

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОСТРОЗЬКА АКАДЕМІЯ»**

**Моделювання та аналіз процесів інформаційної війни  
як інструмент захисту демократії в ЄС і протидії загрозам  
(101238591 – EUDemCT – ERASMUS-JMO-2025-HEI-TCH-RSCH)**

**Нікітін А. В., Красюк Б. В.**

**МОДЕЛЮВАННЯ ТА АНАЛІЗ  
ПРОЦЕСІВ ІНФОРМАЦІЙНОЇ ВІЙНИ  
ЯК ІНСТРУМЕНТ ЗАХИСТУ ДЕМОКРАТІЇ В ЄС  
І ПРОТИДІЇ ЗАГРОЗАМ**

**Дидактичні матеріали**

*(«Математичне та комп'ютерне моделювання процесів  
інформаційної боротьби» та «Математичне моделювання  
інформаційних процесів з використанням Python»)*

Острого  
2026

УДК 004.9:355.40:321.7(4-6ЄС)(075.8)  
ББК 66.4(4ЄС),9 + 32.81я73  
Н 93

*Рекомендовано до друку  
на засіданні кафедри математики та інтелектуальних обчислень  
Національного університету «Острозька академія»  
(протокол № 2 від 18 лютого 2026 року)*

Н 93

**Нікітін А. В., Красюк Б. В.**

Моделювання та аналіз процесів інформаційної війни як інструмент захисту демократії в ЄС і протидії загрозам: Дидактичні матеріали «Математичне моделювання інформаційних процесів з використанням Python». Острог: Видавництво Національного університету «Острозька академія», 2026. 118 с.

DOI 10.25264/18.02.2026

*Дидактичні матеріали підготовлені в межах проекту «Моделювання та аналіз процесів інформаційної війни як інструмент захисту демократії в ЄС і протидії загрозам» (101238591 – EUDemCT – ERASMUS-JMO-2025-HEI-TCH-RSCH) реалізується Національним університетом «Острозька академія» за співфінансування Європейського Союзу та спрямований на усунення критичних прогалин у розумінні механізмів дезінформації, відсутності стандартизованих контрстратегій та низької обізнаності громадськості про інформаційні загрози. Визнаючи інформаційну війну як зростаючий виклик для демократичних інститутів в Україні та ЄС, проект об'єднує освіту, дослідження і міжгалузеву співпрацю з метою підвищення стійкості до гібридних загроз.*

Більше інформації на сайті проекту: <https://eudemct.ua.edu.ua/>

УДК 004.9:355.40:321.7(4-6ЄС)(075.8)  
ББК 66.4(4ЄС),9 + 32.81я73

**Disclaimer:**

*Фінансується Європейським Союзом. Однак висловлені погляди та думки є виключно поглядами та думками автора (авторів) і не обов'язково відображають позицію Європейського Союзу або Європейського виконавчого агентства з питань освіти та культури (EACEA). Європейський Союз та EACEA не несуть відповідальності за них.*



**Co-funded by  
the European Union**

**Назва проєкту:**

Моделювання та аналіз процесів інформаційної війни як інструмент захисту демократії в ЄС і протидії загрозам (101238591- EUDemCT – ERASMUS-JMO-2025-HEI-TCH-RSCH)

**Термін реалізації проєкту:**

01.09.2025-31.08.2028

**Бюджет проєкту:**

13 500,00 євро

**Координатор проєкту:**

Національний університет «Острозька академія»

Проєкт EUDemCT спрямований на усунення критичних прогалин у розумінні механізмів дезінформації, відсутності стандартизованих контрстратегій та низької обізнаності громадськості про інформаційні загрози. Визнаючи інформаційну війну як зростаючий виклик для демократичних інститутів в Україні та ЄС, проєкт об'єднує освіту, дослідження і міжгалузеву співпрацю з метою підвищення стійкості до гібридних загроз.

**Ключові цілі та очікувані результати:**

- Бакалаврський курс з проблематики інформаційної війни та політики безпеки ЄС – протягом трьох років навчання для понад 75 студентів у галузі кібербезпеки, гібридних загроз та обчислювального аналізу, надання їм практичних навичок у сфері захисту даних і цифрової безпеки.
- Навчальні модулі з кібербезпеки і стратегічного прийняття рішень – розроблення міждисциплінарного курсу з моделювання та аналізу інформаційної війни для посилення академічних навчальних програм.
- Семінари для освітян – навчання 180 шкільних та університетських викладачів розпізнавати та протидіяти цифровій дезінформації, покращуючи освіту в галузі медіаграмотності.
- Наукові публікації – підготовка 2 рецензованих статей про інформаційну війну, прийняття рішень та гібридні загрози в міжнародних журналах з кібербезпеки і політичних наук.
- Цифрова платформа поширення інформації – створення вебсайту для розміщення результатів досліджень, а також навчальних матеріалів та аналітичних оглядів; залучає понад 5000 відвідувачів.

Інтегруючи кібербезпеку, політичні науки та стратегічний аналіз, проєкт EUDemCT зміцнює співпрацю між ЄС та Україною, сприяє демократичній стійкості та підвищує інституційний потенціал для протидії гібридним загрозам.

## **МОДУЛЬ 1**

### **Лекція 1: Європейська інтеграція та шлях України до ЄС**

Історичний контекст розширення ЄС та європейські прагнення України. Ключові віхи: Угода про асоціацію, лібералізація візового режиму та статус кандидата в члени ЄС. Виклики та можливості процесу вступу України до ЄС. Вплив війни на інтеграцію України до ЄС.

### **Лекція 2: Політика безпеки ЄС та роль України в європейській оборонній архітектурі**

Спільна політика безпеки та оборони ЄС (СПБО). Гібридні загрози та кібербезпека: уроки з досвіду України. Роль України у зміцненні європейської безпеки.

### **Лекція 3: Економічні виклики та можливості у відносинах між ЄС та Україною**

Торговельна та економічна інтеграція в рамках поглибленої та всеосяжної зони вільної торгівлі (ПВЗВТ). Роль фінансової та гуманітарної допомоги ЄС у відновленні України. Стратегії відбудови та інвестиційні можливості в післявоєнній Україні.

### **Лекція 4: Демократія, верховенство права та реформи врядування в Україні**

Копенгагенські критерії ЄС та прогрес України в реформах. Антикорупційні заходи та судові реформи: досягнення та виклики. Децентралізація та місцеве врядування в Україні: уроки з практики ЄС. Роль громадянського суспільства у просуванні демократичних цінностей.

### **Лекція 5: Підхід ЄС до гібридної війни та досвід України у протидії дезінформації**

Стратегічна комунікація ЄС та реагування на гібридні загрози. Російська дезінформація та операції з впливу в Україні та ЄС. Досвід України у протидії інформаційній війні: успіхи та виклики. Рекомендації щодо політики для зміцнення співпраці між ЄС та Україною в інформаційному просторі.

## **МОДУЛЬ 2**

### **Лекція 1: Вступ до математичного моделювання**

Основи математичних моделей: цілі та підходи. Огляд прикладних проблем інформаційної війни. Python як інструмент моделювання: огляд основних бібліотек (NumPy, SciPy, Matplotlib).

### **Лекція 2: Основи моделювання динамічних систем**

Диференціальні рівняння як основа моделей: Лотки-Вольтерри, Гомперца. Приклади моделей: інформаційна війна, поширення епідемії. Реалізація моделей динамічних систем у Python.

### **Лекція 3: Стохастичні процеси та випадкові збурення**

Вступ до теорії ймовірностей та стохастичних систем. Марківські процеси, стохастичні диференціальні рівняння. Розв'язання стохастичних систем у Python за допомогою NumPy.

### **Лекція 4: Асимптотичний аналіз стохастичних систем**

Усереднення, дифузійне наближення, пуассонівське наближення. Приклади дифузійних моделей. Оптимізація обчислень для довгострокових симуляцій.

### **Лекція 5: Оптимізація обчислень у Python**

Профілювання та оптимізація коду. Використання NumPy для ефективної роботи з великими наборами даних. Векторизація обчислень та паралельна обробка.

### **Лекція 6: Візуалізація результатів моделювання**

Візуалізація даних за допомогою Matplotlib і Plotly. Побудова динамічних графіків для візуалізації стохастичних процесів. Аналіз і інтерпретація результатів.

### **Лекція 7: Моделювання інформаційної війни**

Аналіз моделей інформаційної війни. Реалізація алгоритму Лотка-Вольтера для декількох типів інформаційних загроз. Візуалізація динаміки конкуренції інформаційних потоків.

### **Лекція 8: Розробка системи підтримки прийняття рішень**

Архітектура SPPR для моделювання інформаційних конфліктів. Взаємодія фронтенду та бекенду (ReactJS + Python). API для розрахунків та динамічного налаштування параметрів.

### **Лекція 9: Використання сучасних бібліотек для моделювання**

Огляд Scipy та SimPy для системного моделювання. TensorFlow та PyTorch для моделювання складних динамічних систем. Порівняння готових рішень з самостійно написаними алгоритмами.

### **Лекція 10: Практичні аспекти та реальні приклади**

Статистичний аналіз даних та моделювання на основі реальних зразків. Тестування моделі інформаційної війни для конкретних сценаріїв. Оптимізація моделей для інтерактивної роботи.

## **ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ:**

### **Лабораторія 1: Вступ до Python для математичного моделювання**

Налаштування середовища Python (Jupyter Notebook, PyCharm). Робота з бібліотеками NumPy та Matplotlib. Написання базових програм для розв'язання алгебраїчних задач.

### **Лабораторна робота 2: Розв'язання диференціальних рівнянь**

Реалізація моделей Лотки-Вольтерри в Python. Використання бібліотеки SciPy для розв'язання звичайних диференціальних рівнянь. Візуалізація отриманих траєкторій.

### **Лабораторна робота 3: Моделювання випадкових процесів**

Генерація випадкових чисел і процесів в Python. Реалізація простого марковського процесу. Моделювання стохастичних диференціальних рівнянь за допомогою бібліотеки NumPy.

### **Лабораторна робота 4: Побудова та тестування моделей інформаційної війни**

Реалізація алгоритму конкуренції для двох інформаційних потоків. Тестування моделі для різних параметрів поширення інформації. Аналіз впливу «зовнішніх» та «внутрішніх» каналів.

### **Лабораторна робота 5: Оптимізація обчислень**

Використання профілювання (модуль cProfile) для аналізу продуктивності програми. Векторизація обчислень за допомогою NumPy. Оптимізація циклів та операцій з великими матрицями.

### **Лабораторна робота 6: Використання схем апроксимації**

Реалізація схем усереднення для динамічних систем. Апроксимація дифузії та її програмна реалізація. Аналіз точності апроксимації залежно від параметрів.

### **Лабораторна робота 7: Побудова програмного інтерфейсу для моделювання**

Використання Flask для створення бекенду. Налаштування API для обчислень. Розробка веб-додатку для візуалізації результатів моделювання.

### **Лабораторна робота 8: Робота з великими наборами даних**

Обробка та аналіз великих наборів даних (популярність у соціальних мережах, поведінка користувачів). Використання бібліотек Pandas для роботи з табличними даними. Інтеграція статистичних даних у модель інформаційної війни.

### **Лабораторна робота 9: Тестування та валідація моделі**

Методи тестування моделі: використання контрольних наборів даних. Порівняння моделей з теоретичними результатами. Налаштування параметрів для підвищення точності прогнозування.

### **Лабораторна робота 10: Створення інтерактивного звіту**

Використання Jupyter Notebook для інтеграції коду, графіків та тексту. Створення інтерактивних графіків за допомогою Plotly. Автоматичне генерування звітів з результатами моделювання.

## **ЗМІСТ**

ВСТУП	7
<b>ЧАСТИНА I. ЛЕКЦІЙНИЙ МАТЕРІАЛ</b>	<b>8</b>
Лекція 1. Вступ до математичного моделювання	8
Лекція 2. Основи моделювання динамічних систем	11
Лекція 3. Стохастичні процеси та випадкові збурення	19
Лекція 4. Стохастичні процеси та випадкові збурення (продовження)	36
Лекція 5. Моделювання інформаційної війни	46
Лекція 6. Асимптотичний аналіз стохастичних систем	70
Лекція 7. Оптимізація обчислень у Python	82
Лекція 8. Практичні аспекти та реальні приклади	84
<b>ЧАСТИНА II. ЛАБОРАТОРНІ РОБОТИ</b>	<b>85</b>
Лабораторна робота 1-8	85
ВИСНОВКИ	113
РЕКОМЕНДОВАНА ЛІТЕРАТУРА	114

## ВСТУП

Математичне моделювання є одним з найпотужніших інструментів сучасної науки, що дозволяє досліджувати складні системи та процеси без проведення натурних експериментів. В епоху цифрової трансформації особливої актуальності набуває моделювання інформаційних процесів – від поширення новин у соціальних мережах до протидії інформаційним загрозам.

Даний навчальний курс присвячено математичному моделюванню інформаційних процесів з використанням мови програмування Python. Курс базується на результатах дисертаційного дослідження «Математичне та комп'ютерне моделювання багатовимірних процесів інформаційної боротьби з використанням схем апроксимації та випадкових збурень» та охоплює як класичні підходи до моделювання, так і сучасні методи стохастичного аналізу.

Мета курсу – сформувати у студентів теоретичні знання та практичні навички побудови, дослідження та програмної реалізації математичних моделей інформаційних процесів.

### **Завдання курсу:**

- вивчити основні підходи до математичного моделювання динамічних систем;
- освоїти методи аналітичного та чисельного розв'язання диференціальних рівнянь;
- опанувати теорію стохастичних процесів та методи їх моделювання;
- навчитися застосовувати схеми апроксимації для аналізу складних систем;
- набути практичних навичок програмування на Python з використанням NumPy, SciPy, Matplotlib.

# ЛЕКЦІЯ 1

## Вступ до математичного моделювання

### 1.1. Поняття математичної моделі

Математична модель – це опис системи або процесу мовою математики: рівняннями, нерівностями, функціями, графами тощо. Модель є спрощеним відображенням реальності, що зберігає суттєві властивості досліджуваного об'єкта та ігнорує несуттєві деталі.

За визначенням А. А. Самарського, математичне моделювання – це заміна вивчення оригіналу вивченням його моделі та перенесення результатів такого вивчення на оригінал. Ключовою є вимога адекватності моделі – відповідності її поведінки поведінці реальної системи в рамках прийнятих припущень.

Процес моделювання включає етапи: постановка задачі та визначення мети моделювання; аналіз системи та виділення суттєвих факторів; формалізація – запис математичних співвідношень; вибір методу розв'язання (аналітичний чи чисельний); програмна реалізація; верифікація та валідація моделі; аналіз результатів та інтерпретація.

### 1.2. Класифікація математичних моделей

Математичні моделі класифікують за кількома критеріями. За характером процесів: детерміновані (результат однозначно визначається початковими умовами) та стохастичні (враховують випадкові фактори). За зміною в часі: статичні (описують стан системи) та динамічні (описують еволюцію системи). За неперервністю: з неперервним часом (диференціальні рівняння) та з дискретним часом (різницеві рівняння). За лінійністю: лінійні та нелінійні.

Для моделювання інформаційних процесів найчастіше використовують нелінійні динамічні моделі, оскільки поширення інформації має нелінійний характер (ефект насичення, мережеві ефекти) та суттєво залежить від часу.

### 1.3. Роль моделювання в дослідженні інформаційних процесів

Інформаційні процеси – це процеси створення, збору, обробки, накопичення, зберігання, пошуку, поширення та споживання інформації. В сучасному світі особливого значення набувають процеси поширення інформації в соціальних системах, зокрема в контексті інформаційної безпеки.

Математичне моделювання дозволяє: прогнозувати динаміку поширення інформації; оцінювати ефективність інформаційних кампаній; виявляти критичні параметри системи; розробляти стратегії протидії дезінформації; оптимізувати розподіл ресурсів для інформаційного впливу.

#### 1.4. Огляд інструментів Python для моделювання

Python став стандартом де-факто для наукових обчислень завдяки простоті синтаксису та потужній екосистемі бібліотек. NumPy забезпечує ефективну роботу з багатовимірними масивами та векторизовані операції. SciPy надає алгоритми для чисельного інтегрування, оптимізації, обробки сигналів, статистики. Matplotlib дозволяє створювати візуалізації публікаційної якості. Pandas призначена для аналізу та маніпуляції даними.

Для розв'язання диференціальних рівнянь використовується модуль scipy.integrate, зокрема функція solve\_ivp, що реалізує сучасні адаптивні методи. Для стохастичних моделей застосовуються генератори випадкових чисел numpy.random та спеціалізовані бібліотеки.

#### 1.5. Приклад: експоненціальна модель росту

Найпростіша модель росту – експоненціальна:  $\frac{dN(t)}{dt} = \alpha N(t)$ , де  $N(t)$  – кількість (населення, інформованих осіб тощо),  $\alpha$  – коефіцієнт росту. Ця модель передбачає необмежений ріст, що нереалістично для більшості реальних систем.

Логістична модель враховує обмеженість ресурсів:  $\frac{dN}{dt} = \alpha N(1 - N/K)$ , де  $K$  – ємність середовища. Розв'язок – S-подібна крива з насиченням. Ця модель широко застосовується для опису поширення інновацій, інформації, епідемій.

#### Контрольні питання:

1. Що таке математична модель? Які основні етапи моделювання?
2. Як класифікують математичні моделі?
3. Які переваги Python для наукових обчислень?
4. Порівняйте експоненціальну та логістичну моделі росту.

## ЛЕКЦІЯ 2

### Основи моделювання динамічних систем

#### 2.1. Звичайні диференціальні рівняння

Рівняння першого порядку можна записати у вигляді:

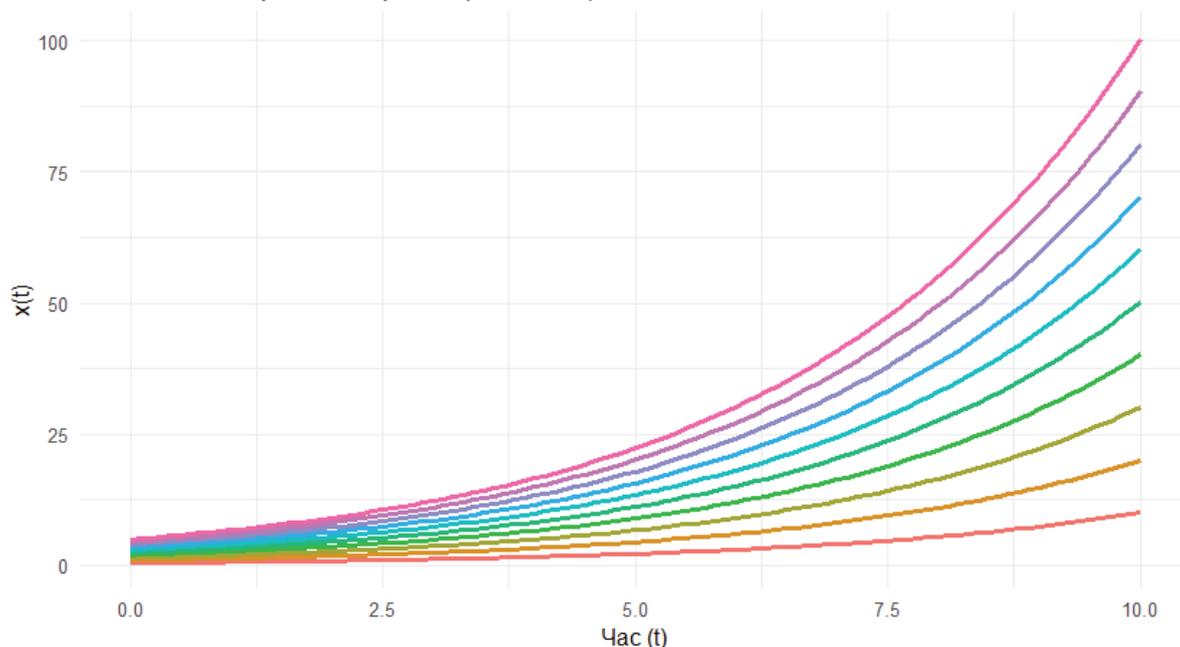
$$\frac{dx}{dt} = f(x, t).$$

Розв'язком диференціального рівняння називається диференційовна функція  $x = \varphi(t)$ , яка задовольняє рівняння, тобто перетворює його в тотожність.

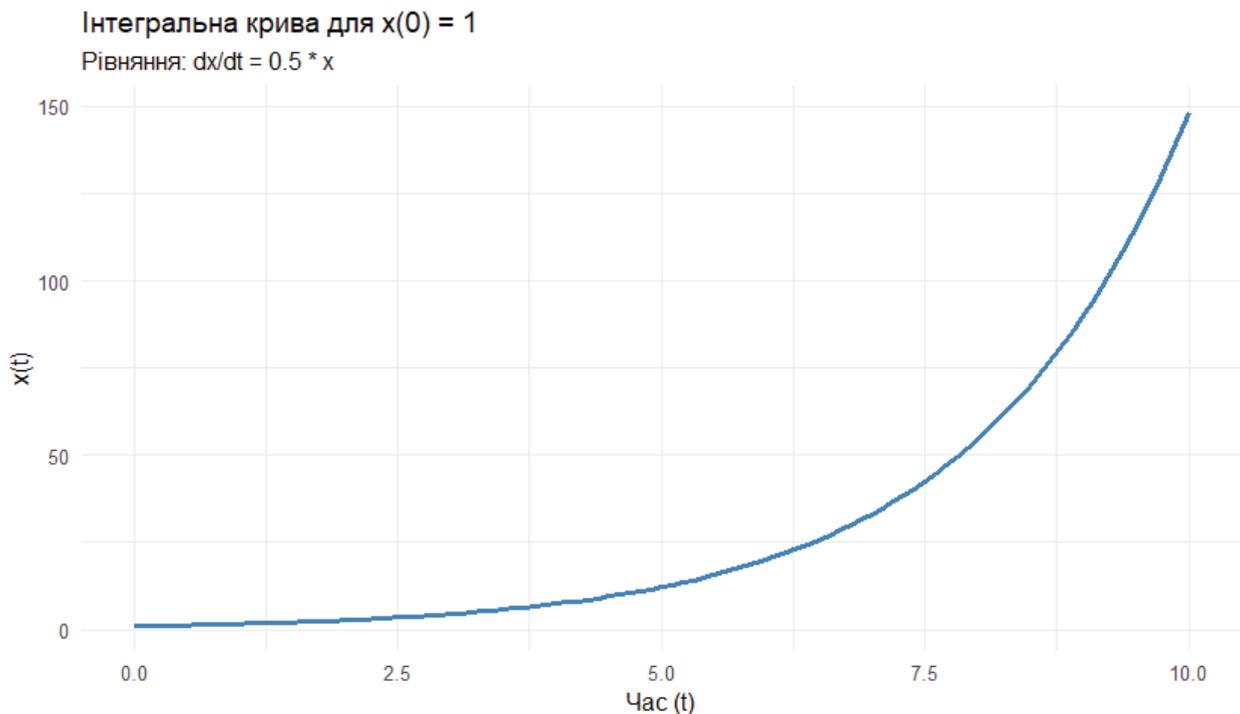
Якщо існує така функція, то розв'язком буде і інша функція,  $x = \varphi(t) + C$ , оскільки  $(\varphi(t) + C)' = x' = \frac{dx}{dt}$ .

Тобто маємо безліч розв'язків, які відрізняються один від одного на сталу. Графік розв'язку називається інтегральною кривою.

Множина інтегральних кривих ( $dx/dt = ax$ )



Якщо зафіксуємо початкову умову, то матимемо одну конкретну інтегральну криву



Важливими поняттями в теорії диференціальних рівнянь є існування, єдиність та стійкість їх розв'язку.

Нехай задане диференціальне рівняння першого порядку

$$\frac{dx}{dt} = f(x, t). \quad (2.1)$$

**Теорема.** Якщо функція  $f(x, t)$  та її похідна  $f'_x(x, t)$  неперервні в деякій області  $D \subset \mathbb{R}^2$ , то в деякому околі довільної внутрішньої точки  $(x_0, t_0)$  цієї області існує єдиний розв'язок  $x = \varphi(t)$  рівняння, який задовольняє умову

$$x(t_0) = x_0.$$

стан системи вважається стійким, якщо при деяких початкових збуреннях система залишається в певному околі стану рівноваги.

**Означення 1.** Розв'язок  $x = \varphi(t)$  системи диференціальних рівнянь (2.1), визначений для всіх  $t > t_0$ , називається стійким за Ляпуновим, якщо

$$\forall \varepsilon > 0 \exists \delta = \delta(\varepsilon) > 0,$$

що для всіх розв'язків  $x = x(t)$ , які задовольняють умову

$$|x(t_0) - \varphi(t_0)| < \delta,$$

виконується нерівність

$$|x(t) - \varphi(t)| < \varepsilon$$

для всіх  $t \geq t_0$ .

**Означення 2.** Розв'язок  $x = \varphi(t)$  системи диференціальних рівнянь (2.1), визначений для всіх  $t > t_0$ , називається асимптотично стійким за Ляпуновим, якщо він є стійким і виконується умова

$$\lim_{t \rightarrow \infty} |x(t) - \varphi(t)| = 0$$

для всіх  $t \geq t_0$ .

Деякі диференціальні рівняння досить легко розв'язуються (інтегруються), наприклад рівняння експоненційного зростання (спадання)

$$\frac{dN(t)}{dt} = \alpha N(t), \quad (2.2)$$

яке разом із початковою умовою

$$N(t_0) = N_0$$

визначає швидкість зміни стану системи в кожен момент часу.

Перенесемо всі члени з  $N$  в один бік, а з  $t$  – в інший (припускаючи, що  $N(t) \neq 0$ ):

$$\frac{dN}{N} = \alpha dt$$

Проінтегруємо обидві частини рівняння:

$$\int \frac{dN}{N} = \int \alpha dt$$

Отримуємо:

$$\ln|N| = \alpha t + C$$

де  $C$  – константа інтегрування.

За властивостями логарифма:

$$|N| = e^{\alpha t + C} = e^C \cdot e^{\alpha t}$$

Позначимо  $e^C$  як нову константу  $C_1$ :

$$N(t) = C_1 e^{\alpha t}$$

Підставимо  $t = t_0$  та  $N(t_0) = N_0$ , щоб знайти значення  $C_1$ :

$$N_0 = C_1 e^{\alpha t_0} \Rightarrow C_1 = N_0 e^{-\alpha t_0}$$

Підставляємо значення  $C_1$  у загальну формулу:

$$N(t) = (N_0 e^{-\alpha t_0}) e^{\alpha t} = k e^{\alpha(t)}$$

Функція  $N(t) > 0$  може описувати людей, охоплених певним інформаційним повідомленням (нарративом).

Рівняння (2.2) означає, що швидкість появи нових «носіїв» інформації прямо пропорційна кількості тих, хто вже її знає. Це класичне «сарафанне радіо» або алгоритмічне просування в соцмережах.

Якщо  $\alpha > 0$ , то це рівняння називають рівнянням зростання, у протилежному випадку – рівнянням розпаду. Додатне  $\alpha$  – наприклад, ситуація активного поширення нарративу означає, що інформація знаходить відгук, і кількість людей, які її сприйняли та поширюють, постійно збільшується. Коли  $\alpha < 0$ , ми маємо справу із затуханням інтересу до теми або результатом успішної контрпропаганди. Інформаційна хвиля не отримує «підживлення» і поступово зникає з інфополя.

У розв'язку присутня довільна константа  $N_0$ , для визначення якої необхідно задати, наприклад, початкове охоплення аудиторії в момент запуску інформаційної операції або публікації новини  $N_0 = N(0) > 0$  у момент часу  $t_0 = 0$  – фіксація точки старту «інформаційного вкиду». Математично ми стверджуємо, що для початку хвилі вже має бути певна ненульова кількість людей, які володіють цією інформацією в момент часу «нуль».

Інтерпретація стійкості – соціальна спільнота перебуває у певному стані стабільності. Вкид інформації – зміна початкового стану на  $x(t_0)$ .

Інтерпретація: Якщо система стійка за Ляпуновим, то навіть потужна інформаційна атака не викличе хаосу чи обвалу системи. Рівень дестабілізації залишатиметься в межах допустимого  $\varepsilon$ . Суспільство "хитнеться", але не розвалиться.

**2. Асимптотична стійкість** – з часом система повністю повертається до норми.

Це ідеальний сценарій, коли критичне мислення громадян, робота медіаграмотності та розвінчання фейків зводять ефект атаки до нуля.

**Інтерпретація:** Навіть якщо в момент  $t_0$  суспільство повірило у фейк, з часом (завдяки внутрішнім механізмам корекції) вплив цієї брехні повністю нівелюється, і стан громадської думки повертається до об'єктивної реальності.

Експоненціальна функція зростає дуже швидко. Якби не було обмежень, одна публікація за короткий час зробила б усю спільноту носієм певної ідеї. Але на практиці ресурс уваги обмежений, а люди з часом «забувають» або втрачають інтерес до теми.

Відносна зміна чисельності популяції

$$\frac{dN}{N} = \alpha dt$$

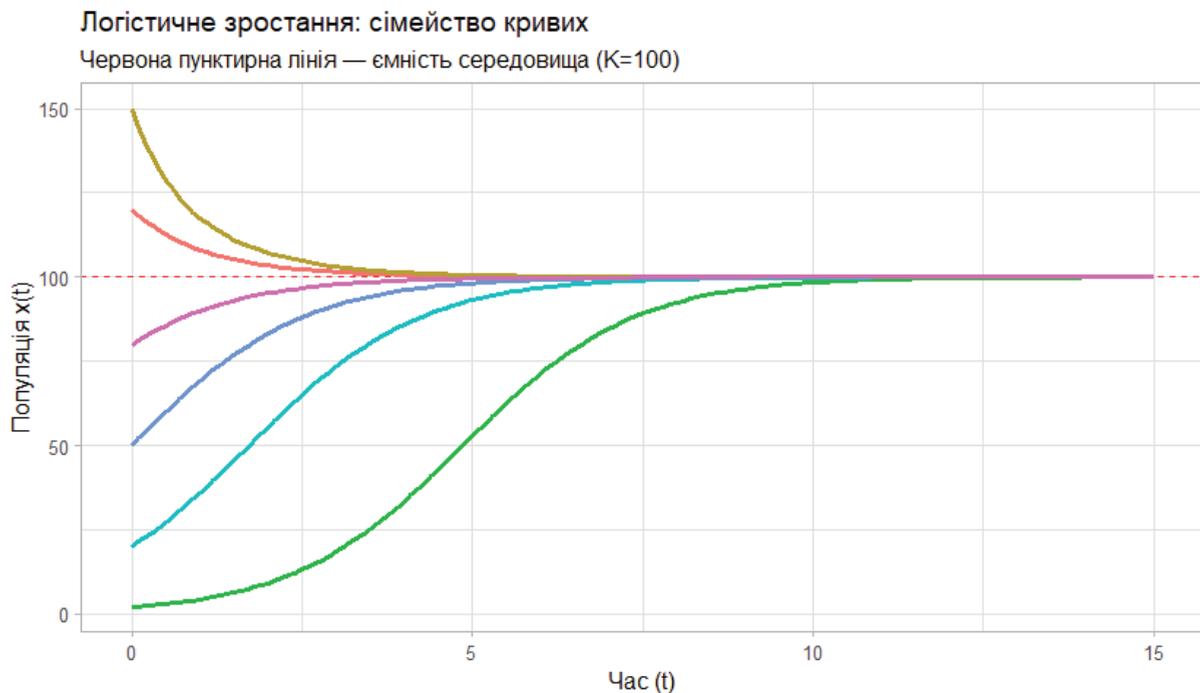
Ферхюльст припустив, що існує певна «стеля» – ємність середовища ( $K$ ). У міру того, як популяція  $N$  наближається до  $K$ , швидкість росту має падати. Він замінив константу  $\alpha$  на функцію, яка зменшується:

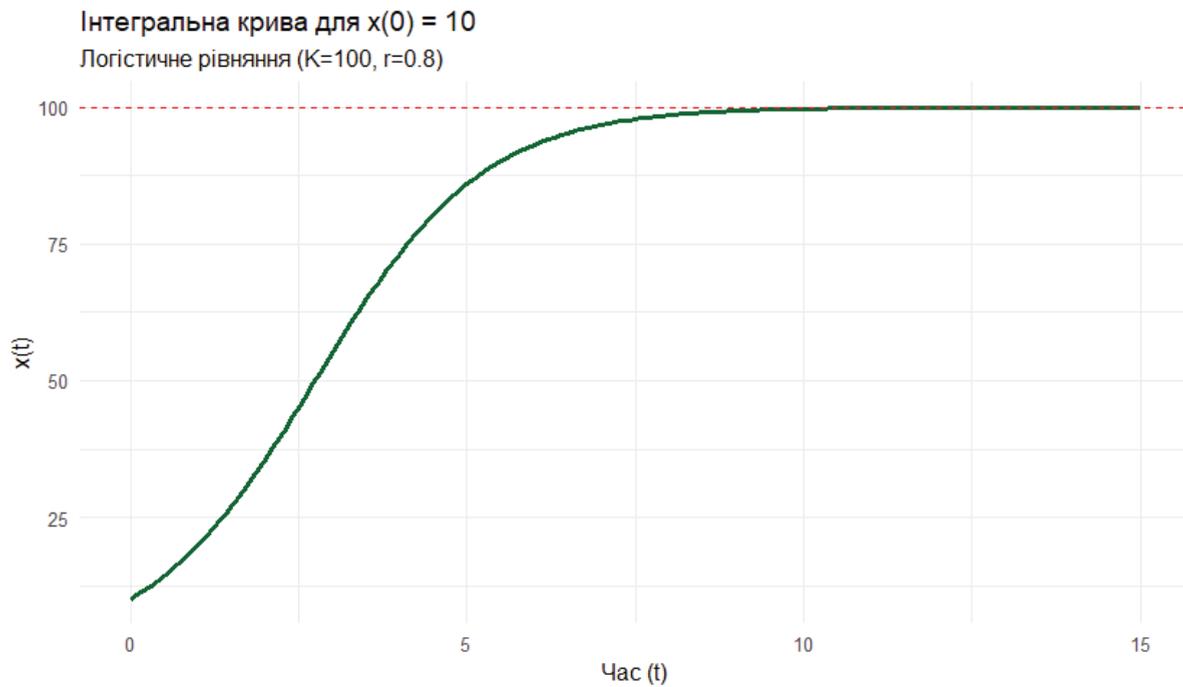
$$\alpha \rightarrow \alpha \left(1 - \frac{N}{K}\right)$$

Якщо підставити цю нову «гальмівну» функцію в початкове рівняння, ми отримаємо:

$$\frac{dx}{dt} = rx \left(1 - \frac{x}{K}\right)$$

Другий доданок має сенс зменшення відносного приросту в результаті знищення природних ресурсів. Це відбувається тим інтенсивніше, чим більша чисельність популяції.





