Наведено означення ермітових сплайнів і рівномірної апроксимації функцій сплайнами. Побудовано ланки многочленного ермітового сплайна та сплайна з експоненціально-степеневою ланкою з чотирма параметрами. Наведено формулу для похибки рівномірної апроксимації ермітовими сплайнами. Написано програму на мові Python, яка будує ланки таких ермітових сплайнів.

The definition of Hermite splines and uniform approximation of function by splines are presented. Links of Hermite polynomial spline and spline with exponential-power link with four parameters are build. The Formula for error of uniform approximation using Hermite splines is shown.. A program in Python, which builds links of this Hermite spline is written.

Зміст

[Вступ 3](#_Toc483996497)

[1. Побудова ланки ермітового сплайна 5](#_Toc483996498)

[1.1 Означення ермітових сплайнів 5](#_Toc483996499)

[1.2 Визначення параметрів ермітового сплайна з експоненціально-степеневою ланкою 6](#_Toc483996500)

[1.3 Знаходження параметрів кубічного ермітового сплайна 8](#_Toc483996501)

[2. Рівномірна апроксимація ермітовими сплайнами 11](#_Toc483996502)

[2.1. Означення рівномірної апроксимації 11](#_Toc483996503)

[2.2. Алгоритми рівномірної апроксимації 11](#_Toc483996504)

[2.3. Похибка апроксимації ермітовими сплайнами 12](#_Toc483996505)

[3. Опис програми 14](#_Toc483996506)

[3.1 Призначення програми 14](#_Toc483996507)

[3.2 Умови застосування 14](#_Toc483996508)

[3.3 Введення вхідних даних 14](#_Toc483996509)

[3.4 Результати 16](#_Toc483996510)

[Висновки 21](#_Toc483996511)

[Список використаної літератури 22](#_Toc483996512)

[Додаток 1. (Код програми) 23](#_Toc483996513)

[Додаток 2 26](#_Toc483996514)

# Вступ

Багатьом із тих, хто стикається з науковими та інженерними розрахунками часто доводиться оперувати наборами значень, отриманих експериментальним шляхом чи методом випадкової вибірки. Як правило, на підставі цих наборів потрібно побудувати функцію, зі значеннями якої могли б з високою точністю збігатися інші отримані значення. Така задача називається апроксимацією кривої. Інтерполяцією називають такий різновид апроксимації, при якій крива побудованої функції проходить точно через наявні точки даних [9,16].

Існує також близька до інтерполяції задача, що полягає в апроксимації якої-небудь складної функції іншою, простішою функцією. Якщо деяка функція занадто складна для продуктивних обчислень, можна спробувати обчислити її значення в декількох точках, а за ними побудувати, тобто інтерполювати, простішу функцію. Зрозуміло, використання спрощеної функції не дозволяє одержати такі ж точні результати, які давала б початкова функція. Але, для деяких класів задач, досягнутий виграш у простоті і швидкості обчислень може переважити отриману похибку у результатах.

Інтерполяція сплайном краща, ніж інтерполяція многочленом, оскільки дає схожі результати навіть при менших степенях поліномів [9,17]. В бакалаврській роботі розглянута інтерполяція сплайнами, що є частинним випадком інтерполяції функції однієї змінної.

Сплайном називали гнучку металеву лінійку, яку використовували креслярі для того, щоб гладко з’єднати окремі точки на кресленні, тобто для графічного виконання інтерполяції [4]. Більше того, крива, що описує деформацію гнучкої лінійки зафіксованої в окремих точках є сплайном. Отже, ми маємо фізичну модель сплайн-функції (або навпаки сплайн-функція є математичною моделлю гнучкої лінійки).

Початок розвитку теорії інтерполяції сплайнами та й сам термін сплайн відраховують з 1946 року зі статті Шонберга [4,8]. Особливо інтенсивний її розвиток відбувся в 50-70 роки, бо традиційною прикладною сферою використання інтерполяційних сплайнів стали в цей час системи автоматизованого проектування. Однак потенційні можливості сплайнів значно ширші ніж просто опис деяких кривих. В реальному світі велика кількість фізичних процесів можуть бути описані сплайнами. В механіці це деформація гнучкої пластини чи стержня зафіксованих в окремих точках; траєкторія руху тіла, якщо сила, що діє на нього змінюється ступінчато (траєкторія штучного космічного об’єкту з активними та інерційними відрізками руху, траєкторія руху літака при ступінчатій зміні тяги двигунів та зміні профілю крила і т.д.). В термодинаміці це теплообмін в стержні складеному з фрагментів з різною теплопередачею. В хімії - дифузія через шари різних речовин. В електриці - поширення електромагнітних полів через різнорідні середовища. Тобто сплайн не надумана математична абстракція, а в багатьох випадках він є розв’язанням диференціальних рівнянь, які описують цілком реальні фізичні процеси.

Для сплайнів є характерними такі ознаки: сплайн складається з фрагментів функцій одного класу, які різняться своїми параметрами; на сусідні фрагменти в точках стикування накладаються певні умови, що зводяться до неперервності значень та перших похідних.

Сплайни, які наближають не тільки функцію, а й її похідні, називають ермітовими [3]. Для підвищення точності наближення вирази, які використовують в ланках сплайна, можуть бути експоненціальною, степеневою, логарифмічною та іншими функціями [5,7,14].

Бакалаврська кваліфікаційна робота присвячена побудові та дослідженню ермітових сплайнів з експоненціально-степеневими і многочленними лаками, а також побудові рівномірних наближень цими сплайнами.

У першому розділі наведено означення ермітових сплайнів з нелінійними за параметрами виразами в ланках. Виведено формули для параметрів ермітових сплайнів з експоненціально-степеневою ланкою та кубічних ермітових сплайнів.

У другому розділі наведено означення рівномірного наближення із заданою похибкою, описано алгоритм побудови такого наближення. Також наведено формулу оцінки похибки рівномірного наближення функцій ермітовими сплайнами з нелінійними за параметрами виразами в ланках з парною кількістю параметрів, вирази для ядер похибок ермітових сплайнів з вказаними ланками.

Третій розділ присвячений опису програми, результати роботи якої наведено в додатках.

У кінці бакалаврської роботи наведено висновки до роботи, перелік використаної літератури, текст і результати роботи програми, що реалізує апроксимацію функцій ермітовими сплайнами.

