МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Курсова робота
з курсу "Чисельні методи"
на тему:
"Мінімаксна апроксимація функцій многочленними сплайнами"

Виконав:
ст. гр. ПМКМ-11
Левантович Богдан
Перевірив:
доц. каф. ПМ
Пізюр Я.В.

Львів 2018

Зміст

Вступ **3**

1 Найкраще чебишовське наближення **4**
1.1 Схема Ремеза побудови чебишовського наближення 5
1.2 Алгоритм Валле-Пуссена заміни точок альтернансу 7

2 Наближення сплайнами **9**
2.1 Рівномірне наближення сплайнами. Точність 9
2.2 Алгоритм побудови рівномірного наближення чебишовськими сплайнами 11

3 Опис програми та отриманих результатів **12**
3.1 Вхідні дані 12
3.2 Вихідні дані 12

Висновки **13**

Список використаної літератури **14**

4 Додатки **15**
4.1 Текст програми 15
4.2 Приклади виконання програми 17

Вступ

Необхідність моделювання функціональних залежностей виникає в багатьох галузях прикладної математики та інформатики. При розв'язуванні багатьох задач науково-технічного 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.

Особливістю моделей, які побудовані на основі мінімаксного (чеби- шовського) наближення, є те, що вони забезпечують найменшу із можливих похибку апроксимації при заданій кількості параметрів. Саме це зумовлює їхню практичну цінність в отриманні розв'язків задач, для яких важлива висока точність відтворення функціональної залежності. Також інтерес викликає побудова мінімаксних сплайнів, які дають змогу суттєво зменшити величину похибки для задач, які вимагають високої точності.

1 Найкраще чебишовське наближення

За теоремою Вейєрштрасса для довільних неперервних на обмеженому проміжку $[a, b]$ функцій $f(x)$ та $w(x) > 0$ і довільного $\epsilon > 0$ можна знайти такий многочлен $P_m(x)$, що

$$|\rho(x)| = \frac{|f(x) - P_m(x)|}{w(x)} < \epsilon, \quad x \in [a, b].$$

Ясно, що найменше при цьому значення степеня m многочлена $P_m(x)$ суттєво залежить від способу наближення. Серед усіх способів наближення функцій найменшу похибку а, значить, і найменше m при заданому ϵ, дає найкраще чебишовське наближення.

Вираз $F(A, x) \in F(B, x)$, для якого максимальне значення абсолютної величини зваженої похибки (**??**) досягає на проміжку $[a, b]$ найменшого значення

$$\min_{c \in B} \max_{x \in [a,b]} \frac{|f(x) - F(C, x)|}{w(x)} = \max_{x \in [a,b]} \frac{|f(x) - F(A, x)|}{w(x)}, \tag{1}$$

звемо найкращим чебишовським зваженим (з вагою $w(x)$) наближенням функції $f(x)$ за допомогою виразу виду $F(A, x)$ на проміжку $[a, b]$.

У цій курсові розглянуто лише найкращі чебишовські наближення. Слова “чебишовські” і “зважені” будемо часом пропускати. При $w(x) = 1$ маємо найкраще абсолютне наближення, при $w(x) = f(x)$ - найкраще відносне.

Величину (1) називатимемо мінімальним (зваженим) відхиленням і позначаємо $E(f, W) \equiv \mu_0$; $E(f, 1) \equiv E(f) \equiv \Delta_0$ - мінімальне абсолютне відхилення; $E(f, f) \equiv \delta_0$ - мінімальне відносне відхилення.

Далі розглянемо властивості найкращих наближень многочленом.

Теорема 1. *Для будь-яких неперервних на проміжку $[a, b]$ функцій $f(x)$ та $w(x) > 0$ і довільного ϵ, існує единий многочлен $P_m(x)$ степеня m, що має найменше відхилення $E(f, w)$.*

Теорема 2. *Нехай на проміжку $[a, b]$ задано неперервні функції $f(x)$ та $w(x) > 0$. Тоді для того, щоб деякий многочлен $P_m(x)$ степеня не вище m був многочленом найкращого чебишовського зваженого наближення функції $f(x)$ на проміжку $[a, b]$ необхідно і достатньо, щоб на цьому проміж-*

ку знайшлась принаймні одна система з $m + 2$ точок
$T = \{t_k\}_{k=0}^{m+1}$ $a \le t_0 < t_1 < t_2 < \ldots \le t_{m+1}$, *у яких зважена різниця* (??) *почергово набувала значень різних знаків і досягала за модулем найбільшого на* $[a, b]$ *значення тобто:*

$$\rho(t_0) = -\rho(t_1) = \rho(t_2) = \ldots = (-1)^{m+1}\rho(t_{m+1}) = \pm E(f, W). \qquad (2)$$

Система точок T із теореми 2 зветься системою точок (чебишовського альтернансу). Для побудови многочлена найкращого наближення необхідно визначити ці точки. Точно визначити їх значення можна тільки у часткових випадках.