У результаті більш реалістичне рівняння приводить до логістичної функції,

### Системи рівнянь

Розглянемо модель

$$\frac{dx(t)}{x} = A dt, \quad x(t_0) = x_0$$

Нехай тепер

$$\mathbf{x}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$$

– вектор станів, а

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

– матриця коефіцієнтів.

Рівняння зі слайда (якщо переписати його у стандартній формі  $\frac{dx}{dt} = Ax$ ) набуває вигляду:

$$\frac{d}{dt} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}.$$



Флуктуації чисельності популяції за рахунок випадкових внутрішніх і зовнішніх факторів, що не враховуються простою моделлю (2.2), насправді дуже великі.

Таким чином, наш світ не є детермінованим. Його справжнє обличчя – імовірнісне:

*Звичайні диференціальні рівняння – це лише перше наближення до реальності. Більш адекватним інструментом дослідження є стохастичні рівняння.*

Більшість економічних, біологічних та фізичних систем може бути описана системою диференціальних рівнянь:

$$dx = a(x, t) dt$$

де  $x(t) = \{x_1(t), \dots, x_n(t)\}$  – вектор змінних стану системи, а  $a(x, t)$  визначає її динаміку.

Будь-які рівняння з похідними вищих порядків можна звести до системи такого вигляду, ввівши нові змінні.

Таке представлення дозволяє легко перейти до чисельного інтегрування. Замінюючи нескінченно малі прирости на малі скінченні:

$$\Delta x = x_{k+1} - x_k, \Delta t = t_{k+1} - t_k,$$

отримуємо ітераційну схему:

$$x_{k+1} = x_k + a(x_k, t_k)\Delta t$$

Чим менший крок  $\Delta t$ , тим ближче чисельний розв'язок до істинного.

## 2.2. Використання SciPy для розв'язання ЗДР

Функція `scipy.integrate.solve_ivp` є сучасним інтерфейсом для чисельного розв'язання задач Коші. Вона підтримує різні методи: RK45 (за замовчуванням), RK23, DOP853, Radau, BDF, LSODA. Параметр `dense_output=True` дозволяє отримати інтерполяційну функцію для обчислення розв'язку в довільних точках.

Приклад виклику: `sol = solve_ivp(fun, t_span, y0, method='RK45', dense_output=True)`, де `fun(t, y)` – права частина рівняння, `t_span = (t0, tf)` – інтервал, `y0` – початкова умова.

### Контрольні питання:

1. Опишіть логістичне рівняння та його фізичний зміст.
2. Як визначити стійкість положення рівноваги?

## ЛЕКЦІЯ 3

### Стохастичні процеси та випадкові збурення

#### 3.1. Обмеження детерміністичних моделей

Детерміністичні моделі передбачають, що майбутній стан системи повністю визначається її поточним станом та керуючими впливами. Однак реальні системи зазнають впливу численних випадкових факторів: флуктуації середовища, невизначеність параметрів, похибки вимірювань, непередбачувана поведінка агентів.

В інформаційних процесах випадковість є невід'ємною складовою: час публікації новини, реакція аудиторії, вірусне поширення – все це має стохастичну природу. Стохастичні моделі дозволяють враховувати цю невизначеність та оцінювати ймовірності різних результатів.

#### 3.2. Випадкові процеси: основні поняття

**Теорія випадкових (стохастичних) процесів** - це математична наука, яка вивчає закономірності випадкових явищ у динаміці їх розвитку.

Вивчаючи явища навколишнього світу, ми часто бачимо процеси, перебіг яких неможливо передбачити заздалегідь. Ця невизначеність (непередбачуваність) виникає внаслідок впливу випадкових факторів, які впливають на хід процесу.

Поняття випадкового процесу є узагальненням поняття випадкової величини.

Припустимо, нам дано ймовірнісний простір  $(\Omega, F, P)$  і деяку непорожню множину  $T \subseteq \mathbb{R}$ .

Відзначимо, що ймовірнісний простір - це трійка, що складається з

$\Omega$  – простір елементарних подій;

$F$  –  $\sigma$ -алгебра підмножин  $\Omega$  називається простором подій,

$P$  – ймовірність, для якої виконуються певні аксіоми

$$P(\Omega) = 1, \quad 0 \leq P(A) \leq 1$$

Набір випадкових величин  $\xi := \{\xi(t), t \in T\}$ , визначених у ймовірнісному просторі  $(\Omega, F, P)$ , називається **випадковим (стохастичним) процесом**

Фіксуючи  $\omega \in \Omega$  маємо функцію  $\xi_\omega(t) := \xi(t)$ , яка називається **траєкторією (реалізацією)** випадкового процесу.

Фіксуючи  $t \in T$  маємо випадкову величину  $\xi(\omega)$ , яка називається перерізом випадкового процесу  $\xi$  в моменті часу  $t$ .

Множину  $T$  називають часовою областю, а множину значень випадкових величин  $\xi(t)$  – фазовим простором випадкового процесу.

Залежно від форми множини  $T$  випадкові процеси поділяють на процеси з дискретним часом і процеси з неперервним часом.

Ми говоримо, що  $\xi \in$  **випадковим процесом з дискретним часом**, якщо множина  $T$  зліченна. Зокрема, якщо  $T = \{0,1,2, \dots\}$ , або  $T = \mathbb{Z}$ , то  $\xi \in$  випадковою послідовністю.

Якщо множина  $T \in$  підмножиною дійсних чисел, ми будемо називати  $\xi$  **випадковим процесом з неперервним часом**.

Таким же чином випадкові процеси можна розділити на два класи (дискретні та неперервні) залежно від типу фазового простору  $E$ .

Таким чином, випадковий процес можна записати як функцію двох аргументів – часу  $t$  і елементарної події  $\omega$ :

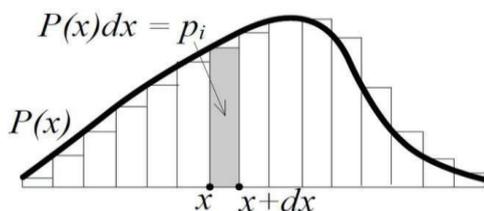
$$f(t) := f(t, \omega), t \in T, \omega \in \Omega, f(t) \in E. \quad (3.1)$$

З теорії ймовірностей ми знаємо, що повною вичерпною характеристикою випадкової величини є її закон розподілу.

Для дискретної випадкової величини це ряд розподілу

$X$	$x_1$	$x_2$	...	$x_n$
$P$	$p_1$	$p_2$	...	$p_n$

Для неперервної випадкової величини це щільність розподілу. Щільність розподілу ймовірностей  $p(x)$  – це функція, яка при множенні на інтервал  $dx$  дає ймовірність  $p_i$ , що значення  $x$  буде в діапазоні від  $x$  до  $x + dx$ .



$$\sum_i p_i = \int_{-\infty}^{\infty} P(x) dx = 1.$$

Універсальною характеристикою як дискретних, так і неперервних випадкових величин є їх функція розподілу

$$F(x) = P\{X < x\},$$

тобто ймовірність того, що випадкова змінна прийме значення, менше заданого  $x$ .

Розглянемо тепер випадковий процес  $X(t)$ . Як ми вже знаємо, переріз випадкового процесу у фіксований момент часу  $t \in$  випадковою величиною з функцією розподілу

$$F(t, x) = P\{X(t) < x\}. \quad (3.2)$$

Функцію (2) називають **одновимірним законом розподілу** випадкового процесу  $X(t)$ .

Загалом закономірності випадкового процесу  $X(t)$  визначаються загальним розподілом ймовірностей його значень  $X(t_1), X(t_2), \dots, X(t_n)$  у різні моменти часу  $t_1, t_2, \dots, t_n$  (скінченновимірні розподіли цього випадкового процесу)

Чим більша розмірність закону розподілу, тим складніше досліджувати випадковий процес, тому на практиці часто використовується не форма закону розподілу, а його основні характеристики, які описують випадковий процес не повно, а частково.

Пам'ятаємо, що багато задач теорії ймовірностей можна розв'язувати, не використовуючи закони розподілу випадкових величин, а лише використовуючи їх числові характеристики – середнє значення, дисперсію, коваріацію, початковий і центральний моменти тощо.

Те ж саме можна зробити і зі випадковими процесами, тільки для них основними характеристиками будуть вже не числа, а не випадкові функції аргументу  $t$ .

Першою і найважливішою характеристикою випадкового процесу є його математичне сподівання.

**Математичним сподіванням (середнім значенням)** випадкового процесу  $X(t)$  називається не випадкова функція  $m_x(t)$ , яка для будь-якого значення аргументу  $t$  дорівнює середньому значенню відповідного перерізу випадкового процесу

$$m_x(t) = \mathbf{E}[X(t)] \quad (3.3)$$

Якщо переріз випадкового процесу є дискретною випадковою величиною, то

$$m_x(t) = \sum_i x_i(t) p_i(t),$$

а у випадку неперервної випадкової величини

$$m_x(t) = \int_{-\infty}^{+\infty} xp(t, x) dx$$

**Центрований стохастичний процес  $\tilde{X}(t)$**  – це процес, який буде отримано, якщо від  $X(t)$  відняти його математичне сподівання

$$\tilde{X}(t) = X(t) - m_x(t) \quad (3.4)$$

Звичайно, математичне сподівання (середнє значення) центрованого випадкового процесу тотожно дорівнює нулю.

Крім середнього значення в теорії випадкових процесів вивчаються й інші характеристики, подібні до числових характеристик випадкових величин, але це будуть вже не числа, а функції.

**Початковим моментом  $k$ -го порядку** випадкового процесу  $X(t)$  називається середнє значення  $k$ -го степеня відповідного перерізу випадкового процесу

$$\alpha_k(t) = \mathbf{E}[(X(t))^k], \quad (3.5)$$

**Центральним моментом  $k$ -го порядку** випадкового процесу  $X(t)$  називається середнє значення  $k$ -го степеня центрованого випадкового процесу

$$\mu_k(t) = M[(\tilde{X}(t))^k] = M[(X(t) - m_x(t))^k]. \quad (3.6)$$

**Дисперсія** випадкового процесу є його другим центральним моментом

$$D_x(t) = M[(\tilde{X}(t))^2] = M[(X(t) - m_x(t))^2] = M[X^2(t)] - m_x^2(t). \quad (3.7)$$

Якщо переріз випадкового процесу є дискретною випадковою величиною, то

$$D_x(t) = \sum_i (x_i - m_x(t))^2 p_i(t),$$

і якщо переріз випадкового процесу є неперервною випадковою величиною, то

$$D_x(t) = \int_{-\infty}^{+\infty} (x - m_x(t))^2 p(t, x) dx \quad (3.8)$$

або

$$D_x(t) = \int_{-\infty}^{+\infty} x^2 p(t, x) dx - m_x^2(t)$$

Таким чином, і середнє, і дисперсія випадкового процесу  $X(t)$  визначаються його одновимірним законом розподілу.

Середнє значення випадкового процесу – це не випадкова «середня функція», навколо якої коливаються реалізації випадкового процесу, а дисперсія випадкового процесу – не випадкова додатна функція, що характеризує ступінь розсіювання реалізацій випадкового процесу. Процес.

**Середнє квадратичне відхилення** – це арифметичне значення квадратного кореня з дисперсії

$$\sigma_x(t) = \sigma[X(t)] = \sqrt{D_x(t)} \quad (3.9)$$

Розглянуті характеристики дуже важливі, але, на жаль, вони не є вичерпними, оскільки вони визначаються одновимірними розподілами.

Як видно з рисунку, при приблизно однакових середніх значеннях і дисперсіях випадкові процеси  $X_1(t)$  і  $X_2(t)$  суттєво відрізняються за внутрішньою структурою.

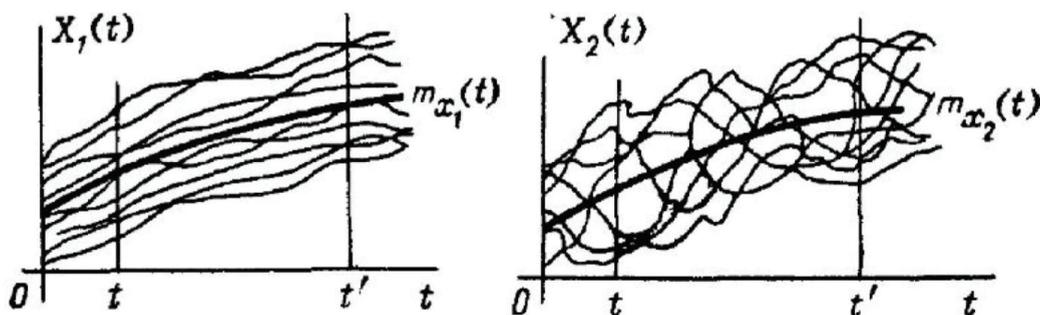


Рис. 3.

Це означає, що ми думаємо про дослідження інших характеристик, які могли б охарактеризувати структуру стохастичних процесів.

З теорії ймовірностей відомо, що ступінь лінійної залежності випадкових величин виражається їх коваріацією

$$K_{xy} = \mathbf{E}[(X - m_x)(Y - m_y)] = \mathbf{E}[XY] - m_x m_y$$

Аналогічна характеристика була введена для випадкових процесів.

Розглянемо дві випадкові величини, а саме дві ділянки випадкового процесу  $X$  для моментів  $t$  і  $t'$ . Ми знаходимо коваріацію для цих двох випадкових величин:

$$K_x(t, t') = M[\tilde{X}(t)\tilde{X}(t')] = M[X(t)X(t')] - m_x(t)m_x(t') \quad (3.10)$$

Функцію (3.10) називають **кореляційною функцією** випадкового процесу  $X(t)$ .

Розглянемо тепер властивості кореляційної функції

1. Для рівних аргументів ( $t = t'$ ) кореляційна функція дорівнює дисперсії випадкового процесу.

Дійсно,

$$K_x(t, t) = \mathbf{E}[\tilde{X}(t)\tilde{X}(t)] = \mathbf{E}[\tilde{X}^2(t)] = D_x(t).$$

Тому немає потреби використовувати дисперсію як окрему незалежну характеристику, а для основних характеристик випадкового процесу достатньо розглянути середню функцію та кореляційну функцію.

### 3.2. Процес Пуассона

#### Процеси з незалежними приростами.

**Означення 1.** Випадковий процес  $\{X(t), t \geq 0\}$  має незалежні прирости, якщо для довільного  $n \in \mathbb{N}$ , для довільних  $t_0 < t_1 < \dots < t_n$ , випадкові величини (прирости)  $X(t_1) - X(t_0), X(t_2) - X(t_1), \dots, X(t_n) - X(t_{n-1})$  утворюють систему незалежних випадкових величин.

Зауважимо, що для випадкових процесів з дискретним часом процеси з незалежними приростами є процесами випадкового блукання, тому процеси з

незалежними приростами слід розглядати як узагальнення випадкових блукань у неперервному часі.

**Означення.** Випадковий процес  $\{v(t), t \geq 0\}$  називається процесом Пуассона з інтенсивністю  $\lambda > 0$ , якщо виконується:

1.  $v(0) = 0$  майже напевно;
2. Випадковий процес  $\{v(t), t \geq 0\}$  має незалежні прирости, тобто для довільних  $t_0 < t_1 < \dots < t_n$  прирости

$$v(t_1) - v(t_0), v(t_2) - v(t_1), \dots, v(t_n) - v(t_{n-1})$$

є незалежними випадковими величинами;

3.  $v(t) - v(s)$ ,  $0 \leq s < t$ , має пуассонівський розподіл з параметром  $\lambda(t - s)$ , тобто

$$P(v(t) - v(s) = k) = \frac{\lambda^k (t - s)^k e^{-\lambda(t-s)}}{k!}, k = 0, 1, 2, \dots$$

4. Траєкторії процесу є неперервними справа майже напевно.

Простіше кажучи, процес Пуассона – це процес з неперервним часом, який описує виникнення рідкісних випадкових подій.

Властивості процесу Пуассона, що впливають з означення

Для будь-якого  $t > 0$ :

$$P(v(t) = k) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}, k = 0, 1, 2, \dots$$

оскільки  $v(t) = v(t) - v(0)$ , а  $v(t) \sim \text{Po}(\lambda t)$ .

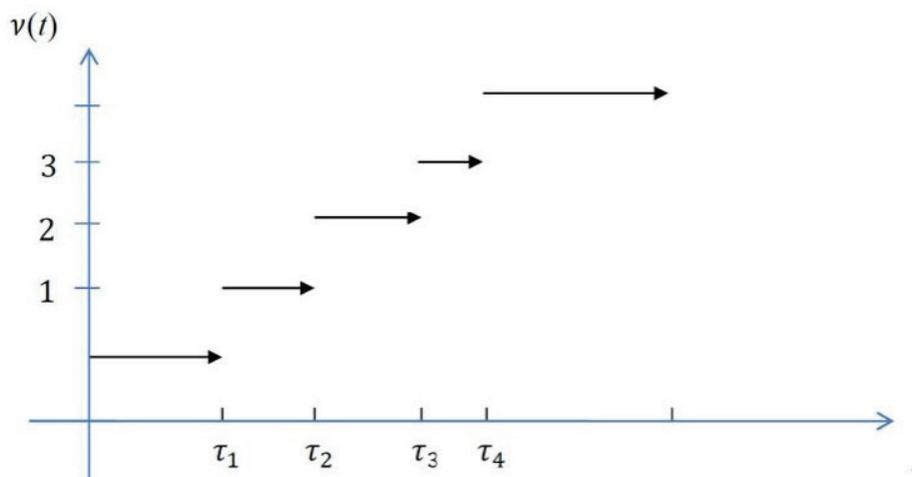
Звідси легко бачити, що:

$$\mathbb{E}[v(t)] = \lambda t, D[v(t)] = \lambda t;$$

$$\lambda = \lim_{\tau \rightarrow 0} \frac{1}{\tau} P(v(\tau) = 1).$$

Параметр  $\lambda$  називається інтенсивністю процесу Пуассона  $v(t)$ .

Пуассонівські випадкові величини є дискретними та невід’ємними, тому траєкторії процесу мають ступінчастий вигляд і є сталими неспадними функціями.



Rys.1.

$$K_v(t, s) = \lambda \min(t, s).$$

**Теорема 2.** (явна конструкція процесу Пуассона).

Нехай  $\{\xi_k, k \geq 1\}$  – послідовність незалежних випадкових величин, кожна з яких має показниковий (експоненційний) розподіл з параметром  $\lambda$

$$\xi_k \sim \text{Exp}(\lambda), f_{\xi_k}(x) = \lambda e^{-\lambda x}, x \geq 0.$$

Покладемо

$$S_0 = 0, S_n = \sum_{k=1}^n \xi_k, n \geq 1.$$

Тоді стохастичний процес  $\{v(t), t \geq 0\}$ , заданий у вигляді

$$v(t) = \sup \{n: S_n \leq t\},$$

є процесом Пуассона з інтенсивністю  $\lambda$ .

Величини  $S_n$  називаються моментами регенерації процесу Пуассона  $v(t)$ .

### 3.3. Процес Вінера

#### Означення 1.

Гаусівський процес або нормальний процес – це випадковий процес, для якого всі скінченновимірні розподіли є нормальними.

Це означає, що якщо задано гаусівський процес  $X(t)$ , то вектор, утворений з його перерізів

$$(X(t_1), X(t_2), \dots, X(t_n)),$$

має багатовимірний нормальний розподіл.

Працюючи з нормальними процесами, необхідно використовувати відомості з теорії ймовірностей про нормальні вектори. Нагадаємо відповідні означення.

**Означення 2.**

Вектор

$$\mathbf{X} = (X(t_1), X(t_2), \dots, X(t_n))$$

називається нормальним, якщо він має щільність розподілу:

$$f(\mathbf{X}) = \frac{1}{(2\pi)^{n/2} \sqrt{\det(\Sigma)}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})\right)$$

де

$$\boldsymbol{\mu} = \mathbb{E}X, \Sigma = \mathbb{E}[\tilde{X}\tilde{X}^T] = \mathbb{E}[(X - \boldsymbol{\mu})(X - \boldsymbol{\mu})^T]$$

– відповідно вектор математичних сподівань і коваріаційна матриця.

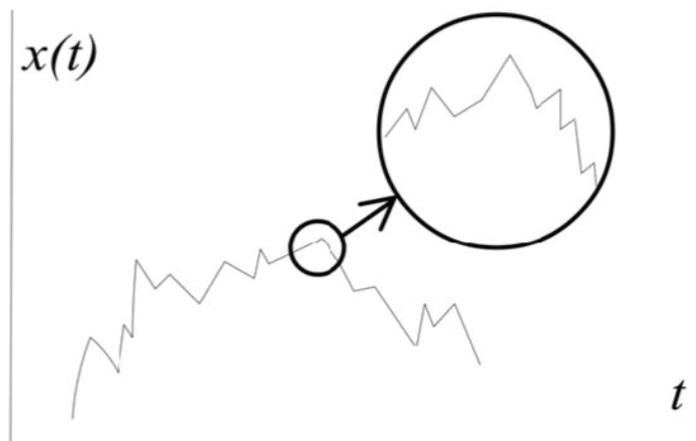
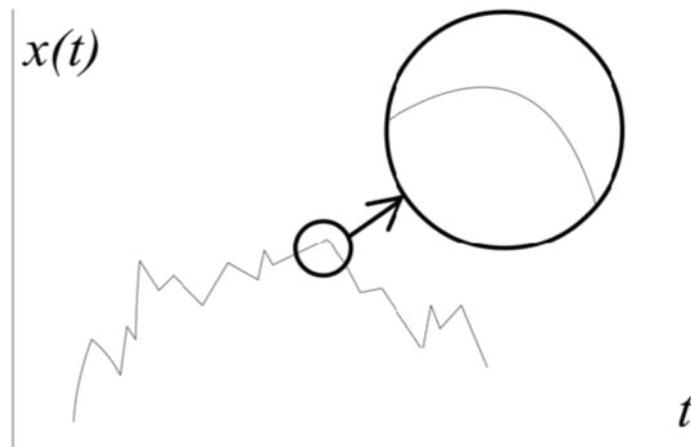
**Означення 3.** Процес  $\{W(t), t \geq 0\}$  називається вінерівським процесом, якщо виконуються наступні умови:

- 1) Випадковий процес  $\{W(t), t \geq 0\}$  має незалежні прирости, тобто для довільних  $t_0 < t_1 < \dots < t_n$ , прирости  $W(t_1) - W(t_0), W(t_2) - W(t_1), \dots, W(t_n) - W(t_{n-1})$  є незалежними;
- 2) Випадкова величина  $W(t) - W(s), s < t$ , є нормально розподілена з математичним сподіванням  $M[W(t) - W(s)] = 0$ , і дисперсією  $D[W(t) - W(s)] = t - s$ ;
- 3) Траєкторії  $\{W(t), t \geq 0\}$  є неперервними.

Ми визначили вінерівський процес аксіоматично, тобто описавши його властивості, і, звичайно, виникає запитання: чи існує математичний об'єкт, який задовольняє цим аксіомам?

Виявляється, процес Вінера служить математичною моделлю для руху молекули під час хаотичних ударів інших молекул, тому його також називають броунівським рухом. Процес Вінера має цікаві, незвичайні властивості.

Процес Вінера має властивість, незвичну для функцій, з якими ми зазвичай маємо справу. Приклад неперервних, але ніде не диференційованих функцій не дуже легко побудувати, і траєкторії вінерівського процесу майже всі такі функції



Траєкторія процесу Вінера значно відрізняється від звичайної функції математичного аналізу, якщо при збільшенні ми розглядаємо звичайну функцію із сильним зломом, ми бачимо, що вона гладка в малих масштабах, стохастична випадкова функція може залишатися ламаною за будь-якого збільшення.

### 3.4. Стаціонарність та ергодичність

#### Стаціонарні процеси

Розглянемо іншу класифікацію стохастичних процесів – стаціонарні та нестаціонарні.

Стаціонарні випадкові процеси є важливим класом стохастичних процесів. На практиці дуже часто трапляються процеси, що відбуваються в часі приблизно однорідно.

Простіше кажучи, це такі процеси, ймовірнісні властивості яких не залежать від початку відліку часу. Їхні розподіли перерізів сьогодні є певними, завтра будуть точно такими самими і післязавтра, тобто ці розподіли не залежать від часу.

Математичні сподівання є сталими, дисперсії – сталими. Багатовимірні розподіли визначаються лише різницею моментів часу перерізів. Такі процеси описують характеристики явищ, усталених у часі.

Траєкторії стаціонарних процесів можуть не мати регулярної форми, але стаціонарність – це не про «гарну форму» траєкторій, а про розподіли.

З іншого боку, можна виділити явно нестаціонарні випадкові процеси – це процеси, що розвиваються в часі. Прикладами є: зміни навантаження електроенергетичної мережі протягом доби, загасаючі коливання в системах автоматичного керування тощо.

Під час дослідження багатьох об'єктів можна отримати точний або наближений опис моделі за допомогою стаціонарних стохастичних процесів.

### **Означення 1.**

Стохастичний процес другого порядку  $\{X(t), t \in T\}$  називається стаціонарним у вузькому сенсі, якщо для будь-яких  $\{t_1, t_2, \dots, t_n\} \subset T$  та будь-якого  $h \in T$  такого, що  $\{t_1 + h, t_2 + h, \dots, t_n + h\} \subset T$ , розподіли випадкових векторів

$$\{X(t_1, \omega), \dots, X(t_n, \omega)\} \text{ та } \{X(t_1 + h, \omega), \dots, X(t_n + h, \omega)\}$$

(скінченновимірні розподіли процесу) є однаковими.

Інакше кажучи, процес є стаціонарним у вузькому сенсі, якщо всі його скінченновимірні розподіли не змінюються при однаковому зсуві часу.

Звідси, зокрема для  $n = 1$ , випливає, що розподіл  $X(t)$  збігається з розподілом  $X(t + h)$  для довільного  $h$ , тобто одновимірний розподіл не залежить від часу. Це означає, що жодна числова характеристика такого розподілу не залежить від часу. Наприклад, математичне сподівання та дисперсія процесу, стаціонарного у вузькому сенсі, є сталими:

$$m_X(t) = \text{const}, D_X(t) = \text{const}.$$

Що стосується двовимірного розподілу процесу, стаціонарного у вузькому сенсі, то він залежить лише від різниці  $t_2 - t_1$ . У результаті всі числові характеристики двовимірного розподілу (наприклад, кореляційна та коваріаційна функції) також залежать лише від різниці між  $t_1$  і  $t_2$ .

Отже, існує функція  $K(\tau)$  одного аргументу така, що кореляційна функція процесу  $X(t)$ , стаціонарного у вузькому сенсі, має вигляд

$$K_X(t_1, t_2) = K(t_2 - t_1).$$

### Означення 2.

Стохастичний процес  $\{X(t), t \in T\}$  називається стаціонарним у широкому сенсі, якщо його математичне сподівання не залежить від часу  $t$ , а кореляційна функція залежить лише від одного аргументу  $\tau = t - t'$ :

$$m_X(t) = m_X = \text{const},$$

$$K_X(t, t + \tau) = k_X(\tau).$$

### Ергодичні процеси

Розглянемо прикладну задачу. Нехай існує єдина, але достатньо довга реалізація випадкового процесу  $X(t)$ :  $t \in [0, T]$  – для неперервного часу, або  $t \in \{1, 2, \dots, N\}$  – для дискретного часу. Потрібно оцінити деякі характеристики цього процесу.

#### Означення 1.

Нехай процес  $\{X(t), t \geq 0\}$  другого порядку має стале математичне сподівання:

$$\mathbb{E}X(t) = m.$$

Нехай він є інтегровним у середньоквадратичному сенсі на будь-якому відрізку  $[0, T]$ .

Розглянемо часову середню:

$$\langle X(t) \rangle_T \equiv \frac{1}{T} \int_0^T X(t) dt.$$

Якщо

$$\langle X(t) \rangle_T \xrightarrow[T \rightarrow \infty]{\text{l.i.m.}} m,$$

то процес  $X(t)$  називається ергодичним у середньоквадратичному сенсі за математичним сподіванням.

Отже, для процесу, ергодичного за математичним сподіванням, добрим оцінюванням параметра  $m_X$  є часова середня єдиної спостережуваної реалізації  $\langle X(t) \rangle_T$ .

## 3.5. Марковські процеси

**Означення.** Марковський процес (або «процес без пам'яті») – це послідовність подій, у якій імовірність кожної події залежить лише від результату попередньої.

Іншими словами, випадковий процес  $X(t)$  називається процесом Маркова, якщо для кожного моменту часу  $t_i$  ймовірність будь-якого майбутнього стану системи залежить лише від стану системи в поточний момент часу  $t_i$  і не залежить від того, як система досягла цього стану.

Для послідовності подій

$$X = (X_n)_{n \in \mathbb{N} \cup \{0\}} = (X_0, X_1, X_2, \dots), X_n \in E, \forall n \in \mathbb{N}$$

властивість Маркова означає, що

$$P(X_{n+1} = s_{n+1} | X_n = s_n, X_{n-1} = s_{n-1}, X_{n-2} = s_{n-2}, \dots) = P(X_{n+1} = s_{n+1} | X_n = s_n)$$

Далі розглянемо систему  $S$ , яка в будь-який момент часу  $t = t_i$  може перебувати в одному з несумісних станів  $s_1, s_2, \dots, s_n$  і перехід цієї системи з одного стану  $s_i$  в інший  $s_j$  може відбуватися в моменти часу  $t_1, t_2, \dots, t_k, \dots$ .

Для такого процесу моменти  $t_1, t_2, \dots, t_k, \dots$  розглядаються як кроки черги процесу, а як аргумент, від якого залежить процес, розглянемо не час, а номер кроку:  $1, 2, \dots, k, \dots$

Цей процес характеризується послідовністю станів

$$S(0), S(1), \dots, S(k), \dots$$

де  $S(0)$  – початковий стан системи (до першого кроку);

$S(1)$  – стан системи після першого кроку;

...

$S(k)$  – стан системи після  $k$ -го кроку;

...

**Означення.** Ланцюг Маркова – це марковський процес з дискретним часом і дискретними станами.

Однією з основних задач дослідження ланцюга Маркова є знаходження ймовірності того, що система  $S$  опиниться в стані  $s_i$  на будь-якому кроці  $k$ , тобто

$$p_i(k) = P[S(k) = s_i], \quad i = 1, 2, \dots, n; \quad k = 0, 1, \dots$$

Щоб знайти ці ймовірності, нам потрібно знати початковий розподіл ймовірностей, який є ймовірностями системи в різних станах при  $t_0 = 0$ :

$$p_1(0), p_2(0), \dots, p_i(0), \dots, p_n(0),$$

та умовні ймовірності прямого переходу системи  $S$  зі стану  $s_i$  у стан  $s_j$  на  $k$ -му кроці:

$$p_{ij}(k) = P[S(k) = s_j | S(k-1) = s_i]$$

Ймовірності умовного переходу  $p_{ij}(k)$  називаються **ймовірностями переходу** ланцюга Маркова.

