

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

ВИПУСКНА РОБОТА

на тему:

**«Прогнозування фінансового ринку за допомогою методів
математичного аналізу»**

**Завідувач
випускаючої кафедри**

Довбиш А.С.

Керівник роботи

Шутильєва О.В.

Студент гр. Ін-71

Голодний В.В.

Суми 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних наук
Затверджую _____
Зав. кафедрою Довбиш А.С.
“ _____ ” _____ 2021 р.

ЗАВДАННЯ
до випускної роботи

Студента четвертого курсу, групи Ін-71 спеціальності «Комп'ютерні науки» денної форми навчання Голодного Владислава Володимировича

Тема: «Прогнозування фінансового ринку за допомогою методів математичного аналізу»

Затверджена наказом по СумДУ

№ _____ від _____ 2021 р.

Зміст пояснювальної записки: 1) Аналіз предметної області;
2) Дослідження сучасного стану і перспектив розвитку математичного аналізу фінансових процесів; 3) Визначення вимог до системи; 4) Оцінка функціональної ефективності системи у рамках математичного моделювання;

Дата видачі завдання “ _____ ” _____ 2021 р.

Керівник випускної роботи _____ Шутильєва О.В.

Завдання прийняв до виконання _____ Голодний В.В.

РЕФЕРАТ

Записка: 37 стор., 15 рис., 5 табл., 1 додаток, 11 джерел.

Об'єкт дослідження – вибірка із не стаціонарних фінансових даних.

Мета роботи – наближення вибірки до вигляду генеральної сукупності.

Методи дослідження – методи математичного аналізу даних, інтерполяція.

Результати – розроблено і програмно реалізовано додаток для виконання інтерполяції даних. При цьому використовується і порівнюються результати трьох різних методів: поліном Ньютона, поліном Лагранжа, сплайн. Для проведення максимально достовірного математичного аналізу були взяті реальні дані, отриманий результат співставляється з генеральною сукупністю.

ЗМІСТ

ВСТУП	5
1 ІНФОРМАЦІЙНИЙ ОГЛЯД.....	6
1.1. Дослідження сучасних систем математичного аналізу фінансів	6
1.2. Огляд методів математичного аналізу	10
1.3. Постановка задачі	16
2. ОПИС МЕТОДУ ДОСЛІДЖЕННЯ	18
2.1. Інтерполяція поліномом Лагранжа	18
2.2. Інтерполяція поліномом Ньютона	19
2.3. Інтерполяція сплайном.....	20
3. ПРОГРАМНА РЕАЛІЗАЦІЯ.....	22
3.1. Вхідний математичний опис	22
3.2. Результати моделювання	23
3.3. Короткий опис програмної реалізації	26
ВИСНОВОК.....	29
СПИСОК ЛІТЕРАТУРИ.....	30
ДОДАТОК.....	31

ВСТУП

Економічна галузь завжди потребувала до себе надзвичайно відповідального ставлення. Так, наприклад, у архівах однієї із найстаріших фондових бірж світу Euronext Amsterdam [1] можна знайти вартість цінних паперів починаючи з XVII століття. Така увага, здавалося б до не актуальних даних, пов'язана із тим, що у інформації з минулого схований ключ до майбутнього.

При роботі із фінансами головною задачею є оцінка ризиків та розрахунок майбутнього прибутку. У деякому сенсі це може нагадувати ворожіння, адже необхідно враховувати безліч факторів, деякі із яких маловірогідні або взагалі ще не відомі. Для того, щоб це зробити потрібно володіти найактуальнішою інформацією та вміти критично аналізувати дані минулих років. Якщо з першим пунктом все досить просто, в наш час дані здатні поширюватися по світу за долі секунди, то з другим все не так однозначно.

Гарний спеціаліст повинен володіти комплексним баченням предметної галузі, вміти знаходити паралелі між різними видами діяльності, та як результат надавати прибуткову стратегію розвитку. Проте тут не достатньо лише суб'єктивної оцінки однієї людини, бажано ще підтвердити та довести своє рішення. Найкраще це зробити за допомогою конкретних чисел.

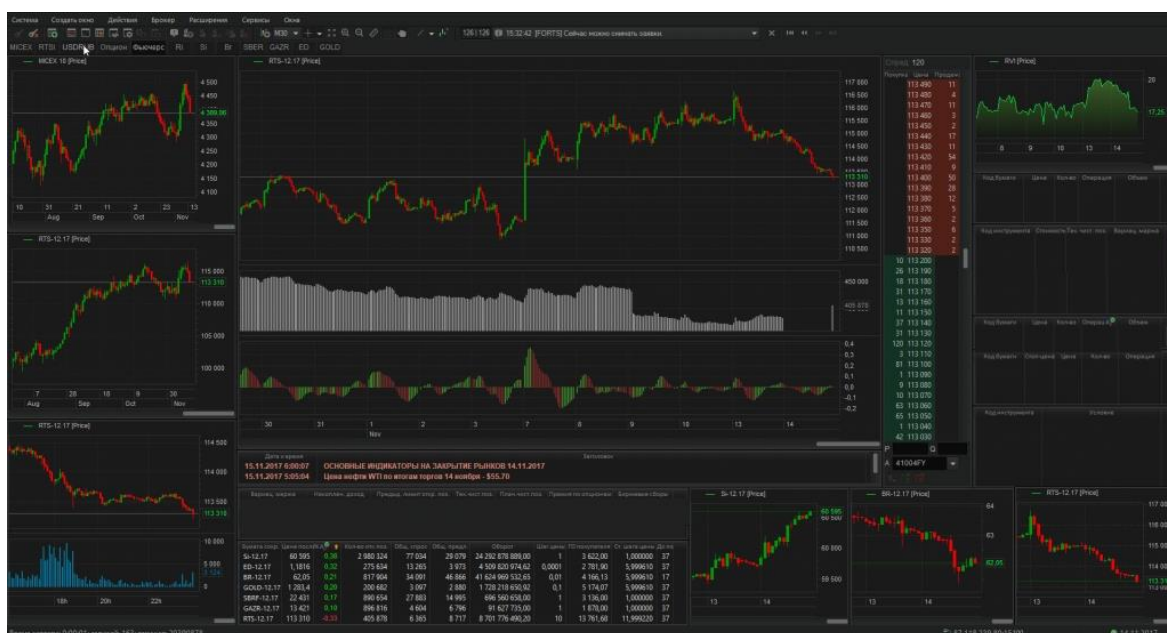
Математика, а саме математичний аналіз та статистика, в тому чи іншому вигляді споконвіку використовуються в економіці. Зазвичай в такий спосіб можна підтвердити або спростувати певне судження, опираючись на результати минулих років. Хоча математичний аналіз є одним із найпотужніших інструментів створених людиною, отриманий в такий спосіб результат залежить від коректності вхідних даних та потребує додаткового тлумачення від спеціаліста.

1 ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Дослідження сучасних систем математичного аналізу фінансів

У економіці використовуються різноманітні спеціалізовані програмні забезпечення. Їх головним призначенням, як правило, є моніторинг бюджету, корегування витрат та оцінка ризиків. Важливо, що при використанні подібних програм остаточне рішення, від якого залежить майбутній успіх, приймає людина.

Торговий термінал QUIK [2] – є одним із найперших загальнодоступних засобів для здійснення інтернет-трейдингу. Програмний комплекс складається із серверної частини та терміналів-клієнтів, які взаємодіють між собою. Головним призначенням програми є своєчасне надання користувачам актуальної інформації з фондового ринку. На рисунку 1.1 можна побачити загальний вигляд терміналу QUIK.



Рисунк 1.1 – Зовнішній вигляд терміналу QUIK

На рисунку 1.1 одразу можна виділити, що інтерфейс є дуже недружнім до користувача. На перший погляд складно зрозуміти що до чого. Проте термінал

QUIK має цікаву особливість у вигляді вбудованої мови програмування QPILE, яка дозволяє створювати торгівельні стратегії, таблиці з розрахованими параметрами, звіти тощо.

Azlex Finance [3] – це типовий представник так званого «домашнього бухгалтера». Програма дозволяє максимально детально розпланувати свій або сімейний бюджет.

Azlex підтримує роботу з усіма валютами світу, більшістю популярних криптовалют та дорогоцінними металами. Усі курси постійно оновлюються. Транзакції можна розділяти на категорії, помічати їх мітками, що дозволить спланувати усі можливі витрати, рисунок 1.2.

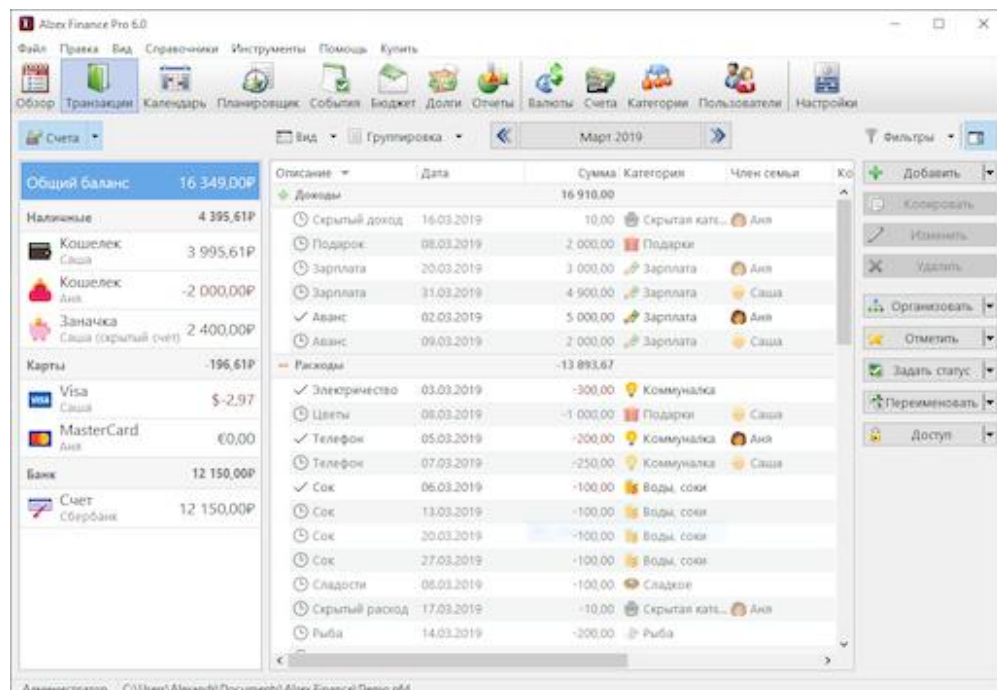


Рисунок 1.2 – Загальний вигляд транзакцій Azlex Finance

Цікавою особливістю програми є функція парсингу СМС, які отримані від банків. Тобто такі дані, залежно від їх змісту, автоматично, потрапляють до заздалегідь визначеної категорії транзакцій. Azlex Finance є кросплатформенною програмою, це дозволяє користувачу взаємодіяти з однією базою даних на різних пристроях.

Debit&Credit [4] – мінімалістичне програмне забезпечення, яке дозволяє детально керувати бюджетом, спостерігати за рухом коштів, фіксувати витрати, створювати та розбивати рахунки на категорії.

Особливу увагу розробники Debit&Credit приділили формуванню звітів. Додаток відображає максимально детальну інформацію про те на що і скільки було витрачено грошей. При цьому програма може прогнозувати майбутній фінансовий стан, для цього вона аналізує поточні, минулі та, на основі наявної інформації, передбачає подальші транзакції користувача.

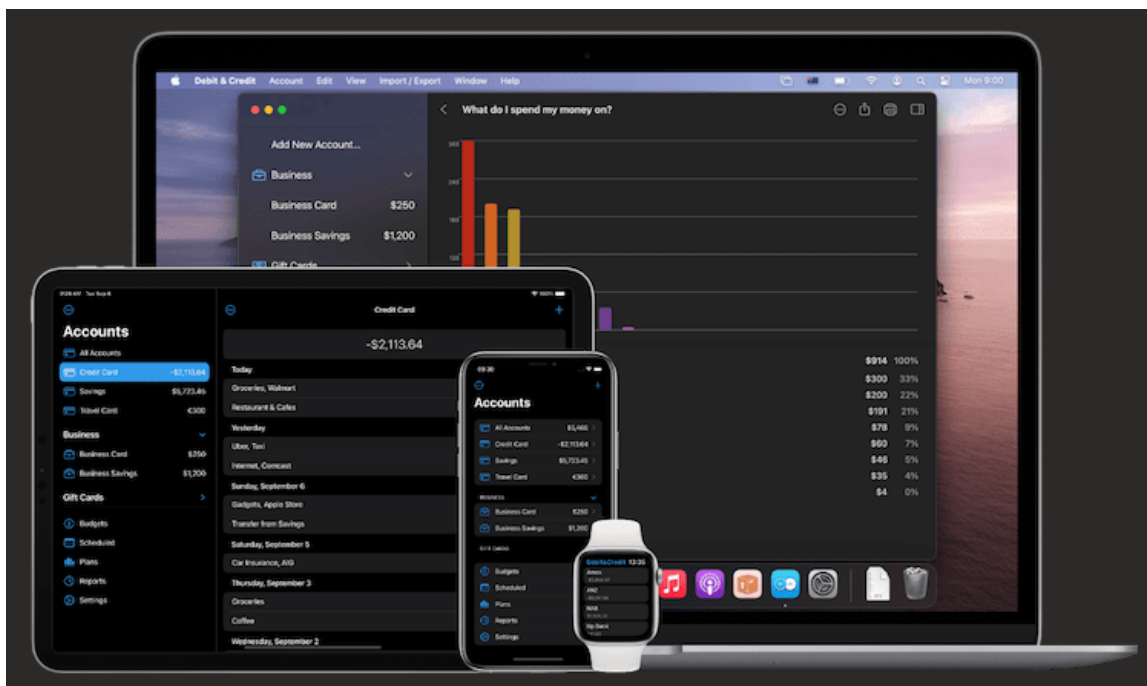


Рисунок 1.2 – Загальний вигляд Debit&Credit на різних пристроях

Слід зазначити, що програма призначена лише для роботи на пристроях компанії Apple.

investing.com [5] – це безкоштовний онлайн ресурс, який надає своїм користувачам надзвичайно повну інформацію про актуальний стан різних активів. Наприклад, це може бути поточна вартість та величина капіталізації певної валюти або компанії, рисунок 1.3.

Top Cryptocurrencies »

Name	Symbol	Price (USD)	Market Cap	Vol (24H)	Total Vol	Chg (24H)	Chg (7D)
Bitcoin	BTC	38,766.9	\$728.05B	\$40.14B	44.61%	+8.59%	+9.16%
Ethereum	ETH	2,497.27	\$292.92B	\$27.14B	30.16%	+5.44%	-6.65%
Tether	USDT	1.0000	\$62.58B	\$62.38B	69.31%	-0.03%	-0.11%
Binance Coin	BNB	363.14	\$56.02B	\$2.13B	2.37%	+5.80%	-6.32%
Cardano	ADA	1.550660	\$49.77B	\$2.77B	3.08%	+4.97%	-6.41%
Dogecoin	DOGE	0.318916	\$41.99B	\$1.46B	1.62%	+3.01%	-12.81%
XRP	XRP	0.88025	\$40.80B	\$3.05B	3.39%	+5.90%	-5.37%
USD Coin	USDC	0.999817	\$23.64B	\$1.71B	1.90%	-0.01%	-0.08%
Polkadot	DOT	21.97000	\$20.99B	\$1.23B	1.37%	+5.07%	-8.05%
Uniswap	UNI	23.1250	\$13.42B	\$399.34M	0.44%	+7.49%	-9.47%

Рисунок 1.3 – Загальна інформація про найпопулярніші криптовалюти

Окрім цього investing додатково надає актуальні новини, які впливають на динаміку активів. Наприклад, можна побачити, що після того як в середині травня 2021 року Китай заявив, що розглядається питання про заборону добутку криптовалют на території країни, її вартість значно впала. Така інформація дає можливості для більш комплексного аналізу ризиків.



Рисунок 1.4 – Приклад графіку ціни біткойна

Проаналізувавши типові програмні забезпечення для моніторингу фінансів можна зробити висновок, що більшість із них виступає в ролі бази даних. Як правило, вони не містять в собі функціоналу для математичного аналізу інформації. Це пов'язано із тим, що більшість задач, де потрібна складна математика, є вузькоспеціалізованими. Тобто потребують чітко сформованих вимог.

Таким чином, перед початком розробки програмного забезпечення потрібно комплексно проаналізувати предметну область, визначити чинники, які впливають або можуть вплинути на результат. Сформувані вимоги до системи.

1.2 Огляд методів математичного аналізу

Математичний аналіз тісно пов'язаний із статистикою, бо саме ця дисципліна займається збиранням, організацією та представленням даних. Чим більш точну і повну вибірку матиме система тим кращим буде результат.

На практиці спеціаліст часто працює з різними наборами даних, для створення більш комплексного бачення поставленої задачі. У таких випадках, щоб показати взаємозв'язок між вибірками використовують коефіцієнт кореляції [7], який представляє із себе середнє арифметичне значення добутку нормованих відхилень за двома досліджуваними ознаками

$$r = \frac{\sum (x - M_x)(y - M_y)}{\sqrt{\sum (x - M_x)^2 \sum (y - M_y)^2}} \quad (1.1)$$

де M_x, M_y – середнє арифметичне вибірок X, Y ; x, y – окремо взяті ознаки вибірок X, Y ;

Значення коефіцієнту кореляції лежить у межах $-1 \leq r \leq 1$. Чим ближче воно наближається до 1, тим тісніший зв'язок між досліджуваними явищами. 0 означає відсутність будь-якої залежності між вибірками. Отже, можна стверджувати, що абсолютна величина характеризує ступінь тісноти зв'язку серед явищ. На рисунку 1.5 показано приклади кореляції.