1.1 Схема Ремеза побудови чебишовського наближення

У загальному випадку процес знаходження точок T побудовано на ітераційних методах. Найбільше практичне значення мають методи розроблені українським математиком Є.Я. Ремезом. Коротко розглянемо один з методів. Він складається з таких етапів.

1. З проміжку $[a, b]$ вибираємо початкове наближення T_0 до альтернансу

   $$T : t_0^{(0)} < t_1^{(0)} < t_2^{(0)} < \ldots < t_{m+1}^{(0)}.$$

   Можна, наприклад, прийняти $t_k^{(0)} = a + \frac{(b-a)k}{m+1}$.

2. Здійснюємо чебишовську інтерполяцію для множини точок
   $T_j = \{t_k\}_{k=0}^{m+1}, t_k^{(j)} < t_{k+1}^{(j)}, k = \overline{0, m}$, тобто визначаємо коефіцієнти многочлена $P_m^i(x)$ і величину μ_j, для яких виконуються умови $\rho(t_k^{(j)}) = (-1)^k \mu_k \quad k = \overline{0, m+1}$. Для знаходження вказаних величин розв'язуємо систему рівнянь:

$$\begin{cases} f(t_0^{(j)}) - a_0 - a_1 t_0^{(j)} - \ldots - a_m (t_0^{(j)})^m = \mu_j w(t_0^{(j)}), \\ f(t_1^{(j)}) - a_0 - a_1 t_1^{(j)} - \ldots - a_m (t_1^{(j)})^m = -\mu_j w(t_1^{(j)}), \\ \ldots\ldots\ldots\ldots\ldots\ldots \qquad\qquad \ldots\ldots\ldots\ldots\ldots\ldots \\ f(t_{m+1}^{(j)}) - a_0 - a_1 t_{m+1}^{(j)} - \ldots - a_m (t_{m+1}^{(j)})^m = (-1)^{m+1} \mu_j w(t_{m+1}^{(j)}). \end{cases} \tag{3}$$

Система є системою $m+2$ алгебраїчних рівнянь з $m+2$ невідомими: $a_0, a_1, \ldots, a_m$ та μ.

3. Перевіряємо виконання рівності

$$|\mu_j| = \max_{x \in [a,b]} |f(x) - P_m^{(j)}(x)| / w(x) \equiv \rho_j. \tag{4}$$

Якщо рівність виконується, то у відповідності з теоремою 2 многочлен $P_m^{(j)}(x)$ і є шуканий многочлен найкращого наближення. При машинній реалізації алгоритму перевірку рівності заміняють перевіркою нерівності

$$\rho_j - |\mu_j| \le \epsilon |\mu_j|, \tag{5}$$

де ϵ - допустима відносна помилка у визначенні похибки наближення. Можна, наприклад, прийняти $\epsilon = 10^{-2}$ чи $\epsilon = 10^{-3}$.

4. Якщо умова 4 чи 5 не виконується, то приймаємо $j := j + 1$ і вибираємо наступне (уточнене) наближення до точок чебишовського альтернансу (наступний V-альтернанс). Далі виконання алгоритму повторюється починаючи з п.2.

При обчисленнях на ЕОМ у цьому пункті іноді додатково перевіряються умови

$$\left| t_k^{(j-1)} - t_k^j \right| < \eta, \quad k = \overline{0, m+1},$$

де η - допустима помилка у визначенні точок альтернансу. Якщо остання нерівність справедлива для всіх точок $k = \overline{0, m+1}$, то вважаємо, що многочлен найкращого наближення знайдено.

1.2 Алгоритм Валле-Пуссена заміни точок альтернансу

Існує кілька методів заміни точок альтернансу. Можлива заміна однієї або кількох точок одночасно. Найпростішим алгоритмом є алгоритм Є.Я. Ремеза з одноточковою заміною (алгоритм Валлє-Пуссена). Опишемо цей алгоритм.

Нехай при виконанні п.3 знайдена точка $\tilde{x}$, для якої справедливо $\rho_j = |\rho(\tilde{x})|$. Можливі три випадки взаємного розміщення точок V-альтернансу та точки $\tilde{x}$:

1. $t_0^{(j)} < \tilde{x} < t_{m+1}^{(j)}$
2. $\tilde{x} < t_0^{(j)}$
3. $\tilde{x} > t_{m+1}^{(j)}$

Розглянемо спосіб заміни точок V-альтернансу у кожному випадку.

1. Знайдемо ціле число v таке, що $t_v^{(j)} < \tilde{x} < t_{v+1}^{(j)}$. Якщо $\operatorname{sign}(\rho(\tilde{x})) = \operatorname{sign}(\rho(t_{m+1}^{(j)}))$, то приймаємо $t_v^{(j+1)} := \tilde{x}$, у протилежному випадку $t_{v+1}^{(j+1)} := \tilde{x}$. Решту точок V-альтерanансу не змінюємо.

