

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра математики
факультету інформатики

Курсова робота
за спеціальністю 113 Прикладна математика
освітня програма «Прикладна математика»

**ПРОГНОЗУВАННЯ ВИПАДКОВИХ ПРОЦЕСІВ В УМОВАХ
НЕВИЗНАЧЕНОСТІ ТА ТЕОРІЯ ІГОР**

Керівник курсової роботи
доц. Щестюк Н.Ю.

(підпис)

Виконав студент
Бойченко В.М.
“17” травня 2021 р.

Київ 2020

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ
Зав.кафедри інформатики,
проф., д.ф.-м.н.
_____ М. М. Глибовець
(підпис)
„_____” _____ 2021 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

студенту Бойченко Вікторії Миколаївні факультету інформатики 3 курсу
ТЕМА Прогнозування випадкових процесів в умовах невизначеності та теорія ігор
Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Вступ

1 Теоретичні відомості зі стаціонарних процесів

2 Теоретичні відомості з теорії ігор

3 Постановка задачі

4 Розробка програми

Висновки

Список літератури

Додатки

Дата видачі „23” жовтня 2020 р. Керівник _____
(підпис)

Завдання отримав _____
(підпис)

Тема: Прогнозування випадкових процесів в умовах невизначеності та теорія ігор

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	23.10.2020	
2.	Огляд технічної літератури за темою роботи.	12.11.2020	
3.	Опрацювання матеріалів	6.01.2021	
4.	Погодження плану робіт з керівником	1.03.2021	
5.	Розв'язання поставленої задачі, аналіз проблеми	5.04.2021	
6.	Розробка комп'ютерної реалізації алгоритму	19.04.2021	
7.	Виконання порівняльного аналізу результатів прогнозування, отриманих за допомогою ігрового підходу.	22.04.2021	
8.	Створення слайдів та написання доповіді	9.05.2021	
9.	Остаточне оформлення роботи	17.05.2021	
10.	Захист курсової роботи	20.05.2021	

Студент: Бойченко В.М.

Керівник: Щестюк Н.Ю.

“23” жовтня 2020 р

ВСТУП

У всі часи тематика прогнозування цікавила людей. Зокрема це стосується аналізу отриманих даних з економічних процесів, цін та попиту на енергетику, фінансового ринку, податкових доходів від продажів, акцій, потенціалу ринку, курсу валют і т.д. Тож це залишається актуальним і досі.

Багато вчених займалися дослідженням прогнозування та розробкою методів для цього. Існує велика кількість літератури, зокрема написані цілі підручники на ці теми. Серед відомих науковців Бокс Дж., Армстронг Дж.С., Дібб С., Саркісян С.А.. Методи дослідження інтерполяції, екстраполяції та фільтрації стаціонарних процесів спочатку запропонував Колмогоров А.Н. Далі його теорію розвинули Вінер Н., Яглом А.М., Розанов Ю.А. Також існують спеціальні пакети та бібліотеки в Python, R (наприклад, Scikit-learn, XGBoost, Keras).

Часто трапляються ситуації, коли ми не довіряємо чи підозрюємо дані на викривленість. Вважаємо, що структура залежності даних викривлена. У такому випадку ми вважаємо, що працюємо в умовах деякої невизначеності. Тоді доцільно застосувати ігровий підхід до прогнозування. Цим теж вже активно займалися. Відправною точкою для розвитку рішень цієї проблеми стала стаття Гренандера У. [1], який вперше застосував мінімаксний підхід до задачі екстраполяції стаціонарних процесів. Пізніше цей метод було поширено на випадок дискретного випадкового процесу в Моклячука М.П. [2], а також на випадок полів представлено в роботі Щестюк Н.Ю. [3], багатовимірних процесів у Моклячук М.П. та Масютки А.Ю. [4].

Цими авторами було зроблено теоретичне обґрунтування та надано приклади використання. Але для застосування теорії ігор до задач оцінювання випадкових процесів залишилась проблема складання алгоритму та його автоматизації. Виходячи з тенденцій та історії розвитку методів прогнозування за мету даної роботи були поставлені розробка нового алгоритму прогнозування, його програмне втілення та порівняння його точності на основі отриманих даних.

Мета роботи зумовила наступне наукове завдання:

- опанувати термінологію і основні положення як теорії стаціонарних процесів, так і теорії ігор
- вивчити теорію Гренандера розв'язка гри, в якій стратегіями першого гравця є спектральні функції стаціонарного процесу x_t , про який відома лише його дисперсія
- розглянути приклади застосування теорії Гренандера для оцінювання функціоналів від випадкових процесів з комп'ютерною реалізацією.

Це наукове завдання було виконано в роботі, структура якої складається зі вступу, чотирьох розділів, висновку та додатків.

У першому та другому розділі розглянуто термінологію стосовно стаціонарних процесів. Також представлено відомі методи прогнозування часових рядів та теоретичні відомості про теорію ігор.

У третьому розділі представлено постановку задачі в ігровій формі, а також розглянуто підхід Гренандера. Наостанок показано ілюстративні приклади розв'язання задачі для 2-х та 3-х елементів у функціоналі для прогнозування на 1 та 2 елементи в майбутньому.

У четвертому розділі показано метод комп'ютерної реалізації задачі на мові Python, застосовано його до реальних даних та проведено аналіз на достовірність та точність прогнозу.

Унаслідок створено програмне представлення розв'язання задачі в ігровій формі, яка призначена для прогнозування реальних часових рядів.

Робота буде корисна для представників різних галузей. Задачу такого роду можна зустріти при кодуванні та обробці сигналів у радіо- та гідролокації, розпізнаванні образів мовних сигналів та зображень, прогнозуванні інформації в умовах фінансового ринку.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1: СТАЦІОНАРНІ ПРОЦЕСИ ТА КЛАСИЧНІ МЕТОДИ ПРОГНОЗУВАННЯ	7
1.1 СТАЦІОНАРНІ ПРОЦЕСИ	7
1.2 ВІДОМІ МЕТОДИ ПРОГНОЗУВАННЯ	8
РОЗДІЛ 2: ДЕЯКІ ВІДОМОСТІ З ТЕОРІЇ ІГОР	11
РОЗДІЛ 3: ІГРОВИЙ ПІДХІД ДО ЗАДАЧ ПРОГНОЗУВАННЯ	14
3.1 ПОСТАНОВКА ЗАДАЧІ	14
3.2 ПІДХІД ГРЕНАНДЕРА ДО ПРОГНОЗУВАННЯ НЕПЕРЕРВНИХ ПРОЦЕСІВ	16
3.3 ІЛЮСТРАТИВНІ ПРИКЛАДИ ДЛЯ ФУНКЦІОНАЛА З 2-Х ТА 3-Х ЗНАЧЕНЬ	17
РОЗДІЛ 4: РЕАЛІЗАЦІЯ ТА ЗАСТОСУВАННЯ ДО РЕАЛЬНИХ ДАНИХ.....	24
ВИСНОВКИ	26
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	27
ДОДАТКИ	29

РОЗДІЛ 1: Стаціонарні процеси та класичні методи прогнозування

1.1 Стаціонарні процеси

Випадковим процесом $X(t)$ називається процес, значення якого при довільному фіксованому $t = t_0$ є випадковою величиною [5].

