgrowth