Міністерство освіти і науки України
Національний університет "Києво-Могилянська академія"
Факультет інформатики
Кафедра математики

Кваліфікаційна робота
освітній ступінь - бакалавр

на тему: **"Розробка асимптотично оптимальних методів сплайн-апроксимації функцій двох змінних"**

Виконала: студентка 4-го року навчання
освітньої програми "Прикладна математика"
спеціальності 113 Прикладна математика
Скаженик Тетяна Андріївна
Керівник: Кашпіровський О.І.,
канд. фіз.-мат. наук, доцент
Рецензент ______________________
(прізвище та ініціали)
Кваліфікаційна робота захищена
з оцінкою ______________________
Секретар ЕК ______________________
“____”__________________ 2022 р.

Київ – 2022

Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра математики

ЗАТВЕРДЖУЮ
Зав.кафедри математики,
проф., д.ф-м.н., Б. В. Олійник

____

(підпис)
"____"________________ 2021

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на кваліфікаційну роботу

студентці 4-го курсу, факультету інформатики
Скаженик Тетяні Андріївні

Тема: Розробка асимптотично оптимальних методів сплайн-апроксимації функцій двох змінних

Вихідні дані: Метод Лигуна-Шумейка побудови асимптотично оптимальної сітки для інтерполяції кубічними сплайнами

Зміст ТЧ до кваліфікаційної роботи:

Зміст
Анотація
Вступ
Інтерполяція функцій однієї змінної сплайнами.
Оптимізація кількості вузлів інтерполяції
Оптимізація інтерполяції сплайнами функцій двох змінних
Висновки
Список літератури
Додатки

Дата видачі "____"__________ 2021 Керівник _______________________
(підпис)
Завдання отримала _______________________
(підпис)

Графік підготовки кваліфікаційної роботи до захисту

Графік узгоджено "____"__________________ 2022 р.

Календарний план виконання роботи:

Номер	Назва етапу кваліфікаційної	Термін виконання етапу	Примітка
1.	Отримання теми кваліфікаційної роботи.	11.09.2021	
2.	Ознайомлення з темою кваліфікаційної.	28.09.2021	
3.	Планування структури роботи. Огляд наукової літератури за темою.	17.11.2021	
4.	Дослідження задачі інтерполяції та опис основних понять.	20.02.2022	
5.	Дослідження застосування сплайнів першої степені двух змінних на прямокутній сітці	01.03.2022	
6.	Розробка асимптотично оптимальних методів сплайн-апроксимації функції двох змінних	10.04.2022	
7.	Надання роботи керівнику на перевірку.	26.05.2022	
10.	Передзахист дипломної роботи.	14.06.2022	
11.	Захист дипломної роботи.	05.07.2022	

Науковий керівник ______________________

(ПІБ)

Виконавець кваліфікаційної роботи ______________________

(ПІБ)

Зміст

Анотація

Сплайни досить розповсюджені у використанні в математичній теорії, різних обчислювальних програмах, а також при вирішенні задач теорії нелінійних електричних ланцюгів.

Сплайн-функції - це функції, отримані гладким склеюванням шматків многочленів. Методи, які базуються на сплайн-функціях займають панівну позицію серед найбільш потужних методів обчислювальної математики. За допомогою сплайнів можна наблизити будь-яку неперервну функцію $f(x)$, задану із певною точністю ε на відповідному відрізку.

Використання сплайнів має багато переваг. По-перше, вони гнучкі у використанні, що робить їх потужними для вирішення задач наближення функцій. А ці задачі самі по собі лежать в основі багатьох методів обчислювальної математики. Часто виникає необхідність апроксимувати характеристики нелінійних елементів ланцюгів. Це пов'язано з тим, що найчастіше ці характеристики задаються таблицями та графіками. Тож, для того, щоб була змога оперувати механічними методами розрахунку ланцюгів, краще представляти їх у вигляді аналітичних виразів. По-друге, алгоритми, які побудовані на основі сплайн-функцій, легко і досить ефективно реалізуються.

Сплайни виникли в той час, коли з'ясувалася неможливість та неефективність застосування класичного методу наближення алгебраїчними та тригонометричними поліномами при розв'язанні багатьох прикла-

(а) (б)

Рис. 1: Контрастні зображення

дних задач. Сплайни від двох змінних інтенсивно використовуються в різних системах комп'ютерного моделювання.

Також сплайни набули широкого розповсюдження й в обробці та аналізі зображень. Сучасні мультимедійні продукти вимагають більшої ємності сховища і мережі. З розвитком систем спостереження, мультимедіа, автомобільної промисловості почали з'являтися нові ефективні підходи стиснення відеопотоків та обробки зображень.

Існує безліч підходів для вирішення даної задачі. Один із них полягає у вилученні контурів та застосуванні ідеї локалізації (див. рис.1). Зображення представляють у вигляді деякої нескінченної системи різницевих рівнянь. Для розв'язання подібних задач можуть бути використані методи локальної сплайн-апроксимації, основою яких є локальні асимптотичні формули для параметрів сплайнів. Для стиснення зображення, нам необхідно виділити найважливішу інформацію (тобто місця особливо щільного скупчення пікселей). Для цього найкраще використовувати машинно-орієнтований підхід для апроксимації нелінійних функцій із заданою наперед точністю за допомогою сплайн-функцій.

При цьому досягається асимптотично мінімальне наближення, тобто наближення із мінімальною кількістю вузлів (стиск зображень) необхідних для забезпечення заданої точності.

Метою цієї роботи є побудова алгоритму вибору асимтотично оптимальних вузлів інтерполяції лінійних сплайнів двох змінних. З цією метою в роботі пропонується узагальнення результатів роботи Лигуна і Шумейко [1], в якій подібна проблема досліджується для випадку апроксимації функцій однієї змінної кубічними сплайнами.

Вступ

Система дослідження сплайнів як математичних об'єктів розпочалася з 1946 року в роботах Шенберга. Сплайни однієї та двох змінних широко використовуються при розв'язанні найрізноманітніших прикладних задач [2].

При обробці великих масивів інформації виникає проблема зменшення її об'єму. У випадку, коли такій інформації відповідає деяка гладка функція однієї змінної $f(x)$, ефективний метод стиску інформації на основі кубічних сплайнів запропоновано в роботі Лигуна і Шумейка [1].

Стиск об'єму інформації в методі Лигуна-Шумейка полягає в заміні початкової рівномірної сітки сплайн-інтерполяції з великою кількістю вузлів на нерівномірну сітку із суттєво меншою кількістю вузлів. При цьому інтерполяція по новій прорідженій системі вузлів має давати прийнятну точність апроксимації $f(x)$.

