State University of Campinas

Institute of Mathematics, Statistics and Scientific Computing

Covid's SCIRD Metapopulation Model and Study of Lockdown Effect

Marina de Souza Ripper

2021.

Monography presented to the Institute of Mathematics, University Statistics and Scientific Computing State of Campinas as part of the requirements for obtaining credits in the Supervised Project subject, under the guidance of Prof. João Frederico da Costa Azevedo Meyer.

Contents

Abstract

Abstract. The COVID-19 virus is a disease caused by Sars-cov-2, and its behaviour varies in different locations, in the present work we focus on São Paulo state and specifically on some towns in its interior. The model here developed is of Kermack-McKendrick type, which considers compartments of Susceptible, Confined, Infected, Recovered - with temporary immunity, and Dead individuals. Such model will allow us to analyze the effects of how a lockdown policy on one town, affects neighbouring ones, and predict the behaviour of individuals in each compartment due to this lockdown. For this purpose we use data obtained from the COVID-19 dynamics, for towns in the interior of São Paulo state, with different public health programs.

Keywords. COVID-19, Lockdown Policy, Metapopulation Models, Kermack-McKendric, Mathematical Modelling, Mathematical Epidemiology, Nonlinear Systems of ODE.

1 Introduction

To analyze the effect of how a lockdown policy on one town affects neighbouring ones, a numerical approximation of a SCIRD-type model, was used to present different possible scenarios. The tests were undertaken in a python3.9.6 environment.

What binds the relationship between cities, is the transit of individuals between them, on that note, the main difficulty found on the development of this work, was to find data that was trustworthy and consistent. At first the idea was to analyze toll data, more specifically DER(Departamento de Estradas de Rodagem)[1] data of 2020, but this was found to be unreliable and inconsistent for the present work. The Data chosen for analysing flow while still not ideal was from IBGE´s analyses of Populational Arrangements, where there was found data of the transit of people that work and study on other cities of the same arrangement, The arrangement chosen was of Americana - Santa Bárbara d´Oeste/SP, for its cities with more homogeneous population size.

The Work is organized as follows: In Section 2, is presented the SCIRD model and its analysis, (R_0); in Section 3 simulations based on the data presented are performed; in 4 the code made in python 3.6.9 for the simulations is presented, in 5 we analyze such results and if the results live up to the expected outcome. Finally in Section 6 conclusions are made and future research is suggested.

2 Methodology

For the present article the model created is presented in figure 1:

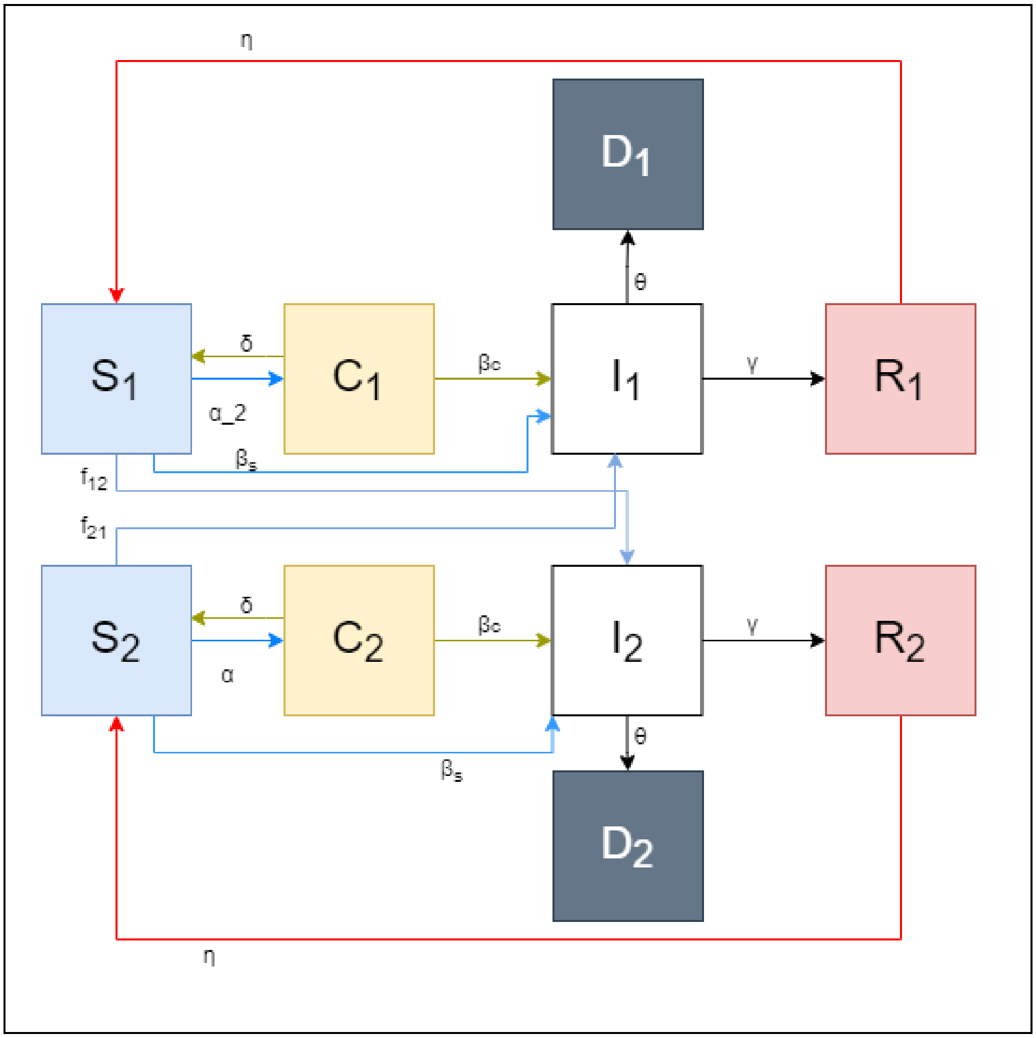


Figure 1: Model SCIRD

In this model, we consider the following classes of individuals: S (Susceptible),C (Confined), I (Infected), R (Recovered - with temporary immunity) and D (Dead), for cities 1, 2, 3...n where n is the total number of cities.

The Model can be written as:

$$
\begin{aligned}
\frac{\partial S_{Ci}}{\partial t} &= \mu(S_{Ci}+C_{Ci}+I_{Ci}+R_{Ci}) - \alpha_k S_{Ci} + \delta C_{Ci} - \beta S_{Ci} I_{Ci} + \eta R_{Ci} - \mu S_{Ci} - \sum_{j\neq i}(f_{ij}\beta I_{Cj} S_{Ci});\\
\frac{\partial C_{Ci}}{\partial t} &= \alpha_k S_{Ci} - \delta C_{Ci} - \mu C_{Ci} - \beta_c C_{Ci} I_{Ci};\\
\frac{\partial I_{Ci}}{\partial t} &= \beta S_{Ci} I_{Ci} - \theta I_{Ci} - \gamma I_{Ci} - \mu I_{Ci} + \beta_c C_{Ci} I_{Ci} + \sum_{j\neq i}(f_{ji}\beta I_{Ci} S_{Cj});\\
\frac{\partial R_{Ci}}{\partial t} &= \gamma I_{Ci} - \eta R_{Ci} - \mu R_{Ci};\\
\frac{\partial D_{Ci}}{\partial t} &= \theta I_{Ci};\\
\forall i &= 1,2,3,...n \quad also \quad k=2 \quad if \quad i=1 \quad and \quad k=0 \quad \forall \quad i\neq 1;
\end{aligned}
$$

Analyzing each term we have that αS corresponds to the confinement of the Susceptible class,$\alpha_2 S$ corresponds to the confinement and lockdown o the susceptible class of city 1. The term β corresponds to the transmission coefficient from susceptible to infected individuals, while $beta_c$ corresponds the transmission coefficient from confined to infected individuals. Since confined individuals are less likely to contract the disease, $beta_c$ is smaller than $beta$. The term $deltaC$ represents the deconfinement of the population and the model considers there to be no permanent immunity, so the immunity loss rate is represented by η.

