МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра прикладної математики

Звіт

із лабораторної роботи №2

з дисципліни «Теорія керування» на тему:

«Математичне моделювання епідемій»

|  |  |
| --- | --- |
| Виконав: | Керівник: |
| студент групи КМ-63 | *Професор Норкін В. І.* |
| *Артеменко Я. К.* |  |

Київ — 2020

# **ЗМІСТ**

[1 ВСТУП 2](#_Toc35981923)

[2 ОСНОВНА ЧАСТИНА 3](#_Toc35981924)

[2.1 Постановка задачі 3](#_Toc35981925)

[2.2 Описання моделі 4](#_Toc35981926)

[2.3 Порядок виконання роботи 6](#_Toc35981927)

[2.4 Контрольний приклад 6](#_Toc35981928)

[2.5 Дослідження змін 8](#_Toc35981929)

[3 ВИСНОВКИ 13](#_Toc35981930)

[СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 14](#_Toc35981931)

[ДОДАТКИ 15](#_Toc35981932)

[Додаток А (код програми) 15](#_Toc35981933)

# **1 ВСТУП**

Метою даної лабораторної роботи являється розробка програмного забезпечення для виконання математичного моделювання епідемій за допомогою SIR-моделі при заданих початкових умовах, візуалізація моделі та дослідження впливу початкових параметрів.

# **2 ОСНОВНА ЧАСТИНА**

# **2.1 Постановка задачі**

За допомогою будь-якої мови програмування потрібно побудувати та зобразити математичну модель епідемії за допомогою розв’язання задачі Коші для диференційних рівнянь.

Вводяться наступні величини:

N - загальне число індивідів (здорових без імунітету (susceptibles), хворих (infected);

S(t) - число здорових індивідів (без імунітету, susceptibles);

R(t) - число індивідів з імунітетом (здорових з імунітетом та тих, що видужали з придбаним імунітетом);

D(t) – число померлих;

u(t) – доля вакцинованих (ізольованих) в одиницю часу;

p(t)dt = p(I(t)) \* dt – qмовірність заразитися здоровому за час dt ;

S(t) \* p(I(t)) \* dt - середнє число заражених за час dt ;

q\*dt - ймовірність одужання інфікованого за час dt ;

q0 \* dt - ймовірність смерті інфікованого за час dt ;

u(t) \* dt - (керований) відсоток вакцинованих за час dt ;

S(t) \* u(t)dt - число вакцинованих за час dt.

Рекомендації до вибору параметрів моделі

Бажано орієнтуватися на епідемічну ситуацію з короно-вірусом COVID-19,

Наприклад,

r [1,50], (дослідити, як радикальне зменшення числа контактів впливає на хід епідемії)

с [0.5,0.9], (у COVID-19 дуже висока ймовірність передачі)

q [0.05,0.1], (у COVID-19 досить повільний процес одужання)

[0.01,0.1] , (відсоток смертельних випадків коливається від 1% до 10%)

u [0,0.1], (для COVID-19 практично ще немає вакцини)

N [103,106] , (як виглядає епідемічний процес для невеликих міст і для міст мільйонників)

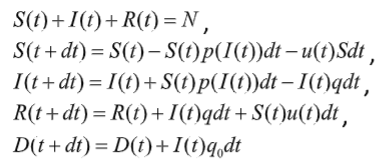
T [30,300], (моделювання потрібно вести до моменту закінчення епідемії).

# **2.2 Описання моделі**

Дана модель подається у вигляді диференційних рівнянь.

Нехай ті, що вже мали захворювання та видужали, набувають стійкого імунітету.

Тоді введені величини будуть пов’язані наступними рівняннями:



Звідси:

В матричній формі:



Початковими умовами для даної системи:

Позначимо:

* – вектор стану системи (або вектор фазових змінних),
* – управління (керована зміна), ,
* – множина допустимих керувань.

# **2.3 Порядок виконання роботи**

1) Обрати значення параметрів моделі відповідно їх змісту (орієнтуватися на пандемію COVID-19);

2) Обрати початкові значення;

3) Обрати проміжок часу [0, T];

4) Обрати програму для розв’язання нелінійної системи диференційних

рівнянь (solver, e.g. ode45 в системі Matlab);

5) Розв’язати задачу Коши на відрізку часу [0, T];

6) Візуалізувати розв’язок, тобто побудувати графіки функцій . На графіках вказати назву графіку, назву осей координат, легенду;

7) Дослідити зміни розвитку епідемії (картину) для різних значень

параметрів моделі ;

8) Підготувати звіт про роботу в електронному вигляді (з графіками,

висновками і лістингом програми);

9) Надіслати звіт викладачеві на електронну адресу.