# 1. Побудова ланки ермітового сплайна

## 1.1 Означення ермітових сплайнів

Апроксимація з допомогою алгебраїчних поліномів використовується для налаштування і дослідження адаптивних телевимірювальних систем при моделюванні вхідних сигналів, які повинні максимально наближатися по своїм характеристикам до реальних сигналів [1,7]. Ця задача вимагає розробки алгоритмів наближення не тільки функцій, що описують сигнал, але і її похідних. Для підвищення точності наближення як ланки сплайна можна використовувати не тільки многочлен, але і нелінійний за параметрами вираз

(1.1)

Нехай . На множині задані значення функції і її похідних до -*го* порядку включно.

**Означення 1.** Ермітовим сплайном з парною кількістю параметрів [3,4] називають функцію

яка задовольняє систему рівнянь

. (1.3)

Через позначено функцію Хевісайда

*.*

З системи (1.3) слідує, що . Вираз називається ланкою ермітового сплайна. Інколи зручно розглядати функцію похибки у вигляді

де вагова функція

## 1.2 Визначення параметрів ермітового сплайна з експоненціально-степеневою ланкою

Запишемо систему рівнянь для знаходження ланки ермітового сплайна з парною кількістю параметрів виду

(1.4)

де , , , , .

З системи (1.4) необхідно знайти невідомі параметри *a , b, c ,d* . Для цього виразимо з першого і третього рівнянь параметр *a* .

З першого рівняння

|  |  |  |
| --- | --- | --- |
|  |  | (1.5) |

а з третього рівняння

|  |  |  |
| --- | --- | --- |
|  |  | (1.6) |

Прирівнявши рівняння (1.5) і (1.6) одержимо

|  |  |
| --- | --- |
|  |  |

Знайдемо параметр *d* , прологарифмувавши цю рівність

|  |  |  |
| --- | --- | --- |
|  |  | (1.7) |

Вираз для параметра *a* із формули (1.5) підставляємо у друге рівняння системи (1.4) , отримуємо

Врахуємо значення параметра *d* із формули (1.7)

|  |  |
| --- | --- |
|  | (1.8) |

Аналогічно вираз (1.6) підставляємо у четверте рівняння системи (1.4) замість параметра *a* .

Підставимо у цю рівність вираз для параметра із (1.7)

|  |  |
| --- | --- |
|  | (1.9) |

З рівняння (1.9) виразимо параметр *b*

|  |  |  |
| --- | --- | --- |
|  |  | (1.10) |

Виразимо з рівняння (1.8) параметр *c*

|  |  |  |
| --- | --- | --- |
|  |  | (1.11) |

Параметр *b* з рівняння (1.10) підставляємо в рівняння (1.11) , зведемо подібні доданки і отримаємо

Звідси знайдемо параметр *с*

|  |  |  |
| --- | --- | --- |
|  | , | (1.12) |

де та обчислюємо за формулами

Отже, для побудови ланки ермітового сплайна обчислюємо параметр за формулою (1.12), далі параметр за формулою (1.10), параметр за формулою (1.7) і параметр за формулою (1.5) або (1.6).

## 1.3 Знаходження параметрів кубічного ермітового сплайна

Запишемо систему рівнянь для знаходження ланки многочленного ермітового сплайна 3-го степеня з парною кількістю параметрів виду

|  |  |  |
| --- | --- | --- |
|  |  | (1.13) |

де , , , , .

З системи (1.13) необхідно знайти невідомі параметри *a ,b, c , d* . Для цього виразимо з першого і третього рівнянь параметр *d* .

З першого рівняння

|  |  |  |
| --- | --- | --- |
|  |  | (1.14) |

а з третього рівняння

|  |  |  |
| --- | --- | --- |
|  |  | (1.15) |

Прирівняємо друге і третє рівняння системи (1.13) та знайдемо параметр *a*

|  |  |  |
| --- | --- | --- |
|  |  | (1.16) |

Вираз для параметра *a* із формули (1.16) підставляємо у друге рівняння системи (1.13) і отримуємо

|  |
| --- |
|  |

Звідси знайдемо параметр *b*

|  |  |  |
| --- | --- | --- |
|  |  | (1.17) |

де вирази та обчислюємо за формулами

Аналогічно вираз для параметра *a* із формули (1.16) підставляємо у четверте рівняння системи (1.13) і отримуємо

Із останньої рівності знайдемо параметр *b*

|  |  |  |
| --- | --- | --- |
|  | , | (1.18) |

де та знаходимо за формулами

Прирівнявши рівняння (1.17) та (1.18) знаходимо параметр *c*

|  |  |  |
| --- | --- | --- |
|  |  | (1.19) |

Отже, для побудови ланки кубічного ермітового сплайна обчислюємо параметр за формулою (1.19), далі параметр за формулою (1.17) або (1.18), параметр за формулою (1.16) і параметр за формулою (1.14) або (1.15).

# 2. Рівномірна апроксимація ермітовими сплайнами

## 2.1. Означення рівномірної апроксимації

Наведемо означення рівномірної апроксимації (наближення) [2,14].

**Означення.** Апроксимація функції  ермітовим сплайном  називається рівномірною із заданою похибкою , якщо

, , , , (2.1)

де  – вага наближення, ,  [1,2].

При такій апроксимації максимальні похибки на всіх ланках крім останньої є рівні між собою і дорівнюють заданій, а на останній отримана похибка не перевищує задану.

## 2.2. Алгоритми рівномірної апроксимації

Цей алгоритм є інваріантним до виразу і кількості параметрів у ланці сплайна [15].

1. На всьому відрізку  будуємо ланку ермітового сплайна. Позначимо його ліву границю , а праву .
2. Обчислимо похибку апроксимації функції ермітовим сплайном за формулою .
3. Якщо отримана похибка менша від заданої , то наближення побудоване. Кінець.
4. Якщо отримана похибка більша від заданої , то зменшуємо довжину інтервалу, тобто зсуваємо праву межу вліво, поки похибка на цьому інтервалі стане меншою від заданої . Нехай, при *k-*му зсуві межі вліво () отримана похибка рівна , а на попередньому *k-1-*му кроці вона більша від заданої, тобто  (позначимо праву межу , ). Завжди можна знайти таку праву межу , на якій отрмана похибка апроксимації  буде як завгодно мало відрізнятись від заданої  . Цю точку  знаходимо наприклад методом дихотомії (ділення відрізка навпіл) [9,11,16].
5. Запам’ятовуємо межі і параметри ланки ермітового сплайна.
6. Лівою межею наступної ланки є права межа ланки, яка їй передувала. Спочатку правою межею можна вибрати кінець інтервалу . Далі довжину ланки можна екстраполювати точкою, значення якої обчислюють за такою формулою , де  - довжина попередньої ланки.
7. Будуємо сплайн на наступній ланці і обчислюємо похибку.
8. Якщо отримана похибка більша від заданої , то переходимо до виконання пункту 4.
9. Якщо отримана похибка менша від заданої  а також , то приймаємо, що , і переходимо до виконання пункту 7
10. В протилежному випадку, якщо , запам’ятовуємо межі та параметри ланки нелінійного ермітового сплайну. Рівномірна апроксимація із заданою похибкою побудована.