2. Якщо $\operatorname{sign}\rho(\tilde{x}) = \operatorname{sign}\rho(t_0^{(j)})$, то приймаємо $t_0^{(j+1)} := \tilde{x}$, а решту точок V-альтернансу не змінюємо. Якщо це не так, то заміняємо усі точки альтернансу за формулами:

$$t_0^{(j+1)} := \tilde{x}; \quad t_k^{(j+1)} := t_{k-1}^{(j)}, \quad k = \overline{1, m+1}.$$

   У цьому випадку із V-альтернансу виключається точка $t_{m+1}^{(j)}$

3. Якщо $\operatorname{sign}\rho(\tilde{x}) = \operatorname{sign}\rho(t_{m+1}^{(j)})$, то приймаємо $t_{m+1}^{(j)} := \tilde{x}$. і решту точок V-альтернансу не змінюємо. Якщо це не так, то замінюємо усі точки V-альтернансу за формулами:

$$t_k^{(j+1)} := t_{k+1}^{(j)}, \quad k = \overline{0, m}; \quad t_{m+1}^{(j+1)} := \tilde{x}.$$

   У цьому випадку із V-альтернансу виключається точка $t_0^{(j)}$.

Отже черговий V-альтернанс відрізняєтся від попереднього тим, що точка $\tilde{x}$, у якій досягається максимум абсолютної величини зваженої похибки, вводиться у V-альтернанс замість однієї із старих точок. Відомо, що алгоритм Валле-Пуссена для заміни точок альтернансу при знаходженні найкращого наближення попередньої функції многочленом на проміжку $[a,b]$ збігається незалежно від початкового наближення до точок альтернансу. Більш того у цьому випадку цей алгоритм збігається зі швидкістю гометричної прогресії у тому сенсі, що знайдуться такі числа A та $0 < q < 1$, що відхилення $E^{(k)}(f,W)$ многочлена $P_m^{(k)}(x)$ від функції $f(x)$ будуть задовольняти нерівності

$$E^{(k)}(f,W) - E(f,W) \leq Aq^k; \quad k = 1, 2, \ldots$$

Фактична швидкість збіжності залежить від диференціальних властивостей функції та використовуваного алгоритму заміни точок альтернансу. Відомо, що коли $f(x) \in C^{m+1}[a,b], w(x) = 1$ або $w(x) = f(x)$ і $f^{(m+1)}(x)$ не змінює знак при $x \in [a,b]$, то граничні точки проміжку $[a,b]$ є точками альтернансу. Тому у цьому випадку алгоритм Валле-Пуссена для наближення многочленами невисоких степенів $m = \overline{0,2}$ практично не програє у швидкості порівняно з іншими алгоритмами типу Є.Я. Ремеза. Зазначимо, що всі перелічені властивості найкращого чебишовського наближення неперервної при $x \in [a,b]$ функції $f(x)$ многочленом справедливі також і для наближення табличної функції. Більш того, при заміні неперервної функції її значенями в точках $x_k = a + \frac{(b-a)k}{N}$ різниця між відповідними відхиленнями при $N \to \infty$ прямує до нуля.

2 Наближення сплайнами

2.1 Рівномірне наближення сплайнами. Точність

На проміжку $[a, b]$ розглянемо множину точок $Z = z_{i}{}_{i=0}^{r}$ $a = z_0 < z_1 < \ldots < z_r = b$. Якщо на кожному проміжку $[z_{i-1}, z_i]$ функцію $f(x)$ наближати за допомогою виразу $F(A_i, x)$ виду

$$F(A, x) = F(a_0, a_1, \ldots, a_m; x), \tag{6}$$

, то на всьому проміжку $[a, b]$ функція $f(x)$ буде наближена сплайном

$$S(F, x) = \sum_{i=1}^{r} F(A_i, x)\theta((x - z_{i-1})(z_i - x)) \tag{7}$$

де

$$\theta(x) = \begin{cases} 0 & \text{при } x < 0 \\ 1 & \text{при } x \geq 0 \end{cases} - \text{функція Хевісайда}$$

Сплайн 7, кожна ланка якого $F(A_i, x), \overline{1, r}$ є найкращим чебишовським наближенням функції $f(x)$ на проміжку $[z_{i-1}, z_i]$, звемо чебишовським сплайном. Можа розглядати многочлений, раціональний чи нелінійний чебишовський сплайн.