# **2.4 Контрольний приклад**

Нехай початкові умови будуть наступними:

* N = 10000
* r = 50
* c = 0.85
* q = 0.09
* u = 0.05
* d = 0.04
* D0 = 0
* R0 = 10
* I0 = 97
* S0 = 10000 - 107

Тоді ми можемо підрахувати значення наступної формули:

Після виконання програми з вказаними початковими даними видно, що найінтенсивніші зміни можна спостерігати протягом перших 80 діб (рис. 2.1).

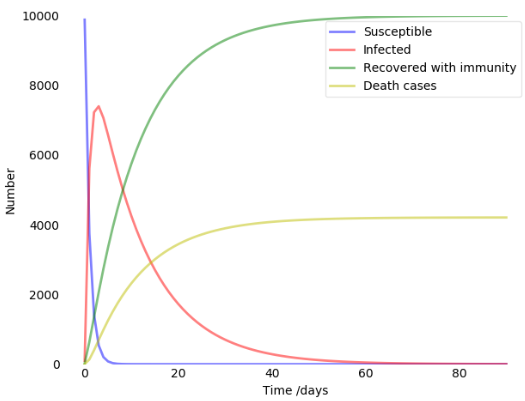


Рисунок 2.1 – Графічне представлення найінтенсивнішого періоду змін математичної моделі епідемії.

Але повне завершення епідемії можна очікувати через 365 діб (рис. 2.2).

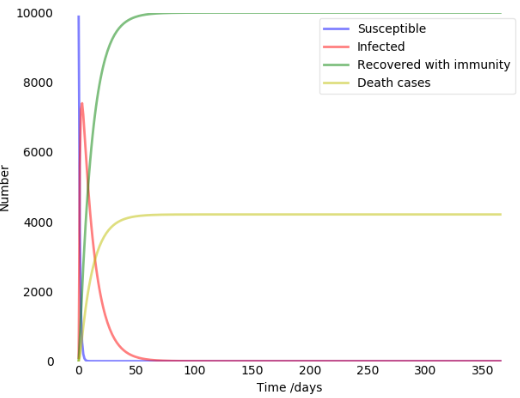


Рисунок 2.2 – Графічне представлення математичної моделі епідемії.

За цей час кількість померлих буде становити 4209.

# **2.5 Дослідження змін**

Далі будуть розглядатись різні випадки і як при певних даних буде реагувати програма при населенні в 10 000 жителів:

1. При зменшенні кількості контактів індивіда до 5.

В результаті отримаємо, що кількість померлих буде 2889, а сама епідемія завершиться через 354 дні.

Найінтенсивніші зміни можна спостерігати протягом перших 80 діб (рис. 2.3).

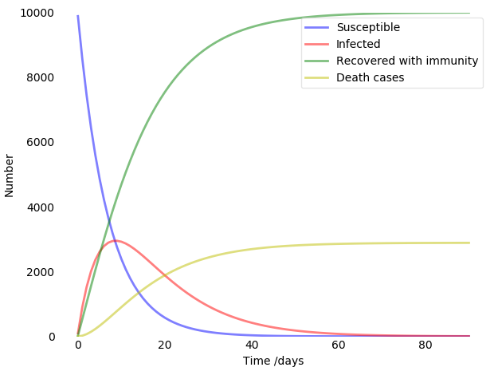


Рисунок 2.3 – Графічне представлення найінтенсивнішого періоду змін математичної моделі епідемії за умови зменшення контактів індивідів.

Але повне завершення епідемії можна очікувати через 354 доби (рис. 2.4).

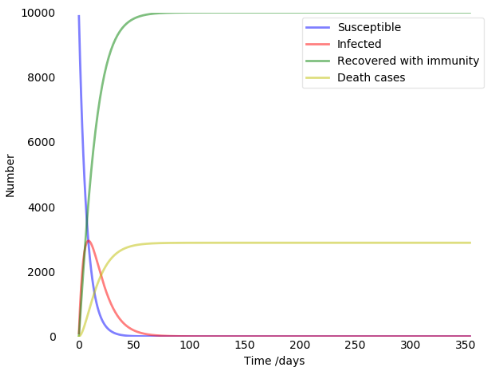


Рисунок 2.4 - Графічне представлення математичної моделі епідемії за умови зменшення контактів індивідів.