Відомо, що наведений алгоритм рівномірної апроксимації має єдиний розв‘язок, за умови коли наближувала функція  і сплайн  такі, що функція похибки , при  є неспадною функцією від . Для цього достатньо, щоб ядро наближення , при  [2,6].

## 2.3. Похибка апроксимації ермітовими сплайнами

Похибка  рівномірної апроксимації (такого наближення, при якому максимальні похибки на всіх інтервалах є однакові) нелінійними ермітовими сплайнами з парною кількістю параметрів у ланці обчислюється за формулою [1,2]

, (2.2)

де - вагова функція ,  - кількість ланок ермітового сплайна на інтервалі , - ядро похибки наближення.

Із теорем, наведених в роботах [6,7], які ще називають обмінними теоремами і властивостей ядер похибок [6] можна показати, що ядро похибки ермітового сплайна з експоненціально-степеневою ланкою



має вигляд [5,6]

. (2.3)

У нашому випадку для ланки ермітового сплайна , а , а тому ядро похибки матиме наступний вигляд

|  |
| --- |
|  |
|  |

Відомо, що ядро похибки наближення кубічним ермітовим сплайном має вигляд [3,4]

 (2.4)

# 3. Опис програми

## 3.1 Призначення програми

Програма призначена для побудови рівномірної апроксимації функцій. Ланка сплайна може мати експоненціально-степеневий вигляд або вигляд кубічного многочлена. Програма дає можливість знайти коефіцієнти ланок, побудувати графіки ланок, якими апроксимована функція, та графік похибки наближення.

## 3.2 Умови застосування

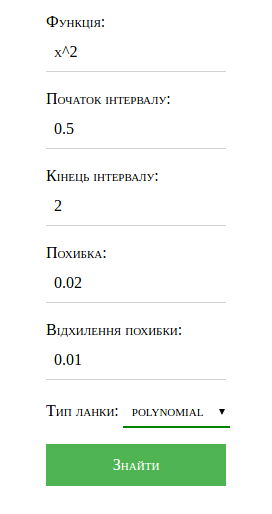
Вимоги до ПК:

* Процесор: 1 (ГГц) Intel Pentium III, 1.2 (ГГц) Intel Celeron або краще
* ОЗП: 1 (ГБ) або більше
* Відеокарта: 128 (МБ) або більше
* ОС: Windows 7/8/10, Linux Ubuntu 14.04/16.04 LTS

Для того щоб скористатися програмою необхідно мати сучасний браузер та доступ до мережі інтернет. Далі достатньо відкрити файл *Approximation.html* з кореневої папки програми.

## 3.3 Введення вхідних даних

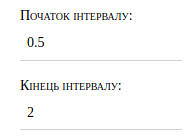
Для знаходження рівномірної апроксимації функції, спершу потрібно ввести цю функцію у відповідне поле.



Приклади вводу функцій:

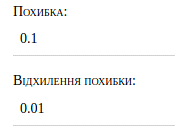
|  |  |
| --- | --- |
|  | X^2 або x\*\*2 |
|  | 1/(x^2 +2) |
|  | x\*cos(x)^2 |
|  | 1/(ln(x)) |

В полях початок та кінець інтервалу потрібно ввести межі відповідного інтервалу на якому наближується функція



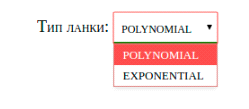
Похибка — допустиме абсолютне значення відхилення ланки сплайна від функції що наближується.

Відхилення похибки — допустиме відносне відхилення похибки наближення, при якому задовільняється точність.



Для прикладу, як бачимо з рисунку похибка рівна 0.1 а відхилення похибки 0.01, отже значення похибки апроксимації може відрізнятися на 1% від отриманого при обчисленні.

Після цього потрібно вибрати тип ланки сплайна, якою ми будемо наближувати функцію.



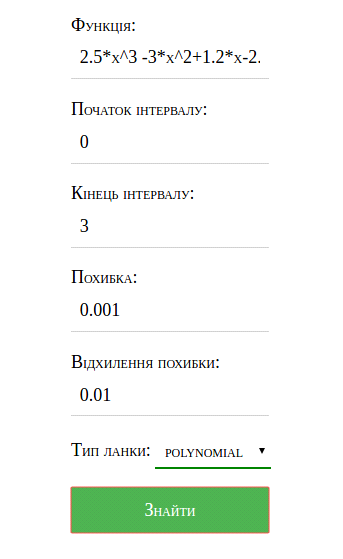
Рівномірну апроксимацію реалізовано двома типами ланок, а саме: ланкою у вигляді кубічного многочлена та експоненціально-степеневою ланкою.

Після введення всіх вхідних даних потрібно натиснути на кнопку “ЗНАЙТИ”, після опрацювання вхідних даних,результати програми можна побачити на екрані.

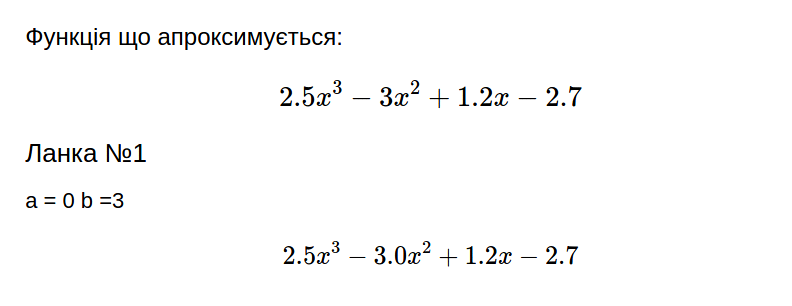
## 3.4 Результати

Приклад отриманих результатів наближення функції ( ) на інтервалі , використовуючи ланку у вигляді кубічного многочлена.