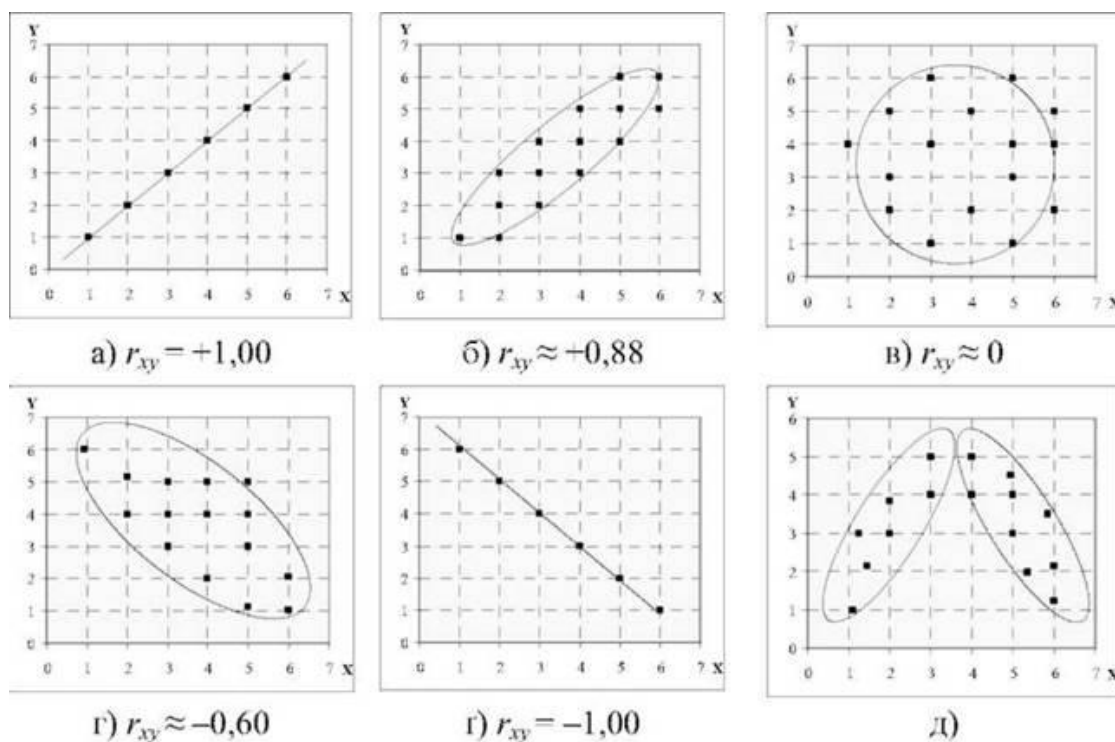


Рисунок 1.5 – Приклади кореляції: а) строга позитивна кореляція; б) сильна позитивна кореляція; в) нульова кореляція; г) помірна негативна кореляція; д) строга негативна кореляція; е) нелінійна кореляція

На практиці досить часто використовують рівняння регресії [8][10], яке демонструє залежність при якій рівномірні зміни параметру x викликають тотожні зміни функції y . При цьому, як видно з рисунку 1.6, чим більший кореляційний зв'язок, тим тісніше точки зосереджені навколо лінії регресії.

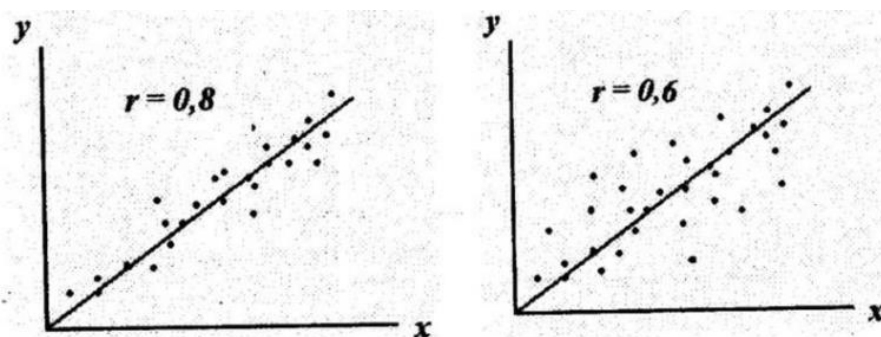


Рисунок 1.6 – Лінійна регресія та залежність від коефіцієнта кореляції

Для виведення рівняння лінійної регресії потрібно встановити на скільки одиниць змінюється одна ознака, порівняно з іншою. Цю умову можна задати у вигляді наступної пропорції, коли обидва параметра x та y задані як відхилення від середніх арифметичних значень M_x, M_y :

$$\frac{x - M_x}{y - M_y} = \frac{1}{b} \quad (1.2)$$

Із пропорції (1.2) можна вивести рівняння регресії:

$$y = M_y + b(x - M_x) \quad (1.3)$$

У рівнянні (1.3) параметр b виступає у ролі, так званого, коефіцієнта регресії, який показує на скільки одиниць зміниться y , якщо x зміниться на одиницю.

$$b_{y/x} = \frac{\Sigma(x - M_x)(y - M_y)}{\Sigma(x - M_x)^2} \quad (1.4)$$

Якщо зв'язок між досліджуваними вибірками не можливо описати лінійною залежністю, то коефіцієнт кореляції не підходить для визначення міри зв'язку. Він може вказати на відсутність взаємозв'язку для даних у яких прослідковується сильна криволінійна залежність. У таких випадках встановлюється кореляція між рівномірними змінами однієї ознаки до нерівномірної, але підпорядкованої певній закономірності, іншої. Зовнішнім проявом нелінійної регресії є те, що функція на графіку приймає вигляд кривих різної форми. Тому потрібний інший показник, який би встановив степінь криволінійної залежності.

Одним із способів дослідження нелінійних залежностей є апроксимація даних [9]. Цей процес передбачає заміну одних математичних об'єктів іншими, які близькі до вихідних. Апроксимація дозволяє досліджувати числові характеристики і якісні властивості об'єкта, зводячи задачу до вивчення більш простих та зручних властивостей ознак.

Метод найменших квадратів (МНК) [10] є одним із групи методів регресійного аналізу, який використовується для статистичного оцінювання параметрів регресійної моделі за емпіричними даними. Згідно з цим методом параметри моделі повинні відповідати такому рівнянню регресії, щоб забезпечувати найменше значення суми квадратів відхилень емпіричних даних від тих, що обчислені за рівнянням регресії. Так, з двох різних наближень тієї ж самої емпіричної функції, що задана у вигляді таблиці, кращим вважається те, для якого сума квадратів відхилення має найменше значення, рисунок 1.7.

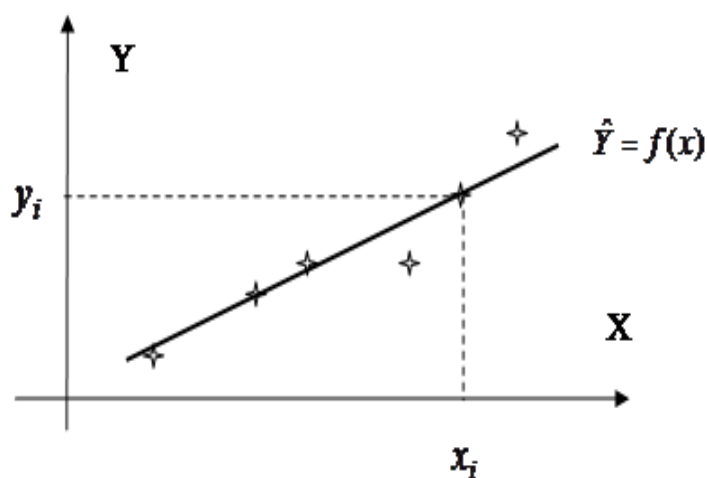


Рисунок 1.7 – Приклад функції регресії побудованої за методом найменших квадратів

Припустимо, що при дослідженні не перервного процесу були отримані дані, які формують таблицю значень невідомої функції $y=y(x)$, які упорядковані за зростанням значення x . Вважається, що результати y_i при точних значеннях x_i може містити випадкові помилки. Будемо припускати, що помилки обчислень є незалежні випадкові величини з однаковими дисперсіями і математичним очікуванням. У цьому випадку, обчислення визначаються як рівномірні, без систематичних помилок. Апроксимація таких даних виглядає як знаходження найбільш доцільної, в даному випадку, лінійної функції, рисунок 1.8. Тобто обрання об'єкту, який максимально точно опише ознаки явища.

