

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики

Кафедра Обчислювальної Математики

Курсова робота

За спеціальністю 113 "Прикладна математика"

на тему:

**Моделювання розвитку епідемії
з використанням чисельних методів**

Виконав студент 3-го курсу

Коломієць Микола

Науковий керівник:

Тимошенко Андрій. . .

Київ, 2023

ЗМІСТ

1	SIR модель	3
1.1	SIR модель	3
1.2	SEIR модель	4
1.3	SEI модель	5
1.4	SEIPR	5
1.5	SEIAPR	5
1.6	SEIPHR	6
1.7	SEIPHRD	6
1.8	Смертність	6
1.9	SIR-подібні моделі з вакцинацією	7
2	Чисельні методи	8
2.1	Методи вирішення системи диф рівнянь	8
2.1.1	Метод Ейлера	8
2.1.2	Класичний метод Рунге Кутті	9
2.2	Методи підбору параметрів	11
2.2.1	Метод рою часток	11
3	Оцінка і порівняння моделей	14
3.1	Справжні дані та оцінка ефективності	14
3.2	SIR	14
3.3	SEIR	14
3.4	SEIPR	14
3.5	SEIAPR	14
3.6	SEIAPHR	14
3.7	SEIAPHRD	14
3.8	Порівняння моделей та результати	14

ВСТУП

Оцінка сучасного стану об'єкта дослідження. Актуальність роботи та підстави для її виконання. Мета й завдання роботи. Метою є Об'єкт, методи та засоби дослідження. Можливі сфери застосування.

РОЗДІЛ 1 SIR МОДЕЛЬ

В першому розгляді розглянемо SIR-подібні моделі. Почнемо з базової SIR моделі і поступово будемо ускладнювати модель шляхом додавання нових груп, що дозволить поступово наблизити модель до реалій протікання хвороби.

1.1 SIR модель

Одною з найпростіших моделей є SIR модель. В ній всі люди розглядаються у вигляді трьох груп: здорові (susceptible), хворі (infectious) та ті, що одужали (removed). Форсування людей між цими групами відбувається зі здорових у хворі і з хворих до одужаних.(див рис. 1.1)

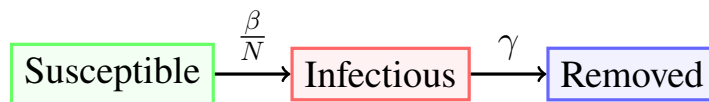


Рис. 1.1: SIR model

Зв'язок між цими групами записують за допомогою системи диференціальних рівнянь

$$\begin{cases} \frac{dS}{dt} = -\frac{\beta}{N}SI \\ \frac{dI}{dt} = \frac{\beta}{N}SI - \gamma I \\ \frac{dR}{dt} = \gamma I \end{cases}$$

Перше рівняння описує, що кількість здорових зменшується на величину пропорційну добутку хворих на здорових - кількість можливих контактів. Друге рівняння показує, що кількість хворих зростає на ту саму величину і зменшується на величину пропорційну до кількості хворих (кількість одужаних). Третє рівняння збільшується на ту саму кількість одужаних[1].

Додавши всі рівняння системи отримаємо - $\frac{dS}{dt} + \frac{dI}{dt} + \frac{dR}{dt} = 0$. Тож населення у нашій моделі не змінюється з часом: $N = S + I + R = const$.

1.2 SEIR модель

Часто хвороба характеризується не лише станами - хворий, здоровий, перехворівший. Для подібних випадків додають додаткові стани. Наприклад, якщо додати у SIR модель ще одну групу «заражені, але ще не взмозі інфікувати інших» то вийде SEIR модель. Тоді у порівнянні з SIR моделлю у схему додається проміжний етап. (див рис. 1.2)

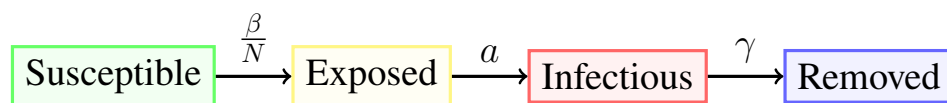


Рис. 1.2: Схема роботи SIER моделі

У вихідну систему додається додаткове рівняння

$$\begin{cases} \frac{dS}{dt} = -\frac{\beta}{N}SI \\ \frac{dE}{dt} = \frac{\beta}{N}SI - aE \\ \frac{dI}{dt} = aE - \gamma I \\ \frac{dR}{dt} = \gamma I \end{cases}$$

$E(t)$ функція кількості заражених, коефіцієнт a відповідає оберненому середньому періоду переходу людини від зараженої, до повноцінного інфікованого. Так само населення залишається незмінним: $N = S + E + I + R$.

Подіюні прийоми застосовують для більш точного моделювання відповідно до протікання хвороби - часто є певний період під час якого інфікований, ще не заражає оточуючих або заражає з меншим шансом, більш тісним контактом, або період до проявів перших симптомів і початком домашньої ізоляції, носіння маски. Все це впливає на параметри та рівняння моделі і на точність результату. Буде складніше підібрати вірні параметри, але це може позитивно вплинути на точність.

1.3 SEI модель

Іноді моделі не ускладнюють, а спрощують. Наприклад, якщо для хвороби виробити імунітет навічно не можливо, як от нежить, одужаних розглядати немає сенсу. Таким чином схема спрощується (див рис. 1.3)

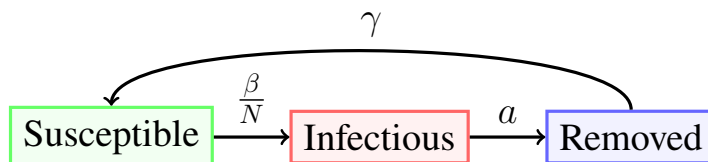


Рис. 1.3: Схема роботи SEI моделі

Отже і система буде складатись з меншої кількості рівнянь.

$$\begin{cases} \frac{dS}{dt} = -\frac{\beta}{N}SI + \gamma I \\ \frac{dE}{dt} = \frac{\beta}{N}SI - aE \\ \frac{dI}{dt} = aE - \gamma I \end{cases}$$

1.4 SEIPR

$$\begin{cases} \frac{dS}{dt} = -\beta \frac{I}{N}S - \beta' \frac{P}{N}S, \\ \frac{dE}{dt} = \beta \frac{I}{N}S + \beta' \frac{P}{N}S - \kappa E, \\ \frac{dI}{dt} = \kappa \rho_1 E - \gamma_i I, \\ \frac{dP}{dt} = \kappa (1 - \rho_1) E - \gamma_i P, \\ \frac{dR}{dt} = \gamma_i (I + P) \end{cases}$$

1.5 SEIAPR

$$\begin{cases} \frac{dS}{dt} = -\beta \frac{I}{N}S - \beta' \frac{P}{N}S, \\ \frac{dE}{dt} = \beta \frac{I}{N}S + \beta' \frac{P}{N}S - aE, \\ \frac{dI}{dt} = a \rho_1 E - \gamma I, \\ \frac{dP}{dt} = a \rho_2 E - \gamma P, \\ \frac{dA}{dt} = a (1 - \rho_1 - \rho_2) E, \\ \frac{dR}{dt} = \gamma (I + P) \end{cases}$$

1.6 SEIPHR

1.7 SEIPHRD

$$\left\{ \begin{array}{l} \frac{dS}{dt} = -\beta \frac{I}{N} S - l\beta \frac{H}{N} S - \beta' \frac{P}{N} S, \\ \frac{dE}{dt} = \beta \frac{I}{N} S + l\beta \frac{H}{N} S + \beta' \frac{P}{N} S - aE, \\ \frac{dI}{dt} = a\rho_1 E - (\gamma_a + \gamma_i) I - \delta_i I, \\ \frac{dP}{dt} = a\rho_2 E - (\gamma_a + \gamma_i) P - \delta_p P, \\ \frac{dA}{dt} = a(1 - \rho_1 - \rho_2) E, \\ \frac{dH}{dt} = \gamma_a(I + P) - \gamma_r H - \delta_h H, \\ \frac{dR}{dt} = \gamma_i(I + P) + \gamma_r H, \\ \frac{dF}{dt} = \delta_i I + \delta_p P + \delta_h H, \end{array} \right.$$

1.8 Смертність

Якщо зробити припущення, що людина у кожній групі може померти з ймовірністю μ , то кожна група буде зменшуватися на добуток μ та популяцію цієї групи. В таких моделях також додають до розгляду новонароджених. Якщо припустити, що у середньому народжуваність за момент часу dt дорівнює Λ , одержимо нуву систему рівнянь.

$$\left\{ \begin{array}{l} \frac{dS}{dt} = \Lambda - \mu S - \frac{\beta}{N} SI \\ \frac{dI}{dt} = \frac{\beta}{N} SI - \gamma I - \mu I \\ \frac{dR}{dt} = \gamma I - \mu R \end{array} \right.$$

У даній моделі населення вже буде змінюватись.

$$\frac{dN(t)}{dt} = \Lambda - \mu N(t) = \Lambda - \mu(S(t) + I(t) + R(t))$$

Іноді для спрощення встановлюють $\Lambda = \mu$ - за таких значень населення не змінюється, проте трохи збільшується потенціал інфікування (з приростом нового покоління у категорію здорових)

1.9 SIR-подібні моделі з вакцинацією

$$\begin{cases} \frac{dS}{dt} = vN(1 - P) - \frac{\beta}{N}SI \\ \frac{dI}{dt} = \frac{\beta}{N}SI - \gamma I \\ \frac{dR}{dt} = \gamma I \\ \frac{dV}{dt} = vNP \end{cases}$$

РОЗДІЛ 2 ЧИСЕЛЬНІ МЕТОДИ

У другому розділі розглянемо чисельні методи для розв’язання даних систем диф рівнянь та для визначення оптимальних параметрів, що якнайкраще симулюють розвиток захворюваності.

2.1 Методи вирішення системи диф рівнянь

Розв’язати подібні системи в загальному вигляді від параметрів не можна через нелінійність та не стійкість розв’язків, тож у данному розділі ми розглянемо деякі чисельні методи для розв’язку подібних систем відносно наперед відомих параметрів.

2.1.1 Метод Ейлера

Метод Ейлера найбільш базовий чисельний метод для розв’язання нелінійних систем диференціальних рівнянь, що застосовується при відомих початкових значеннях. В данному випадку нам відома кількість хворих на початку епідемії. Це частковий випадок методів Рунге Кутті. Загалом метод можна описати формулою $dS \approx S_{n+1} - S_n$. Зазвичай береться певний крок dt і диференціальне рівняння апроксимується відносно нього:

$$\frac{dS}{dt} = f(t, S) \Rightarrow S_{n+1} = S_n + f(t_n, S_n) dt$$

Ітерації починаються при відомому S_0 . Метод є не стабільним при великому кроці dt .

Як приклад візьмемо SIR модель при розв’язанні моделі отримаємо:

$$\begin{cases} S_{n+1} = S_n - \frac{\beta}{N} S_n I_n dt \\ I_{n+1} = I_n + \frac{\beta}{N} S_n I_n dt - \gamma I_n dt \\ R_{n+1} = R_n + \gamma I_n dt \\ S_0 = N - 1, I_0 = 1, R_0 = 0 \end{cases}$$

Функція написана на С для отримання даних з моделі відносно її параметрів кроку та початкових умов.

```

1 float** sir(int size, float beta, float gama, int N, float dt){
2     float* S = malloc(size * sizeof(float));
3     float* I = malloc(size * sizeof(float));
4     float* R = malloc(size * sizeof(float));
5     float** answer = malloc(3 * sizeof(float*));
6     answer[0] = S; answer[1] = I; answer[2] = R;
7     S[0] = N - 1; I[0] = 1; R[0] = 0; answer[0] = 1;
8     int step = (int)(1 / dt); float sum = 1;
9     for (int i = 0; i < size - 1; i++){
10         S[i + 1] = S[i] - beta * S[i] * I[i] * dt / N;
11         I[i + 1] = I[i] + (beta * S[i] * I[i] / N - gama * I[i]) * dt;
12         R[i + 1] = R[i] + gama * I[i] * dt;
13     return answer;
14 }

```

Мова програмування С була обрана для написання цієї функції завдяки її швидкості виконання та відносної простоти написання коду. Потреба у високій швидкості виконання обумовлена потребами задля швидкого знаходження оптимальних параметрів - через складну залежність від параметрів моделі використати градієнтні методи не вийде, а не градієнтні методи можуть багато разів викликати данну функцію на кожній ітерації.

Функції, що розв'язують інші моделі відрізняються лише кількістю масивів та основним циклом, загальна структура функцій однакова. Час виконання програми такий то.

2.1.2 Класичний метод Рунге Кутті

Ідеологічно методи Рунге Кутті це алгоритми влучного застосування методу Ейлера певну кількість разів. Від кількості разів, яку він застосовується залежить порядок методу. Наприклад класичним методом Рунге Кутті називають саме метод четвертого порядку.

Нехай у нас є система диференціальних рівнянь $\frac{dx}{dt} = f(x, t)$. Тоді,

якщо скористатися методом Ейлера отримаємо $x_{k+1} = x_k + f(x_k, t_k)$. Замість цього ми рахуємо лише доданок $k_1 = f(x_k, t_k)$. Далі рахуємо, ще декілька доданків

$$\begin{cases} k_1 = f(x_k, t_k) \\ k_2 = f(x_k + k_1 \frac{dt}{2}, t_k + \frac{dt}{2}) \\ k_3 = f(x_k + k_2 \frac{dt}{2}, t_k + \frac{dt}{2}) \\ k_4 = f(x_k + k_3 dt, t_k + dt) \end{cases}$$

І вже тоді додаємо, отримуючи наближене значення

$$x_{k+1} = x_k + \frac{dt}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Розглянемо роботу методу на звичайній SIR моделі. Для простоти введемо функції:

$$\begin{cases} f_1(S_k, I_k, R_k) = -\beta \frac{S_k}{N} I_k \\ f_2(S_k, I_k, R_k) = \beta \frac{S_k}{N} I_k - \gamma I_k \\ f_3(S_k, I_k, R_k) = \gamma I_k \end{cases}$$

Розрахунок наближень 4-рьох порядків:

$$\begin{cases} s_1 = f_1(S_k, I_k, R_k) \\ i_1 = f_2(S_k, I_k, R_k) \\ r_1 = f_3(S_k, I_k, R_k) \end{cases}$$

$$\begin{cases} s_2 = f_1(S_k + \frac{dt}{2}s_1, I_k + \frac{dt}{2}i_1, R_k + \frac{dt}{2}r_1) \\ i_2 = f_2(S_k + \frac{dt}{2}s_1, I_k + \frac{dt}{2}i_1, R_k + \frac{dt}{2}r_1) \\ r_2 = f_3(S_k + \frac{dt}{2}s_1, I_k + \frac{dt}{2}i_1, R_k + \frac{dt}{2}r_1) \end{cases}$$

$$\begin{cases} s_3 = f_1(S_k + \frac{dt}{2}s_2, I_k + \frac{dt}{2}i_2, R_k + \frac{dt}{2}r_2) \\ i_3 = f_2(S_k + \frac{dt}{2}s_2, I_k + \frac{dt}{2}i_2, R_k + \frac{dt}{2}r_2) \\ r_3 = f_3(S_k + \frac{dt}{2}s_2, I_k + \frac{dt}{2}i_2, R_k + \frac{dt}{2}r_2) \end{cases}$$

$$\begin{cases} s_4 = f_1(S_k + dt s_3, I_k + dt i_3, R_k + r_3) \\ i_4 = f_2(S_k + dt s_3, I_k + dt i_3, R_k + r_3) \\ r_4 = f_3(S_k + dt s_3, I_k + dt i_3, R_k + r_3) \end{cases}$$

Отримане наближення:

$$\begin{cases} S_{k+1} = S_k + \frac{dt}{6}(s_1 + 2s_2 + 2s_3 + s_4) \\ I_{k+1} = I_k + \frac{dt}{6}(i_1 + 2i_2 + 2i_3 + i_4) \\ R_{k+1} = R_k + \frac{dt}{6}(r_1 + 2r_2 + 2r_3 + r_4) \end{cases}$$

Задля застосування данного методу на інших моделях треба лише додати потрібні допоміжні функції та розрахунок додаткових змінних, принцип залишається незмінним.