Наближення функції $f(x)$ сплайном $S(F, x)$ звемо рівномірним наближенням із заданою похибкою μ, якщо

$$\mu_i = max_{x \in [z_{i-1}, z_i]} |f(x) - S(F, x)|/w(x) = \mu, i = \overline{1, r-1}, \mu_r \leq \mu \tag{8}$$

Наближення функції $f(x)$ сплайном $S(F, x)$ називаємо рівномірним наближенням із заданою кількість r ланок, якщо

$$max_{x \in [z_0, z_1]} \frac{|f(x) - S(F, x)|}{w(x)} = max_{x \in [z_{i-1}, z_i]} \frac{|f(x) - S(F, x)|}{w(x)}, i = \overline{2, r} \tag{9}$$

З умов рівномірності 8 чи 9 визначаються границі ланок (вузли Z сплайна $S(F, x)$. Рівномірні наближення неперервної функції чебишовським сплайном є і оптимальні у тому сенсі, що при заданій похибці одержуємо мінімально можливу кількість ланок, а при заданій кількості ланок - мінімально можливу похибку. Ця властивість справедлива і для наближення деякими многочленним сплайнами степеня m, якщо $f^{(m+1)(x)} \neq 0$ при $x \in [a, b]$. В таких випадках говорять також про оптимальний розподіл вузлів сітки. Рівномірне наближення із заданою кількістю ланок будемо іноді називати просто рівномірним наближенням сплайнами. Встановимо точність рівномірного наближення сплайнами

Теорема. Нехай $f(x) \in C^{m+2}[a, b], w(x) \in C^1[a, b]$,

$$\eta(f, F) = \eta(f(x), F) \in C^1[a, b], w(x) > 0, \eta(f, F) \neq 0$$

при $x \in [a, b]$ і максимальна зважена похибка $\mu_i = max_{x \in [z_{i-1}, z_i]} \frac{|f(x) - S(F, x)|}{w(x)}$

Наближення сплайном $S(F, x)$ функції $f(x)$ на кожному проміжку $[z_{i-1}, z_i]$ може бути представлена у вигляді

$$\mu_i = \frac{1}{\lambda_i} \frac{|\eta(f(\xi_i), F)|}{w(\xi_i)} \Delta x_i^{m+1},$$

де $\Delta x_i = z_i - z_{i-1}, \xi_i \in (z_{i-1}, z_i), \lambda_i$ - константа не залежна від $f(x), \lambda_i > 0, i = \overline{1, r}$. Тоді при $r \to \infty$ похибка μ рівномірного наближення функції $f(x)$ сплайном $S(F, x)$ з r ланками визначається на проміжку $[a, b]$ за формулою

$$\mu = S_r^{m-1} \left(\int_a^b |\eta(f, F)/w(x)|^{\frac{1}{m+1}} dx \right)^{m+1} \left[1 + O\left(\frac{b-a}{r} \right) \right],$$

де $S_r = \sum_{j=1}^{r} \lambda_j^{1/m+1)}$

2.2 Алгоритм побудови рівномірного наближення чебишовськими сплайнами

Наведемо алгоритм рівномірного наближення чебишовськими сплайнами із заданою похибкою. Алгоритм не залежить від виду сплайна.

1. Будуємо ланку нелінійного чебишовського сплайна на всьому інтервалі $[a, b]$. Ліва границя $z_e = a$, права $z_p = b$.

2. Знаходимо похибку наближення $\mu_1 = max \left| \frac{f(x)-S(A,x)}{w(x)} \right|$.

3. Якщо $\mu_1 < \mu$, то наближення побудоване. Кінець

4. Якщо $\mu_1 > \mu$, то зсуваємо праву границю інтервалу вліво, поки похибка на даному інтервалі не стане меншою від заданої похибки μ. Допустимо, що при k-му зсуві границі вліво (т. z^-) похибка рівна $\mu_k < \mu$, а на попередньому кроці $\mu_{k-1} > \mu$ (права границя $z^+, z^+ > z^-$). Тоді можна знайти таку праву границю $z_p \in [z^-, z^+]$, при якій похибка μ^* буде як завгодно мало відрізнятись від заданої $|\mu - \mu^*| < \epsilon (\epsilon = O(\mu))$. Точку z_p можна знайти одним із відомих способів, наприклад методом ділення відрізка навпіл або методом хорд.

5. Запам'ятовуємо границі ланки і параметри чебишовського сплайна.

6. Лівою границею наступної ланки є права границя попередньої ланки. Правою границею можна завжди вважати точку b, але можна також екстраполюватися точкою $z_p = z_p + \Delta z$, де Δz - довжина попередньої ланки.

7. Будуємо сплайн і знаходимо похибку.

8. Якщо $\mu_1 > \mu$, то переходимо до пункту 4.

Очевидно, що описаний алгоритм приводить до єдиного рішення, якщо наближувана функція $f(x)$ і сплайн $S(A, x)$ такі, що функція похибки $\rho(b) = max_{x \in [z_e, b]} |(f(x) - S(A, x))/w(x)|$ є неспадною функцію від b. Для цього достатньо, щоб ядро наближення $\eta(f, F) \neq 0$ при $x \in [a, b]$.

3 Опис програми та отриманих результатів

Мета програми: знаходження найкращого чебишовського наближення сплайнами для заданої функції і допустимої похибки на кожній ланці сплайна.
Програма написана на мові програмування *Python* з використанням таких бібліотек: *Sympy, Numpy, Plotly*.