Часовий ряд — реалізація випадкового процесу, яка реально спостерігається.

Стаціонарність:

а) У вузькому розумінні — з зсувом часового ряду числові характеристики не змінюються.

б) У широкому розумінні:

1. $E(\xi(t)) = \text{const}$ (немає тренду)

2. $\text{cov}(\xi(t_1), \xi(t_2)) = \text{cov}(\xi(t), \xi(t+k)) = R(k)$ — кореляційна функція залежить лише від різниці $t_1 - t_2$ (звичайно, вважається, що $E|\xi(t)|^2 < \infty$) [6]

Наслідок:

$\text{cov}(\xi(t_1), \xi(t_2)) = D(t) = R(0) = \text{const}$ (стабільна дисперсія)

Білий шум:

$$E(\varepsilon_t) = 0$$

$$E(\varepsilon_t, \varepsilon_{t-k}) = R(k) = 0, k \neq 0$$

$$\text{Var}(\varepsilon_t) = \sigma^2$$

$$\text{cov}(\varepsilon_t, \varepsilon_{t-k}) = \begin{cases} 0, & k \neq 0 \\ \sigma^2, & k = 0 \end{cases}$$

Випадковий процес $X(t)$ називається *стаціонарним випадковим процесом*, якщо всі його ймовірнісні характеристики не змінюються з плином часу (наприклад, розподіл ймовірностей величини $X(t)$ для всіх t є одним й тим самим, сумісний розподіл ймовірностей величин $X(t_1)$ та $X(t_2)$ залежить лише від довжини проміжку часу $t_2 - t_1$, тобто розподіл пар величин $\{X(t_1), X(t_2)\}$ та $\{X(t_1 + s), X(t_2 + s)\}$ однакові для довільних t_2, t_1 та s) [7].

Приклади випадкових процесів — криві стаціонарного процесу (Рисунок 1), і нестаціонарного (Рисунок 2) [5].

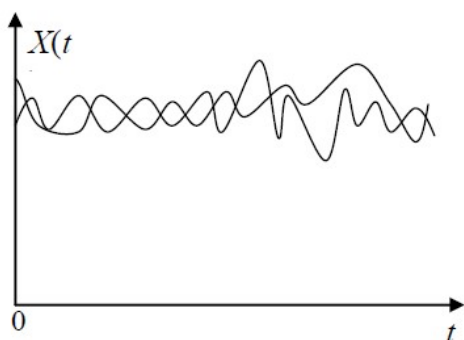


Рисунок 1

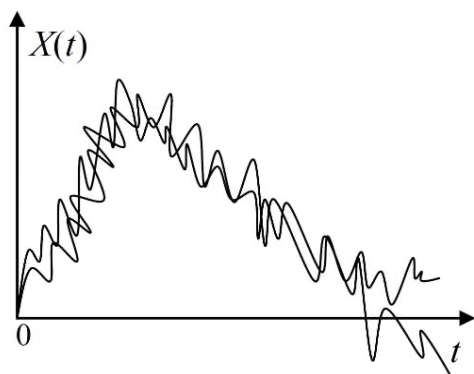


Рисунок 2

Прикладами стаціонарних випадкових процесів є коливання напруги в електромережі, випадкові шуми в радіоприймачі, процес хитаючи корабля.

1.2 Відомі методи прогнозування

Існує понад 100 методів та моделей, що використовуються в цілях прогнозування. Залежно від умов та мети дослідження нам потрібні різні варіанти, оскільки в кожного свої особливості. Вони розрізняються також за науковою обґрунтованістю і призначенням. Ми візьмемо до розгляду моделі прогнозування часових рядів та їх розподіл на статистичні та структурні розглянутий в роботі Jingfei Yang [8].

Зокрема, вже було проведено дослідження методів прогнозування часових рядів та зібрано в сукупність їх для класифікації в роботі Чучуєвою І.К. [9]. У статистичних моделях присутня функціональна залежність між майбутніми та реальними

значеннями, а також зовнішніми факторами задано аналітично. До статистичних відносять регресійні, авторегресійні та моделі експоненційного згладжування.

Для задач, де потрібно дослідити відношення між двома та більшою кількістю змінних використовується регресійний аналіз. Він набув широкого розповсюдження, враховуючи задачі прогнозування та управління. Принцип полягає в тому, щоб встановити залежність між початковою змінною та множиною зовнішніх факторів (регресорів). Серед способів визначення коефіцієнтів є метод найменших квадратів чи метод максимальної правдоподібності. Серед варіантів є лінійна, множинна та нелінійна регресійні моделі.

Авторегресійна модель (Autoregressive model, AR) — модель часових рядів, в якій значення часового ряду в даний момент лінійно залежать від попередніх значень цього ж ряду. Авторегресійний процес порядку p визначається наступним чином [10]:

$$X_t = c + \sum_{i=1}^p \varphi_i X_{t-i} + \varepsilon_t$$

де $\varphi_i, i = \overline{1, p}$ є параметрами моделі, c є сталою, а випадкова величина ε_t є білим шумом.

Ковзне середнє (Moving Average, MA) — функція, значення якої в кожній точці визначене, як середнє значення початкової функції за попередній період.

$$X_t = \mu + \sum_{i=0}^q \varphi_i \varepsilon_{t-i} + \varepsilon_t$$

де $\varphi_i, i = \overline{1, q}$ є параметрами моделі, μ є математичним сподіванням X_t (яке часто є 0), а випадкова величина $\varepsilon_{t-i}, \varepsilon_t$ є білим шумом.

Часто для досягнення більшої точності досить зручно об'єднати кілька моделей. Зокрема, так була отримана ARMA(q,p), що містить собі фільтр у вигляді ковзкого середнього порядку q та авторегресію фільтрованих значень процесу порядку p [9].

ARIMA (модель Бокса-Дженкінса) – є розширенням ARMA для нестационарних часових рядів, які можна зробити стаціонарними при взятті різниці деякого порядку від початкового часового ряду [11].

Експоненційне згладжування містить в собі принцип, що при обробці всіх попередніх значень ряду враховується старіння інформації відносно того як відхиляємось від прогнозного проміжку. Таким чином, останні спостереження мають найбільший вплив на прогноз [12].

У структурних задання відбувається структурно та до них належать нейромережні моделі, на основі ланцюгів Маркова та класифікаційно-регресійних деревах.

