**Full Code Appendix - Team el185**

**Team:**

**Ebrahim Lambat (el185 - 199029365)**

**Saira Qurashi (sq28 -199044741 )**

**Shamia Sanjania (sms87 - 199011720)**

**Appendix 1.1 – Importing Data and Smoothing the data:**

% Importing data from the document provided by the Professor

Alldata = readtable("WHO-COVID-19-global-data.csv"); %import data

format long

TT = table2array(Alldata(135433:136059,6)); % Cumulative Cases for Trinidad and Tobago

SA = table2array(Alldata(117877:118503,6)); % Cumulative Cases for Saudi Arabia

CY = table2array(Alldata(32605:33231,6)); % Cumulative Cases for Cyprus

time = linspace(1,627,627); % Creating a timeframe for the graphs

% Removing any anomalous results by averaging out the values

CY(107) = 748;

CY(290) = 2512;

CY(301) = 3934;

CY(359) = 19422;

CY(360) = 19528;

CY(361) = 19634;

CY(487) = 67717;

CY(627) = 119003;

SA(77) = 258;

TT(221) = 253;

TT(237) = 1164;

TT(238) = 1297;

TT(239) = 1430;

TT(242) = 1652;

TT(260) = 3403;

TT(267) = 4153;

TT(281) = 4920;

TT(289) = 5072;

TT(293) = 5315;

TT(306) = 5721;

TT(307) = 5738;

TT(308) = 5755;

TT(314) = 5865;

TT(327) = 6400;

TT(344) = 6821;

TT(402) = 7612;

TT(498) = 14308;

TT(559) = 34906;

TT(600) = 43054;

TT(622) = 47247;

**Appendix 1.2 – Plotting Graphs for each country:**

% Plotting raw data of each country

figure;

plot(time, TT);

title("Number of cases over time for Trinidad and Tobago")

xlabel("Days since 03/01/2020")

ylabel("Cumulative Cases")

xlim([0 640])

figure;

plot(time, SA);

title("Number of cases over time for Saudi Arabia")

xlabel("Days since 03/01/2020")

ylabel("Cumulative Cases")

xlim([0 640])

figure;

plot(time, CY);

title("Number of cases over time for Cyprus")

xlabel("Days since 03/01/2020")

ylabel("Cumulative Cases")

xlim([0 640])

**Appendix 1.3 – Plotting Graphs for each country for our start of COVID:**

% Cumulative graphs from positive cases being greater than or equal to 200

TTCC = TT(218:end,:); % Start of TT cases over 200

SACC = SA(76:end,:); % Start of SA cases over 200

CYCC = CY(88:end,:); % Start of CY cases over 200

figure;

timeTTCC = linspace(1,numel(TTCC),numel(TTCC));

plot(timeTTCC, TTCC);

title("Number of cases over time since start of COVID for TT")

xlabel("Days since start of COVID")

ylabel("Cumulative Cases")

figure;

timeSACC = linspace(1,numel(SACC),numel(SACC));

plot(timeSACC, SACC);

title("Number of cases over time since start of COVID for SA")

xlabel("Days since start of COVID")

ylabel("Cumulative Cases")

figure;

timeCYCC = linspace(1,numel(CYCC),numel(CYCC));

plot(timeCYCC, CYCC);

title("Number of cases over time since start of COVID for CY")

xlabel("Days since start of COVID")

ylabel("Cumulative Cases")

**Appendix 1.4 – Standardising the Cumulative cases for each country:**

% Standardising the cumulative cases with the corresponding population for that region)

Population\_CY=1207359; % total population of Cyprus

N\_CY=CYCC/Population\_CY;

figure;

plot(N\_CY);

title("Normalised Cumulative Cases for Cyprus");

xlabel("Days since start of COVID");

ylabel("Cumulative Fraction");

Population\_SA=34813871;% total population Saudi Arabia

N\_SA=SACC/Population\_SA;

figure;

plot(N\_SA);

title("Normalised Cumulative Cases for Saudi Arabia");

xlabel("Days since start of COVID");

ylabel("Cumulative Fraction");

Population\_TT=1399488; % total population of Trinidad and Tobago

N\_TT=TTCC/Population\_TT;

figure;

plot(N\_TT);

title("Normalised Cumulative Cases for Trinidad & Tobago");

xlabel("Days since start of COVID");

ylabel("Cumulative Fraction");

%Combining the graphs

y1=plot(N\_CY);

title('Normalised Cumulative Cases ')

xlabel("Days since start of COVID");

ylabel("Cumulative Fraction");

hold on

y2 = plot(N\_SA);

hold on

y3 = plot(N\_TT);

hold off

legend('Cyprus','Saudi Arabia','Trinidad & Tobago')

**Appendix 1.5 – Identifying number and size of waves for each country:**

smoothSA=smoothdata(N\_SA);

for i = 3:size(N\_SA, 1)-2

smoothSA(i, 1) = (N\_SA(i-2, 1) + N\_SA(i-1, 1) + N\_SA(i, 1) + N\_SA(i+1, 1) + N\_SA(i+2))./5;

end

AttemptSmooth1\_SA = zeros(size(N\_SA, 1), 1);

for i = 1:size(N\_SA, 1)

AttemptSmooth1\_SA(i, 1) = N\_SA(i, 1) < smoothSA(i, 1);

end

figure;

plot(medfilt1(AttemptSmooth1\_SA, 120))

title("SA Splitting of Waves")

xlabel("Days since start of COVID")

xlim([-100 600])

ylim([-0.5 1.5])

smoothCY = smoothdata(N\_CY);

for i = 3:size(N\_CY, 1)-2

smoothCY(i, 1) = (N\_CY(i-2, 1) + N\_CY(i-1, 1) + N\_CY(i, 1) + N\_CY(i+1, 1) + N\_CY(i+2))./5;

end

AttemptSmooth1\_CY = zeros(size(N\_CY, 1), 1);

for i = 1:size(N\_CY, 1)

AttemptSmooth1\_CY(i, 1) = N\_CY(i, 1) < smoothCY(i, 1);

end

figure;

plot(medfilt1(AttemptSmooth1\_CY, 80))

title("CY splitting of waves")

xlabel("Days since start of COVID")

xlim([-100 600])

ylim([-0.5 1.5])

smoothTT = smoothdata(N\_TT);

for i = 3:size(N\_TT, 1)-2

smoothTT(i, 1) = (N\_TT(i-2, 1) + N\_TT(i-1, 1) + N\_TT(i, 1) + N\_TT(i+1, 1) + N\_TT(i+2))./5;

end

AttemptSmooth1\_TT = zeros(size(N\_TT, 1), 1);

for i = 1:size(N\_TT, 1)

AttemptSmooth1\_TT(i, 1) = N\_TT(i, 1) < smoothTT(i, 1);

end

figure;

plot(medfilt1(AttemptSmooth1\_TT, 109))

xlim([-100 500])

ylim([-0.5 1.5])

title("TT splitting of waves")

xlabel("Days since start of COVID")

**Appendix 2.1 – Creating the variables for the coming figures (CY):**

CYCC1 = N\_CY(1:137);

days1\_CY = 1:137;

CYCC2 = CYCC(138:348);

CYCC2 = CYCC2 - 1291;

CYCC2 = CYCC2/Population\_CY;

days2\_CY = 1:211;

CYCC3 = CYCC(349:420);

CYCC3 = CYCC3 - 37880;

CYCC3 = CYCC3/Population\_CY;

days3\_CY = 1:71;

CYCC4 = CYCC(421:540);

CYCC4 = CYCC4 - 35032;

CYCC4 = CYCC4/Population\_CY;

days4\_CY = 1:120;

**Appendix 2.2 – Wave 1 (CY):**

% %Wave 1

CY\_Wave1 = N\_CY(1:137,1);

%Plotting normalised wave

figure;

plot(CY\_Wave1);

title("Wave 1 of Normalised Cumulative Cases for Cyprus");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

xlim([0 140]);

% Plotting log wave

figure;

logwave1\_CY = log(CY\_Wave1);

plot(logwave1\_CY);

title("Logarithm Plot for Wave 1 of Normalised Cumulative Cases for CY");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

% a and r

mdl1a\_CY = fitlm(1:13, log(CYCC1(1:13)));

r\_CY1 = table2array(mdl1a\_CY.Coefficients(2, 1));

a\_CY1 = table2array(mdl1a\_CY.Coefficients(1, 1));

% Exponential model prediction

figure;

hold on;

CY\_Wave1 = N\_CY(1:137, 1); %normalised

plot(CY\_Wave1);

CYexp1 = zeros(137, 1);

for i = 1:137

CYexp1(i, 1) = exp(a\_CY1+(r\_CY1.\*(i))); % exp

end

plot(CYexp1);

hold off;

title("CY Wave 1 Exp Model Prediction");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

legend('Observed','Predicted');

ylim([0 0.00060]);

%  Error

figure;

Err\_CY1 = zeros(137, 1);

for i = 1:137

Err\_CY1(i, 1) = CY\_Wave1(i)-CYexp1(i); % need an array, not a matrix

end

plot(Err\_CY1);

title("Error for Wave 1 CY");

xlabel("Days since wave beginning");

ylabel("Error of Cumulative Fraction");

% K

figure;

K\_CY1 = zeros(84, 1);

for i = 1:137

K\_CY1(i, 1) = CY\_Wave1(i)\*(1+CYexp1(i))/CYexp1(i);

end

plot(K\_CY1);

title("K for Wave 1 CY");

xlabel("Days since wave beginning");

ylabel("Carrying Capacity");