3.1 Вхідні дані

1. Початок інтервалу.
2. Кінець інтервалу.
3. Степінь многочлена.
4. Функція для апроксимації.
5. Точність (за замовчуванням 10^{-2}).
6. Допустима похибка на одній ланці сплайна.

3.2 Вихідні дані

1. Коефіцієнти многочлена і похибки на кожній ланці сплайна.
2. Графік похибоки.
3. Графік сплайна і функції.

Висновки

У цій курсовій я розглянув найкраще чебишовське наближення многочленами сплайнами. Написав програму для знаходження ланок цього сплайна при заданій допустимій похибці. Також в програмі реалізув побудову графіків похибки та наближення сплайном, вивід максимальної похибки на кожній ланці, коєфіцієнтів ланок сплайна та їх інтервали.

Список використаної літератури

1. Демьянов В.Ф., Малоземов В.Н. Введение в минимакс. -М.: Наука, 1972. - 368 с.

2. Попов Б.А. Равномерное приближение сплайнами. -Киев: Наук. думка, 1989. - 272 с.

3. Попов Б.А., Теслер Г.С. Приближение функций для технических приложений. - Киев: Наук. думка, 1980. - 352 с.

4. Ремез Е.Я. Основы численных методов чебышовского приближения. Киев: Наук. думка, 1969, - 623 с.

5. Попов Б.О. Чисельні методи рівномірного наближення сплайнами. Конспект лекцій. -Львів: ЛДУ, 1992. - 92 с.

6. Самарский А.А., Гулин А.В. Численные методы. -М.: Наука, 1989. - 432 с.

7. https://plot.ly/ - для побудови графіків

8. http://www.sympy.org/ - для розв'язування систем

9. http://www.numpy.org/ - для наукових розрахунків

4 Додатки

4.1 Текст програми

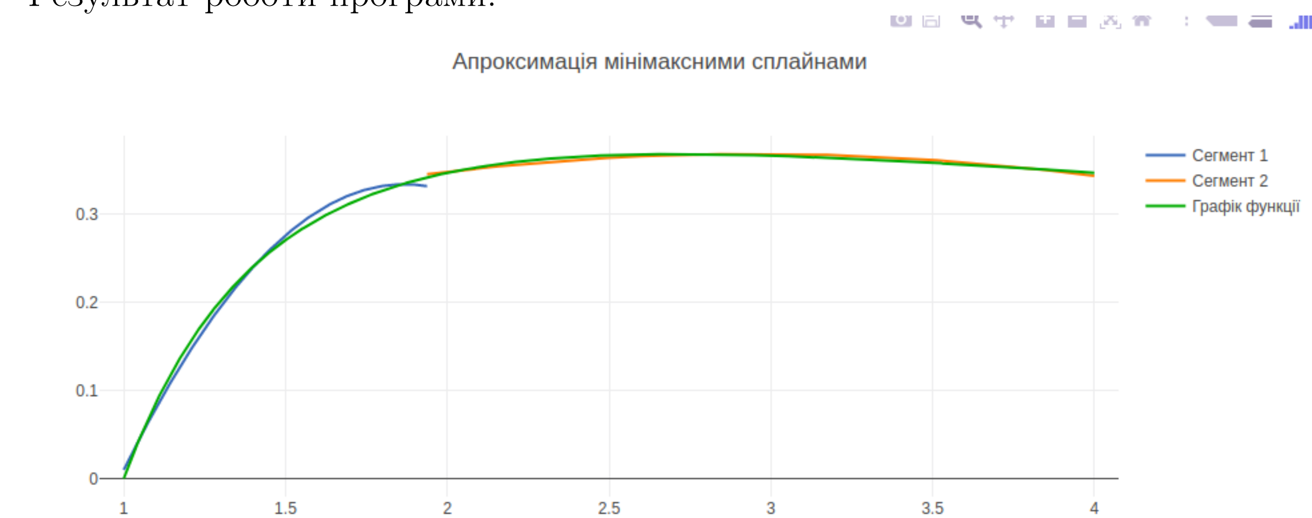
```python
import minmax

def getError(result):
  iterations = len(result)
  error = result[str(iterations)]["max_err"]
  return abs(error)

def shrinkInterval(interval, history = []):
  [start, end] = interval
  if (start > end):
    print('Begining of interval is greater than its end')
    return
  left_boundaries = sorted(filter(lambda x: x < end, map(lambda x: x[1], history)))
  if len(left_boundaries) > 0:
    nearest_left_neighbor = left_boundaries[-1]
    delta = (end - nearest_left_neighbor) / 2.0
    return [start, end - delta]
  else:
    mid = (end - start) / 2.0
    return [start, start + mid]

def expandInterval(interval, history):
  [start, end] = interval
  if (start > end):
    print('Begining of interval is greater than its end')
    return
  if len(history) == 0:
    print('when expanding there should be history')
    return

  right_boundaries = sorted(filter(lambda x: x > end, map(lambda x: x[1], history)))
  if len(right_boundaries) > 0:
    nearest_right_neighbor = right_boundaries[0]
    delta = (nearest_right_neighbor - end) / 2.0
    return [start, end + delta]


def main(func, deg, start, end, precision, allowed_error):
  interval = [start, end]
  historyOfIntervals = []
  splines = []

  def approximate(interval):
    if type(interval) is list:
```