Нейронна мережа - це послідовність нейронів, з'єднаних між собою синапсами. Синапс – це зв'язок між двома нейронами. У синапсів є один параметр – вага. Завдяки йому вхідна інформація змінюється, коли передається від одного нейрона до іншого. Штучні нейронні мережі являють собою систему з'єднаних і взаємодіючих простих процесорів (штучних нейронів). Кожен процесор подібної мережі має справу тільки із сигналами, які він періодично отримує, і сигналами, які він періодично посиляє іншим процесорам. Це досить потужний інструмент для виконання складних задач при використанні у великій мережі [13].

РОЗДІЛ 2: Деякі відомості з теорії ігор

Теорія ігор – це розділ прикладної математики, що вивчає математичні моделі прийняття рішень за умови деякого конфлікту [14].

Основним завданням теорії ігор є знаходження оптимальної стратегії поведінки в даній конфліктній ситуації для кожного з гравців. Під стратегією розуміють послідовність певних правил поведінки, що обумовлюють хід кожного гравця в конкретний момент часу гри.

Виграш (програш) гри визначає її функція виграшу. Вона задається деяким аналітичним виразом або таблицею (матрицею).

Антагоністичними або іграми з нульовою сумою називають ігри, у яких виграш одного гравця – це програш іншого, за умови скінченної кількості стратегій.

Розглянемо деяку антагоністичну гру між гравцями A і B зі стратегіями A_i і B_j , де відповідно $i = \overline{1, n}$, $j = \overline{1, n}$ та матрицею стратегій наступного вигляду:

$$M = \begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{pmatrix}$$

Кожен елемент a_{ij} дорівнює величині виграшу гравця A (програшу гравця B). Тоді завданням гравця A є вибір такої стратегії, щоб максимізувала величину a_{ij} , а гравця B – такої, щоб її мінімізувала. Такі ігри називаються *матричними* [15].

Тобто перший гравець намагатиметься спочатку вибрати мінімальне значення виграшу по кожній стратегії

$$\alpha_i = \min_j a_{ij}, i = \overline{1, n}$$

А потім серед цих виграшів шукатиме той, який забезпечує максимум:

$$\alpha = \max_i \alpha_i = \max_i \min_j a_{ij}$$

Величину α називають *нижньою чистою ціною* гри (максиміном). Їй відповідає стратегія, дотримання якої першим гравцем при будь-яких стратегіях другого гравця, гарантує для нього виграш, не менший, ніж α . Тобто це гарантований виграш першого гравця за умови будь-яких стратегій другого гравця.

У свою чергу, другий учасник максимізує свій програш по кожній стратегії:

$$\beta_i = \min_j a_{ij}, j = \overline{1, m}$$

А потім серед усіх стратегій вибирає ту, при якій програш найменший:

$$\beta = \min_i \beta_i = \min_i \max_j a_{ij}$$

Величину β називають *верхньою чистою ціною* гри (мінімаксом). Їй відповідає мінімаксна стратегія другого гравця, дотримання якої за умови будь-якої стратегії першого гравця, забезпечує йому програш, не більший, ніж β . Тобто це гарантований програш другого гравця за умови будь-яких стратегій першого гравця.

Оптимальним розв'язком цієї гри є *сідлова точка*, тобто елемент a_{kl} матриці задовольняє умову $a_{il} \leq a_{kl} \leq a_{kj}$ для довільних i та j .

Якщо $\beta = \alpha$, тобто верхня ціна гри дорівнює нижній ціні гри, то відповідні чисті стратегії називають *оптимальними*, а про гру кажуть, що вона має *сідлову точку*.

У більшості матричних ігор відсутні сідлові точки, тобто не виконується умова:

$$\max_i \min_j a_{ij} \leq \min_j \max_i a_{ij}$$

У таких випадках використовуються *змішані стратегії*, які є схемою випадкового вибору стратегій.

Змішаною стратегією S_A першого гравця називають застосування чистих стратегій $A_1, A_2, \dots, A_n$ з ймовірностями $p_1, p_2, \dots, p_n$, причому

$$\sum_{i=1}^n p_i = 1$$

Змішану стратегію S_A гравця А записують у вигляді матриці:

$$S_A = \begin{pmatrix} A_1 & A_2 & \cdots & A_n \\ p_1 & p_2 & \cdots & p_m \end{pmatrix}$$

де $p_i \geq 0, i = \overline{1, n}, p_i$ – ймовірність використання чистої стратегії A_i .

Аналогічно стратегію S_B гравця В записують у вигляді матриці:

$$S_A = \begin{pmatrix} B_1 & B_2 & \cdots & B_n \\ q_1 & q_2 & \cdots & q_m \end{pmatrix},$$

де $q_j \geq 0, j = \overline{1, m}, \sum_{j=1}^m q_j = 1, q_j$ – ймовірність використання чистої стратегії B_j [16].

РОЗДІЛ 3: Ігровий підхід до задач прогнозування

3.1 Постановка задачі

Ігри – це строго визначені математичні об'єкти. Гра утворюється гравцями, набором стратегій для кожного гравця та визначення виграшів для кожної комбінації стратегій.

Відносно нещодавно було запропоновано застосувати ігровий підхід до моделювання фінансових процесів, розглядаючи там відповідних суб'єктів та аналізуючи їхню поведінку, прогнозуючи майбутні ситуації.

На відміну від теорії ймовірностей, теорія ігор у фінансовій математиці розглядає не припущення про існування деякого механізму випадковості, а робить припущення, що можна робити ставки на майбутній хід подій.

Серед суб'єктів фінансового ринку теорія ігор розглядає двох гравців: «фінансового диявола» з одного боку та спекулянта (Скептика) з іншого. «Фінансового диявола» вважаємо таким, чиї дії не можна передбачити (можна сказати, аналог механізму випадковості в теорії ймовірностей), в його інтересах не дати виграти нескінченно велику суму іншому суб'єкту – спекулянту. Отже, спекулянт прагне отримати якомога більшу суму, проте він досить обережний у своїх діях, тому ніколи не банкрутує.

Складніші моделі розглядають Скептика, Передбачувача та Реальність, їхня взаємодія, відповідно є складнішою та описується складнішими формулами. Кожен із суб'єктів має свої власні інтереси та намагається певними діями їх досягти. Спробуємо побачити у їх діях деякі закономірності та математично їх описати.

Розглянемо модель фінансової гри з «фінансовим дияволом» (гравець X) та Скептиком (гравець P). Нехай фінансовий ринок працює дискретно, у ньому є деякий фінансовий актив і ми розглядаємо його ціну, що змінюється в кожному конкретний момент часу. Позначимо цю ціну S_n , де n позначає відповідний момент часу. Вважаємо, що цей процес контролює гравець X . Вперше задача подібного класу була поставлена У.Гренадером у термінах теорії ігор [1]. Її розв'язання можна інтерпретувати як розв'язання задачі прогнозування стаціонарного випадкового

процесу, що спостерігається без шуму, точне значення спектральної щільності невідомо, але відомо дисперсію сигналу, що і визначає в даному випадку параметричний клас спектральних щільностей.