(Зауважимо, що умовна ймовірність настання події  $A$  при настанні події  $B$  є

$$P(A|B) = \frac{P(A \cdot B)}{P(B)}$$

Ймовірність  $p_{ii}(k)$  – це ймовірність того, що на  $k$ -му кроці система залишиться в стані  $s_i$ .

Запишемо ймовірності  $p_i(0)$  у вигляді вектора

$$p = (p_1(0), p_2(0), \dots, p_i(0), \dots, p_n(0))$$

де  $p_i(0) \geq 0, \sum_{i=1}^n p_i(0) = 1$ .

Перехідні ймовірності  $p_{ij}(k)$  можна записати у вигляді квадратної матриці

$$P = \begin{pmatrix} p_{11}(k) & \cdots & p_{1n}(k) \\ \vdots & \ddots & \vdots \\ p_{n1}(k) & \cdots & p_{nn}(k) \end{pmatrix}, k = 1, 2, \dots, n$$

де  $p_{ij}(k) \geq 0$ .

Оскільки випадкові події - перехід системи з фіксованого стану  $s_i$  в будь-який можливий стан  $s_j$ , створюють повну групу подій, то

$$\sum_{j=1}^n p_{ij}(k) = 1$$

(ми також зазначимо, що набір подій називається повним, якщо передбачається, що принаймні одна з них відбудеться в результаті тесту.

Якщо події  $B_1, B_2, \dots, B_n$  є попарно несумісні і мають додатні ймовірності, які в сумі дорівнюють одиниці, то для будь-якої події  $A$  застосовується наступна формула:

$$P(A) = P(A|B_1) \cdot P(B_1) + P(A|B_2) \cdot P(B_2) + \cdots + P(A|B_n) \cdot P(B_n)$$

**Означення.** Ланцюг Маркова називається однорідним, якщо умовна ймовірність  $p_{ij}(k)$  переходу системи зі стану  $s_i$  в стан  $s_j$  не залежить від кроку системи  $k$ , тобто

$$p_{ij}(k) = p_{ij} = \text{const.}$$

Далі ми розглянемо однорідні ланцюги Маркова.

Матриця перехідних ймовірностей для однорідних ланцюгів Маркова має вигляд

$$P = \begin{pmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{pmatrix}.$$

Властивості матриці цілком очевидні

- 1) усі елементи матриці невід'ємні;
- 2) для кожного рядка матриці проходів виконується наступна умова:

$$\sum_{j=1}^n p_{ij} = 1$$

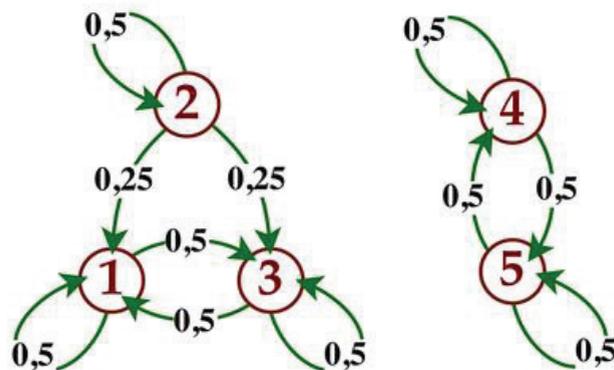
Ланцюг Маркова вважається заданим, якщо задано початковий розподіл ймовірностей і матрицю перехідних ймовірностей.

**Означення.** Вектор з невід'ємними координатами, сума яких дорівнює одиниці, називається стохастичним.

**Означення.** Квадратна матриця з невід'ємними елементами, сума яких у кожному рядку дорівнює одиниці, називається стохастичною.

Для наочності стану ланцюгів Маркова і ймовірності переходу системи з одного стану в інший зручно навести імовірнісний граф. У загальному випадку графи зображуються вершинами, які розміщені на площині в певному порядку, і лініями, що з'єднують вершини – ребрами. Вершина графа інформує про стан, в якому може перебувати система, а ребро графа, що з'єднує дві вершини, вказує на стан, в який система може перейти з певною ймовірністю.

$$P = \begin{pmatrix} 1/2 & 0 & 1/2 & 0 & 0 \\ 1/4 & 1/2 & 1/4 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 1/2 \end{pmatrix}$$



Розглянемо наступну задачу: за ймовірностями переходу  $p_{ij}$  потрібно знайти ймовірності переходу системи  $S$  зі стану  $s_i$  в стан  $s_j$  за  $l$  кроків.

**Приклад.** Знайти матрицю ймовірностей переходу від стану до стану ланцюга Маркова після 2 і 3 кроків, якщо матриця переходу після першого кроку рівна:

$$P = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

**Розв'язання.** Оскільки  $\|p_{ij}^{(2)}\| = P^2$ , то, помноживши матриці, отримаємо:

$$P^2 = P \cdot P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 1/2 \\ 0 & 1/2 & 1/2 \end{pmatrix},$$

$$P^3 = P^2 \cdot P = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

Позначимо  $P_i(m)$  – безумовну ймовірність, що система за  $m$  кроків перейде до стану  $s_i$ . Тоді множина ймовірностей  $P_i(m)$  утворить вектор, який назвемо вектором розподілу ймовірностей відносно станів:

$$P(m) = (P_1(m), P_2(m), \dots, P_k(m)).$$

Для цього вектора виконуються стандартні властивості розподілу ймовірностей

$$0 \leq P_i(m) \leq 1, \quad i = 1, 2, \dots, k;$$

$$\sum_{i=1}^k P_i(m) = 1$$

Виявляється, що

$$P(m) = P(0) \cdot P^m,$$

де  $P(0)$  – початковий вектор ймовірностей станів системи

Розподіл ймовірностей станів системи  $P_{st} = (P_1, P_2, \dots, P_k)$  називається стаціонарним, якщо він не змінюється від кроку до кроку, тобто

$$P_{st} = P_{st} \cdot P$$

або

$$P_{st} \cdot (P - E) = 0,$$

де  $E$  – одинична матриця.

Цьому матричному рівнянню відповідає система лінійних однорідних рівнянь, яка має ненульовий розв'язок за умови

$$\det(P - E) = 0.$$

Доповненням цієї системи рівнянь повинен бути умова нормалізації ймовірностей

$$\sum_{i=1}^k P_i = 1.$$

**Приклад.** Матриця ймовірностей переходу ланцюга Маркова має вигляд

$$P = \begin{pmatrix} 0.4 & 0.6 \\ 0.5 & 0.5 \end{pmatrix}$$

Розподіл ймовірностей згідно станів у початковому моменті часу характеризується вектором

$$p = (0.2, 0.8),$$

Треба знайти розподіл ймовірностей згідно станів на 2-му кроці, а також стаціонарний розподіл ймовірностей.

**Розв'язання.** Маємо

$$P(2) = P(0) \cdot P^2;$$

$$P^2 = \begin{pmatrix} 0.4 & 0.6 \\ 0.5 & 0.5 \end{pmatrix} \begin{pmatrix} 0.4 & 0.6 \\ 0.5 & 0.5 \end{pmatrix} = \\ = \begin{pmatrix} 0.4 \cdot 0.4 + 0.6 \cdot 0.5 & 0.4 \cdot 0.6 + 0.6 \cdot 0.5 \\ 0.5 \cdot 0.4 + 0.5 \cdot 0.5 & 0.5 \cdot 0.6 + 0.5 \cdot 0.5 \end{pmatrix} = \begin{pmatrix} 0.46 & 0.54 \\ 0.45 & 0.55 \end{pmatrix};$$

Отже,

$$P(2) = (0.2, 0.8) \cdot \begin{pmatrix} 0.46 & 0.54 \\ 0.45 & 0.55 \end{pmatrix} = \\ = (0.2 \cdot 0.46 + 0.8 \cdot 0.45, 0.2 \cdot 0.54 + 0.8 \cdot 0.55) = \\ = (0.452, 0.548)$$

і отримали розподіл ймовірностей згідно станів після двох кроків

Щоб обчислити стаціонарний розподіл, сконструюємо систему рівнянь

$$\begin{cases} (P_1, P_2) \begin{pmatrix} 0.4 & 0.6 \\ 0.5 & 0.5 \end{pmatrix} = (P_1, P_2), \\ P_1 + P_2 = 1. \end{cases}$$

тобто

$$\begin{cases} 0.4P_1 + 0.5P_2 = P_1, \\ 0.6P_1 + 0.5P_2 = P_2, \\ P_1 + P_2 = 1. \end{cases}$$

В останній системі перше та друге рівняння є лінійно залежними, тому одне з них можна вилучити. З першого рівняння маємо

$$P_2 = 1.2 \cdot P_1$$

І, беручи до уваги третє рівняння, отримаємо

$$P_1 = \frac{1}{2.2} = 5/11, \quad P_2 = 1 - P_1 = 6/11,$$

а значить, стаціонарний розподіл має вигляд

$$P_{st} = \left( \frac{5}{11}, \frac{6}{11} \right).$$

Значення отриманого результату є наступне: в стані стаціонарним система проводить 5/11 часу в стані  $s_1$ , і 6/11 часу в стані  $s_2$ .

### 3.6. Генерація випадкових процесів у Python

NumPy надає генератори базових розподілів: `np.random.randn(n)` для  $N(0,1)$ , `np.random.exponential(scale, n)` для  $\text{Exp}(1/\text{scale})$ , `np.random.poisson(lam, n)` для  $\text{Poisson}(\text{lam})$ . Для відтворюваності використовують `np.random.seed(value)` або сучасніший генератор `np.random.default_rng(seed)`.

Вінерівський процес генерується як кумулятивна сума приростів: `dt = T/N`;  
`dW = np.sqrt(dt) * np.random.randn(N)`; `W = np.concatenate([[0], np.cumsum(dW)])`.

Для пуассонівського процесу генерують інтервали: `intervals = np.random.exponential(1/lam, n_events)`; `times = np.cumsum(intervals)`.

**Контрольні питання:**

1. Які властивості має вінерівський процес?
2. Як пов'язані інтервали між подіями та інтенсивність пуассонівського процесу?
3. Як згенерувати траєкторію вінерівського процесу в Python?

## ЛЕКЦІЯ 4

### Стохастичні процеси та випадкові збурення (продовження)

#### 4.1. Дифузійні процеси

**Означення.** Випадковий процес Маркова називається **дифузійним процесом**, якщо виконуються умови:

$$\begin{aligned}\lim_{t \rightarrow t_0+} \frac{1}{t - t_0} \int_{|x-x_0| \geq \delta} P(t_0, x_0, t, dx) &= 0, \\ \lim_{t \rightarrow t_0+} \frac{1}{t - t_0} \int_{|x-x_0| < \delta} (x - x_0) P(t_0, x_0, t, dx) &= a(t_0, x_0), \\ \lim_{t \rightarrow t_0+} \frac{1}{t - t_0} \int_{|x-x_0| < \delta} (x - x_0)^2 P(t_0, x_0, t, dx) &= b(t_0, x_0),\end{aligned}$$

для довільних  $\delta > 0, t_0, x_0$ .

Функція  $a(t_0, x_0)$  називається коефіцієнтом дрейфу, функція  $b(t_0, x_0)$  – коефіцієнтом дифузії.

Як початковий розподіл і матриця переходів задають інформацію для дискретного ланцюга, так функції  $a$  і  $b$  задають основну інформацію про дифузійний процес.

Можна довести, що процес Вінера є дифузійним процесом.

Процес Вінера є дифузійним процесом:

$$a \equiv 0, b \equiv 1.$$

Існують альтернативні підходи до дослідження дифузійних процесів:

**Підхід Колмогорова** (рівняння в частинних похідних): описує еволюцію густини ймовірності, працює з гладкими функціями;

**Стохастичний підхід** (стохастичні диференціальні рівняння): описує окремі траєкторії  $X(t)$ , які є недиференційовними.

#### 4.2. Стохастичні диференціальні рівняння

Іто запропонував означення інтеграла

$$\int_0^T \eta(t) dW(t)$$

як границі інтегральних сум

$$\sum_{i=0}^n \eta(t_i)(W(t_{i+1}) - W(t_i)),$$

де  $0 = t_0 < \dots < t_n = T$ .

Якщо замінити  $\eta(t_i)$  на  $\eta(t_{i+1})$ , границя змінюється, що принципово відрізняє стохастичний інтеграл від інтеграла Рімана.

Стохастичне диференціальне рівняння записують у диференціалах:

$$dX = F(X)dt + G(X)dW(t),$$

але за змістом це інтегральне рівняння:

$$X(t) = X(0) + \int_0^t F(X(s))ds + \int_0^t G(X(s))dW(s).$$

**Приклад.** Процес Орнштейна–Уленбека

$$dx = -\beta(x - \alpha)dt + \sigma dW$$

описує блукання з притяганням до рівня рівноваги  $\alpha$ .

Розв'язок:

$$x(t) = \alpha + (x_0 - \alpha)e^{-\beta t} + \frac{\sigma}{\sqrt{2\beta}} \sqrt{1 - e^{-2\beta t}} \varepsilon.$$

Процес має нормальний розподіл із середнім, що прямує до  $\alpha$ , та сталою дисперсією.

При малих  $\beta$  процес Орнштейна–Уленбека наближається до вінерівського блукання, але не відходить у нескінченність.

### 4.3. Процес Леві як розв'язок стохастичного рівняння Іто–Скорохода

Почнемо з введення означень, рівнянь і позначень.

**Означення 1.** Процес  $\{X_t, t \geq 0\}$  називається **процесом Леві**, якщо:

- траєкторії  $X_t$  є правосталими та мають ліві границі майже всюди;
- $P(X_0 = 0) = 1$ ;
- для  $s < t$  розподіл  $X_t - X_s$  збігається з розподілом  $X_{t-s}$ ;
- для  $0 \leq s \leq t$  розподіл  $X_t - X_s$  не залежить від  $\{X_u, u \leq s\}$ .

Розглядається процес Леві, заданий стохастичним диференціальним рівнянням Іто–Скорохода:

$$dX_t = \mu(X_t, t)dt + \sigma(X_t, t)dW_t + \int_U \gamma(X_t, t) \tilde{\nu}(dt, du), X_0 = 0.$$

Тут  $W_t$  – стандартний броунівський процес,  $\mu$  – зсув,  $\sigma$  – дифузійний коефіцієнт.

**Означення.** Міра Пуассона  $\nu(t, A)$  на  $\mathbb{R}^+ \times U$  така, що:

а) для фіксованого  $A$  процес  $\nu(t, A)$  є пуассонівським:

$$P(\nu(t, A) - \nu(s, A) = k) = \frac{\gamma^k (t - s)^k}{k!} e^{-\gamma(t-s)};$$

б) для будь-яких  $A_1, \dots, A_n$  відповідні процеси є незалежними.

Центрована міра Пуассона визначається як

$$\tilde{\nu}(dt, du) = \nu(dt, du) - E\nu(dt, du),$$

з інтенсивністю

$$\lambda(t) = \int_U \frac{dudt}{u^2}.$$

Міра  $\tilde{\nu}$  може бути побудована наступним чином:

Нехай  $X_t$  – однорідний процес із незалежними приростами, що набуває значень у  $\mathbb{R}$ , для якого виконується наступна рівність:

$$\begin{aligned} E\{\exp(i(z, X_t))\} \\ = \exp \left\{ t \left[ \int_{|u| < 1} (\exp(i(z, u)) - 1 - i(z, u)) \frac{du}{u^2} + \int_{|u| \geq 1} (\exp(i(z, u)) - 1) \frac{du}{u^2} \right] \right\} \end{aligned}$$

Такий процес  $X_t$  з ймовірністю одиниця не має розривів другого роду.

Якщо ми позначимо

$$A = [0, T] \times \mathbb{R}^1$$

і  $\nu(A)$  – кількість розривів процесу  $X_t$ , для яких точка

$$(t, X_{t+0} - X_{t-0})$$

належить до  $A$ , тоді  $\nu$  матиме розподіл Пуассона з параметром (2), а для множин  $A_k$ , що не перетинаються, значення  $\nu(A_k)$  будуть незалежними.

Нагадаємо також визначення інтеграла за центрованою мірою Пуассона для східчастих функцій.

Якщо є

$$0 < t_1 < t_2 < \dots < T_n$$

та

$$\{B_1, B_2, \dots, B_n\} \subset \mathbb{R}, \bigcup_j B_j = \mathbb{R},$$

таких, що  $\gamma(t, u)$  є константами на множинах

$$[t_k, t_{k+1}) \times B_j,$$

тоді, беручи  $u_j \in B_j$ , ми припускаємо:

$$\int_0^T \int_{\mathbb{R}} \gamma(t, u) \tilde{v}(dt, du) = \sum_{\substack{0 \leq k \leq n-1 \\ 1 \leq j \leq n}} \gamma(t_k, u_j) \tilde{v}([t_k, t_{k+1}) \times B_j)$$

Також важливо розуміти, за яких умов рівняння (1) матиме розв'язок.

Для рівняння (1) умови існування та єдиності виглядають так.

**Теорема 1.** Нехай коефіцієнти рівняння (1) задовольняють наступні умови:

$$\mu(s, 0), \sigma(s, 0), \int_U \gamma(s, 0, \theta) \Pi(d\theta)$$

є локально обмеженими за  $s$ ,

існує зростаюча функція  $l(s)$  така, що

$$\frac{|\mu(s, x) - \mu(s, y)|^2 + |\sigma(s, x) - \sigma(s, y)|^2 + \int_U |\gamma(s, x, \theta) - \gamma(s, y, \theta)|^2 \Pi(d\theta)}{|x - y|^2} \leq l(s)$$

Позначимо  $\tilde{F}_t \sigma$ -алгебру, породжену змінними

$$\xi(0), \nu(d\theta \times ds), w(s), s \leq t.$$

Якщо  $\xi(0)$  не залежить від сукупності змінних  $w(s), \nu(d\theta \times ds)$  та

$$E |\xi(0)|^2 < \infty,$$

тоді рівняння має  $\tilde{F}_t$ -вимірний розв'язок, і

$$E |\xi(s)|^2 < \infty.$$

**Теорема 2.** Ми припускаємо, що коефіцієнти рівняння (1) задовольняють наступну умову: для кожного  $c > 0$  існує константа  $l_c$ , для якої

$$\frac{|\mu(s, x) - \mu(s, y)|^2 + |\sigma(s, x) - \sigma(s, y)|^2 + \int_U |\gamma(s, x, \theta) - \gamma(s, y, \theta)|^2 \Pi(d\theta)}{|x - y|^2} \leq l_c$$

як тільки

$$|x| \leq r, |y| \leq r, s \leq r.$$

Тоді розв'язок рівняння (1) для даного  $\xi(0)$  є єдиним.

Розглядається випадкова еволюція у вигляді стохастичного диференціального рівняння в ергодичному середовищі Маркова:

$$du^\varepsilon(t) = C(u^\varepsilon, x(t/\varepsilon^2))dt + d\eta^\varepsilon(t), u^\varepsilon(t) \in \mathbb{R}. \quad (1)$$

Це складна система, визначена стохастичним диференціальним рівнянням із додатками імпульсного типу.

Складність системи виникає з присутності перемикання Маркова і схеми апроксимації.

Ось  $x(t)$  – рівномірно ергодичний процес Маркова в стандартному фазовому просторі  $(X, \mathcal{X})$ , визначений генератором

$$Q\varphi(x) = q(x) \int_X P(x, dy) [\varphi(y) - \varphi(x)]$$

на просторі Банаха  $B(X)$  дійсних значень функцій  $\varphi(x)$ , обмежених максимумом норми

$$\|\varphi(x)\| = \max_{x \in X} |\varphi(x)|.$$

Стохастичне ядро (stochastic kernel)

$$P(x, B), x \in X, B \in \mathcal{X}$$

визначає рівномірно ергодичний вкладений ланцюг Маркова

$$x_n = x(\tau_n), n \geq 0,$$

зі стаціонарним розподілом  $\rho(B)$ .

Стаціонарний розподіл  $\pi(B)$  процесу Маркова  $x(t)$  можна визначити з реляції

$$\pi(dx)q(x) = q\rho(dx), q = \int_X \pi(dx)q(x).$$

Визначимо  $R_0$  як потенціальний оператор для генератора  $Q$ :

$$R_0 = \Pi - (\Pi + Q)^{-1},$$

де

$$P\varphi(x) = \int_X \pi(dy) \varphi(y) 1(x)$$

є проектором у підпросторі

$$N_Q = \{\varphi: Q\varphi = 0\}.$$

**Ще кілька зауважень про генератор  $Q$ :** важливою властивістю такого генератора є те, що простір, в якому він визначений, розкладається на безпосередню суму його нульового підпростору (тобто ті функції, які оператор  $Q$  перетворює на нуль, що настає завдяки інваріантній мірі) і підпростору значень (тобто ті функції, на яких оператор діє в спосіб нетривіальний). Отже, в підпросторі ядер знаходиться проектор.

Потенціальний оператор  $R_0$  був запроваджений, оскільки в присутності нульового підпростору ми не можемо на всій траєкторії сконструювати оператора, оберненого до оператора  $Q$ .

А конструкція оператора  $R_0$  уможлиблює побудову оператора, оберненого до  $Q$  в підпросторі вартості, і трансформацію на нуль функцій з підпростору ядер.

**Приклад.** Процес Орнштейна–Уленбека  $x(t)$  даний через розв'язання стохастичного рівняння

$$dx(t) = -cx(t) + \sigma dW(t),$$

де  $c > 0, \sigma > 0, W(t)$  – стандартний процес Вінера.

В просторі подвійних функцій диференціювання процес Орнштейна–Уленбека даний через генератор

$$Q\varphi(x) = \frac{\sigma^2}{2} \varphi''(x) - cx\varphi'(x).$$

Позначимо  $C_0(\mathbb{R})$  простір неперервних функцій вигляду  $\varphi + c, c = \text{const}$ , де  $\varphi$  дорівнює нулю в нескінченності.

Стаціонарна міра процесу Орнштейна–Уленбека має вигляд

$$\pi(x) = \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-x^2/2\sigma_0^2}, \sigma_0^2 = \frac{\sigma^2}{2c}.$$

Потенціальний оператор процесу має вигляд:

$$R_0\varphi(x) = \int_{-\infty}^{\infty} R_0(x, y)\varphi(y) dy,$$

де

$$R_0(x, y) = \frac{2}{\sigma^2} \int_{-\infty}^{x \wedge y} \exp \left[ -\frac{y^2 - z^2}{2\sigma_0^2} \right] dz.$$

Розглянемо граничну поведінку випадкової еволюції  $u^\varepsilon(t)$ , подану через рівняння

$$\frac{d}{dt} u^\varepsilon(t) = C^\varepsilon(u^\varepsilon(t), x(t/\varepsilon^2)), u(0) = u \in \mathbb{R}, \varepsilon \rightarrow 0, \varepsilon > 0,$$

де  $x(t)$  є процесом перемикавання Орнштейна–Уленбека, а

$$C^\varepsilon(u, x) = C(u, x) + \varepsilon^{-1} C_0(x).$$

Функція  $C(u, x)$  має фізичне значення швидкості еволюції з огляду на наступну умову рівноваги:

$$PC_0(x) = \int_{-\infty}^{\infty} \pi(dx) C_0(x) = 0.$$

Це генерує флуктуації, а в консеквенції – дифузю в граничному процесі.

Коли умова рівноваги виконана, граничний оператор

$$\hat{L}\varphi(u) = [PC(u, x)P + PC_0(x)PR_0C_0(x)P]\varphi(u)$$

має вигляд

$$\hat{L}\varphi(u) = \hat{C}(u)\varphi'(u) + \frac{1}{2}B\varphi''(u),$$

і визначає певний гауссівський процес, а компоненти генератора можуть бути явно обчислені для конкретних функцій  $C(u, x)$ ,  $C_0(x)$ .

### Потенціал ланцюга Маркова

Розглянемо ергодичний ланцюг Маркова

$$\{\xi_n, n \geq 0\}$$

на скінченному фазовому просторі

$$X = \{1, 2, \dots, m\}.$$

Матриця ймовірностей переходу

$$P = \| p_{ij} \|_{i,j \in X}$$

задовольняє умови ергодичності і має ергодичний розподіл

$$\vec{\pi} = (\pi_1, \dots, \pi_m).$$

**Означення.** Матриця

$$\Pi = \begin{pmatrix} \pi_1 & \pi_2 & \dots & \pi_m \\ \pi_1 & \pi_2 & \dots & \pi_m \\ \vdots & \vdots & \ddots & \vdots \\ \pi_1 & \pi_2 & \dots & \pi_m \end{pmatrix}$$

розміру  $m \times m$  називається власним проектором матриці  $I - P$ .

Нехай

$$\vec{1} = (1, \dots, 1)^T.$$

Тоді

$$\Pi = \vec{1}\vec{\pi} = \vec{1} \otimes \vec{\pi}.$$

Проектор має наступні властивості:

$$\Pi^2 = \Pi, P\Pi = \Pi, \vec{\pi}\Pi = \vec{\pi}, (I - P)\Pi = \Pi(I - P) = 0.$$

Матриця

$$I - P + \Pi$$

є невиродженою, тобто існує обернена матриця

$$(I - P + \Pi)^{-1}.$$

**Означення.** Матриця

$$R_0 = (I - P + \Pi)^{-1} - \Pi$$

називається **потенціалом ланцюга Маркова**  $\{\xi_n, n \geq 0\}$ .

Матриця  $R_0$  також називається узагальненою оберненою матрицею до  $I - P$ .

**Приклад.** Матриця ймовірностей переходу має вигляд

$$P = \begin{pmatrix} 0.4 & 0.6 \\ 0.3 & 0.7 \end{pmatrix}.$$

Знайти потенціал ланцюга Маркова  $\xi_n$ .

**Розв'язання.** Знайдемо інфінітезимальну матрицю:

$$A = I - P = \begin{pmatrix} 0.6 & -0.6 \\ -0.3 & 0.3 \end{pmatrix}.$$

Знайдемо стаціонарний розподіл:

$$(\pi_1, \pi_2) = (\pi_1, \pi_2) \begin{pmatrix} 0.4 & 0.6 \\ 0.3 & 0.7 \end{pmatrix},$$

тобто

$$\begin{cases} \pi_1 = 0.4\pi_1 + 0.3\pi_2, \\ \pi_2 = 0.6\pi_1 + 0.7\pi_2. \end{cases}$$

Звідси

$$\vec{\pi} = \left(\frac{1}{3}, \frac{2}{3}\right).$$

За означенням

$$\Pi = \vec{1}\vec{\pi} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} \frac{1}{3} & \frac{2}{3} \end{pmatrix} = \begin{pmatrix} \frac{1}{3} & \frac{2}{3} \\ \frac{1}{3} & \frac{2}{3} \end{pmatrix}.$$

Спочатку:

$$I - P = \begin{pmatrix} 0.6 & -0.6 \\ -0.3 & 0.3 \end{pmatrix}.$$

$$I - P + \Pi = \begin{pmatrix} 0.6 + \frac{1}{3} & -0.6 + \frac{2}{3} \\ -0.3 + \frac{1}{3} & 0.3 + \frac{2}{3} \end{pmatrix} = \begin{pmatrix} \frac{14}{15} & \frac{1}{15} \\ \frac{1}{30} & \frac{29}{30} \end{pmatrix}.$$

Обчислюємо визначник:

$$\det = \frac{14}{15} \cdot \frac{29}{30} - \frac{1}{15} \cdot \frac{1}{30} = \frac{405}{450} = \frac{9}{10}.$$

Отже,

$$\left( I - P + \Pi \right)^{-1} = \frac{10}{9} \begin{pmatrix} \frac{29}{30} & -\frac{1}{15} \\ -\frac{1}{30} & \frac{14}{15} \end{pmatrix} = \begin{pmatrix} \frac{29}{27} & -\frac{2}{27} \\ -\frac{1}{27} & \frac{28}{27} \end{pmatrix}.$$

За означенням:

$$R_0 = (I - P + \Pi)^{-1} - \Pi.$$

Віднімаємо П:

$$R_0 = \begin{pmatrix} \frac{29}{27} & -\frac{2}{27} \\ -\frac{1}{27} & \frac{28}{27} \end{pmatrix} - \begin{pmatrix} \frac{1}{3} & \frac{2}{3} \\ \frac{1}{3} & \frac{2}{3} \end{pmatrix} = \begin{pmatrix} \frac{20}{27} & -\frac{20}{27} \\ -\frac{10}{27} & \frac{10}{27} \end{pmatrix}.$$

**Контрольні питання:**

1. Опишіть алгоритм методу знаходження потенціалу.
2. Чи є пуассонівський процес марковським? Чому?
3. З яких елементів складається стохастичне рівняння Іто?
4. Що таке однорідний ланцюг Маркова?

## ЛЕКЦІЯ 5

### Моделювання інформаційної війни

#### 5.1. Модель поширення інформації

Нехай є деяка соціальна спільнота чисельністю  $N_0$  (культурна, вікова, політична, члени Інтернет-спільноти тощо), потенційно схильна до впливу деякої інформаційної хвилі, тобто, наприклад, загрози зміни її стану шляхом передачі інформаційними каналами відомостей, важливих для даної групи.

У момент часу  $t = t_0$  ( $t_0 = 0$ ) загроза актуалізується, тобто джерело інформаційної хвилі вживає дій щодо її реалізації, внаслідок чого інформаційна хвиля починає поширюватися серед спільноти.

Основною поточною характеристикою ступеня поширеності (небезпеки) інформаційна хвиля вважається величина  $N(t)$  – залежне від часу  $t$  число «адептів», які прийняли нові відомості, цінності, ідеї, норми тощо, що поширюються джерелом інформаційної хвилі ( $N(t_0) = N(0) = 0$ ).

Основні модельні припущення такі:

Інформаційна хвиля поширюється серед спільноти двома інформаційними каналами.

**а)** перший з них – «зовнішній» по відношенню до спільноти, наприклад, рекламна кампанія, що проводиться через ЗМІ. Її інтенсивність (число рівноцінних інформаційних актів в одиницю часу) характеризується параметром  $\alpha_{11} > 0$ , який вважається незалежним від часу.

**б)** другий, «внутрішній» канал – міжособистісне спілкування членів соціальної спільноти (його інтенсивність, тобто число рівноцінних інформаційних контактів, характеризується параметром  $\alpha_{12} > 0$ , також незалежним від часу).

В результаті особистого спілкування вже «завербовані» адепти (їх число дорівнює величині  $N(t)$ ), впливаючи на ще не завербованих членів (їх чисельність дорівнює величині  $N_0 - N(t)$ ), вносять свій додатковий «особистий» внесок у процес вербування.

Зауважимо, стосовно першого каналу тут вважається, що спільнота, яка піддається впливу, знаходиться у свого роду «всеосяжному нелокальному інформаційному полі», тобто будь-який зі ще не завербованих членів спільноти (їх число дорівнює  $N_0 - N(t)$ ) завжди має можливість отримати інформацію, що поширюється за зовнішнім каналом, і з певною ймовірністю її сприйняти.

Тим самим, швидкість зовнішнього вербування визначається, крім фактора  $N_0 - N(t)$  і величини  $\alpha_{11}$ , також і величиною  $\alpha_{12}$  – згаданою ймовірністю (схильністю) сприйняття інформації, що залежить, наприклад, від довіри до неї.

Зазначимо також, що на відміну від зовнішнього, внутрішній канал має локальний характер (від людини до людини), але швидкість вербування, як і в першому випадку, знову-таки пропорційна числу ще не завербованих членів спільності – величині  $N_0 - N(t)$ , інтенсивності контактів  $\alpha_{21}$  і схильності до сприйняття інформації по другому каналу – величині  $\alpha_{22}$ .