Головна частина відхилення кубічного сплайна з рівномірною сіткою пропорційна добутку $f^{IV}(x)$ на h^4 ($h > 0$ крок рівномірної сітки). Отже, при великих коливаннях значень $|f^{IV}(x)|$ розподіл похибки сплайн-апроксимації на рівномірній сітці є нерівномірним. Звідси і виникає ідея побудови нерівномірної сітки з кроком h_k, які мають бути пропорційні значенням $|f^{IV}(x)|^{-\frac{1}{4}}$. Отримана в такий спосіб нерівномірна сітка є близькою до найкращої (асимптотично оптимальною).

Така сітка забезпечує майже рівномірний розподіл відхилення сплайна від інтерпольованої функції $f(x)$ по всій області визначення.

Зауважимо, що похідну $f^{IV}(x)$ в методі Лигуна-Шумейка можна замінити відповідними різницевими відношеннями.

В даній роботі пропонується узагальнення методу Лигуна-Шумейка на випадок, коли ми маємо великий масив інформації, якому відповідає функція двох змінних, що визначена на деякому прямокутнику

$$\Pi = [0, A] \times [0, B]$$

Припускаємо, що на рівномірній двовимірній сітці у вузлах (x_i, y_j) маємо значення $f(x, y)$.

Для апроксимації $f(x, y)$ використовуємо лінійний інтерполяційний сплайн $s_1(x, y)$.

Для такого сплайна відхилення від $f(x, y)$ оцінюється величиною $\frac{1}{3}\phi(x, y)h^2$, де

$$\phi(x, y) = max(|f''_{xx}(x, y)|, |f''_{xy}(x, y)|, |f''_{yy}(x, y)|).$$

Таким чином для вибору оптимальних вузлів сплайн інтерполяції прямокутник Π слід розрізати на менші прямокутники π_k, площі яких мають бути пропорційні $\phi(x, y)^{-1}$ (більш точні формулювання наведено пізніше).

Зауважимо, що метод Лигуна-Шумейка в одновимірному випадку визначає асимптотично оптимальну сітку однозначно.

У двовимірному випадку на форму асимптотично оптимальних прямокутників π_k доводиться накладати додаткому умову на їх розміри.

В даній роботі розглядаються прямокутники π_k з однаковими горизонтальними сторонами.

Розділ 1

Інтерполяція функцій однієї змінної сплайнами. Оптимізація кількості вузлів інтерполяції

1.1 Означення та деякі властивості поліноміальних сплайнів однієї змінної

Нехай деяка функція $f(x)$ задана графічно чи таблично. Складність аналітичного представлення функції залежить від розміру інтервала та необхідної точності наближення.

Одне з найголовніших питань при наближенні заданої функції є питання вибору метода апроксимації. Зазвичай для апроксимації функцій використовуються степеневі та експоненціальні поліноми, раціональні дроби, деякі спеціальні функції.

Проте такі методи є неточними, оскільки при збільшенні кількості даних, похибки при обчисленні зростають.

Останнім часом все більшого визнання набувають сплайн-функції, які отримуються "склеюванням"кусків многочленів.

Означення 1.1.1. *Неперервну функцію з сіткою* $\Delta = x_i, i = \overline{0, N}$, *яка на* *кожному проміжку* $[x_{i-1}, x_i]$ *збігається з деяким поліномом степеня* k *називають сплайн-функцією степеня* k.

Простота реалізації на ЕОМ та визначні локальні властивості сплайнів дають можливість оптимально визначати характеристики апроксимуючої функції, а також автоматизувати дослідження таблиці значень або графіка функції. При цьому досягається асимтотичне оптимальне наближення з мінімальною кількістю вузлів, необхідних для забезпечення заданої точності.

Означення 1.1.2. *Ермітова інтерполяція – це метод поліноміальної інтерполяції із застосуванням похідних функції.*

Суть методу полягає в тому, щоб побудувати поліном, значення якого в точках інтерполяції збігаються із значеннями початкової функції, так само збігаються і значення їх перших похідних.

Означення 1.1.3. *Кубічним сплайном мінімального дефекта (дефекта 1) називають сплайн* $S_3(x)$, *оскільки з трьох похідних* $S_3(x)$ *лише третя похідна є розривною.*

1.2 Метод Лигуна-Шумейка побудови асимптотично оптимальної сітки для інтерполяції кубічними сплайнами

Нехай на прямокутнику $\prod = [0; A] \times [0; B] \subset R^1$ визначена гладка функція $z = f(x, y)$. Припустимо, що ми маємо таблицю наближених значень

$$z_{ij} \approx f(x_i, y_i) + \varepsilon_{ij}$$

в точках двовимірної сітки рівновіддалених вузлів

$$\triangle = \{(x_i, y_j) : x_i = ih, y_j = jh\} i = \overline{0, N_1}, j = \overline{0, N_2},$$

де $h = \frac{A}{N_1} = \frac{B}{N_2}$.

Припустимо щодо шуму ε_{ij} виконання умови

$$max|\varepsilon_{ij}| \leq \varepsilon$$

$$i = \overline{0, N_1} \qquad j = \overline{0, N_2}$$

де ε достатньо мале число.

При великих N_1, N_2 виникає потреба в зменшенні об'єму таблиці значень z_{ij}.

За умови неоднакової поведінки $f(x, y)$ в різних частинах прямокутника Π перехід нової сітки рівновіддалених вузлів $\triangle$ з більшим кроком h не є виправданим. В залежності від локальної "швидкості"зміни функції $f(x, y)$ прямокутник Π слід розрізати на шматки π_k.

$$\Pi = \bigcup_{k=\overline{1,N_3}} \pi_k$$

де N_3 має бути суттєво меншим добутку $N_1 \cdot N_2$, а π_k - прямокутники різних розмірів. Частинам прямокутника Π, в яких спостерігається швидка змінна $f(x, y)$ мають відповідати малі розміри. В зонах повільної зміни $f(x, y)$ розміри π_k мають бути більшими.

Для розв'язання такої проблеми в даній роботі пропонується використати сплайни 1-го порядку від двох змінних.

Для вибору вузлів інтерполяції використаємо узагальнення методів побудови асимптотично оптимальної апроксимації кубічними сплайнами однієї змінної, яка запропонована в роботі Лигуна і Шумейко [1]. Дамо короткий опис методу Лигуна-Шумейко.

Нехай задані значення

$$y_k = f(x_k), k = \overline{0; N}$$

$$x_k = kh, A = Nh,$$

де $f \in C^{IV}[0; A]$.