Гравець P (Скептик) спостерігає за процесом до даного моменту часу $t = 0$ і прогнозує

$$A_N \xi = \sum_{j=0}^N a(j) \xi(j)$$

де $\xi(j)$ - послідовність значень, вказаних фінансовим дияволом,

Простір стратегій гравця X - клас спектральних щільностей Ξ , що містить всі стаціонарні випадкові процеси ξ з $E\xi = 0$ та $D\xi \leq P$.

Простір стратегій гравця P – клас Λ , що містить всі лінійні оцінки $\hat{A}_N \xi$.

Функція гри - $\Delta(\xi, \hat{A}_N) = E|A_N \xi - \hat{A}_N \xi|^2$ - середньоквадратична похибка оцінки $\hat{A}_N \xi$ функціонала $A_N \xi$.

За допомогою результатів статей [2] та [4] можемо переконатись, що справедлива наступна теорема для корельованої послідовності:

Теорема :

Функція $\Delta(\xi, \hat{A}_N)$ на множині $\Xi \times L$ має сідлову точку. При цьому:

$$\min_{\hat{A}_N \in \Lambda} \max_{\xi \in \Xi} \Delta(\xi, \hat{A}_N) = \max_{\xi \in \Xi} \min_{\hat{A}_N \in \Lambda} \Delta(\xi, \hat{A}_N) = P v_N^2.$$

Найменш сприятливою в класі Ξ для оптимального оцінювання функціонала $\hat{A}_N \xi$ стохастичною послідовністю є послідовність одностороннього рухомого середнього порядку $N + 1$, що має вигляд

$$\xi(m) = \sum_{u=m-N}^m \varphi(m-u) \eta(u).$$

Тут v_N^2 - найбільше власне значення, а $\{\varphi(u): u = 0, 1, \dots, N\}$ - власний вектор, що відповідає v_N^2 , самоспряженого компактного оператора Q_N у просторі C^{N+1} , що визначається матрицею з елементами :

$$Q_{N(p,q)} = \sum_{u=0}^{\min(N-p, N-q)} a(p+u) \overline{a(q+u)} \quad , \quad p, q = \overline{0, N}$$

$\eta(u)$ - стаціонарна послідовність з ортогональними значеннями (білий шум).

Оператор Q_N - самоспряжений (його матриця ермітова) неперервний оператор. Його можна подати у вигляді $Q_N = A_N \cdot A_N^*$, де оператор A_N визначається матрицею з елементами

$$a_{pq} = \begin{cases} a_{p+q}, & p+q \leq N, \\ 0, & p+q > N. \end{cases}$$

Тому оператор Q_N має дійсні невід'ємні власні значення.

3.2 Підхід Гренандера до прогнозування неперервних процесів

Двоє людей, X і P , грають у наступну гру. X керує випадковим механізмом, що створює стаціонарний стохастичний процес $x = \{x_t\}$. Дано середнє значення $Ex(t) = 0$ і дисперсію $Ex^2(t) = 1$, але форму спектра процесу можна повністю визначити за допомогою X . Гравець P дотримується значень $x(t)$ для $t < 0$ і хоче передбачити значення :

$$z = \int_0^1 a(t)x_t dt = Ax$$

де $a(t)$ – дійсна і неперервна функція t в $(0, 1)$. Для цієї мети P має всі лінійні предиктори, утворені зі стохастичних змінних x_t , $t \leq 0$. Позначимо вибраний предиктор через p , а передбачуване значення – px .

Функція окупності гри становить $(Ax - px)^2$. X хоче максимально збільшити кількість $\|Ax - px\|^2 = E(Ax - px)^2$ і P намагається мінімізувати той самий вираз.

Теорема: Гра з нульовою сумою для двох осіб, що визначена вище за допомогою функції виплат $(Ax - px)^2$ – визначена. Рішення гри подано в чистих стратегіях:

$$\begin{cases} x_t = \int_{-\infty}^t \gamma(t-u) d\xi(u) \\ px = \int_0^1 a(t) x_t^* dt \end{cases}$$

де $\gamma(t)$ – власна функція, що відповідає найбільшому власному значенню v симетричного ядра

$$K(x, y) = \int_0^{\min(1-x, 1-y)} a(x+u)a(y+u)du$$

Ціною гри є значення v . Коли v є простим власним значенням, то рішення гри унікальне. Задача зводиться до розв’язання рівнянь частинних похідних [1].

3.3 Ілюстративні приклади для функціонала з 2-х та 3-х значень

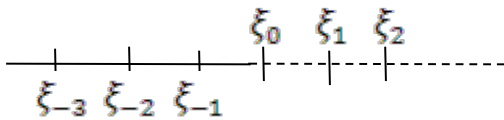
Для початку згенеруємо стаціонарний ряд з $M\xi = 0, D\xi \leq 2,5$ (див. Додаток А).

0	-0.768651
1	-0.006537
2	1.658654
3	-2.815931
4	-1.963251
5	-0.197305

6	1.525510
7	-1.175910
8	2.321404
9	1.414653

Задача для функціонала з 2-х значень

Нехай дано певний набір значень, взятих у рівномірно віддалені моменти часу до моменту t_0 .



Дано:

Дано випадковий процес з $M\xi = 0, D\xi \leq 2,5$.

Функціонал: $A_1\xi = 2\xi(0) + 3\xi(1)$.

Знайти : $\widehat{A_N\xi}$ - оцінку.

Розв'язання:

Отже, $N = 1$, коефіцієнти: $a_0 = 2, a_1 = 3$

Матриця оператора визначається так:

$$Q_{N(p,q)} = \sum_{u=0}^{\min(N-p, N-q)} a(p+u) \overline{a(q+u)} , \quad p, q = \overline{0, N}$$

Її можна подати так:

$$a_{pq} = \begin{cases} a(p+q), & p+q \leq N, \\ 0, & p+q > N. \end{cases}$$

p	q	$p+Q$	
0	0	0	$\leq N$
0	1	1	$\leq N$
1	0	1	$\leq N$
1	1	2	$> N$

$$a_{00} = a_0, 0 + 0 = 0 \leq N = 1$$

$$a_{10} = a_1, 1 + 0 = 1 \leq 1$$

$$a_{01} = a_1, 0 + 1 = 1 \leq 1$$

$$a_{11} = a_2, 1 + 1 = 2 > 1$$

$$a_0 = 2, a_1 = 3, a_2 = 0$$

Отже, матриця виглядає так:

$$Q = \begin{pmatrix} a_{00} & a_{10} \\ a_{01} & a_{11} \end{pmatrix} = \begin{pmatrix} a_0 & a_1 \\ a_1 & a_2 \end{pmatrix} = \begin{pmatrix} 2 & 3 \\ 3 & 0 \end{pmatrix}$$