Вхідні дані:

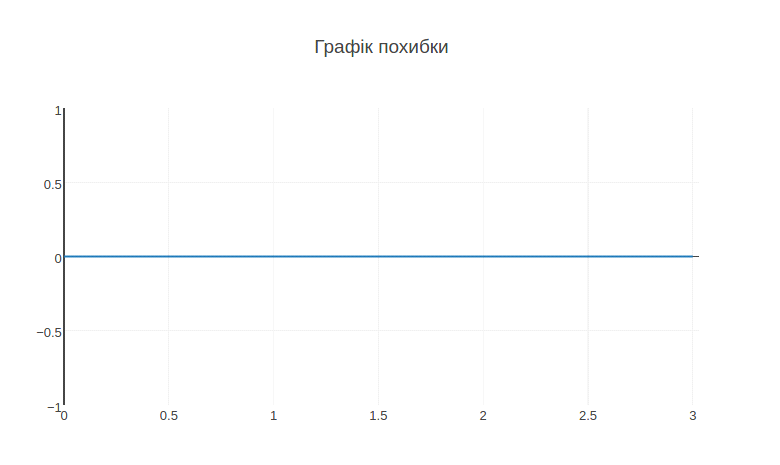


Вихідні дані:



Як можна побачити з рисунку, результатом роботи програми є вивід на екран аналітичного вигляду функції що апроксимується та відповідний аналітичний вигляд ланки сплайна. Оскільки функція є многочленом третього степеня то вона апроксимується точно, однією ланкою. Також на екран виводяться графіки ланок сплайну та графік функції похибки.

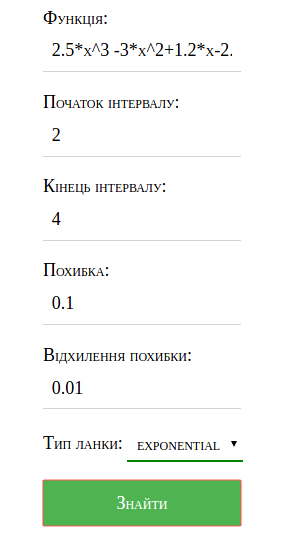




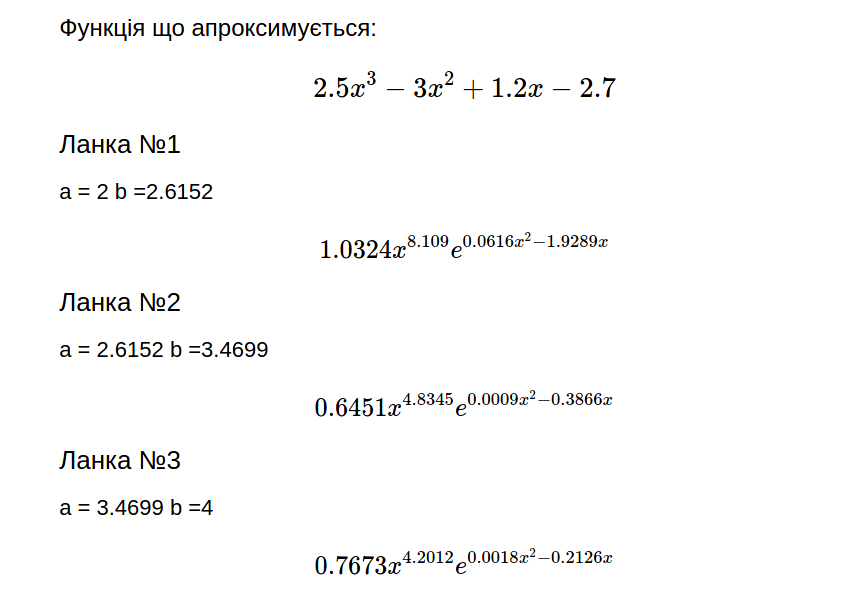
Як і можна було очікувати графік функції та сплайна повністю наклалися, а функція похибки рівна нулю.

Приклад отриманих результатів наближення цієї ж функції на інтервалі [2,4], використовуючи експоненціально-степеневу ланку.

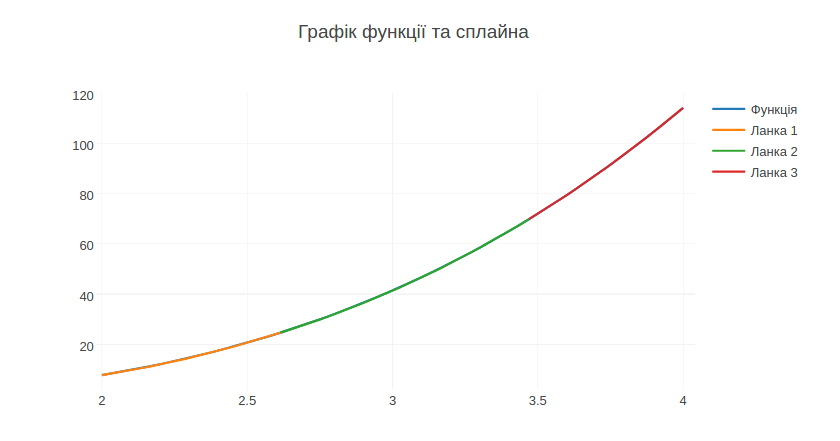
Вхідні дані:



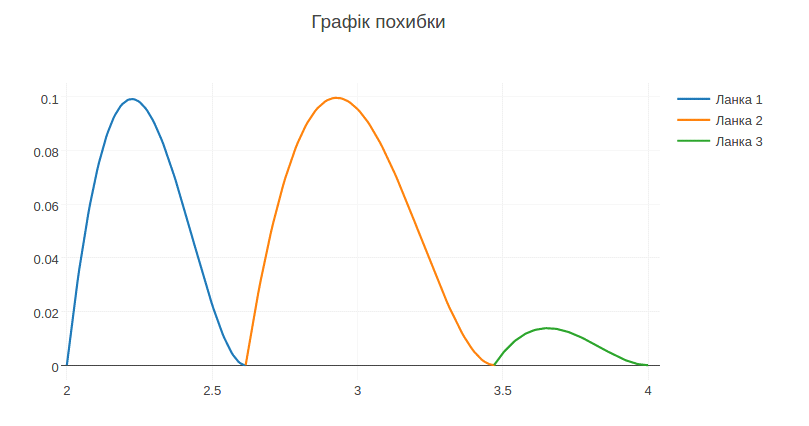
Вихідні дані:



Оскільки вигляд функції відрізняється від вигляду ланки сплайна то наближення здійснюється із заданою похибкою,будуючи кожну ланку з однаковою похибкою. Як бачимо з рисунку початок кожної наступної ланки є кінцем попередньої.