1. Збільшення інтенсивності вакцинування до 0.09, коли кількість контактів r = 50.

В результаті отримаємо, що кількість померлих буде 4044, а сама епідемія завершиться через 319 днів.

Найінтенсивніші зміни можна спостерігати протягом перших 80 діб (рис. 2.5).

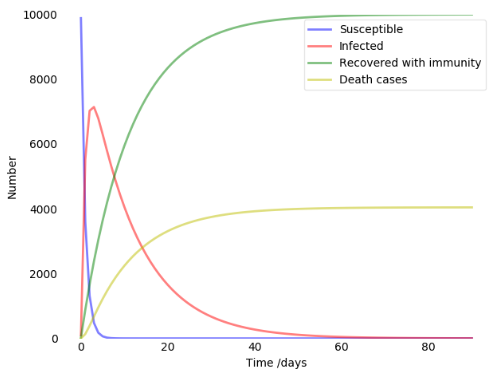


Рисунок 2.5 - Графічне представлення найінтенсивнішого періоду змін математичної моделі епідемії за умови збільшення інтенсивності вакцинації.

Але повне завершення епідемії можна очікувати через 319 діб (рис. 2.6).

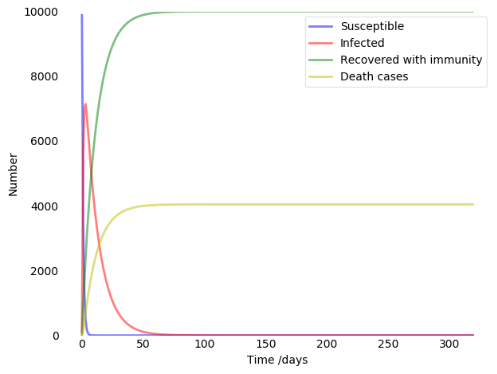


Рисунок 2.6 - Графічне представлення математичної моделі епідемії за умови збільшення інтенсивності вакцинації.

1. При збільшенні інтенсивності вакцинування до 0.09 та при зменшенні кількості контактів r до 5.

В результаті отримаємо, що кількість померлих буде 2263, а сама епідемія завершиться через 349 днів.

Найінтенсивніші зміни можна спостерігати протягом перших 80 діб (рис. 2.7).

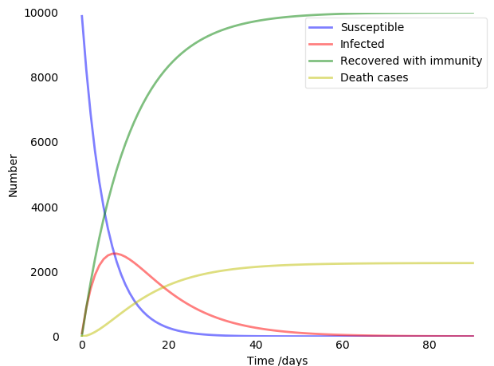


Рисунок 2.7 - Графічне представлення найінтенсивнішого періоду змін математичної моделі епідемії за умови збільшення інтенсивності вакцинації та зменшення контактів індивідів.

Але повне завершення епідемії можна очікувати через 349 діб (рис. 2.6)

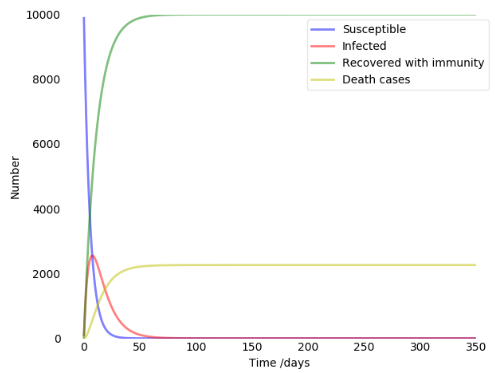


Рисунок 2.8 - Графічне представлення математичної моделі епідемії за умови збільшення інтенсивності вакцинації та зменшення контактів індивідів.

# **3 ВИСНОВКИ**

Під час виконання даної лабораторної роботи було розроблено програмне забезпечення для створення математичного моделювання епідемії із заданими початковими параметрами. В пункті «2.5 Дослідження змін» видно, як впливають різні значення вхідних параметрів на результат виконання програми. Також видно те, що самим ефективним методом боротьби з епідемією являється менша кількість контактів з іншими індивідами та вища інтенсивність вакцинування населення.