%Used value for k

myK\_CY1 = K\_CY1(137); %K(137) first and then change this for optimum

%ln(P/(K-P)) graph

set(0,'DefaultLegendAutoUpdate','off')

figure;

myLn\_CY1 = zeros(137, 1);

for i = 1:137

myLn\_CY1(i, 1) = log(abs(CY\_Wave1(i)/(myK\_CY1-CY\_Wave1(i))));

end

plot(myLn\_CY1);

hold on;

mdl1b\_CY = fitlm(days1\_CY(1:end), myLn\_CY1(1:end));  
lnR\_CY1 = table2array(mdl1b\_CY.Coefficients(2, 1));  
lnA\_CY1 = table2array(mdl1b\_CY.Coefficients(1, 1));

art\_CY1 = zeros(137, 1);

for i = 1:137

art\_CY1(i, 1) = lnA\_CY1+(lnR\_CY1\*(i));

end

plot(art\_CY1)

hold off;

title("ln(P/(K-P)) graph for Wave 1 CY");

xlabel("Days since wave beginning");

ylabel("ln(P/(K-P)");

legend('Observed','Predicted');

% % Normalised and Logistical model

% Normalised and Logistical model

figure;

hold on;

expart\_CY1 = exp(art\_CY1);

logistic\_CY1 = zeros(137, 1);

for i = 1:137

  logistic\_CY1(i, 1) = (myK\_CY1\*expart\_CY1(i))/(1+expart\_CY1(i));

end

plot(CY\_Wave1);

plot(logistic\_CY1);

title("Logistical model prediction for Wave 1 CY");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

legend('observed','Predicted')

hold off;

% Error

Error\_CY1 = CY\_Wave1 - logistic\_CY1;

SumError\_CY1 = sum(Error\_CY1);

SumSQError\_CY1 = sumsqr(Error\_CY1);

Diff\_CY1 = SumError\_CY1 - SumSQError\_CY1;

**Appendix 2.3 – Wave 2 (CY):**

%Wave 2

CY\_Wave2 = zeros(210, 1);

for i = 1:210

CY\_Wave2(i, 1) = CYCC(i+137,1);

CY\_Wave2(i, 1) = CY\_Wave2(i, 1) - 1291;

CY\_Wave2(i, 1) = CY\_Wave2(i, 1)./1207359;

end

CY\_Wave2 = N\_CY(138:348,1);

%Plotting normalised

figure;

plot(CY\_Wave2);

title("Wave 2 of Normalised Cumulative Cases for Cyprus");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

%Plotting log wave

figure;

logwave2\_CY = log(CY\_Wave2);

plot(logwave2\_CY);

title("Logarithm Plot for Wave 2 of Normalised Cumulative Cases for Cyprus");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

%Plotting normalised wave

figure;

plot(CY\_Wave2);

title("Wave 2 of Normalised Cumulative Cases for Cyprus");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

%Plotting log wave

figure;

logwave2\_CY = log(CY\_Wave2);

plot(logwave2\_CY);

title("Logarithm Plot for Wave 2 of Normalised Cumulative Cases for Cyprus");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

% a and r

mdl2a\_CY = fitlm(71:130, log(CYCC2(71:130)));

r\_CY2 = table2array(mdl2a\_CY.Coefficients(2, 1));

a\_CY2 = table2array(mdl2a\_CY.Coefficients(1, 1));

% Exponential model prediction

figure;

hold on;

plot(CY\_Wave2);

CYexp2 = zeros(210, 1);

for i = 1:210

 CYexp2(i, 1) = exp(a\_CY2+(r\_CY2.\*(i))); % exp

end

plot(CYexp2);

ylim([0 0.01]);

hold off;

title("CY wave 2 exp model prediction");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

legend('Observed','Predicted');

% Error

figure;

Err\_CY2 = zeros(210, 1);

for i = 1:210

Err\_CY2(i, 1) = CY\_Wave2(i)-CYexp2(i); % need an array, not a matrix

end

plot(Err\_CY2);

title("Error for Wave 2 CY");

xlabel("Days since wave beginning");

ylabel("Error of Cumulative Fraction");

% K

figure;

K\_CY2 = zeros(210, 1);

for i = 1:210

K\_CY2(i, 1) = CY\_Wave2(i)\*(1+CYexp2(i))/CYexp2(i);

end

plot(K\_CY2);

title("K for Wave 2 CY");

xlabel("Days since wave beginning");

ylabel("Carrying Capacity");

%  %Used value for k

%

myK\_CY2 = K\_CY1(210); %K(210) first and then change this for optimum

%ln(P/(K-P)) graph

figure;

myLn\_CY2 = zeros(210, 1);

for i = 1:210

myLn\_CY2(i, 1) = log(abs(CY\_Wave2(i)/(myK\_CY2-CY\_Wave2(i))));

end

plot(myLn\_CY2);

hold on;

% title("ln(P/(K-P)) graph for Wave 2 CY");

%

% xlabel("Days since wave beginning");

%

% ylabel("ln(P/(K-P)");

% legend('Observed','Predicted');

mdl2b\_CY = fitlm(days2\_CY(1:end), myLn\_CY2(1:end));  
lnR\_CY2 = table2array(mdl2b\_CY.Coefficients(2, 1));  
lnA\_CY2 = table2array(mdl2b\_CY.Coefficients(1, 1));

art\_CY2 = zeros(210, 1);

for i = 1:210

art\_CY2(i, 1) = lnA\_CY2+(lnR\_CY2\*(i));

end

plot(art\_CY2)

hold off;

title("ln(P/(K-P)) graph for Wave 2 CY");

xlabel("Days since wave beginning");

ylabel("ln(P/(K-P)");

legend('Observed','Predicted');

% Normalised and Logistical model

figure;

hold on;

expart\_CY2 = exp(art\_CY2);

logistic\_CY2 = zeros(210, 1);

for i = 1:210

logistic\_CY2(i, 1) = (myK\_CY2.\*expart\_CY2(i))./(1+expart\_CY2(i));

end

plot(CY\_Wave2);

plot(logistic\_CY2);

title("Normalised and Logistical model for Wave 2 CY");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

legend('Observed','Predicted');

% ylim([0 0.006]);

hold off;

% Error

myError\_CY2 = CY\_Wave2 - logistic\_CY2;

SumError\_CY2 = sum(myError\_CY2);

SumSQError\_CY2 = sumsqr(myError\_CY2);

Diff\_CY2 = SumError\_CY2 - SumSQError\_CY2;

**Appendix 2.4 – Wave 3 (CY):**

%Wave 3

CY\_Wave3 = zeros(71, 1);

for i = 1:71

CY\_Wave3(i, 1) = CYCC(i+348,1);

CY\_Wave3(i, 1) = CY\_Wave3(i, 1) - 37880;

CY\_Wave3(i, 1) = CY\_Wave3(i, 1)./1207359;

end

CY\_Wave3 = N\_CY(349:420,1);

%Plotting normalised wave

figure;

plot(CY\_Wave3);

title("Wave 3 of Normalised Cumulative Cases for Cyprus");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

%Plotting log wave

figure;

logwave3\_CY = log(CY\_Wave3);

plot(logwave3\_CY);

title("Logarithm Plot for Wave 3 of Normalised Cumulative Cases for Cyprus");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

% a and r

mdl3a\_CY = fitlm(1:20, log(CYCC3(1:20)));

r\_CY3 = table2array(mdl3a\_CY.Coefficients(2, 1));

a\_CY3 = table2array(mdl3a\_CY.Coefficients(1, 1));

% Exponential model prediction

figure;

hold on;

plot(CY\_Wave3);

CYexp3 = zeros(71, 1);

for i = 1:71

 CYexp3(i, 1) = exp(a\_CY3+(r\_CY3.\*(i))); % exp

end

plot(CYexp3);

hold off;

title("CY wave 3 exp model prediction");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

legend('Observed','Predicted');

% Error

figure;

Err\_CY3 = zeros(71, 1);

for i = 1:71

Err\_CY3(i, 1) = CY\_Wave3(i)- CYexp3(i); % need an array, not a matrix

end

plot(Err\_CY3);

title("Error for Wave 3 CY");

xlabel("Days since wave beginning");

ylabel("Error of Cumulative Fraction");

% K

figure;

K\_CY3 = zeros(71, 1);

for i = 1:71

K\_CY3(i, 1) = CY\_Wave3(i)\*(1+CYexp3(i))/CYexp3(i);

end

plot(K\_CY3);

title("K for Wave 3 CY");

xlabel("Days since wave beginning");

ylabel("Carrying Capacity");

% %  %Used value for k

% %

myK\_CY3 = K\_CY3(71); %K(71) first and then change this for optimum

ln(P/(K-P)) graph

set(0,'DefaultLegendAutoUpdate','off')

figure;

myLn\_CY3 = zeros(71, 1);

for i = 1:71

myLn\_CY3(i, 1) = log(abs(CY\_Wave3(i)/(myK\_CY3-CY\_Wave3(i))));

end

plot(myLn\_CY3);

hold on;

mdl3b\_CY = fitlm(days3\_CY(1:end), myLn\_CY3(1:end));  
lnR\_CY3 = table2array(mdl3b\_CY.Coefficients(2, 1));  
lnA\_CY3 = table2array(mdl3b\_CY.Coefficients(1, 1));

art\_CY3 = zeros(71, 1);

for i = 1:71

art\_CY3(i, 1) = lnA\_CY3+(lnR\_CY3\*(i));

end

plot(art\_CY3)

hold off;

title("ln(P/(K-P)) graph for Wave 3 CY");

xlabel("Days since wave beginning");

ylabel("ln(P/(K-P)");

legend('Observed','Predicted');