При інтерполяції $f(x)$ кубічним сплайном мінімального дефекту $S_3(f, x)$ на кожному проміжку $[x_k, x_{k+1}]$ величина відхилення сплайна $S_3(f, x)$ від $f(x)$ оцінюється величиною

$$\frac{5}{384} \cdot h^4 |f^{IV}(\varepsilon_k)|$$

$$\varepsilon_k \in [x_k, x_{k+1}]$$

Для отримання асимптотично оптимальних вузлів інтерполяції розглядається функція

$$\phi(x) = \int_0^x |f^{IV}(t)|^{\frac{1}{4}} dt, x \in [0, A],$$

яка є первісною від

$$|f^{IV}(t)|^{\frac{1}{4}}.$$

Асимптотично оптимальні вузли z_j $0 \leq j \leq M$, $M \leq N$ за методом Лигуна-Шумейко обираються так, щоб прирости функції $\phi(x)$ на проміжках $[z_{j-1}, z_j]$ були однаковими, тобто

$$\int_0^{z_1} |f^{IV}(t)|^{\frac{1}{4}} dt = \int_{z_1}^{z_2} |f^{IV}(t)|^{\frac{1}{4}} dt = ... = \int_{z_{M-1}}^{A} |f^{IV}(t)|^{\frac{1}{4}} dt. \qquad (1.1)$$

Враховуючи, що $\phi(x)$ монотонно зростаюча функція, то вузли z_j,

$j = \overline{1, M}$ визначаються однозначно.

Замість невідомої похідної $f^{IV}(x)$ при чисельній реалізації метода Лигуна-Шумейко використовується кусково-стала функція.

На кожному проміжку $[x_{k-1}, x_k]$ $f^{IV}(x)$ замінюється розділеною різницею 2-го порядку

$$\frac{M_{k+1} - 2M_k + M_{k-1}}{h^2} \stackrel{\text{def}}{=} P_k,$$

де M_k значення похідної 2-го порядку інтерполяційного сплайна $S_3(f, x)$

$$M_k = S_3{}''(f, x_k).$$

Інтерполяційний сплайн $S_3(f, x)$ в данній ситуації можна замінити локальним кубічним сплайном [2].

Зауважимо, що у випадку інтерполяційного сплайна $S_3(f, x)$ і при його заміні на локальний сплайн значення P_k відрізняються від $f^{IV}(x_k)$ на величину $O(h^4)$.

При заміні $f^{IV}(x)$ кусково-сталою графік функції $\phi(x)$ стає ламаною лінією. Враховуючи, що $\phi(x)$ монотонно зростаюча, пошук вузлів z_j, що задовольняють умові (1.1), є достатньо простою задачою.

Враховуючи, що інтерполяційні сплайни від двох змінних мають суттєво складнішу природу порівняно зі сплайнами однієї змінної, спробуємо узагальнити метод Лигуна-Шумейко на лінійні сплайни від двох змінних, які мають достатньо просту конструкцію. Такі сплайни використовуються в методі скінченних елементів [3].

Для побудови таких сплайнів використовується трьохточкова інтерполяція функцій двох змінних. Нехай z_0, z_1, z_2 значення функції $f(x, y)$ в точках (x_0, y_0), (x_1, y_1), (x_2, y_2), які не лежать на одній прямій.

Лінійному многочлену

$$z = a + bx + cy,$$

що в точках (x_k, y_k) приймає значення $z_k, k = 0; 1; 2$ відповідає площина, що визначається точками $(x_k, y_k, z_k),\ k = 0; 1; 2.$

Точка $M(x, y, z)$ лежить в цій площині, якщо вектори

$$\overrightarrow{M_0M}, \overrightarrow{M_0M_1}, \overrightarrow{M_0M_2}$$

є лінійно залежними, а отже

$$\begin{vmatrix} x - x_0 & y - y_0 & z - z_0 \\ x_1 - x_0 & y_1 - y_0 & z_1 - z_0 \\ x_2 - x_0 & y_2 - y_0 & z_2 - z_0 \end{vmatrix} = 0. \tag{1.2}$$

Звідси отримаємо рівняння,

$$\begin{vmatrix} y_1 - y_0 & z_1 - z_0 \\ y_2 - y_0 & z_2 - z_0 \end{vmatrix} \cdot (x - x_0) - \begin{vmatrix} x_1 - x_0 & z_1 - z_0 \\ x_2 - x_0 & z_2 - z_0 \end{vmatrix} \cdot (y - y_0) +$$

$$+ \begin{vmatrix} x_1 - x_0 & y_1 - y_0 \\ x_2 - x_0 & y_2 - y_0 \end{vmatrix} \cdot (z - z_0) = 0,$$

яке еквівалентне (1.2).

Розділ 2

Оптимізація інтерполяції сплайнами функцій двох змінних

2.1 Сплайни 1-го степеня двох змінних. Оцінка залишкового члена при інтерполяції функцій лінійними сплайнами двох змінних