The term $\mu(S_{C1}+C_{C1}+I_{C1}+R_{C1})$ represents the growth of the population, and μS_{Cn}, μC_{Cn}, μI_{Cn} and μR_{Cn}) where n = 1,2,3 represents the natural death of each compartment. For the transit between cities 1, 2 and 3 we have:

1. $-f_{ij}\beta I_{Cj} S_{Ci}$ corresponds to the transit from susceptibles from city i to the infected in city j, where f_{ij} is the transit computed in accord to the data in IBGE[2]. considering city i as the origin of the transit, ind_j as the cities j integration index with the population arrangement and f_i the quantity of people of city i who work and study in other municipalities in the arrangement, we have:

$$
f_{ij} = f_i \cdot \frac{100 \cdot ind_j}{\sum\limits_{k\neq i}(ind_k)}
$$

2. $+f_{ji}\beta I_{Ci} S_{Cj}$ corresponds to the transit of susceptibles from city j to the infected in city i, the calculation is the same as for f_{ji} switching j with i.

Finally, the death rate of the infected individuals is given by θ and the recuperation rate is given by γ.

3 Tests and Results

The tests were made for 3 cities, City 1: Americana (SP), City 2: Nova Odessa (SP), City 3: Santa Bárbara d´Oeste (SP), simulating the lockdown in Americana, the Model was coded in python 3.9. For the transit the calculations made out of table 1 are:

Population Arrangements	Population	People who work and study in other municipalities in the arrangement	Municipality's integration index with the arrangement
Americana (SP)	210.638	41.118	0,27
Nova Odessa (SP)	51.242	10.848	0,29
Santa Bárbara d´Oeste (SP)	180.009	35.560	0,27

Figure 2: Data from IBGE[2] referencing Population Arrangements.

$$\begin{aligned}
f12 &= \frac{41118 * 100 * 0,29}{0,27 + 0,29} = 20970; \\
f13 &= \frac{41118 * 100 * 0,27}{0,27 + 0,29} = 19818,876; \\
f21 &= \frac{10848 * 100 * 0,27}{0,27 + 0,27} = 5424; \\
f23 &= \frac{10848 * 100 * 0,27}{0,27 + 0,27} = 5424; \\
f32 &= \frac{35560 * 100 * 0,29}{0,29 + 0,27} = 20970; \\
f31 &= \frac{35560 * 100 * 0,27}{0,29 + 0,27} = 19818,876;
\end{aligned}$$

The tests were made with the following parameters:

Parâmetros								
delta	eta	mi	alfa	theta	beta	gamma	omega	beta_c
0.4	0.02	0.0000040849	0.3	0.03	4.3e-6	0.97	0.0000058	4.3e-7

Figure 3: Tests Parameters.

As mentioned $4.3e - 6 = \beta > \beta_c = 4.3e - 7$, theta is the death rate of COVID and is approximately 0.3 [3], which makes the recovery rate $gamma = 1 - 0.3 = 0.97$.

Fluxo/1000					
f12	f21	f23	f32	f31	f13
20.970	19.818	5.424	5.424	20.970	19.818

Figure 4: Tests Transit.

Valores Iniciais				
C0_C1	I0_C1	R0_C1	D0_C1	S0_C1
0	50	0	0	N_C1 - (C0_C1+I0_C1+R0_C1+D0_C1)
C0_C2	I0_C2	R0_C2	D0_C2	S0_C2
0	40	0	0	N_C2 - (C0_C2+I0_C2+R0_C2+D0_C2)
C0_C3	I0_C3	R0_C3	D0_C3	S0_C3
0	30	0	0	N_C3 - (C0_C3+I0_C3+R0_C3+D0_C3)

Figure 5: Initial Values.

3.1 Tests

3.1.1 Test 1

Lockdown of 15 days with 60% confinement ($\alpha_2 = 0.6$)