# **СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**

1. Методичні вказівки до лабораторної роботи.
2. A. Huppert, G. Katriel, Mathematical modelling and prediction in infectious disease epidemiology, Clin. Microbiol. Infection 19 (2013), 999-1005.
3. Яушева О.А. 2016 - Математическая модель эпидемии лихорадки Эбола (SIR модель). Санкт-Петербургский государственный университет, Кафедра диагностики функциональных систем. Санкт-Петербург, 2016.
4. И.Д.Колесин, Е.М.Житкова. Математические модели эпидемий: Учебное пособие. - СПб.: НИИФ СПбГУ, 2004. - с.92.

# **ДОДАТКИ**

# **Додаток А (код програми)**

import numpy as np  
import matplotlib.pyplot as plt  
  
from math import log  
from scipy.integrate import odeint  
  
def deriv(y, t, N, p, q, u, d):  
 S, I, R, D = y  
 dSdt = -S \* p - S \* u  
 dIdt = S \* p - q \* I  
 dRdt = q \* I + u \* S  
 dDdt = I \* d  
 return dSdt, dIdt, dRdt, dDdt  
  
N = 10000 # населення  
  
r = 50 # кількість контактів  
c = 0.85 # ймовірність передачі  
q = 0.09 # процес одужання  
u = 0.05 # вакцинування  
d = 0.04 # ймовірність смертельного випадку  
  
D0 = 0 # кількість смертей (0)  
R0 = 20 # кількість індивідів з імунітетом  
I0 = 97 # кількість хворих (I > 0)  
S0 = 10000 - 117 # кількість здорових (S0 < N)  
  
  
  
p = -(r\*I0\*log(1 - c))/N  
print(p)  
  
t = np.linspace(0, 500, 501)  
t1 = np.linspace(0, 90, 91)  
  
y0 = S0, I0, R0, D0  
  
ret = odeint(deriv, y0, t, args=(N, p, q, u, d))  
S, I, R, D = ret.T  
for i in zip(t, I):  
 if i[1] <= 0:  
 e = int(i[0])  
 print(f'Кількість смертей - {D[e]}')  
 y = np.linspace(0, e, e+1)  
 print(e)  
 break  
  
t2 = np.linspace(0, e, e+1)  
  
fig = plt.figure(facecolor='w')  
ax = fig.add\_subplot(111, axisbelow=True)  
ax.plot(t1, S[:len(t1)], 'b', alpha=0.5, lw=2, label='Susceptible')  
ax.plot(t1, I[:len(t1)], 'r', alpha=0.5, lw=2, label='Infected')  
ax.plot(t1, R[:len(t1)], 'g', alpha=0.5, lw=2, label='Recovered with immunity')  
ax.plot(t1, D[:len(t1)], 'y', alpha=0.5, lw=2, label='Death cases')  
ax.set\_xlabel('Time /days')  
ax.set\_ylabel('Number')  
ax.set\_ylim(0,N)  
ax.yaxis.set\_tick\_params(length=0)  
ax.xaxis.set\_tick\_params(length=0)  
ax.grid(b=True, which='major', c='w', lw=2, ls='-')  
legend = ax.legend()  
legend.get\_frame().set\_alpha(0.5)  
for spine in ('top', 'right', 'bottom', 'left'):  
 ax.spines[spine].set\_visible(False)  
plt.show()  
  
fig = plt.figure(facecolor='w')  
ax = fig.add\_subplot(111, axisbelow=True)  
ax.plot(t2, S[:len(t2)], 'b', alpha=0.5, lw=2, label='Susceptible')  
ax.plot(t2, I[:len(t2)], 'r', alpha=0.5, lw=2, label='Infected')  
ax.plot(t2, R[:len(t2)], 'g', alpha=0.5, lw=2, label='Recovered with immunity')  
ax.plot(t2, D[:len(t2)], 'y', alpha=0.5, lw=2, label='Death cases')  
ax.set\_xlabel('Time /days')  
ax.set\_ylabel('Number')  
ax.set\_ylim(0, N)  
ax.yaxis.set\_tick\_params(length=0)  
ax.xaxis.set\_tick\_params(length=0)  
ax.grid(b=True, which='major', c='w', lw=2, ls='-')  
legend = ax.legend()  
legend.get\_frame().set\_alpha(0.5)  
for spine in ('top', 'right', 'bottom', 'left'):  
 ax.spines[spine].set\_visible(False)  
plt.show()