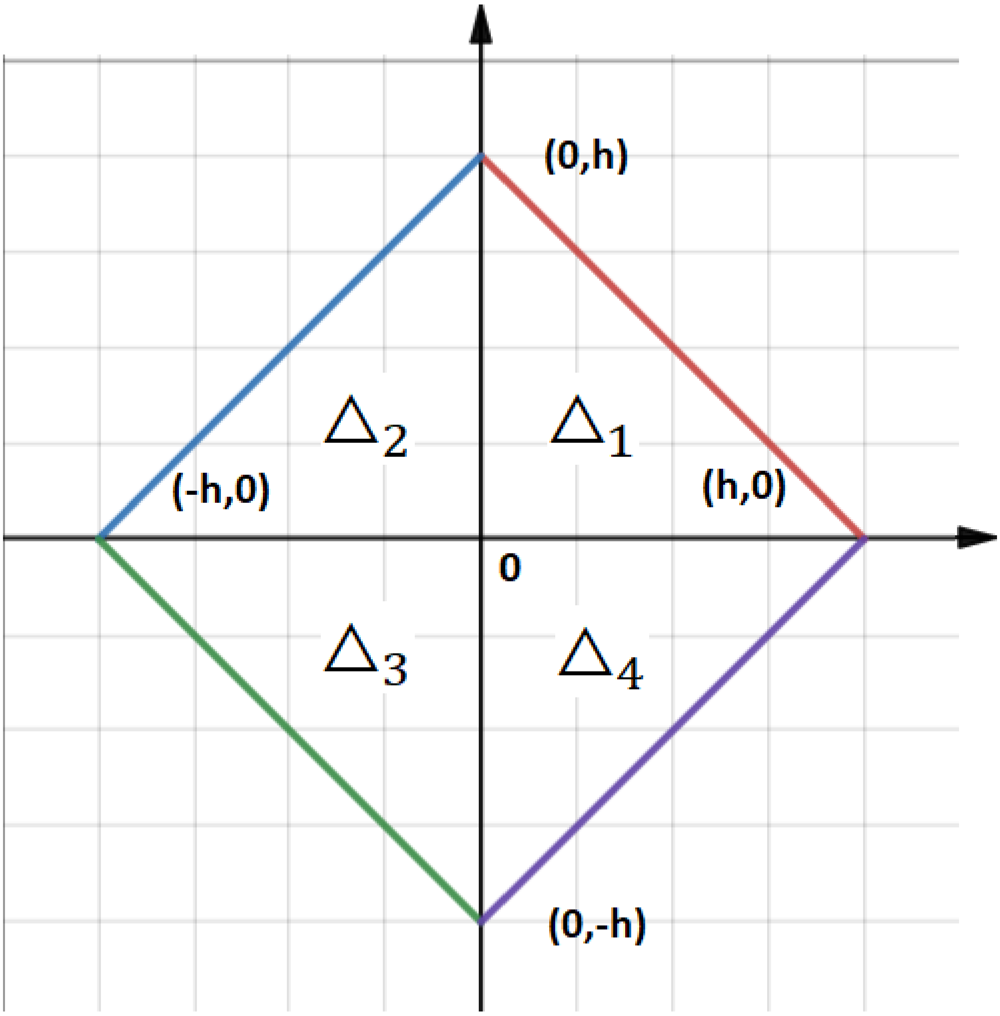
Для побудови лінійних сплайнів двох змінних запишемо лінійні інтерполяційні многочлени з вузлами інтерполяції у вершинах рівнобедрених прямокутних трикутників з катетами довжини h ($h > 0$).

1. У випадку трикутника $\triangle_1$ з вершинами $(0;0)$, $(h;0)$, $(0;h)$ маємо

$$L_{11}(x,y) = f(0,0) + \frac{f(h,0) - f(0,0)}{h} \cdot x + \frac{f(0,h) - f(0,0)}{h} \cdot y.$$

2. У випадку трикутника $\triangle_2$ з вершинами $(0;0)$, $(-h;0)$, $(0;h)$ маємо

$$L_{11}(x,y) = f(0,0) + \frac{f(0,0) - f(-h,0)}{h} \cdot x + \frac{f(0,h) - f(0,0)}{h} \cdot y.$$



3. У випадку трикутника Δ_3 з вершинами $(0;0)$, $(-h;0)$, $(0;-h)$ маємо

$$L_{11}(x,y) = f(0,0) + \frac{f(0,0) - f(-h,0)}{h} \cdot x + \frac{f(0,0) - f(0,-h)}{h} \cdot y.$$

4. У випадку трикутника Δ_4 з вершинами $(0;0)$, $(h;0)$, $(0;-h)$ маємо

$$L_{11}(x,y) = f(0,0) + \frac{f(h,0) - f(0,0)}{h} \cdot x + \frac{f(0,0) - f(0,-h)}{h} \cdot y.$$

Як слідує із Зав'ялова [2] ст.89 всередині трикутника $\Delta_k (k = \overline{1,4})$ для довільної двічі неперервно диференційованої функції $f(x,y)$ $f \in C^2(\triangle_k)$ справедлива така оцінка наближення

$$|L_{11}(x,y) - f(x,y)| \leq \frac{2}{3}h^2 \cdot M_{2x}$$
$$\forall (x,y) \in \triangle_k, k = \overline{1,4},$$

де

$$M_{2k} = max_{(u,v)\in\triangle_k}(|f''_{xx}(u,v)|, |f''_{xy}(u,v)|, |f''_{yy}(u,v)|)$$

При побудові лінійного інтерполяційного сплайна $S_{11}(x,y)$ на прямокутнику $\Pi = [0,A] \times [0,B]$ з сіткою рівновіддалених вузлів (x_i, y_j)

$$x_i = ih, i = \overline{0, N_1},$$
$$y_j = jh, j = \overline{0, N_2}$$

на кожному з трикутників типу $\triangle_k$ сплайн $S_{11}(x,y)$ збігається з описаним раніше лінійним інтерполяційним многочленом $L_{11}(x,y)$. Для оцінки відхилень сплайна $S_{11}(x,y)$ замінимо величини M_{2k} максимумом модулів різницевих похідних 2-го порядку функції $f(x,y)$ в точці (x_i, y_j)

$$D_{xx}(x_i, y_j) = h^{-2} \cdot (f(x_{i+1}, y_j) + f(x_{i-1}, y_j) - 2f(x_i, y_j))$$
$$D_{xy}(x_i, y_j) = 0,25h^{-2}(f(x_{i+1}, y_{j+1}) + f(x_{i-1}, y_{j-1})$$
$$-f(x_{i-1}, y_{j+1}) - f(x_{i+1}, y_{j-1}))$$
$$D_{yy}(x_i, y_j) = h^{-2}(f(x_i, y_{j+1}) + f(x_i, y_{j-1}) - 2f(x_i, y_j)),$$

де

$$i = \overline{1, N_1 - 1}, j = \overline{1; N_2 - 1}$$

Нехай

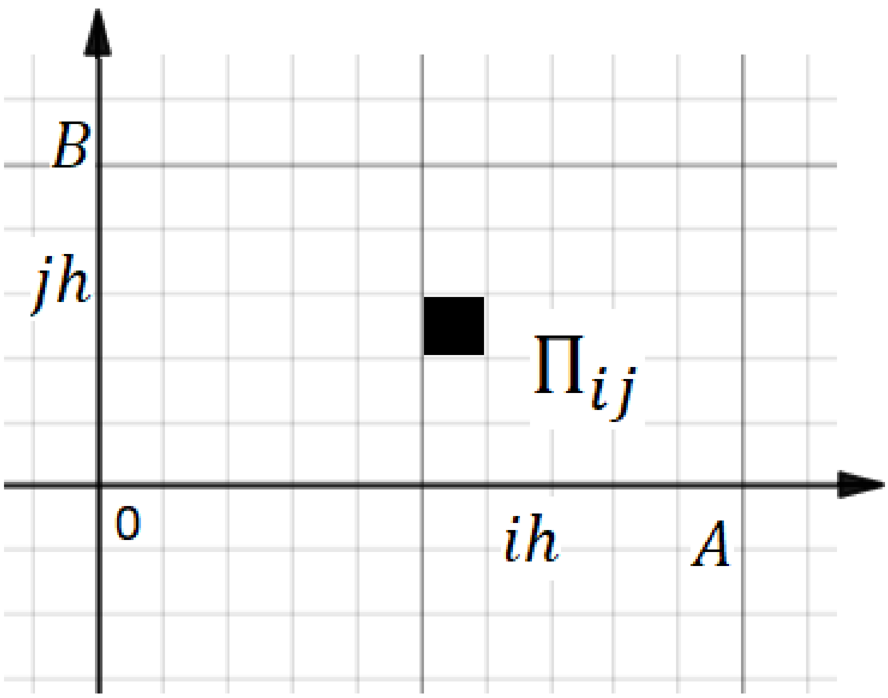
$$m_{ij} = max(|D_{xx}(x_i, y_j)|, |D_{xy}(x_i, y_j)|, |D_{yy}(x_i, y_j)|)$$