Знайдемо власні числа:

$$\lambda_{1,2} = 1 \pm \sqrt{10}$$

Найбільше власне значення

$$\lambda_1 = 1 + \sqrt{10} = v_1^2$$

Знайдемо загальний розв'язок системи:

$$\begin{pmatrix} 1 - \sqrt{10} & 3 \\ 3 & -(1 + \sqrt{10}) \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \vec{0}$$

$$v_1^2 + v_2^2 = P$$

$$\begin{cases} (1 + \sqrt{10})v_1 + 3v_2 = 0 \\ 3v_1 - (1 + \sqrt{10})v_2 = 0 \end{cases}$$

$$v_2 = \frac{(\sqrt{10} - 1)v_1}{3}$$

Для формули $v_1^2 + v_2^2 = 2,5$ з підстановками

$$v_2^2 + \frac{(\sqrt{10} - 1)^2 v_1^2}{9} = 2,5$$

$$v_2^2(9 + 10 - 2\sqrt{10} + 1) = 22,5$$

$$v_2^2 = \frac{22,5}{20 - 2\sqrt{10}} = \frac{11,25}{10 - \sqrt{10}}$$

$$v_2 = \pm \sqrt{\frac{11,25}{10 - \sqrt{10}}} \cong 1,2826$$

Знайдемо власний вектор:

$$\varphi = (\varphi(0), \varphi(1))$$

$$\varphi(0) = \sqrt{\frac{11,25}{10 - \sqrt{10}}} \cong 1,2826$$

$$\begin{aligned}\varphi(1) &= \frac{(\sqrt{10} - 1)\varphi(0)}{3} = \sqrt{\frac{11,25}{10 - \sqrt{10}}} * \frac{(\sqrt{10} - 1)}{3} = \frac{1}{3} \sqrt{\frac{11,25(\sqrt{10} - 1)^2}{10 - \sqrt{10}}} \\ &= \sqrt{\frac{22,5(\sqrt{10} - 1)^2}{2 * 9\sqrt{10}(\sqrt{10} - 1)}} = \sqrt{\frac{2,5(\sqrt{10} - 1)}{2 * \sqrt{10}}} = \frac{1}{2} \sqrt{\frac{5(\sqrt{10} - 1)}{\sqrt{10}}} \\ &= \frac{1}{2} \sqrt{5 - \frac{5}{\sqrt{10}}} \cong 0,9245\end{aligned}$$

Найменш сприятлива стаціонарна послідовність:

$$\xi(j) = \sum_{u=j-N}^j \varphi(j-u)\eta(u), \quad j = 0 \dots N$$

$$\xi(j) = \varphi(0)\eta(j) + \varphi(1)\eta(j-1) = \sqrt{\frac{11,25}{10 - \sqrt{10}}}\eta(j) + \frac{1}{2}\sqrt{5 - \frac{5}{\sqrt{10}}}\eta(j-1)$$

Оптимальна лінійна мінімаксна оцінка:

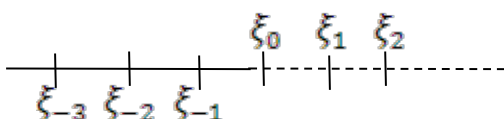
$$\widehat{A}_1\xi = \varphi(1)\eta(-1) = \frac{1}{2}\sqrt{5 - \frac{5}{\sqrt{10}}} * 1.414653 = 1,3078$$

Похибка $\widehat{A}_1\xi$ не перевищує $P * v^2 \cong 2,5 + 2,5\sqrt{10} \sim 10,4057$

Шукані прогнозні значення $seq = [0.32192054, -0.71900874]$ (див. Додаток Б)

Задача для функціонала з 3-х значень

Нехай дано певний набір значень, взятих у рівномірно віддалені моменти часу до моменту t_0 .



Дано:

Дано випадковий процес з $M\xi = 0, D\xi \leq 2,5$.

Функціонал: $A_2\xi = 2\xi(0) + 3\xi(1) + 0.5\xi(2)$.

Знайти : $\widehat{A_2\xi}$ - оцінку.

Розв'язання:

Отже, $N = 2$, коефіцієнти: $a_0 = 2, a_1 = 3, a_2 = 0.5$

Матриця оператора визначається так:

$$Q_{N(p,q)} = \sum_{u=0}^{\min(N-p, N-q)} a(p+u) \overline{a(q+u)}, \quad p, q = \overline{0, N}$$

Її можна подати так:

$$a_{pq} = \begin{cases} a(p+q), & p+q \leq N, \\ 0, & p+q > N. \end{cases}$$

Отже, отримуємо матрицю вигляду:

$$Q = \begin{pmatrix} 2 & 3 & 0.5 \\ 3 & 0.5 & 0 \\ 0.5 & 0 & 0 \end{pmatrix}$$

Знайдемо власні числа:

$$\lambda_1 \sim -1.893$$

$$\lambda_2 \sim 0.015$$

$$\lambda_3 \sim 4.378$$

Найбільше власне значення

$$\lambda_3 = 4.378 = v_1^2$$

Знайдемо загальний розв'язок системи:

$$\begin{pmatrix} 2 - 4.378 & 3 & 0.5 \\ 3 & 0.5 - 4.378 & 0 \\ 0.5 & 0 & -4.378 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \vec{0}$$

Має виконуватись умова, що $v_1^2 + v_2^2 + v_3^2 = P$, де $P = 1,3$ – обмеження дисперсії

$$\begin{pmatrix} -2.378 & 3 & 0.5 \\ 3 & -3.878 & 0 \\ 0.5 & 0 & -4.378 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \vec{0}$$

$$\begin{cases} -2.378 * v_1 + 3 * v_2 + 0.5 * v_3 = 0 \\ 3 * v_1 - 3.878 * v_2 = 0 \\ 0.5 * v_1 - 4.378 * v_3 = 0 \end{cases}$$

$$\begin{cases} -2.378 * v_1 + 3 * v_2 + 0.5 * v_3 = 0 \\ v_2 \cong 0.773 * v_1 \\ v_3 \cong 0.114 v_1 \end{cases}$$

$$\begin{cases} -2.378 * v_1 + 3 * v_2 + 0.5 * v_3 = 0 \\ v_2 \cong 0.773 * v_1 \\ v_3 \cong 0.114 v_1 \end{cases}$$

$$v_1^2 + v_2^2 + v_3^2 = 2.5$$

$$v_1^2 + 0.773^2 v_1^2 + 0.114^2 v_1^2 = 2.5$$

$$v_1^2 + 0.597529 v_1^2 + 0.012996 v_1^2 = 2.5$$

$$1.610525 * v_1^2 = 2.5$$

$$v_1^2 \cong 1.552$$

$$v_1 \cong 1.246$$

Знайдемо власний вектор:

$$\varphi = (\varphi(0), \varphi(1), \varphi(2))$$

$$\varphi(0) = v_1 \cong 1.246$$

$$\varphi(1) = v_2 \cong 0.963$$

$$\varphi(2) = v_3 \cong 0.142$$

Найменш сприятлива стаціонарна послідовність:

$$\xi(j) = \sum_{u=j-N}^j \varphi(j-u) \eta(u), \quad j = 0 \dots N$$

$$\xi(j) = \varphi(0) \eta(j) + \varphi(1) \eta(j-1) + \varphi(2) \eta(j-2)$$

$$\cong 1.246 * \eta(j) + 0.963 * \eta(j-1) + 0.142 * \eta(j-2)$$

(стратегія для фін.диявола)

Оптимальна лінійна мінімаксна оцінка:

$$\widehat{A}_2 \xi = \varphi(1) \eta(-1) + \varphi(2) \eta(-2) = 0.963 * 1.415 + 0.142 * 2.321 = 1.692$$

(відповідь для скептика)

Похибка $\widehat{A}_2\xi$ не перевищує: $P * v^2 = 2,5 * 4,378 = 10,945$

Шукані прогнозні значення $seq = [0.73593535, -0.54754004, 1.95025723]$ (див. в Додатку Б).

