Наведено означення мінімаксного наближення функцій многочленами, алгоритм його побудови, оцінку похибки наближення. Написано програму на мові Python, яка реалізує мінімаксний метод та метод найменших квадратів. З її допомогою наближено деякі характеристики газів цими методами.

The definition of minimax approximation for functions using polynomials, algorithm of its construction, error estimation of such approximation are presented. Python program that implements minmax method and least squares method is written. With help of this program some characteristics of gas using these methods are approximated.

Зміст

[Вступ 3](#_Toc483941999)

[1. Основні поняття мінімаксного наближення 4](#_Toc483942000)

[1.1. Способи задання функцій. Норма похибки 4](#_Toc483942001)

[1.2. Існування мінімаксного многочленного наближення 5](#_Toc483942002)

[1.3. Побудова мінімаксного наближення за схемою Ремеза 7](#_Toc483942003)

[1.4. Заміна точок альтернансу 8](#_Toc483942004)

[1.5. Частинні випадки побудови мінімаксних наближень 10](#_Toc483942005)

[1.6. Похибка мінімаксного наближення 11](#_Toc483942006)

[2. Апроксимація теплофізичних характеристик технічно важливих газів 13](#_Toc483942007)

[2.1. Отримання теплофізичних характеристик 13](#_Toc483942008)

[2.2. Метод найменших квадратів 15](#_Toc483942009)

[2.3. Порівняння результатів наближення 17](#_Toc483942010)

[3. Опис програми і отриманих результатів 19](#_Toc483942011)

[3.1. Призначення програми 19](#_Toc483942012)

[3.2. Умови застосування 19](#_Toc483942013)

[3.3. Запуск програми та задання вхідних даних 19](#_Toc483942014)

[3.4. Опис отриманих результатів 24](#_Toc483942015)

[Висновки 28](#_Toc483942016)

[Додатки 29](#_Toc483942017)

[Додаток 1. (Код програми) 29](#_Toc483942018)

[Додаток 2. Приклади виконання програми 32](#_Toc483942019)

# Вступ

Необхідність моделювання функціональних залежностей виникає в багатьох галузях прикладної математики та інформатики. При розв’язуваннi багатьох задач науково-технiчного характеру доводиться використовувати функції задані таблицею. Проте часто необхідно мати значення функції в точках, яких немає в таблиці. Також виникає необхідність використання простої функції замість складної.

Є два загальні підходи до моделювання функціональних залежностей, які відрізняються один від одного величиною похибки між цією залежністю і табличними даними. Перший підхід, коли в цих даних є певний ступінь похибки або «шуму» , це підібрати таку функціональну залежність яка показує загальний тренд даних. Це робиться тому що кожна окрема точка з цієї таблиці може мати похибку, і ми не намагаємось побудувати функцію, яка точно проходить через ці точки, бо ми будемо повторювати ці похибки. Один метод з цього підходу це метод найменших квадратів.

Другий підхід, коли ми знаємо що табличні дані точні, ми намагаємось побудувати криву, яка точно проходить через табличні точки. Прикладами цього є: значення густини води або теплоємність газу як функція від температури. Оцінка значень в точках, які лежать між точними табличними даними називається інтерполяцією.

У цій бакалаврській роботі розглянуто методи першого підходу, а саме моделювання функціональних залежностей мінімаксними многочленними наближеннями. Саме мінімаксне наближення забезпечує найменшу похибку апроксимації функції на заданому відрізку.

# 1. Основні поняття мінімаксного наближення

## 1.1. Способи задання функціональних залежностей. Норма похибки

Наближувана функціональна залежність  у практичних обчисленнях найчастіше задається або аналітично або дискретно (тобто у вигляді табличного задання) [20,22]. Таблично задану функцію можна представити у вигляді



де значення аргумента . Припускатимемо, що аргументи упорядковані за зростанням:



При застосуванні чисельних методів на комп’ютері врахувати всі значення функції  у всіх точках  неможливо, бо кількість чисел, які можна представити є обмежена. Тому обчислювальні методи будують так, щоб розв’язок задачі на відрізку  був еквівалентний її розв’язку на певній підмножині , яка містить обмежену кількость точок.

Для наближення функції  використовуємо простіший вираз

 (1)

з  параметром [37]. Частинним випадком виразу (1) є многочлен степеня 

 (2)

Точність наближення функції  за допомогою виразу (1) на проміжку  xарактеризується віддалю між цими функціями. Спосіб виміру цієї віддалі визначає норму похибки наближення функції  за допомогою виразу (1) на проміжку  (або на множині ). У загальному випадку у виразах для похибки використовують зважену віддаль (зважену різницю)

 (3)

де вага  при , .

Вибір певного виду норми похибки залежить передусім від конкретних задач, які ставлять при наближенні функцій. У теоретичних дослідженнях часто використовується норма похибки  [1,23]



Найчастіше використовують норми ,  та .

Норму  вживають для зменшення суми площ, які обмежуються кривими  та  [20]



Норму  або середньоквадратичну похибку зазвичай використовують при обробці дослідних даних



Норму  (чебишовська норма або норма ) використовують, щоб найточніше представити кожне значення наближуваної функції , за умови, що її значення відомі достатньо точно



Функцію (3) називають функцією похибки, а її графік - кривою похибки.

## 1.2. Існування мінімаксного многочленного наближення

За теоремою Вейєрштрасса [1,38] для довільних неперервних на обмеженому проміжку  функцій  та  і довільного  можна знайти такий многочлен , що



Очевидно, що найменше значення степеня  многочлена  суттєво залежить від способу наближення. Серед усіх способів наближення функцій найменшу похибку a, отже, і найменший степінь  при заданій точності , дає мінімаксне ( його ще називають найкраще чебишовське) наближення [22].

Вираз , для якого максимальне значення абсолютної величини зваженої похибки (3) досягає на проміжку  найменшого значення

 (4)

називають мінімаксним або найкращим чебишовським зваженим (з вагою ) наближенням функції  виразом  на проміжку . При  маємо мінімаксне абсолютне наближення, при  - мінімаксне відносне.

Величину (4) називатимемо мінімальним (зваженим) відхиленням і позначаємо ;  - мінімальне абсолютне відхилення;  - мінімальне відносне відхилення.

Розглянемо властивості мінімаксних наближень многочленами [20,22,33].

**Теорема 1.** Для будь-яких неперервних на проміжку  функцій  та  і довільного , існує єдиний многочлен  степеня , що має найменше відхилення .

**Теорема 2.** Нехай на проміжку  задано неперервні функції  та . Тоді для того, щоб деякий многочлен  степеня не вище  був многочленом мінімаксного зваженого наближення функції  на проміжку  необхідно і достатньо, щоб на цьому проміжку знайшлась хоча б одна система з  точок , у яких зважена різниця (3) почергово набувала значень різних знаків і досягала за модулем найбільшого на  значення тобто

 (5)

Систему точок  називають системою точок (чебишовського) альтернансу. Для побудови многочлена мінімаксного наближення необхідно визначити точки альтернансу. Точне їх значення можна знайти тільки у часткових випадках.

## 1.3. Побудова мінімаксного наближення за схемою Ремеза

Процес знаходження точок альтернансу будується ітераційними методами. Найчастіше на практиці застосовують методи розроблені українським математиком Є.Я. Ремезом [39, 22]. Наведу один із методів. Він складається з таких етапів.

1. На проміжку  вибираємо початкове наближення  до точок альтернансу



Точки альтернансу можна, наприклад, обчислити за такою формулою .

2. Здійснюємо чебишовську інтерполяцію для множини точок , тобто визначаємо коефіцієнти многочлена  і величину , для яких виконуються умови . Для знаходження вказаних величин розв’язуємо систему рівнянь:

 (6)

Вона є системою  алгебраїчних рівнянь з  невідомими:  та 

3. Перевіряємо виконання рівності

 (7)

Якщо рівність (7) виконується, то за теоремою 2 многочлен  і є многочлен найкращого (мінімаксного) наближення. При комп’ютерній реалізації алгоритму перевірку умови (7) заміняють перевіркою нерівності

 (8)

де  - допустима відносна помилка у визначенні похибки наближення. Її приймають рівною  або .

4. Якщо умова 7 чи 8 не виконується, то приймаємо  і вибираємо наступне (уточнене) наближення до точок чебишовського альтернансу (переходимо до наступної ітерації). Далі алгоритм повторюємо продовжуючи виконувати з п.2.

При обчисленні на комп’ютері у цьому пункті іноді ще перевіряють умови



де  - допустима помилка у визначенні точок альтернансу. Якщо остання нерівність справедлива для всіх точок  (тобто точки альтернансу змінюються несуттєво), то вважаємо, що многочлен мінімаксного наближення знайдено.

## 1.4. Заміна точок альтернансу

Існує кілька методів заміни точок альтернансу. Можлива заміна одної або кількох точок одночасно. Найпростішим алгоритмом є алгоритм Є.Я. Ремеза з одноточковою заміною (алгоритм Валлє-Пуссена) [39,22,33]. Опишемо цей алгоритм.

Нехай при виконанні пункту 3 ми знайшли точку , для якої виконується . Можливі три випадки взаємного розташування точок альтернансу та точки :

1. 

2. 

3. 

Розглянемо спосіб заміни точок альтернансу для кожного випадку.

1. Знайдемо ціле число  таке, що . Якщо , то робимо присвоєння , у протилежному випадку . Решту точок альтеранансу не змінюємо.

2. Якщо , то присвоюємо . Решту точок альтернансу не змінюємо. Якщо це не так, то заміняємо усі точки альтернансу за формулами:



У цьому випадку із альтернансу виключається остання точка 

3. Якщо , то приймаємо  і решту точок альтернансу не змінюємо. Якщо це не так, то замінюємо усі точки альтернансу за формулами:



У цьому випадку із V-альтернансу виключається перша точка .