**2. Швидкість зміни числа adeptів  $N(t)$**  (тобто число завербованих за одиницю часу) складається з:

**а)** швидкості зовнішнього вербування (вона пропорційна добутку інтенсивності  $\alpha_{11}$  на ймовірність  $\alpha_{12}$  і на число ще не завербованих членів  $N_0 - N(t)$ ), тобто величини

$$\alpha_{11}\alpha_{12}(N_0 - N(t)) = \alpha_1(N_0 - N(t)),$$

де  $\alpha_1 = \alpha_{11}\alpha_{12}$ ;

**б)** швидкості внутрішнього вербування (вона пропорційна добутку інтенсивності  $\alpha_{21}$  на ймовірність  $\alpha_{22}$ , на число діючих adeptів  $N(t)$  і на число не завербованих  $N_0 - N(t)$ ), тобто величини

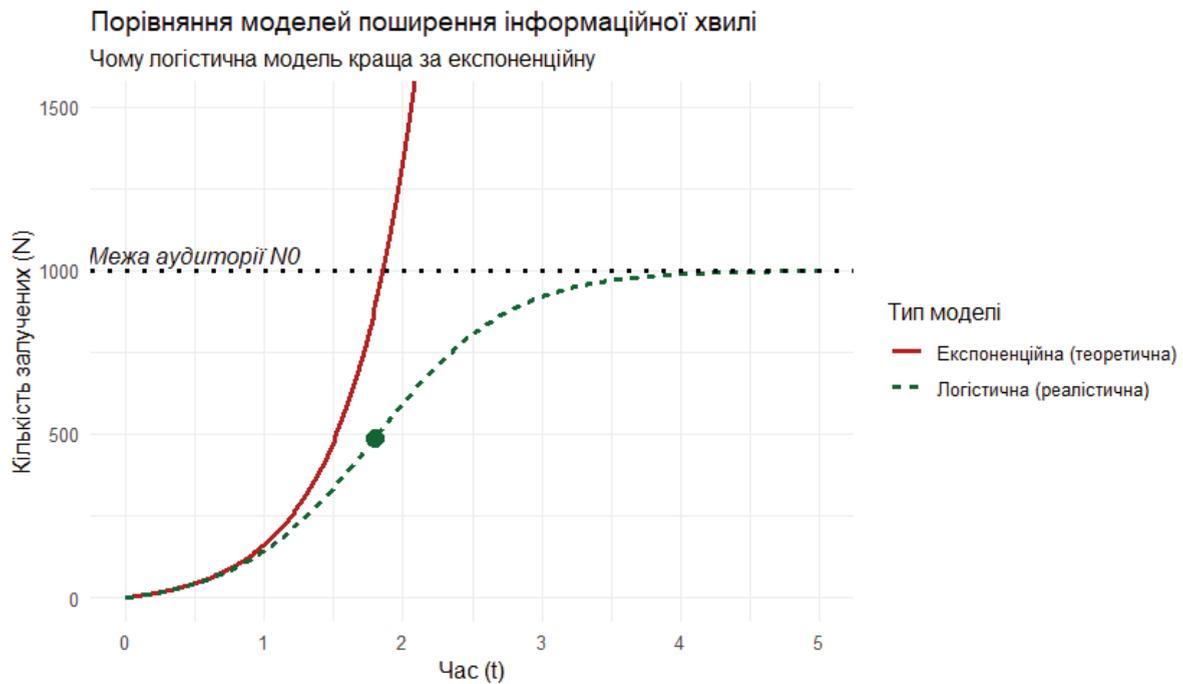
$$\alpha_{21}\alpha_{22}N(t)(N_0 - N(t)) = \alpha_2N(t)(N_0 - N(t)),$$

де  $\alpha_2 = \alpha_{21}\alpha_{22}$ .

Підсумовуючи припущення 1 і 2, приходимо до моделі:

$$\frac{dN(t)}{dt} = (\alpha_1 + \alpha_2N(t))(N_0 - N(t)), N(0) = 0, t > 0.$$

У математичному відношенні вона має явний розв'язок - узагальнення добре відомої логістичної моделі (що отримується з рівняння при  $\alpha_1 = 0$ , тобто за відсутності зовнішнього впливу).



Розв'яжемо задачу при  $t \geq t_0 = 0$ , що задовольняє початкову умову  $N(0) = 0$ :

$$\frac{dN}{dt} = (\alpha_1 + \alpha_2 N)(N_0 - N), N(0) = 0$$

Відокремлюємо змінні:

$$\frac{dN}{(\alpha_1 + \alpha_2 N)(N_0 - N)} = dt$$

Розклад на елементарні дроби

$$\frac{1}{(\alpha_1 + \alpha_2 N)(N_0 - N)} = \frac{A}{\alpha_1 + \alpha_2 N} + \frac{B}{N_0 - N}$$

Зведемо до спільного знаменника:

$$1 = A(N_0 - N) + B(\alpha_1 + \alpha_2 N)$$

Знайдемо коефіцієнти при  $N = N_0$

$$1 = B(\alpha_1 + \alpha_2 N_0) \Rightarrow B = \frac{1}{\alpha_1 + \alpha_2 N_0}$$

та при  $N = -\frac{\alpha_1}{\alpha_2}$

$$1 = A\left(N_0 + \frac{\alpha_1}{\alpha_2}\right) = A\left(\frac{\alpha_2 N_0 + \alpha_1}{\alpha_2}\right) \Rightarrow A = \frac{\alpha_2}{\alpha_1 + \alpha_2 N_0}$$

Отже, інтеграл набуває вигляду:

$$\frac{1}{\alpha_1 + \alpha_2 N_0} \int \left( \frac{\alpha_2}{\alpha_1 + \alpha_2 N} + \frac{1}{N_0 - N} \right) dN = \int dt$$
$$\frac{1}{\alpha_1 + \alpha_2 N_0} [\ln(\alpha_1 + \alpha_2 N) - \ln(N_0 - N)] = t + C$$
$$\ln \left( \frac{\alpha_1 + \alpha_2 N}{N_0 - N} \right) = (\alpha_1 + \alpha_2 N_0)t + C_1$$

Використаємо початкову умову  $N(0) = 0$

$$\ln \left( \frac{\alpha_1}{N_0} \right) = C_1;$$

$$\ln \left( \frac{\alpha_1 + \alpha_2 N}{N_0 - N} \right) = (\alpha_1 + \alpha_2 N_0)t + \ln \left( \frac{\alpha_1}{N_0} \right)$$

Знайдемо  $N(t)$  ( $e^{\ln x} = x$ ):

$$\frac{N_0(\alpha_1 + \alpha_2 N)}{\alpha_1(N_0 - N)} = e^{(\alpha_1 + \alpha_2 N_0)t}$$

Остаточна формула:

$$N(t) = N_0 \frac{1 - e^{-(\alpha_1 + \alpha_2 N_0)t}}{1 + \frac{\alpha_2 N_0}{\alpha_1} e^{-(\alpha_1 + \alpha_2 N_0)t}}$$

або

$$N(t) = \frac{N_0 \exp((\alpha_1 + \alpha_2 N_0)t) - \alpha_1}{(\alpha_2 + \alpha_1/N_0) \exp((\alpha_1 + \alpha_2 N_0)t)}$$

Це розв'язок, і подивимось асимптотику:

$$\lim_{t \rightarrow \infty} N(t) = N_0 \frac{1 - \lim_{t \rightarrow \infty} e^{-(\alpha_1 + \alpha_2 N_0)t}}{1 + \frac{\alpha_2 N_0}{\alpha_1} \lim_{t \rightarrow \infty} e^{-(\alpha_1 + \alpha_2 N_0)t}}$$
$$= N_0 \frac{1 - 0}{1 + \frac{\alpha_2 N_0}{\alpha_1} \cdot 0} = N_0 \cdot \frac{1}{1} = N_0$$

Модель описує поширення інформаційної хвилі у режимі, коли в кінцевому підсумку (за досить великий час, формально при  $t \rightarrow \infty$ ) усі члени групи будуть «завербовані».

Стан рівноваги:

$$\frac{dN(t)}{dt} = (\alpha_1 + \alpha_2 N(t))(N_0 - N(t)) = 0,$$

тобто ми маємо дві критичні точки рівноваги:

$$N_1^* = N_0$$

– стан повного насичення, коли інформаційна хвиля охопила всю потенційну аудиторію.

$$N_2^* = -\frac{\alpha_1}{\alpha_2}$$

– формальна математична точка рівноваги, яка лежить поза межами змістовного аналізу для реальних процесів (оскільки кількість людей не може бути від'ємною).

Проаналізуємо тепер швидкість наростання інформаційної хвилі залежно від параметрів моделі. З цією метою скористаємося апаратом математичного аналізу, зокрема дослідженням другої похідної. Це дозволить нам визначити момент найвищої інтенсивності процесу та зміну його темпів.

#### **Дослідження функції за допомогою другої похідної**

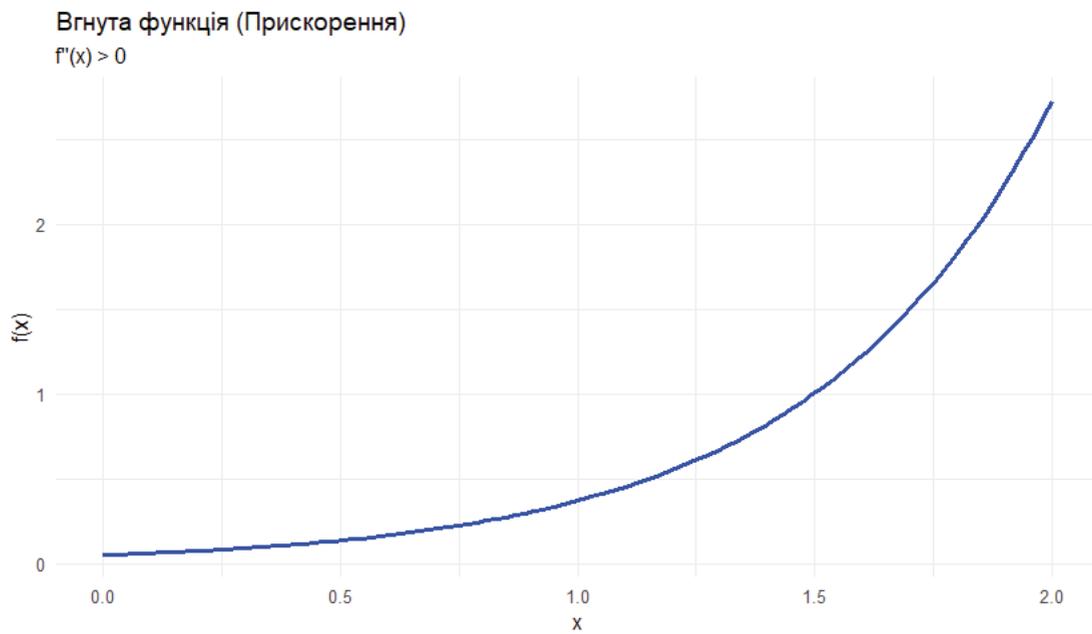
Для аналізу характеру зміни функції (прискорення або сповільнення росту) у математичному аналізі використовують другу похідну  $f''(x)$ .

#### **Опуклість та вгнутість**

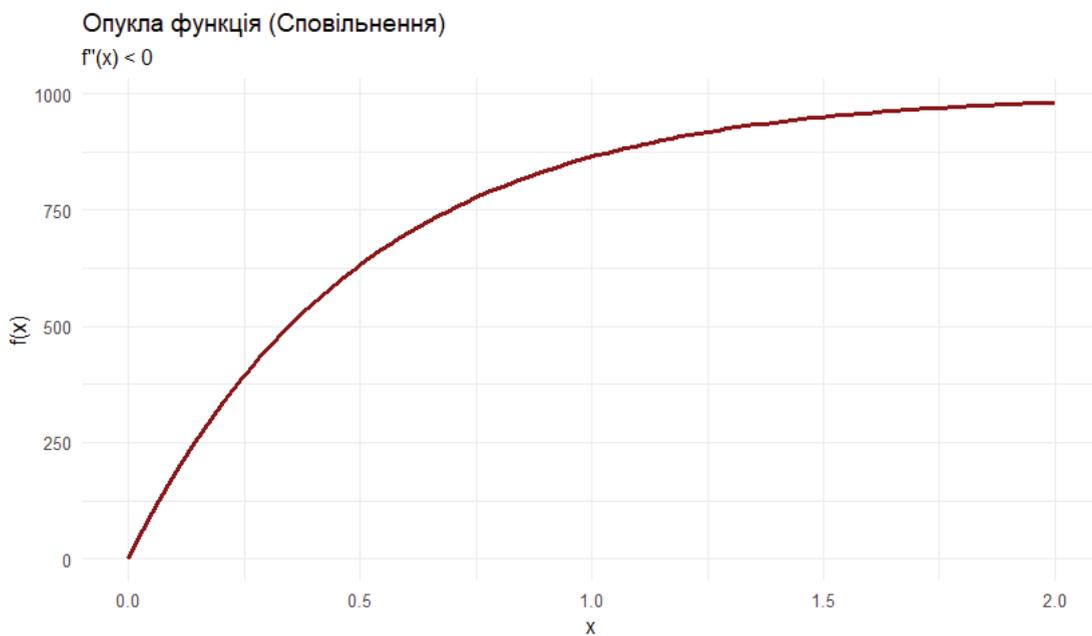
Вгнутість (опуклість донизу). Якщо на певному інтервалі

$$f''(x) > 0,$$

то графік функції є вгнутим. Це означає, що швидкість росту самої функції збільшується (процес прискорюється).

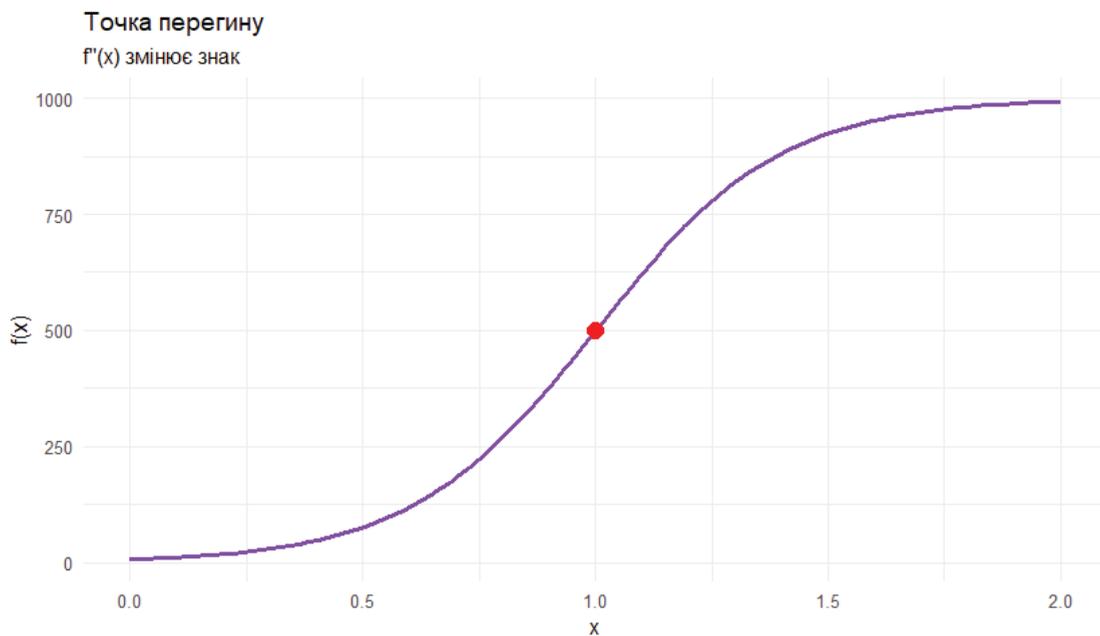


Опуклість (опуклість догори). Якщо  
 $f''(x) < 0$ ,  
то графік є опуклим. У цьому випадку швидкість росту зменшується (процес сповільнюється).



### Точка перегину

Точкою перегину називається точка, в якій графік функції змінює свій характер з вгнутості на опуклість (або навпаки).



**Необхідна умова:** у точці перегину  $x_0$  друга похідна або дорівнює нулю,  
 $f''(x_0) = 0$ ,

або не існує.

**Достатня умова:**

при переході через точку  $x_0$  друга похідна  $f''(x)$  повинна змінювати свій знак.

Друга похідна  $N''(t)$  показує прискорення процесу зміни кількості завербованих осіб  $N(t)$ .

Якщо  $N''(t) > 0$ , швидкість зростає. Це фаза активного прискорення, коли кожен новий учасник залучає ще більше людей через внутрішні канали.

Якщо  $N''(t) < 0$ , швидкість зростає сповільнено або падає. Це фаза гальмування, що свідчить про насичення аудиторії.

Це момент, коли  $N''(t) = 0$ . У цій точці швидкість вербовки  $\frac{dN}{dt}$  досягає свого максимуму (піку). Після цієї точки процес починає сповільнюватися.

Повернемось до закону еволюції

$$\frac{dN(t)}{dt} = (\alpha_1 + \alpha_2 N)(N_0 - N)$$

Для знаходження  $N''(t)$  диференціюємо обидві частини по  $t$ , застосовуючи правило диференціювання складеної функції:

$$N''(t) = \frac{d}{dN} \left( \frac{dN}{dt} \right) \cdot \frac{dN}{dt}$$

Обчислимо похідну від швидкості по  $N$ :

$$f(N) = (\alpha_1 + \alpha_2 N)(N_0 - N) = \alpha_1 N_0 - \alpha_1 N + \alpha_2 N_0 N - \alpha_2 N^2$$

$$f'(N) = -\alpha_1 + \alpha_2 N_0 - 2\alpha_2 N$$

Запишемо вираз для другої похідної:

$$N''(t) = (\alpha_2 N_0 - \alpha_1 - 2\alpha_2 N) \cdot \frac{dN}{dt}$$

Знайдемо точку перегину ( $N''(t) = 0$ ):

Оскільки швидкість  $\frac{dN}{dt} \neq 0$  під час процесу, прирівнюємо до нуля перший множник:

$$\begin{aligned}\alpha_2 N_0 - \alpha_1 - 2\alpha_2 N &= 0 \\ 2\alpha_2 N &= \alpha_2 N_0 - \alpha_1;\end{aligned}$$

звідки

$$N_{peak} = \frac{\alpha_2 N_0 - \alpha_1}{2\alpha_2}$$

Момент часу  $t_{peak}$ , у який швидкість вербовки досягає свого максимуму (тобто коли  $N(t) = N_{peak}$ ), визначається параметрами зовнішнього та внутрішнього впливу.

Щоб знайти цей час, ми беремо наш аналітичний розв'язок для логістичної моделі та прирівнюємо його до значення в точці перегину

$$N_{peak} = \frac{N_0}{2} - \frac{\alpha_1}{2\alpha_2}.$$

Після алгебраїчних перетворень отримуємо:

$$t_{peak} = \frac{1}{\alpha_1 + \alpha_2 N_0} \ln \left( \frac{\alpha_2 N_0}{\alpha_1} \right)$$

$N_{peak} = N(t_{peak})$  – чисельність членів, завербованих у момент  $t_{peak}$ . Це точка максимуму «ажіотажу» (вона існує не для будь-яких значень параметрів моделі).

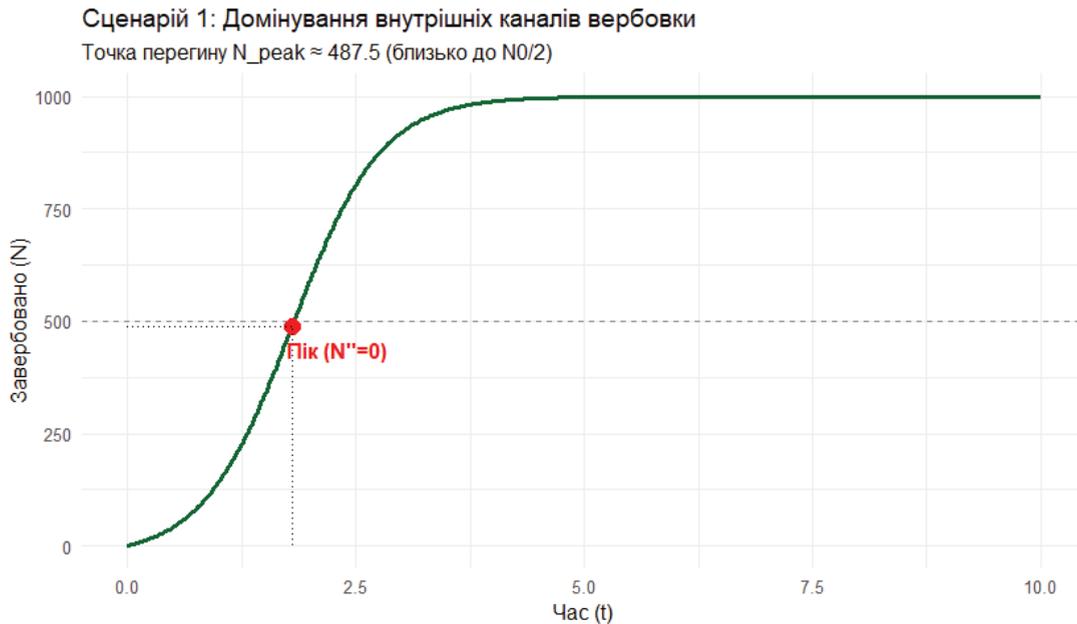
Отримана формула пікової точки  $N_{peak}$  дозволяє провести градацію режимів інформаційної операції.

1) Режим домінування внутрішніх каналів ( $\alpha_2 > \alpha_1$ )  
Якщо зовнішній вплив мінімальний, точка перегину знаходиться близько до середини:

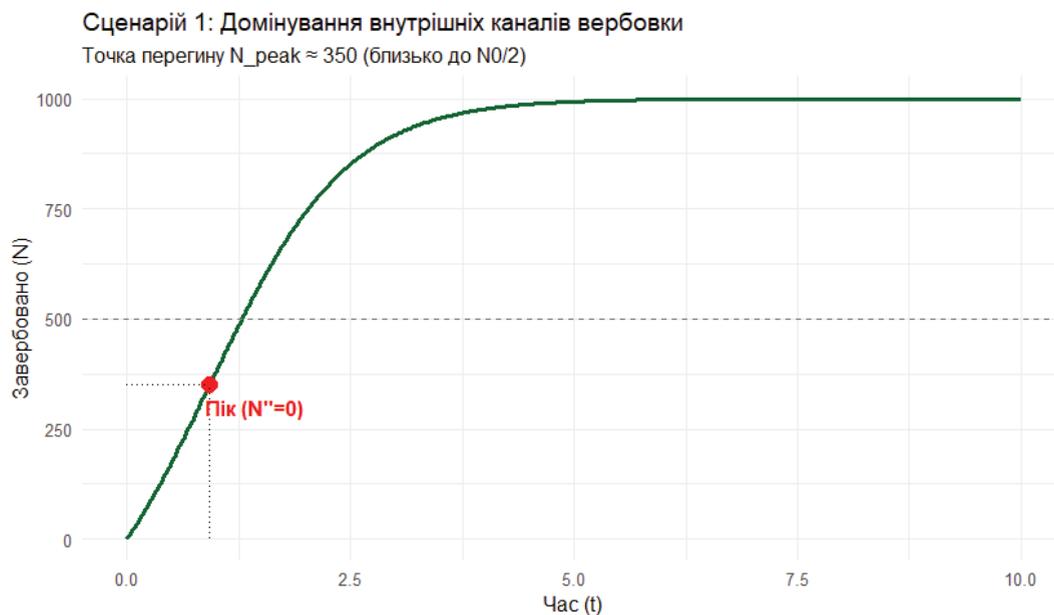
$$N_{peak} \approx \frac{N_0}{2}.$$

Процес розвивається як класична "сарафанна" вербовка, де основна маса людей залучається через горизонтальні зв'язки всередині системи.

```
alpha1 <- 0.05 # Мінімальний зовнішній вплив  
alpha2 <- 0.002 # Потужні внутрішні канали вербовки ( $\times N_0 = 1000$ )
```



```
alpha1 <- 0.3 # Мінімальний зовнішній вплив  
alpha2 <- 0.001 # Потужні внутрішні канали вербовки ( $\times N_0 = 1000$ )
```



2) Режим переваги зовнішніх джерел ( $\alpha_1 > \alpha_2$ ). Зі збільшенням інтенсивності зовнішніх каналів ( $\alpha_1$ ), точка піку зміщується вліво:

$$N_{reak} < \frac{N_0}{2}.$$

Це означає, що максимальна швидкість вербовки досягається раніше, ніж буде охоплено половину аудиторії. Зовнішній ресурс "перетискає" природний темп адаптації.

Критичний сценарій "Миттєвого охоплення" ( $\alpha_1 \geq \alpha_2 N_0$ ). Якщо зовнішній канал настільки потужний, що  $\alpha_1 \geq \alpha_2 N_0$ , то

$$N_{reak} \leq 0.$$

Це означає:

у системи немає фази прискорення ( $N''(t)$  завжди від'ємне);

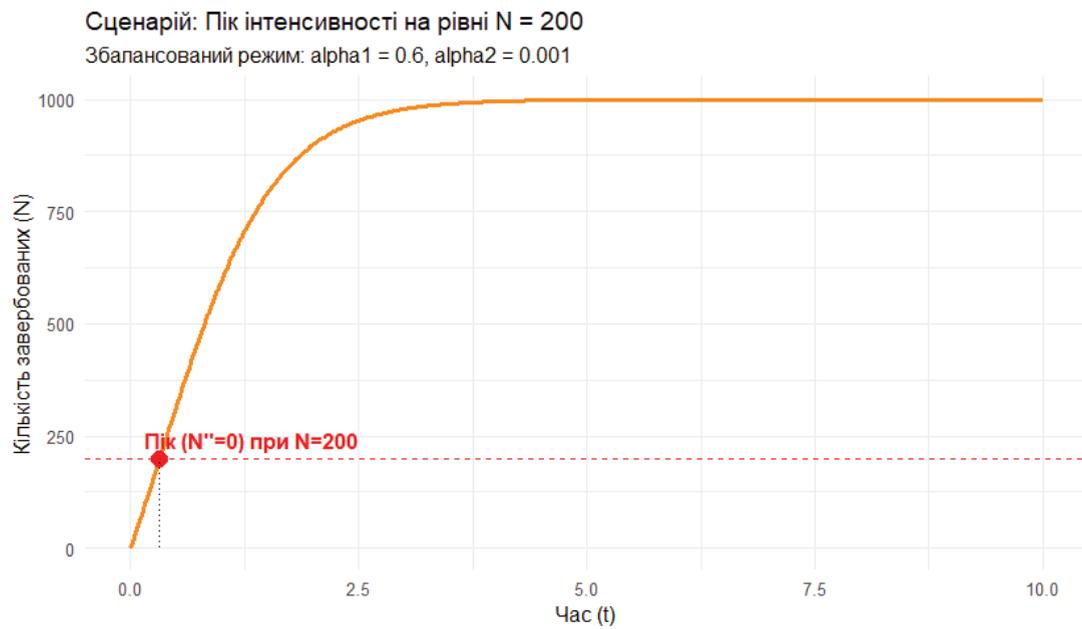
швидкість вербовки є максимальною в перший момент  $t = 0$  і надалі лише падає.

Це характерно для масованих інформаційних ударів, де внутрішні канали вже не відіграють ролі у "розігріві" аудиторії.

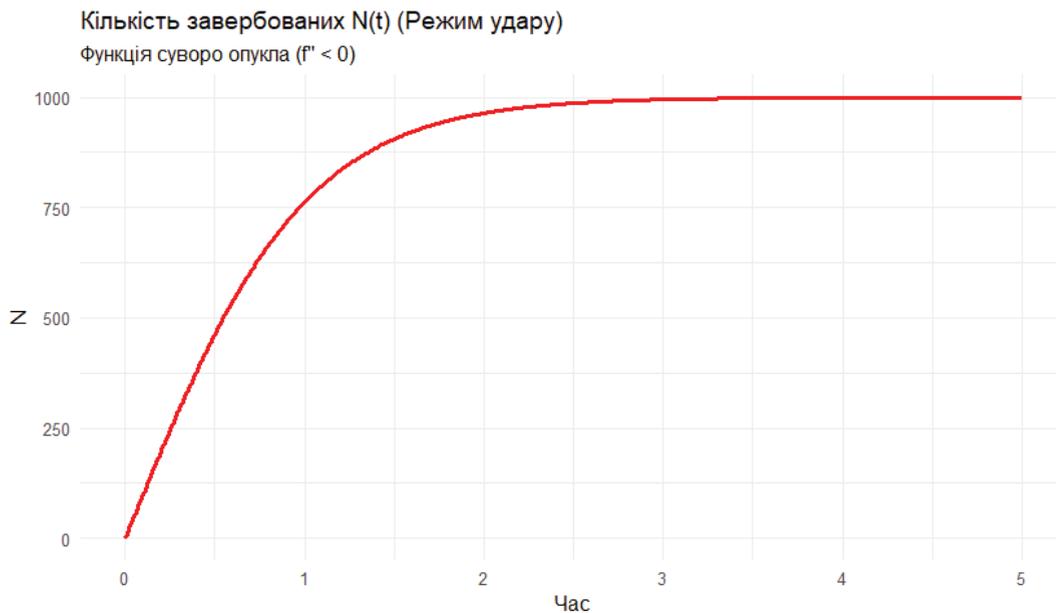
`alpha1 <- 0.97` # Значно посилений зовнішній вплив

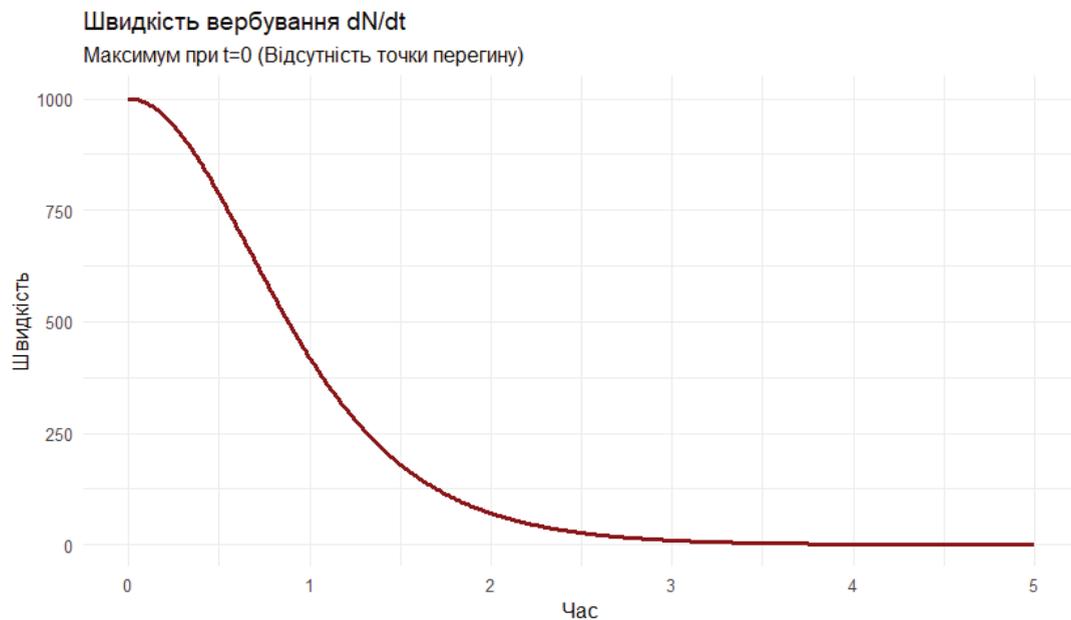
`alpha2 <- 0.001` # Помірна внутрішня вербовка





3) Інтенсивність зовнішніх джерел інформаційної хвилі дорівнює інтенсивності внутрішніх джерел: тут  $N_{\text{peak}} = 0$ , це означає, що швидкість вербування в початковий момент часу  $t = 0$  максимально можлива, а при  $t > 0$  вона зменшується. Це режим «бліцкригу». Ви отримуєте максимально швидкий результат на початку, але швидкість залучення нових людей стрімко падає з кожною секундою, оскільки ресурс аудиторії вичерпується надзвичайно швидко.





### Про ефективність витрат на поширення інформаційного впливу

Для аналізу введемо два типи ефективності, які трактуються як «прибуток» на одиницю «витрат» (де витратами вважаються параметри зовнішнього каналу  $\alpha_1$ ):

**Поточна (інтегральна) ефективність  $E(t)$**

$$E(t) = \frac{N(t)}{\alpha_1 t}.$$

Показує загальну віддачу від операції до моменту часу  $t$ .