РОЗДІЛ 4: Реалізація та застосування до реальних даних

Для створення алгоритму було використано дані від Google Trends, щоб проаналізувати як швидко люди закидують свої цілі поставлені на Новий Рік [17]. Також серед можливих реалізацій прогноз для фінансового ринку. Зокрема існують варіанти прогнозу для цін акцій

Під час створення виникла проблема в обрахунку загального розв'язку системи лінійних рівнянь. Були пошуки алгоритмів для обрахування, спеціальних бібліотек, для збереження змінних. Були спроби за методом обчислення Гауса-Жордано. Також у процесі знайдено спосіб необхідної матриці за допомогою SVD або QR розкладів [18]. Ще був варіант з використання RREF-бібліотеки. Проте такі чисельні методи зводили розв'язок до тривіального (нульового). Часткове обчислення матриці давала хибні результати.

У моєму випадку обрахована для задачі матриця розмірністю $N \times N$ зі системи $(Q - \lambda E) \cdot \vec{x} = \vec{0}$ завжди матиме нескінченну кількість розв'язків для системи [19]. Але використання за умовою задачі обмеження дисперсії зводить кількість розв'язків до одного.

Врешті-решт виникла ідея використати примітивні засоби для вираження змінних методом підстановки. Для цього достатньо було використати засоби SymPy-бібліотеки - через задання кортежу n-символів вектору $\vec{x}$ та використання функції solve(). Однак дозволяє відносно швидко знаходити великі кількості змінних виражаючи через елемент x_0 .

Далі виникли труднощі з заданням значень білого шуму для ілюстративних прикладів. Було використано метод задання гаусового розподілу з математичним сподіванням 0 та заданою дисперсією для n-кількості значень.

Було зроблено тестування на швидкість виконання програми при заданих для функціонала коефіцієнтів та згенерованого стаціонарного ряду зі заданою дисперсією:

$$\forall i = \overline{0, N-1} : a(i) = 1, D(\xi) = 2$$

При заданні кількості елементів $N = 10$ код виконувався протягом 0,5 секунд, $N = 100$ — 40 секунд. За інтерполяційним многочленом Лагранжа для $N = 200$ займає більше 3,5 год. Виходить експоненційне зростання (див. в Додатку В). Тобто, для маленьких чисел обрахування ще не вимагає багато часу, утім, для великих чисел не підходить — потрібна подальша оптимізація. Розроблений програмний код простий для розуміння та реалізації, але не підходить для обробки та прогнозувань на основі великої кількості даних.

Вхідними даними є кількість елементів, сам стаціонарний часовий ряд для η , варіація, коефіцієнти функціонала. Вихідними даними є оцінка функціонала, похибка, та послідовність $\xi(i), i = \overline{0, N}$. Після отримання цього значення необхідно перетворити прогноз на початковий варіант даних (зворотній шлях до стаціонарного ряду). Код працює при $N \geq 1$. При $N = 0$ розв'язання задачі нівелюється за відсутності даних та необхідності прогнозу. Власні значення для створеної матриці завжди будуть дійсними невід'ємними внаслідок властивості самоспряженого компактного оператора [20].

Було перевірено точність роботи методу на основі порівняння з отриманими на вхід даними та з прогнозованими. Тобто, проміжок значень з набору даних був використаний для перевірки точності прогнозування на прикладі останніх двох значень. Для прикладу взяли вже згадані дані від Google Trends, проміжок [100,110] зі стаціонарними значеннями. Прогноз виконували на $N = 2$ кроки, на 111 та 112 елементи. Для коефіцієнтів взяли значення 1, обмеження дисперсії дорівнює 1:

$$\forall i = \overline{0, N} : a(i) = 1, D(\xi) = 1$$

Під час обрахування прогнозу було знайдено похибку 2,247. Після знайденого розв'язку задачі обчислені прогнозні значення були перетворені назад зі стаціонарних у вхідні. Далі їх було порівняно з готовими значеннями та перевірено, що отримані значення знаходяться в межах похибки (див. в Додатку Г).

Висновки

Отже, в результаті проведеної роботи, в якій завдяки вивченні теорії та її аплікування до практики, досліджено різноманітну літературу стосовно випадкових процесів, часових рядів, методів їх опрацювання та отримання прогнозів. Досліджено вже відомі методи для знаходження рішення за різних умов (наявності чи відсутності інформації в задачі).

У процесі роботи над курсовою було розглянуто термінологію та основні положення необхідні для подальшого вивчення теорії стаціонарних процесів та теорії ігор. Особливу увагу в роботі було приділено теорії Гренадера. У результаті опрацьованого матеріалу надалі досліджувалось розв'язання для дискретних випадкових процесів робіт Моклячука.

Було застосовано теорію ігор до задач оцінювання випадкових процесів. Зокрема розв'язано приклади для функціонала з 2-х та 3-х значень на основі згенерованого стаціонарного ряду. Було зроблено алгоритм для реалізації розв'язання цієї задачі, запрограмована на основі мови Python та протестовано на реальних даних. Серед знайдених недоліків це швидкість роботи алгоритма та доступних програмних методів від Python.

Складений алгоритм та програмна реалізація поставленої задачі роблять актуальною дану роботу, оскільки до цього були лише розв'язки теоретичного характеру. У подальшому ці обчислювання можна оптимізувати для більшої швидкодії. До того ж можна розширити для використання у задачах такого класу, наприклад, на двомірний процес, випадкове поле.