Таким чином наступна система точок альтернансу відрізняєтся від попередньої тим, що точка , у якій є максимум абсолютної величини зваженої похибки, вводиться у альтернанс замість однієї із старих точок.

Відомо, що алгоритм Валле-Пуссена для заміни точок альтернансу при знаходженні мінімаксного наближення неперервної функції многочленом на проміжку  збігається незалежно від початкового наближення до точок альтернансу. Він збігається зі швидкістю гометричної прогресії у тому сенсі, що знайдуться такі числа  та , що відхилення  многочлена  від функції  будуть задовольняти нерівності



Фактична швидкість збіжності залежить від диференціальних властивостей функції та використовуваного алгоритму заміни точок альтернансу. Відомо, що коли  або  і  не змінює знак при , то граничні точки проміжку  є точками альтернансу [39, 22]. Тому у цьому випадку алгоритм Валле-Пуссена для наближення многочленами невисоких степенів  практично не програє у швидкості порівняно з іншими алгоритмами типу Є.Я. Ремеза.

Зауважимо, що наведені властивості мінімаксного наближення непервної на  функції  многочленом виконуються і для наближення табличної функції.

## 1.5. Частинні випадки побудови мінімаксних наближень

Приклад 1 [20]. Знайдемо мінімаксне абсолютне наближення сталою: . У цьому випадку система рівнянь (6) має вигляд

.

Додамо два рівняння цієї системи і одержимо вираз для .

Підставимо це значення у перше рівняння системи отримаємо вираз для похибки . Очевидно, що точки альтернансу у цьому випадку співпадатимуть із точками мінімуму та максимуму функції . Тому  
Якщо функція монотонна, то ці значення досягаються на краях проміжку . Тому у цьому випадку .

Приклад 2. Знайдемо мінімаксне абсолютне наближення многочленом першого степеня (прямою): . Система рівнянь (6) у цьому випадку складається із трьох рівнянь: , .

Віднімемо від третього рівняння системи перше і отримаємо вираз для параметра  
. Додамо два перші рівняння системи, отримаємо вираз для параметра

Якщо і не змінює знак, то , а в центральній точці альтернанcу функція похибки має екстремум і Тому . Прирівнюючи два вирази для коефіцієнта отримаємо трансцендентне рівняння для визначення точки :. Після визначення цієї точки знаходимо та за формулами:

.

Розкладаючи у виразі для функцію в ряд в околі точки :

,

де маємо

.

## 1.6. Похибка мінімаксного наближення

Порівняємо максимальну похибку мінімаксного наближення многочленом з іншими наближеннями многочленом. Нехай і при . Відомо, що у цьому випадку [22,23]

. (10)

Наблизимо таку функцію відрізком ряду Тейлора з коефіцієнтом в околі точки

де . Очевидно, що при максимальне значення залишкового члену ряду рівне [1,37]

(11)

де . За цією ж формулою обчислюється похибка наближення ланкою ермітового сплайну непарного степеня. Із наведених формул (10) і (11) можна бачити, що використання мінімаксного наближення замість наближення іншими способами суттєво зменшує одержувану при цьому похибку.

# 2. Апроксимація теплофізичних характеристик технічно важливих газів

## 2.1. Отримання теплофізичних характеристик

В наш час у зв’язку з інтенсивним використанням технічно важливих газів у багатьох областях сучасної науки, техніки і технології, зокрема в енергетиці, геології, хімічній технології, газовій промисловості тощо, значно зростає потреба в більш точному визначенні їх параметрів і характеристик. Отримання надійних довідкових даних про теплофізичні властивості стиснутих газів при високих температурах пов’язано із значними труднощами. Це пояснюється тим, що існуючі експериментальні дані отримані в обмеженому температурному інтервалі з верхньою границею, яка не перевищує 800-1000К. Ця обставина не дозволяє використовувати традиційні методи для розрахунку параметрів при високих температурах, бо такі методи полягають в побудові емпіричних рівнянь за результатами експериментальних даних і розрахунку по ним дискретних значень параметрів і характеристик, оскільки ці рівняння непридатні для отримання даних за межами експериментально досліджуваної області [28,29].

Для усунення вказаних недоліків були створені методи отримання рівнянь, які відображають властивості реальних газів і які є придатні для екстраполяційних розрахунків теплофізичних властивостей. Для цих розрахунків доцільно застосовувати теоретично обґрунтовані рівняння, які дозволяють розраховувати будь-які теплофізичні властивості газів, якщо відомо закон міжмолекулярної взаємодії, і які містять мінімальну кількість невідомих констант – параметрів модельного потенціалу.

Таким рівнянням для газів є віріальне рівняння стану, віріальні коефіцієнти якого можуть бути розраховані на основі прийнятих функцій міжмолекулярної взаємодії, а параметри потенціалу визначаються із експериментальних значень густини. Ці рівняння дуже громіздкі і розв’язуються чисельно з використанням обчислювальної техніки. Багато коефіцієнтів виражаються через кратні інтеграли, які також не можуть бути обчислені аналітично, а тому обчислюються за допомогою квадратурних формул на комп’ютерах [1,17,37]. В результаті обчислень отримують дискретні значення певних характеристик газів (густина, фактор стискуваності, ізохорна та ізобарна теплоємності, швидкість звуку, теплопровідність, в’язкість тощо), при певних значеннях тиску і температури, що змінюються з доволі великим кроком. Для обчислення характеристик газів у проміжних точках потрібно затратити багато часу і зусиль. Тому доцільно апроксимувати отримані дані, наприклад, за мінімаксним критерієм або методом найменших квадратів і, маючи готові аналітичні вирази, обчислювати потрібні характеристики газів у проміжних точках з достатньо великою точністю [22,24,33].

Далі в таблиці 1 наведено залежності щільності *ρ*, ізохорної і ізобарної теплоємностей водню H від зміни тиску при температурі Т=500К. У таблиці 2 наведено значення щільності, фактора стискуваності , швидкості звуку , в’язкості *μ* та теплопровідості [λ](https://en.wiktionary.org/wiki/%CE%BB) кисню при температурі Т=500К і зміні тиску *p*.

Таблиця 1. Характеристики водню при Т=500К.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **ρ** |  |  |  |  |
| 0.1 | 0.048 | 1.0004 | 72.293 | 10.389 | 14.514 |
| 1.0 | 0.483 | 1.0040 | 62.794 | 10.389 | 14.518 |
| 2.0 | 0.962 | 1.0080 | 59.933 | 10.389 | 14.522 |
| 3.0 | 1.438 | 1.0120 | 58.259 | 10.388 | 14.526 |
| 4.0 | 1.909 | 1.0160 | 57.070 | 10.388 | 14.530 |
| 5.0 | 2.377 | 1.0200 | 56.147 | 10.388 | 14.534 |
| 6.0 | 2.841 | 1.0240 | 55.393 | 10.388 | 14.538 |
| 8.0 | 3.759 | 1.0320 | 54.203 | 10.388 | 14.546 |
| 10.0 | 4.663 | 1.0400 | 53.278 | 10.388 | 14.554 |
| 12.0 | 5.553 | 1.0480 | 52.522 | 10.388 | 14.561 |
| 16.0 | 7.293 | 1.0639 | 51.327 | 10.388 | 14.574 |
| 20.0 | 8.982 | 1.0797 | 50.399 | 10.389 | 14.587 |
| 25.0 | 11.026 | 1.0995 | 49.470 | 10.390 | 14.601 |
| 30.0 | 12.998 | 1.1192 | 48.709 | 10.391 | 14.614 |
| 35.0 | 14.903 | 1.1389 | 48.065 | 10.393 | 14.626 |
| 40.0 | 16.745 | 1.1584 | 47.506 | 10.396 | 14.636 |
| 45.0 | 18.526 | 1.1779 | 47.013 | 10.399 | 14.646 |
| 50.0 | 20.251 | 1.1973 | 46.571 | 10.402 | 14.655 |
| 60.0 | 23.542 | 1.2359 | 45.805 | 10.409 | 14.671 |
| 70.0 | 26.640 | 1.2742 | 45.157 | 10.417 | 14.685 |
| 80.0 | 29.564 | 1.3122 | 44.594 | 10.425 | 14.697 |

Таблиця 2. Характеристики кисню при Т=500К.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **ρ** |  |  |  | [**λ**](https://en.wiktionary.org/wiki/%CE%BB) |
| 0.1 | 0.77 | 1.0002 | 421.1 | 303.5 | 41.0 |
| 1.0 | 7.68 | 1.0018 | 422.5 | 304.3 | 41.2 |
| 2.0 | 15.34 | 1.0037 | 424.1 | 305.3 | 41.4 |
| 3.0 | 22.96 | 1.0057 | 425.8 | 306.3 | 41.6 |
| 4.0 | 30.55 | 1.0078 | 427.5 | 307.4 | 41.8 |
| 5.0 | 38.11 | 1.0099 | 429.3 | 308.4 | 42.1 |
| 6.0 | 45.63 | 1.0122 | 431.1 | 309.6 | 42.3 |
| 8.0 | 60.55 | 1.0169 | 434.8 | 311.9 | 42.8 |
| 10.0 | 75.31 | 1.0221 | 438.7 | 314.4 | 43.2 |
| 12.0 | 89.89 | 1.0275 | 442.8 | 316.9 | 43.7 |
| 16.0 | 118.49 | 1.0394 | 451.5 | 322.4 | 44.8 |
| 20.0 | 146.26 | 1.0525 | 460.8 | 328.3 | 45.8 |
| 25.0 | 179.75 | 1.0706 | 473.1 | 336.0 | 47.2 |
| 30.0 | 211.80 | 1.0903 | 486.2 | 344.2 | 48.6 |
| 35.0 | 242.39 | 1.1115 | 499.3 | 352.7 | 50.0 |
| 40.0 | 271.53 | 1.1339 | 513.8 | 361.3 | 51.4 |
| 45.0 | 299.25 | 1.1575 | 528.2 | 370.1 | 52.8 |
| 50.0 | 325.60 | 1.1820 | 542.8 | 378.9 | 54.2 |

## 2.2. Метод найменших квадратів

Нехай в результатi вимiрювань деякої фізичної залежності, яка описується функцiєю  в точках  отримаємо дискретні значення  [1,17,37]. За цими дискретними (табличними) даними потрібно побудувати аналiтичну формулу

 (1)

яка залежить вiд  параметрiв , причому функцiя  має "досить добре" наближати функцiю  на всьому промiжку . Вигляд функцiї  i кiлькiсть параметрiв у деяких випадках вiдомi на основi додаткових мiркувань. У iнших випадках параметри і вигляд функцiї  визначаються за графiком, побудованим за значеннями  так, щоб залежнiсть (1) була досить простою i добре вiдображала величини, отримані в результаті вимірювань.