Аппроксимирующий многочлен				
$Y = A + BX$				
№	Эмпирическая формула	Линеаризующие преобразования $i = \overline{1, m}$	Соответствие между коэффициентами эмпирической формулы и аппроксимирующего многочлена	Ограничения, $i = \overline{1, m}$
1	$y = a + bx$	$x_i = X_i, y_i = Y_i$	$a = A, b = B$	нет
2	$y = a + \frac{b}{x}$	$\frac{1}{x_i} = X_i, y_i = Y_i$	$a = A, b = B$	$0 \notin [x_1, x_m]$
3	$y = \frac{1}{a + bx}$	$x_i = X_i, \frac{1}{y_i} = Y_i$	$a = A, b = B$	$0 \notin E(y)$
4	$y = \frac{x}{a + bx}$	$x_i = X_i, \frac{x_i}{y_i} = Y_i$	$a = A, b = B$	$\begin{cases} 0 \notin [x_1, x_m], \\ 0 \notin E(y) \end{cases}$
5	$Y = a \cdot b^x$	$x_i = X_i, \ln y_i = Y_i$	$a = e^A, b = e^B$	$y_i > 0$
6	$y = a + b \cdot \ln x$	$\ln x_i = X_i, y_i = Y_i$	$a = A, b = B$	$x_i > 0$
7	$y = \exp(a + b \cdot e^{-x})$	$e^{-x_i} = X_i, \ln y_i = Y_i$	$a = A, b = B$	$y_i > 0$
8	$y = \exp\left(a + \frac{b}{x}\right)$	$\frac{1}{x_i} = X_i, \ln y_i = Y_i$	$a = A, b = B$	$\begin{cases} 0 \notin [x_1, x_m], \\ y_i > 0 \end{cases}$
9	$Y = a \cdot x^b$	$\ln x_i = X_i, \ln y_i = Y_i$	$a = e^A, b = B$	$\begin{cases} x_i > 0, \\ y_i > 0 \end{cases}$

Рисунок 1.8 – Список лінійних функцій, які можна використати при апроксимації даних

Для знаходження коефіцієнтів функції регресії використовують наступні формули:

$$\begin{cases} a = \frac{\sum_{i=1}^n x_i y_i - \frac{1}{n} \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sum_{i=1}^n x_i^2 - \frac{1}{n} (\sum_{i=1}^n x_i)^2} \\ b = \frac{1}{n} \sum_{i=1}^n y_i - \frac{a}{n} \sum_{i=1}^n x_i \end{cases}, \quad (1.5)$$

де x_i, y_i – це ознаки певного об'єкту; n – кількість ознак; a, b – коефіцієнти рівняння регресії;

У фінансах дуже часто застосовують інтерполяцію даних. Це математичний спосіб знаходження проміжних значень величин за наявним дискретним набором відомих значень. Розрізняють інтерполяцію за допомогою поліномів та сплайнів, рисунок 1.9.

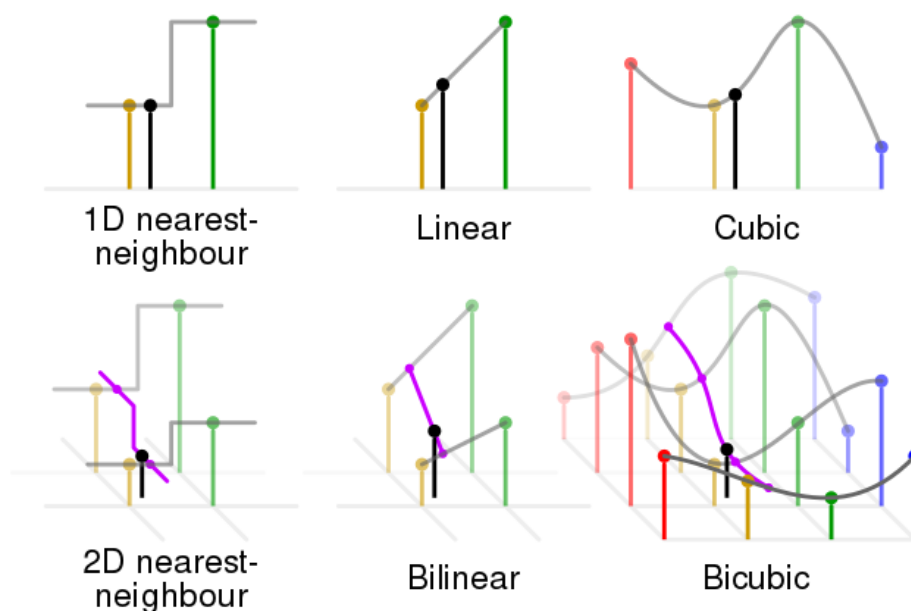


Рисунок 1.9 – Приклади інтерполяції даних

Для того, щоб робити максимально повний економічний аналіз потрібно ознайомитися з деякими фундаментальними явищами у статистиці. Перше з них – це функція розподілу $P(X < x)$, яка показує вірогідність того, що випадкова величина прийме значення менше ніж змінна x . Знаючи деякий закон розподілу дискретних величин можна спрогнозувати вірогідність отримання того або іншого значення не лише в конкретній точці а й, наприклад, в деякому інтервалі, рисунок 1.10.



Рисунок 1.10 – Приклади функції розподілу

При прийнятті будь-яких рішень, які мають вірогідну природу, слід пам'ятати, що більшість таких об'єктів підпорядковуються нормальному закону розподілу [11], який відповідає наступній відомій формулі, коли $\mu = 0$ і $\sigma = 1$:

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}, \text{ де} \quad (1.6)$$

$\frac{1}{\sqrt{2\pi}}$ в даному виразі гарантує, що отримана площа буде дорівнювати 1; $\frac{1}{2}$ в показнику експоненти гарантує, що дисперсія буде дорівнювати 1, а звідси маємо, що і стандартне відхилення буде одиничним;

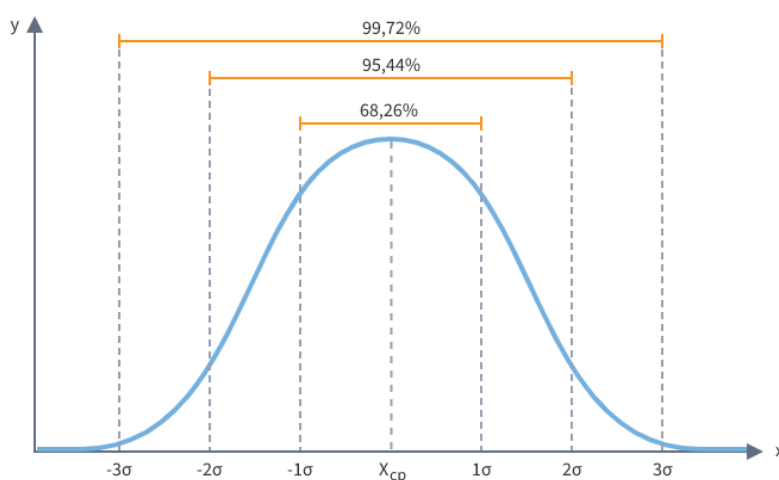


Рисунок 1.11 – Приклади нормального розподілу

До будь-якого закону розподілу, в тому числі і до нормального, можна застосувати правило трьох сігм, рисунок 1.11. Згідно нього вірогідність того, що випадкова величина відхилиться від свого центру математичного очікування більше ніж на три середньоквадратичних відхилень, майже дорівнює нулю. Хоча це й малоймовірно, але на практиці таке трапляється, економісти називають такі ситуації «чорними лебедями». Потрібно завжди пам'ятати про їх існування.

1.3 Постановка задачі

У рамках випускної роботи необхідно розробити програмне забезпечення, яке на основі наявних даних знаходило б проміжні результати. Наприклад,

користувач має звіти по дохідності свого бізнесу за кожен тиждень місяця, але його цікавить скільки приблизно він заробив у окремий день.

Для вирішення цієї задачі найкраще підійдуть методи інтерполяції даних. Більше того, окрім проміжних значень програма повинна бути спроможна надати користувачу приблизний вигляд графіку його дохідності протягом заданого періоду. Тобто, візуалізувати результат інтерполяції.

Для вирішення поставленої задачі треба:

- провести аналіз методів інтерполяції даних;
- реалізувати алгоритм інтерполяції на мові програмування C#;
- розробити користувацький інтерфейс;
- реалізувати візуалізацію результатів.

2. ОПИС МЕТОДУ ДОСЛІДЖЕННЯ

2.1 Інтерполяція поліномом Лагранжа

Нехай, початкова сіткова функція задана за допомогою $(n+1)$ точок $\Omega_n: y_i = f(x_i), i = \overline{0, n}$, де $x_i \in [a, b] = [x_0, x_n]$ – в загальному випадку нерівномірно розташовані вузли, які визначаються кроком h .

Для прикладу розглянемо частковий спосіб інтерполяції. Тут і далі будемо використовувати скорочення: $f_i = f(x_i)$.

Виділимо «вікно» або частковий відрізок $[x_i, x_{i+1}]$, який складається із двох точок. Тоді поліном Лагранжа, який буде виконувати інтерполяцію вхідної функції, матиме наступний вигляд:

$$L_1(x) = \frac{(x-x_{i+1})}{(x_i-x_{i+1})} f_i + \frac{(x-x_i)}{x_{i+1}-x_i} f_{i+1} = P_{1i}(x) * f_i + P_{1i+1}(x) * f_{i+1}, \text{ де } (2.1)$$

$P_{1i}(x), P_{1i+1}(x)$ – це коефіцієнти.