Π_{ij} - квадрат із стороною h, верхньою правою вершиною якої є точка $(x_i, y_j) = (ih, jh)$, $i = \overline{1, N_1}$, $j = \overline{1, N_2}$.

Таким чином прямокутник

$$\Pi = [0, A] \times [0, B]$$

розрізається $N_1 \times N_2$ квадрати зі стороною h.



На прямокутнику Π визначимо кусково-сталу функцію

$$\phi(x,y) = \begin{cases} m_{ij}, (x,y) \in \Pi_{ij}, i = \overline{1, N_1 - 1}, j = \overline{1, N_2 - 1} \\ m_{N_1-1,j}, (x,y) \in \Pi_{N_1 j}, j = \overline{1, N_2 - 1} \\ m_{i,N_2-1}, (x,y) \in \Pi_{iN_2}, i = \overline{1, N_1 - 1} \\ m_{N_1-1,N_2-1}, (x,y) \in \Pi_{N_1 N_2} \end{cases}$$

Функція $\phi(x,y)$ дає емпіричне наближення для різниці $f(x,y) - S_{11}(x,y)$. Зауважимо, на відрізках прямих

$$x = x_i, i = \overline{1, N_1 - 2}$$

$$y = y_j, j = \overline{1, N_2 - 2}$$

такий спосіб визначення $\phi(x,y)$ дає два різних значення, а у вузлах (x_i, y_j) навіть чотири різних значення. Але така неоднозначність визначення функції $\phi(x,y)$ відбувається на множині міри Лебега нуль на площині.

Оскільки в подальшому використовуються подвійні інтеграли від $\sqrt{\phi(x,y)}$, неоднозначність визначення функції $\phi(x,y)$ не є принциповою для визначення асимптотично оптимальних прямокутників

$$\pi_k, k = \overline{1, m}, m < N_1 \cdot N_2.$$

2.2 Узагальнення методу Лигуна-Шумейка на випадок оптимізації вузлів інтерполяції лінійних сплайнів двох змінних

Для сплайн-апроксимації функції $f(x,y)$ застосуємо аналог рівності (1.1) із заміною в підінтегральному виразі $\sqrt[4]{f^{IV}(x)}$ на $\sqrt{\phi(x,y)}$ інтегралів кратності одиниця на подвійні інтеграли

$$\iint_{\pi_1} \sqrt{\phi(x,y)}\,dx\,dy = \iint_{\pi_2} \sqrt{\phi(x,y)}\,dx\,dy$$

$$= \ldots = \iint_{\pi_m} \sqrt{f(x,y)}\,dx\,dy. \tag{2.1}$$

Зауважимо, що із системи рівнянь (1.1) асимптотично оптимальні вузли z_j визначаються за методом Лигуна-Шумейка однозначно.

У випадку системи (2.1) така однозначність вже відсутня. Тому для визначення прямокутників потрібно накласти якісь додаткові умови на прямокутники π_k. Наприклад, можна вимагати, щоб всі горизонтальні сторони прямокутників π_k були однаковими.

Припустимо, що всі горизонтальні сторони прямокутників π_k дорівнюють H, де H ділить націло число A і є кратним h

$$A = pH, H = qh, p, q \in N.$$

Опишемо алгоритм визначення прямокутників π_k. Знайдемо величину

$$d = \frac{1}{Mh^2} \iint_{\Pi} \sqrt{\phi(x,y)}\,dx\,dy = M^{-1} \cdot \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sqrt{\phi(x_i, y_j)}.$$

Отримавши значення d, будемо поступово визначати розміри бічних сторін прямокутників π_k, які позначимо τ_k. Надалі τ_k називатимемо висотами прямокутників π_k.

Прямокутник $\Pi = [0, A] \times [0, B]$ розріжемо на вертикальні смуги

$$\Pi_r = [(r-1)H, rH] \times [0; B], r = \overline{1, p}.$$

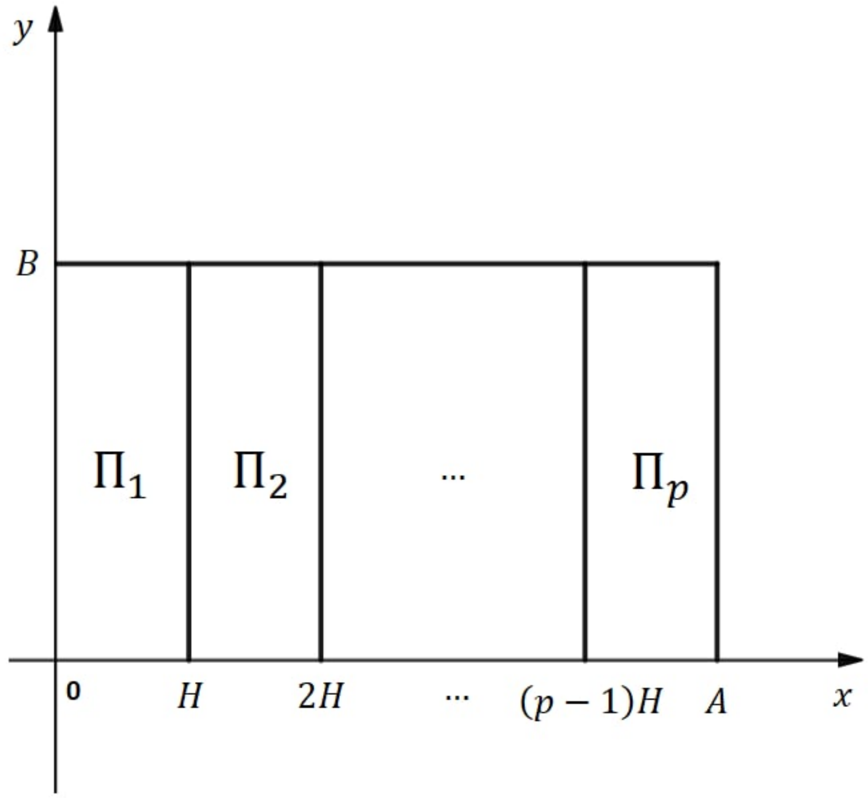


Рис. 2.1: Поділ прямокутника $\Pi = [0; A] \times [0; B]$ *на смуги* Π_r *шириною* H