Normalised and Logistical model

figure;

hold on;

expart\_CY3 = exp(art\_CY3);

logistic\_CY3 = zeros(71, 1);

for i = 1:71

logistic\_CY3(i, 1) = (myK\_CY3.\*expart\_CY3(i))./(1+expart\_CY3(i));

end

plot(CY\_Wave3);

plot(logistic\_CY3);

title("Normalised and Logistical model for Wave 3 CY");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

legend('Observed','Predicted');

ylim([0.030 0.05]);

hold off;

%Error

myError\_CY3 = CY\_Wave3 - logistic\_CY3;

SumError\_CY3 = sum(myError\_CY3);

SumSQError\_CY3= sumsqr(myError\_CY3);

Diff\_CY3 = SumError\_CY3 - SumSQError\_CY3;

**Appendix 2.5 – Wave 4 (CY):**

% Wave 4

CY\_Wave4 = zeros(119, 1);

for i = 1:119

CY\_Wave4(i, 1) = CYCC(i+420,1);

CY\_Wave4(i, 1) = CY\_Wave4(i, 1) - 35032;

CY\_Wave4(i, 1) = CY\_Wave4(i, 1)./1207359;

end

CY\_Wave4 = N\_CY(421:540,1);

%Plotting normalised wave

figure;

plot(CY\_Wave4);

title("Wave 4 of Normalised Cumulative Cases for Cyprus");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

%Plotting log wave

figure;

logwave4\_CY = log(CY\_Wave4);

plot(logwave4\_CY);

title("Logarithm Plot for Wave 4 of Normalised Cumulative Cases for Cyprus");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

% a and r

mdl4a\_CY = fitlm(1:18, log(CYCC1(1:18)));

r\_CY4 = table2array(mdl4a\_CY.Coefficients(2, 1));

a\_CY4 = table2array(mdl4a\_CY.Coefficients(1, 1));

% Exponential model prediction

figure;

hold on;

plot(CY\_Wave4);

CYexp4 = zeros(119, 1);

for i = 1:119

 CYexp4(i, 1) = exp(a\_CY4+(r\_CY4.\*(i))); % exp

end

plot(CYexp4);

hold off;

title("CY wave 4 exp model prediction");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

legend('Observed','Predicted');

ylim([0.06,0.065])

% Error

figure;

Err\_CY4 = zeros(119, 1);

for i = 1:119

 Err\_CY4(i, 1) = CY\_Wave4(i)-CYexp4(i); % need an array, not a matrix

end

plot(Err\_CY4);

title("Error for Wave 4 CY");

xlabel("Days since wave beginning");

ylabel("Error of Cumulative Fraction");

% K

figure;

K\_CY4 = zeros(119, 1);

for i = 1:119

 K\_CY4(i, 1) = CY\_Wave4(i)\*(1+CYexp4(i))/CYexp4(i);

end

plot(K\_CY4);

title("K for Wave 4 CY");

xlabel("Days since wave beginning");

ylabel("Carrying Capacity");

% %  %Used value for k

myK\_CY4 = K\_CY4(119); %K(119) first and then change this for optimum

%ln(P/(K-P)) graph

figure;

myLn\_CY4 = zeros(119, 1);

for i = 1:119

myLn\_CY4(i, 1) = log(abs(CY\_Wave4(i)/(myK\_CY4-CY\_Wave4(i))));

end

plot(myLn\_CY4);

hold on;

mdl4b\_CY = fitlm(days4\_CY(1:end), myLn\_CY4(1:end));  
lnR\_CY4 = table2array(mdl4b\_CY.Coefficients(2, 1));  
lnA\_CY4 = table2array(mdl4b\_CY.Coefficients(1, 1));

art\_CY4 = zeros(119, 1);

for i = 1:119

art\_CY4(i, 1) = lnA\_CY4+(lnR\_CY4\*(i));

end

plot(art\_CY4)

hold off;

title("ln(P/(K-P)) graph for Wave 4 CY");

xlabel("Days since wave beginning");

ylabel("ln(P/(K-P)");

legend('Observed','Predicted');

% Normalised and Logistical model

figure;

hold on;

expart\_CY4 = exp(art\_CY4);

logistic\_CY4 = zeros(119, 1);

for i = 1:119

logistic\_CY4(i, 1) = (myK\_CY4.\*expart\_CY4(i))./(1+expart\_CY4(i));

end

plot(CY\_Wave4);

plot(logistic\_CY4);

title("Normalised and Logistical model for Wave 4 CY");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

legend('Observed','Predicted');

hold off;

% Error

myError\_CY4 = CY\_Wave4 - logistic\_CY4;

SumError\_CY4 = sum(myError\_CY4);

SumSQError\_CY4= sumsqr(myError\_CY4);

Diff\_CY4 = SumError\_CY4 - SumSQError\_CY4;

**Appendix 2.6 – Initial Conditions for SIR Model (CY):**

%% Defining our initial conditions

R0 = 58/Population\_CY

I0 = N\_CY(1) - R0;

P0 = (I0 + R0);

S0 = 1 - P0;

**Appendix 2.7 – Parameters estimation for SIR Model (CY):**

%% Parameters estimation

from = 20; % 1 20

to = 35; % 19 35 77$

waveEnd = 77; % my wave 1 end

b = 0.1; % my chosen value for b as covid is 10 days

% Calculating my linear regression variables

mdl = fitlm(from:to, log(N\_CY(from:to)));

r = table2array(mdl.Coefficients(2, 1));

c = table2array(mdl.Coefficients(1, 1)); % what used to be a

% The last coefficient

a = r + b;

% Integrate system of ODE

[t, y] = ode45(@(t, y) sir(t, y, a, b), 1:length(N\_CY), [S0, R0]);

% Accuracy estimation

p = 1 - y(:, 1);

MSE = sum((N\_CY(1:waveEnd) - p(1:waveEnd)) .^ 2) / waveEnd

% Draw figure

figure;

plot(N\_CY);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

**Command Window (for this section):**

>> S19 = 1 - (N\_CY(19) + N\_CY(9))

S19 =  
0.9990  
>>S35 = 1 - (N\_CY(35) + N\_CY(25))

S35 =  
0.9986

>>R0 = R0 = 58/Population\_CY

R0 =  
 4.8039e-05

>> S0=S19

S0 =  
0.999006095121666

**Appendix 2.8 – Testing initial conditions – I(0) (CY):**

%% Play with original conditions

figure

plot(N\_CY);

hold on;

lims = ylim();

labs = {'Observed'};

I00 = I0 / 2;

for k = 1:5

S0 = 1 - I00 - R0;

[t, y] = ode45(@(t, y) sir(t, y, a, b), 1:length(N\_CY), [S0, R0]);

p = 1 - y(:, 1);

plot(p);

MSE = sum((N\_CY(from:to) - p(from:to)) .^ 2) / (to-from)

labs = [labs, sprintf("I(0) = %.4g", I00)];

I00 = I00 \* 2;

end

legend(labs, 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

title("Observing different values for I(0) for CY")

**Appendix 2.9 – Testing initial conditions – I(0) and a (CY):**

%% Play with original conditions with optimisation

figure

plot(N\_CY);

hold on;

lims = ylim();

labs = {'Observed'};

I00 = I0 / 2;

for k = 1:5

S0 = 1 - I00 - R0;

aOpt = fminbnd(@(x) forSearch(x, length(N\_CY), b, [S0, R0], N\_CY), b, 3 \* b);

[t, y] = ode45(@(t, y) sir(t, y, aOpt, b), 1:length(N\_CY), [S0, R0]);

p = 1 - y(:, 1);

plot(p);

MSE = sum((N\_CY(from:to) - p(from:to)) .^ 2) / (to-from)

labs = [labs, sprintf("I(0) = %.4g, a = %.4f", I00, aOpt)];

I00 = I00 \* 2;

end

legend(labs, 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

title("Observing different values for I(0) for CY with varying values of a");

**Appendix 2.10 – Essential functions for Appendix 2.7-2.9 (CY):**

% functions

function dydt = sir(t, y, a, b)

I = 1 - sum(y);

dydt = [ - a \* y(1) \* I;

b \* I];

end

function err = forSearch(a, wEnd, b, y0, N\_CY)

[t, y] = ode45(@(t, y) sir(t, y, a, b), 1:wEnd, y0);

p = 1 - y(:, 1);

err = sum((N\_CY(1:wEnd) - p(1:wEnd)) .^ 2) / wEnd;

end

**Appendix 2.11 – Initial Conditions for SIRPs Model (CY):**

%% Initial Conditions

R0 = 58/Population\_CY;

I0 = 0.00012920;

P0 = (I0 + R0);

S0 = 1 - P0;

**Appendix 2.12 – Parameters estimation for SIRPs Model (CY):**

%% Parameters estimation

from = 1; %

to = 540; %

b = 0.1; % my chosen value for b as covid is 10 days

% Linear regression

mdl = fitlm(from:to, log(N\_CY(from:to)));

r = table2array(mdl.Coefficients(2, 1));

c = table2array(mdl.Coefficients(1, 1));

% The last coefficient

a=0.1094;

% Define my coefficients

k2 = 1;

k3 = 1/50;

k6 = 1/100;