Можна побачити, що (2.1) це алгебраїчний многочлен першої степені, який задовольняє умови функціональної інтерполяції:

$$L_1(x_i) = f_i, L_1(x_{i+1}) = f_{i+1} \quad (2.2)$$

Виділимо «вікно» у вигляді подвійного часткового відрізка $[x_i, x_{i+2}]$. У такому випадку многочлен Лагранжа прийме наступний вигляд:

$$L_2(x) = \frac{(x-x_{i+1})(x-x_{i+2})}{(x_i-x_{i+1})(x_i-x_{i+2})} f_i + \frac{(x-x_i)(x-x_{i+2})}{(x_{i+1}-x_i)(x_{i+1}-x_{i+2})} f_{i+1} + \frac{(x-x_i)(x-x_{i+1})}{(x_{i+2}-x_i)(x_{i+2}-x_{i+1})} f_{i+2} = P_{2i}(x) * f_i + P_{2i+1}(x) * f_{i+1} + P_{2i+2}(x) * f_{i+2}, \text{ де } (2.3)$$

$P_{2i}(x), P_{2i+1}(x), P_{2i+2}(x)$ – це коефіцієнти;

Знову ж таки можна перевірити, що отриманий многочлен (2.3) відповідає умовам функціональної інтерполяції:

$$L_2(x_i) = f_i, L_2(x_{i+1}) = f_{i+1}, L_2(x_{i+2}) = f_{i+2} \quad (2.4)$$

Якщо узагальнити записи (2.1) і (2.3), то отримаємо, що многочлен для довільного відрізка $[x_i, x_k]$ можна записати наступним чином:

$$L_k(x) = \sum_{m=i}^{i+k} \frac{(x-x_i)(x-x_{i+1}) \dots (x-x_{m-1})(x-x_{m+1}) \dots (x-x_k)}{(x_m-x_i)(x_m-x_{i+1}) \dots (x_m-x_{m-1})(x_m-x_{m+1}) \dots (x_m-x_k)} f_m = \sum_{m=i}^{i+k} P_{km}(x) * f_m, \text{ де}$$

$P_{km}(x)$ – коефіцієнти Лагранжа.

Легко перевірити, що $P_{km}(x)$ задовольняє умови функціональної інтерполяції:

$$P_{km}(x_j) = \begin{cases} 1, & j = m, \\ 0, & i \leq j \leq i+k \end{cases} \quad (2.5)$$

Переваги:

- число арифметичних операцій, які необхідні для побудови многочлена, пропорційно n^2 , що є найменшим для всіх форм запису;
- формула в явному вигляді містить значення функції у вузлах інтерполяції;
- можна використати для рівновіддалених і не рівновіддалених вузлів;
- зручно використовувати, коли значення функції змінюється а вузли інтерполяції ні.

Недоліки:

- при збільшенні вузлів, збільшується степінь полінома;
- якщо степінь полінома більша 5-ти, то з'являється хвилястість, ефект Рунге – Мерей.

2.2 Інтерполяція поліномом Ньютона

На практиці функціонального інтерполювання іноді зручніше використовувати поліном Ньютона, степінь якого можна послідовно збільшувати шляхом додавання нових вузлів, які мають більш високу степінь. Такі не симетричні многочлени, альтернативні симетричним поліномам Лагранжа.

Роздільні різниці вводяться для функції $y_i = f(x_i) = f_i, i = \overline{0, n}$, яка задана на не рівномірній сітці, а кінцеві різниці для функції $y_i = f(x_i) = f_i, i = \overline{0, n}$, яка знаходиться на рівномірній.

Обравши всередині рівномірної або не рівномірної сітки часткові відрізки, які будемо досліджувати, введемо наступні визначення для кінцевих та роздільних різниць:

- роздільна різниця k -го порядку:

$$f(x_i, x_{i+1}, \dots, x_{i+k}) = \frac{f(x_i, x_{i+1}, \dots, x_{i+k}) - f(x_i, x_{i+1}, \dots, x_{i+k-1})}{x_{i-k} - x_i}$$

- кінцева різниця k -го порядку:

$$\Delta^k f_i = \Delta(\Delta^{k-1} f_i) = \sum_{j=0}^k (-1)^j C_k^j f_{i+j}, \text{ де } C_k^j = \frac{k!}{(k-j)! j!}$$

Для гладких функцій числові значення $f(x_j, x_{j+1}, \dots, x_{j+k})$ та $\Delta^k f_i$ при зростанні k зменшуються і прямують до нуля, тобто $f(x_j, x_{j+1}, \dots, x_{j+k}) \rightarrow 0$ і $\Delta^k f_i \rightarrow 0$ при $k \rightarrow \infty$.

Зв'язок між роздільними і кінцевими різницями k -го порядку, при $h = \text{const}$, встановлюється наступною рівністю:

$$f(x_i, x_{i+1}, \dots, x_{i+k}) = \frac{\Delta^k f_i}{k! h^k} \quad (2.6)$$

2.3 Інтерполяція сплайном

Сплайн – функція, область визначення якої розбита на шматки, кожен із яких є деяким поліномом (многочленом).

Підвищення точності інтерполювання вимагає збільшення вузлів інтерполяції. Це призведе до зростання степеня інтерполяційних многочленів, але, в умовах відсутності додаткової інформації про задану табличну функцію, останні дають досить значну похибку. У цьому випадку більш ефективним є використання сплайнів, що на проміжку між вузлами інтерполювання є поліномом невисокого степеня.

На практиці широкого застосування набули кубічні сплайни. Доведено, що такий інтерполюючий сплайн – єдина функція з мінімальною кривизною серед усіх функцій, які інтерполюють задані дані.

Отже, якщо $h = \frac{b-a}{n}$, $x_i = x_0 + ih$, $x \in [x_i, x_{i+1}]$, то кубічний сплайн на цьому відрізку буде мати наступний вигляд:

$$S_3(x) = \frac{(x_{i+1} - x)^2(2(x - x_i) + h)}{h^3} f(x_i) + \frac{(x - x_i)^2(2(x_{i+1} - x) + h)}{h^3} f(x_{i+1}) \\ + \frac{(x_{i+1} - x)^2(x - x_i)}{h^2} m_i + \frac{(x - x_i)^2(x - x_{i+1})}{h^2} m_{i+1}$$

Тут $m_i = S'_3(x_i)$, $m_{i+1} = S'_3(x_{i+1})$. Для їх визначення накладають умови неперервності другої похідної в точці x_i та обмеження на значення сплайна і його похідних на кінцях проміжку $[a, b]$ – крайові умови.

Граничні (крайові) умови 1-го типу. Якщо відомо, що $S'_3(a) = f'(a)$; $S'_3(b) = f'(b)$, то для визначення m_i маємо систему рівнянь:

$$\begin{cases} m_0 = f'_0 \\ m_n = f'_n \\ m_{i-1} + 4m_i + m_{i+1} = \frac{3}{h}(f(x_{i+1}) - f(x_i)) \end{cases}$$

3. ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Вхідний математичний опис

За допомогою онлайн-сервісу [investing.com](https://www.investing.com) було взято графік ціни в доларах на ф'ючерси золота [6] за останні пів року, рисунок 3.1. Для прикладу реалізації алгоритму інтерполяції розглядатимуться дані з відрізка в один місяць, на рисунку 3.1 вони виділені червоним кольором.



Рисунок 3.1 – Графік ціни на ф'ючерси золота

У таблиці 3.1 наведено ціни на ф'ючерси золота за травень місяць, при цьому навмисно було зроблено не однорідний крок між датами. Таким чином, в подальшому, вдасться перевірити точність інтерполяції порівнявши результат з реальними даними.

Таблиця 3.1 – Вибірка із цін на ф'ючерси золота

Дата	02.05.21	04.05.21	10.05.21	14.05.21	20.05.21
Ціна	1687	1760	1795	1794	1838

У рамках математичного моделювання, на основі інформації із вибірки, таблиця 3.1, потрібно знайти яка ціна x^* була на актив 12 травня 2021 року. Для перевірки результату буде використана реальна інформація про ціну на ф'ючерси золота в цей день, $x = 1756.64$ долари.

3.2 Результати моделювання

Для початку використаємо і порівняємо результати інтерполяції за допомогою поліномів Ньютона і Лагранжа. Слід зазначити, що дати, які наведені в таблиці 3.1, під час дослідження були переведені в прості числа. Наприклад, 02.05.21 -> 2; 04.05.21 -> 4 і так далі. Це було зроблено для спрощення їх обробки.

На рисунку 3.2 показаний результат інтерполяції даних за допомогою поліному Лагранжа.