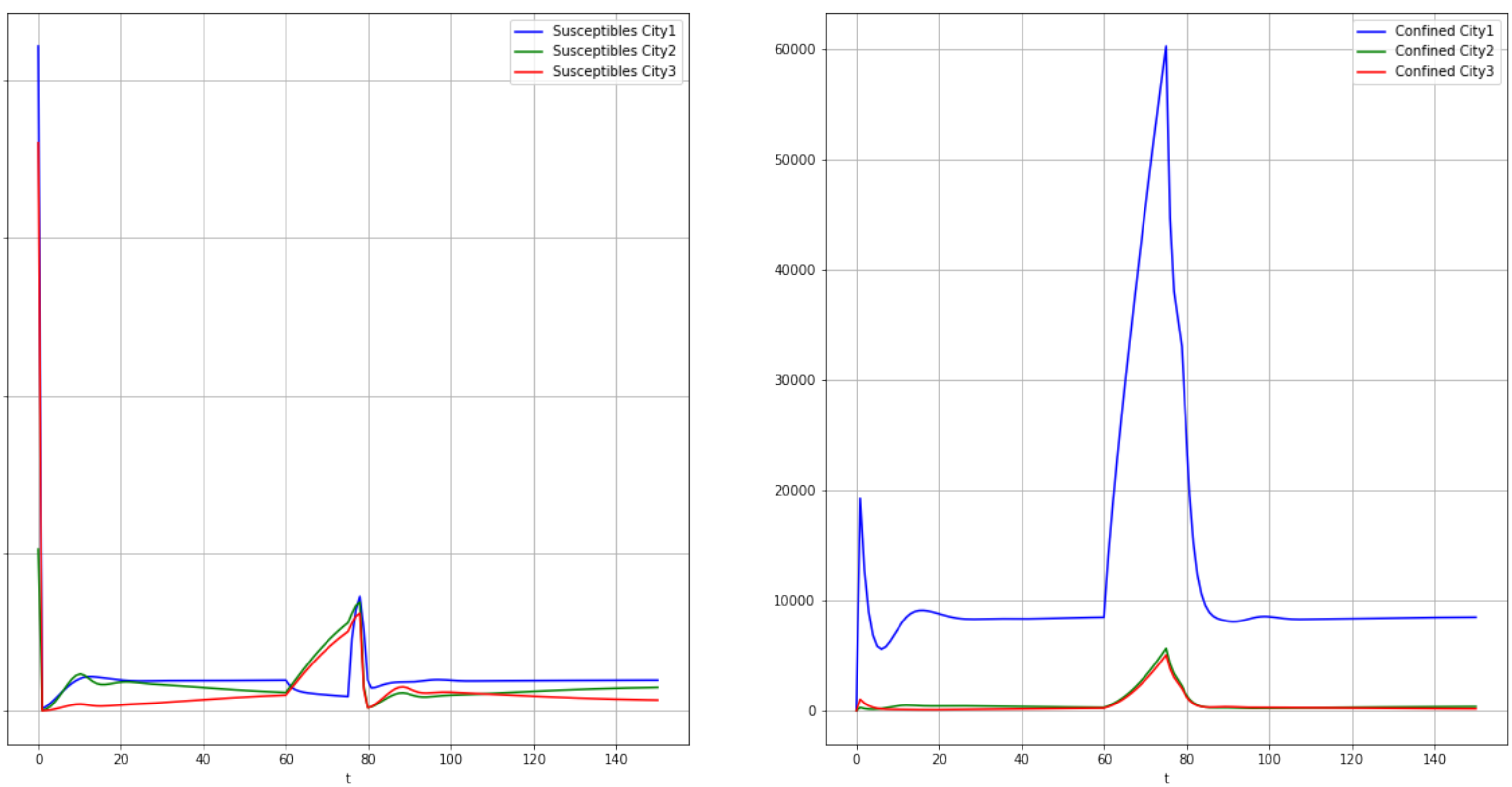


Figure 6: Test 1:Susceptible and Confined

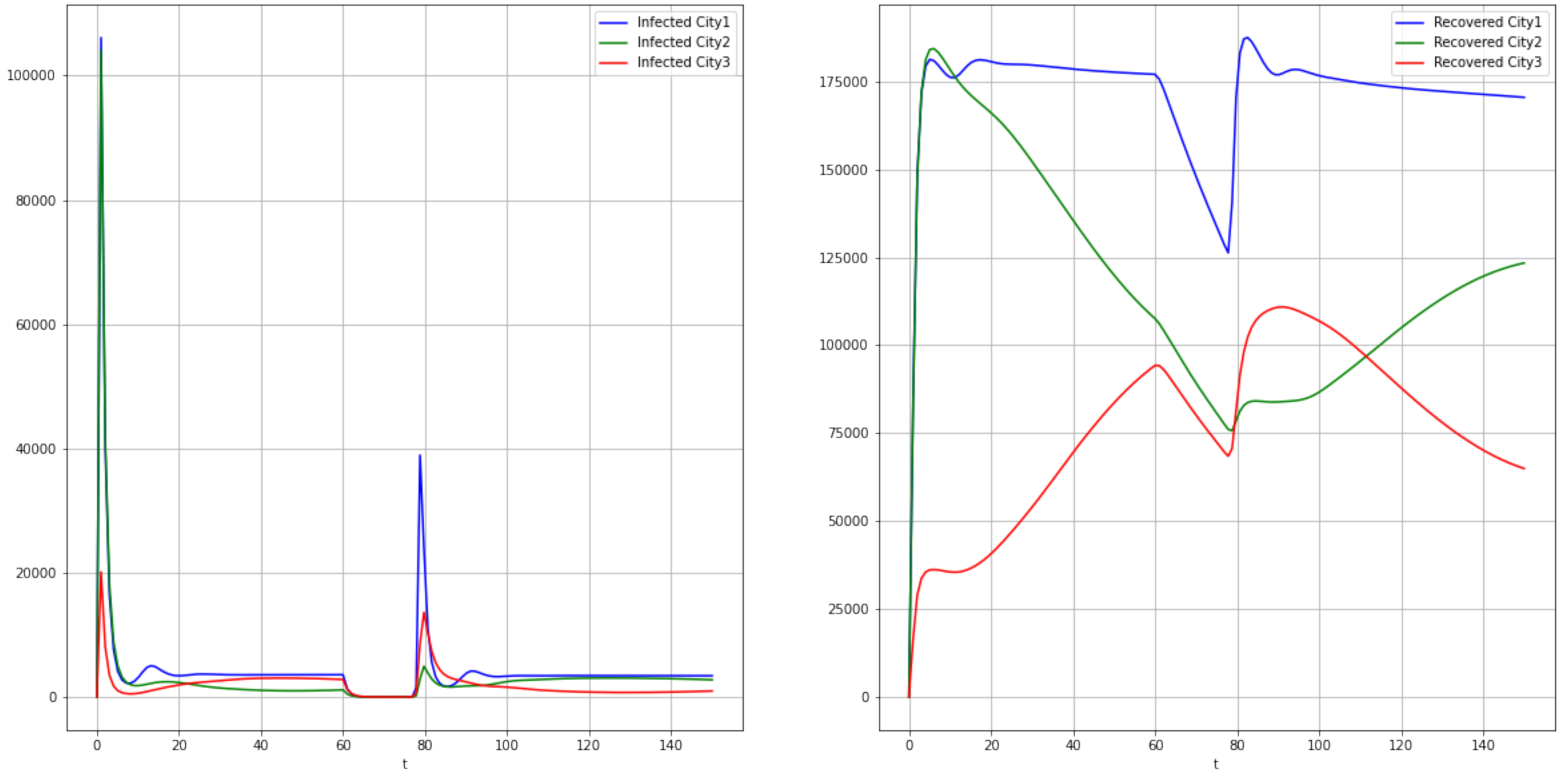
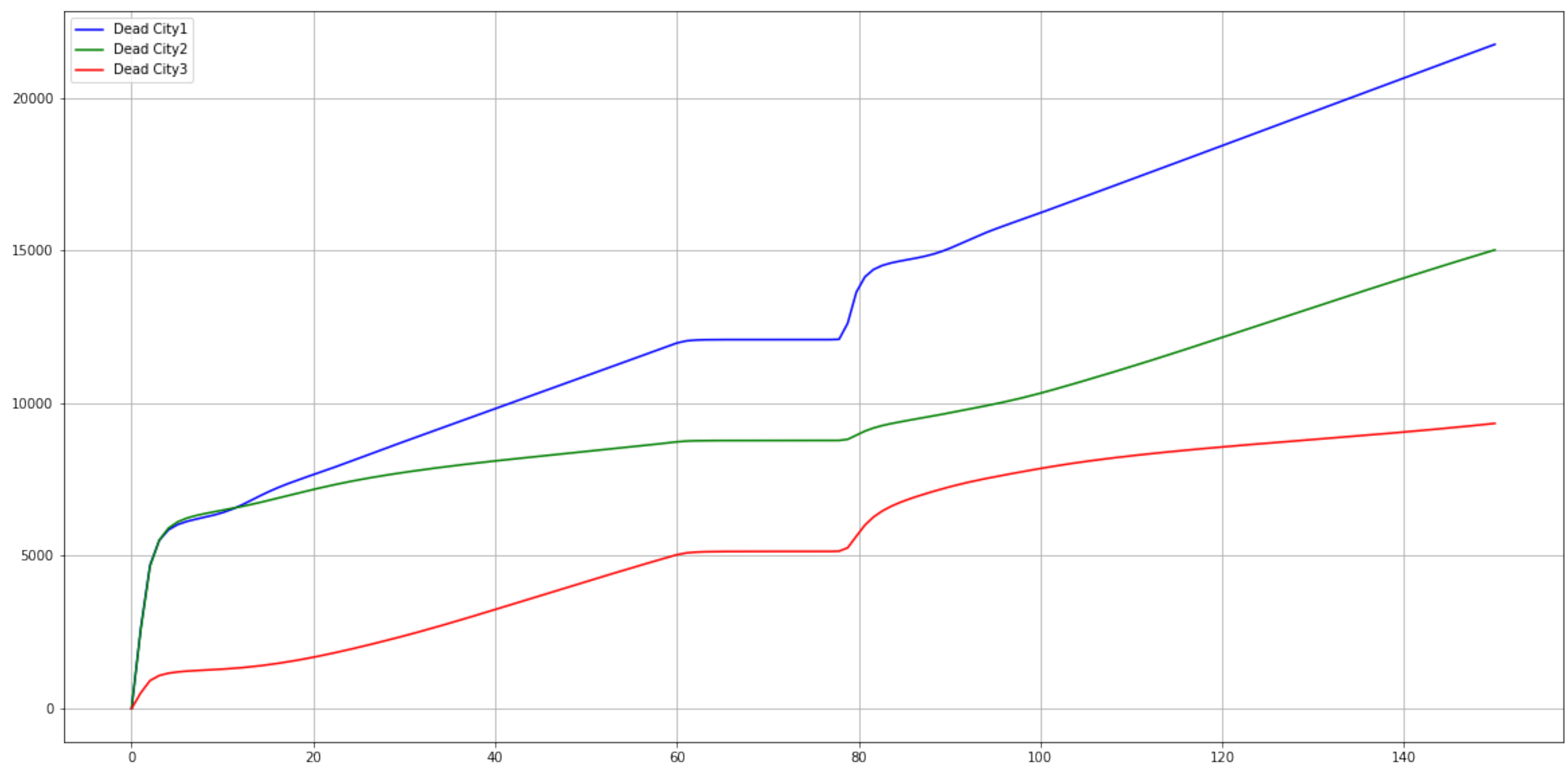