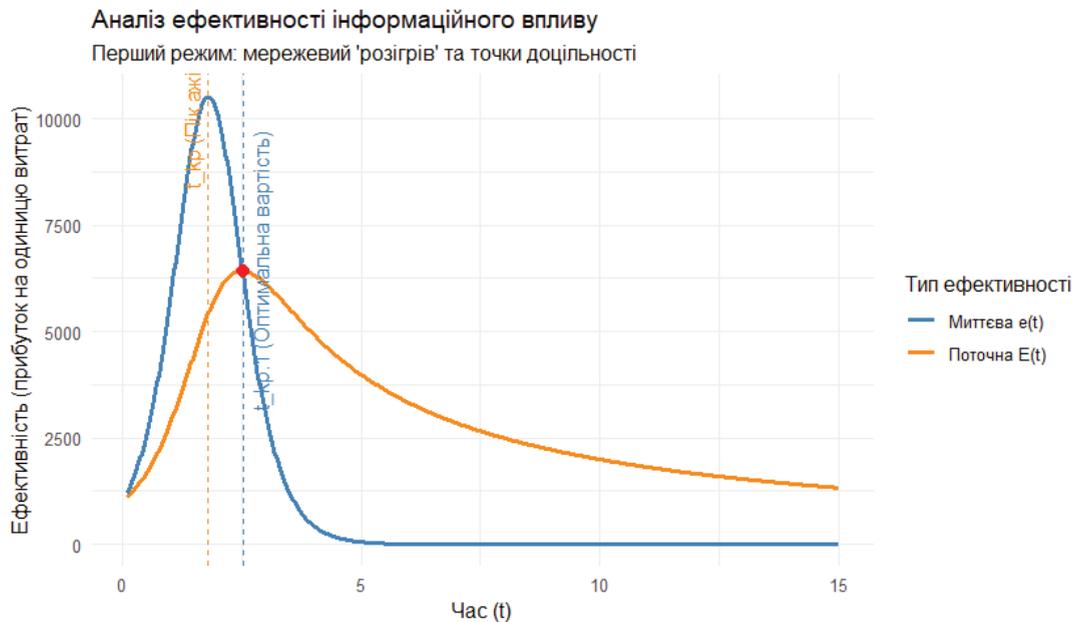
**Миттєва ефективність  $e(t)$**

$$e(t) = \frac{dN/dt}{\alpha_1}.$$

Розглянемо перший режим, коли потенціал внутрішньої вербовки (мережевого ефекту) переважає потужність зовнішнього джерела.

Система проявляє найбільш цікаву нелінійну динаміку: Ефект «розігріву». Процес не просто відбувається, він самоприскорюється завдяки внутрішнім зв'язкам.

Саме тут виникає момент, коли внутрішні канали починають працювати настільки ефективно, що зовнішні витрати ( $\alpha_1$ ) дають максимальну віддачу



У точці  $t_{\text{peak}}$  швидкість залучення нових людей є максимальною. Це момент максимального тріумфу інформаційної хвилі.

У точці  $t_{\text{peak1}}$  вартість залучення одного адепта (з урахуванням усього часу операції) стає оптимальною. У цей момент значення  $E(t)$  та  $e(t)$  рівні.

Після  $t_{\text{peak1}}$  проведення подальших заходів стає дедалі менш доцільним, оскільки витрачаються ті самі ресурси ( $\alpha_1$ ), а кількість нових учасників зростає все повільніше.

Таким чином, як видно з аналізу навіть дуже простої математичної моделі поширення інформаційного впливу, учасники даного процесу залежно від поставлених цілей, наявних ресурсів, існуючих часових рамок та поведінкових характеристик мають у своєму розпорядженні певні можливості для його організації в потрібному режимі.

## 5.2. Модель інформаційної протидії

Нехай тепер, маємо соціальну спільноту чисельністю  $N_0$ , потенційно схильну до впливу не одного, а двох інформаційних потоків, що не збігаються між собою за змістом (в екстремальному випадку інформація типу 1 ( $I_1$ ) і типу 2 ( $I_2$ ) – діаметрально протилежні одна одній). Нехай у момент часу  $t_0 = 0$  два джерела різної інформації одночасно починають її транслявати, внаслідок чого обидва інформаційні потоки розповсюджуються серед спільноти.

Оскільки хвилі  $I_1$  і  $I_2$  не тотожні одна одній, то даний процес природно розглядати як інформаційне протиборство (конкуренцію, суперництво). Побудуємо математичну модель цієї боротьби, з якої можна було б отримати динаміку її розвитку в часі (тобто залежні від часу  $t$  величини  $N_1(t)$  і  $N_2(t)$  числа

«адептів», які прийняли інформацію, що розповсюджується джерелами «1» і «2»), а також визначити її кінцевий результат – «переможця» або «переможеного».

Переможцем вважається той, хто до моменту повного охоплення досліджуваної спільноти обома видами інформації зумів розповсюдити свою інформацію серед більшого, ніж суперник, числа членів спільноти, тобто величини більшої, ніж  $N_0/2$  – половини від загальної чисельності спільноти  $N_0$ .

Основні модельні припущення (цілком аналогічні припущенням, зробленим при побудові моделі поширення інформації

Кожен із потоків  $I_1$  і  $I_2$  розповсюджується серед спільноти двома інформаційними каналами:

а) перший із них – «зовнішній» по відношенню до спільноти. Інтенсивність розповсюдження інформації цим каналом для  $I_1$  характеризується параметром  $\alpha_1 > 0$ , а для  $I_2$  – параметром  $\alpha_2 > 0$ , які вважаються незалежними від часу;

б) другий, «внутрішній» канал – міжособистісне спілкування членів соціальної спільноти (його інтенсивність для  $I_1$  характеризується параметром  $\beta_1 > 0$ , а для  $I_2$  – параметром  $\beta_2 > 0$ ).

В результаті такого спілкування вже завербовані ідеєю «1» адепти (їхнє число дорівнює величині  $N_1(t)$ ), впливаючи на ще не завербованих членів (їхня чисельність дорівнює величині  $(N_0 - N_1(t) - N_2(t))$ ), вносять свій «особистий» внесок у процес вербування.

Точно так само адепти ідеї «2» (їхнє число дорівнює величині  $N_2(t)$ ), впливаючи на ще не завербованих членів (їхня чисельність також дорівнює величині  $(N_0 - N_1(t) - N_2(t))$ ), вносять свій «особистий» внесок у процес вербування.

Швидкості зміни числа адептів  $N_1(t)$ ,  $N_2(t)$  (тобто число завербованих за одиницю часу  $I_1$  і  $I_2$  складаються з:

а) швидкостей зовнішнього вербування (вони пропорційні добуткам інтенсивностей  $\alpha_1$  і  $\alpha_2$  на число діючих членів  $(N_0 - N_1(t) - N_2(t))$ , тобто величинам

$$\alpha_1(N_0 - N_1(t) - N_2(t)) \text{ та}$$

$$\alpha_2(N_0 - N_1(t) - N_2(t)) \text{ відповідно для } I_1 \text{ і } I_2;$$

б) швидкостей внутрішнього вербування (вони пропорційні добуткам інтенсивностей  $\beta_1$  і  $\beta_2$  на число діючих адептів  $N_1(t)$  і  $N_2(t)$  та на число незавербованих  $(N_0 - N_1(t) - N_2(t))$ , тобто величинам  $\beta_1 N_1(t)(N_0 - N_1(t) - N_2(t))$  та  $\beta_2 N_2(t)(N_0 - N_1(t) - N_2(t))$  відповідно для  $I_1$  і  $I_2$ ).

Чисельність ще не завербованих членів спільноти дорівнює величині  $N_0$  за винятком числа членів, що прийняли вже не один, а обидва види інформації (тобто за вирахуванням суми  $N_1(t) + N_2(t)$ ).

Як і раніше, параметри  $\alpha_1, \beta_1$  та  $\alpha_2, \beta_2$  характеризують не тільки інтенсивність інформаційного впливу, а й схильність до його сприйняття. Таким чином, ще незавербована до моменту  $t$  частина спільноти (її гіпотетичний «середній» представник, спочатку нейтральний по відношенню як до  $I_1$ , так і до  $I_2$ ) приймає інформацію тим швидше, чим більші величини  $\alpha_1, \beta_1, \alpha_2, \beta_2$ .

При цьому навіть якщо вплив  $I_1$  наперед сильніший за вплив  $I_2$  ( $\alpha_1 > \alpha_2, \beta_1 > \beta_2$ ), частина членів спільноти все одно приймає  $I_2$  (немає повної монополії одного виду інформації по відношенню до іншого).

Підсумовуючи припущення 1 і 2, приходимо до моделі (5.1):

$$\frac{dN_1}{dt} = (\alpha_1 + \beta_1 N_1(t))(N_0 - N_1(t) - N_2(t)), N_1(t_0 = 0) = N_1(0) > 0, \quad (5.1)$$

$$\frac{dN_2}{dt} = (\alpha_2 + \beta_2 N_2(t))(N_0 - N_1(t) - N_2(t)), N_2(t_0 = 0) = N_2(0) > 0.$$

Система звичайних диференціальних рівнянь (5.1) (автономна динамічна система 2-го порядку) слугує початковою моделлю досліджуваного процесу. З неї при відомих параметрах  $N_0, \alpha_1, \beta_1, \alpha_2, \beta_2$  і початкових значеннях чисельностей  $N_1(0)$  і  $N_2(0)$  аналітично або чисельно знаходяться всі шукані характеристики. Будемо називати її моделлю інформаційної війни.

Ділячи друге рівняння (5.1) на перше, отримуємо:

$$\frac{dN_2}{dN_1} = \frac{\alpha_2 + \beta_2 N_2}{\alpha_1 + \beta_1 N_1}; \quad (5.2)$$

$$\frac{dN_2}{\alpha_2 + \beta_2 N_2} = \frac{dN_1}{\alpha_1 + \beta_1 N_1};$$

$$\int \frac{1}{\alpha_2 + \beta_2 N_2} dN_2 = \int \frac{1}{\alpha_1 + \beta_1 N_1} dN_1;$$

Для інтегрування використовуємо правило:

$$\int \frac{1}{ax + b} dx = \frac{1}{a} \ln |ax + b|.$$

Отримуємо:

$$\frac{1}{\beta_2} \ln(\alpha_2 + \beta_2 N_2) = \frac{1}{\beta_1} \ln(\alpha_1 + \beta_1 N_1) + C^*;$$

$$\ln(\alpha_2 + \beta_2 N_2) = \frac{\beta_2}{\beta_1} \ln(\alpha_1 + \beta_1 N_1) + \beta_2 C^*.$$

За властивістю логарифмів ( $k \ln A = \ln A^k$ ) коефіцієнт  $\frac{\beta_2}{\beta_1}$  заносимо в степінь аргументу:

$$\ln(\alpha_2 + \beta_2 N_2) = \ln((\alpha_1 + \beta_1 N_1)^{\beta_2/\beta_1}) + C^{**}$$

(тут  $C^{**} = \beta_2 C^*$ ).

Представимо константу  $C^{**}$  як  $\ln C$ . Тоді:

$$\ln(\alpha_2 + \beta_2 N_2) = \ln(C \cdot (\alpha_1 + \beta_1 N_1)^{\beta_2/\beta_1})$$

Прибираємо логарифми з обох боків:

$$\alpha_2 + \beta_2 N_2 = C \cdot (\alpha_1 + \beta_1 N_1)^{\beta_2/\beta_1}$$

$$\beta_2 N_2 = C \cdot (\alpha_1 + \beta_1 N_1)^{\beta_2/\beta_1} - \alpha_2. \quad (5.3)$$

Підставимо значення в початковий момент часу  $t = 0$ , коли чисельності дорівнюють  $N_1(0)$  та  $N_2(0)$ :

$$\alpha_2 + \beta_2 N_2(0) = C \cdot (\alpha_1 + \beta_1 N_1(0))^{\beta_2/\beta_1}$$

Звідси:

$$C = \frac{\alpha_2 + \beta_2 N_2(0)}{(\alpha_1 + \beta_1 N_1(0))^{\beta_2/\beta_1}}$$

Якщо ж ми починаємо «з нуля» ( $N_1(0) = 0$ ,  $N_2(0) = 0$ ), формула має вигляд:

$$C_0 = \frac{\alpha_2}{\alpha_1^{\beta_2/\beta_1}}.$$

а для розв'язку справедливо:

$$\frac{\beta_2}{\alpha_2} N_2(t) = \left(1 + \frac{\beta_1}{\alpha_1} N_1(t)\right)^{\beta_2/\beta_1} - 1. \quad (5.4)$$

Не розглядаючи часову динаміку процесу детально, вивчимо питання про «переможця» та «переможеного», коли до моменту повного охоплення досліджуваної спільноти обома видами інформації «переможець» зумів поширити свою інформацію серед більшого, ніж суперник, числа членів спільноти.

Необхідна і достатня умова «перемоги»  $I_1$  над  $I_2$  (тобто виконання нерівності

$$N_1^f > \frac{N_0}{2} > N_2^f,$$

де  $N_1^f, N_2^f$  – значення чисельностей  $N_1(t), N_2(t)$  у момент закінчення процесу, коли

$$N_1^f + N_2^f = N_0,$$

впливає з (2.4) яке має єдиний розв'язок  $N_1^f < N_0$ .

У випадку

$$N_1(0) = N_2(0) = 0$$

воно задається наступною нерівністю:

$$\frac{\beta_1}{\ln \left( 1 + \frac{\beta_1 N_0}{2\alpha_1} \right)} > \frac{\beta_2}{\ln \left( 1 + \frac{\beta_2 N_0}{2\alpha_2} \right)}. \quad (5.5)$$

Відповідно, нерівність протилежного знака означає «перемогу»  $I_2$  (тобто «поразку»  $I_1$ ), а рівність – «нічию».

Зауважимо, що при ненульових початкових чисельностях adeptів  $I_1$  та  $I_2$  умова перемоги  $I_1$  над  $I_2$  визначається також і значеннями

$$N_1(0), N_2(0).$$

Обмежимося далі випадком при  $N_1(0) = N_2(0) = 0$ ,

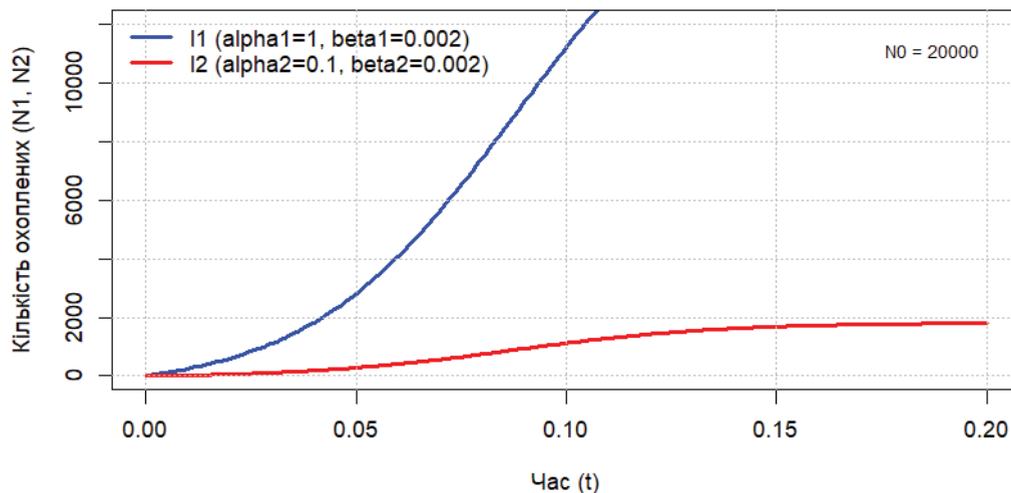
Проаналізуємо умови перемоги (5.5), розглянувши чотири сценарії.

1) Нехай

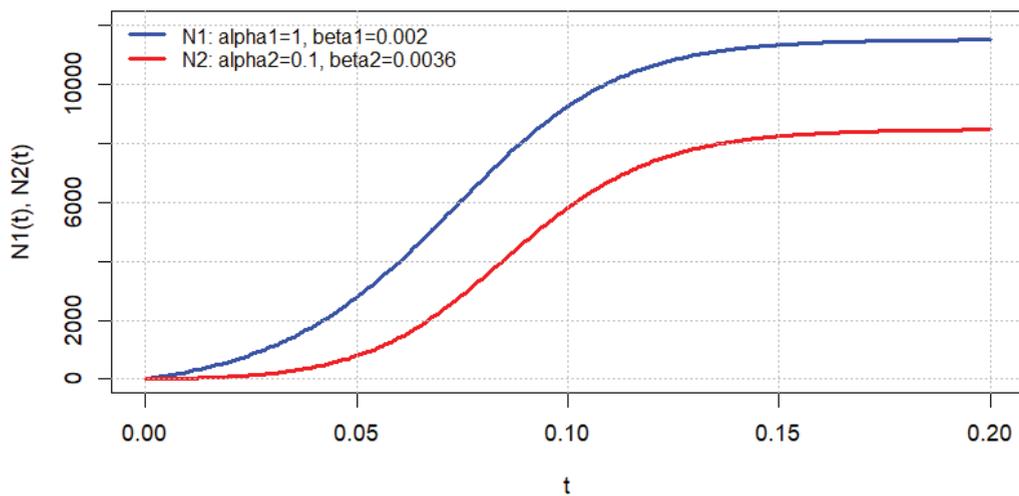
$$\beta_2 = \beta_1,$$

тобто інтенсивності внутрішніх (міжособистісних) каналів поширення  $I_1$  та  $I_2$  однакові. Тоді (2.5) перетворюється на очевидну нерівність  $\alpha_1 > \alpha_2$  для перемоги  $I_1$ ,  $\alpha_1 < \alpha_2$  для перемоги  $I_2$ . У цій ситуації перемагає той, хто забезпечує сильніший зовнішній інформаційний вплив.

**Моделювання при рівних  $\beta_1 = \beta_2 = 0.002$**



Моделювання сценарію №1 (N0 = 20000 )

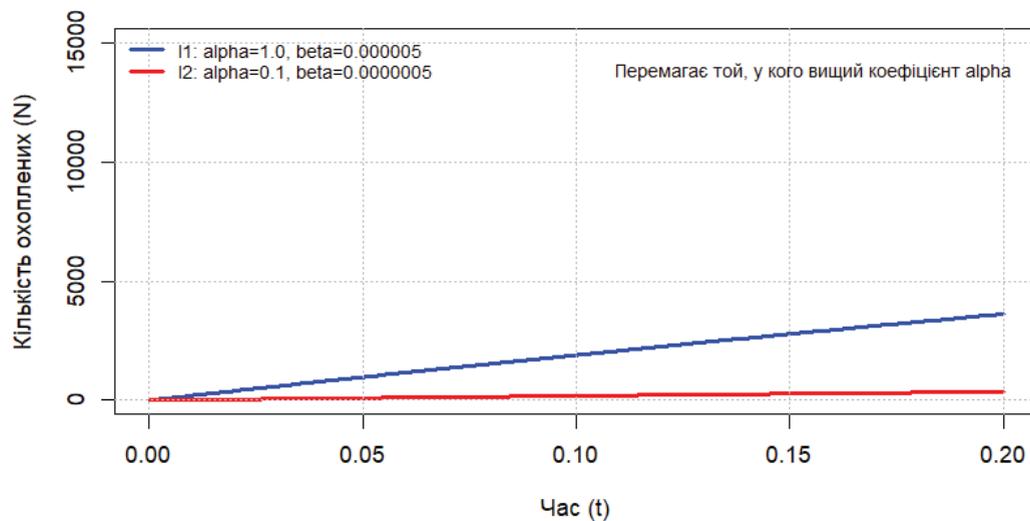


2) Нехай

$$\frac{\beta_1 N_0}{2\alpha_1} \ll 1, \frac{\beta_2 N_0}{2\alpha_2} \ll 1,$$

тобто для обох видів інформації інтенсивність поширення внутрішніми каналами суттєво менша за зовнішню. Тоді перемагає той, у кого більший коефіцієнт зовнішнього впливу  $\alpha$ .

Сценарій 2: Домінування зовнішнього впливу (beta << alpha)



3. Нехай тепер

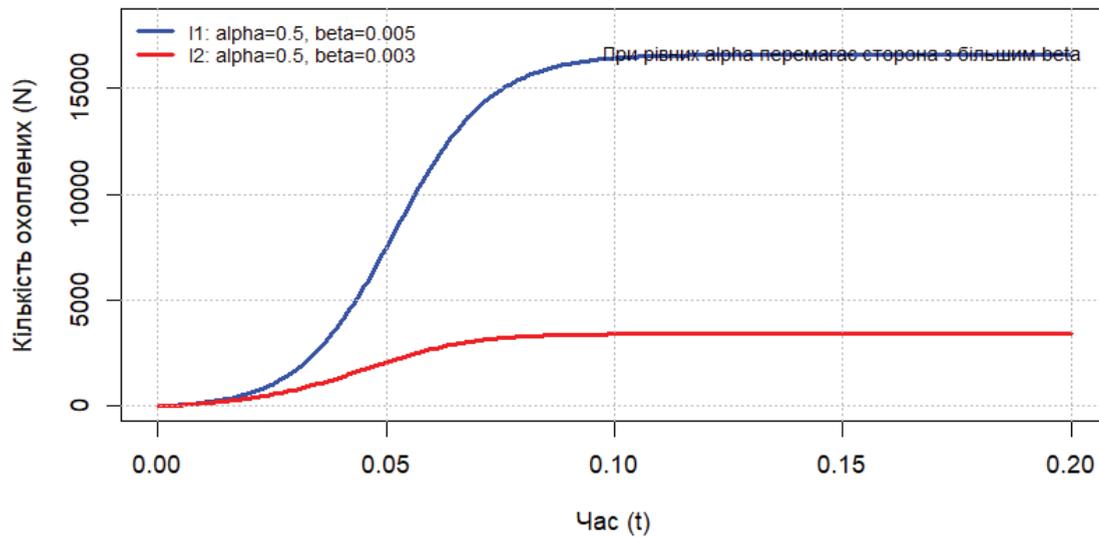
$$\alpha_1 = \alpha_2,$$

тобто зовнішні канали однакові. У цій ситуації перемагає той, чия інформація інтенсивніше поширюється внутрішніми (міжособистісними) каналами:

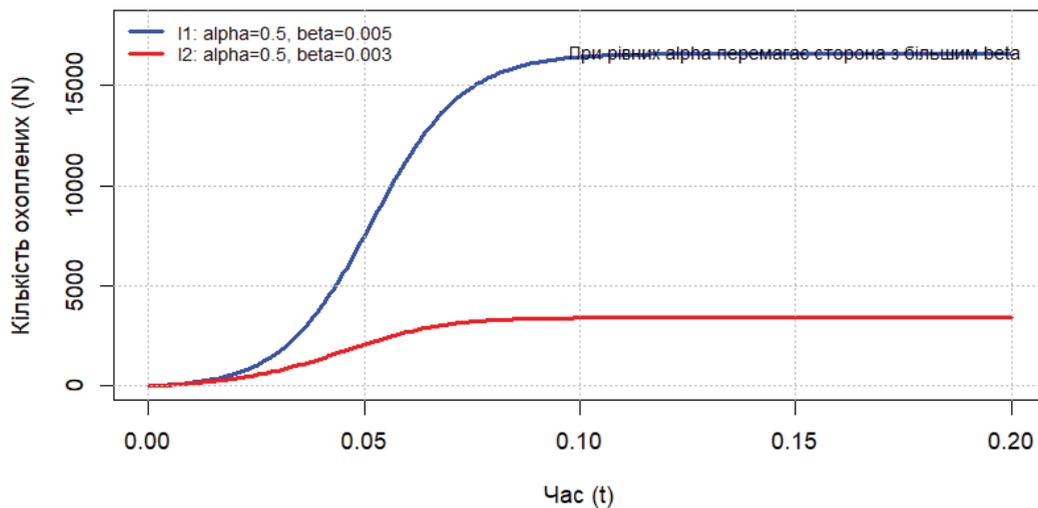
при  $\beta_1 > \beta_2$  - перемагає  $I_1$ ,

при  $\beta_2 > \beta_1$  - перемагає  $I_2$ .

### Сценарій 3: Домінування внутрішнього каналу (beta)



### Сценарій 3: Домінування внутрішнього каналу (beta)



4) Набагато різноманітніші результати мають місце в загальному випадку, коли результат визначається всіма характеристиками інтенсивності поширення  $I_1$  та  $I_2$  – як зовнішніми, так і внутрішніми.

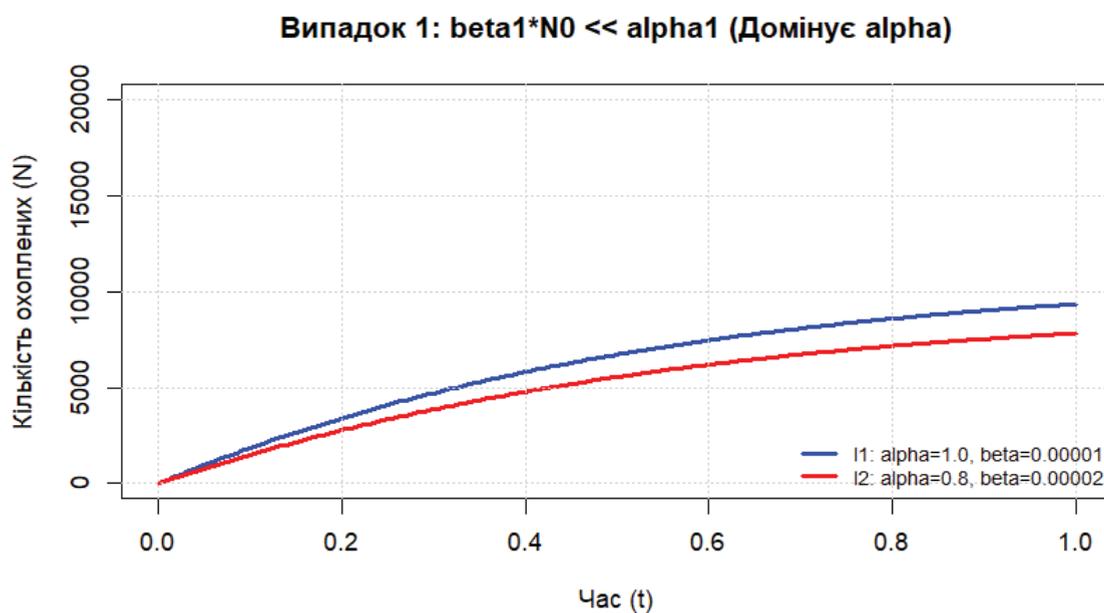
Нехай

$$\beta_2 = 2\beta_1,$$

тобто інтенсивність внутрішнього каналу для  $I_2$  у два рази більша, ніж для  $I_1$ . Умова «перемоги» інформації  $I_1$  набуває вигляду:

$$\alpha_2 < \frac{4\alpha_1^2}{4\alpha_1 + \beta_1 N_0}.$$

При малих значеннях внутрішнього поширення  $\beta_1 N_0 \ll \alpha_1$  умова спрощується до  $\alpha_1 > \alpha_2$ .



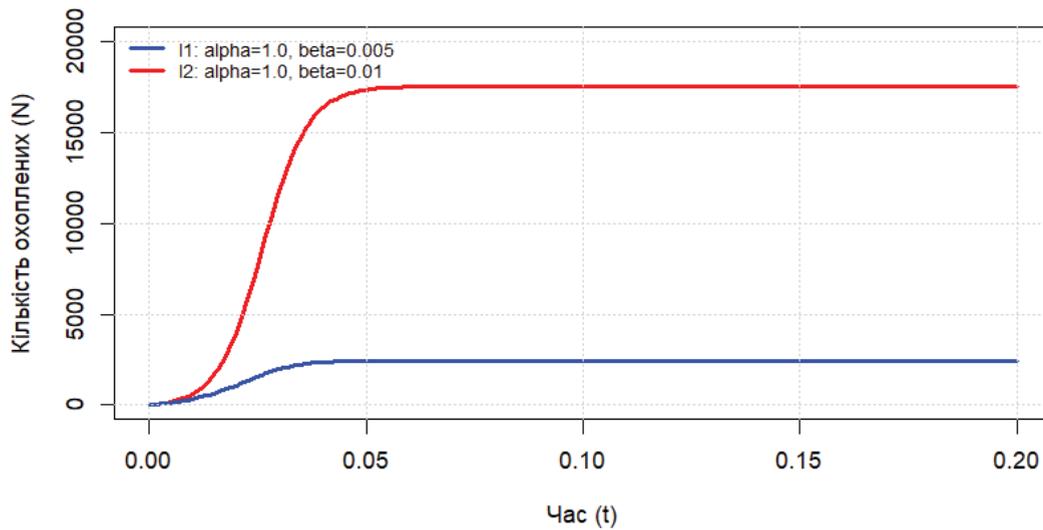
Незважаючи на перевагу другої сторони у внутрішніх каналах, перша перемагає за рахунок зовнішнього впливу.

У протилежному випадку, коли

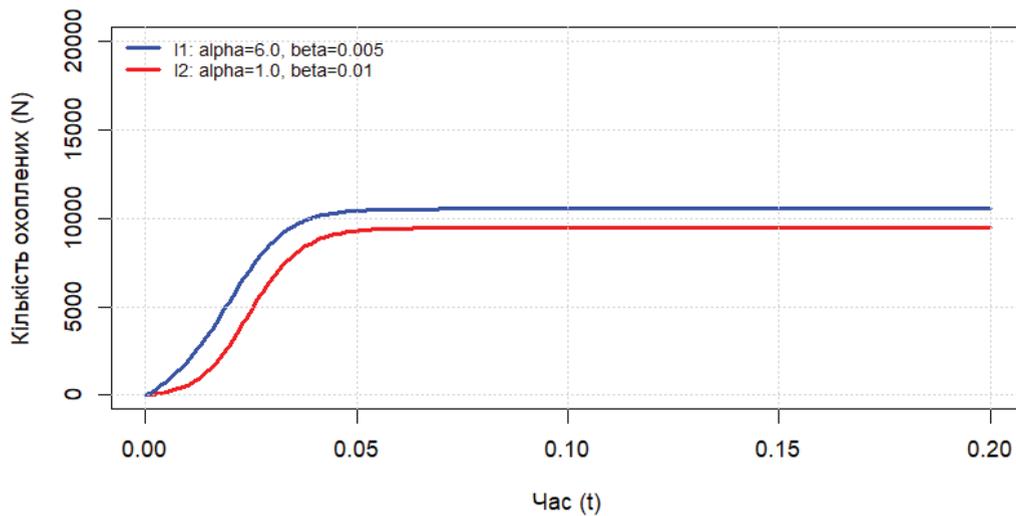
$$\beta_1 N_0 \gg \alpha_1,$$

перша сторона повинна мати значну перевагу у зовнішніх каналах для досягнення перемоги.

**Випадок 2:  $\beta_1 \cdot N_0 \gg \alpha_1$  (Домінує мережевий ефект)**



**Випадок 2:  $\beta_1 \cdot N_0 \gg \alpha_1$  (Домінує мережевий ефект)**



У сторін, що ведуть інформаційну війну, існують (зокрема залежно «ємності» соціальної спільноти) вельми різноманітні можливості досягати перемоги, системно змінюючи характеристики процесу.

Визначальне значення має правильний вибір співвідношення між потужностями зовнішніх («вертикальних») та внутрішніх («горизонтальних», мережевих) інформаційних каналів.

### 5.3. Багатовимірні моделі інформаційної боротьби

Модель інформаційної війни узагальнюється на ситуацію протиборства не двох, а більшої кількості видів інформації  $I_i, i = 1, 2, \dots, M$ , де  $M > 2$ . У цьому випадку модель є системою  $M$  нелінійних звичайних диференціальних рівнянь:

$$\frac{dN_i}{dt} = (\alpha_i + \beta_i N_i(t)) \left( N_0 - \sum_{j=1}^M N_j(t) \right),$$

$$N_i(t_0 = 0) = N_i(0) \geq 0, 1 \leq i \leq M$$

де  $N_i(t)$ ,  $N_i(t_0 = 0)$  - поточні та початкові значення кількості adeptів інформації  $I_i$ , а  $\alpha_i, \beta_i$  - інтенсивності її поширення по зовнішньому та внутрішньому каналах відповідно.