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_CY(t,y,a,b,k2,k3,k6), 1:length(N\_CY), [S0, 0, I0, R0]);

% Accuracy estimation

p = y(:,3) +y(:,4); % I + R as defined previously

MSE = sum((N\_CY(from:to) - p(from:to)).^2)/(to-from)

figure;

plot(N\_CY);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

**Appendix 2.13 – Testing initial conditions – K2, K3 and K6 (CY):**

%% Modifications to the values of k2 and applying:

% Define my coefficients

k2 = 1/4;

k3 = 1/50; % after 50 days in "Resistance" state people become tired

k6 = 1/100; % after 100 days in "Exhaustion" state people return to the initial "Ignorant" state and become sensitive to the alarm signals

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_CY(t,y,a,b,k2,k3,k6), 1:length(N\_CY), [S0, 0, I0, R0]);

% Accuracy estimation

p = y(:,3) +y(:,4); % I + R as defined previously

MSE = sum((N\_CY(from:to) - p(from:to)).^2)/(to-from)

figure;

plot(N\_CY);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

title('SIRPs model with modified constants for CY')

%% Modifications to the values of k3 and applying

% Define my coefficients

k2 = 1;

k3 = 1/10; % after 10 days in "Resistance" state people become tired

k6 = 1/100; % after 1100 days in "Exhaustion" state people return to the initial "Ignorant" state and become sensitive to the alarm signals

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_CY(t,y,a,b,k2,k3,k6), 1:length(N\_CY), [S0, 0, I0, R0]);

% Accuracy estimation

p = y(:,3) +y(:,4); % I + R as defined previously

MSE = sum((N\_CY(from:to) - p(from:to)).^2)/(to-from)

figure;

plot(N\_CY);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

title('SIRPs model with modified constants for CY')

%% Modifications to the values of k6 and applying:

% Define my coefficients

k2 = 1;

k3 = 1/50; % after 50 days in "Resistance" state people become tired

k6 = 1/200; % after 100 days in "Exhaustion" state people return to the initial "Ignorant" state and become sensitive to the alarm signals

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_CY(t,y,a,b,k2,k3,k6), 1:length(N\_CY), [S0, 0, I0, R0]);

% Accuracy estimation

p = y(:,3) +y(:,4); % I + R as defined previously

MSE = sum((N\_CY(from:to) - p(from:to)).^2)/(to-from)

figure;

plot(N\_CY);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

title('SIRPs model with modified constants for CY')

%% SIRPs Model with the modifications to K2, K3 and K6:

% Define my coefficients

k2 = 1/4;

k3 = 1/10; % after 50 days in "Resistance" state people become tired

k6 = 1/200; % after 100 days in "Exhaustion" state people return to the initial "Ignorant" state and become sensitive to the alarm signals

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_CY(t,y,a,b,k2,k3,k6), 1:length(N\_CY), [S0, 0, I0, R0]);

% Accuracy estimation

p = y(:,3) +y(:,4); % I + R as defined previously

MSE = sum((N\_CY(from:to) - p(from:to)).^2)/(to-from)

figure;

plot(N\_CY);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

title('SIRPs model with modified constants for CY')

%% SIRPs Model with the modifications b:

%Define parameters

R0 = 0.0094;

I0 = 0.00012920;

P0 = (I0 + R0);

S0 = 1 - P0;

b = 0.1; % 0.15 %0.08

% Define my coefficients

k2 = 1;

k3 = 1/50;

k6 = 1/100;

Ip=0.1

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_CY(t,y,a,b,k2,k3,k6,Ip), 1:length(N\_CY), [S0, 0, I0, R0]);

p = y(:,3) +y(:,4); % I + R as defined previously

MSE = sum((N\_CY(from:to) - p(from:to)).^2)/(to-from)

figure;

plot(N\_CY);

hold on;

plot(p);

hold off;

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

legend('Observed','Predicted')

title('SIRPs model with modified constant Ip for CY')

**Appendix 2.14 – Applying Crowd effect to SIRPs model (CY):**

% Parameters estimation

from = 1; %

to = 540; %

%Define parameters

R0 = 0.0094;

I0 = 0.00012920;

P0 = (I0 + R0);

S0 = 1 - P0;

b = 0.1;

% Define my coefficients

k2 = 1/4;

k3 = 1/10;

k6 = 1/200;

Ip=0.1

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_CY(t,y,a,b,k2,k3,k6,Ip), 1:length(N\_CY), [S0, 0, I0, R0]);

p = y(:,3) +y(:,4); % I + R as defined previously

MSE = sum((N\_CY(from:to) - p(from:to)).^2)/(to-from)

figure;

plot(N\_CY);

hold on;

plot(p);

hold off;

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

legend('Observed','Predicted')

title('SIRPs model with modified constant Ip for CY')

**Appendix 2.15 – Essential functions for Appendix 2.12-2.14 (TT):**

end

%% function without Ip (no Crowd Effect

function dydt = sirPs\_CY(t,y,a,b,k2,k3,k6)

% y(1)=S\_ign(t)

% y(2) = S\_res(t)

% y(3) = I(t)

% y(4) = R(t) k

S\_exh = 1 - sum(y);

a1 = -a\*y(1)\*y(3) - k2\*y(1)\*y(3) + k6\*S\_exh;

a2 = k2\*y(1)\*y(3)-k3\*y(2);

a3 = a\*y(1)\*y(3) + a\*S\_exh\*y(3) -b\*y(3);

a4 = b\*y(3);

dydt = [ a1; a2; a3; a4];

end

%% function with crowd effect

function dydt = sirPs\_CY(t,y,a,b,k2,k3,k6,Ip)

% y(1)=S\_ign(t)

% y(2) = S\_res(t)

% y(3) = I(t)

% y(4) = R(t) k

S\_exh = 1 - sum(y);

a1 = -a\*y(1)\*y(3) - k2\*y(1)\*y(3).^2 + k6\*S\_exh;

a2 = (k2/Ip)\*y(1)\*y(3).^2-k3\*y(2);

a3 = a\*y(1)\*y(3) + a\*S\_exh\*y(3) -b\*y(3);

a4 = b\*y(3);

dydt = [ a1; a2; a3; a4];

**Appendix 3.1 – Creating the variables for the coming figures (SA):**

% Creating the waves in accordance to above intervals identified in Appendix 1.5

SACC1 = N\_SA(1:265);

days1\_SA = 1:265;

SACC2 = SACC(266:552);

SACC2 = SACC2 - 358713;

SACC2 = SACC2/Population\_SA;

days2\_SA = 1:287;

**Appendix 3.2 – Wave 1 (SA):**

%Wave 1

SA\_Wave1 = N\_SA(1:265,1);

%Plotting normalised wave

figure;

plot(SA\_Wave1);

title("Wave 1 of Normalised Cumulative Cases for Saudi Arabia");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

%Plotting log wave

figure;

logwave1\_SA = log(SA\_Wave1);

plot(logwave1\_SA);

title("Logarithm Plot for Wave 1 of Normalised Cumulative Cases for Saudi Arabia");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

% a and r

mdl1a\_SA = fitlm(8:68, log(SACC1(8:68)));  
r\_SA1 = table2array(mdl1a\_SA.Coefficients(2, 1));  
a\_SA1 = table2array(mdl1a\_SA.Coefficients(1, 1));  
% Exponential model prediction  
figure;  
hold on;  
SA\_Wave1 = N\_SA(1:265,1); %normalised

plot(SA\_Wave1);

SAexp1 = zeros(265, 1);

for i = 1:265

 SAexp1(i, 1) = exp(a\_SA1+(r\_SA1.\*(i))); % exp

end

plot(SAexp1);

ylim([0 0.012]);

hold off;

title("SA Wave 1 Exp Model Prediction");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

legend('Observed','Predicted');

% Error

figure;

Err\_SA1 = zeros(265, 1);

for i = 1:265

   Err\_SA1(i, 1) = SA\_Wave1(i)-SAexp1(i); % need an array, not a matrix

end

plot(Err\_SA1);

title("Error for Wave 1 SA");

xlabel("Days since wave beginning");

ylabel("Error of Cumulative Fraction");

% K

figure;

K\_SA1 = zeros(265, 1);

for i = 1:265

   K\_SA1(i) = SA\_Wave1(i)\*((1+SAexp1(i))/SAexp1(i));

end

plot(K\_SA1);

title("K for Wave 1 SA");

xlabel("Days since wave beginning");

ylabel("Carrying Capacity");

%Used value for k

myK\_SA1 = K\_SA1(265); %K(265) first and then change this for optimum

%ln(P/(K-P)) graph

figure;

hold on;

myLn\_SA1 = zeros(265, 1);

for i = 1:265

 myLn\_SA1(i, 1) = log(abs((SA\_Wave1(i)/(myK\_SA1-SA\_Wave1(i)))));

end

plot(myLn\_SA1);

title("ln(P/(K-P)) graph for Wave 1 SA");

xlabel("Days since wave beginning");

ylabel("ln(P/(K-P)");

mdl1b\_SA = fitlm(days1\_SA(1:end), myLn\_SA1(1:end));  
lnR\_SA1 = table2array(mdl1b\_SA.Coefficients(2, 1));  
lnA\_SA1 = table2array(mdl1b\_SA.Coefficients(1, 1));  
art\_SA1 = zeros(265, 1);

for i = 1:265

 art\_SA1(i, 1) = lnA\_SA1+(lnR\_SA1\*(i));

end

plot(art\_SA1)

hold off;