Кожна ланка сплайна виділена іншим кольором на графіку.



Оскільки ланок було декілька, то всі ланки (крім останньої) мають похибку наближення задану при введені, у цьому прикладі (0.1)

# Висновки

1. Наведено означення ермітових сплайнів з нелінійними за параметрами виразами в ланках.
2. Побудовано ермітові сплайни з многочленними (3-го степеня) та експоненціально-степеневими ланками з чотирма параметрами.
3. Наведено означення рівномірної апроксимації функцій ермітовими сплайнами з заданою похибкою.
4. Програмно реалізовано побудову ланок цих ермітових сплайнів і алгоритм рівномірної апроксимації функції із заданою похибкою.
5. Як видно із отриманих результатів деякі функції доцільно наближати ермітовими сплайнами з експоненціально-степеневою ланкою, бо при цьому досягається краща точність наближення ніж при апроксимації многочленним ермітовим сплайном.

# Список використаної літератури

1. Пизюр Я. В., Попов Б. О. Рівномірне наближення ермітовими сплайнами з парною кількістю параметрів// Контрольно-вимірювальна техніка.– 1993.–№50.–С. 8 – 13.
2. Попов Б. А. Равномерное приближение сплайнами.– Киев: Наук. думка, 1989.– 272 с.
3. Де Бор К. Практическое руководство по сплайнам.– М.: Радио и связь, 1985.–304 с.
4. Завьялов Ю. С., Квасов Б. И., Мирошниченко В. Л. Методы сплайн-функций.– М.: Наука 1980.–352 с.
5. Пизюр Я.В. Приближение эрмитовым сплайном с звеном в виде произведения степенной и экспоненциональной функции / «Применение информатики и вычислительной техники при решении народнохозяйственных задач». Тезисы докладов Республиканской конференции молодых ученых и специалистов. – Минск, 1989.– С. 112.
6. Попов Б. О. Чисельні методи рівномірного наближення сплайнами: Конспект лекцій. – Львів: ЛДУ, 1992. – 92 с.
7. Пизюр Я. В., Попов Б. А. Свойства степенных эрмитовых сплайнов// Контрольно-измерительная техника.– 1988.–№44.– С. 21 – 28.
8. Альберг Дж., Нильсон Э., Уолш Дж. Теория сплайнов и ее приложения: Пер. с англ.– М.: Мир, 1972.– 320 с.
9. Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. Численные методы.– М.: Наука, 1987.– 600 с.
10. Василенко В. А. Сплайн-функции: Теория, алгоритмы, программы.– Новосибирск: Наука, 1983.– 216 с.
11. Кутнів М.В. Чисельні методи: Навчальний посібник.– Львів: Видавництво «Растр-7», 2010.– 288 с.
12. Лигун А.А., Сторчай В.Ф. О наилучшем выборе узлов при приближении функций локальными эрмитовыми сплайнами. // Укр. мат. журн. -1980. -32, №6. –с.824-830.
13. Лигун А. А., Шумейко А. А. Оптимальный выбор узлов при приближении функций сплайнами // Докл. АН УССР.- 1984.- А, № 6.- C. 18-22.
14. Пізюр Я.В. Наближення функцій ермітовими сплайнами з експоненціальними ланками // Вісник НУ «Львівська політехніка», «Фізико-математичні науки», -2007.- №566.- С. 68-75.
15. ПізюрЯ.В. Модифікований алгоритм балансної апроксимації функцій ермітовими сплайнами // Шоста відкрита наукова конференція професорсько-викладацького складу Інституту прикладної математики та фундаментальних наук. (Львів, 11-13 жовтня 2007). Тези доповідей. – Львів: Видавництво Національного університету “Львівська політехніка”, 2007. – С. 48.
16. Самарский А. А., Гулин А. В., Численные методы.– М.: Наука, Гл. ред. физ.-мат. лит., 1989.– 432 с.
17. Корнейчук Н. П. Сплайны в теории приближения.– М.: Наука, 1984.– 352 с.

# Додаток 1. (Код програми)

Підпрограма для відправлення запиту на сервер та відображення отриманих результатів, а саме: аналітичний вигляд ланок сплайна , графік наближення функції та графік функції похибки.

document.getElementById("go").addEventListener("click", function() {

document.getElementById('spiner').style = 'display: block'

func = document.getElementById('func').value

mu = +document.getElementById('mu').value

start = +document.getElementById('start').value

end = +document.getElementById('end').value

eps\_mu = +document.getElementById('eps\_mu').value

typ = document.getElementById('typ').value

iterations = document.getElementById('iterations')

fetch("http://localhost:5000/diplom", {

method: 'POST',

body: JSON.stringify({

func,mu,start, end, eps\_mu,typ

})

}).then(res => res.json())

.then(res => {

console.log(res)

document.getElementById('spiner').style = 'display: none'

let func\_div = document.getElementById('formula\_div')

func\_div.innerHTML = 'Функція що апроксимується:' + '$$' + res[0].formula\_func + '$$'

MathJax.Hub.Queue(["Typeset", MathJax.Hub, func\_div])

let formulas = res.map((iter, i) => {

return `

<h3>Ланка №${i+1}</h2>

<h4>a = ${iter.start} b =${iter.end}</h4>

<h4 id="formula${i}">$$${iter.formula}$$</h4> `

})

iterations.innerHTML = formulas.join('')

MathJax.Hub.Queue(["Typeset", MathJax.Hub, iterations])

Plotly.newPlot('plot', [{

x: res[0].func\_x,

y: res[0].func\_y,

name: 'Функція'

}, ...res.map((el, i) => ({

x: el.x,

y: el.y,

name: 'Ланка ' + (i + 1)

}))], {

title: 'Графік функції та сплайна',

xaxis: {

range: [start , end ]}

});

Plotly.newPlot('error\_plot', res.map((el, i) => ({

x: el.residual\_x,

y: el.residual\_y,

name: 'Ланка ' + (i + 1)