Розгорнутий вигляд системи (для  $M$  видів інформації):

$$\begin{cases} \frac{dN_1}{dt} = (\alpha_1 + \beta_1 N_1)(N_0 - N_1 - N_2 - \dots - N_M) \\ \frac{dN_2}{dt} = (\alpha_2 + \beta_2 N_2)(N_0 - N_1 - N_2 - \dots - N_M) \\ \dots \\ \frac{dN_M}{dt} = (\alpha_M + \beta_M N_M)(N_0 - N_1 - N_2 - \dots - N_M) \end{cases}$$

Матричний вигляд системи:

$$\frac{d\mathbf{N}(t)}{dt} = \mathbf{C}(\mathbf{N}(t))$$

Вектор станів

$$\mathbf{N}(t) = \begin{pmatrix} N_1(t) \\ N_2(t) \\ \vdots \\ N_M(t) \end{pmatrix}$$

Праву частину Можна представити у вигляді

$$\mathbf{C}(\mathbf{N}) = \mathbf{A}(\mathbf{N})\mathbf{N} + \mathbf{F}$$

Повний запис матриці  $\mathbf{C}(\mathbf{N})$

$$\begin{pmatrix} -\alpha_1 + \beta_1 N_0 - \beta_1 N_1(t) & -\alpha_1 - \beta_1 N_1(t) & \dots & -\alpha_1 - \beta_1 N_1(t) \\ -\alpha_2 - \beta_2 N_2(t) & -\alpha_2 + \beta_2 N_0 - \beta_2 N_2(t) & \dots & -\alpha_2 - \beta_2 N_2(t) \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & \dots & -\alpha_M + \beta_M N_0 - \beta_M N_M(t) \end{pmatrix} \times \begin{pmatrix} N_1 \\ N_2 \\ \vdots \\ N_M \end{pmatrix} + \begin{pmatrix} \alpha_1 N_0 \\ \alpha_2 N_0 \\ \vdots \\ \alpha_M N_0 \end{pmatrix}$$

У багатовимірному випадку система має загальний розв'язок (записаний у рекурентному вигляді):

$$\begin{aligned} \beta_{i+1} N_{i+1}(t) &= C_i (\alpha_i + \beta_i N_i(t))^{\beta_{i+1}/\beta_i} - \alpha_{i+1}, 1 \leq i \leq M-1 \\ C_i &= [\alpha_{i+1} + \beta_{i+1} N_{i+1}(0)] / [\alpha_i + \beta_i N_i(0)]^{\beta_{i+1}/\beta_i}, 1 \leq i \leq M-1 \end{aligned}$$

З моделі розв'язків визначається динаміка інформаційної боротьби між  $M$  учасниками та її підсумки. Зокрема, легко знаходиться достатня умова завідомої перемоги одного з конкурентів над усіма іншими разом узятими. Це умова досягнення в момент  $t_f$  чисельності членів спільноти, що сприйняли інформацію переможця, величини більшої за  $N_0/2$  (тобто отримання «партією» переможця «парламентської більшості»).

Введемо «функцію перемоги»:

$$V_i = V_i(\alpha_i, \beta_i, N_i(0), N_0), 1 \leq i \leq M$$

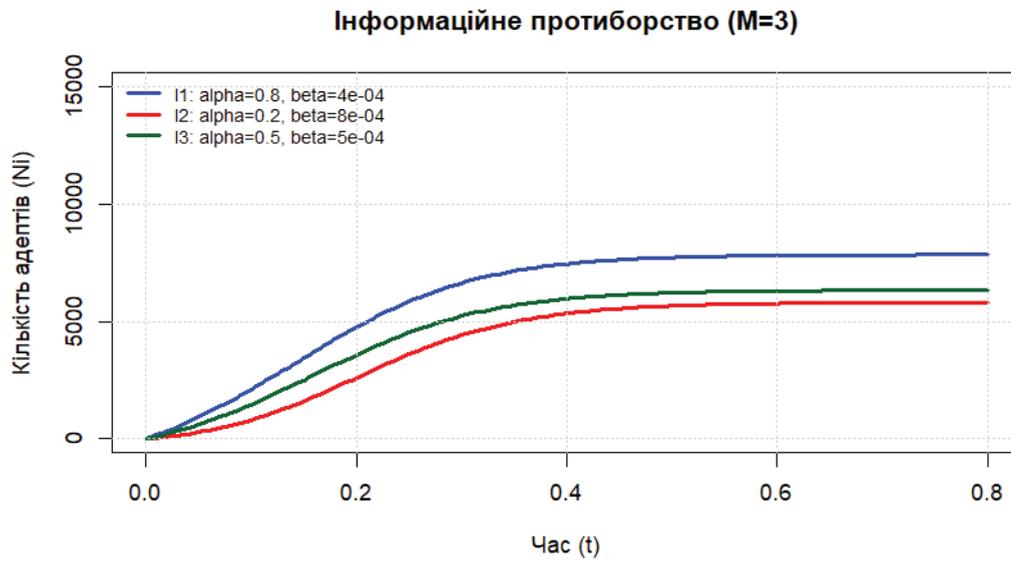
за правилом:

$$V_i = \beta_i \ln \left[ \left( 1 + \frac{\beta_i N_0}{2\alpha_i} \right) / \left( 1 + \frac{\beta_i N_i(0)}{\alpha_i} \right) \right]^{-1}$$

Тоді переможцем є учасник з номером  $i_0$  ( $1 \leq i_0 \leq M$  з найбільшим серед усіх конкурентів значенням «функції перемоги», тобто такий, що має властивість:

$$V_{i_0} = \max_{1 \leq i \leq M} V_i \quad (V_{i_0} > V_i, 1 \leq i \leq M, i \neq i_0)$$

У загальному випадку, як і у випадку двох вимірів, переможець заздалегідь визначається з вхідних даних. При цьому величини чисельностей  $N_i(t)$  монотонно зростають з часом, а їх сума  $N_t = \sum_i^M N_j(t)$  прямує до  $N_0$  при  $t \rightarrow \infty$ .



У розглянутій моделі не враховується багато факторів – неоднорідність соціальної спільноти, ефекти «забування» інформації, можливість взаємозалежності величин  $\alpha_1$  і  $\alpha_2$ , можливість свідомої протидії інформаційній хвилі та інші фактори, що можуть бути важливими для досліджуваного процесу.

Подальші стохастичні моделі є більш природними.

**Контрольні питання:**

1. Запишіть та поясніть базову модель інформаційної боротьби.
2. Який фізичний зміст параметрів  $\alpha$  та  $\beta$ ?

## ЛЕКЦІЯ 6

### Асимптотичний аналіз стохастичних систем

#### 6.1. Схема усереднення

Еволюційні системи з випадковими збуреннями є широко розповсюдженими математичними моделями стохастичних систем, що описують реальні процеси в часі у різноманітних галузях – і в техніці, фізиці та суспільних науках.

Основна проблема, що виникає при дослідженні стохастичних систем, – це вивчення поведінки системи на зростаючих інтервалах часу, що призводить до ефекту усереднення випадкових збурень.

Ефективним методом асимптотичного аналізу стохастичних систем на зростаючих інтервалах часу є метод малого параметру з використанням теорії випадкових еволюцій в банаховому просторі тест-функцій.

В даній лекції пропонується огляд основних алгоритмів асимптотичного аналізу стохастичних систем: усереднення, дифузійна апроксимація, а також неklasичні схеми – Пуассона та Леві для еволюційних систем з марковськими випадковими збуреннями, а також застосування до моделі, представленої у лекції 5.

Еволюційна система з випадковими збуреннями у схемі серій з малим параметром серії  $\varepsilon \rightarrow 0$  ( $\varepsilon > 0$ ) задається розв'язком еволюційного рівняння

$$\frac{du^\varepsilon(t)}{dt} = C\left(u^\varepsilon(t), x\left(\frac{t}{\varepsilon}\right)\right), t \geq 0, \quad (6.1)$$

функція швидкості  $C(u; x)$ ,  $u \in R^d$ ,  $x \in E$ , є вектор-функцією двох аргументів: аргументу еволюційної системи  $u$ , що приймає значення в просторі  $R^d$ ,  $d \geq 1$ ; та аргументу випадкових збурень  $x$ , що приймає значення у фазовому просторі  $(X, \mathcal{F})$  який є повним метричним простором  $X$  з системою вимірних множин  $\mathcal{F}$ . Вважаємо, що окремі точки  $x$  простору  $X$  належать  $\mathcal{F}$ .

Щодо параметру  $\varepsilon \rightarrow 0$ . Вираз  $x\left(\frac{t}{\varepsilon}\right)$  означає, що випадкове збурення  $x$  змінюється дуже швидко порівняно з основною системою  $u$ . Це створює ефект «високочастотного шуму»

Припускається існування розв'язку сім'ї рівнянь

$$\frac{du_x(t)}{dt} = C(u_x(t), x), x \in X. \quad (6.2)$$

Важливою властивістю рівнянь (6.2) є півгрупова властивість розв'язків:

$$u_x(t_1 + t_2; u) = u_x(t_2; u_x(t_1; u)). \quad (6.3)$$

(майбутнє системи визначається лише її поточним станом, а не тим, як вона до нього прийшла.)

Тут  $u_x(t; u)$  є розв'язком рівняння (6.2) з початковою умовою  $u_x(0; u) = u$ .

Півгрупова властивість (3) породжує півгрупу операторів в банаховому просторі тест-функцій  $\mathbb{B}(R^d)$ :

$$C_t(x)\varphi(u) = \varphi(u_x(t; u)), t \geq 0, \varphi(u) \in \mathbb{B}(R^d). \quad (6.4)$$

(банахів простір вводиться для загальності математичного обґрунтування. Оператори діють не на числа, а на функції, які визначені на станах системи. Простір таких функцій (тест-функцій) якраз і є Банаховим простором.)

Півгрупа операторів (6.4) визначається генератором

$$C(x)\varphi(u) := \lim_{t \rightarrow 0} \frac{1}{t} [C_t(x)\varphi(u) - \varphi(u)],$$

який задається формулою

$$C(x)\varphi(u) = C(u; x)\varphi'(u).$$

Тут за означенням

$$C(u; x)\varphi'(u) = \sum_{k=1}^d C_k(u; x) \frac{\partial \varphi(u)}{\partial u_k}.$$

Це як миттєва швидкість: ми дивимося, як змінюється оператор за нескінченно малий час, тому  $C$  часто називають функцією швидкості.

Тепер залишається визначитися з характеристикою випадкових збурень. Будемо розглядати, в основному, випадкові збурення, що визначаються марковським стрибковим процесом  $x(t)$ ,  $t \geq 0$ , у стандартному фазовому просторі  $(X, \mathcal{F})$ , що задається породжуючим оператором (генератором) у банаховому просторі  $\mathbb{B}(E)$  тест-функцій  $\varphi(x) \in \mathbb{B}(E)$ , які приймають числові значення:

$$Q\varphi(x) = q(x) \int_E P(x, dy) [\varphi(y) - \varphi(x)].$$

Тут функція  $q(x)$ ,  $x \in X$ , задає інтенсивність моментів стрибків процесу  $x(t)$ ,  $t \geq 0$ , що мають показниковий розподіл; стохастичне ядро  $P(x, dy)$ ,  $x \in X$ ,  $dy \in \mathcal{F}$ , розподіл ймовірностей переходу процесу  $x(t)$  зі стану  $x$  в множину станів  $dy$ .

Стохастичне ядро  $P(x, B)$ ,  $x \in X$ ,  $B \in \mathcal{F}$  визначає рівномірно ергодичний вбудований марковський ланцюг  $x_n = x(\tau_n)$ ,  $n \geq 0$ , зі стаціонарним розподілом  $\rho(B)$ ,  $B \in \mathcal{F}$ . Стаціонарний розподіл  $\pi(B)$ ,  $B \in \mathcal{F}$  марковських процесів  $x(t)$ ,  $t \geq 0$  визначається відношенням

$$\pi(dx)q(x) = q\rho(dx), q = \int_X \pi(dx)q(x)$$

**Принцип усереднення.** Якщо марковський процес збурень  $x(t)$ ,  $t \geq 0$ , є рівномірно ергодичним зі стаціонарним розподілом  $\pi(B)$ ,  $B \in \mathcal{E}$ , то при виконанні сформульованих вище умов має місце слабка збіжність

$$u^\varepsilon(t) \xrightarrow{d} u^0(t), \varepsilon \rightarrow 0.$$

Гранична еволюція  $u^0(t)$ ,  $t \geq 0$ , визначається розв'язком детермінованого еволюційного рівняння

$$\frac{du^0(t)}{dt} = \hat{C}(u^0(t)), t \geq 0.$$

Усереднена функція швидкості визначається співвідношенням

$$\hat{C}(u) = \int_E \pi(dx) C(u; x).$$

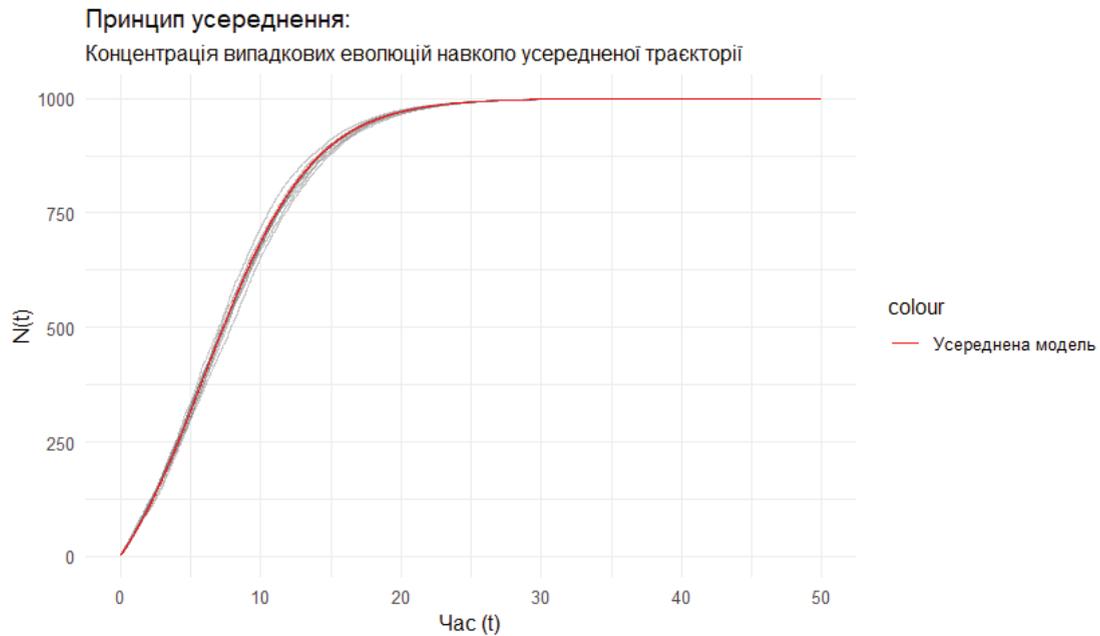
**Приклад.** Модель поширення інформації.

У базовій моделі параметри  $\alpha_1$  (зовнішній вплив) та  $\alpha_2$  (внутрішнє вербування) є сталими. Щоб застосувати схему усереднення, ми припустимо, що ці параметри тепер залежать від «швидкого» випадкового середовища  $x(t/\varepsilon)$  яке моделює хаотичні зміни в інформаційному просторі (наприклад, зміна алгоритмів соцмереж або настроїв аудиторії).

$$\frac{dN(t)}{dt} = (\alpha_1(x) + \alpha_2(x)N(t))(N_0 - N(t)), N(0) = 0, t > 0.$$

Повільна змінна:  $N(t)$  – число «адептів».

Швидка змінна:  $x(t/\varepsilon)$  – марковський процес, що перемикає стани середовища.



## 6.2. Дифузійна апроксимація

### Флуктуації динамічної системи за умови рівноваги.

Умова рівноваги(балансу) для динамічної системи

$$\frac{du^\varepsilon(t)}{dt} = C\left(u^\varepsilon(t), x\left(\frac{t}{\varepsilon}\right)\right), t \geq 0,$$

що визначається функцією швидкості  $C(u; x)$ ,  $u \in R^d$ ,  $x \in X$ , означає, що усереднена швидкість дорівнює тотожно нулю:

$$\hat{C}(u) = \int_E \pi(dx)C(u; x) \equiv 0.$$

За умови балансу усереднена система зберігає сталі значення на необмеженому інтервалі часу. У зв'язку з цим виникає проблема вивчення флуктуацій динамічної системи, яку в умовах балансу слід розглядати з таким нормуванням

$$\frac{du^\varepsilon(t)}{dt} = \varepsilon^{-1}C\left(u^\varepsilon(t), x\left(\frac{t}{\varepsilon^2}\right)\right). \quad (6.5)$$

Нормування динамічної системи (6.5) можна пояснити тим, що флуктуації за умови балансу стануть помітними, якщо збільшити швидкість системи (множником  $\varepsilon^{-1}$ ); крім того, усереднення флуктуацій проявиться, якщо збільшити інтервал часу спостережень (множником  $\varepsilon^{-2}$ ).

**Дифузійна апроксимація динамічної системи за умови балансу.** Якщо марковський процес збурень  $x(t)$ ,  $t \geq 0$ , є рівномірно ергодичним зі стаціонарним розподілом  $\pi(B)$ ,  $B \in X$ , то за умови балансу (1) має місце слабка збіжність

$$u^\varepsilon(t) \xrightarrow{d} \zeta^0(t), \varepsilon \rightarrow 0.$$

Граничний дифузійний процес  $\zeta^0(t)$ ,  $t \geq 0$ , визначається генератором

$$\mathbf{L}\varphi(u) = b(u)\varphi'(u) + \frac{1}{2}B(u)\varphi''(u);$$

де параметр зсуву –

$$b(u) = \int_X \pi(dx)b(u; x), b(u; x) = C(u; x)R_0 C'_u(u; x); \quad (5)$$

дифузійний коефіцієнт –

$$B(u) = \int_X \pi(dx)B(u; x); B(u; x) = C(u; x)R_0 C(u; x); \quad (6)$$

і потенціал ергодичного марковського процесу є

$$R_0 = - \int_0^{\infty} [\exp \mathbf{Q}t - \Pi] dt, \Pi \varphi(x) = \int_E \pi(dx) \varphi(x). (7)$$

**Зауваження.** Граничний дифузійний процес  $\zeta^0(t)$ ,  $t \geq 0$ , задається також розв'язком стохастичного диференціального рівняння

$$d\zeta^0(t) = b(\zeta^0(t))dt + \sigma(\zeta^0(t))dW(t). (8)$$

Тут  $\sigma^2(u) = B(u)$ ,  $W(t)$  – стандартний вінерівський процес.

#### Пояснення

Якщо попередня схема усереднення показувала нам "головний тренд" (детерміновану криву), то ця схема описує випадкові коливання навколо цього тренду за умови рівноваги. Дифузійна апроксимація застосовується тоді, коли "середня швидкість" системи дорівнює нулю, тобто, в середньому система нікуди не рухається, і її стан зберігає сталі значення на великих інтервалах часу. Але насправді вона не стоїть на місці – вона коливається через швидкі випадкові збурення.

Щоб ці коливання стали помітними, застосовується спеціальне нормування, і це дозволяє нам побачити "мікроскопічні" рухи, які раніше зникали при звичайному усередненні.

За таких умов дограничний випадковий процес  $u^\varepsilon$  слабо збігається до граничного процесу  $\zeta^0$ . Цей процес вже не є детермінованим, а стохастичним, і визначається генератором  $\mathbf{L}$

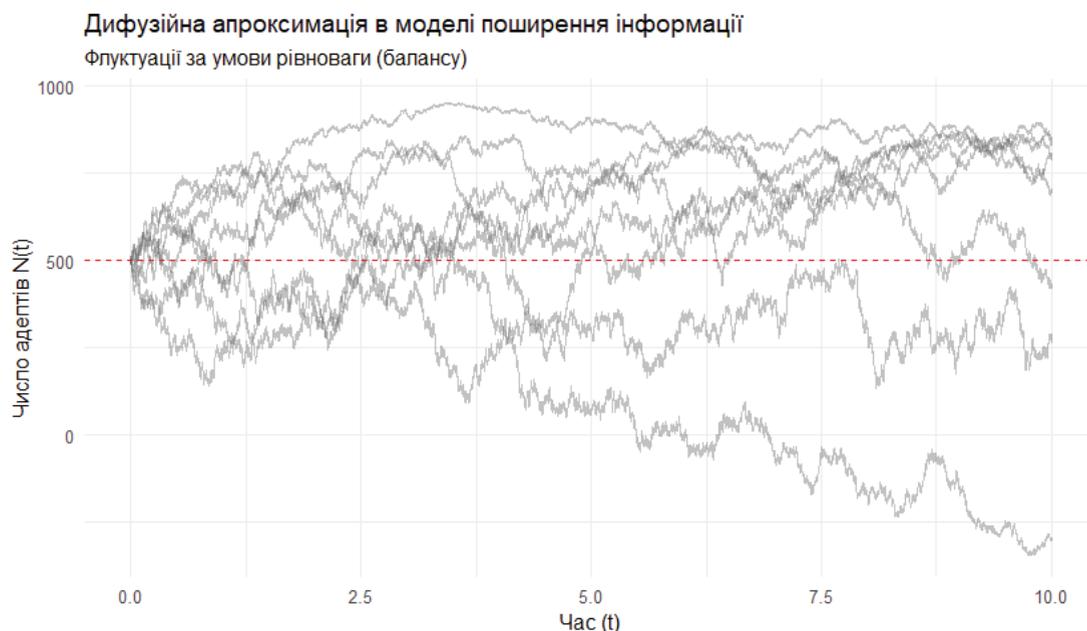
#### Приклад. Модель поширення інформації.

Якщо у схемі усереднення ми бачили гладку S-подібну криву, то дифузійна апроксимація – це модель того, як ця крива "тремтить". Це критично важливо для прогнозування ризиків: наприклад, коли середня швидкість поширення новини нульова, але через випадкові флуктуації може статися раптовий "вибух" популярності.

Умова балансу – припускаємо, що середня швидкість поширення інформації  $\hat{C}(N) \equiv 0$ . Це означає, що зовнішній вплив і внутрішні контакти в середньому компенсують одне одного (наприклад, новина застаріває так само швидко, як і поширюється).

Флуктуації – оскільки середній тренд нульовий, на перший план виходять випадкові відхилення. Замість гладкої лінії ми побачимо «тремтіння» навколо початкового стану.

Масштабування - вводим множники  $\varepsilon^{-1}$  для амплітуди коливань та  $\varepsilon^{-2}$  для часу, щоб зробити ці флуктуації помітними на макрорівні.



### 6.3. Апроксимації Леві

Далі збурення системи визначаються імпульсним процесом за неklasичною схемою апроксимації Леві і приділяємо особливу увагу асимптотичній поведінці генератора для цієї системи.

Стохастична еволюційна система в ергодичному марковському середовищі визначається стохастичним диференціальним рівнянням

$$du^\varepsilon(t) = C(u^\varepsilon, x(t/\varepsilon^2))dt + d\eta^\varepsilon(t), \quad u^\varepsilon(t) \in \mathbb{R} \quad (6.6)$$

де рівномірно ергодичний марковський процес  $x(t)$ ,  $t \geq 0$  у стандартному фазовому просторі  $(X, \mathcal{F})$ , визначається генератором

$$\mathbf{Q}\varphi(x) = q(x) \int_X P(x, dy)[\varphi(y) - \varphi(x)]$$

у банаховому просторі дійснозначних обмежених функцій  $\varphi(x)$  з супремум нормою  $\|\varphi\| = \max_{x \in X} |\varphi(x)|$ .

Стохастичне ядро  $P(x, B)$ ,  $x \in X$ ,  $B \in \mathcal{F}$  визначає рівномірно ергодичний вкладений марковський ланцюг  $x_n = x(\tau_n)$ ,  $n \geq 0$ , зі стаціонарним розподілом  $\rho(B)$ ,  $B \in \mathcal{F}$ . Стаціонарний розподіл  $\pi(B)$ ,  $B \in \mathcal{F}$  марковських процесів  $x(t)$ ,  $t \geq 0$  визначається відношенням

$$\pi(dx)q(x) = q\rho(dx), \quad q = \int_X \pi(dx)q(x)$$

Нехай  $R_0$  – потенційний оператор генератора  $\mathbf{Q}$ , що визначається рівністю

$$R_0 = \Pi - (\Pi + \mathbf{Q})^{-1},$$

де  $\Pi\varphi(x) = \int_X \pi(dy)\varphi(y)1(x)$  проектор нулів генератором  $\mathbf{Q}$  на підпростір  $N_{\mathbf{Q}} = \varphi: \mathbf{Q}\varphi = 0$ .

Імпульсний процес збурення  $\eta^\varepsilon(t), t \geq 0$

$$\eta^\varepsilon(t) = \int_0^t \eta^\varepsilon(ds, x(s/\varepsilon^2)) \quad (2)$$

де множина процесів з незалежними приростами  $\eta_\varepsilon(t, x), t \geq 0, x \in X$ , визначається генераторами

$$\Gamma^\varepsilon(x)\varphi(w) = \varepsilon^2 \int_R (\varphi(w+v) - \varphi(w))\Gamma^\varepsilon(dv, x), x \in X \quad (3)$$

і задовольняє властивості апроксимації Леві [2]

**L1.** Апроксимація середніх

$$\int_R v\Gamma^\varepsilon(dv, x) = \varepsilon a_1(x) + \varepsilon^2(a_2(x) + \theta_a(x)), \theta_a(x) \rightarrow 0, \varepsilon \rightarrow 0,$$

і

$$\int_R v^2\Gamma^\varepsilon(dv, x) = \varepsilon(b(x) + \theta_b(x)), \theta_b(x) \rightarrow 0, \varepsilon \rightarrow 0,$$

**L2.** Умова на функцію розподілу

$$\int_R g(v)\Gamma^\varepsilon(dv, x) = \varepsilon^2(\Gamma_g(x) + \theta_g(x)), \theta_g(x) \rightarrow 0, \varepsilon \rightarrow 0,$$

для усіх  $g(v) \in C^2(\mathbb{R})$  (простір реальних обмежених функцій, таких як  $g(v)/|v|^2 \rightarrow 0, |v| \rightarrow 0$ ), де міра  $\Gamma_g(x)$  обмежена для всіх  $g(v) \in C^2(\mathbb{R})$  і визначається співвідношенням (функції з простору  $C^2(\mathbb{R})$  розділяють міру):

$$\Gamma_g(x) = \int_R g(v)\Gamma_0(dv, x), g(v) \in C^3(\mathbb{R});$$

**L3.** Рівномірна квадратична інтегрованість

$$\sup_{|v|>c} \lim_{c \rightarrow \infty} \int v^2\Gamma_0(dv, x) = 0;$$

**Приклад** випадкової величини  $\xi$ , що задовольняє умови апроксимації Леві

$\xi$	$b$	$\varepsilon a_1 + \varepsilon^2 a_2$
$\mathbf{P}(\xi)$	$\varepsilon^2 p$	$1 - \varepsilon^2 p$

$$\mathbf{P}\{\xi = b\} = \varepsilon^2 p,$$

$$\mathbf{P}\{\xi = \varepsilon a_1 + \varepsilon^2 a_2\} = 1 - \varepsilon^2 p.$$

$$\begin{aligned} \mathbf{E}[\xi] &= b\varepsilon^2 p + (\varepsilon a_1 + \varepsilon^2 a_2)(1 - \varepsilon^2 p) = \\ &= \varepsilon a_1 + \varepsilon^2(a_2 + bp) + o(\varepsilon^2), \\ \mathbf{E}[\xi^2] &= b^2\varepsilon^2 p + (\varepsilon a_1 + \varepsilon^2 a_2)^2(1 - \varepsilon^2 p) = \\ &= \varepsilon^2(a_1^2 + b^2 p) + o(\varepsilon^2). \end{aligned}$$

Нехай виконана умова балансу

$$\hat{a}_1 := \int_X \pi(dx) a_1(x) = 0, \quad (6.7)$$

розглянемо асимптотичні властивості процесу збурень.

**Теорема 2.1.** Якщо задовольняються умова балансу (6.7) та умови L1 – L3, то для імпульсного процесу збурення має місце слабка збіжність

$$\eta^\varepsilon(t) \xrightarrow{d} \eta^0(t), \varepsilon \rightarrow 0$$

Граничний процес  $\eta^0(t)$  визначається генератором

$$\Gamma\varphi(w) = \hat{a}_2\varphi'(w) + \frac{1}{2}\sigma^2\varphi''(w) + \int_R [\varphi(w+v) - \varphi(w)]\hat{\Gamma}_0(dv)$$

де

$$\hat{a} = \int_X \pi(dx)(a_2(x) - a_0(x)),$$

$$\sigma^2 = \int_X \pi(dx)(b(x) - b_0(x)) + 2 \int_X \pi(dx)a_1(x)R_0a_1(x),$$

$$a_0(x) = \int_X v\Gamma_0(dv, x),$$

$$b_0(x) = \int_R v^2\Gamma_0(dv, x),$$

$$\hat{\Gamma}_0(v) = \int_R \pi(dx)\Gamma_0(v, x),$$

і це процес Леві з трьома компонентами – детермінованим зсувом, дифузійною складовою та стрибковою частиною Пуассона.

**Теорема 2.** При виконанні умов балансу (4) та умов L1 – L3 має місце слабка збіжність

$$u^\varepsilon(t) \xrightarrow{d} \hat{u}(t), \varepsilon \rightarrow 0.$$

Граничний процес  $\hat{u}(t)$  визначається генератором

$$\mathbf{L}\varphi(w) = \hat{C}(u)\varphi'(w) + \Gamma\varphi(w),$$

де  $\hat{C}(u) = \int_X \pi(dx)C(u, x)$ .

**Коментар 1.** Слабка збіжність процесів  $u^\varepsilon(t) \xrightarrow{d} \hat{u}(t)$ ,  $\varepsilon \rightarrow 0$ , впливає зі збіжності відповідних генераторів, коли зберігається компактність граничної множини процесів  $u^\varepsilon(t)$ .

**Коментар 2.** Граничний процес  $\hat{u}(t)$  може бути задано стохастичним диференціальним рівнянням

$$d\hat{u}_d(t) = [\hat{C}(\hat{u}(t)) + \hat{a}^2]dt + \sigma dW(t) + \int_{\mathbb{R}} v\tilde{\nu}(dt, dv),$$

де  $E\tilde{\nu}(dt, dv) = dt\tilde{\Gamma}_0(dv)$ .

**Коментар 3.** Граничний процес  $\hat{u}(t)$  має три компоненти. Детермінований зсув визначається розв'язанням диференціального рівняння

$$d\hat{u}_d(t) = [\hat{C}(\hat{u}_d(t)) + \hat{a}_2]dt,$$

де додатковий член  $\hat{a}_2$  з'являється внаслідок накопичення з нормалізованим часом  $t/\varepsilon^2$ ,  $\varepsilon \rightarrow 0$  невеликих стрибків імпульсного процесу, які відбуваються з ймовірністю, близькою до одиниці. Друга, дифузійна складова, визначається параметром  $\sigma$  і виникає через накопичення з ростом нормованого часу  $t/\varepsilon^2$ ,  $\varepsilon \rightarrow 0$  невеликих стрибків порядку  $\varepsilon$ , які також відбуваються з ймовірністю близькою до одиниці.