Figure 7: Test 1:Infected and Recovered



[H]

Figure 8: Test 1: Dead

3.1.2 Test 2

: Lockdown of 30 days with 60% confinement ($\alpha_2 = 0.6$)

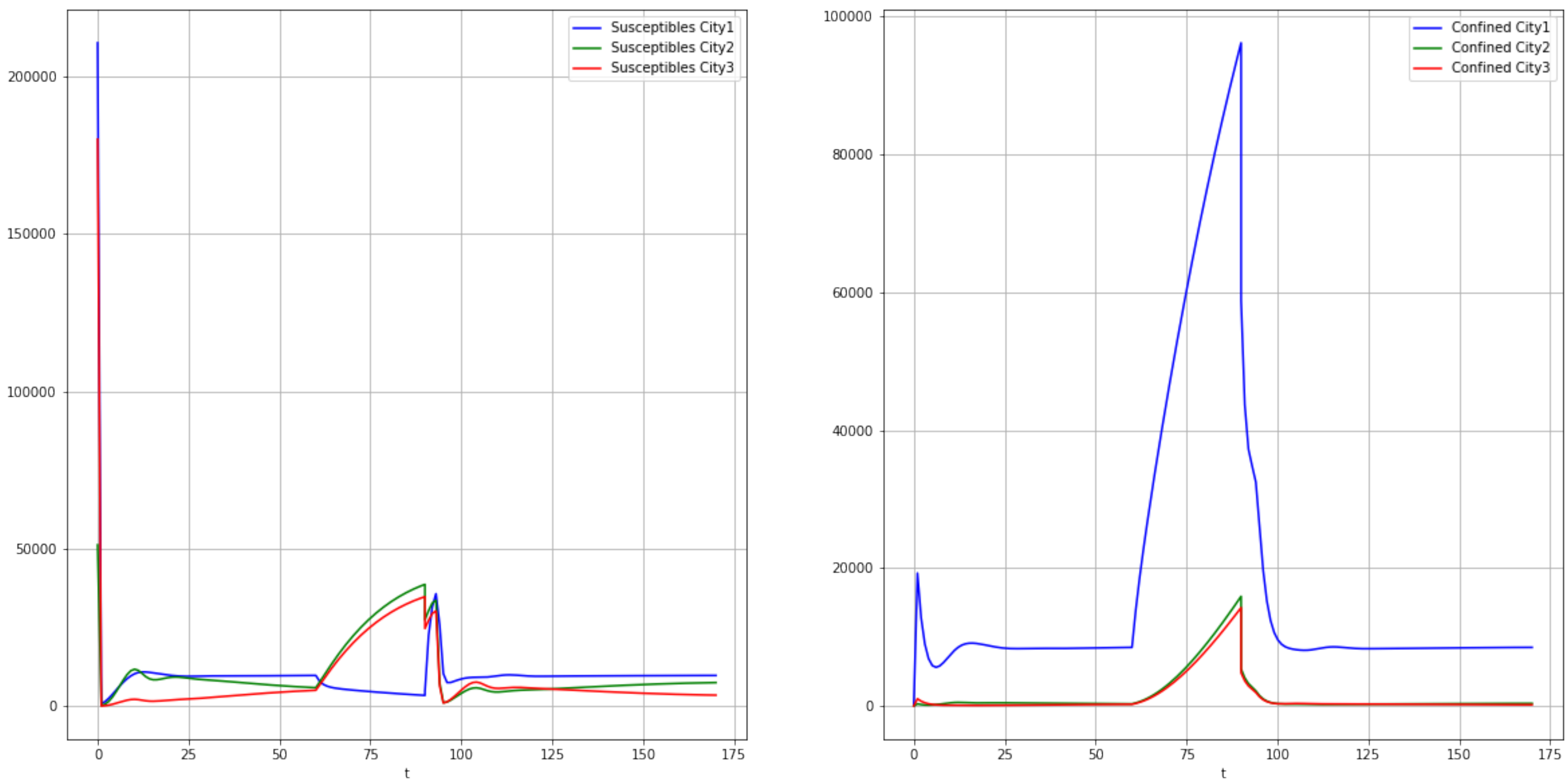


Figure 9: Test 2:Susceptible and Confined