% Normalised and Logistical model

figure;

hold on;

expart\_SA1 = exp(art\_SA1);

logistic\_SA1 = zeros(265, 1);

for i = 1:265

   logistic\_SA1(i, 1) = (myK\_SA1\*expart\_SA1(i))/(1+expart\_SA1(i));

end

plot(SA\_Wave1);

plot(logistic\_SA1);

title("Logistical model prediction for Wave 1 SA");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

hold off;

% Error

Error\_SA1 = SA\_Wave1- logistic\_SA1;

SumError\_SA1 = sum(Error\_SA1(1:265));

SumSQError\_SA1 = sumsqr(Error\_SA1(1:265));

Diff\_SA1 = SumError\_SA1 - SumSQError\_SA1;

**Appendix 3.3 – Wave 2 (SA):**

%Wave 2

mySA\_Wave2 = zeros(287, 1);

for i = 1:287

   mySA\_Wave2(i, 1) = SACC(i+265, 1);

   mySA\_Wave2(i, 1) = mySA\_Wave2(i, 1)-358713;

end

for i = 1:287

   SA\_Wave2(i, 1) = mySA\_Wave2(i, 1)./34813871;

end

%Plotting normalised wave

figure;

plot(SA\_Wave2);

title("Wave 2 of Normalised Cumulative Cases for Saudi Arabia");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

%Plotting log wave

figure;

logwave2\_SA = log(SA\_Wave2);

plot(logwave2\_SA);

title("Logarithm Plot for Wave 2 of Normalised Cumulative Cases for Saudi Arabia");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

% a and r   
mdl2a\_SA = fitlm(2:18, log(SACC2(2:18)));  
r\_SA2 = table2array(mdl2a\_SA.Coefficients(2, 1));  
a\_SA2 = table2array(mdl2a\_SA.Coefficients(1, 1));  
% Exponential model prediction  
figure;

hold on;

plot(SA\_Wave2);

SAexp2 = zeros(287, 1);

for i = 1:287

  SAexp2(i, 1) = exp(a\_SA2+(r\_SA2.\*(i))); % exp

end

plot(SAexp2);

ylim([0 0.001]);

hold off;

title("Normalised and exp for Wave 2 SA");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

legend('Observed','Predicted');

% Error

figure;

Err\_SA2 = zeros(287, 1);

for i = 1:287

 Err\_SA2(i, 1) = SA\_Wave2(i)-SAexp2(i); % need an array, not a matrix

end

plot(Err\_SA2);

title("Error for Wave 2 SA");

xlabel("Days since wave beginning");

ylabel("Error of Cumulative Fraction");

% K

figure;

K\_SA2 = zeros(287, 1);

for i = 1:287

 K\_SA2(i, 1) = SA\_Wave2(i)\*(1+SAexp2(i))/SAexp2(i);

end

plot(K\_SA2);

title("K for Wave 2 SA");

xlabel("Days since wave beginning");

ylabel("Carrying Capacity");

%Used value for k

myK\_SA2 = K\_SA2(287); %K(267) first and then change this for optimum

%ln(P/(K-P)) graph

figure;

hold on;

myLn\_SA2 = zeros(287, 1);

for i = 1:287

 myLn\_SA2(i, 1) = log(abs(SA\_Wave2(i)/(myK\_SA2-SA\_Wave2(i))));

end

plot(myLn\_SA2);

title("ln(P/(K-P)) graph for Wave 2 SA");

xlabel("Days since wave beginning");

ylabel("ln(P/(K-P)");

mdl2b\_SA = fitlm(days2\_SA(1:end), myLn\_SA2(1:end));  
lnR\_SA2 = table2array(mdl2b\_SA.Coefficients(2, 1));  
lnA\_SA2 = table2array(mdl2b\_SA.Coefficients(1, 1));  
art\_SA2 = zeros(287, 1);

for i = 1:287

 art\_SA2(i, 1) = lnA\_SA2+(lnR\_SA2\*(i));

end

plot(art\_SA2)

hold off;

% Normalised and Logistical model

figure;

hold on;

expart\_SA2 = exp(art\_SA2);

logistic\_SA2 = zeros(287, 1);

for i = 1:287

 logistic\_SA2(i, 1) = (myK\_SA2.\*expart\_SA2(i))./(1+expart\_SA2(i));

end

plot(SA\_Wave2);

plot(logistic\_SA2);

title("Normalised and Logistical model for Wave 2 SA");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

hold off;

% Error

myError\_SA2 = SA\_Wave2 - logistic\_SA2;

SumError\_SA2 = sum(myError\_SA2);

SumSQError\_SA2 = sumsqr(myError\_SA2);

Diff\_SA2 = SumError\_SA2 - SumSQError\_SA2;

**Appendix 3.4 – Initial Conditions for SIR Model (SA):**

%% Defining our initial conditions

R0 = 15/Population\_SA;

I0 = N\_SA(1) - R0;

P0 = (I0 + R0);

S0 = 1 - P0;

**Appendix 3.5 – Parameters estimation for SIR Model (SA):**

%% Parameters estimation

from = 1; % 1 76

to = 552; % 75 128 552

waveEnd = 265; % my wave 1 end

b = 0.1; % my chosen value for b as covid is 10 days

% Calculating my linear regression variables

mdl = fitlm(from:to, log(N\_SA(from:to)));

r = table2array(mdl.Coefficients(2, 1));

c = table2array(mdl.Coefficients(1, 1)); % what used to be a

% The last coefficient

a = r + b;

% Integrate system of ODE

[t, y] = ode45(@(t, y) sir(t, y, a, b), 1:length(N\_SA), [S0, R0]);

% Accuracy estimation

p = 1 - y(:, 1);

MSE = sum((N\_SA(1:waveEnd) - p(1:waveEnd)) .^ 2) / waveEnd

% Draw figure

figure;

plot(N\_SA);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

**Command Window (for this section):**

>> S0

>> S128 = 1 - (N\_SA(128)-N\_SA(118))

>> SA(66)

>>R0 = 15/ 34813871

R0 =  
 4.3086e-07

>> N\_SA(1)

ans =  
 6.9513e-06

>> I0 = N\_SA(1)- R0

I0 =  
 6.5204e-06

**Appendix 3.6 – Testing initial conditions – I(0) (SA):**

%% Play with original conditions

% Remove semi-colons next to MSE to identify MSE values for each iteration.

figure

plot(N\_SA);

hold on;

lims = ylim();

labs = {'Observed'};

I00 = I0 / 2;

for k = 1:5

S0 = 1 - I00 - R0;

[t, y] = ode45(@(t, y) sir(t, y, a, b), 1:length(N\_SA), [S0, R0]);

p = 1 - y(:, 1);

plot(p);

MSE = sum((N\_SA(from:to) - p(from:to)) .^ 2) / (to-from)

labs = [labs, sprintf("I(0) = %.4g", I00)];

I00 = I00 \* 2;

end

legend(labs, 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

**Appendix 3.7 – Testing initial conditions – I(0) and a (SA):**

%% Play with original conditions with optimisation

% Remove semi-colons next to MSE to identify MSE values for each iteration.

figure

plot(N\_SA);

hold on;

lims = ylim();

labs = {'Observed'};

I00 = I0 / 2;

for k = 1:5

S0 = 1 - I00 - R0;

aOpt = fminbnd(@(x) forSearch(x, length(N\_SA), b, [S0, R0], N\_SA), b, 3 \* b);

[t, y] = ode45(@(t, y) sir(t, y, aOpt, b), 1:length(N\_SA), [S0, R0]);

p = 1 - y(:, 1);

plot(p);

MSE = sum((N\_SA(from:to) - p(from:to)) .^ 2) / (to-from)

labs = [labs, sprintf("I(0) = %.4g, a = %.4f", I00, aOpt)];

I00 = I00 \* 2;

end

legend(labs, 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

**Appendix 3.8 – Essential functions for Appendix 3.5-3.7 (SA):**

% functions

function dydt = sir(t, y, a, b)

I = 1 - sum(y);

dydt = [ - a \* y(1) \* I;

b \* I];

end

function err = forSearch(a, wEnd, b, y0, N\_SA)

[t, y] = ode45(@(t, y) sir(t, y, a, b), 1:wEnd, y0);

p = 1 - y(:, 1);

err = sum((N\_SA(1:wEnd) - p(1:wEnd)) .^ 2) / wEnd;

end

**Appendix 3.9 – Initial Conditions for SIRPs Model (SA):**

%% Defining our initial conditions

R0 = 15/Population\_SA;

I0 = 0.00000652; % N\_SA(1) - R0; %6.52x10-6

P0 = (I0 + R0);

S0 = 1 - P0;

**Appendix 3.10 – Parameters estimation for SIRPs Model (SA):**

%% Parameters estimation

from = 1; % 1 76

to = 128; % 75 128

b = 0.1; % my chosen value for b as covid is 10 days

% Calculating my linear regression variables

mdl = fitlm(from:to, log(N\_SA(from:to)));

r = table2array(mdl.Coefficients(2, 1));

c = table2array(mdl.Coefficients(1, 1)); % what used to be a

% The optimum coefficient

%a = r + b;

a = 0.1111;

% Define my coefficients

k2 = 1;

k3 = 1/50;

k6 = 1/100;

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_SA(t,y,a,b,k2,k3,k6), 1:length(N\_SA), [S0, 0, I0, R0]);

% Accuracy estimation

MSE = sum((N\_SA(from:to) - p(from:to)).^2)/(to-from)