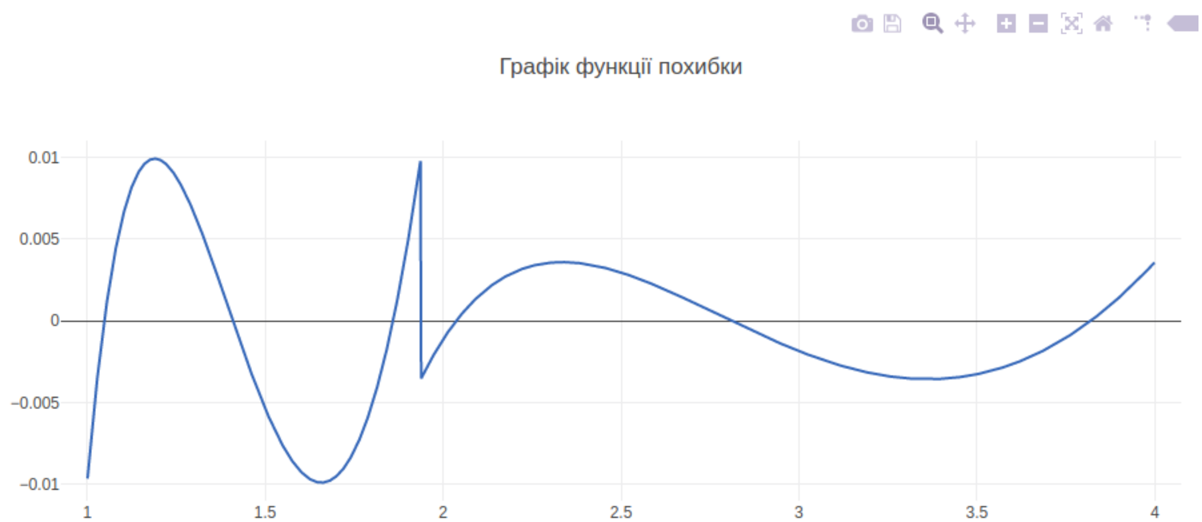
```
      return minmax.main(f_str=func, start=interval[0], end=interval[1], degree=deg, precision=precision)

  def make_approximation_on_one_segment(overallInterval):
    if not type(overallInterval) is list:
      print overallInterval
      return
    result = approximate(overallInterval)
    max_error = getError(result)
    print("Interval {}".format(overallInterval))
    # print("max_error: {} Interval {} history {}".format(max_error, overallInterval, historyOfIntervals))
    condition = abs(abs(max_error) - allowed_error)

    if condition > (allowed_error / 10):

      if (max_error > allowed_error):
        shrinkedInterval = shrinkInterval(overallInterval, historyOfIntervals)
        if len(historyOfIntervals) == 0:
          historyOfIntervals.append(overallInterval)
        historyOfIntervals.append(shrinkedInterval)
        make_approximation_on_one_segment(shrinkedInterval)
      else:
        if overallInterval[1] != interval[1]:
          expandedInterval = expandInterval(overallInterval, historyOfIntervals)
          historyOfIntervals.append(expandedInterval)
          make_approximation_on_one_segment(expandedInterval)
        else:
          splines.append({
            "interval": overallInterval,
            "spline": result,
            "max_error": max_error
          })
    else:
      splines.append({
        "interval": overallInterval,
        "spline": result,
        "max_error": max_error
      })
      if overallInterval[1] < interval[1]:
        historyOfIntervals[:] = []
        make_approximation_on_one_segment([overallInterval[1], interval[1]])

  make_approximation_on_one_segment(interval)
  return splines

# print main('sin(x)', deg=2, start=1, end=4, precision=0.1, allowed_error=0.001)
```

4.2 Приклади виконання програми

Приклад 1 Знайдемо чебишовське наближення поліномом степеня 2 для функції $f(x) = \frac{ln(x)}{x}$ на проміжку $[1, 4]$. Точність ($\epsilon = 0.01$). Максимально допустима похибка на ланці сплайна 0.01

Результат роботи програми:





Сегмент	Максимальна похибка	Інтервал	Аналітичний вигляд
1	0.010	$[1.000; 1.938]$	$-0.4309x^2 + 1.6089x - 1.1680$
2	0.004	$[1.938; 4.000]$	$-0.0227x^2 + 0.1338x + 0.1707$

Приклад 2 Вхідні дані:

Функція, яку апроксимуємо

sin(x)

Степінь многочлена

1

Початок інтервалу

0

Кінець інтервалу

3.1415

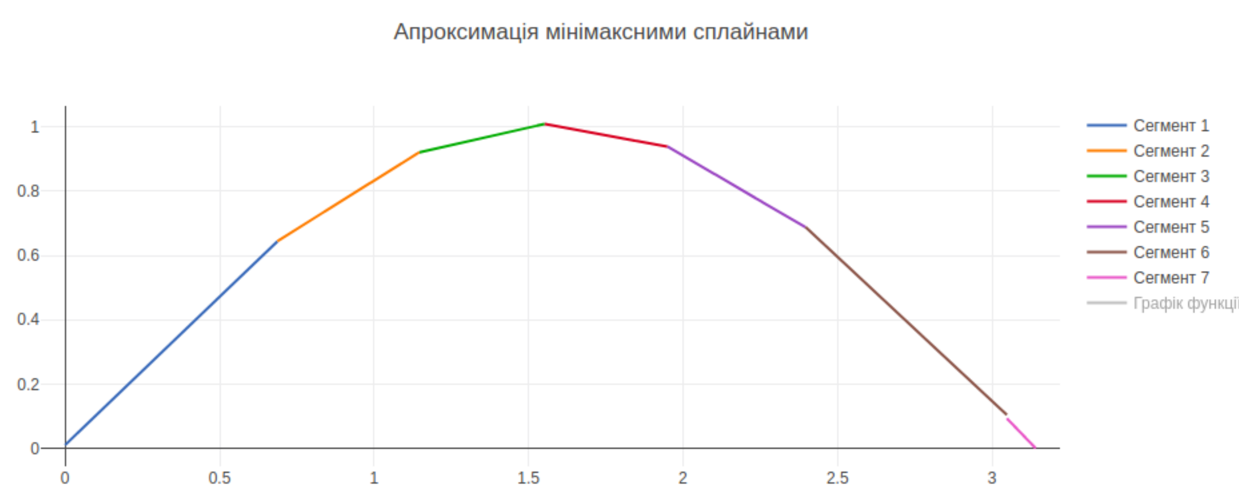
Точність

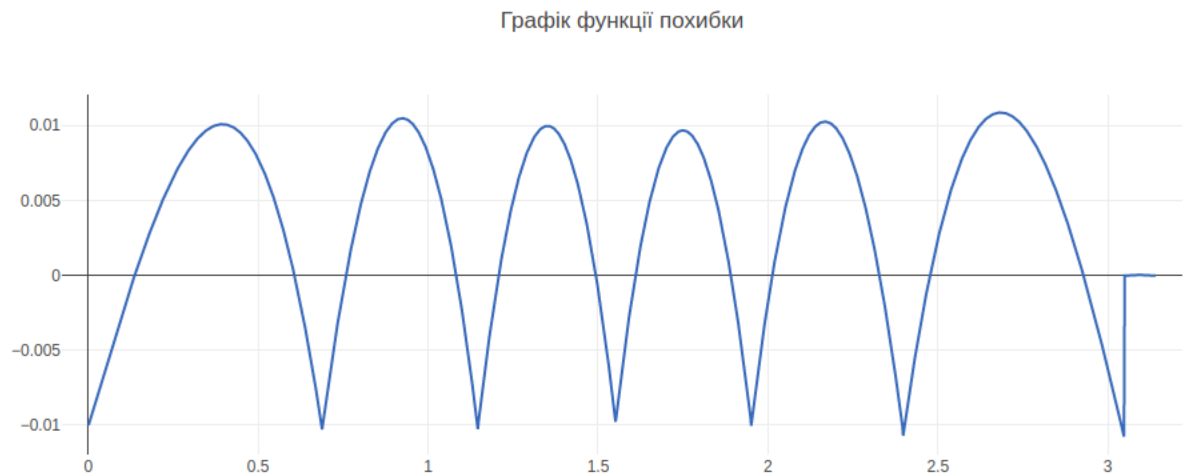
0.01

Допустима похибка на одному відрізку сплайна

0.01

Результат роботи програми:





Сегмент	Максимальна похибка	Інтервал	Аналітичний вигляд
1	0.01009	$[0.000; 0.687]$	$0.9231x + 0.0100$
2	0.01048	$[0.687; 1.147]$	$0.6026x + 0.2307$
3	0.00997	$[1.147; 1.552]$	$0.2175x + 0.6719$
4	0.00967	$[1.552; 1.950]$	$-0.1781x + 1.2860$
5	0.01026	$[1.950; 2.397]$	$-0.5618x + 2.0348$
6	0.01086	$[2.397; 3.048]$	$-0.8973x + 2.8394$
7	0.00003	$[3.048; 3.142]$	$-0.9985x + 3.1370$

Приклад 3 Вхідні дані:

Функція, яку апроксимуємо

e^x

Степінь многочлена

2

Початок інтервалу

1

Кінець інтервалу

4

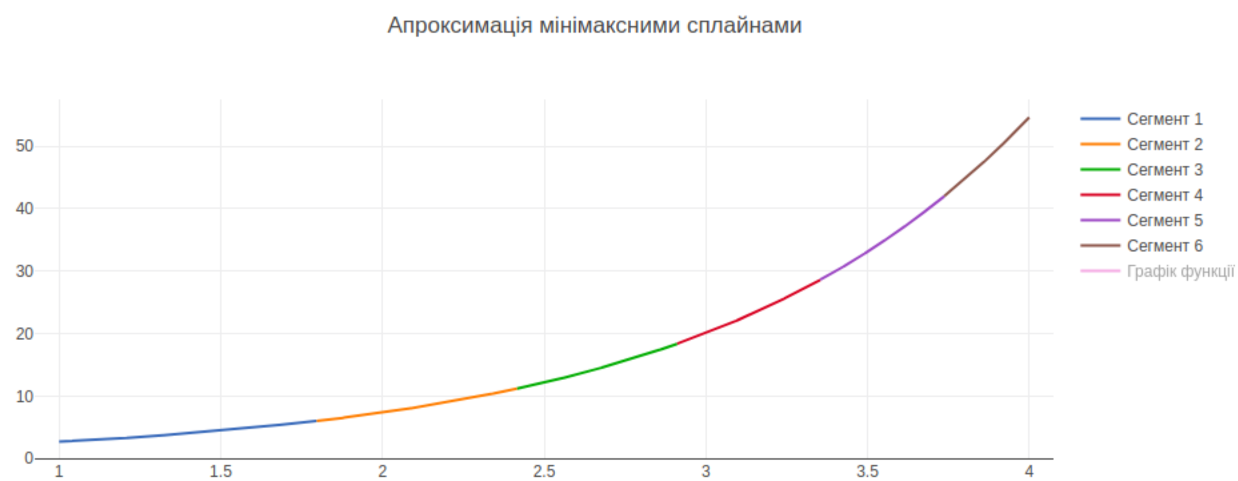
Точність

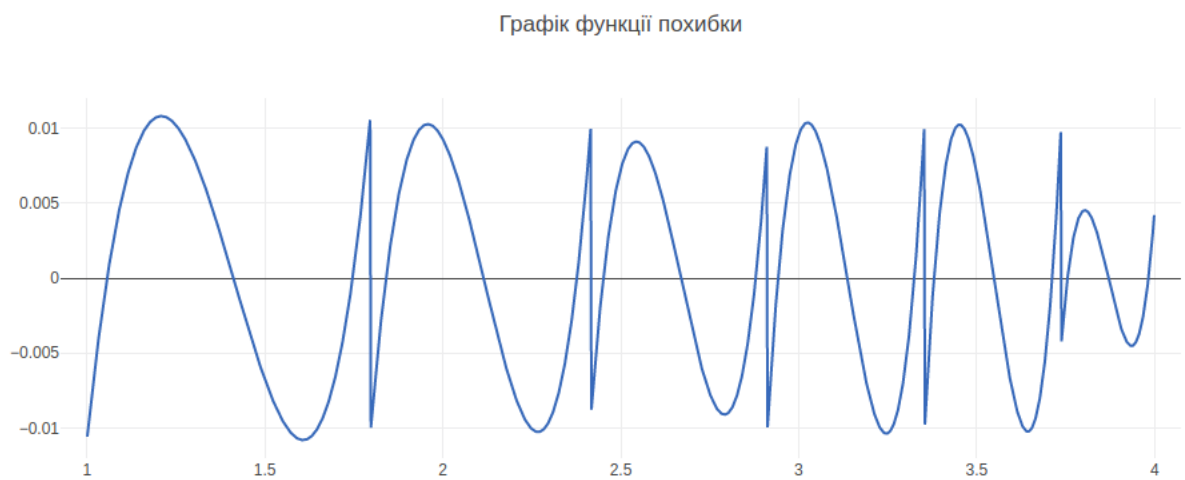
0.01

Допустима похибка на одному відрізку сплайна

0.01

Результат роботи програми:





Сегмент	Максимальна похибка	Інтервал	Аналітичний вигляд
1	0.01080	$[1.000; 1.797]$	$2.0581x^2 - 1.6265x + 2.2975$
2	0.01026	$[1.797; 2.417]$	$4.1517x^2 - 9.1727x + 9.1184$
3	0.00910	$[2.417; 2.911]$	$7.2221x^2 - 24.0156x + 27.0762$
4	0.01037	$[2.911; 3.354]$	$11.5238x^2 - 49.1252x + 63.7371$
5	0.01024	$[3.354; 3.737]$	$17.3951x^2 - 88.5323x + 129.8808$
6	0.00452	$[3.737; 4.000]$	$23.9830x^2 - 137.5833x + 221.1976$