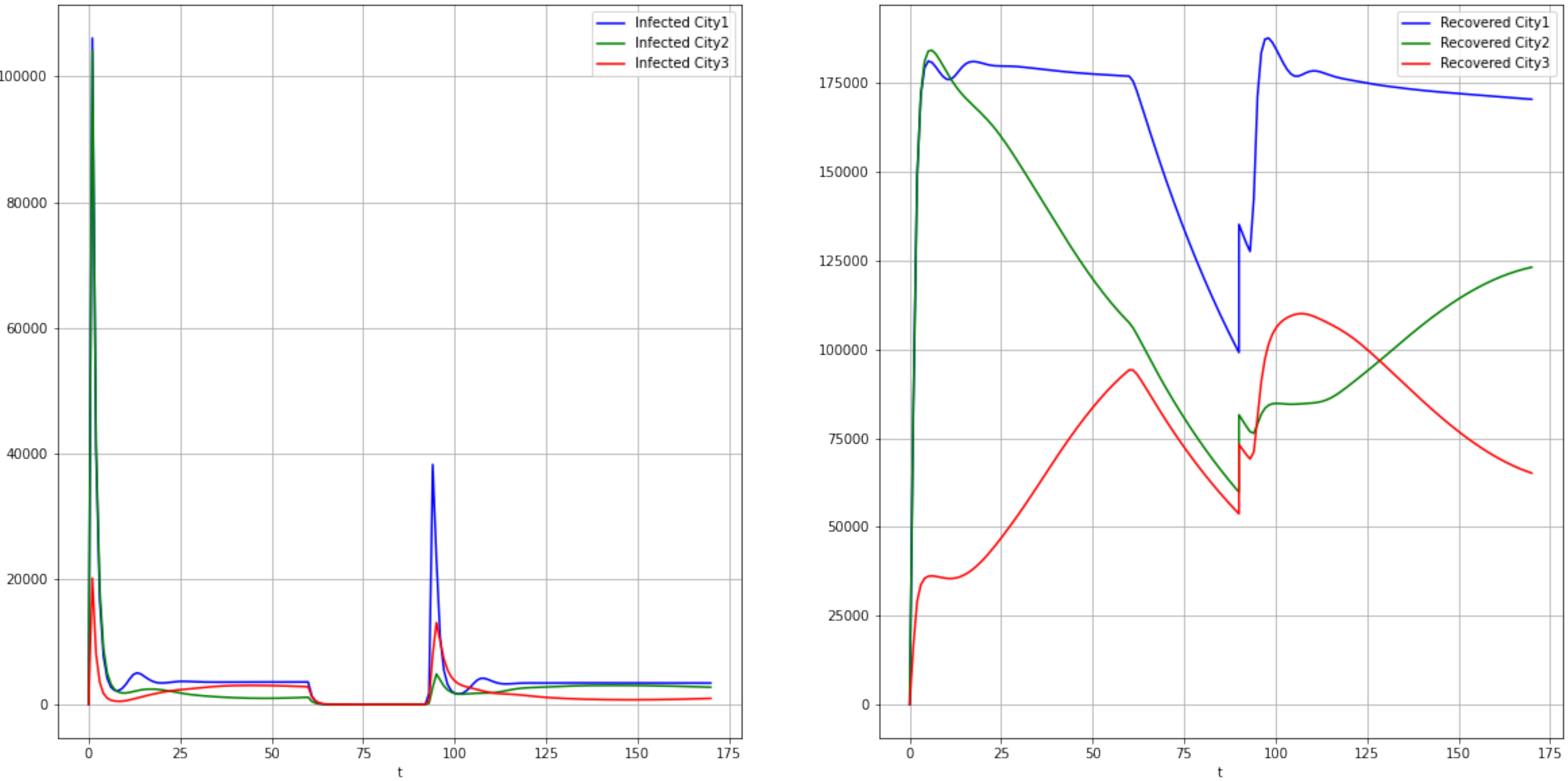


Figure 10: Test 2:Infected and Recovered

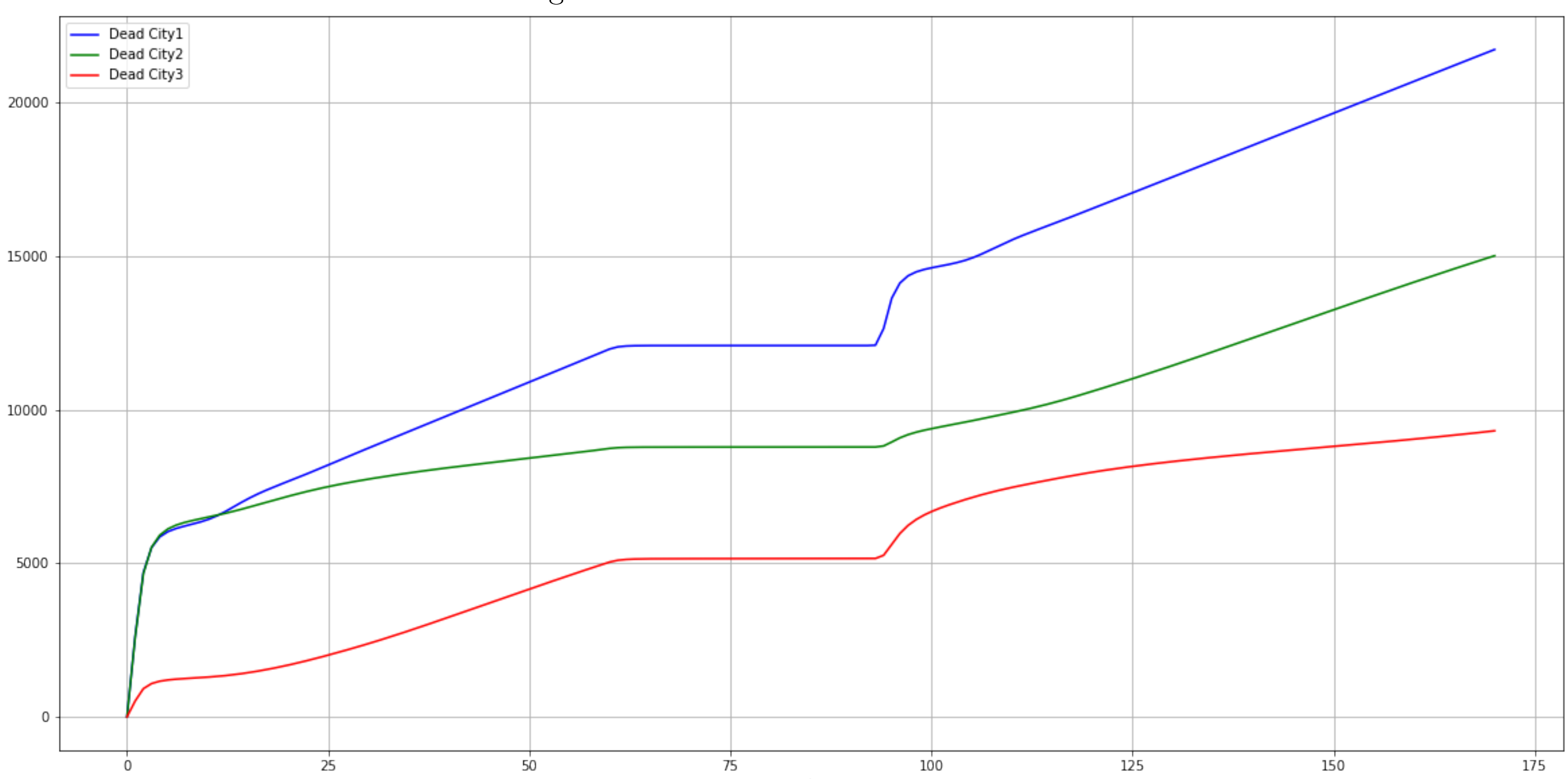
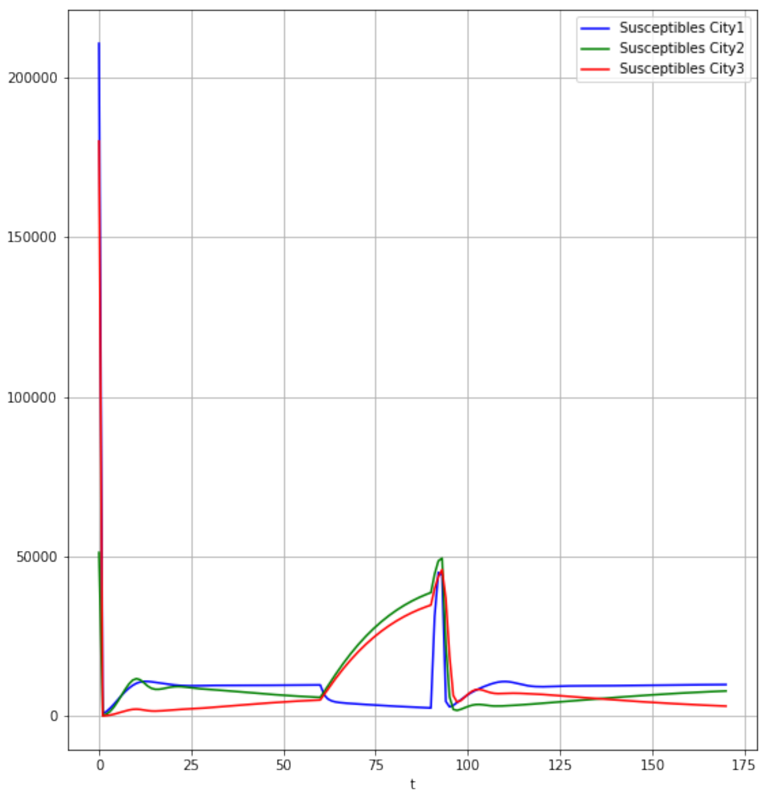


Figure 11: Test 2: Dead

3.1.3 Test 3:

Lockdown of 30 days with 80% confinement ($\alpha_2 = 0.8$)