Висоти τ_k будемо визначати як числа кратні h

$$\tau_k = s_k h, s_k \in N.$$

Значення s_1 визначимо так, щоб виконувалась подвійна нерівність

$$\sum_{i=1}^{q} \sum_{j=1}^{s_1-1} \sqrt{\phi(x_i, y_j)} < d \leq \sum_{i=1}^{q} \sum_{j=1}^{s_1} \sqrt{\phi(x_i, y_j)}$$

Для прямокутника $\pi_2 \subset \Pi_1$, значення s_2 визначимо з умови

$$\sum_{i=1}^{q}\sum_{j=1}^{s_1+s_2-1}\sqrt{\phi(x_i,y_j)} < 2d \leq \sum_{i=1}^{q}\sum_{s_1+1}^{s_1+s_2}\sqrt{\phi(x_i,y_j)}$$

В аналогічний спосіб визначаються s_k і для наступних прямокутників π_k із смуги Π_1. Кількість прямокутників π_k, які лежать в Π_1 визначимо так

$$M_1 = [h^{-2}d^{-1}\iint\limits_{\Pi_1}\sqrt{\phi(x,y)}dxdy] + 1,$$

де знак $[\cdot]$ в данному випадку означає цілу частину виразу, що знаходиться в дужках.

При такому поділі смуги Π_1 на прямокутники π_k $k = \overline{1, M_1}$ для прямокутника π_{M_1} може виконуватись умова

$$\iint\limits_{\pi_{M_1}}\sqrt{\phi(x,y)}dxdy < h^2 d.$$

Зауважимо, що для $1 \leq k < M_1$ матимемо

$$\iint\limits_{\pi_k} -\sqrt{\phi(x,y)}dxdy \geq h^2 d.$$

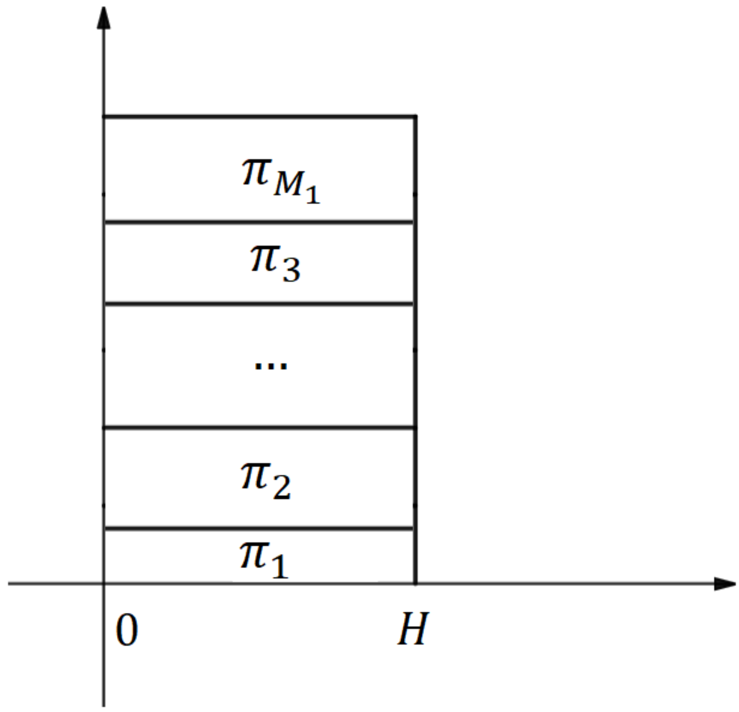


Рис. 2.2: Смуга Π_1 *розрізана на* M_1 *прямокутників* π_k *з висотами* $\tau_1 = s_1 h$, $\tau_2 = s_2 h$, .. $\tau_{M_1} = S_{M_1} h$

В аналогічний спосіб на прямокутники π_k, $k > M_1$ розрізаються смуги $\prod_r$, $1 < r \leq p$.

На рисунку (2.3) зображено прямокутник Π розрізаний на

$$\tilde{M} = \sum_{r=1}^{p} M_r$$

прямокутників π_k, де M_r- кількість прямокутників π_k в смузі Π_r.

Зауважимо, що число $\tilde{M}$ може бути більшим за M, причому має місце нерівність