})), {

title: 'Графік похибки',

xaxis: {

range: [start , end ]}

});

})

Підпрограма для опрацювання введених користувачем даних та реалізації алгоритму рівномірної апроксимації.

def main (start,end,mu,eps\_mu,func,typ):

z\_a = start

z\_b = end

global a,b,c,d

x\_val\_func = np.linspace(start,end,50)

y\_val\_func = np.vectorize(lambdify(x,func))(x\_val\_func)

temp\_a=start

temp\_b=end

dfunc=diff(func)

array = []

while(True):

f0=func.subs(x,z\_a)

f1=func.subs(x,z\_b)

d0=dfunc.subs(x,z\_a)

d1=dfunc.subs(x,z\_b)

if typ == "polynomial":

system = Matrix(( (z\_a\*\*3,z\_a\*\*2,z\_a,1,f0), (z\_b\*\*3,z\_b\*\*2,z\_b,1,f1),

(3\*z\_a\*\*2,2\*z\_a,1,0,d0),(3\*z\_b\*\*2,2\*z\_b,1,0,d1)))

roots = solve\_linear\_system(system, e, f, g, r)

a,b,c,d=N(roots[e],5),N(roots[f],5),N(roots[g],5),N(roots[r],5)

result = a\*x\*\*3 +b\*x\*\*2+c\*x+d

if typ == "exponential":

A = ( (d0\*z\_a\*ln(z\_b/z\_a)-f0\*ln(f1/f0)) /

(f0\*(2\*ln(z\_b/z\_a)\*z\_a\*\*2 - z\_b\*\*2 +z\_a\*\*2))) /\

((f1\*(z\_a\*ln(z\_b/z\_a)-z\_b+z\_a)\*(2\*ln(z\_b/z\_a)\*z\_b\*\*2-z\_b\*\*2+z\_a\*\*2)) -

(( z\_a\*ln(z\_b/z\_a)-z\_b+z\_a)\*(2\*ln(z\_b/z\_a)\*z\_b\*\*2-z\_b\*\*2+z\_a\*\*2))/

(z\_b\*ln(z\_b/z\_a)-z\_b+z\_a )\*(2\*ln(z\_b/z\_a)\*z\_a\*\*2-z\_b\*\*2+z\_a\*\*2))

B = (z\_b\*d1\*ln(z\_b/z\_a)-ln(f1/f0)\*f1) /\

(f1\*(2\*ln(z\_b/z\_a)\*z\_b\*\*2-z\_b\*\*2+z\_a\*\*2)\*

(f1\*(2\*ln(z\_b/z\_a)\*z\_a\*\*2-z\_b\*\*2+z\_a\*\*2)\*(z\_b\*ln(z\_b/z\_a)-z\_b+z\_a)-1))

c = N(A-B,5)

b = N(((z\_b\*d1\*ln(z\_b/z\_a))/(f1\*(z\_b\*ln(z\_b/z\_a)-z\_b+z\_a))) -

c\*((2\*ln(z\_b/z\_a)\*z\_b\*\*2-z\_b\*\*2+z\_a\*\*2)/(z\_b\*ln(z\_b/z\_a)-z\_b+z\_a))-

ln(f1/f0)/(z\_b\*ln(z\_b/z\_a)-z\_b+z\_a),5)

d = N((ln(f1/f0)-b\*(z\_b-z\_a)-c\*(z\_b\*\*2-z\_a\*\*2))/(ln(z\_b/z\_a)),5)

a = N(f0/(exp(b\*z\_a+c\*z\_a\*\*2)\*z\_a\*\*d),5)

result = a\*exp(b\*x+c\*x\*\*2)\*(x\*\*d)

max\_mu = residual(z\_a,z\_b,func,result)

if (max\_mu > mu):

z\_b = (temp\_a + temp\_b)/ 2

temp\_b=z\_b

continue

elif ((max\_mu < mu\*(1-eps\_mu))&(temp\_b < end)&(max\_mu!=0)):

print("max = {0}, a = {1}, b = {2}".format(max\_mu,z\_a,z\_b))

temp = z\_b

z\_b = (3\*temp\_b-temp\_a)/2

temp\_b=z\_b

temp\_a=temp

continue

else:

t=z\_a

h=(z\_b-z\_a)/100

array\_x = []

array\_y = []

residual\_x=[]

residual\_y=[]

while(t <= z\_b):

array\_x.append(t)

array\_y.append(result.subs(x,t))

residual\_x.append(t)

if abs(result.subs(x,t)-func.subs(x,t))<0.00001:

residual\_y.append(0)

else:

residual\_y.append(abs(result.subs(x,t)-func.subs(x,t)))

t+=h

array\_x.append(z\_b)

residual\_x.append(z\_b)

array\_y.append(result.subs(x,z\_b))

if abs(result.subs(x,t)-func.subs(x,t))<0.00001:

residual\_y.append(0)

else:

residual\_y.append(abs(result.subs(x,t)-func.subs(x,t)))

if (typ=="polynomial"):

latech = latex(pres(a)\*x\*\*3 + pres(b)\*x\*\*2 + pres(c)\*x + pres(d))

else:

latech = latex(pres(a)\*(exp( pres(b)\*x + pres(c)\*x\*\*2 ))\*(x\*\*pres(d)))

array.append({

'start': pres(z\_a),

'end': pres(z\_b),

'formula': latech,

'x': list(map(lambda x: float(x), array\_x)),

'y': list(map(lambda y: float(y), array\_y)),

'func\_x': list(x\_val\_func),

'func\_y': list(y\_val\_func),

'residual\_x': list(map(lambda x: float(x), residual\_x)),

'residual\_y': list(map(lambda y: float(y), residual\_y)),

'formula\_func': latex(simplify(func))