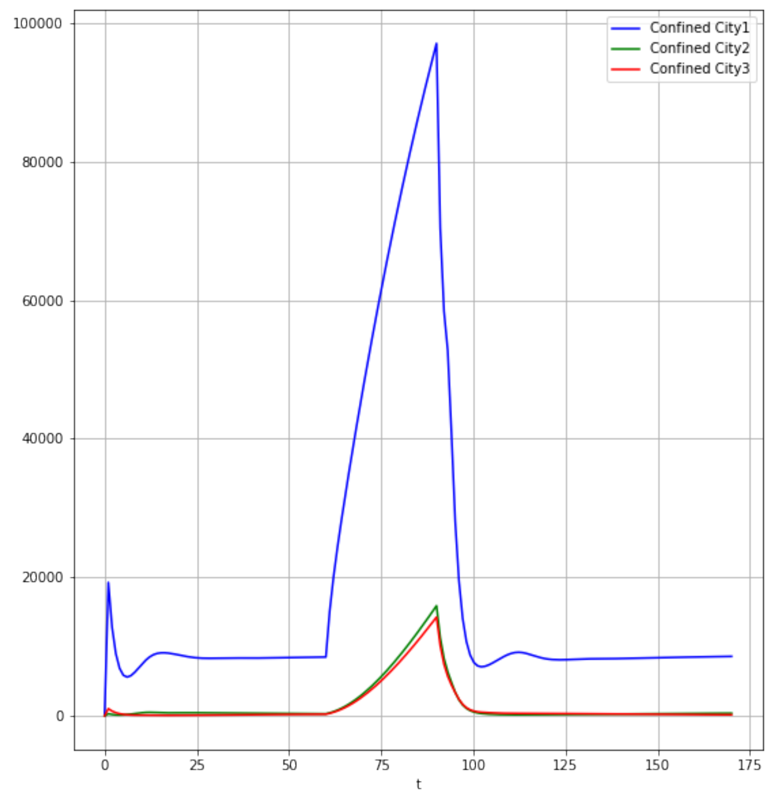


Figure 12: Test 3:Susceptible and Confined

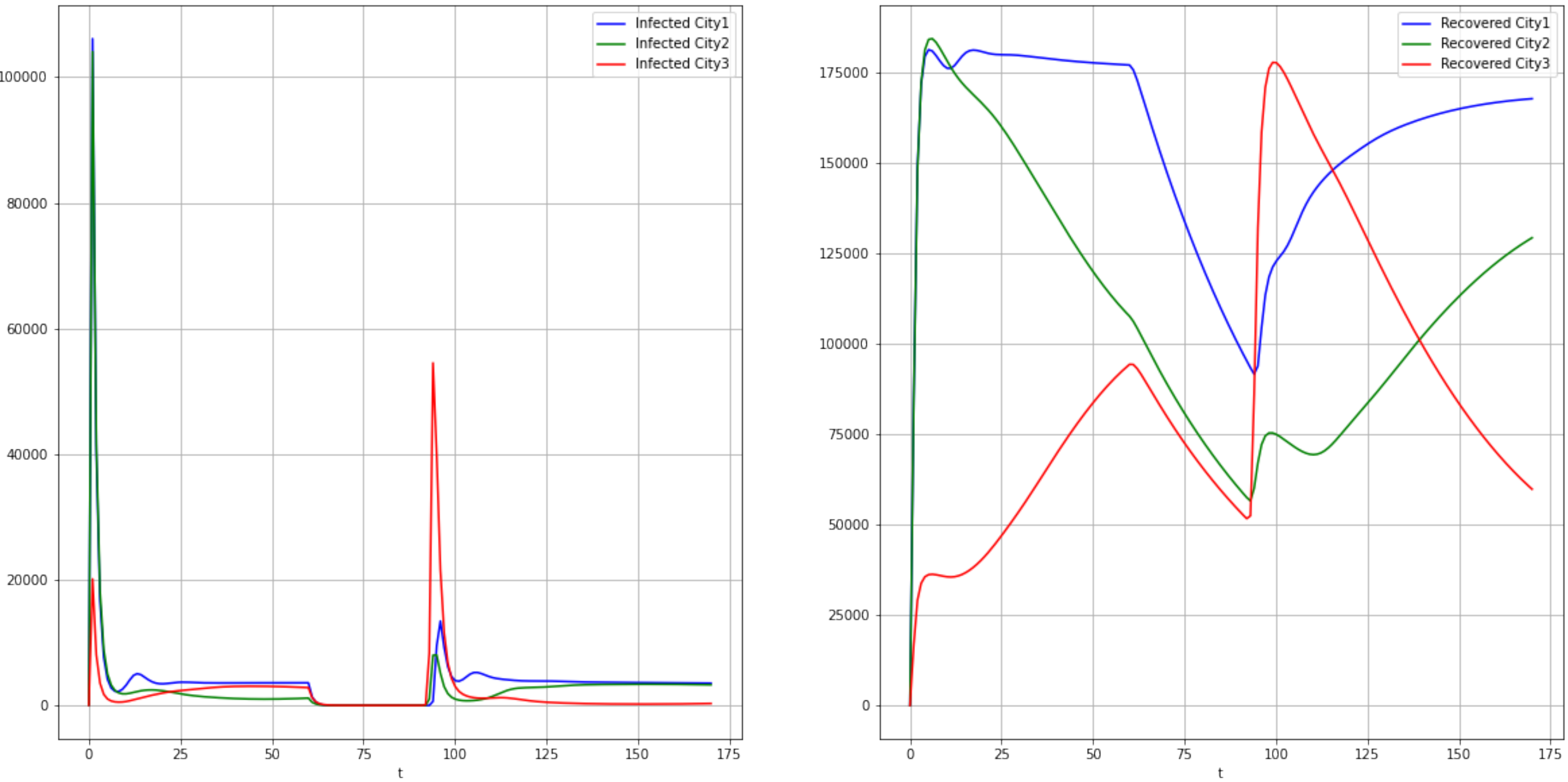
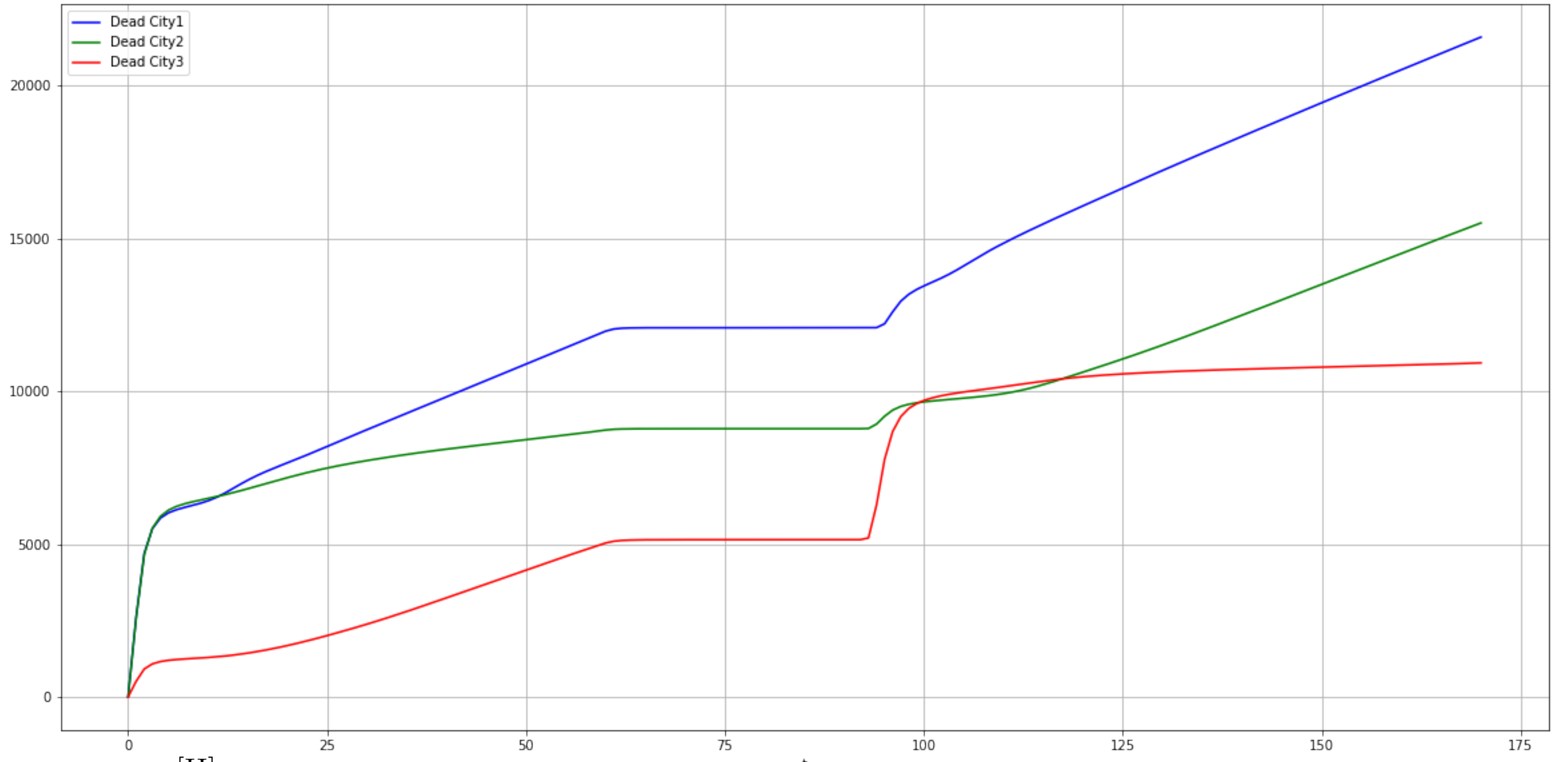