По виміряних значеннях будуємо систему рівнянь

 (2)

Якщо вона має єдиний розв’язок, то його знаходять з будь-яких *m* рiвнянь системи (2). Однак, у загальному випадку  є отримані з деякою похибкою i точний вигляд залежностi  невiдомий. Тому система (2) зазвичай є несумiсною. Знайдемо параметри  так, щоб у деякому розумiннi всi рiвняння системи (2) задовольнялися з найменшою похибкою, тобто, щоб параметри  мiнiмiзували функцiю



Такий метод розв’язування системи (2) називають методом найменших квадратiв. Якщо функцiя  досягає абсолютного мiнiмуму в областi змiни параметрiв , то, розв’язуючи систему



знаходимо точки, в яких може бути екстремум. Вибравши той розв’язок, який належить областi змiни параметрiв  i в якому функцiя  має абсолютний мiнiмум, знаходимо незалежнi значення параметрів .

Якщо  лiнiйно залежить вiд параметрiв , тобто



то система (2) набуває вигляду

 (3)

Розв’язуємо систему (3) токим чином, щоб визначити невiдомi, якi мiнiмiзують суму квадратiв нев’язок, тобто суму вигляду



Для виконання умови мiнiмуму величини , яка є функцiєю вiд параметрів , розв’язуємо систему лiнiйних алгебраїчних рiвнянь



яку для зручності представляємо у вигляді

 (4)

Розв’язок системи лiнiйних алгебраїчних рiвнянь (4) з  невiдомими вважаємо наближеним розв’язком системи (2) за методом найменших квадратів. Найчастіше в якості  вибирають функції . Систему лінійних алгебраїчних рівнянь (4) розв’язують методом Гауса або методом Гауса з вибором головного елемента [1,37].

## 2.3. Порівняння результатів наближення

Було проведено мінімаксне наближення та наближення методом найменших квадратів експериментально отриманих залежностей: щільності кисню *ρ* від тиску *p* при температурі . Результати наведено у таблицях 3 і 4.

Таблиця 3. Мінімаксне наближення.

|  |  |  |
| --- | --- | --- |
| Степінь многочлена | Вигляд многочлена | Максимальна похибка |
| 1 |  | 8,4452 |
| 2 |  | 0.3173 |
| 3 |  | 0.1185 |
| 4 |  | 0.0075 |

Таблиця 4. Метод найменших квадратів.

|  |  |  |
| --- | --- | --- |
| Степінь многочлена | Вигляд многочлена | Максимальна похибка |
| 1 |  | 12.0199 |
| 2 |  | 0.4187 |
| 3 |  | 0.1328 |
| 4 |  | 0.0114 |

Графіки наближуваних даних і многочленів побудованих за мінімаксним критерієм і метод найменших квадратів, наведено у додатку 2. Порівнюючи інформацію із таблиць 3 і 4 видно, що мінімаксне наближення є кращим ніж наближення за методом найменших квадратів, бо дає менші похибки для многочленів однакового степеня. Також перевага мінімаксного наближення над методом найменших квадратів має місце і для аналітично заданих функцій (див. додаток 2).

# 3. Опис програми і отриманих результатів

## 3.1. Призначення програми

Програма призначена для побудови апроксимацій функцій многочленами двома способами: мінімаксним наближенням, та методом найменших квадратів. Функція може бути задана двома способами: дискретним (у вигляді таблиці), або неперервним (аналітично). Програма дає змогу знайти коефіцієнти многочленів, максимальні похибки, побудувати графіки многочлена, яким наближуємо функцію, функції, яка наближується.  
Є можливість порівняти два наближення для певної функції.

## 3.2. Умови застосування

Вимоги до ПК:

* 32-розрядний (x86) або 64-розрядний (x64) процесор із тактовою частотою 1 ГГц або швидший;
* 1 гігабайт (ГБ) RAM (для 32-розрядної версії) або 2 ГБ (для 64-розрядної версії);

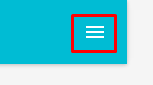
ОС: Windows 7/8/10

Для того, що запустити програму на комп’ютері достатньо мати сучасний браузер, наприклад, Google Chrome, Mozilla Firefox та доступ до мережі інтернет. Далі достатньо зайти на веб сторінку:

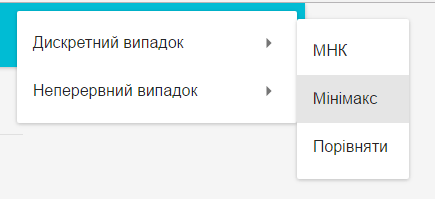
https://bodya17.github.io/diplom/index.html

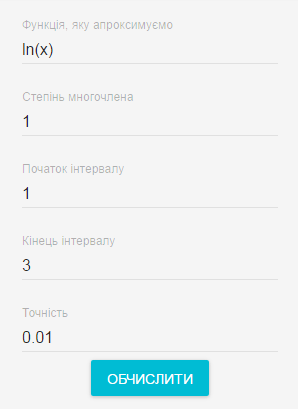
## 3.3. Запуск програми та задання вхідних даних

Для знаходження наближення функції, спочатку потрібно вибрати, яким чином задана функція (таблично чи аналітично). Це можна зробити натиснувши кнопку з правої сторони головного меню сайту.



Далі необхідно вибрати метод яким потрібно апроксимувати функцію.



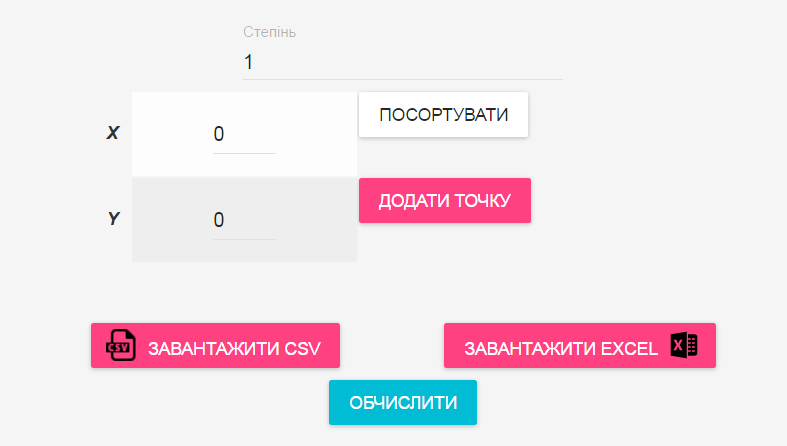
Для неперервного випадку, потрібно заповнити наступну форму:  


Як видно з рисунку, користувачу потрібно ввести функцію для апроксимації. Приклади вводу функцій:

|  |  |
| --- | --- |
|  | e^x |
|  | sqrt(x) |
|  | cos(x)^2 або (cos(x))^2 |
|  | 1/x |

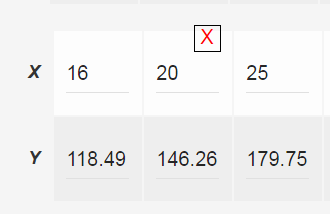
Точність – допустима відносна похибка у визначенні похибки наближення у мінімаксному наближенні.

Для дискретного випадку:



Тут можна задати степінь апроксимуючого многочлена та задати табличну функцію. Це можна зробити двома способами:

1) Вручну. За допомогою кнопки “ДОДАТИ ТОЧКУ” можна додати до таблиці, яка знаходиться лівіше, ще одну точку. Редагувати точки можна відразу в таблиці. При необхідності можна також вилучити точку навівши курсор на неї.



Також є можливість посортувати ці точки (по змінній x), натиснувши на відповідну кнопку.