Третя складова – це рідкісні великі стрибки, які відбуваються з майже нульовою ймовірністю і визначаються з точки зору усередненої міри стрибків  $\hat{\Gamma}_0(dv)$  генератором

$$\Gamma_j\varphi(w) = \int_{\mathbb{R}} [\varphi(w+v) - \varphi(w)]\hat{\Gamma}_0(dv).$$

**Приклад.** Модель поширення інформації в схемі Леві.

$$dN^\varepsilon(t) = C\left(N^\varepsilon, x\left(\frac{t}{\varepsilon^2}\right)\right)dt + d\eta^\varepsilon(t), \quad u^\varepsilon(t) \in \mathbb{R},$$

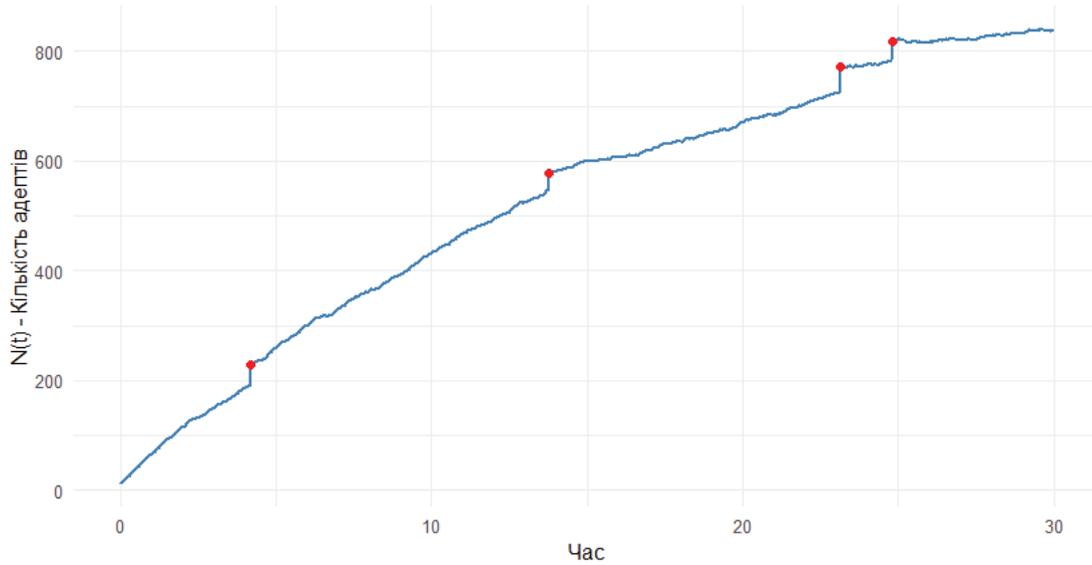
де

$$C(N, x) = (\alpha_1(x) + \alpha_2(x)N(t))(N_0 - N(t)), N(0) = 0$$

і виконуються усі припущення вище

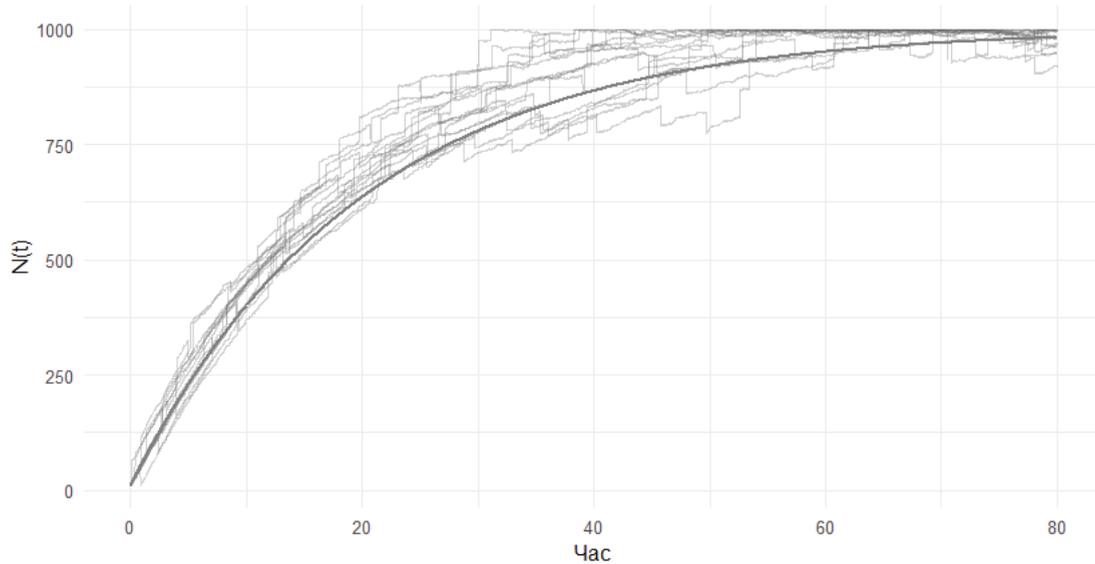
### Апроксимація Леві в інформаційній моделі

Комбінація зсуву (тренд), дифузії (шум) та стрибків (вірусні події)



### Апроксимація Леві:

Порівняння з детермінованою траєкторією без збурень



### Інтерпретації

1. Зсув (Drift): Це «стратегічне планування». Навіть якщо навколо хаос, система має внутрішній вектор розвитку (усереднений тренд).
2. Дифузія (Diffusion): Це «тактичний шум». Постійні дрібні коливання, які роблять прогноз імовірнісним, а не точним.
3. Стрибки (Jumps): Це «інформаційні події». Тепер ми визнаємо, що деякі події настільки потужні, що змінюють стан системи миттєво, не чекаючи накопичення дрібних змін.

**Приклад.** Модель інформаційної боротьби

$$dN^\varepsilon(t) = C(N^\varepsilon(t), x(t / \varepsilon^2))dt + d\eta^\varepsilon(t),$$

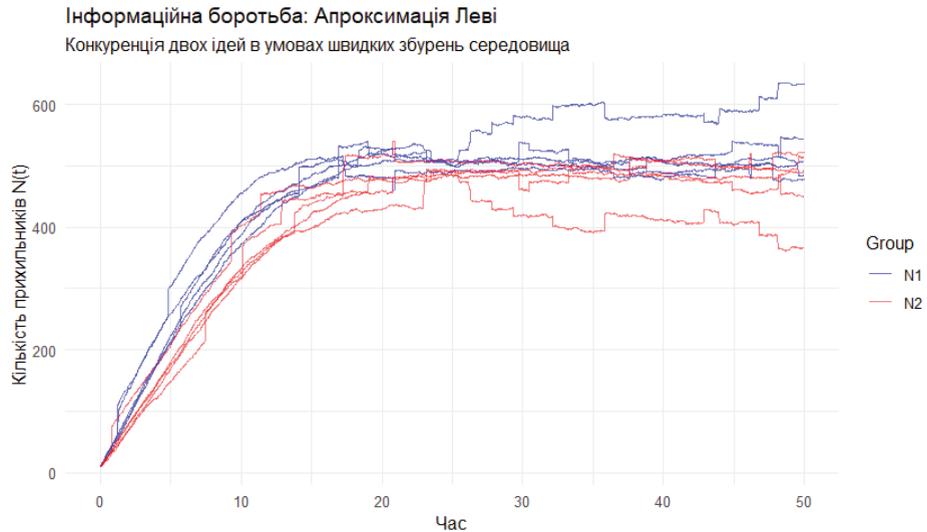
де

$$C(N^\varepsilon(t), x(t / \varepsilon^2)) = \begin{pmatrix} -\alpha_1(x) + \beta_1(x)N_0 - \beta_1(x)N_1^\varepsilon(t) & -\alpha_1(x) - \beta_1(x)N_1^\varepsilon(t) \\ -\alpha_2(x) - \beta_2(x)N_2^\varepsilon(t) & -\alpha_2(x) + \beta_2(x)N_0 - \beta_2(x)N_2^\varepsilon(t) \end{pmatrix} \begin{pmatrix} N_1^\varepsilon(t) \\ N_2^\varepsilon(t) \end{pmatrix} + \begin{pmatrix} \alpha_1 N_0 \\ \alpha_2 N_0 \end{pmatrix},$$

тут

–  $N^\varepsilon(t)$  є двовимірним вектором розв’язків, компонентами якого є кількості прихильників різних ідей;

–  $x(t / \varepsilon^2)$  є рівномірно ергодичним марковським процесом який моделює вплив середовища на інтенсивності розповсюдження інформації. Це означає, що в період між моментами відновлення такого марковського процесу значення  $\alpha_1(x), \alpha_2(x), \beta_1(x), \beta_2(x)$  є сталими, як у класичній моделі, натомість в моменти відновлення значення миттєво змінюються. Це моделює випадкові події які відбуваються незалежно від суспільства та суттєво впливають на погляди людей.



**Приклад** У випадку перемикаючого процесу Орнштейна-Уленбека, представлені вище формули можемо явно обрахувати інтегруванням по стаціонарній мірі (3) з урахуванням вигляду (4) потенціального оператора  $R_0$ .

Наприклад, якщо припустити, що функції  $b(x) := x^2$ ,  $b_0(x) = a_1(x) := x$ , то

$$\int_R x^2 \pi(dx) = \sigma_0^2, \int_R x \pi(dx) = 0, 2 \int_R x R_0 x \pi(dx) = \frac{2}{c} \sigma_0^2,$$

$$\text{а отже } \sigma^2 = \sigma_0^2 + \frac{2}{c} \sigma_0^2 = \frac{2+c}{c} \sigma_0^2.$$

Аналогічно можуть бути обраховані інші характеристики граничного процесу для різних перемикаючих марковських процесів з відомою стаціонарною мірою та потенціальним оператором.

**Інтерпретація.**

**Конкуренція за ресурси:** Оскільки загальна кількість людей  $N_0$  обмежена, зростання однієї групи безпосередньо або опосередковано впливає на можливості іншої.

**Роль випадкових подій:** Швидкі коливання середовища  $x(t)$  можуть призвести до того, що навіть за однакових початкових умов одна з ідей отримає перевагу через вдалий збіг зовнішніх факторів у критичний момент.

**Стійкість системи:** Використання схеми Леві дозволяє врахувати не лише плавні зміни, а й різкі стрибки (наприклад, "інформаційні бомби", які можуть кардинально змінити баланс сил у протистоянні).

**Контрольні питання:**

1. Сформулюйте теорему усереднення.
2. Що описує дифузійна апроксимація?
3. За яких умов застосовна апроксимація Леві?
4. Як схеми апроксимації обґрунтовують стохастичну модель?

## ЛЕКЦІЯ 7

### Оптимізація обчислень у Python

#### 7.1. Векторизація з NumPy

Векторизація – заміна явних циклів Python на операції з масивами NumPy. Масиви зберігаються в суміжних блоках пам'яті, а операції виконуються оптимізованим C-кодом. Прискорення може становити 10-100 разів.

Приклад: замість циклу `for` використовуйте `y = x**2`. Арифметичні операції, функції (`sin`, `exp`), порівняння – все застосовується поелементно до масивів. Broadcasting розширює масиви різних форм для поелементних операцій.

#### 7.2. Паралельна генерація траєкторій

Для методу Монте-Карло потрібно генерувати  $M$  незалежних траєкторій. Векторизований підхід: зберігати всі траєкторії в масиві  $(M, N+1)$  та оновлювати всі одночасно. `dW = np.sqrt(dt) * np.random.randn(M, N)` генерує всі прирости одразу.

Для СДР зі станозалежними коефіцієнтами внутрішній цикл по траєкторіях векторизується:  $X\_all[k+1] = X\_all[k] + a(X\_all[k])*dt + b(X\_all[k])*dW\_all[:,k]$ .

#### 7.3. Профілювання коду

Перед оптимізацією необхідно виявити вузькі місця. Модуль `cProfile` показує час виконання кожної функції. Команда `%timeit` в Jupyter вимірює час виконання копірки. `line_profiler` показує час кожного рядка.

#### 7.4. Компіляція з Numba

Numba – JIT-компілятор для Python, що перетворює функції на машинний код. Декоратор `@numba.jit(nopython=True)` компілює функцію при першому виклику. Прискорення може становити 10-1000 разів для циклів.

#### 7.5. Паралельні обчислення

`multiprocessing` запускає окремі процеси Python для обходу GIL. `joblib` надає зручну обгортку: `Parallel(n_jobs=-1)(delayed(func)(arg) for arg in args)`. Numba з `@numba.jit(parallel=True)` автоматично паралелізує цикли.

#### 7.6. Оптимізація пам'яті

Типи даних: `float64` займає 8 байт, `float32` – 4 байти. Для багатьох задач `float32` достатньо. Уникайте копіювання: операції на місці (`x += 1`), `views` замість копій. Для великих симуляцій зберігайте тільки необхідне.

**Контрольні питання:**

1. Що таке векторизація? Чому вона швидша за цикли Python?
2. Як використати Numba для прискорення коду?
3. Які інструменти профілювання існують у Python?
4. Як паралелізувати обчислення методу Монте-Карло?

## ЛЕКЦІЯ 8

### Практичні аспекти та реальні приклади

#### 8.1. Валідація та верифікація моделей

Верифікація – перевірка правильності програмної реалізації. Валідація – перевірка відповідності моделі реальності. Методи: порівняння з аналітичними розв'язками, експериментальними даними, ретроспективне прогнозування.

#### 8.2. Калібрування параметрів

Калібрування – підбір параметрів моделі для найкращого опису даних. Метод найменших квадратів мінімізує суму квадратів відхилень. `scipy.optimize.minimize`, `scipy.optimize.curve_fit` – основні інструменти.

#### 8.3. Метрики якості моделі

MSE – середньоквадратична похибка; MAE – середня абсолютна похибка; R2 – коефіцієнт детермінації (частка поясненої дисперсії). R2 = 1 означає ідеальну модель, R2 = 0 – модель не краща за константу.

#### 8.4. Аналіз залишків

Залишки – різниця між спостереженнями та прогнозами. Для адекватної моделі залишки мають бути випадковими, нормально розподіленими, з нульовим середнім. Тест Шапіро-Вілка перевіряє нормальність (`scipy.stats.shapiro`).

#### 8.5. Реальний приклад: аналіз інформаційної кампанії

Етапи: попередній аналіз даних, вибір моделі, калібрування параметрів, валідація, прогнозування з довірчими інтервалами, аналіз чутливості. Результати: графіки, таблиці параметрів, імовірнісні оцінки результату.

#### 8.6. Підготовка звіту з моделювання

Структура звіту: Вступ, Математична модель, Чисельні методи, Результати, Обговорення, Висновки. Візуалізації: якісні графіки (`dpi>=300`), підписи осей. Відтворюваність: фіксація `seed`, версій коду, параметрів експериментів.

#### Контрольні питання:

1. Чим верифікація відрізняється від валідації?
2. Як інтерпретувати коефіцієнт R2?
3. Які властивості мають мати залишки адекватної моделі?
4. Опишіть структуру наукового звіту з моделювання.

# ЧАСТИНА II. ЛАБОРАТОРНІ РОБОТИ

## ЛАБОРАТОРНА РОБОТА 1

### Основи Python для математичного моделювання

#### Мета роботи

Ознайомитись з базовими можливостями Python та бібліотек NumPy, Matplotlib для чисельних обчислень та візуалізації.

#### Теоретичні відомості

Python – високорівнева мова програмування. NumPy надає підтримку багатовимірних масивів та векторизованих операцій. Matplotlib – бібліотека для візуалізації даних.

#### Завдання 1. Побудова графіка функції

Створіть масив чисел від 0 до 10 з кроком 0.1. Обчисліть  $f(x) = \sin(x) + 0.5 \cdot \sin(3x)$ . Побудуйте графік.

#### Покрокова інструкція

**Крок 1. Імпорт бібліотек.** Підключаємо NumPy та Matplotlib:

```
import numpy as np
import matplotlib.pyplot as plt
```

**Крок 2. Створення масиву x.** Функція np.arange створює рівномірну сітку:

```
# Масив від 0 до 10 з кроком 0.1
x = np.arange(0, 10, 0.1)
print(f'Кількість точок: {len(x)}') # 100 точок
```

**Крок 3. Обчислення функції.** Векторизовані операції NumPy:

```
# f(x) застосовується до всього масиву одночасно
y = np.sin(x) + 0.5 * np.sin(3 * x)
```

**Крок 4. Побудова графіка.** Налаштовуємо візуалізацію:

```
plt.figure(figsize=(10, 6))
plt.plot(x, y, 'b-', linewidth=1.5, label='f(x)=sin(x)+0.5sin(3x)')
plt.xlabel('x', fontsize=12)
plt.ylabel('f(x)', fontsize=12)
plt.title('Графік функції', fontsize=14)
plt.legend()
```

```
plt.grid(True, alpha=0.3)  
plt.show()
```

## Завдання 2. Генерація випадкових чисел

Згенеруйте 1000 випадкових чисел з  $N(0,1)$ . Побудуйте гістограму та обчисліть статистики.

### Покрокова інструкція

**Крок 1. Генерація даних.** Встановлюємо seed для відтворюваності:

```
np.random.seed(42)  
data = np.random.randn(1000) # 1000 чисел з N(0,1)
```

**Крок 2. Обчислення статистик.** NumPy має вбудовані функції:

```
mean = np.mean(data) # Середнє  
var = np.var(data) # Дисперсія  
std = np.std(data) # Стандартне відхилення  
min_val = np.min(data) # Мінімум  
max_val = np.max(data) # Максимум  
  
print(f'Середнє: {mean:.4f}')  
print(f'Дисперсія: {var:.4f}')  
print(f'Min: {min_val:.4f}, Max: {max_val:.4f}')
```

**Крок 3. Побудова гістограми.** Порівнюємо з теоретичним розподілом:

```
plt.figure(figsize=(10, 6))  
plt.hist(data, bins=30, density=True, alpha=0.7, edgecolor='black')  
  
# Теоретична крива N(0,1)  
x_theory = np.linspace(-4, 4, 100)  
y_theory = np.exp(-x_theory**2/2) / np.sqrt(2*np.pi)  
plt.plot(x_theory, y_theory, 'r-', lw=2, label='N(0,1)')  
  
plt.xlabel('Значення')  
plt.ylabel('Густина')  
plt.legend()  
plt.show()
```

## Завдання 3. Обчислення факторіала

Реалізуйте факторіал рекурсивно та ітеративно. Порівняйте час виконання.

### Покрокова інструкція

**Крок 1. Ітеративна реалізація.** Використовуємо цикл:

```
def factorial_iter(n):
```

```
result = 1
for i in range(2, n + 1):
    result *= i
return result
```

### Крок 2. Рекурсивна реалізація. Функція викликає сама себе:

```
import sys
sys.setrecursionlimit(2000)

def factorial_rec(n):
    if n <= 1:
        return 1
    return n * factorial_rec(n - 1)
```

### Крок 3. Порівняння часу. Вимірюємо час виконання:

```
import time

for n in [100, 500, 1000]:
    # Ітеративний
    start = time.perf_counter()
    for _ in range(100): factorial_iter(n)
    t_iter = (time.perf_counter() - start) / 100 * 1000

    # Рекурсивний
    start = time.perf_counter()
    for _ in range(100): factorial_rec(n)
    t_rec = (time.perf_counter() - start) / 100 * 1000

    print(f'n={n}: iter={t_iter:.4f}ms, rec={t_rec:.4f}ms')
```

## Завдання 4 (індивідуальне). Параметричні криві

За варіантом побудуйте: 1-5 – спіраль Архімеда; 6-10 – равлик Паскаля; 11-15 – кардіоїда; 16-20 – лемніската; 21-25 – лімакон.

### Покрокова інструкція

**Крок 1. Реалізація кривих.** Кожна крива задається в полярних координатах:

```
# Спіраль Архімеда:  $r = a + b \cdot \theta$ 
def archimedes(theta, a=1, b=0.5):
    r = a + b * theta
    return r * np.cos(theta), r * np.sin(theta)

# Кардіоїда:  $r = a \cdot (1 + \cos(\theta))$ 
def cardioid(theta, a=2):
    r = a * (1 + np.cos(theta))
```

```
return r * np.cos(theta), r * np.sin(theta)

# Лемніскага:  $r^2 = a^2 \cos(2\theta)$ 
def lemniscate(theta, a=2):
    r2 = np.maximum(0, a**2 * np.cos(2*theta))
    r = np.sqrt(r2)
    return r * np.cos(theta), r * np.sin(theta)
```

## Крок 2. Побудова графіка. Приклад для кардіоїди:

```
theta = np.linspace(0, 2*np.pi, 1000)
x, y = cardioid(theta)

plt.figure(figsize=(8, 8))
plt.plot(x, y, 'b-', linewidth=2)
plt.axis('equal')
plt.title('Кардіоїда')
plt.grid(True)
plt.show()
```

## Критерії оцінювання

Максимум 100 балів: Завдання 1 – 20 б.; Завдання 2 – 25 б.; Завдання 3 – 20 б.;  
Завдання 4 – 25 б.; Оформлення – 10 б.

## Контрольні питання:

1. Чим масив NumPy відрізняється від списку Python?
2. Що таке векторизація у NumPy?
3. Як створити підграфіки в Matplotlib?
4. Порівняйте рекурсію та ітерацію.

## ЛАБОРАТОРНА РОБОТА 2

### Чисельне розв'язання диференціальних рівнянь

#### Мета роботи

Освоїти методи Ейлера та Рунге-Кутти для чисельного розв'язання ЗДР.  
Навчитись використовувати SciPy.

#### Теоретичні відомості

Метод Ейлера:  $y_{n+1} = y_n + h \cdot f(t_n, y_n)$ , точність  $O(h)$ . Метод RK4 має точність  $O(h^4)$  завдяки обчисленню чотирьох проміжних значень  $k_1, k_2, k_3, k_4$ .

#### Завдання 1. Метод Ейлера

Реалізуйте метод Ейлера для  $dy/dt = -2y$ ,  $y(0) = 1$ . Порівняйте з точним розв'язком  $y(t) = e^{-2t}$ .

#### Покрокова інструкція

**Крок 1. Реалізація методу Ейлера.** Основний цикл методу:

```
def euler_method(f, y0, t_span, h):  
    """Метод Ейлера для  $dy/dt = f(t, y)$ """  
    t0, tf = t_span  
    t = np.arange(t0, tf + h, h)  
    y = np.zeros(len(t))  
    y[0] = y0  
  
    for i in range(len(t) - 1):  
        y[i+1] = y[i] + h * f(t[i], y[i])  
  
    return t, y
```

**Крок 2. Визначення рівняння.** Права частина та точний розв'язок:

```
def f(t, y):  
    return -2 * y  
  
def exact(t):  
    return np.exp(-2 * t)
```

**Крок 3. Порівняння при різних  $h$ .** Аналіз похибки:

```
y0, t_span = 1.0, (0, 3)  
  
for h in [0.1, 0.01, 0.001]:  
    t, y = euler_method(f, y0, t_span, h)
```

```
error = abs(y[-1] - exact(t[-1]))  
print(f'h={h}: похибка = {error:.6e}')
```

```
# Графік порівняння  
t, y = euler_method(f, y0, t_span, 0.1)  
t_ex = np.linspace(0, 3, 100)  
plt.plot(t_ex, exact(t_ex), 'b-', label='Точний')  
plt.plot(t, y, 'ro--', label='Ейлер')  
plt.legend(); plt.grid()  
plt.show()
```

## Завдання 2. Метод Рунге-Кутти 4-го порядку

Реалізуйте RK4 та порівняйте точність з методом Ейлера при однаковому кроці.

### Покрокова інструкція

**Крок 1. Реалізація RK4.** Обчислюємо  $k_1, k_2, k_3, k_4$ :

```
def rk4_method(f, y0, t_span, h):  
    """Метод Рунге-Кутти 4-го порядку"""  
    t0, tf = t_span  
    t = np.arange(t0, tf + h, h)  
    y = np.zeros(len(t))  
    y[0] = y0  
  
    for i in range(len(t) - 1):  
        k1 = f(t[i], y[i])  
        k2 = f(t[i] + h/2, y[i] + h*k1/2)  
        k3 = f(t[i] + h/2, y[i] + h*k2/2)  
        k4 = f(t[i] + h, y[i] + h*k3)  
        y[i+1] = y[i] + h*(k1 + 2*k2 + 2*k3 + k4)/6  
  
    return t, y
```

**Крок 2. Порівняння методів.** При однаковому кроці  $h$ :

```
h = 0.1  
t_e, y_e = euler_method(f, y0, t_span, h)  
t_rk, y_rk = rk4_method(f, y0, t_span, h)  
  
err_euler = abs(y_e[-1] - exact(3))  
err_rk4 = abs(y_rk[-1] - exact(3))  
print(f'Ейлер: {err_euler:.6e}')  
print(f'RK4: {err_rk4:.6e}')
```

### Завдання 3. Система Лотки-Вольтерри

Розв'яжіть систему хижак-жертва:  $dx/dt = \alpha x - \beta xy$ ,  $dy/dt = \delta xy - \gamma y$  за допомогою `solve_ivp`.

#### Покрокова інструкція

**Крок 1. Визначення системи.** Права частина системи ДР:

```
from scipy.integrate import solve_ivp

def lotka_volterra(t, y, alpha, beta, gamma, delta):
    """Система Лотки-Вольтерри"""
    x, y_pred = y
    dxdt = alpha*x - beta*x*y_pred # Жертви
    dydt = delta*x*y_pred - gamma*y_pred # Хижаки
    return [dxdt, dydt]
```

**Крок 2. Розв'язання та візуалізація.** Використовуємо `solve_ivp`:

```
# Параметри
alpha, beta, gamma, delta = 1.1, 0.4, 0.4, 0.1
y0 = [10, 10] # Початкові популяції
t_span = (0, 100)

# Розв'язуємо
sol = solve_ivp(lotka_volterra, t_span, y0,
                args=(alpha, beta, gamma, delta),
                dense_output=True, max_step=0.1)

# Візуалізація
t = np.linspace(0, 100, 1000)
z = sol.sol(t)

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 5))

# Динаміка популяцій
ax1.plot(t, z[0], 'b-', label='Жертви')
ax1.plot(t, z[1], 'r-', label='Хижаки')
ax1.set_xlabel('Час'); ax1.legend()

# Фазовий портрет
ax2.plot(z[0], z[1], 'g-')
ax2.set_xlabel('Жертви'); ax2.set_ylabel('Хижаки')
plt.show()
```

#### Завдання 4 (індивідуальне). Дослідження системи

Дослідіть поведінку системи при різних початкових умовах (варіанти задають  $x(0)$  та  $y(0)$  від 5 до 50).

##### Покрокова інструкція

**Крок 1. Побудова фазового портрета.** Декілька траєкторій з різними початковими умовами:

```
plt.figure(figsize=(10, 10))

# Різні початкові умови
initial_conditions = [(10, 5), (20, 10), (30, 15), (40, 20)]

for x0, y0_init in initial_conditions:
    sol = solve_ivp(lotka_volterra, (0, 100), [x0, y0_init],
                    args=(alpha, beta, gamma, delta),
                    dense_output=True)
    t = np.linspace(0, 100, 500)
    z = sol.sol(t)
    plt.plot(z[0], z[1], label=f'({x0}, {y0_init})')

plt.xlabel('Жертви'); plt.ylabel('Хижак')
plt.title('Фазовий портрет системи Лотки-Вольтерри')
plt.legend()
plt.show()
```

##### Критерії оцінювання

Максимум 100 балів: Завдання 1 – 20 б.; Завдання 2 – 25 б.; Завдання 3 – 25 б.;  
Завдання 4 – 20 б.; Оформлення – 10 б.

##### Контрольні питання:

1. Який порядок точності має метод Ейлера?
2. Поясніть алгоритм RK4.
3. Що таке жорсткі системи ДР?
4. Яка фізична інтерпретація системи Лотки-Вольтерри?

## ЛАБОРАТОРНА РОБОТА 3

### Моделювання випадкових процесів

#### Мета роботи

Освоїти методи генерації випадкових процесів. Змоделювати вінерівський та пуассонівський процеси.

#### Теоретичні відомості

Вінерівський процес  $W(t)$ :  $W(0)=0$ , прирости  $W(t)-W(s) \sim N(0, t-s)$ , незалежні на неперекривних інтервалах. Пуассонівський процес  $N(t)$  з інтенсивністю  $\lambda$  – процес лічби подій з експоненціальними інтервалами.

#### Завдання 1. Вінерівський процес

Згенеруйте 5 траєкторій вінерівського процесу на  $[0, 1]$  з  $N=1000$  кроків.

#### Покрокова інструкція

**Крок 1. Генерація вінерівського процесу.** Прирости мають розподіл  $N(0, \Delta t)$ :

```
def wiener_process(T, N, seed=None):
    """
    Генерація вінерівського процесу.
    T - кінцевий час, N - кількість кроків
    """
    if seed is not None:
        np.random.seed(seed)

    dt = T / N # Крок по часу

    # Прирости: dW ~ N(0, dt)
    dW = np.sqrt(dt) * np.random.randn(N)

    # Кумулятивна сума приростів
    W = np.concatenate([[0], np.cumsum(dW)])
    t = np.linspace(0, T, N + 1)

    return t, W
```

**Крок 2. Побудова кількох траєкторій.** Генеруємо 5 реалізацій:

```
plt.figure(figsize=(12, 6))

for i in range(5):
    t, W = wiener_process(T=1, N=1000, seed=i)
```

```
plt.plot(t, W, label=f'Траєкторія {i+1}')

plt.xlabel('Час t')
plt.ylabel('W(t)')
plt.title('5 траєкторій вінерівського процесу')
plt.legend()
plt.grid(True, alpha=0.3)
plt.show()
```

## Завдання 2. Перевірка властивостей

Перевірте  $E[W(1)] \approx 0$  та  $\text{Var}[W(1)] \approx 1$  за 1000 реалізацій.

### Покрокова інструкція

**Крок 1. Генерація 1000 реалізацій.** Зберігаємо кінцеві значення  $W(1)$ :

```
n_simulations = 1000
W_final = np.zeros(n_simulations)

for i in range(n_simulations):
    t, W = wiener_process(T=1, N=1000)
    W_final[i] = W[-1] # Значення W(1)

# Статистики
mean_W = np.mean(W_final)
var_W = np.var(W_final)

print(f'E[W(1)] = {mean_W:.4f} (теор: 0)')
print(f'Var[W(1)] = {var_W:.4f} (теор: 1)')
```

**Крок 2. Гістограма  $W(1)$ .** Порівняння з  $N(0,1)$ :

```
plt.figure(figsize=(10, 6))
plt.hist(W_final, bins=40, density=True, alpha=0.7)

x = np.linspace(-4, 4, 100)
plt.plot(x, np.exp(-x**2/2)/np.sqrt(2*np.pi), 'r-', lw=2)
plt.title('Розподіл W(1)')
plt.show()
```

## Завдання 3. Пуассонівський процес

Змоделюйте пуассонівський процес з  $\lambda=5$  на  $[0, 10]$ . Побудуйте  $N(t)$  та гістограму інтервалів.

### Покрокова інструкція

**Крок 1. Генерація пуассонівського процесу.** Інтервали між подіями  $\sim \text{Exp}(\lambda)$ :

```
def poisson_process(lam, T, seed=None):
    """
    Генерація пуассонівського процесу.
    lam - інтенсивність, T - кінцевий час
    """
    if seed is not None:
        np.random.seed(seed)

    # Генеруємо інтервали між подіями
    intervals = []
    current_time = 0

    while current_time < T:
        # Інтервал ~ Exp(lam)
        dt = np.random.exponential(1/lam)
        current_time += dt
        if current_time < T:
            intervals.append(dt)

    # Часи подій
    event_times = np.cumsum(intervals)

    return event_times, intervals
```

## Крок 2. Візуалізація $N(t)$ . Ступінчаста функція:

```
lam, T = 5, 10
events, intervals = poisson_process(lam, T, seed=42)

# Будуємо  $N(t)$ 
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 5))

# Ступінчастий графік
t_plot = np.concatenate([[0], events, [T]])
N_plot = np.arange(len(events) + 2)
N_plot[-1] = N_plot[-2]
ax1.step(t_plot, N_plot, where='post')
ax1.set_xlabel('Час t'); ax1.set_ylabel('N(t)')
ax1.set_title('Пуассонівський процес N(t)')

# Гістограма інтервалів
ax2.hist(intervals, bins=20, density=True, alpha=0.7)
x = np.linspace(0, max(intervals), 100)
ax2.plot(x, lam*np.exp(-lam*x), 'r-', lw=2, label=f'Exp({lam})')
ax2.legend()
plt.show()
```

## Завдання 4 (індивідуальне). Складений пуассонівський процес

За варіантом: 1-8 –  $N(\mu, \sigma^2)$ ; 9-16 –  $U(a, b)$ ; 17-25 –  $\text{Exp}(\lambda)$  для величин стрибків.

### Покрокова інструкція

**Крок 1. Складений процес.**  $S(t) = \sum X_i$ , де  $X_i$  – випадкові величини:

```
def compound_poisson(lam, T, jump_dist='normal', params=(0, 1)):
    """Складений пуассонівський процес"""
    events, _ = poisson_process(lam, T)
    n_jumps = len(events)

    # Генеруємо величини стрибків
    if jump_dist == 'normal':
        mu, sigma = params
        jumps = np.random.normal(mu, sigma, n_jumps)
    elif jump_dist == 'uniform':
        a, b = params
        jumps = np.random.uniform(a, b, n_jumps)
    else: # exponential
        rate = params[0]
        jumps = np.random.exponential(1/rate, n_jumps)

    # Кумулятивна сума
    S = np.cumsum(jumps)
    return events, S
```

**Крок 2. Візуалізація.** Побудова траєкторії  $S(t)$ :

```
events, S = compound_poisson(5, 10, 'normal', (10, 5))

plt.figure(figsize=(10, 6))
plt.step(np.concatenate([[0], events]), np.concatenate([[0], S]),
        where='post')
plt.xlabel('Час'); plt.ylabel('S(t)')
plt.title('Складений пуассонівський процес')
plt.show()
```

### Критерії оцінювання

Максимум 100 балів: Завдання 1 – 20 б.; Завдання 2 – 25 б.; Завдання 3 – 25 б.;  
Завдання 4 – 20 б.; Оформлення – 10 б.

### Контрольні питання:

1. Які властивості має вінерівський процес?
2. Як генерувати прирости вінерівського процесу?
3. Що таке інтенсивність пуассонівського процесу?
4. Який розподіл мають інтервали між подіями?

## ЛАБОРАТОРНА РОБОТА 4

### Реалізація моделі інформаційної боротьби

#### Мета роботи

Реалізувати детерміністичну модель інформаційної боротьби для  $n$  типів загроз та дослідити її поведінку.

#### Теоретичні відомості

Модель:  $dN_i/dt = (\alpha_i + \beta_i N_i)(N_0 - \sum N_j)$ , де  $N_0$  – популяція,  $N_i$  – прихильники  $i$ -го типу,  $\alpha_i$  – зовнішній канал,  $\beta_i$  – внутрішній канал поширення.

#### Завдання 1. Модель для 2 типів

Реалізуйте модель з  $N_0=10000$ ,  $\alpha_1=0.001$ ,  $\alpha_2=0.0012$ ,  $\beta_1=0.00001$ ,  $\beta_2=0.000008$ .