2.2 Методи підбору параметрів

Тепер коли ми можемо порахувати різницю між прогнозованою величиною, яку видає наша модель і реальними даними, можемо використати алгоритми оптимізації для підбору оптимальних параметрів. За функцію ціни візьмемо відстань між графіками реальних даних та прогнозованих. Розглянемо лише неградієнтні методи оптимізації, адеж порахувати градієнт функції наближеної методами з першої частини глави не можна через складність обрахунків та кількість задіяної пам'яті.

2.2.1 Метод рою часток

Метод рою частинок не передбачає розрахунок градієнта функції, тому його швидкість роботи більш ніж задовільняє нашу задачу. Данний метод використовує "частинки кожна з яких символізує один розв'язок задачі. На кожній ітерації частинка намагається рухатись у оптимальному напрямі, тут присутній елемент стахастики адеж випадковим є те чи буде вона рухатись до глобального оптимума (за весь час роботи алгоритма) чи до локального оптимума данної частинки. Функцію зміни параметрів частинки можна описати такою формулою:

$$\Delta x_{k+1}^{(i)} = \alpha \Delta x_k^{(i)} + \beta (x_{best}^{(i)} - x_k) + \gamma (x_{best} - x_k^{(i)})$$

У формулі $x_{best}^{(i)}$ найкраще значення цієї частинки, x_{best} найкраще значення всіх частинок.

Загальна схема алгоритму виглядає так:

Лістинг 2.1: Python example

```
def roy(f, n, N, iteration, w, a1, a2, A, B, eps, first):
    #
    X = np.array([[random.uniform(A[i], B[i])
                    for i in range(n)] for j in range(N)])
    #
    V = np.array([[random.uniform(-eps, eps) for i in range(n)
                    for j in range(N)])])
    #
    #
    for i in range(n):
        X[0][i] = first[i]
    #
    P = np.copy(X)
    #
    res = np.array([f(X[i]) for i in range(N)])
    #
    b = P[np.argmin(res)]
    #
    for i in tqdm(range(iteration)):
        #
        V = w * V + a1 * random.random() * (P - X) +
            a2 * random.random() * (b - X)
        #
        X = X + V
        #
        #
        for j in range(N):
```

```

        res1 = f(X[j])
        if res1 < res[j]:
            P[j] = X[j]
            res[j] = res1
#
    b = P[np.argmin(res)]
#
    return b

```

РОЗДІЛ 3 ОЦІНКА І ПОРІВНЯННЯ МОДЕЛЕЙ

У ході курсової роботи для оцінки та візуалізації моделей була створена програма, яка дозволяє користувачу регулювати параметри та одразу бачити, як вони впливають на прогнози. Також реалізована кнопка, яка починає оптимізацію початкових параметрів, налаштованих користувачем, методом рою часток (одна з часток ініціалізована початковими параметрами). Також є опція зберегти параметри у форматі json. Основна програма написана на мові програмування Python, функції поділювання написані на C задля збільшення швидкості оптимізації.

3.1 Справжні дані та оцінка ефективності

Прогнозувати будемо розвиток коронавірусу у Люксембургу через те, що країна досить не велика та має середню густоту населення. Підбирати параметри будемо за перші 70 днів епідемії і відповідний прогноз будемо робити на наступні два тижні.

3.2 SIR

3.3 SEIR

3.4 SEIPR

3.5 SEIAPR

3.6 SEIAPHR

3.7 SEIAPHRD

3.8 Порівняння моделей та результати

БІБЛІОГРАФІЯ

- [1] Ayoob Salimipour, Toktam Mehraban, Hevi Seerwan Ghafour, Noreen Izza Arshad, and M. J. Ebadi. SIR model for the spread of COVID-19: A case study. *Operations Research Perspectives*, 10:100265, 2023. Publisher: Elsevier.