% Draw figure

figure;

plot(N\_SA);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

title("Social Psychology additions to SIR models for SA")

**Appendix 3.11 – Testing initial conditions – K2, K3 and K6 (SA):**

%% Modifications to the values of k2 and applying:

% Define my coefficients

k2 = 1/2;

k3 = 1/50;

k6 = 1/100;

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_SA(t,y,a,b,k2,k3,k6), 1:length(N\_SA), [S0, 0, I0, R0]);

% Accuracy estimation

p = y(:,3) +y(:,4); % I + R as defined previously

MSE = sum((N\_SA(from:to) - p(from:to)).^2)/(to-from)

% Draw figure

figure;

plot(N\_SA);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

title("SIRPs model with modified constants for SA")

%% Modifications to the values of k3 and applying

% Define my coefficients

k2 = 1;

k3 = 1/100;

k6 = 1/100;

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_SA(t,y,a,b,k2,k3,k6), 1:length(N\_SA), [S0, 0, I0, R0]);

% Accuracy estimation

p = y(:,3) +y(:,4); % I + R as defined previously

MSE = sum((N\_SA(from:to) - p(from:to)).^2)/(to-from)

% Draw figure

figure;

plot(N\_SA);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

title("SIRPs model with modified constants for SA")

%% Modifications to the values of k6 and applying:

% Define my coefficients

k2 = 1;

k3 = 1/50;

k6 = 1/200;

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_SA(t,y,a,b,k2,k3,k6), 1:length(N\_SA), [S0, 0, I0, R0]);

% Accuracy estimation

p = y(:,3) +y(:,4); % I + R as defined previously

MSE = sum((N\_SA(from:to) - p(from:to)).^2)/(to-from)

% Draw figure

figure;

plot(N\_SA);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

title("SIRPs model with modified constants for SA")

%% SIRPs Model with the modifications to K2, K3 and K6:

% Define my coefficients

% Define my coefficients

k2 = 1/2;

k3 = 1/17;

k6 = 1/300;

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_SA(t,y,a,b,k2,k3,k6), 1:length(N\_SA), [S0, 0, I0, R0]);

% Accuracy estimation

p = y(:,3) +y(:,4); % I + R as defined previously

MSE = sum((N\_SA(from:to) - p(from:to)).^2)/(to-from)

% Draw figure

figure;

plot(N\_SA);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

title("SIRPs model with modified constants for SA")

%% SIRPs Model with the modifications b:

%Define parameters

R0 = 0.0094;

I0 = 0.00012920;

P0 = (I0 + R0);

S0 = 1 - P0;

b = 0.1; % 0.15 %0.08

% Define my coefficients

k2 = 1/2;

k3 = 1/17;

k6 = 1/300;

b=0.2

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_SA(t,y,a,b,k2,k3,k6), 1:length(N\_SA), [S0, 0, I0, R0]);

% Accuracy estimation

p = y(:,3) +y(:,4); % I + R as defined previously

MSE = sum((N\_SA(from:to) - p(from:to)).^2)/(to-from)

% Draw figure

figure;

plot(N\_SA);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

title("SIRPs model with modified constants for SA")

**Appendix 3.12 – Applying Crowd effect to SIRPs model (SA):**

%% SIRPs\_q Model with the modifications to K2, K3 and K6:

% Define my coefficients

k2 = 1/4; %1/2

k3 = 1/50; %1/17

k6 = 1/100; %1/300

Ip = 0.02;

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_q\_SA(t,y,a,b,k2,k3,k6,Ip), 1:length(N\_SA), [S0, 0, I0, R0]);

% Accuracy estimation

p = y(:,3) +y(:,4); % I + R as defined previously

MSE = sum((N\_SA(from:to) - p(from:to)).^2)/(to-from)

% Draw figure

figure;

plot(N\_SA);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

title("SIRPs model with modified constants for SA")

**Appendix 3.13 – Essential functions for Appendix 3.10-3.12 (SA):**

% functions

function dydt = sirPs\_SA(t,y,a,b,k2,k3,k6)

% we now have 5 equations as 5 substances but also

% y(1) = S\_ign(t)

% y(2) = S\_res(t)

% y(3) = I(t)

% y(4) = R(t)

% our y(5) is infact simply S\_exh = 1 – sum(y), this is substituted by algebraic equation explored.

% use these variables to now define each diff eqn with this variables in one [] but split using semicolon ;

S\_exh = 1 - sum(y);

a1 = -a\*y(1)\*y(3) - k2\*y(1)\*y(3) + k6\*S\_exh;

a2 = k2\*y(1)\*y(3)-k3\*y(2);

a3 = a\*y(1)\*y(3) + a\*S\_exh\*y(3) -b\*y(3);

a4 = b\*y(3);

dydt = [a1; a2; a3; a4]; % only 4 as S\_exh excluded and c alculated algebraically above

end

function dydt\_q = sirPs\_q\_SA(t,y,a,b,k2,k3,k6,Ip)

% we now have 5 equations as 5 substances but also

% y(1) = S\_ign(t)

% y(2) = S\_res(t)

% y(3) = I(t)

% y(4) = R(t)

% our y(5) is infact simply S\_exh = 1 – sum(y), this is substituted by algebraic equation explored.

% use these variables to now define each diff eqn with this variables in one [] but split using semicolon ;

% q = k2/Ip

S\_exh = 1 - sum(y);

a1 = -a\*y(1)\*y(3) - (k2/Ip)\*y(1)\*y(3).^2 + k6\*S\_exh;

a2 = (k2/Ip)\*y(1)\*y(3).^2-k3\*y(2);

a3 = a\*y(1)\*y(3) + a\*S\_exh\*y(3) -b\*y(3);

a4 = b\*y(3);

dydt\_q = [a1; a2; a3; a4]; % only 4 as S\_exh excluded and c alculated algebraically above

end

**Appendix 4.1 – Calling the intervals as variables (TT):**

% Creating the waves in accordance to above intervals identified in Appendix 1.5

TTCC1 = N\_TT(1:84);

days1\_TT = 1:84;

TTCC2 = TTCC(85:169);

TTCC2 = TTCC2 - 5568;

TTCC2 = TTCC2/Population\_TT;

days2\_TT = 1:85;

TTCC3 = TTCC(170:end);

TTCC3 = TTCC3 - 7430;

TTCC3 = TTCC3/Population\_TT;

days3\_TT = 1:241;

**Appendix 4.2 – Wave 1 (TT):**

% Wave 1 data/variables for TT and plots

TT\_Wave1 = N\_TT(1:84,1);

%Plotting normalised wave

figure;

plot(TT\_Wave1);

title("Wave 1 of Normalised Cumulative Cases for Trinidad and Tobago");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

%Plotting log wave

figure;

logwave1\_TT = log(TT\_Wave1);

plot(logwave1\_TT);

title("Logarithm Plot for Wave 1 of Normalised Cumulative Cases for Trinidad and Tobago");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

% a and r values

mdl1a\_TT = fitlm(2:8, log(TTCC1(2:8)));

r\_TT1 = table2array(mdl1a\_TT.Coefficients(2, 1));

a\_TT1 = table2array(mdl1a\_TT.Coefficients(1, 1));

% Exponential model prediction

figure;

hold on;

TT\_Wave1 = N\_TT(1:84, 1); %normalised

plot(TT\_Wave1);

TTexp1 = zeros(84, 1);

for i = 1:84

TTexp1(i, 1) = exp(a\_TT1+(r\_TT1.\*(i))); % exp

end

plot(TTexp1);

hold off;

title("TT Wave 1 Exp Model Prediction");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

legend('Observed','Predicted');

ylim([0 0.001]);

% Error

figure;

Err\_TT1 = zeros(84, 1);

for i = 1:84

Err\_TT1(i, 1) = TT\_Wave1(i)-TTexp1(i); % need an array, not a matrix

end

plot(Err\_TT1);

title("Error for Wave 1 TT");

xlabel("Days since wave beginning");

ylabel("Error of Cumulative Fraction");

% K

figure;

K\_TT1 = zeros(84, 1);

for i = 1:84

K\_TT1(i, 1) = TT\_Wave1(i)\*(1+TTexp1(i))/TTexp1(i);

end

plot(K\_TT1);

title("K for Wave 1 TT");

xlabel("Days since wave beginning");

ylabel("Carrying Capacity");

%Used value for k

myK\_TT1 = 0.0175; %K(84) first and then change this for optimum

%ln(P/(K-P)) graph

figure;

hold on;

myLn\_TT1 = zeros(84, 1);

for i = 1:84

myLn\_TT1(i, 1) = log(abs(TT\_Wave1(i)/(myK\_TT1-TT\_Wave1(i))));

end

plot(myLn\_TT1);

title("ln(P/(K-P)) graph for Wave 1 TT");

xlabel("Days since wave beginning");

ylabel("ln(P/(K-P)");

legend('Observed','Predicted');

% ln a and ln r values

mdl1b\_TT = fitlm(days1\_TT(1:end), myLn\_TT1(1:end));

lnR\_TT1 = table2array(mdl1b\_TT.Coefficients(2, 1));

lnA\_TT1 = table2array(mdl1b\_TT.Coefficients(1, 1));

art\_TT1 = zeros(84, 1);

for i = 1:84

art\_TT1(i, 1) = lnA\_TT1+(lnR\_TT1\*(i));

end

plot(art\_TT1)

hold off;