#### Покрокова інструкція

**Крок 1. Визначення моделі.** Права частина системи ДР:

```
from scipy.integrate import solve_ivp

def info_warfare(t, N, N0, alphas, betas):
    """
    Модель інформаційної боротьби.
    N - вектор [N1, N2, ..., Nn] кількості прихильників
    N0 - загальна популяція
    alphas - [ $\alpha_1$ ,  $\alpha_2$ , ...] інтенсивності зовнішнього каналу
    betas - [ $\beta_1$ ,  $\beta_2$ , ...] інтенсивності внутрішнього каналу
    """
    n_types = len(alphas)
    N_free = N0 - np.sum(N) # Нерекрутовані
    N_free = max(0, N_free) # Не може бути від'ємним

    dN = np.zeros(n_types)
    for i in range(n_types):
        dN[i] = (alphas[i] + betas[i]*N[i]) * N_free

    return dN
```

**Крок 2. Розв'язання та візуалізація.** Задаємо параметри та будуємо графік:

```
# Параметри
N0 = 10000
alphas = [0.001, 0.0012]
betas = [0.00001, 0.000008]
N_init = [100, 100] # Початкові кількості
```

```
# Розв'язуємо
sol = solve_ivp(info_warfare, [0, 500], N_init,
               args=(N0, alphas, betas),
               dense_output=True, max_step=1)

t = np.linspace(0, 500, 1000)
N = sol.sol(t)

# Графік
plt.figure(figsize=(12, 5))
plt.plot(t, N[0], 'b-', label='Тип 1', linewidth=2)
plt.plot(t, N[1], 'r-', label='Тип 2', linewidth=2)
plt.plot(t, N0 - N[0] - N[1], 'g--', label='Нерекрутовані')
plt.xlabel('Час'); plt.ylabel('Кількість')
plt.title('Динаміка інформаційної боротьби')
plt.legend(); plt.grid(True)
plt.show()
```

## Завдання 2. Розширення до 4 типів

Додайте ще 2 типи інформаційних загроз та дослідіть динаміку.

### Покрокова інструкція

**Крок 1. Модель для 4 типів.** Розширюємо параметри:

```
# 4 типи загроз
alphas_4 = [0.001, 0.0012, 0.0008, 0.0015]
betas_4 = [0.00001, 0.000008, 0.000012, 0.000006]
N_init_4 = [50, 50, 50, 50]

sol = solve_ivp(info_warfare, [0, 500], N_init_4,
               args=(N0, alphas_4, betas_4),
               dense_output=True)

t = np.linspace(0, 500, 1000)
N = sol.sol(t)

plt.figure(figsize=(12, 5))
colors = ['b', 'r', 'g', 'm']
for i in range(4):
    plt.plot(t, N[i], colors[i], label=f'Тип {i+1}')
plt.legend(); plt.grid()
plt.show()
```

### Завдання 3. Аналіз чутливості

Дослідіть, як зміна  $\alpha$  та  $\beta$  на  $\pm 10\%$ ,  $\pm 20\%$  впливає на результат.

#### Покрокова інструкція

**Крок 1. Варіації параметрів.** Систематичний аналіз:

```
# Базові параметри
alpha_base = 0.001
beta_base = 0.00001
variations = [0.8, 0.9, 1.0, 1.1, 1.2] # -20%, -10%, 0%, +10%, +20%

results = []

for var in variations:
    alphas = [alpha_base * var, 0.0012]
    betas = [beta_base * var, 0.000008]

    sol = solve_ivp(info_warfare, [0, 500], [100, 100],
                    args=(N0, alphas, betas), dense_output=True)
    N_final = sol.sol(500)
    results.append(N_final[0]) # Фінальна кількість типу 1

# Візуалізація
plt.figure(figsize=(10, 5))
plt.bar(['-20%', '-10%', '0%', '+10%', '+20%'], results)
plt.ylabel('Фінальна кількість типу 1')
plt.title('Аналіз чутливості до параметрів')
plt.show()
```

### Завдання 4 (індивідуальне). Модель для n типів

Реалізуйте модель для n = варіант + 2 типів та визначте переможця.

#### Покрокова інструкція

**Крок 1. Генерація параметрів.** Випадкові параметри для n типів:

```
variant = 5 # Ваш варіант
n_types = variant + 2

np.random.seed(variant)
alphas = np.random.uniform(0.0005, 0.002, n_types)
betas = np.random.uniform(0.000005, 0.00002, n_types)
N_init = np.ones(n_types) * (N0 / (n_types * 10))

sol = solve_ivp(info_warfare, [0, 1000], N_init,
                args=(N0, alphas, betas), dense_output=True)
```

```
t = np.linspace(0, 1000, 2000)
N = sol.sol(t)

# Визначаємо переможця
winner = np.argmax(N[:, -1]) + 1
print(f'Переможець: Тип {winner}')
print(f'Параметри:  $\alpha$ ={alphas[winner-1]:.6f},  $\beta$ ={betas[winner-1]:.6f}')
```

### Критерії оцінювання

Максимум 100 балів: Завдання 1 – 20 б.; Завдання 2 – 20 б.; Завдання 3 – 25 б.;  
Завдання 4 – 25 б.; Оформлення – 10 б.

### Контрольні питання:

1. Поясніть фізичний зміст параметрів  $\alpha$  та  $\beta$ .
2. Що означає 'зовнішній' та 'внутрішній' канали?
3. Як визначити переможця інформаційної боротьби?
4. Які обмеження має детерміністична модель?

## ЛАБОРАТОРНА РОБОТА 5

### Стохастична модель інформаційної боротьби

#### Мета роботи

Реалізувати стохастичну модель з дифузійними збуреннями методом Ейлера-Маруяма.

#### Теоретичні відомості

Стохастична модель:  $dN_i = (\alpha_i + \beta_i N_i)(N_0 - \sum N_j)dt + \sigma_i N_i dW_i$ . Метод Ейлера-Маруяма:  $N(t+\Delta t) = N(t) + f(N)\Delta t + g(N)\Delta W$ , де  $\Delta W \sim N(0, \Delta t)$ .

#### Завдання 1. Метод Ейлера-Маруяма

Реалізуйте стохастичну модель для 2 типів та порівняйте з детерміністичною.

#### Покрокова інструкція

**Крок 1. Реалізація методу.** Стохастична схема інтегрування:

```
def stochastic_info_warfare(N0, alphas, betas, sigmas, N_init, T, dt):
```

```
    """
```

```
    Стохастична модель методом Ейлера-Маруяма.
```

```
    sigmas - коефіцієнти волатильності [ $\sigma_1, \sigma_2, \dots$ ]
```

```
    """
```

```
    n_types = len(alphas)
```

```
    steps = int(T / dt)
```

```
    N = np.zeros((steps + 1, n_types))
```

```
    N[0] = N_init
```

```
    sqrt_dt = np.sqrt(dt)
```

```
    for k in range(steps):
```

```
        # Нерекрутована частина
```

```
        N_free = max(0, N0 - np.sum(N[k]))
```

```
        # Вінерівські прирости
```

```
        dW = sqrt_dt * np.random.randn(n_types)
```

```
        for i in range(n_types):
```

```
            # Детерміністична частина (drift)
```

```
            drift = (alphas[i] + betas[i]*N[k,i]) * N_free
```

```
            # Стохастична частина (diffusion)
```

```
            diffusion = sigmas[i] * N[k,i]
```

```
            # Оновлення (Ейлер-Маруяма)
```

```
N[k+1,i] = max(0, N[k,i] + drift*dt + diffusion*dW[i])
```

```
return N
```

## Крок 2. Порівняння моделей. Детерміністична vs стохастична:

```
# Параметри
N0 = 10000
alphas = [0.001, 0.0012]
betas = [0.00001, 0.000008]
sigmas = [0.05, 0.05] # Волатильність 5%
N_init = [100, 100]
T, dt = 500, 0.5

# Детерміністична модель
sol_det = solve_ivp(info_warfare, [0, T], N_init,
                    args=(N0, alphas, betas), dense_output=True)
t_det = np.linspace(0, T, 1000)
N_det = sol_det.sol(t_det)

# Стохастична модель
N_stoch = stochastic_info_warfare(N0, alphas, betas, sigmas, N_init, T, dt)
t_stoch = np.linspace(0, T, len(N_stoch))

# Графік
plt.figure(figsize=(12, 5))
plt.plot(t_det, N_det[0], 'b-', lw=2, label='Детерм. тип 1')
plt.plot(t_stoch, N_stoch[:,0], 'b--', alpha=0.7, label='Стохаст. тип 1')
plt.legend(); plt.grid()
plt.show()
```

## Завдання 2. Довірчий інтервал

За 100 реалізаціями обчисліть середню траєкторію та 5%-95% квантилі.

### Покрокова інструкція

#### Крок 1. Множинні реалізації. Генеруємо 100 траєкторій:

```
n_simulations = 100
all_trajectories = []

for _ in range(n_simulations):
    N = stochastic_info_warfare(N0, alphas, betas, sigmas, N_init, T, dt)
    all_trajectories.append(N[:, 0]) # Тип 1

all_trajectories = np.array(all_trajectories)
```

## Крок 2. Статистики та візуалізація. Обчислення квантилей:

```
# Середня траєкторія та квантилі
mean_traj = np.mean(all_trajectories, axis=0)
q05 = np.percentile(all_trajectories, 5, axis=0)
q95 = np.percentile(all_trajectories, 95, axis=0)

t = np.linspace(0, T, len(mean_traj))

plt.figure(figsize=(12, 6))
plt.fill_between(t, q05, q95, alpha=0.3, label='90% довірчий інтервал')
plt.plot(t, mean_traj, 'b-', lw=2, label='Середня траєкторія')
plt.plot(t_det, N_det[0], 'r--', lw=2, label='Детерміністична')
plt.xlabel('Час'); plt.ylabel('N1')
plt.legend(); plt.grid()
plt.show()
```

## Завдання 3. Вплив волатильності

Дослідіть поведінку при  $\sigma = 0.01, 0.05, 0.1, 0.2$ .

### Покрокова інструкція

#### Крок 1. Порівняння різних $\sigma$ . Систематичне дослідження:

```
sigma_values = [0.01, 0.05, 0.1, 0.2]

fig, axes = plt.subplots(2, 2, figsize=(14, 10))
axes = axes.flatten()

for idx, sigma in enumerate(sigma_values):
    sigmas_test = [sigma, sigma]

    # 10 траєкторій для кожного  $\sigma$ 
    for _ in range(10):
        N = stochastic_info_warfare(N0, alphas, betas, sigmas_test,
                                   N_init, T, dt)
        t = np.linspace(0, T, len(N))
        axes[idx].plot(t, N[:,0], 'b-', alpha=0.3)

    axes[idx].plot(t_det, N_det[0], 'r--', lw=2)
    axes[idx].set_title(f' $\sigma = \{sigma\}$ ')
    axes[idx].set_xlabel('Час'); axes[idx].grid()

plt.tight_layout()
plt.show()
```

## Завдання 4 (індивідуальне). Ймовірність перемоги

За 1000 симуляцій оцініть ймовірність перемоги кожного типу.

### Покрокова інструкція

**Крок 1. Масове моделювання.** Підраховуємо переможців:

```
n_simulations = 1000
winners = []

for _ in range(n_simulations):
    N = stochastic_info_warfare(N0, alphas, betas, sigmas, N_init, T, dt)
    winner = np.argmax(N[-1])
    winners.append(winner)

# Ймовірності перемоги
from collections import Counter
counts = Counter(winners)

for type_id in range(len(alphas)):
    prob = counts[type_id] / n_simulations
    print(f'Тип {type_id+1}: P(перемога) = {prob:.3f}')
```

### Критерії оцінювання

Макимум 100 балів: Завдання 1 – 25 б.; Завдання 2 – 25 б.; Завдання 3 – 20 б.;  
Завдання 4 – 20 б.; Оформлення – 10 б.

### Контрольні питання:

1. Поясніть метод Ейлера-Маруяма.
2. Що означає коефіцієнт волатильності  $\sigma$ ?
3. Як обчислити довірчий інтервал?
4. Чому стохастична модель краща за детерміністичну?

## ЛАБОРАТОРНА РОБОТА 6

### Модель з пуассонівськими стрибками

#### Мета роботи

Реалізувати модель з пуассонівськими стрибками для моделювання раптових подій.

#### Теоретичні відомості

Пуассонівські стрибки моделюють вірусне поширення, раптові новини тощо. Модель:  $dN_i = C(N_i)dt + J_i dP_i(t)$ , де  $P_i(t)$  – пуассонівський процес з  $\lambda_i$ ,  $J_i$  – величина стрибка.

#### Завдання 1. Модель зі стрибками

Реалізуйте модель з  $\lambda=5$  та стрибками з  $N(100, 50)$ .

#### Покрокова інструкція

**Крок 1. Реалізація моделі.** Додаємо пуассонівські стрибки:

```
def info_warfare_jumps(N0, alphas, betas, lambdas, jump_params,
                      N_init, T, dt):
    """
    Модель з пуассонівськими стрибками.
    lambdas - інтенсивності стрибків [ $\lambda_1, \lambda_2, \dots$ ]
    jump_params - параметри стрибків [ $(\mu_1, \sigma_1), (\mu_2, \sigma_2), \dots$ ]
    """
    n_types = len(alphas)
    steps = int(T / dt)
    N = np.zeros((steps + 1, n_types))
    N[0] = N_init

    for k in range(steps):
        N_free = max(0, N0 - np.sum(N[k]))

        for i in range(n_types):
            # Детерміністична частина
            drift = (alphas[i] + betas[i]*N[k,i]) * N_free * dt

            # Пуассонівські стрибки
            n_jumps = np.random.poisson(lambdas[i] * dt)
            if n_jumps > 0:
                mu, sigma = jump_params[i]
                jumps = np.sum(np.random.normal(mu, sigma, n_jumps))
            else:
```

```
jumps = 0

N[k+1,i] = max(0, N[k,i] + drift + jumps)

return N
```

## Крок 2. Тестування моделі. Візуалізація траєкторії:

```
# Параметри
N0 = 10000
alphas = [0.001, 0.0012]
betas = [0.00001, 0.000008]
lambdas = [5, 5] # λ = 5 стрибків на одиницю часу
jump_params = [(100, 50), (100, 50)] # N(100, 50)
N_init = [100, 100]
T, dt = 500, 0.5

N = info_warfare_jumps(N0, alphas, betas, lambdas, jump_params,
                       N_init, T, dt)
t = np.linspace(0, T, len(N))

plt.figure(figsize=(12, 5))
plt.plot(t, N[:,0], 'b-', label='Тип 1')
plt.plot(t, N[:,1], 'r-', label='Тип 2')
plt.xlabel('Час'); plt.ylabel('Кількість')
plt.title('Модель з пуассонівськими стрибками')
plt.legend(); plt.grid()
plt.show()
```

## Завдання 2. Порівняння моделей

Порівняйте модель зі стрибками та без стрибків на одному графіку.

### Покрокова інструкція

#### Крок 1. Порівняльний аналіз. Детерміністична vs стрибкова:

```
# Детерміністична
sol_det = solve_ivp(info_warfare, [0, T], N_init,
                    args=(N0, alphas, betas), dense_output=True)
t_det = np.linspace(0, T, 1000)
N_det = sol_det.sol(t_det)

# Зі стрибками
N_jump = info_warfare_jumps(N0, alphas, betas, lambdas, jump_params,
                             N_init, T, dt)
t_jump = np.linspace(0, T, len(N_jump))
```

```
plt.figure(figsize=(12, 5))
plt.plot(t_det, N_det[0], 'b-', lw=2, label='Детерм. тип 1')
plt.plot(t_jump, N_jump[:,0], 'b--', alpha=0.7, label='Стрибки тип 1')
plt.plot(t_det, N_det[1], 'r-', lw=2, label='Детерм. тип 2')
plt.plot(t_jump, N_jump[:,1], 'r--', alpha=0.7, label='Стрибки тип 2')
plt.legend(); plt.grid()
plt.show()
```

### Завдання 3. Вплив інтенсивності $\lambda$

Дослідіть поведінку при  $\lambda = 1, 5, 10, 20$ .

#### Покрокова інструкція

**Крок 1. Систематичне дослідження.** Порівняння різних  $\lambda$ :

```
lambda_values = [1, 5, 10, 20]

fig, axes = plt.subplots(2, 2, figsize=(14, 10))
axes = axes.flatten()

for idx, lam in enumerate(lambda_values):
    lambdas_test = [lam, lam]

    # 5 траєкторій для кожного  $\lambda$ 
    for _ in range(5):
        N = info_warfare_jumps(N0, alphas, betas, lambdas_test,
                               jump_params, N_init, T, dt)
        t = np.linspace(0, T, len(N))
        axes[idx].plot(t, N[:,0], 'b-', alpha=0.5)

    axes[idx].plot(t_det, N_det[0], 'r-', lw=2)
    axes[idx].set_title(f' $\lambda = \{lam\}$ ')
    axes[idx].grid()

plt.tight_layout()
plt.show()
```

### Завдання 4 (індивідуальне). Асиметричні стрибки

Реалізуйте модель з різними  $\lambda$  для кожного типу ( $n = \text{варіант mod } 5 + 3$ ).

#### Покрокова інструкція

**Крок 1. Асиметрична модель.** Різні інтенсивності для кожного типу:

```
variant = 5
n_types = variant % 5 + 3
```

```
# Генеруємо різні параметри
np.random.seed(variant)
alphas_n = np.random.uniform(0.0005, 0.002, n_types)
betas_n = np.random.uniform(0.000005, 0.00002, n_types)
lambdas_n = np.random.uniform(1, 20, n_types) # Різні λ!
jump_params_n = [(np.random.uniform(50, 150),
                   np.random.uniform(20, 80)) for _ in range(n_types)]
N_init_n = np.ones(n_types) * 50

# Моделювання
N = info_warfare_jumps(N0, alphas_n, betas_n, lambdas_n,
                       jump_params_n, N_init_n, T, dt)

# Визначення переможця
winner = np.argmax(N[-1]) + 1
print(f'Переможець: Тип {winner}')
print(f'λ переможця: {lambdas_n[winner-1]:.2f}')
```

### Критерії оцінювання

Максимум 100 балів: Завдання 1 – 25 б.; Завдання 2 – 20 б.; Завдання 3 – 25 б.;  
Завдання 4 – 20 б.; Оформлення – 10 б.

### Контрольні питання:

1. Що моделюють пуассонівські стрибки?
2. Як параметр  $\lambda$  впливає на частоту стрибків?
3. Чим складений процес відрізняється від простого?
4. Наведіть приклади реальних подій-стрибків.

## ЛАБОРАТОРНА РОБОТА 7

### Комплексна модель з процесами Леві

#### Мета роботи

Реалізувати комплексну модель, що поєднує детерміністичну динаміку, дифузію та стрибки.

#### Теоретичні відомості

Процес Леві = детермінований зсув + дифузія (вінерівський процес) + стрибки (пуассонівський). Комплексна модель:  $dN_i = C(N_i)dt + \sigma_i N_i dW_i + J_i dP_i$ .

#### Завдання 1. Комплексна модель

Реалізуйте модель з усіма трьома компонентами:  $\sigma=0.05$ ,  $\lambda=3$ , стрибки  $N(50, 25)$ .

#### Покрокова інструкція

**Крок 1. Повна модель Леві.** Об'єднуємо всі компоненти:

```
def levy_info_warfare(N0, alphas, betas, sigmas, lambdas, jump_params,
                      N_init, T, dt):
    n_types = len(alphas)
    steps = int(T / dt)
    N = np.zeros((steps + 1, n_types))
    N[0] = N_init
    sqrt_dt = np.sqrt(dt)

    for k in range(steps):
        N_free = max(0, N0 - np.sum(N[k]))
        dW = sqrt_dt * np.random.randn(n_types)

        for i in range(n_types):
            drift = (alphas[i] + betas[i]*N[k,i]) * N_free * dt
            diffusion = sigmas[i] * N[k,i] * dW[i]
            n_jumps = np.random.poisson(lambdas[i] * dt)
            jumps = np.sum(np.random.normal(jump_params[i][0],
                                             jump_params[i][1], n_jumps)) if n_jumps > 0 else 0
            N[k+1,i] = max(0, N[k,i] + drift + diffusion + jumps)
    return N
```

**Крок 2. Запуск моделі. Тестування:**

```
N0, T, dt = 10000, 500, 0.5
alphas, betas = [0.001, 0.0012], [0.00001, 0.000008]
sigmas, lambdas = [0.05, 0.05], [3, 3]
jump_params = [(50, 25), (50, 25)]
```

```
N_init = [100, 100]

N = levy_info_warfare(N0, alphas, betas, sigmas, lambdas,
                      jump_params, N_init, T, dt)
t = np.linspace(0, T, len(N))
plt.plot(t, N[:,0], 'b-', label='Тип 1')
plt.plot(t, N[:,1], 'r-', label='Тип 2')
plt.legend(); plt.show()
```

### Завдання 2-4

Завдання 2: Порівняйте детерміністичну модель, модель з дифузією та повну модель Леві. Завдання 3: За 500 симуляцій обчисліть статистики та розподіл переможців. Завдання 4: Визначте,  $\sigma$  чи  $\lambda$  має більший вплив на непередбачуваність.

### Критерії оцінювання

Максимум 100 балів: Завдання 1 – 25 б.; Завдання 2 – 25 б.; Завдання 3 – 20 б.; Завдання 4 – 20 б.; Оформлення – 10 б.

### Контрольні питання:

1. Що таке процес Леві та його компоненти?
2. Чому комплексна модель краще описує реальність?

## ЛАБОРАТОРНА РОБОТА 8

### Валідація моделі та аналіз результатів

#### Мета роботи

Навчитись методам валідації математичних моделей та підготовки звітів.

#### Теоретичні відомості

Валідація – перевірка відповідності моделі реальності. Метрики: MSE, MAE,  $R^2$ .

Аналіз залишків показує систематичні відхилення.

#### Завдання 1. Відновлення параметрів

Створіть синтетичні дані та спробуйте відновити параметри моделі методом найменших квадратів.

#### Покрокова інструкція

##### Крок 1. Генерація даних. Модель + шум:

```
from scipy.optimize import minimize

true_alpha, true_beta = 0.001, 0.00001
sol = solve_ivp(lambda t, y: (true_alpha + true_beta*y)*(10000-y),
                [0, 500], [100], dense_output=True)
t_obs = np.linspace(0, 500, 50)
y_true = sol.sol(t_obs)[0]
y_obs = y_true + np.random.normal(0, 100, len(t_obs))
```

##### Крок 2. Оптимізація. Метод Нелдера-Міда:

```
def objective(params, t_obs, y_obs):
    alpha, beta = params
    sol = solve_ivp(lambda t, y: (alpha + beta*y)*(10000-y),
                    [0, 500], [100], t_eval=t_obs)
    return np.mean((y_obs - sol.y[0])**2)

result = minimize(objective, [0.0005, 0.000005], args=(t_obs, y_obs))
print(f'Оцінки:  $\alpha$ ={result.x[0]:.6f},  $\beta$ ={result.x[1]:.8f}')
```

#### Завдання 2. Метрики якості

Обчисліть MSE, MAE,  $R^2$  для відновленої моделі.

### Покрокова інструкція

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

y_pred = sol_pred.y[0]
print(f'MSE: {mean_squared_error(y_obs, y_pred):.2f}')
print(f'MAE: {mean_absolute_error(y_obs, y_pred):.2f}')
print(f'R2: {r2_score(y_obs, y_pred):.4f}')
```

### Завдання 3. Аналіз залишків

Побудуйте гістограму залишків, Q-Q plot, проведіть тест Шапіро-Вілکا.

### Покрокова інструкція

```
from scipy import stats
residuals = y_obs - y_pred

stat, p_value = stats.shapiro(residuals)
print(f'Shapiro-Wilk: p-value={p_value:.4f}')
if p_value > 0.05:
    print('Залишки нормально розподілені')
```

### Завдання 4 (індивідуальне). Звіт

Підготуйте повний звіт: вступ, модель, результати, аналіз чутливості, висновки (5-7 слайдів).

### Критерії оцінювання

Максимум 100 балів: Завдання 1 – 20 б.; Завдання 2 – 20 б.; Завдання 3 – 20 б.;  
Завдання 4 – 30 б.; Оформлення – 10 б.

### Контрольні питання:

1. Що таке валідація моделі?
2. Як інтерпретувати  $R^2$ ?
3. Що показує тест Шапіро-Вілکا?

## ВИСНОВКИ

Навчальний курс «Математичне моделювання інформаційних процесів з використанням Python» охоплює широке коло тем, пов'язаних із сучасними методами математичного моделювання складних соціальних та інформаційних процесів.

У лекційній частині курсу студенти ознайомилися з основами математичного моделювання, класичними моделями типу Лотки-Вольтерри, методами чисельного розв'язання диференціальних рівнянь, стохастичними процесами та різними схемами апроксимації – усередненням, дифузійною апроксимацією, пуассонівською апроксимацією та апроксимацією Леві.

Лабораторні роботи дозволили студентам отримати практичні навички програмування на Python, роботи з бібліотеками NumPy, SciPy та Matplotlib, реалізації чисельних методів та візуалізації результатів моделювання.

Особливу увагу приділено моделі інформаційної боротьби для  $n$  типів інформаційних загроз, яка є актуальною в контексті сучасних викликів інформаційної безпеки. Студенти навчилися враховувати випадкові фактори – дифузійні процеси та стрибки, що дозволяє більш адекватно моделювати реальні інформаційні процеси.

Матеріали курсу базуються на дисертаційному дослідженні «Математичне та комп'ютерне моделювання багатовимірних процесів інформаційної боротьби з використанням схем апроксимації та випадкових збурень» і відображають сучасний рівень наукових досліджень у цій галузі.

## РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Korolyuk V.S. Stochastic Systems in Merging Phase Space / V.S. Korolyuk, N. Limnios // World Scientific, 2005. – 330 с.
2. Papanicolaou G., Stroock D., Varadhan S.R.S., Martingale approach to some limit theorems, Duke turbulence conference (Durham, NC, April 23-25, 1976), Duke University Mathematics Series III, New York: Duke University, 1977, 120 p.
3. Chabanyuk Y., Nikitin A., Khimka U. Asymptotic analyzes for complex evolutionary systems with Markov and Semi-Markov switching using approximation schemes. Wiley-ISTE. 2020 240 p.
4. Lotka A. J. Elements of Physical Biology. – Baltimore: Williams & Wilkins Company, 1925.
5. Volterra V. Fluctuations in the Abundance of a Species considered Mathematically // Nature. – 1926. – Vol. 118. – P. 558-560.
6. Kloeden P. E., Platen E. Numerical Solution of Stochastic Differential Equations. – Berlin: Springer, 1992.
7. Коваленко І. Н., Кузнецов Н. Ю., Шуренков В. М. Випадкові процеси: Довідник. – К.: Наукова думка, 1983. – 366 с.
8. McKinney W. Python for Data Analysis. – O'Reilly Media, 2022.
9. VanderPlas J. Python Data Science Handbook. – O'Reilly Media, 2016.
10. Свешніков А. А. Прикладні методи теорії випадкових функцій. – М.: Наука, 1968. – 464 с.
11. Красюк Б. В. Математичне та комп'ютерне моделювання багатовимірних процесів інформаційної боротьби з використанням схем апроксимації та випадкових збурень: дисертація. – К.: КНУ ім. Тараса Шевченка, 2024. – 134 с.
12. Документація NumPy: <https://numpy.org/doc/>
13. Документація SciPy: <https://docs.scipy.org/doc/scipy/>
14. Документація Matplotlib: <https://matplotlib.org/stable/contents.html>

## About ERASMUS+ Jean Monnet Actions



Jean Monnet Programme has transformed into Jean Monnet Actions under ERASMUS+ Programme since 2014.

The **Jean Monnet actions** offer opportunities in the field of higher education and in other fields of education and training. The Jean Monnet actions contribute to spread knowledge about the European Union integration matters. The following actions are supported:

- Jean Monnet Actions in the field of higher education
- Jean Monnet Actions in other fields of education and training
- Jean Monnet policy debate (higher education and other fields of education and training)

These actions will be implemented by **the European Education and Culture Executive Agency (EACEA)**.

**The Jean Monnet Actions** in the field of Higher Education supports teaching and research in the field of European Union studies worldwide.

European Union studies refers to the teaching, learning and research about the European Union, its history, aims, structures, functions and/or its policies.

The Jean Monnet actions also strive to function as a vector of public diplomacy towards third countries, promoting EU values and enhancing the visibility of what the European Union stands for and what it intends to achieve.

**The Jean Monnet “Teaching and Research”** actions will:

- promote excellence in teaching and research in the field of European Union studies worldwide;
- foster the dialogue between the academic world and society, including local, regional, state and EU level policy-makers, civil servants, civil society actors, representatives of the different levels of education and of the media;
- generate knowledge and insights in support of EU policy-making and strengthen the role of the EU within Europe and in a globalised world;
- reach out to a wider public and spread knowledge about the EU to the wider society (beyond academia and specialised audiences) bringing the EU closer to the public.

The actions also strive to function as a vector for public diplomacy towards third countries not associated to the Programme, promoting EU values and enhancing the visibility of what the European Union actually stands for and what it intends to achieve.

**The Jean Monnet “Teaching and Research”** must take one of the following forms: Modules, Chairs, Centres of Excellence

- **Modules** are short teaching programmes or courses in the field of European Union studies at offered at a higher education institution. Each Module has a minimum duration of 40 teaching hours per academic year for a duration of three years. Modules may concentrate on one particular discipline in European studies or be multidisciplinary in approach and therefore call upon the academic input of several professors and experts. They can also take the form of short specialised or summer programmes.
- **Chairs** are teaching posts with a specialisation in European Union studies (as described above) for university professors for a duration of three years. A Jean Monnet Chair is held by only one professor, who provides the minimum of 90 teaching hours per academic year. The Chair may also have a team to support and enhance the activities of the Chair, including the provision of additional teaching hours.
- **Jean Monnet Centres of Excellence** are focal points of competence and knowledge on European Union subjects. They should gather the expertise and competences of high-level experts aiming to at develop synergies between the various disciplines and resources in European studies (as described above) as well as at creating joint transnational activities, they also ensure openness to civil society. Jean Monnet Centres of Excellence have a major role in reaching out to students from faculties not normally dealing with European Union issues as well as to policy makers, civil servants, organised civil society and the general public at large.

#### **Sources:**

Jean Monnet Actions: <https://erasmus-plus.ec.europa.eu/programme-guide/part-b/jean-monnet-actions>

Jean Monnet actions in the field of higher education: <https://erasmus-plus.ec.europa.eu/programme-guide/part-b/jean-monnet-actions/higher-education>

#### **More information:**

Erasmus+ (EU programme for education, training, youth and sport) (2021-2027): [https://www.eacea.ec.europa.eu/grants/2021-2027/erasmus\\_en](https://www.eacea.ec.europa.eu/grants/2021-2027/erasmus_en)

Erasmus+ Programme Guide: <https://erasmus-plus.ec.europa.eu/erasmus-programme-guide>

Jean Monnet Actions: <https://erasmus-plus.ec.europa.eu/programme-guide/part-b/jean-monnet-actions>

Jean Monnet actions in the field of higher education: <https://erasmus-plus.ec.europa.eu/programme-guide/part-b/jean-monnet-actions/higher-education>

Jean Monnet Activities - Database from 1995 – 2021: [https://www.eacea.ec.europa.eu/grants/2021-2027/erasmus/jean-monnet-activities-database\\_en](https://www.eacea.ec.europa.eu/grants/2021-2027/erasmus/jean-monnet-activities-database_en)



*Навчальне видання*

**Анатолій Володимирович НІКІТІН**

**Богдан Віталійович КРАСЮК**

**МОДЕЛЮВАННЯ ТА АНАЛІЗ ПРОЦЕСІВ  
ІНФОРМАЦІЙНОЇ ВІЙНИ ЯК ІНСТРУМЕНТ ЗАХИСТУ ДЕМОКРАТІЇ  
В ЄС І ПРОТИДІЇ ЗАГРОЗАМ**

*Дидактичні матеріали*

*(«Математичне та комп'ютерне моделювання процесів інформаційної боротьби»  
та «Математичне моделювання інформаційних процесів з використанням Python»)*

**Комп'ютерна верстка** *Наталії Крушинської*

Формат 60x84/8. Ум. друк. арк. 13,72. Наклад 100 прим. Зам. № 8-26.  
Папір офсетний. Друк цифровий. Гарнітура «Cambria».

Оригінал-макет виготовлено у видавництві  
Національного університету «Острозька академія»,  
Україна, 35800, Рівненська обл., м. Острог, вул. Семінарська, 2.  
Свідоцтво суб'єкта видавничої справи РВ № 1 від 8 серпня 2000 року.