})

if(z\_b == end):

break

z\_a = z\_b

z\_b = end

temp\_a = z\_a

temp\_b = z\_b

continue

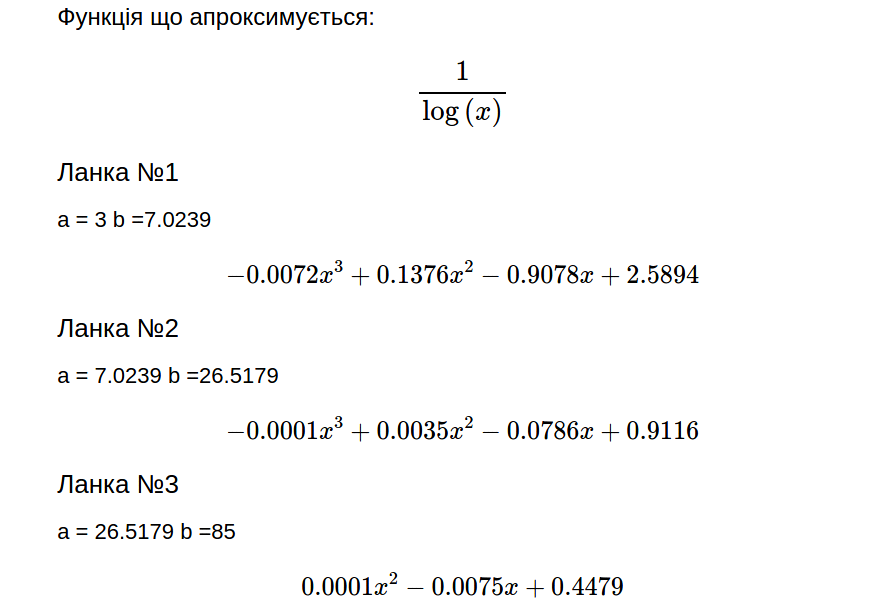
return array

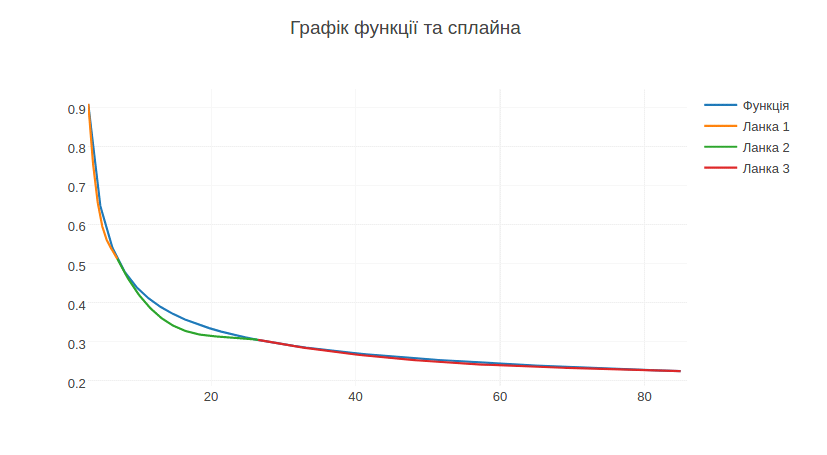
# Додаток 2

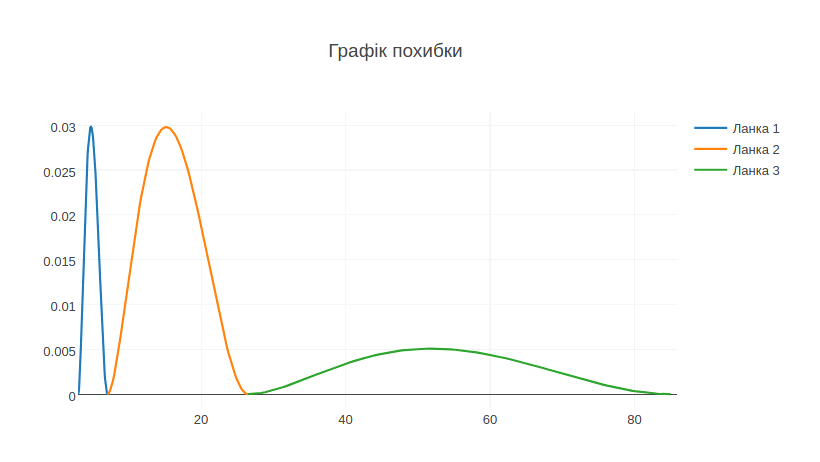
Приклади виконання програми

Приклад 1. Побудуємо рівномірну апроксимацію з ланкою у вигляді кубічного многочлена, функції 1/(ln(x)) на інтервалі [3,85], з похибкою 0.03 та відхиленням похибки 0.01

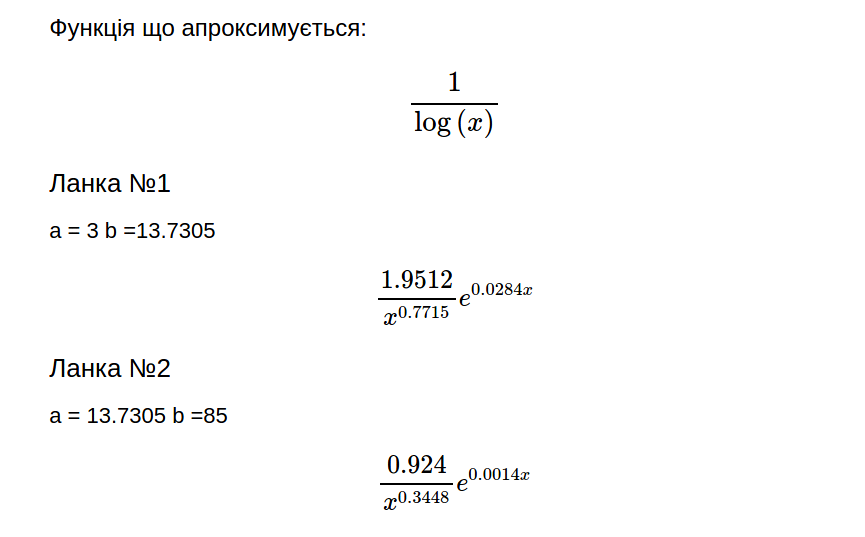
Результат роботи програми:

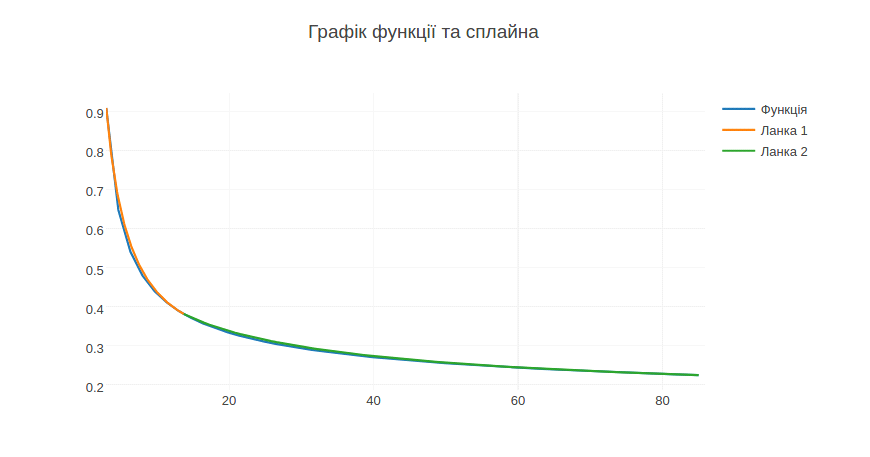


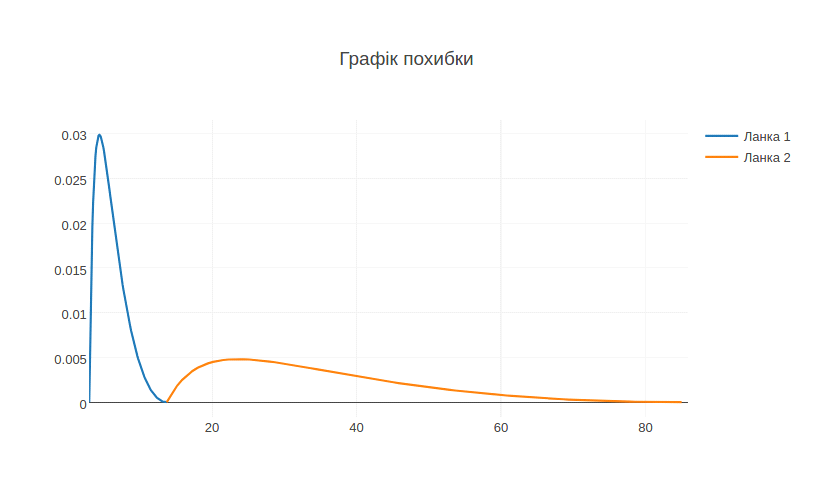




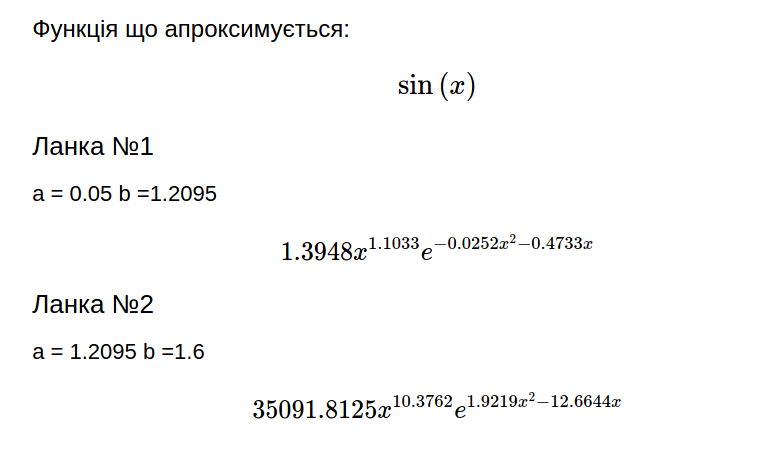
Приклад 2. Побудуємо рівномірну апроксимацію цієї ж функції з такими ж початковими умовами але експоненціально-степеневою ланкою.

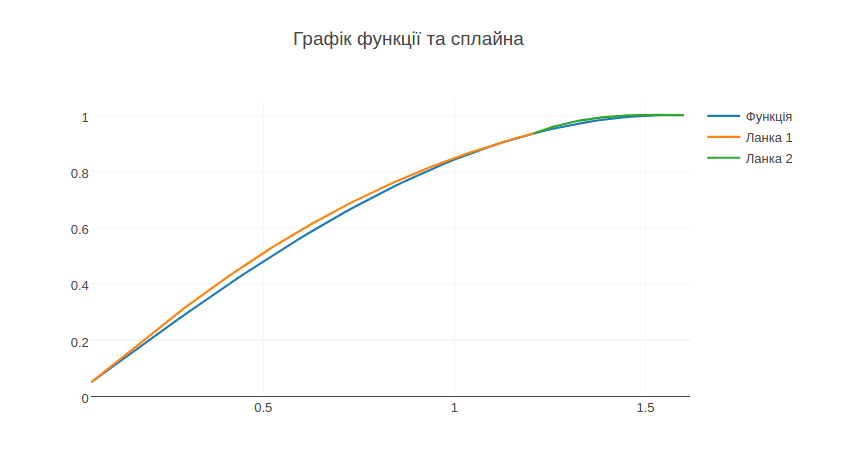


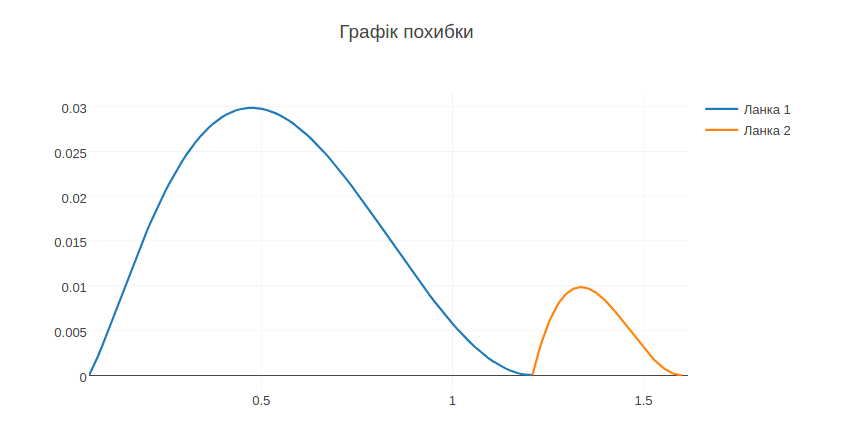




Приклад 3. Побудуємо рівномірну апроксимацію з експоненціально-степеневою ланкою функції sin(x) на інтервалі [0.05,1.6], з похибкою 0.03 та відхиленням похибки 0.01







Приклад 4. Побудуємо рівномірну апроксимацію з ланкою у вигляді кубічного многочлена, функції на інтервалі , з похибкою 0.003 та відхиленням похибки 0.01