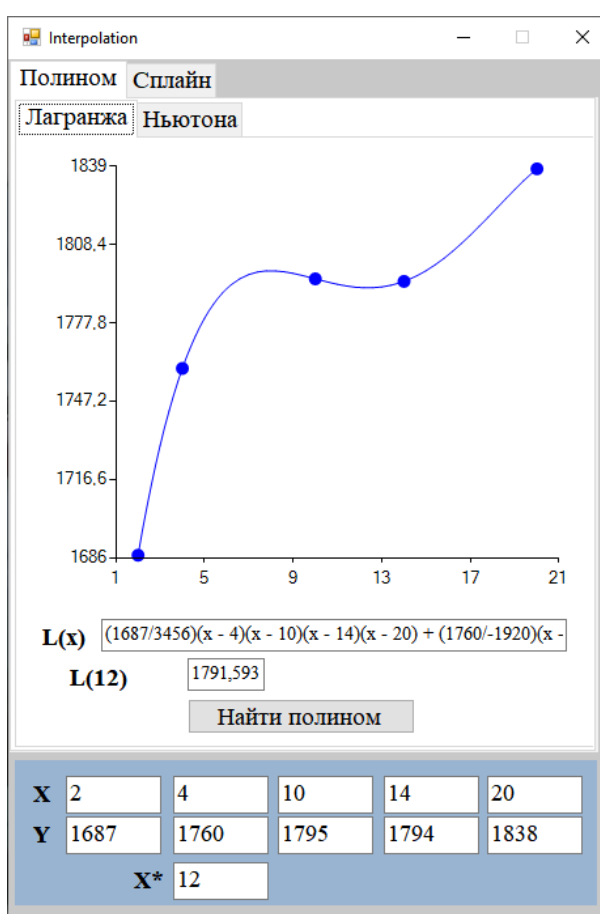


Рисунок 3.2 – Інтерполяція поліномом Лагранжа

У результаті був отриманий наступний поліном Лагранжа:

$$L(x) = \left(\frac{1687}{3456}\right)(x-4)(x-10)(x-14)(x-20) + \left(\frac{1760}{-1920}\right)(x-2) * (x-10)(x-14)(x-20) + \left(\frac{1795}{1920}\right)(x-2)(x-4)(x-14)(x-20) + \left(\frac{1794}{-2880}\right)(x-2) * (x-4)(x-14)(x-20) + \left(\frac{1838}{1}\right)(x-2)(x-4)(x-10)(x-14)$$

$$(x - 4) * (x - 10)(x - 20) + \left(\frac{1838}{17280}\right)(x - 2)(x - 4) * (x - 10)(x - 14)$$

Згідно отриманого результату у визначену дату ціна активу буде становити $x^*=1791.59$ доларів. Для того, щоб в подальшому порівняти точність між різними методами інтерполяції, знайдемо різницю між реальною ціною x та очікуваною x^* .
 $L = |x - x^*| = 1791.59 - 1756.64 = 34.95$.

На рисунку 3.3 показаний результат інтерполяції за допомогою поліному Ньютона.

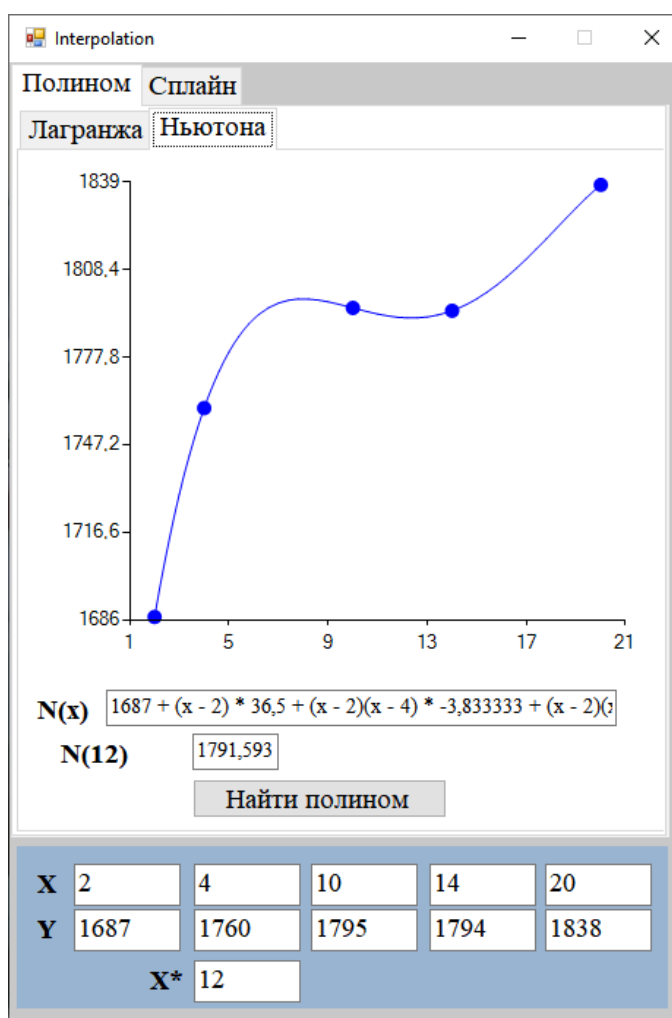


Рисунок 3.3 – Інтерполяція поліномом Ньютона

У результаті роботи програми було отримано поліном Ньютона, який має наступний вигляд:

$$N(x) = 1687 + (x - 2) * 36,5 + (x - 2)(x - 4) * (-3,833333) + (x - 2)(x - 4) * (x - 10) * 0,26875 + (x - 2)(x - 4)(x - 10)(x - 14) * (-0,01018518)$$

Одразу можна помітити, що зовнішній вигляд поліномів Лагранжа та Ньютона є повністю ідентичним, рисунок 3.2 і 3.3. Це пов'язано із тим, що принцип за яким виконується інтерполяція в цих методах є однаковим. Потрібно побудувати многочлен n-го ступеня, який існує лише один для n-вузлів. Однак тут є дві головні відмінності:

- поліном Лагранжа в явному вигляді містить значення функції у вузлах інтерполяції;
- степінь поліному Ньютона можна поступово збільшувати шляхом додавання нових вузлів;

На рисунку 3.4 показаний результат інтерполяції за допомогою сплайну.

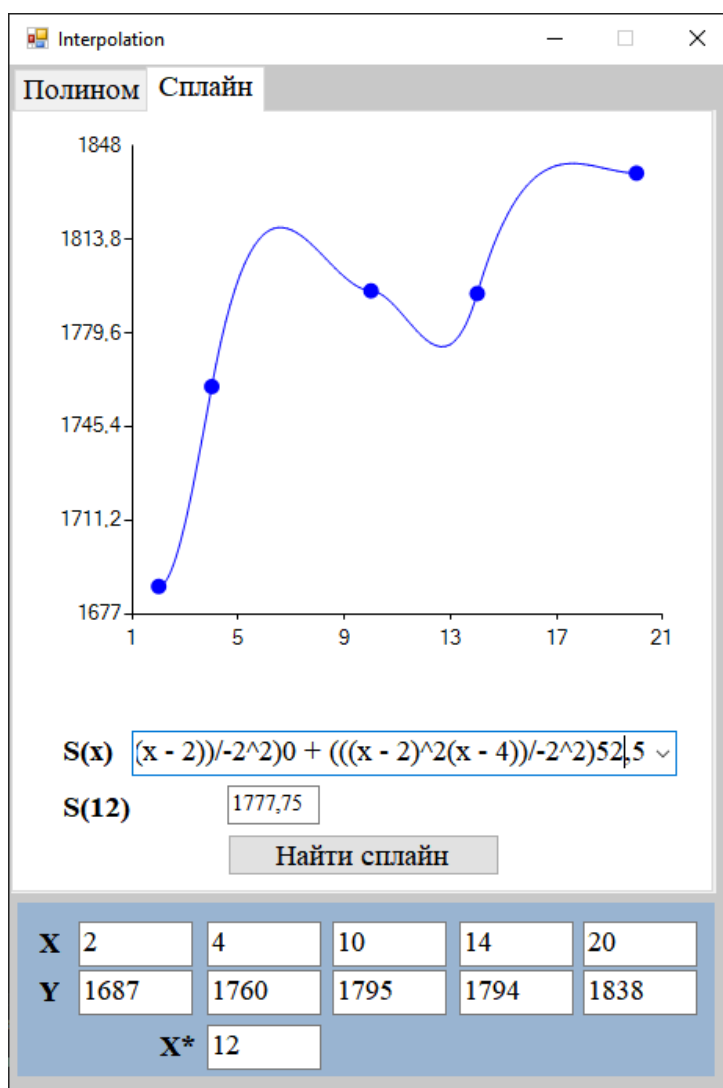


Рисунок 3.4 – Інтерполяція сплайном

При аналізі рисунку 3.4 одразу можна помітити, що отриманий графік максимально подібний до того, який ми досліджуємо, рисунок 3.1. Це говорить про те, що не зважаючи на обмежену вибірку даних, таблиця 3.1, вдалося відтворити, або хоча б наблизитися, до початкового вигляду функції.