$$M \leq \tilde{M} \leq M + p.$$

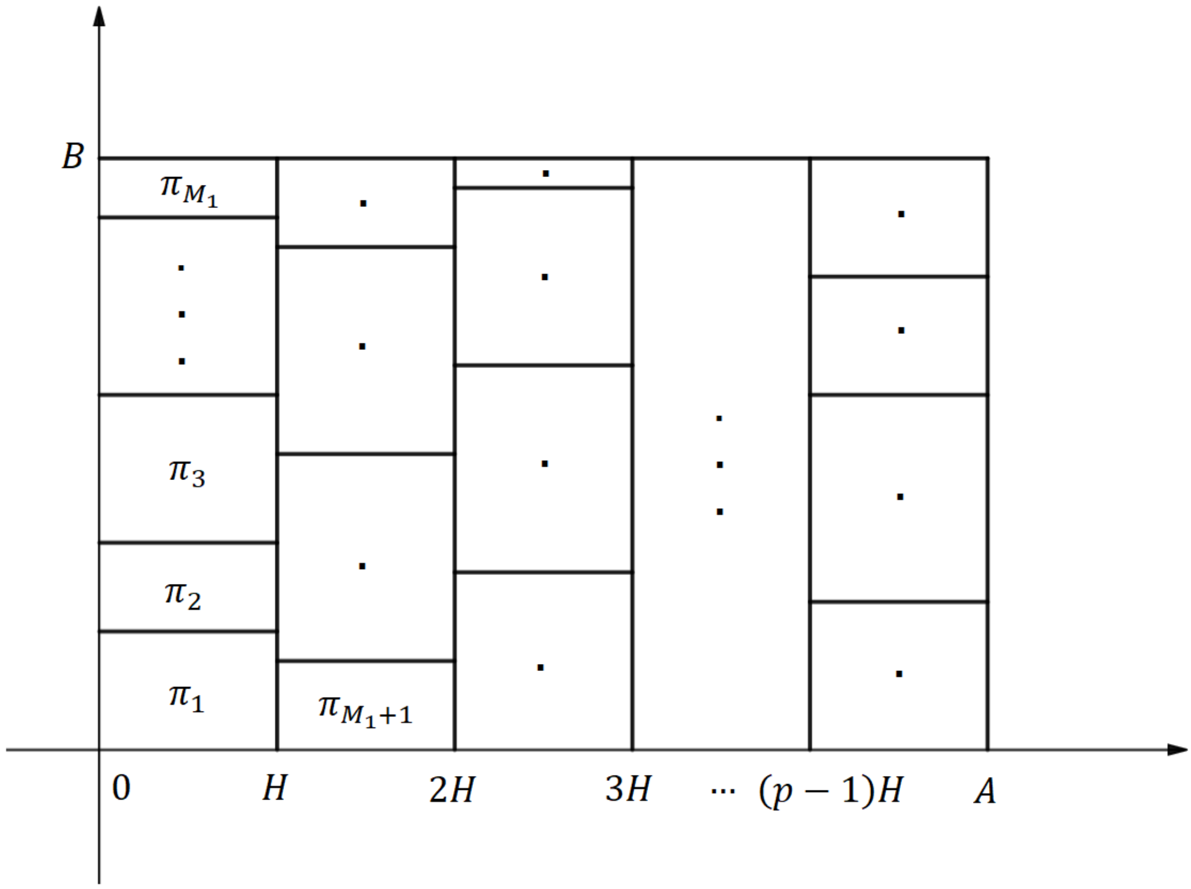


Рис. 2.3

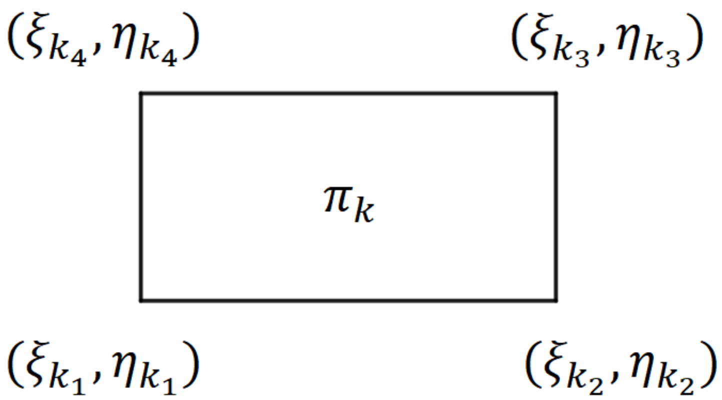
Таким чином на початку ми передбачали розрізати прямокутник Π на M прямокутників π_k, але запропонований алгоритм розрізання може збільшити кількість прямокутників Π_k. Будемо підбирати числа M і p так, щоб p було суттєво меншим M.

Числа M та $\tilde{M}$ мають бути суттєво меншими від $N_1 \cdot N_2$. Нагадаємо, що $N_1 \cdot N_2$ є кількістю малих квадратів із стороною h, з яких складається прямокутник Π.

Таким чином після виконання узагальненого алгоритму Лигуна-Шумейка ми отримали $\tilde{M}$ прямокутників Π_k, за допомогою яких можна отримати асимптотично оптимальну апроксимацію функції $f(x, y)$ білінійним сплайном [2].

Враховуючи, що білінійні сплайни є локальними, то достатньо навести формули для побудови білінійного сплайна для одного прямокутника π_k.

Зафіксуємо деякий прямокутник π_k. Позначимо його вершини (ξ_{k_1}, η_{k_1}), (ξ_{k_2}, η_{k_2}), (ξ_{k_3}, η_{k_3}), (ξ_{k_4}, η_{k_4}), а значення функції в цих точках позначимо відповідно ζ_{k_1}, ζ_{k_2}, ζ_{k_3}, ζ_{k_4}.



$$\xi_{k_2} - \xi_{k_1} = \xi_{k_3} - \xi_{k_4} = H$$

$$\eta_{k_4} - \eta_{k_1} = \eta_{k_3} - \eta_{k_2} = \tau_k$$

Введемо нормовані змінні

$$u = H^{-1} \cdot (x - \xi_{k_1}), v = \tau_k^{-1} \cdot (y - \eta_{k_1})$$

Для точок $(x, y) \in \Pi_k$ маємо $(u, v) \in [0; 1] \times [0; 1] = [0; 1]^2$.

Зображення білінійного сплайна $S_{11} \cdot (x, y)$ має вигляд

$$S_{11}(x, y) = (1 - v)[(1 - u)\zeta_{k_1} + u\zeta_{k_2}] + v[(1 - u)\zeta_{k_4} + u\zeta_{k_3}].$$

Таким чином $S_{11}(x, y)$ на π_k є многочленом 1-го степеня відносно u і v, а отже і відносно x та y.

Разом з тим вираз для $S_{11}(x, y)$ містить і нелінійну складову у вигляді добутку uv.

2.3 Опис узагальненого алгоритму Лигуна-Шумейка для пошуку асимптотично оптимальних вузлів сплайн-інтерполяції функцій двох змінних

Нехай $z = f(x, y)$ двічі неперервно диференційовна функція двох змінних, що визначена на квадраті

$$Q = [0; 1] \times [0; 1]$$

На Q задана решітка вузлів інтерполяції (x_i, y_j)

$$x_i = 0.01i, \qquad y_j = 0,01j, \qquad i, j = \overline{0; 100}$$

$\{F(i, j)\}_{i,j=\overline{0,100}}$ - двовимірний масив значень функції $f(x, y)$ в точках (x_i, y_j)

$$F(i, j) = f(x_i, y_j).$$

$H = 0,1$ - ширина вертикальних смуг Π_r, $r = \overline{1, 10}$, на які розрізається квадрат Q. Визначення двовимірного масиву $G(i, j)$ $i, j = 1, 100$.