Figure 13: Test 3:Infected and Recovered



[H]

Figure 14: Test 3: Dead

4 Code

In [ ]:

```
#Abaixo definimos os parâmetros:
delta = 0.4
eta = 0.02
mi = 0.0000040849
alfa = 0.02
theta = 0.03
beta = 4.3e-6
beta_2 = 4.3e-5
beta_3 = 2e-5
gamma = 0.97
omega = 0.0000058
beta_c = 4.3e-7
```

Figure 15: Test 3:Susceptible and Confined

In [ ]:

```
#Definimos os valores incicais das cidades 1: americana, 2: nova odessa e 3: santa barbara
N_C1 = 210638

N_C2 = 51242

N_C3 = 180009


#Abaixo os valores iniciais de cada compartimento C: Confinados, I: Infectados, R: Recuperados, D: Mortos e S: Sucetíveis
C0_C1 = 0
I0_C1 = 50
R0_C1 = 0
D0_C1 = 0
S0_C1 = N_C1 - (C0_C1+I0_C1+R0_C1+D0_C1)
C0_C2 = 0
I0_C2 = 40
R0_C2 = 0
D0_C2 = 0
S0_C2 = N_C2 - (C0_C2+I0_C2+R0_C2+D0_C2)
C0_C3 = 0
I0_C3 = 30
R0_C3 = 0
D0_C3 = 0
S0_C3 = N_C3 - (C0_C3+I0_C3+R0_C3+D0_C3)
#Colocamos os valores iniciais em uma array:
y0 = [S0_C1,C0_C1,I0_C1,R0_C1,D0_C1,S0_C2,C0_C2,I0_C2,R0_C2,D0_C2,S0_C3,C0_C3,I0_C3,R0_C3,D0_C3]
f12 = 20.970 # I_C2*S_C1
f21 = 5.424 #I_C1* S_C2
f23 = 5.424 #I_C3* S_C2
f32 = 20.970 #I_C2*S_C3
f31 = 19.818 #I_C1* S_C3
f13 = 19.818 #I_C3*S_C1
```

In [ ]:

```
#Criamos uma função com o modelo:
def deriv(y,t,beta, gamma, delta, eta, mi, alfa, theta, alfa_2,beta_c,f12,f21,f23,f32,f31,f13):
    S_C1,C_C1,I_C1,R_C1,D_C1,S_C2,C_C2,I_C2,R_C2,D_C2,S_C3,C_C3,I_C3,R_C3,D_C3= y
    dSdt_C1 = mi * (S_C1+C_C1+I_C1+R_C1) - alfa_2 * S_C1 + delta * C_C1 - beta * S_C1 * I_C1 + eta * R_C1 - mi * S_C1 - f12*beta*I_C2*S_C1 - f13*beta*I_C3*S_C
    dCdt_C1 = alfa_2 * S_C1 - delta * C_C1 - mi * C_C1 - beta_c * C_C1 * I_C1
    dIdt_C1 = beta * S_C1 * I_C1 - theta * I_C1 - gamma * I_C1 - mi * I_C1 + beta_c * C_C1 * I_C1 + f21*beta*I_C1* S_C2 + f31*beta*I_C1* S_C3
    dRdt_C1 = gamma * I_C1 - eta * R_C1 - mi * R_C1
    dDdt_C1 = theta * I_C1
    dSdt_C2 = mi * (S_C2+C_C2+I_C2+R_C2) - alfa * S_C2 + delta * C_C2 - beta * S_C2 * I_C2 + eta * R_C2 - mi * S_C2 - f21*beta*I_C1* S_C2 - f23*beta*I_C3* S_C
    dCdt_C2 = alfa * S_C2 - delta * C_C2 - mi * C_C2 - beta_c * C_C2 * I_C2
    dIdt_C2 = beta * S_C2 * I_C2 - theta * I_C2 - gamma * I_C2 - mi * I_C2 + beta_c * C_C2 * I_C2 +  f12*beta*I_C2*S_C1  + f32*beta*I_C2*S_C3
    dRdt_C2 = gamma * I_C2 - eta * R_C2 - mi * R_C2
    dDdt_C2 = theta * I_C2
    dSdt_C3 = mi * (S_C3+C_C3+I_C3+R_C3) - alfa * S_C3 + delta * C_C3 - beta * S_C3 * I_C3 + eta * R_C3 - mi * S_C3 - f32*beta*I_C2*S_C3 - f31*beta*I_C1* S_C3
    dCdt_C3 = alfa * S_C3 - delta * C_C3 - mi * C_C3 - beta_c * C_C3 * I_C3
    dIdt_C3 = beta * S_C3 * I_C3 - theta * I_C3 - gamma * I_C3 - mi * I_C3 + beta_c * C_C3 * I_C3 + f23*beta*I_C3* S_C2 + f13*beta*I_C3*S_C1
    dRdt_C3 = gamma * I_C3 - eta * R_C3 - mi * R_C3
    dDdt_C3 = theta * I_C3
    return [dSdt_C1,dCdt_C1,dIdt_C1,dRdt_C1,dDdt_C1,dSdt_C2,dCdt_C2,dIdt_C2,dRdt_C2,dDdt_C2,dSdt_C3,dCdt_C3,dIdt_C3,dRdt_C3,dDdt_C3]
```