Інтерполяція сплайном потребує створення поліномів не високої степені між вузлами, тому в результаті роботи програми було отримано 4 функції, які описують відповідні многочлени:

$$\text{a) } \left((4 - x)^2 \frac{(2(x-2)+(-2))}{-2} \right) 1687 + \left((x - 2)^2 \frac{(2*(4-x)+(-2))}{-2^3} \right) 1760 + \\ + \left((4 - x)^2 \frac{(x-2)}{-2^2} \right) 0 + \left((x - 2)^2 \frac{(x-4)}{-2^2} \right) 52,5.$$

$$\text{b) } \left((10 - x)^2 \left(\frac{(2(x-4)+(-6))}{-6} \right) \right) 1760 + \left((x - 4)^2 \left(\frac{(2(10-x)+(-6))}{-6^3} \right) \right) 1795 + \\ + \left((10 - x)^2 \left(\frac{(x-4)}{-6^2} \right) \right) 52,5 + \left((x - 4)^2 \left(\frac{(x-10)}{-6^2} \right) \right) - 0,5$$

$$\text{c) } \left((14 - x)^2 \left(\frac{(2(x-10)+(-4))}{-4} \right) \right) 1795 + \left((x - 10)^2 \left(\frac{(2*(14-x)+(-4))}{-4^3} \right) \right) 1794 + \\ + \left((14 - x)^2 \left(\frac{(x-10)}{-4^2} \right) \right) (-0,5) + \left((x - 10)^2 \left(\frac{(x-14)}{-4^2} \right) \right) 33$$

$$\text{d) } \left((20 - x)^2 \left(\frac{(2(x-14)+(-6))}{-6} \right) \right) 1794 + \left((x - 14)^2 \left(\frac{(2*(20-x)+(-6))}{-6^3} \right) \right) 1838 + \\ + \left((20 - x)^2 \left(\frac{(x-14)}{-6^2} \right) \right) 33 + \left((x - 14)^2 \left(\frac{(x-20)}{-6^2} \right) \right) 0$$

Знайдемо різницю між очікуваним проміжним значенням ціни x^* та реальними даними x . $S = |x - x^*| = |1756.64 - 1777,75| = 21.11$. Отриманий результат показує не значну похибку, що свідчить про достовірність виконаної інтерполяції.

3.3 Короткий опис програмної реалізації

Програмне забезпечення було написане на мові програмування C#. Для зручності сприйняття короткий опис коду наведений нижче в таблицях, також він знаходиться в Додатку у повному своєму обсязі.

Клас *interpolation* виконує взаємодію із користувацьким інтерфейсом, оброблює вхідні дані та надає результати інтерполяції.

Таблиця 3.2 – Опис складових класу *interpolation*

ЗМІННІ			
Назва	Тип	Модифікатор доступу	Призначення
x	float[]	private	Масив координат вузлів інтерполяції по вісі абсцис
y	float[]	private	Масив координат вузлів інтерполяції по вісі ординат
x_min	float	private	Ліве обмеження вісі абсцис
x_max	float	private	Праве обмеження вісі абсцис

Методи	
Назва	Призначення
interpolation	Конструктор класу. Виконує початкове налаштування
initialize_x_y	Ініціалізація вузлів
Button_Lagrange	Обчислення поліному Лагранжа та виведення його у вигляді графіку
Button_Newton	Обчислення поліному Ньютона та виведення його у вигляді графіку
Button_Spline	Обчислення сплайну та виведення його у вигляді графіку

Клас *Spline* виконує обчислення сплайну на основі наявних вузлів.

Таблиця 3.3 – Опис складових класу *Spline*

ЗМІННІ			
Назва	Тип	Модифікатор доступу	Призначення

c	float[]	private	Розрахунок коефіцієнтів сплайну
Методи			
Назва		Призначення	
public void method(float[] x, float[] y, ref float res, ref string[] spline_str, float X = 0, bool fg = true)		Виконує розрахунок сплайну на основі наявних вузлів	
void initialize_c_a(float[] y, float[] x)		Розраховує коефіцієнти функції	

Клас *Newton* виконує обчислення поліному Ньютона.

Таблиця 3.4 – Опис складових класу *Newton*

Методи	
Назва	Призначення
public void method(float[] x, float[] y, float X, ref float res, ref string polinom, bool fg = true)	Виконує розрахунок поліному на основі наявних вузлів

Клас *Lagrange* виконує обчислення поліному Лагранжа.

Таблиця 3.5 – Опис складових класу *Lagrange*

Методи	
Назва	Призначення
public void method(float[] x, float[] y, float X, ref float res, ref string polinom, bool fg = true)	Виконує розрахунок поліному на основі наявних вузлів

ВИСНОВОК

Під час виконання випускної роботи було розроблено програмне забезпечення, яке дозволяє своєму користувачу виконувати інтерполяцію даних, що являється досить поширеною задачею при роботі із фінансами. Такий тип завдань актуальний, коли у розпорядженні є лише часткова вибірка із даних і потрібно відновити відсутню.

Для проведення математичного моделювання із генеральної сукупності цін на актив була зроблена вибірка із не рівномірним кроком. Таким чином перевірялося чи вдасться системі створити функцію, яка своїм зовнішнім виглядом наближалася б до початкової. Це вдалося максимально точно зробити при використанні інтерполяції сплайном. Можна дійти висновку, що такий метод дослідження цінних паперів є виправданим, бо цей вид активів дуже часто не є стаціонарним за ціною а їх графіки можуть різко змінюватися. Сплайн дуже точно передає ці зміни.

У перспективі можна дослідити інші методи інтерполяції, об'єднати їх. Додати більше математичних показників, які б вказували на доцільність отриманого результату. Спробувати розвинути програму в систему підтримки прийняття рішень, шляхом додавання інтелектуальної складової.

СПИСОК ЛІТЕРАТУРИ

1. Euronext [Електронний ресурс] – <https://www.euronext.com/en>
2. Программний комплекс QUICK [Електронний ресурс] – <https://arqatech.com/ru/products/quik/>
3. Azlex Finance [Електронний ресурс] – <https://www.personalfinances.ru/>
4. Debit & Credit [Електронний ресурс] – <https://debitandcredit.app/>
5. Investing [Електронний ресурс] – <https://investing.com/>
6. Investing, gold contract [Електронний ресурс] – <https://www.investing.com/commodities/gold>
7. Mahdavi Damghani, Babak (2012). The Misleading Value of Measured Correlation. *Wilmott* 2012 (1): 64–73. doi:10.1002/wilm.10167
8. Сеньо П. С. (2007). Теорія ймовірностей та математична статистика (вид. 2-ге, перероб. і доп.). Київ: Знання. с. 446.
9. Українська радянська енциклопедія : у 12 т. / гол. ред. М. П. Бажан; редкол.: О.К. Антонов та ін. — 2-ге вид. — К. : Головна редакція УРЕ, 1974–1985.
10. Любчак, В.О. Методи та алгоритми обчислень: навч. посіб. / В.О. Любчак, Л.Д. Назаренко. - Суми: СумДУ, 2008. - 313 с.
11. Weisstein, Eric W. Normal Distribution Function

ДОДАТОК

```

using System;
using System.Drawing;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;

namespace interpolation
{
    public partial class interpolation : Form
    {
        float[] x = new float[5];
        float[] y = new float[5];

        float x_min;
        float x_max;
        public interpolation()
        {
            InitializeComponent();

            FormBorderStyle = FormBorderStyle.Fixed3D;
            MaximizeBox = false;
        }

        private void initialize_x_y() {
            x[0] = (float)Convert.ToDouble(textBox1.Text);
            x[1] = (float)Convert.ToDouble(textBox2.Text);
            x[2] = (float)Convert.ToDouble(textBox3.Text);
            x[3] = (float)Convert.ToDouble(textBox4.Text);
            x[4] = (float)Convert.ToDouble(textBox5.Text);

            y[0] = (float)Convert.ToDouble(textBox6.Text);
            y[1] = (float)Convert.ToDouble(textBox7.Text);
            y[2] = (float)Convert.ToDouble(textBox8.Text);
            y[3] = (float)Convert.ToDouble(textBox9.Text);
            y[4] = (float)Convert.ToDouble(textBox10.Text);

            x_min = Math.Min(x[0], Math.Min(x[1], Math.Min(x[2], Math.Min(x[3],
x[4])))) - 1;
            x_max = Math.Max(x[0], Math.Max(x[1], Math.Max(x[2], Math.Max(x[3],
x[4])))) + 1;
        }
        private void Button_Lagrange(object sender, EventArgs e)
        {
            initialize_x_y();

            Lagrange lagrange = new Lagrange();
            float res = 0;
            string polinom = "";

            lagrange.method(x, y, (float)Convert.ToDouble(textBox11.Text), ref
res, ref polinom);

            textBox12.Text = polinom;
        }
    }
}

```

```

label7.Text = "L(" + textBox11.Text + ")";
textBox13.Text = res.ToString();

//Подготовка Chart1
chart1.Legends[0].Enabled = false;
chart1.ChartAreas[0].AxisX.MajorGrid.Enabled = false;
chart1.ChartAreas[0].AxisY.MajorGrid.Enabled = false;
chart1.ChartAreas[0].AxisX.Crossing = 0;
chart1.ChartAreas[0].AxisY.Crossing = 0;
chart1.Series.Clear();

//Точки
Series S = new Series();
S.ChartType = SeriesChartType.Point;
S.MarkerSize = 10;
S.Color = Color.Blue;
S.MarkerStyle = MarkerStyle.Circle;
chart1.Series.Add(S);

for (int i = 0; i < x.Length; i++)
    S.Points.AddXY(x[i], y[i]);

//for function
Series S1 = new Series();
S1.ChartType = SeriesChartType.Spline;
S1.Color = Color.Blue;
chart1.Series.Add(S1);

chart1.ChartAreas[0].AxisX.Maximum = x_max;
chart1.ChartAreas[0].AxisX.Minimum = x_min;
chart1.ChartAreas[0].AxisY.Maximum = Math.Max(y[0], Math.Max(y[1],
Math.Max(y[2], Math.Max(y[3], y[4])))) + 1;
chart1.ChartAreas[0].AxisY.Minimum = Math.Min(y[0], Math.Min(y[1],
Math.Min(y[2], Math.Min(y[3], y[4])))) - 1;

float step = (x_max - x_min) / 50;
for (; x_min < x_max; x_min += step)
{
    lagrange.method(x, y, x_min, ref res, ref polinom, false);
    S1.Points.AddXY(x_min, res);
}
}

private void Button_Newton(object sender, EventArgs e)
{
    initialize_x_y();
    Newton newton = new Newton();

    float res = 0;
    string polinom = "";

    newton.method(x, y, (float)Convert.ToDouble(textBox11.Text), ref
res, ref polinom);

    textBox6.Text = polinom;
    label8.Text = "N(" + textBox11.Text + ")";