2) Завантажити з файлу. Файл повинен бути у форматі CSV(Comma Separated Values), тобто значення які розділені комою. Перший стовпець – це значення , другий – . Приклад файлу CSV:

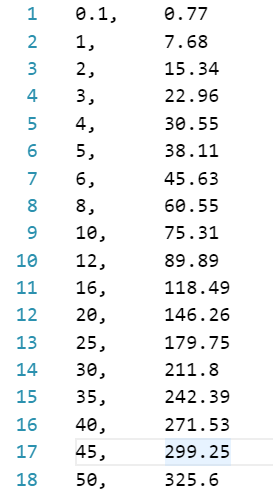
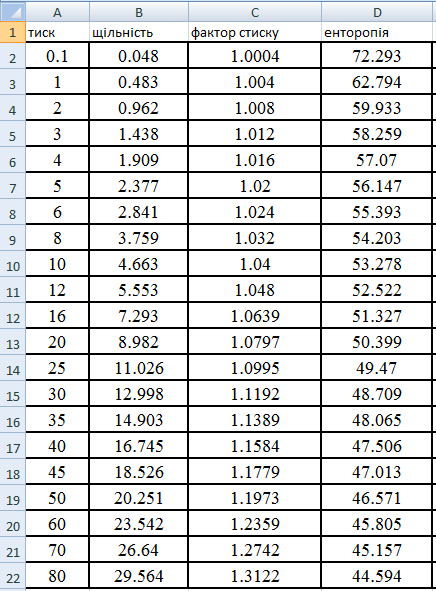
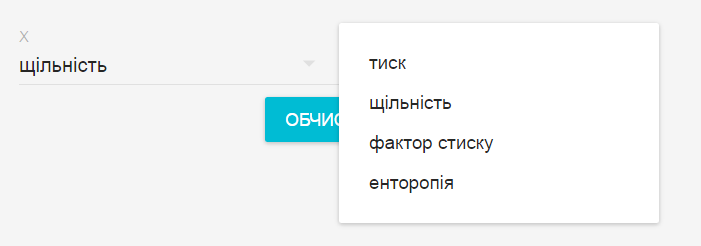


Рис. 1

Також файл може бути у форматі xlsx (Excel). Приклад Excel файлу.



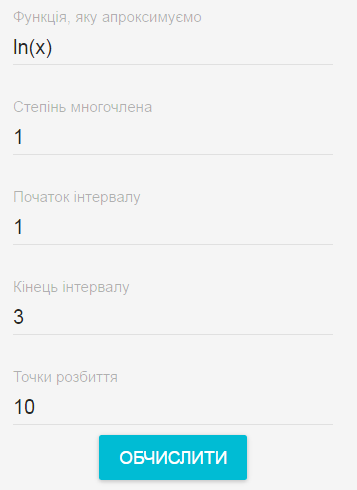
Далі необхідно вибрати яка величина – змінна , а яка – .



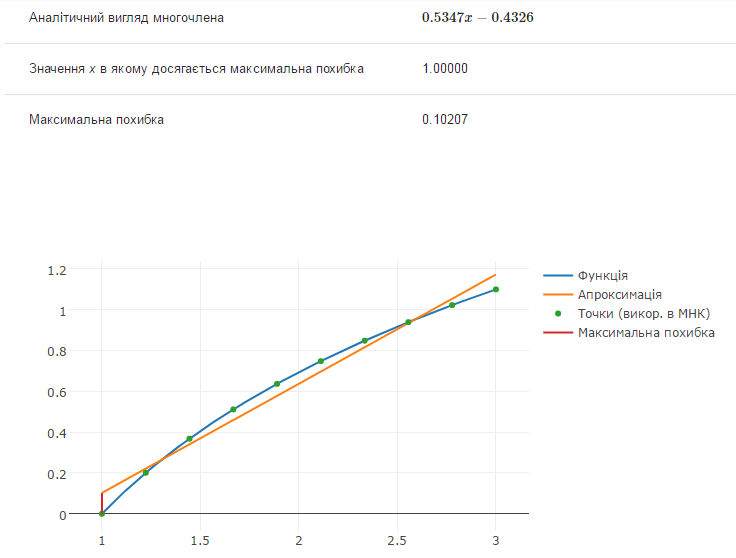
Після цього, незважаючи на спосіб, яким задали функцію (вручну чи завантажили з файлу), потрібно натиснути кнопку “ОБЧИСЛИТИ”. Коли запит обробиться на сервері, результати можна побачити на екрані.

## 3.4. Опис отриманих результатів

Приклад отриманих результатів наближення функції , лінійним многочленом на проміжку , використовуючи метод найменших квадратів. Вхідні дані:

**

Вихідні дані:

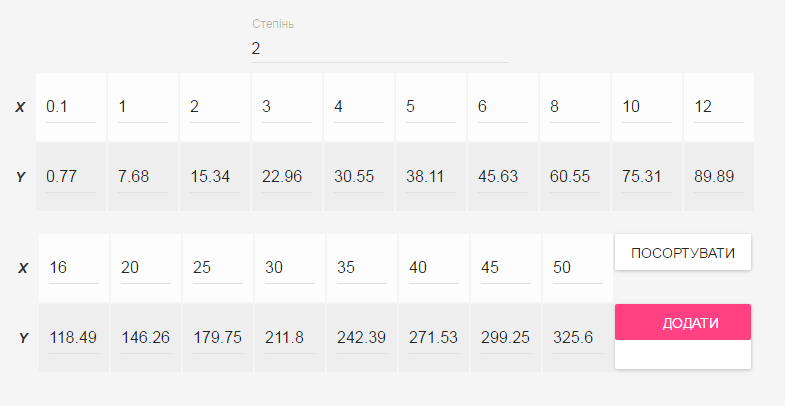


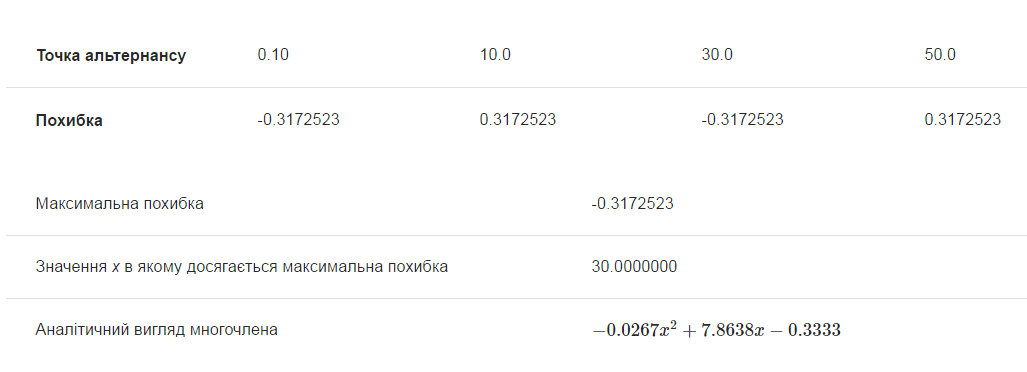
Як можна побачити з рисунку, результатом роботи програми є вивід на екран аналітичного вигляду многочлена, значення в якій досягається максимальна похибка та величини цієї похибки. Також на екран виводяться такі графіки:

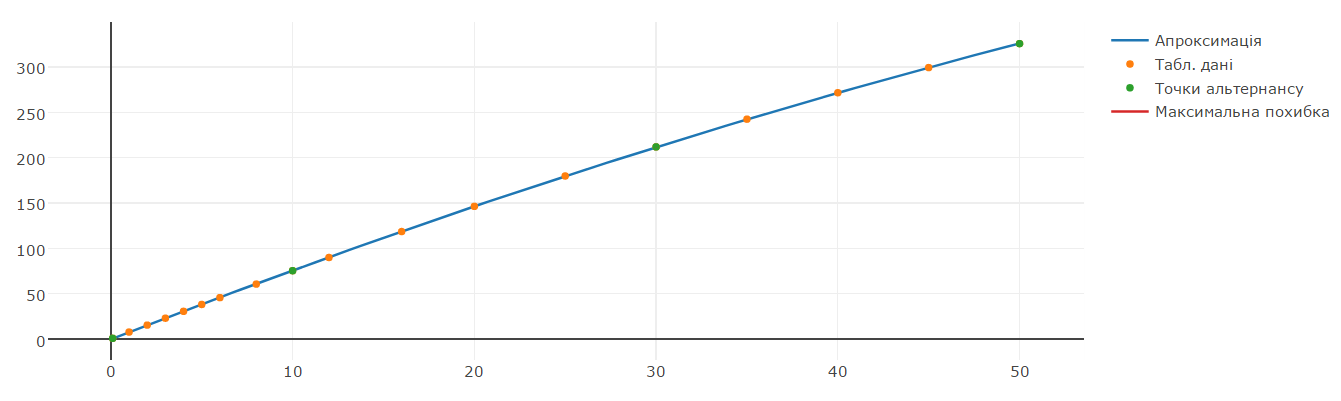
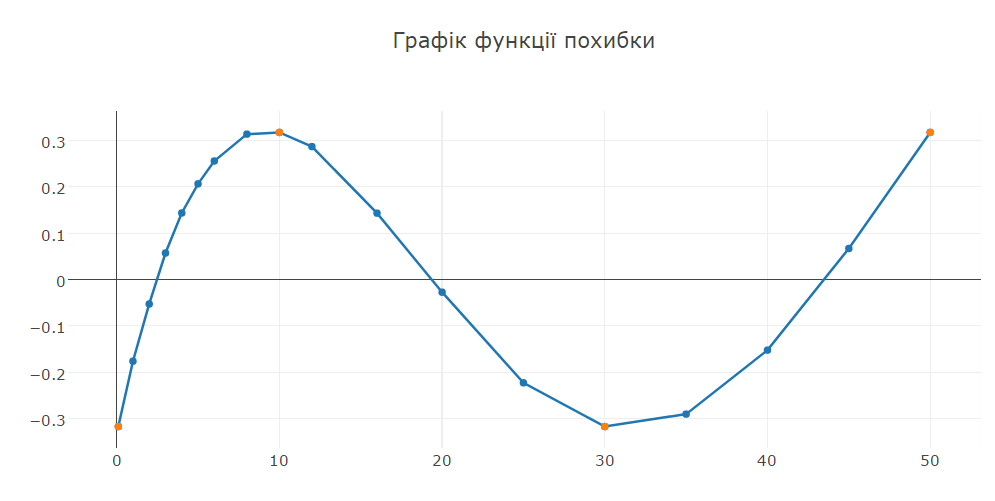
1. Графік функції що апроксимується (синій колір).
2. Графік апроксимуючого многочлена, в даному випадку лінійний (оранжевий колір).
3. Точки, які використовуються в програмі при обчислені коефіцієнтів многочлена (зелений колір).
4. Максимальна похибка (червоний колір).