% Normalised and Logistical model

figure;

hold on;

expart\_TT1 = exp(art\_TT1);

logistic\_TT1 = zeros(84, 1);

for i = 1:84

logistic\_TT1(i, 1) = (myK\_TT1.\*expart\_TT1(i))./(1+expart\_TT1(i));

end

plot(TT\_Wave1);

plot(logistic\_TT1);

title("Normalised and Logistical model for Wave 1 TT");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

legend('Observed','Predicted');

hold off;

% Error

Error\_TT1 = TT\_Wave1 - logistic\_TT1;

SumError\_TT1 = sum(Error\_TT1);

SumSQError\_TT1 = sumsqr(Error\_TT1);

Diff\_TT1 = SumError\_TT1 - SumSQError\_TT1;

**Appendix 4.3 – Wave 2 (TT):**

% Wave 2 data/variables for TT and plots

TT\_Wave2 = zeros(85, 1);

for i = 1:85

TT\_Wave2(i, 1) = TTCC(i+84, 1);

TT\_Wave2(i, 1) = TT\_Wave2(i, 1) - 5568;

TT\_Wave2(i, 1) = TT\_Wave2(i, 1)./1399488;

end

%Plotting normalised wave

figure;

plot(TT\_Wave2);

title("Wave 2 of Normalised Cumulative Cases for Trinidad and Tobago");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

%Plotting log wave

figure;

logwave2\_TT = log(TT\_Wave2);

plot(logwave2\_TT);

title("Logarithm Plot for Wave 2 of Normalised Cumulative Cases for Trinidad and Tobago");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

% calculating a and r

mdl2a\_TT = fitlm(1:9, log(TTCC2(1:9)));

r\_TT2 = table2array(mdl2a\_TT.Coefficients(2, 1));

a\_TT2 = table2array(mdl2a\_TT.Coefficients(1, 1));

% Exponential model prediction

figure;

hold on;

TT\_Wave2 = N\_TT(1:85, 1); %normalised

plot(TT\_Wave2);

TTexp2 = zeros(85, 1);

for i = 1:85

TTexp2(i, 1) = exp(a\_TT2+(r\_TT2.\*(i))); % exp

end

plot(TTexp2);

hold off;

title("TT Wave 2 Exp Model Prediction");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

legend('Observed','Predicted');

ylim([0 0.001]);

% Error

figure;

Err\_TT2 = zeros(85, 1);

for i = 1:85

Err\_TT2(i, 1) = TT\_Wave2(i)-TTexp2(i); % need an array, not a matrix

end

plot(Err\_TT2);

title("Error for Wave 2 TT");

xlabel("Days since wave beginning");

ylabel("Error of Cumulative Fraction");

% K

figure;

K\_TT2 = zeros(85, 1);

for i = 1:85

K\_TT2(i, 1) = TT\_Wave2(i)\*(1+TTexp2(i))/TTexp2(i);

end

plot(K\_TT2);

title("K for Wave 2 TT");

xlabel("Days since wave beginning");

ylabel("Carrying Capacity");

%Used value for k

myK\_TT2 = K\_TT2(85); %K(85) first and then change this for optimum

%ln(P/(K-P)) graph

figure;

hold on;

myLn\_TT2 = zeros(85, 1);

for i = 1:85

myLn\_TT2(i, 1) = log(abs(TT\_Wave2(i)/(myK\_TT2-TT\_Wave2(i))));

end

plot(myLn\_TT2);

title("ln(P/(K-P)) graph for Wave 2 TT");

xlabel("Days since wave beginning");

ylabel("ln(P/(K-P)");

legend('Observed','Predicted');

% Identifying ln a and ln r

mdl2b\_TT = fitlm(days2\_TT(1:end), myLn\_TT2(1:end));

lnR\_TT2 = table2array(mdl2b\_TT.Coefficients(2, 1));

lnA\_TT2 = table2array(mdl2b\_TT.Coefficients(1, 1));

art\_TT2 = zeros(85, 1);

for i = 1:85

art\_TT2(i, 1) = lnA\_TT2+(lnR\_TT2\*(i));

end

plot(art\_TT2)

hold off;

% Normalised and Logistical model

figure;

hold on;

expart\_TT2 = exp(art\_TT2);

logistic\_TT2 = zeros(85, 1);

for i = 1:85

logistic\_TT2(i, 1) = (myK\_TT2.\*expart\_TT2(i))./(1+expart\_TT2(i));

end

plot(TT\_Wave2);

plot(logistic\_TT2);

title("Normalised and Logistical model for Wave 2 TT");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

legend('Observed','Predicted');

hold off;

% Error

Error\_TT2 = TT\_Wave2 - logistic\_TT2;

SumError\_TT2 = sum(Error\_TT2);

SumSQError\_TT2 = sumsqr(Error\_TT2);

Diff\_TT2 = SumError\_TT2 - SumSQError\_TT2;

**Appendix 4.4 – Wave 3 (TT):**

% Wave 3 data/variables for TT and plots

TT\_Wave3 = zeros(241, 1);

for i = 1:241

TT\_Wave3(i, 1) = TTCC(i+169, 1);

TT\_Wave3(i, 1) = TT\_Wave3(i, 1) - 7430;

TT\_Wave3(i, 1) = TT\_Wave3(i, 1)./1399488;

end

%Plotting normalised wave

figure;

plot(TT\_Wave3);

title("Wave 3 of Normalised Cumulative Cases for Trinidad and Tobago");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

%Plotting log wave

figure;

logwave3\_TT = log(TT\_Wave3);

plot(logwave3\_TT);

title("Logarithm Plot for Wave 3 of Normalised Cumulative Cases for Trinidad and Tobago");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

% Identifying a and r

mdl3a\_TT = fitlm(1:8, log(TTCC3(1:8)));

r\_TT3 = table2array(mdl3a\_TT.Coefficients(2, 1));

a\_TT3 = table2array(mdl3a\_TT.Coefficients(1, 1));

% Exponential model prediction

figure;

hold on;

TT\_Wave3 = N\_TT(1:241, 1); %normalised

plot(TT\_Wave3);

TTexp3 = zeros(241, 1);

for i = 1:241

TTexp3(i, 1) = exp(a\_TT3+(r\_TT3.\*(i))); % exp

end

plot(TTexp3);

hold off;

title("TT Wave 3 Exp Model Prediction");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

legend('Observed','Predicted');

ylim([0 0.001]);

% Error

figure;

Err\_TT3 = zeros(241, 1);

for i = 1:241

Err\_TT3(i, 1) = TT\_Wave3(i)-TTexp3(i); % need an array, not a matrix

end

plot(Err\_TT3);

title("Error for Wave 3 TT");

xlabel("Days since wave beginning");

ylabel("Error of Cumulative Fraction");

% K

figure;

K\_TT3 = zeros(241, 1);

for i = 1:241

K\_TT3(i, 1) = TT\_Wave3(i)\*(1+TTexp3(i))/TTexp3(i);

end

plot(K\_TT3);

title("K for Wave 3 TT");

xlabel("Days since wave beginning");

ylabel("Carrying Capacity");

%Used value for k

myK\_TT3 = K\_TT3(241); %K(241) first and then change this for optimum

%ln(P/(K-P)) graph

figure;

hold on;

myLn\_TT3 = zeros(241, 1);

for i = 1:241

myLn\_TT3(i, 1) = log(abs(TT\_Wave3(i)/(myK\_TT3-TT\_Wave3(i))));

end

plot(myLn\_TT3);

title("ln(P/(K-P)) graph for Wave 3 TT");

xlabel("Days since wave beginning");

ylabel("ln(P/(K-P)");

legend('Observed','Predicted');

% Identifying ln a and ln r

mdl3b\_TT = fitlm(days3\_TT(1:end), myLn\_TT3(1:end));

lnR\_TT3 = table2array(mdl3b\_TT.Coefficients(2, 1));

lnA\_TT3 = table2array(mdl3b\_TT.Coefficients(1, 1));

art\_TT3 = zeros(241, 1);

for i = 1:241

art\_TT3(i, 1) = lnA\_TT3+(lnR\_TT3\*(i));

end

plot(art\_TT3)

hold off;

% Normalised and Logistical model

figure;

hold on;

expart\_TT3 = exp(art\_TT3);

logistic\_TT3 = zeros(241, 1);

for i = 1:241

logistic\_TT3(i, 1) = (myK\_TT3.\*expart\_TT3(i))./(1+expart\_TT3(i));

end

plot(TT\_Wave3);

plot(logistic\_TT3);

title("Normalised and Logistical model for Wave 3 TT");

xlabel("Days since wave beginning");

ylabel("Cumulative Fraction");

legend('Observed','Predicted');

hold off;

% Error

Error\_TT3 = TT\_Wave3 - logistic\_TT3;

SumError\_TT3 = sum(Error\_TT3);

SumSQError\_TT3 = sumsqr(Error\_TT3);

Diff\_TT3 = SumError\_TT3 - SumSQError\_TT3;

**Appendix 4.5 – Initial Conditions for SIR Model (TT):**

%% INITIAL CONDITIONS

R0 = 148/POPULATION\_TT;

I0 = N\_TT(1) - R0;

P0 = (I0 + R0);

S0 = 1 - P0;

**Appendix 4.6 – Parameters estimation for SIR Model (TT):**

%% Parameters estimation

% Remove semi-colons next to MSE to identify MSE values.