Список використаної літератури

- [1] U. Grenander, “A prediction problem in game theory,” *Ark. för Mat.*, vol. 3, no. 4, pp. 371–379, 1957, doi: 10.1007/BF02589429.
- [2] M. P. Moklyachuk, “Estimation of Linear Functionals of a Stationary Stochastic Process and a Two Person Zero Sum Game,” 1981. [Online]. Available: https://scholar.google.com.tw/scholar?cluster=33210472385875531&hl=en&as_sdt=2005.
- [3] Н. Щестюк, “Екстраполяція функціоналів від однорідного випадкового поля, яке спостерігають у півплощині,” *Вісник Львівського університету*, pp. 51–61, 2007, [Online]. Available: <http://ekmair.ukma.edu.ua/handle/123456789/12394>.
- [4] A. M. Mikhail Moklyachuk, “Minimax prediction problem for multidimensional stationary stochastic sequences.pdf,” *Dep. Probab. Theory Math. Stat. Kyiv Natl. Taras Shevchenko Univ.*, vol. 14(30), no. 4, pp. 89–109, 2008, [Online]. Available: <http://mi.mathnet.ru/thsp216>.
- [5] І. В. Новицький and С. А. Ус, *ББК 22.171 я73 Н73 Рекомендовано редакційною радою НГУ як навчальний посібник з дисципліни «Випадкові процеси» для студентів напряму підготовки 6.040303 Системний аналіз (протокол № 9 від 2.09. 2010).* 2011.
- [6] Ю. А. Розанов, *Стационарные случайные процессы*. Наука. Гл. ред. физ.-мат. лит., 1990.
- [7] А. Я. Хинчин, “Теория корреляции стационарных стохастических процессов,” *Успехи математических наук*, vol. 5, pp. 42–51, 1938, [Online]. Available: <http://www.mathnet.ru/rus/agreement>.
- [8] V. Fachbereich, von M. Sc Jingfei Yang, and J. Yang, “Power System Short-term Load Forecasting,” 1974.
- [9] Чучуева И.А., “Модель прогнозирования временных рядов по выборке максимального подобия,” Москва, 2012. [Online]. Available: <https://www.mbureau.ru/sites/default/files/pdf/Chuchueva-Dissertation.pdf>.
- [10] “5 Statistical Methods For Forecasting Quantitative Time Series | Bista INC,” May

- 31, 2016. <https://www.bistasolutions.com/resources/blogs/5-statistical-methods-for-forecasting-quantitative-time-series/>.
- [11] А. С. Перерва and Т. В. Ковалюк, “ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ ПРОГНОЗУВАННЯ СТАЦІОНАРНИХ ТА НЕСТАЦІОНАРНИХ ЧАСОВИХ РЯДІВ,” *Міжнародний науковий журнал Науковий огляд* 3.46, pp. 101–112, 2018, [Online]. Available: <https://naukajournal.org/index.php/naukajournal/article/view/1503>.
- [12] А. А. Грешилов, В. А. Стакун, and А. А. Стакун, *Математические методы построения прогнозов*. М.: Радио и связь, 1997.
- [13] А. О. Долгих and О. Г. Байбуз, “Аналіз методів , моделей та програмних засобів прогнозування часових рядів,” *Открытые информационные и компьютерные интегрированные технологии*, vol. 79, pp. 74–87, 2018.
- [14] Васильев В.А., *Модели экономического обмена и кооперативные игры*. Новосибирск: Изд-во НГУ 96, 1984.
- [15] “Конспект лекції № 11 . Тема №11. ТЕОРІЯ ІГОР.” <https://financial.lnu.edu.ua/wp-content/uploads/2019/09/ME-lektsiia-11.pdf>.
- [16] “Теорія масового обслуговування.” <https://arhivinfo.ru/2-103406.html>.
- [17] M. Chalabi, “How Fast You’ll Abandon Your New Year’s Resolutions | FiveThirtyEight,” Jan. 01, 2015. <https://fivethirtyeight.com/features/how-fast-youll-abandon-your-new-years-resolutions/>.
- [18] S. Van Huffel, J. Vandewalle, and A. Haegemans, “An efficient and reliable algorithm for computing the singular subspace of a matrix, associated with its smallest singular values,” 1987.
- [19] J. Stewart, “Calculus (Early Transcendentals) 3rd Ed.,” 1995. <http://sites.science.oregonstate.edu/math/home/programs/undergrad/CalculusQuestStudyGuides/vcalc/system/system.html>.
- [20] D. Whittaker and J. B. Conway, “A Course in Functional Analysis,” *Math. Gaz.*, vol. 75, no. 474, p. 488, Dec. 1991, doi: 10.2307/3618662.

Додатки

Додаток А

Генерація стаціонарного ряду

```
from random import gauss
from random import seed
import pandas as pd
import numpy as np

np.random.seed(7)

mean = 0
variance = 2.5

n = 10

random_numbers = [gauss(mean, np.sqrt(variance)) for i in range(n)]

noise = {'Values': random_numbers}
df_noise = pd.DataFrame (noise, columns = ['Values'])
df_noise
```

0	-0.768651
1	-0.006537
2	1.658654
3	-2.815931
4	-1.963251
5	-0.197305
6	1.525510
7	-1.175910
8	2.321404
9	1.414653

Додаток Б

1) Розв'язання першого прикладу

```

1 game_approach(arr_N, var, eta) # predicted values of game approach

N = 1
Q_l =
[[-2.16227766  3.          ]
 [ 3.          -4.16227766]]
expr = [3.0*x0 - 4.16227766016838*x1]
results = {x0: 1.38742588672279*x1}
x1
coefs = [1.38742589 1.          ]
x_coef = 2.924950591148529
x_res = 0.9245081354314588
phi = [1.28268652 0.92450814]
u = -1
seq = [ 0.32192054 -0.71900874]
estim = 1.3078585689099542
min = -0.719008738855667 max = 0.32192053734956017
error = (10.405694150420949+0j)