Приклад отриманих результатів наближення табличної функції (рис. 1) многочленом другого степеня, використовуючи метод мінімаксного наближення.

Вхідні дані:



Вихідні дані:  


Як можна побачити з рисунку, результатом роботи програми є вивід на екран аналітичного вигляду многочлена, значення в якій досягається максимальна похибка та величини цієї похибки. Також на екран виводяться такі графіки:

1. Графік апроксимуючого многочлена, в даному випадку многочлен другого степеня (синій колір).
2. Табличні дані (оранжевий колір).
3. Точки альтернансу (зелений колір).
4. Максимальна похибка (червоний колір).
5. Графік функції похибки.

# Висновки

1. В роботі наведено поняття мінімаксного наближення функцій многочленами та метод найменших квадратів. Описано алгоритм Ремеза мінімаксної апроксимації та алгоритм Валлє-Пуссена заміни точок альтернанту.
2. Здійснено програмну реалізацію мінімаксного наближення неперервних функцій та дискретних даних многочленами довільної степені.
3. Здійснено програмну реалізацію методу найменших квадратів многочленами довільної степені.
4. Результати моделювання функціональних залежностей заданих як неперервними функціями, так і заданих у вигляді дискретних даних показали більшу ефективність (кращу точність) мінімаксного наближення над методом найменших квадратів. Тому його доцільно застосовувати, незважаючи на більшу складність при програмній реалізації.

# Додатки

## Додаток 1. (Код програми)

Підпрограма для створення початкового альтернанcу (дискретний випадок).  
Вхідні параметри:  
1) масив значень x  
2) degree – степінь апроксимуючого многочлена  
Результат виконання функції: початковий альтернанс

**def** make\_initial\_alternance(x, degree):

alternance = [x[0]] *# add first*

busy = []

for i in range(degree):

rand\_index = np.random.randint(1, len(x)-1)

while rand\_index in busy:

rand\_index = np.random.randint(1, len(x)-1)

busy.append(rand\_index)

alternance.append(x[rand\_index])

alternance.append(x[len(x)-1]) *# add last*

alternance.sort()

return alternance

В цій функції використана бібліотека *numpy(*[*http://www.numpy.org/*](http://www.numpy.org/)*).*

Підпрограма для визначення максимального значення функції в дискретних точках.  
Вхідні параметри:  
1) func – функція  
2) x\_vals – масив дискретних точок

**def** max\_error(func, x\_vals):

y\_vals = func(x\_vals)

neg\_err = min(y\_vals)

pos\_err = max(y\_vals)

if abs(neg\_err) > pos\_err:

e\_max = neg\_err

else:

e\_max = pos\_err

return e\_max

Результат виконання функції: максимальне значення заданої функції

Підпрограма для заміни точок альтернансу.  
Вхідні параметри:  
1) err\_func – функція похибки  
2) alternance – попередній альтернанс  
3) x\_vals – значення x для таблично заданої функції

Результат виконання функції: новий альтернанс

**def** change\_alternance(err\_func, alternance, x\_vals):

x\_err = x\_of\_max\_error(err\_func, x\_vals)

temp = alternance[:]

temp.append(x\_err)

temp.sort()

index\_of\_x\_err = temp.index(x\_err)

if index\_of\_x\_err != 0 and index\_of\_x\_err != (len(temp)-1):

if sign(err\_func(temp[index\_of\_x\_err])) == sign(err\_func(temp[index\_of\_x\_err-1])):

del temp[index\_of\_x\_err-1]

else: del temp[index\_of\_x\_err+1]

elif index\_of\_x\_err == 0:

if sign(err\_func(temp[index\_of\_x\_err])) == sign(err\_func(temp[1])):

del temp[1]

else:

del temp[len(temp)-1]

elif index\_of\_x\_err == (len(temp)-1):

if sign(err\_func(temp[index\_of\_x\_err])) == sign(err\_func(temp[index\_of\_x\_err-1])):

del temp[index\_of\_x\_err-1]

else:

del temp[0]

return temp

Підпрограма для побудови аналітичного вигляду многочлена.

Вхідні параметри:  
1) t – альтернанс  
2) degree – степінь апроксимуючого многочлена  
3) f\_discrete – дискретна функція

**def** pol(t, degree, f\_discrete):

x = Symbol('x')

e = Symbol('e')

vars\_str = ' '.join(['a' + str(i) for i in range(degree+1)])

variables = symbols(vars\_str)

eqs = []

for i in range(degree+2):

eqs.append(make\_eq(variables, t[i], f\_discrete(t[i])) + e)

e \*= -1

if degree % 2 == 1:

e \*= -1

solution = solve(eqs, variables + (e,))

error\_on\_iteration = solution[e]

polynom = x - x

for i, v in enumerate(variables):

polynom += solution[v] \* x\*\*i

return [polynom, error\_on\_iteration]

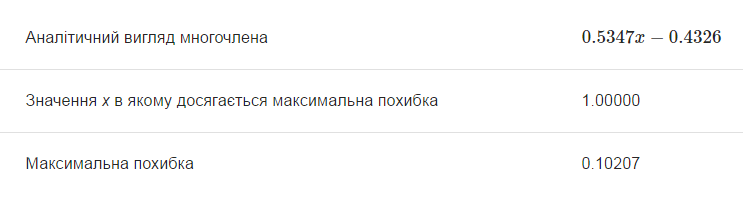
Результат виконання функції: аналітичний вигляд многочлена.

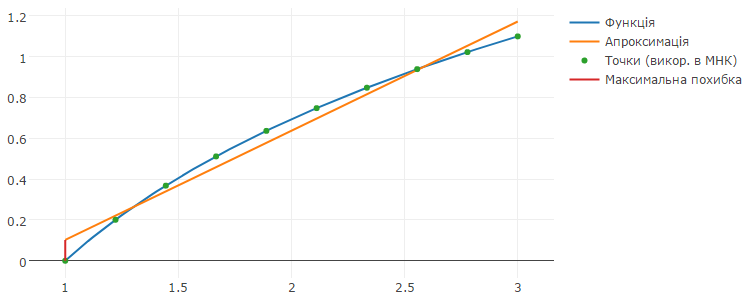
## Додаток 2. Приклади виконання програми

**Неперервний випадок**

Приклад 1. Знайдемо наближення функції , лінійним многочленом на проміжку , використовуючи метод найменших квадратів.

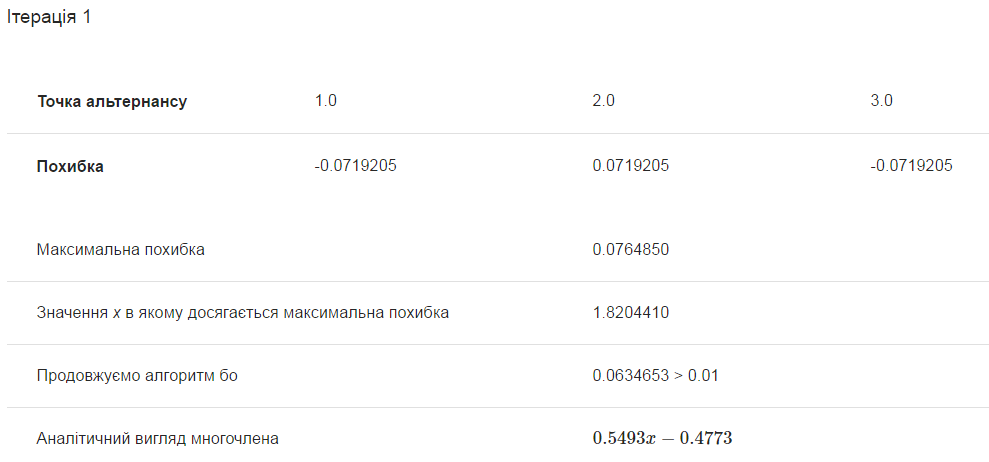
Результат роботи програми:

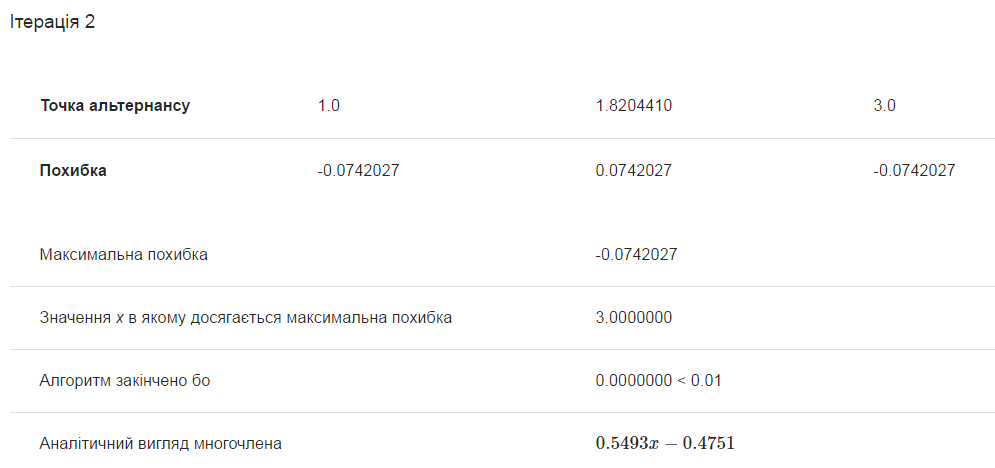


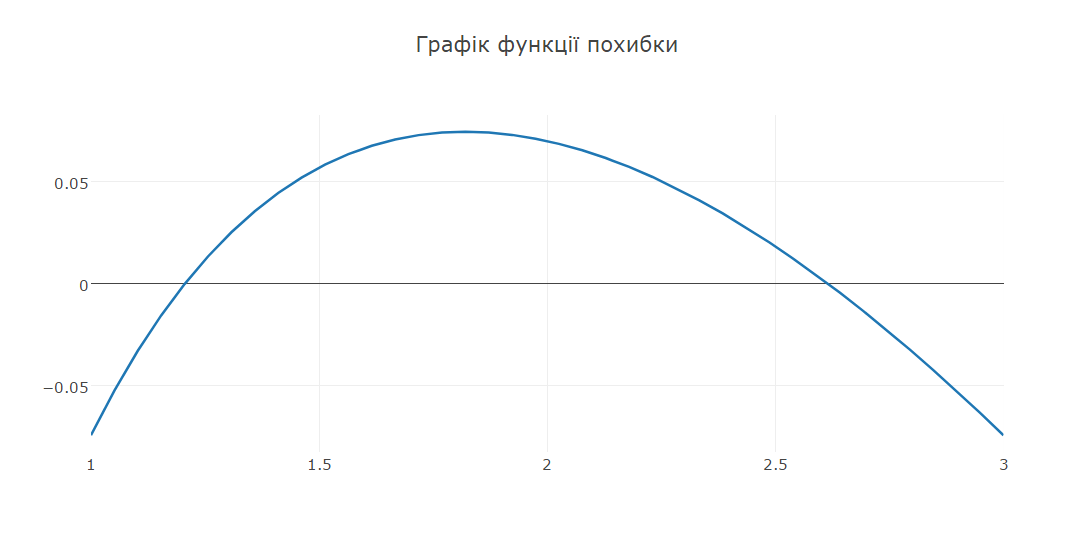


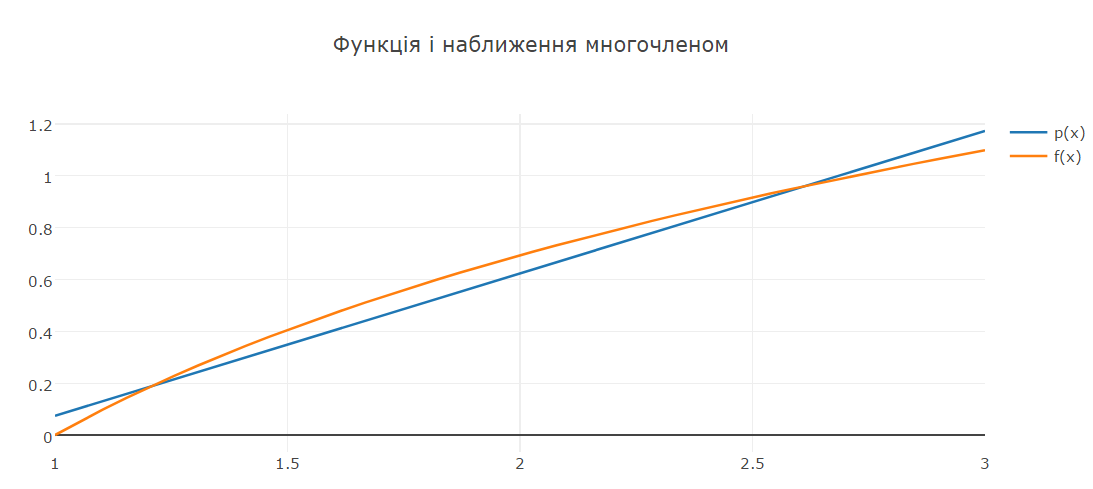
Приклад 2. Знайдемо наближення функції , лінійним многочленом на проміжку , використовуючи мінімаксний метод.

Результат роботи програми:



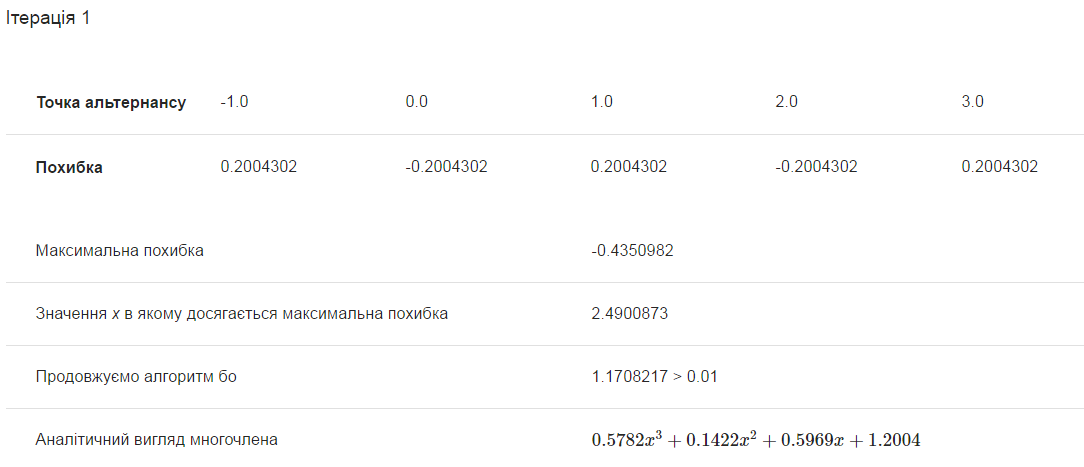


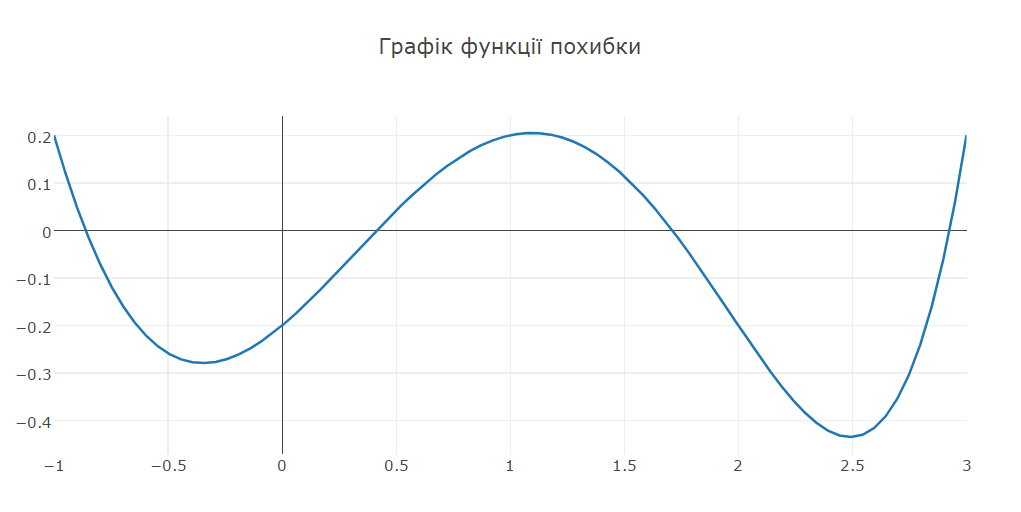


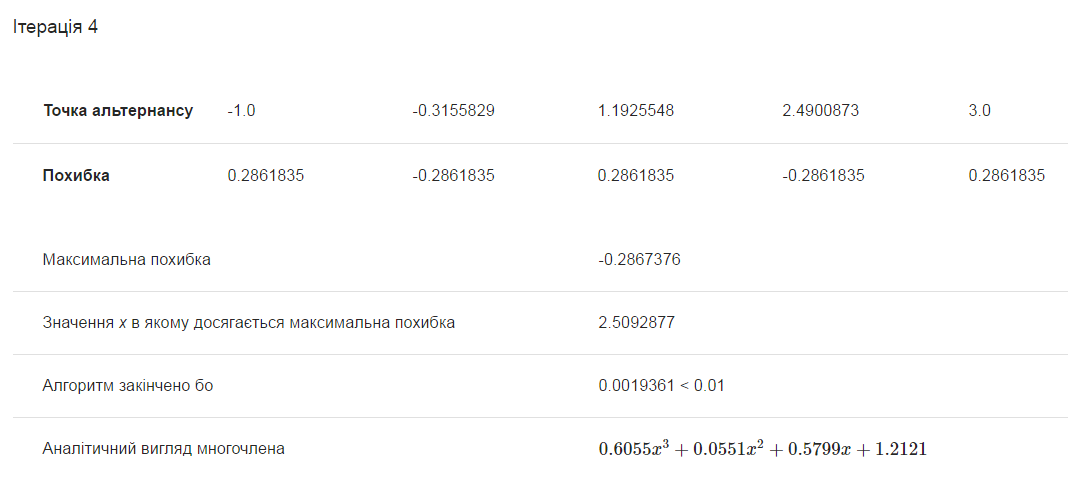


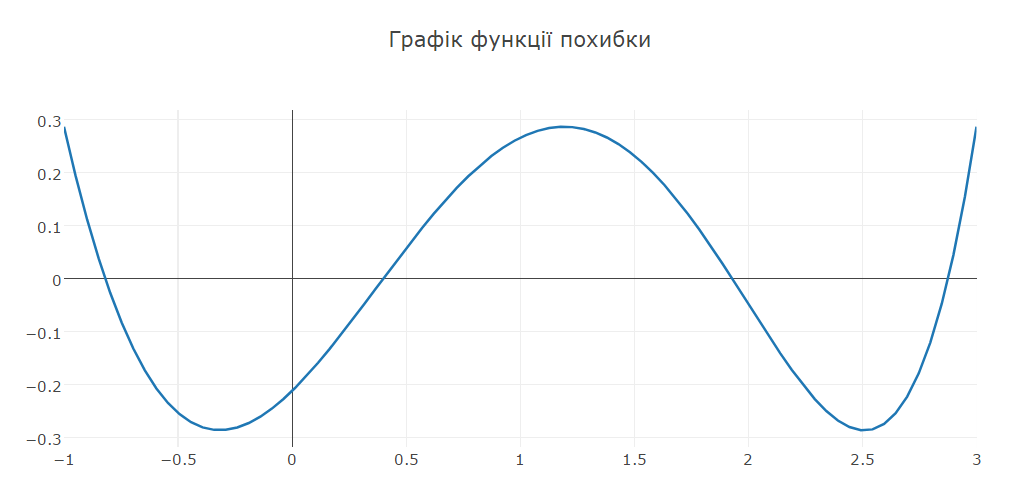
Приклад 3. Знайдемо наближення функції многочленом третього степеня на проміжку , використовуючи мінімаксний метод.