```



```

textBox3.Text = res.ToString();

//Подготовка Chart2
chart2.Legends[0].Enabled = false;
chart2.ChartAreas[0].AxisX.MajorGrid.Enabled = false;
chart2.ChartAreas[0].AxisY.MajorGrid.Enabled = false;
chart2.ChartAreas[0].AxisX.Crossing = 0;
chart2.ChartAreas[0].AxisY.Crossing = 0;
chart2.Series.Clear();

//Точки
Series S = new Series();
S.ChartType = SeriesChartType.Point;
S.MarkerSize = 10;
S.Color = Color.Blue;
S.MarkerStyle = MarkerStyle.Circle;
chart2.Series.Add(S);

for (int i = 0; i < x.Length; i++)
    S.Points.AddXY(x[i], y[i]);

//for function
Series S1 = new Series();
S1.ChartType = SeriesChartType.Spline;
S1.Color = Color.Blue;
chart2.Series.Add(S1);

chart2.ChartAreas[0].AxisX.Maximum = x_max;
chart2.ChartAreas[0].AxisX.Minimum = x_min;
chart2.ChartAreas[0].AxisY.Maximum = Math.Max(y[0], Math.Max(y[1],
Math.Max(y[2], Math.Max(y[3], y[4])))) + 1;
chart2.ChartAreas[0].AxisY.Minimum = Math.Min(y[0], Math.Min(y[1],
Math.Min(y[2], Math.Min(y[3], y[4])))) - 1;

float step = (x_max - x_min) / 50;
for (; x_min < x_max; x_min += step)
{
    newton.method(x, y, x_min, ref res, ref polinom, false);
    S1.Points.AddXY(x_min, res);
}
}

private void Button_Spline(object sender, EventArgs e)
{
    initialize_x_y();

    float res = 0;
    string[] spline_str = new string[x.Length - 1];
    Spline spline = new Spline();

    spline.method(x, y, ref res, ref spline_str);
    spline.method(x, y, ref res, ref spline_str,
(float)Convert.ToDouble(textBox11.Text), false);

    label10.Text = "S(" + textBox11.Text + ")";
    textBox7.Text = res.ToString();
}

```

```

comboBox1.Items.Clear();
for (int i = 0; i < x.Length - 1; i++)
    comboBox1.Items.Add($"{x[i]} -> {x[i + 1]})" + spline_str[i]);

//Подготовка Chart3
chart3.Legends[0].Enabled = false;
chart3.ChartAreas[0].AxisX.MajorGrid.Enabled = false;
chart3.ChartAreas[0].AxisY.MajorGrid.Enabled = false;
chart3.ChartAreas[0].AxisX.Crossing = 0;
chart3.ChartAreas[0].AxisY.Crossing = 0;
chart3.Series.Clear();

//Точки
Series S = new Series();
S.ChartType = SeriesChartType.Point;
S.MarkerSize = 10;
S.Color = Color.Blue;
S.MarkerStyle = MarkerStyle.Circle;
chart3.Series.Add(S);

for (int i = 0; i < x.Length; i++)
    S.Points.AddXY(x[i], y[i]);

//for function
Series S1 = new Series();
S1.ChartType = SeriesChartType.Spline;
S1.Color = Color.Blue;
chart3.Series.Add(S1);

chart3.ChartAreas[0].AxisX.Maximum = x_max;
chart3.ChartAreas[0].AxisX.Minimum = x_min;
chart3.ChartAreas[0].AxisY.Maximum = Math.Max(y[0], Math.Max(y[1],
Math.Max(y[2], Math.Max(y[3], y[4]))) + 10;
chart3.ChartAreas[0].AxisY.Minimum = Math.Min(y[0], Math.Min(y[1],
Math.Min(y[2], Math.Min(y[3], y[4]))) - 10;

//равновесие
x_max -= 1;
x_min += 1;
float step = (x_max - x_min) / 100;
for (; x_min < x_max; x_min += step)
{
    spline.method(x, y, ref res, ref spline_str, x_min, false);
    S1.Points.AddXY(x_min, res);
}
}
}
class Spline {
    float[] c;

    public void method(float[] x, float[] y, ref float res, ref string[]
spline_str, float X = 0, bool fg = true)
    {
        if (fg)
        {

```

```

        initialize_c_a(y, x);

        //для списка
        for (int i = 0; i < c.Length - 1; i++)
        {
            float h = x[i] - x[i + 1];
            spline_str[i] += $"((x[i + 1] - x)^2(2(x - x[i]) +
{h})/h){y[i]} + " +
            $"((x - x[i])^2(2*(x[i + 1] - x) +
{h}))/h^3){y[i + 1]} + " +
            $"(((x[i + 1] - x)^2(x - x[i]))/h^2){c[i]} + " +
            $"(((x - x[i])^2(x - x[i + 1]))/h^2){c[i + 1]}";
        }
    }
    else
    {
        int ind = 0;
        float h = 0;
        for (int i = 0; i < x.Length; i++) {
            if (X < x[0]) {
                ind = 0;
                break;
            }
            else if (X == x[i])
            {
                ind = i;
                break;
            }
            else if (X < x[i])
            {
                ind = i - 1;
                break;
            }
        }
        if (ind >= 0 && ind < x.Length - 1) h = x[ind + 1] - x[ind];
        else h = x[ind] - x[ind - 1];
        //точка f(x)
        res = (float)((Math.Pow(x[ind + 1] - X, 2) * (2 * (X - x[ind])
+ h)) / Math.Pow(h, 3)) * y[ind] +
            ((Math.Pow(X - x[ind], 2) * (2 * (x[ind + 1] - X) + h)) /
Math.Pow(h, 3)) * y[ind + 1] +
            ((Math.Pow(x[ind + 1] - X, 2) * (X - x[ind])) / Math.Pow(h,
2)) * c[ind] +
            ((Math.Pow(X - x[ind], 2) * (X - x[ind + 1])) / Math.Pow(h,
2)) * c[ind + 1]);
    }

}

void initialize_c_a(float[] y, float[] x) {
    c = new float[y.Length];
    c[0] = 0;
    c[c.Length - 1] = 0;
    for (int i = 1; i < c.Length - 1; i++)
        c[i] = (3 / (x[i] - x[i - 1])) * (y[i + 1] - y[i]);
}

```

```

}

class Newton
{
    public void method(float[] x, float[] y, float X, ref float res, ref
string polinom, bool fg = true)
    {
        float[] now_function = new float[x.Length];
        float[] res_function = new float[x.Length - 1];

        //инициализация
        for (int i = 0; i < x.Length; i++)
            now_function[i] = y[i];

        //разделенная разность n - го порядка
        for (int i = 0; i < x.Length - 1; i++)
        {
            for (int j = 1; j < x.Length - i; j++)
            {
                now_function[j - 1] = (now_function[j] - now_function[j -
1]) / (x[j + i] - x[j - 1]);
            }
            res_function[i] = now_function[0];
        }

        //полином Ньютона
        res = y[0];
        if (fg) polinom += $"{y[0]} + ";
        for (int i = 1; i < x.Length; i++)
        {
            float var = 1;
            for (int j = 0; j < i; j++)
            {
                var *= (X - x[j]);
                if (fg) polinom += $"(x - {x[j]})";
            }

            res += var * res_function[i - 1];
            if (fg) polinom += $" * {res_function[i - 1]}";
            if (i != x.Length - 1) polinom += " + ";
        }
    }
}

class Lagrange
{
    public void method(float[] x, float[] y, float X, ref float res, ref
string polinom, bool fg = true)
    {
        float[] difference_x = new float[x.Length];
        float[] mult_x = new float[x.Length];
        res = 0;
        for (int i = 0; i < x.Length; i++)
        {

```

```
//разница
for (int j = 0; j < x.Length; j++)
{
    if (j != i) difference_x[j] = x[i] - x[j];
}

//произведение элементов
mult_x[i] = 1;
for (int j = 0; j < x.Length; j++)
{
    if (j != i) mult_x[i] *= difference_x[j];
}

}

//Значение в точке
for (int i = 0; i < x.Length; i++)
{
    float var = 1;
    var *= y[i] / mult_x[i];
    if (fg) polinom += $"({y[i].ToString()}/{mult_x[i]})";

    for (int j = 0; j < x.Length; j++) {
        if (j != i)
        {
            var *= (X - x[j]);
            if (fg) polinom += $"(x - {x[j]})";
        }
    }

    res += var;
    if (i != (x.Length - 1) && fg) polinom += " + ";
}

}

}
```