In [ ]:

```
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

# Integrate the equations over the time grid, t.
tspan = np.linspace(0,60,60)
#Integramos a função no tempo para confinamento de Araraquara de 35% em 60 dias
sol1 = odeint(deriv, y0, tspan, args=( beta, gamma, delta, eta, mi, alfa, theta, 0.35,beta_c, f12,f21,f23,f32,f31,f13))

tspan2 = np.linspace(60,75,15)
tspan21 = np.linspace(0,15,15)

y02 = [sol1.T[0][59],sol1.T[1][59],sol1.T[2][59],sol1.T[3][59],sol1.T[4][59],sol1.T[5][59],sol1.T[6][59],sol1.T[7][59],
       sol1.T[8][59],sol1.T[9][59],sol1.T[10][59],sol1.T[11][59],sol1.T[12][59],sol1.T[13][59],sol1.T[14][59]]
#Integramos a função no tempo para confinamento de Araraquara de 65% em 15 dias
sol2 = odeint(deriv, y02, tspan21, args=( beta, gamma, 0, eta, mi, alfa, theta, 0.60,beta_c,0,0,f23,f32,0,0))

tspan3 = np.linspace(75,100,80)
tspan31 = np.linspace(0,80,80)

y03 = [sol2.T[0][14],sol2.T[1][14],sol2.T[2][14],sol2.T[3][14],sol2.T[4][14],sol2.T[5][14],sol2.T[6][14],sol2.T[7][14],
       sol2.T[8][14],sol2.T[9][14],sol2.T[10][14],sol2.T[11][14],sol2.T[12][14],sol2.T[13][14],sol2.T[14][14]]
#Integramos a função no tempo para confinamento de Araraquara de 40% em 25 dias
sol3 = odeint(deriv, y03, tspan31, args=( beta, gamma, delta, eta, mi, alfa, theta, 0.35,beta_c,f12,f21,f23,f32,f31,f13))

#Junção dos tempos
tspan = tspan.tolist() + tspan2.tolist() + tspan3.tolist()
```

In [ ]:

```
#Junção dos dados em cada período de confinamento:
dSdt_C1 = sol1.T[0].tolist() + sol2.T[0].tolist() + sol3.T[0].tolist()
dCdt_C1 = sol1.T[1].tolist() + sol2.T[1].tolist() + sol3.T[1].tolist()
dIdt_C1 = sol1.T[2].tolist() + sol2.T[2].tolist() + sol3.T[2].tolist()
dRdt_C1 = sol1.T[3].tolist() + sol2.T[3].tolist() + sol3.T[3].tolist()
dDdt_C1 = sol1.T[4].tolist() + sol2.T[4].tolist() + sol3.T[4].tolist()
dSdt_C2 = sol1.T[5].tolist() + sol2.T[5].tolist() + sol3.T[5].tolist()
dCdt_C2 = sol1.T[6].tolist() + sol2.T[6].tolist() + sol3.T[6].tolist()
dIdt_C2 = sol1.T[7].tolist() + sol2.T[7].tolist() + sol3.T[7].tolist()
dRdt_C2 = sol1.T[8].tolist() + sol2.T[8].tolist() + sol3.T[8].tolist()
dDdt_C2 = sol1.T[9].tolist() + sol2.T[9].tolist() + sol3.T[9].tolist()
dSdt_C3 = sol1.T[10].tolist() + sol2.T[10].tolist() + sol3.T[10].tolist()
dCdt_C3 = sol1.T[11].tolist() + sol2.T[11].tolist() + sol3.T[11].tolist()
dIdt_C3 = sol1.T[12].tolist() + sol2.T[12].tolist() + sol3.T[12].tolist()
dRdt_C3 = sol1.T[13].tolist() + sol2.T[13].tolist() + sol3.T[13].tolist()
dDdt_C3 = sol1.T[14].tolist() + sol2.T[14].tolist() + sol3.T[14].tolist()
```

In [ ]:
```
#Plotamos todos os compartimentos:
plt.rcParams["figure.figsize"] = (20,10)
plt.subplot(1, 2, 1)
plt.plot(tspan, dSdt_C1, 'b', label='Susceptibles City1')
plt.plot(tspan, dSdt_C2, 'g', label='Susceptibles City2')
plt.plot(tspan, dSdt_C3, 'r', label='Susceptibles City3')

plt.legend(loc='best')
plt.xlabel('t')
plt.grid()

plt.subplot(1, 2, 2)
plt.plot(tspan, dCdt_C1 , 'b', label='Confined City1')
plt.plot(tspan, dCdt_C2 , 'g', label='Confined City2')
plt.plot(tspan, dCdt_C3 , 'r', label='Confined City3')

plt.legend(loc='best')
plt.xlabel('t')
plt.grid()
plt.show()
```

In [ ]:
```
plt.rcParams["figure.figsize"] = (20,10)
plt.subplot(1, 2, 1)
plt.plot(tspan, dIdt_C1 , 'b', label='Infected City1')
plt.plot(tspan, dIdt_C2 , 'g', label='Infected City2')
plt.plot(tspan, dIdt_C3 , 'r', label='Infected City3')

plt.legend(loc='best')
plt.xlabel('t')
plt.grid()

plt.subplot(1, 2, 2)
plt.plot(tspan, dRdt_C1 , 'b', label='Recovered City1')
plt.plot(tspan, dRdt_C2 , 'g', label='Recovered City2')
plt.plot(tspan, dRdt_C3 , 'r', label='Recovered City3')

plt.legend(loc='best')
plt.xlabel('t')
plt.grid()

plt.show()
```

In [ ]:
```
plt.plot(tspan, dDdt_C1 , 'b', label='Dead City1')
plt.plot(tspan, dDdt_C2 , 'g', label='Dead City2')
plt.plot(tspan, dDdt_C3 , 'r', label='Dead City3')

plt.legend(loc='best')
plt.xlabel('t')
plt.grid()
```

5 Analysis of the Results

From Test one Figure 6 we can observe that when city 1 is confined, cities 3 and 2 also have a growth in the confinement, this is expected since the flow to city 2 and 3 to city one is cut off, so more people now don't have the option to travel to city 1, The Susceptible class in city 1 as also expected, decreases significantly, since more people are confined, now cities 1 and 2 though have a decrease,also have a slow increase, this happens due to the fact that even though they can´t travel to city 1, still can travel between cities 2 and 3 and also maintain the confinement to 35%. What is interesting is that after the confinement period, all three cities have a significant decrease in the susceptible class.

In figure 7 the infected decrees almost to 0 during the confinement period, having a peek after such period which is naturally caused by the abrupt opening, but so happens to decrease significantly again and maintain quite low, which means the lockdown was successful in lowering the infectious class. The Recovered of city 1 have a slow decrease during the confinement period since there are less infected, as does happen in city 2 and 3, and both increase again after the confinement because of the infectious peak caused by the abrupt opening of commerce and transit, city 1 recovered decrease very slowly and city 2 increases probably due to the significant decrease of recovered on city 2 caused by its infected going almost to zero.

The accumulated dead in all three cities decrease during the confinement period, to then increase again slowly in all three tests, In tests two and three in particular, the accumulated deaths in city 3 stabilize due to the infected going almost to zero.

The difference in tests 1 and 2 lie in that, the behaviour of every compartment is that of test 1 though more enhanced, but there s not that much more of a difference, except on the stabilization of the dead in city 3. This is also observed comparing tests 1 and 3.

6 Conclusion

By analyzing the simulations, we can see that an effective lockdown for a period of 15 days and 60% confinement would have a good effect on the development of the disease, but I could not say the data is very conclusive, the infectious except on city 3 don´t have that much of a significant change, for future studies it would be interesting to compare the data between larger cities, and more homogeneous population number, also I find important to see the affect vaccination would bring to this dynamic.

7 Acknowledgments

I would like to thank Nelson Quispe Cuba for the important contribution to this research and my advisor João Frederico da Costa de Azevedo Meyer, this work would not have been possible without either of them.

References

[1] "Departamento de Estradas de Rodagem volume diario medio de trafego (vdm)." `http://200.144.30.103/vdm/SFCG_Concessionaria.asp?CodRodovia=`. Accessed: 2021-12-04.

[2] "IBGE population arrangements." `https://www.ibge.gov.br/en/geosciences/territorial-organization/regional-division/18116-population-arrangements-and-urban-concentrations-in-brazil.html?=&t=acesso-ao-produto`. Accessed: 2021-07-19.

[3] "WorldOMeter coronavirus (covid-19) mortality rate." https://www.worldometers.info/coronavirus/coronavirus-death-rate/. Accessed: 2021-12-04.