Результат роботи програми:



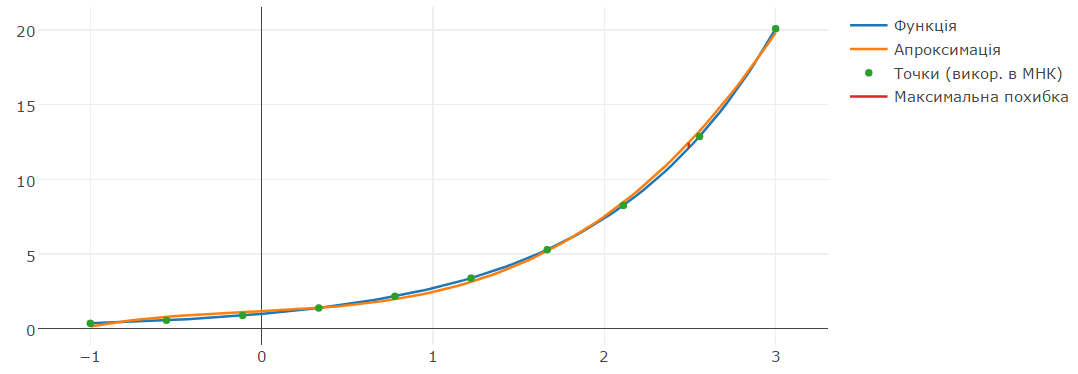
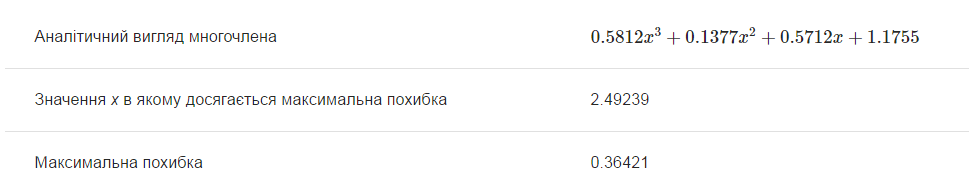






Приклад 4. Знайдемо наближення функції многочленом третього на проміжку , використовуючи метод найменших квадратів.

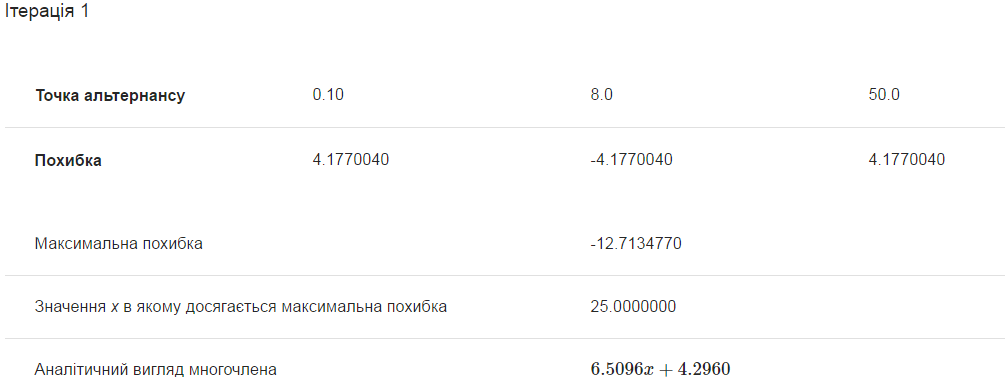
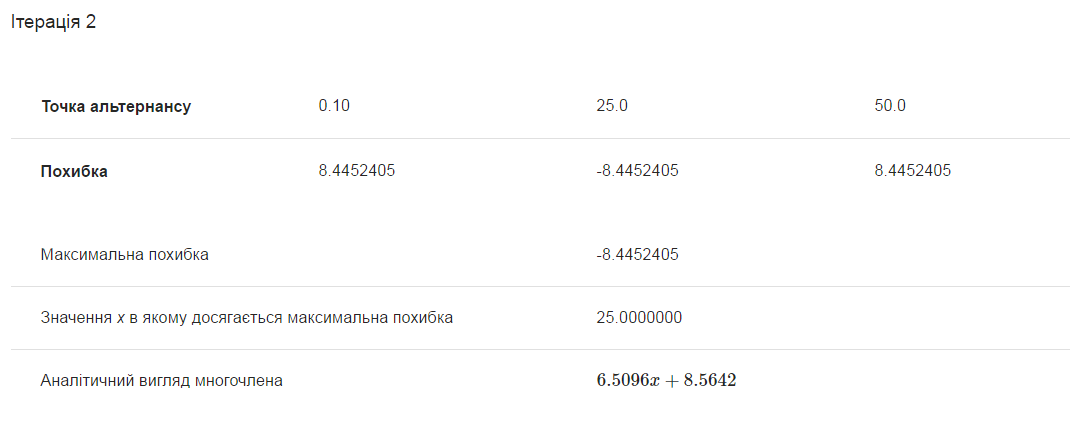
Результат роботи програми:

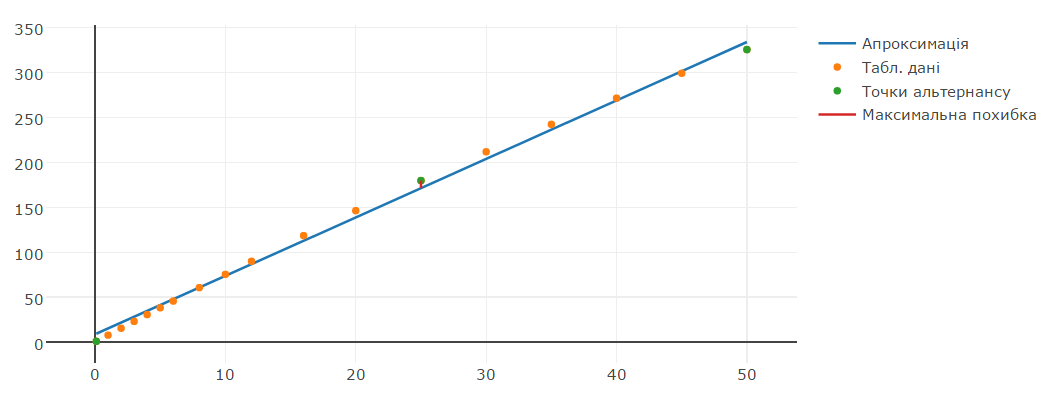


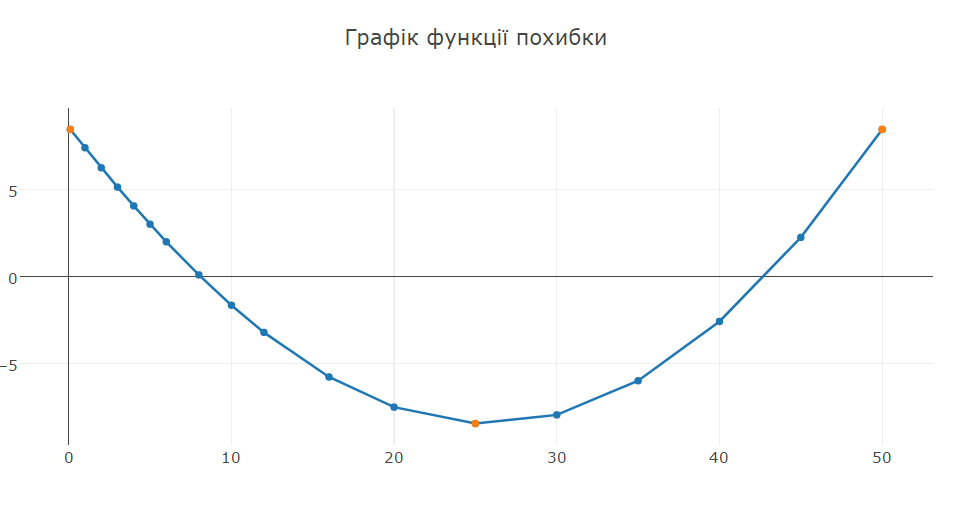
**Дискретний випадок**

Приклад 5. Знайдемо наближення многочленом першого степеня, використовуючи мінімаксний метод для експериментально отриманих залежностей: щільності кисню *ρ* від тиску *p* при температурі .