% Remember to reset values for I0 and R0 and S0 on each iteration if producing

% more than 1 set of graphs

from = 1; % 1 47

to = 101; % 48 101

waveEnd = 163;

b = 0.1;

% Linear regression

mdl = fitlm(from:to, log(N\_TT(from:to)));

r = table2array(mdl.Coefficients(2, 1));

c = table2array(mdl.Coefficients(1, 1));

% The last coefficient

a = r + b;

% Integrate system of ODE

[t, y] = ode45(@(t, y) sir(t, y, a, b), 1:length(N\_TT), [S0, R0]);

% Accuracy estimation

p = 1 - y(:, 1);

MSE = sum((N\_TT(from:to) - p(from:to)) .^ 2) / (to-from);

% Draw figure

figure;

plot(N\_TT);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

**Command Window (for this section):**

>> S0

S0 =  
  
 0.999852088763891

>> S101 = 1 - (N\_TT(101) + N\_TT(91))

S101 =  
  
 0.991614790551973

**Appendix 4.7 – Testing initial conditions – I(0) (TT):**

%% Play with original conditions

% Remove semi-colons next to MSE to identify MSE values for each iteration.

figure

plot(N\_TT);

hold on;

lims = ylim();

labs = {'Observed'};

I00 = I0 / 2;

for k = 1:5

S0 = 1 - I00 - R0;

[t, y] = ode45(@(t, y) sir(t, y, a, b), 1:length(N\_TT), [S0, R0]);

p = 1 - y(:, 1);

plot(p);

MSE = sum((N\_TT(from:to) - p(from:to)) .^ 2) / (to-from);

labs = [labs, sprintf("I(0) = %.4g", I00)];

I00 = I00 \* 2;

end

legend(labs, 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

title("Observing different values for I(0) for TT")

**Appendix 4.8 – Testing initial conditions – I(0) and a (TT):**

%% Play with original conditions with optimisation

% Remove semi-colons next to MSE to identify MSE values for each iteration.

figure

plot(N\_TT);

hold on;

lims = ylim();

labs = {'Observed'};

I00 = I0 / 2;

for k = 1:5

S0 = 1 - I00 - R0;

aOpt = fminbnd(@(x) forSearch(x, length(N\_TT), b, [S0, R0], N\_TT), b, 3 \* b);

[t, y] = ode45(@(t, y) sir(t, y, aOpt, b), 1:length(N\_TT), [S0, R0]);

p = 1 - y(:, 1);

plot(p);

MSE = sum((N\_TT(from:to) - p(from:to)) .^ 2) / (to-from);

labs = [labs, sprintf("I(0) = %.4g, a = %.4f", I00, aOpt)];

I00 = I00 \* 2;

end

legend(labs, 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

title("Observing different values for I(0) for TT with varying values of a");

**Appendix 4.9 – Essential functions for Appendix 4.6-4.8 (TT):**

%% functions

function dydt = sir(t, y, a, b)

I = 1 - sum(y);

dydt = [ - a \* y(1) \* I;

b \* I];

end

function err = forSearch(a, wEnd, b, y0, N\_TT)

[t, y] = ode45(@(t, y) sir(t, y, a, b), 1:wEnd, y0);

p = 1 - y(:, 1);

err = sum((N\_TT(1:wEnd) - p(1:wEnd)) .^ 2) / wEnd;

end

**Appendix 4.10 – Initial Conditions for SIRPs Model (TT):**

%% INITIAL CONDITIONS

R0 = 148/POPULATION\_TT;

I0 = 0.0001686;

P0 = (I0 + R0);

S0 = 1 - P0;

**Appendix 4.11 – Parameters estimation for SIRPs Model (TT):**

%% Parameters estimation

% Remove semi-colons next to MSE to identify MSE values.

% Remember to reset values for I0 and R0 and S0 on each iteration if producing

% more than 1 set of graphs

from = 1; % 1 47

to = length(N\_TT); % 48 101

% waveEnd = 163;

b = 0.1;

% Linear regression

mdl = fitlm(from:to, log(N\_TT(from:to)));

r = table2array(mdl.Coefficients(2, 1));

c = table2array(mdl.Coefficients(1, 1));

% The last coefficient

a = 0.1071;

% Define my coefficients

k2 = 1;

k3 = 1/50;

k6 = 1/100;

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_TT(t,y,a,b,k2,k3,k6), 1:length(N\_TT), [S0, 0, I0, R0]);

% Accuracy estimation

p = y(:,3) +y(:,4); % I + R as defined previously

MSE = sum((N\_TT(from:to) - p(from:to)).^2)/(to-from)

%% Draw figure of Pyschology on SIR Model

figure;

plot(N\_TT);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

title("Social Psychology additions to SIR models for TT");

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

**Appendix 4.12 – Testing initial conditions – K2, K3 and K6 (TT):**

%% Modifications to the values of k2 and applying:

k2 = 1/4;

k3 = 1/50;

k6 = 1/100;

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_TT(t,y,a,b,k2,k3,k6), 1:length(N\_TT), [S0, 0, I0, R0]);

% Accuracy estimation

p = y(:,3) +y(:,4); % I + R as defined previously

MSE = sum((N\_TT(from:to) - p(from:to)).^2)/(to-from)

figure;

plot(N\_TT);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

title("Social Psychology additions to SIR models for TT: K2 = 1/4. 4 days to modify behaviour");

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

%% Modifications to the values of k3 and applying:

k2 = 1;

k3 = 1;

k6 = 1/100;

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_TT(t,y,a,b,k2,k3,k6), 1:length(N\_TT), [S0, 0, I0, R0]);

% Accuracy estimation

p = y(:,3) +y(:,4); % I + R as defined previously

MSE = sum((N\_TT(from:to) - p(from:to)).^2)/(to-from)

figure;

plot(N\_TT);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

title("Social Psychology additions to SIR models for TT: K3 = 1. State of resistance for 1 day");

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

%% Modifications to the values of k6 and applying:

k2 = 1;

k3 = 1/50;

k6 = 1/250;

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_TT(t,y,a,b,k2,k3,k6), 1:length(N\_TT), [S0, 0, I0, R0]);

% Accuracy estimation

p = y(:,3) +y(:,4); % I + R as defined previously

MSE = sum((N\_TT(from:to) - p(from:to)).^2)/(to-from)

figure;

plot(N\_TT);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

title("Social Psychology additions to SIR models for TT: K6 = 1/250. State of exhaustion of 250 days");

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

%% SIRPs Model with the modifications to K2, K3 and K6:

k2 = 1/4;

k3 = 1;

k6 = 1/250;

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_TT(t,y,a,b,k2,k3,k6), 1:length(N\_TT), [S0, 0, I0, R0]);

% Accuracy estimation

p = y(:,3) +y(:,4); % I + R as defined previously

MSE = sum((N\_TT(from:to) - p(from:to)).^2)/(to-from)

figure;

plot(N\_TT);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

title("Social Psychology additions to SIR models for TT with optimal values for K2, K3 and K6");

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

**Appendix 4.13 – Applying Crowd effect to SIRPs model (TT):**

%% SIRPs\_with Crowd effect:

% Define my coefficients

k2 = 1;

k3 = 1/50;

k6 = 1/100;

Ip = 0.02; % Optimal after 0.19

% Integrate the system of ODE

[t,y] = ode45( @(t,y) sirPs\_q\_TT(t,y,a,b,k2,k3,k6,Ip), 1:length(N\_TT), [S0, 0, I0, R0]);

% Accuracy estimation

p = y(:,3) +y(:,4); % I + R as defined previously

MSE = sum((N\_TT(from:to) - p(from:to)).^2)/(to-from)

% Draw figure

figure;

plot(N\_TT);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

title("SIRPs model with crowd effect for TT")

**Appendix 4.14 – Essential functions for Appendix 4.11-4.13 (TT):**

%% functions

function dydt = sirPs\_TT(t,y,a,b,k2,k3,k6)

% we now have 5 equations as 5 substances but also

% y(1) = S\_ign(t)

% y(2) = S\_res(t)

% y(3) = I(t)

% y(4) = R(t)

% our y(5) is infact simply S\_exh = 1 – sum(y), this is substituted by algebraic equation explored.

% use these variables to now define each diff eqn with this variables in one [] but split using semicolon ;

S\_exh = 1 - sum(y);

a1 = -a\*y(1)\*y(3) - k2\*y(1)\*y(3) + k6\*S\_exh;

a2 = k2\*y(1)\*y(3)-k3\*y(2);

a3 = a\*y(1)\*y(3) + a\*S\_exh\*y(3) -b\*y(3);

a4 = b\*y(3);

dydt = [ a1; a2; a3; a4]; % only 4 as S\_exh excluded and calculated algebraically above

end

function dydt\_q = sirPs\_q\_TT(t,y,a,b,k2,k3,k6,Ip)

% we now have 5 equations as 5 substances but also

% y(1) = S\_ign(t)

% y(2) = S\_res(t)

% y(3) = I(t)

% y(4) = R(t)

% our y(5) is infact simply S\_exh = 1 – sum(y), this is substituted by algebraic equation explored.

% use these variables to now define each diff eqn with this variables in one [] but split using semicolon ;

% q = k2/Ip

S\_exh = 1 - sum(y);

aq1 = -a\*y(1)\*y(3) - (k2/Ip)\*y(1)\*y(3).^2 + k6\*S\_exh;

aq2 = (k2/Ip)\*y(1)\*y(3).^2-k3\*y(2);

aq3 = a\*y(1)\*y(3) + a\*S\_exh\*y(3) -b\*y(3);

aq4 = b\*y(3);

dydt\_q = [aq1; aq2; aq3; aq4]; % only 4 as S\_exh excluded and c alculated algebraically above

end