```

2) Розв'язання другого прикладу

```

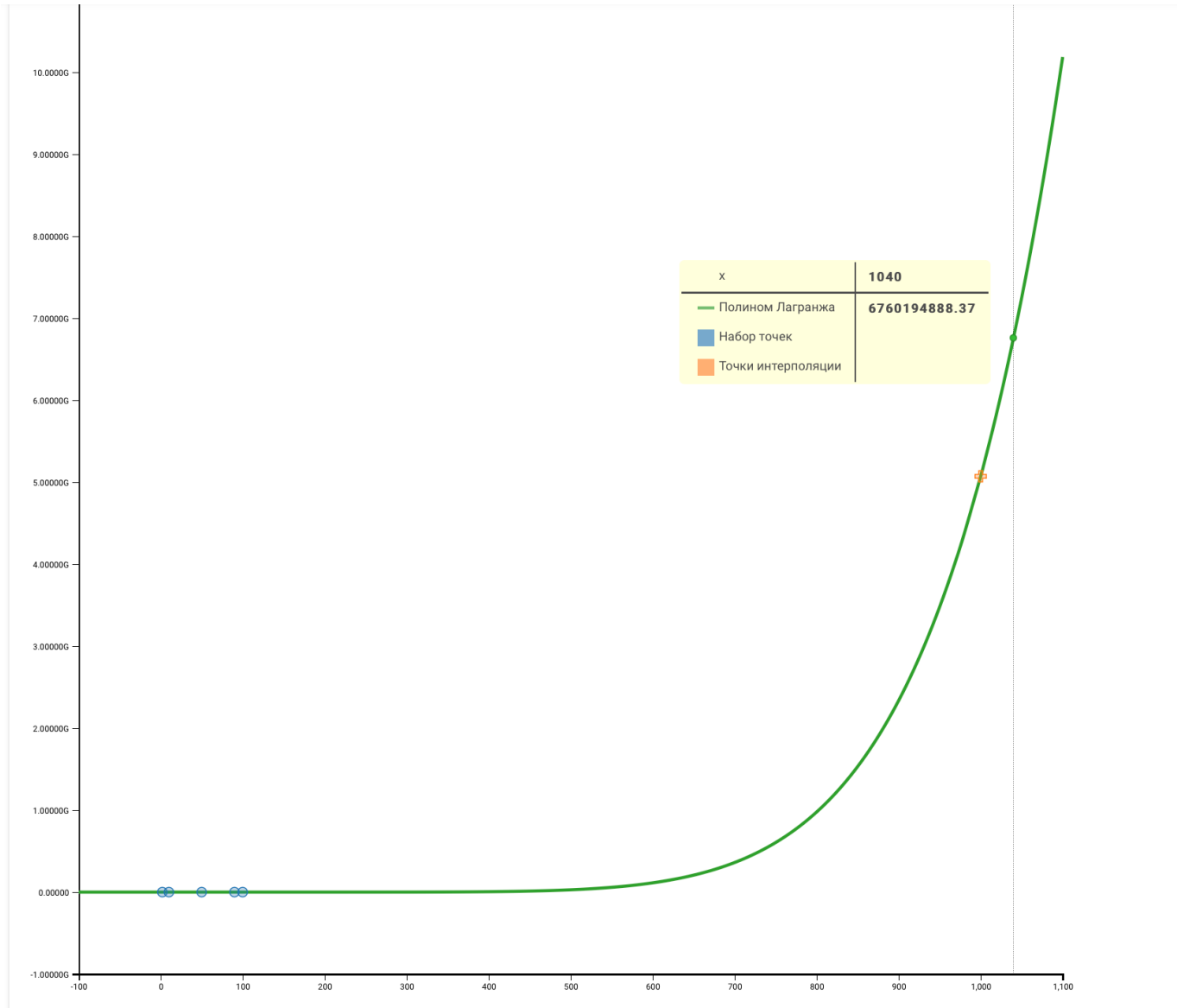
1 game_approach(arr_N, var, eta) # predicted values of game approach

N = 2
Q_l =
[[-2.37793026  3.          0.5          ]
 [ 3.          -3.87793026  0.          ]
 [ 0.5          0.          -4.37793026]]
expr = [3.0*x0 - 3.87793025737385*x1, 0.5*x0 - 4.37793025737385*x2]
results = {x0: 8.7558605147477*x2, x1: 6.77360854912115*x2}
x2
coefs = [8.75586051 6.77360855 1.          ]
x_coef = 123.54686613044503
x_res = 0.14225061003369796
phi = [1.2455265 0.96354995 0.14225061]
u = -2
u = -1
seq = [ 0.73593535 -0.54754004 1.95025723]
estim = 1.6933103027382166
min = -0.5475400401253001 max = 1.9502572309873583
error = (10.944825643434616+0j)

```

Додаток В

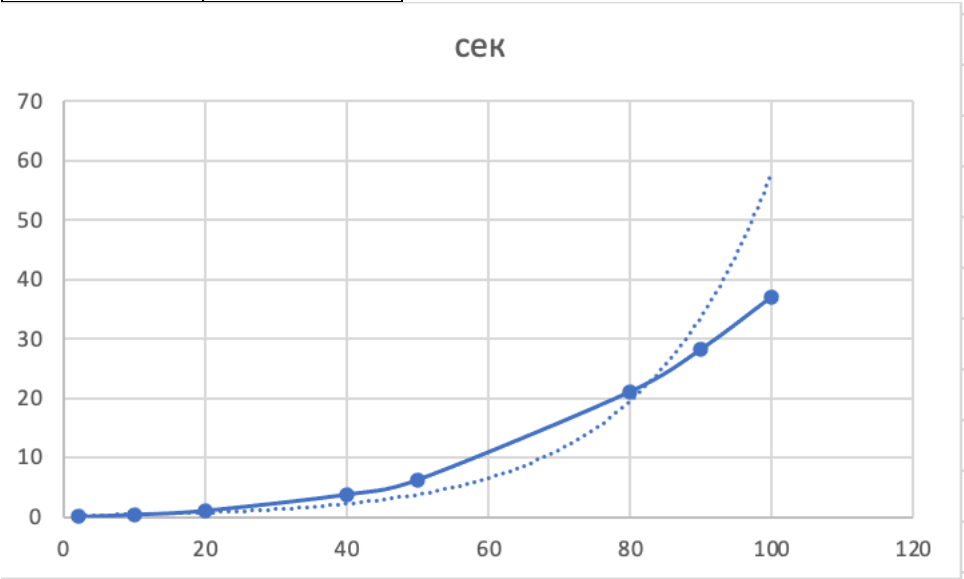
За поліномом Лагранжа



Таблиця з вихідними значеннями, де N – кількість прогнозів, сек – час розв’язання задачі за допомогою коду.

N	сек
2	0,1
10	0,47
20	1,13
40	3,84
50	6,31
80	21,08

90	28,19
100	37



Додаток Г

- 1) Проміжок значень з набору даних, що був використаний для перевірки точності прогнозування на прикладі останніх двох значень.

```
n = 10,
a = 100 b = 112
init df =      diet
0   66.0
1   67.0
2   66.0
3   65.0
4   65.0
5   65.0
6   66.0
7   70.0
8   67.0
9   70.0
10  69.0
11  68.0
var = 0.0009169111726805568
```

- 2) Отримані обрахунки для прогнозних значень

```
len_df = 10
Game approach: start
N = 2
Q_l =
[[-1.2469796  1.          1.          ]
 [ 1.          -1.2469796  0.          ]
 [ 1.          0.          -2.2469796]]
expr = [1.0*x0 - 1.24697960371747*x1, 1.0*x0 - 2.24697960371747*x2]
coefs = [2.2469796  1.80193774 1.          ]
phi = [0.73697623 0.59100905 0.32798528]
seq = [ 0.00043838  0.01656155 -0.0071273 ]
estim = 0.011521035954910601
min = -0.0071273034449705315 max = 0.01656155327569693
error = (2.2469796037174676+0j)
func_res = [ 0.00043838  0.01656155 -0.0071273 ]
Game approach: end
y_9 = 70.03069315233316
y_10 = 71.20016763421961
y_11 = 70.69450657462865
```

- 3) Таблиця порівнянь прогнозних значень з справжніми значеннями, їх різницею та чи знаходяться вони в межах похибки:

Error = 2.2469796037174676

	Prediction	Real	Difference	Check
0	70.030693	70.0	0.030693	True
1	71.168962	69.0	2.168962	True
2	69.502862	68.0	1.502862	True