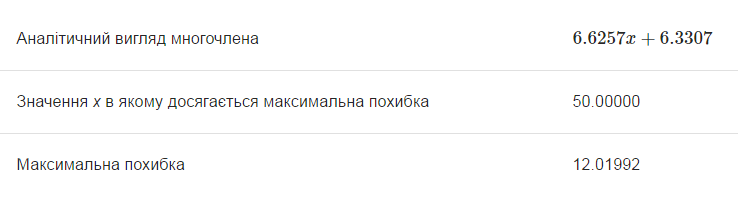
|  |  |
| --- | --- |
|  | **ρ** |
| 0.1 | 0.77 |
| 1.0 | 7.68 |
| 2.0 | 15.34 |
| 3.0 | 22.96 |
| 4.0 | 30.55 |
| 5.0 | 38.11 |
| 6.0 | 45.63 |
| 8.0 | 60.55 |
| 10.0 | 75.31 |
| 12.0 | 89.89 |
| 16.0 | 118.49 |
| 20.0 | 146.26 |
| 25.0 | 179.75 |
| 30.0 | 211.80 |
| 35.0 | 242.39 |
| 40.0 | 271.53 |
| 45.0 | 299.25 |
| 50.0 | 325.60 |

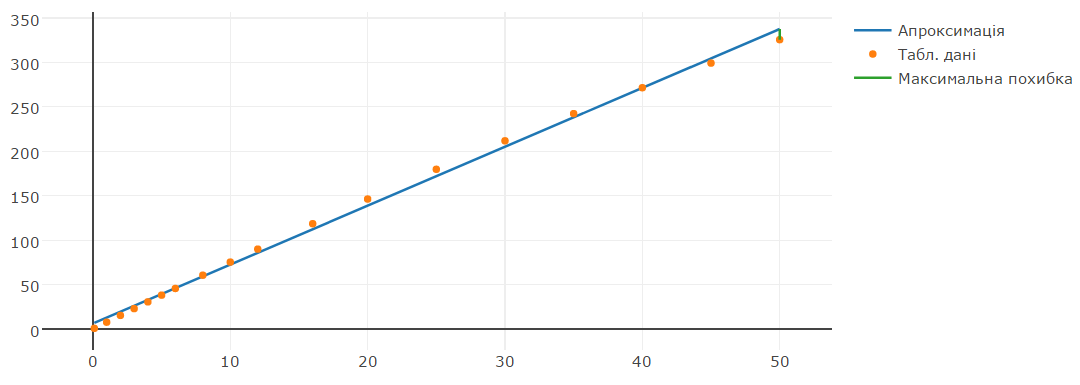
** **





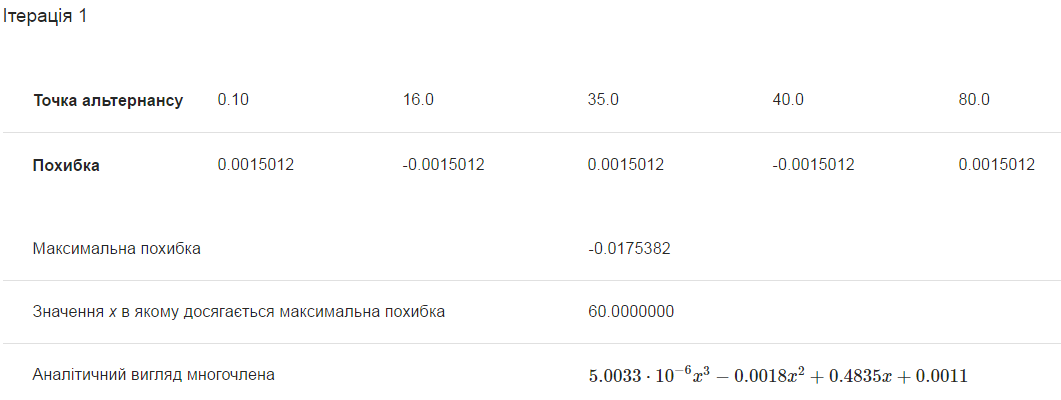
Приклад 6. Знайдемо наближення многочленом першого степеня, використовуючи метод найменших квадратів для тих самих даних які були використані у прикладі 5.

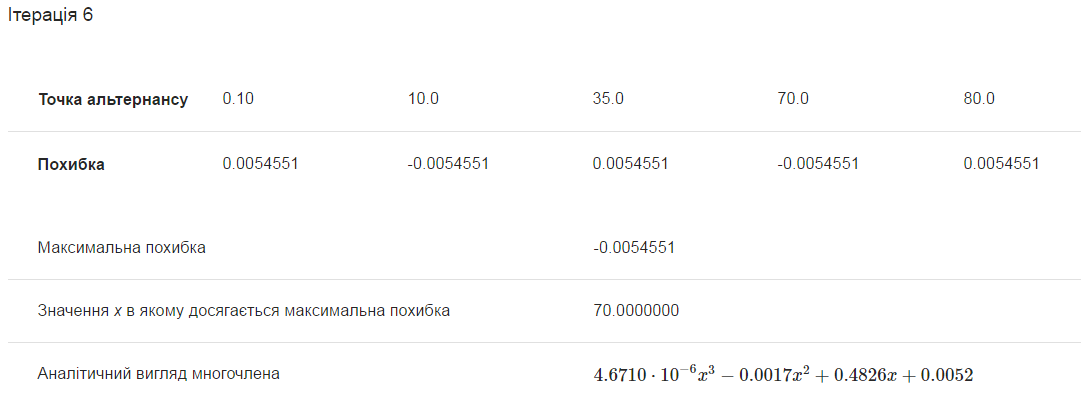


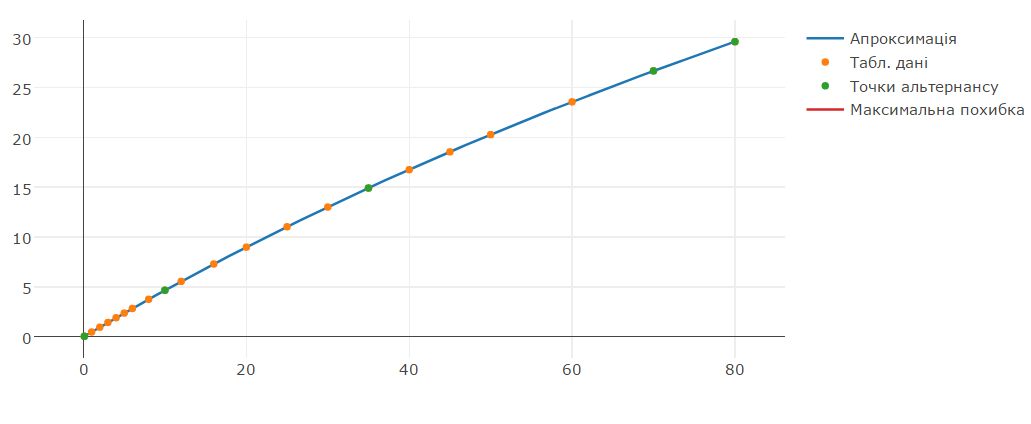


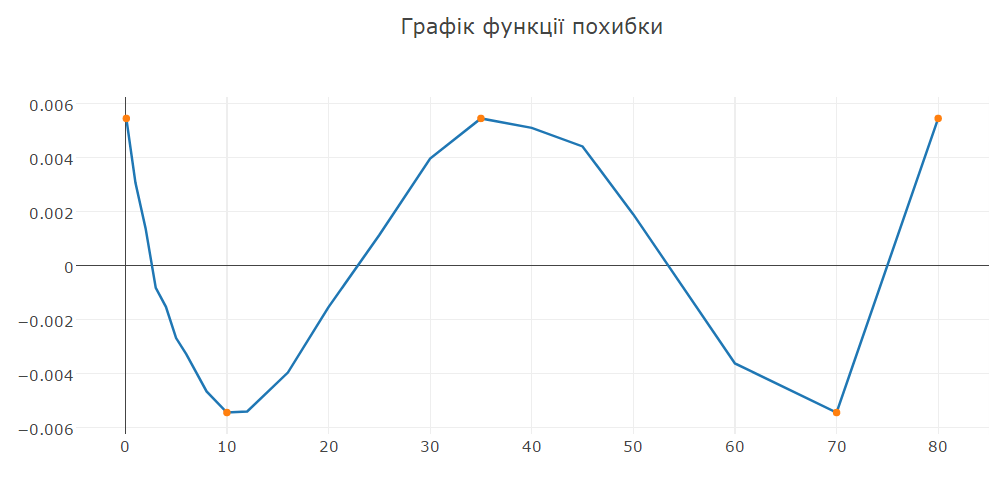
Приклад 7. Знайдемо наближення многочленом третього степеня, використовуючи мінімаксний метод для експериментально отриманих залежностей: щільності водню *ρ* від тиску *p* при температурі .

|  |  |
| --- | --- |
|  | **ρ** |
| 0.1 | 0.048 |
| 1.0 | 0.483 |
| 2.0 | 0.962 |
| 3.0 | 1.438 |
| 4.0 | 1.909 |
| 5.0 | 2.377 |
| 6.0 | 2.841 |
| 8.0 | 3.759 |
| 10.0 | 4.663 |
| 12.0 | 5.553 |
| 16.0 | 7.293 |
| 20.0 | 8.982 |
| 25.0 | 11.026 |
| 30.0 | 12.998 |
| 35.0 | 14.903 |
| 40.0 | 16.745 |
| 45.0 | 18.526 |
| 50.0 | 20.251 |
| 60.0 | 23.542 |
| 70.0 | 26.640 |
| 80.0 | 29.564 |









Приклад 8. Знайдемо наближення многочленом третього степеня, використовуючи метод найменших квадратів для тих самих даних які були використані у прикладі 7.