$$DO \qquad i = \overline{1, 99}$$

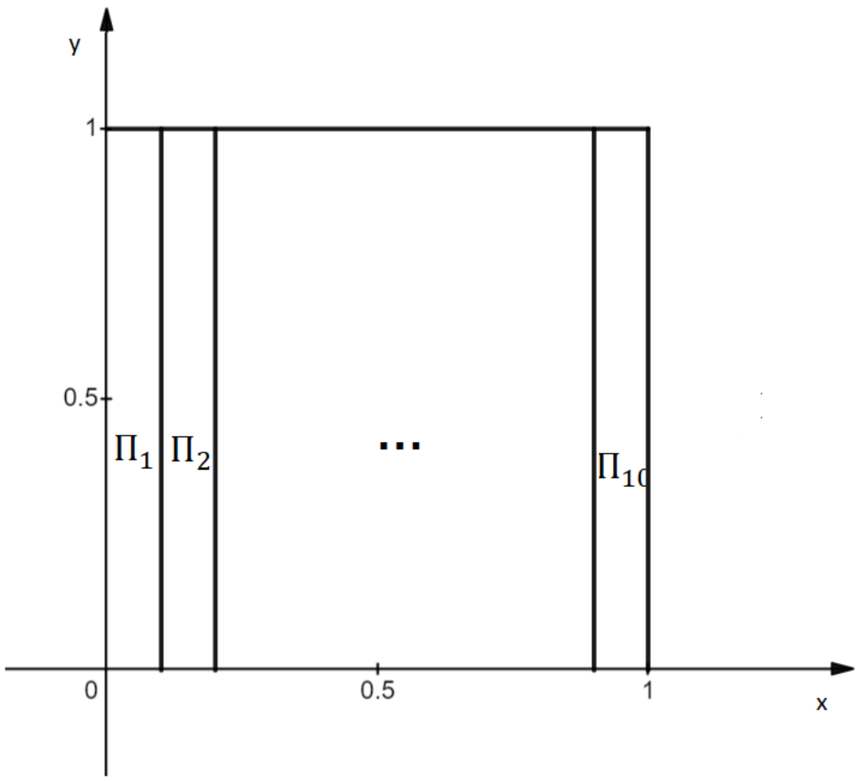


Рис. 2.4: Розрізання квадрата Q на 10 вертикальних смуг Π_r

$$DO \qquad j = \overline{1,99}$$

$$D_{xx} = F(i+1,j) - 2F(i,j) + F(i-1,j),$$

$$D_{yy} = F(i,j+1) - 2F(i,j) + F(i,j-1),$$

$$D_{xy} = 0.25(F(i+1,j+1) + F(i-1,j-1) - F(i+1,j-1) - F(i-1,j+1))$$

$$R = max(|D_{xx}|, |D_{yy}|, |D_{xy}|)$$

$$G(i,j) = R$$

$$end \qquad end$$

Додатково визначимо

$$G(100;100) = G(99;99)$$

$$DO \qquad i = \overline{1;99}$$

$$G(i,100) = G(i,99),$$

$$G(100, i) = G(99, i),$$

$$end$$

Визначимо двовимірний масив $S(r, j)$ $r = \overline{1; 10}$ $j = \overline{0; 100}$.

$$DO \qquad r = \overline{1, 10}$$

$$S(r; 0) = 0$$

$$DO \qquad j = \overline{1, 100}$$

$$S(r, j) = S(r, j - 1) + \sum_{10(r-1)+1}^{10r} G(i, j)$$

$$end \qquad end$$

$$T = 0.02 \cdot \sum_{r=1}^{10} S(r; 100)$$

$Y(k)$ - ордината верхньої сторони прямокутника π_k.

$z(k)$ - ордината нижньої сторони прямокутника π_k.

$r1(k)$ - номер смуги, якому належить π_k.

$E(k)$ - відхилення сплайна від $f(x, y)$ на π_k. $k = \overline{0, 60}$

$k = 0, B = 0.$

$$DO \qquad r = \overline{1, 10}$$

$$k = k + 1$$

$$A = T$$

$$DO \qquad j = \overline{1, 99}$$

$$IF \qquad S(r, j) < A \qquad THEN \qquad i = i + 1$$

$$FALSE \qquad Y(k) = 0.01i,$$

$$r1(k) = r, \qquad z(k) = B$$

$$B = Y(k), \qquad k = k + 1$$

$$A = A + T$$

$$end$$

$$k = k + 1, \qquad Y(k) = 1, \qquad z(k) = B$$

$$B = 0$$

$$end$$

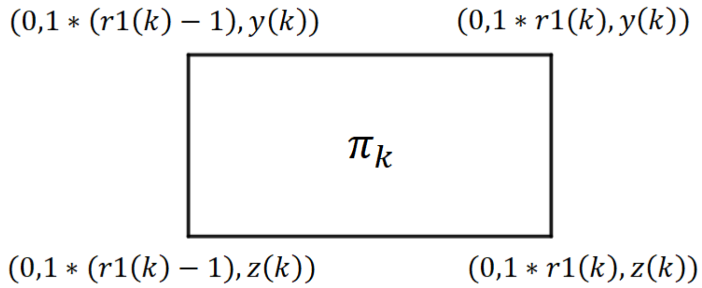


Рис. 2.5: Координати вершин прямокутника π_k

Двовимірний масив $S11[i, j]$ - значення сплайна в точці (x_i, y_j), $i, j = \overline{0, 100}$.

KM - максимальне $k(50 \leq KM \leq 60)$

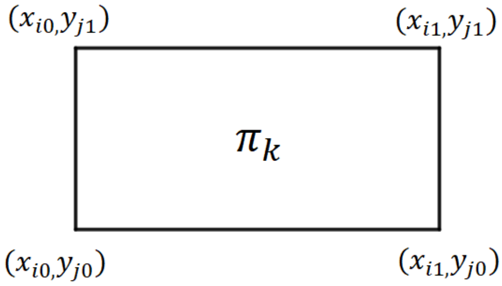
$$DO \qquad k = \overline{1, KM}$$

$$E[k] = 0$$

$$i0 = 10 \cdot (r1(k) - 1), \qquad i1 = 10 \cdot r1(k)$$

$$j0 = 10 \cdot z(k), \qquad j1 = 10 \cdot y(k)$$

$$H = (j1 - j0)^{-1}$$



$$A = F(i0, j0), \qquad B = F(i1, j0)$$

$$A1 = F(i0, j1), \qquad B1 = F(i1, j1)$$

$$DO \qquad i = \overline{i0, i1}$$

$$u = 0.1 \cdot (i - i0)$$

$$DO \qquad i = \overline{j0, j1}$$

$$v = (j - j0) \cdot H$$

$$S11[i, j] = (1 - v) \cdot ((1 - u)A + uB) + v \cdot ((1 - u)A1 + uB1)$$

$$E[k] = max(E[k], |F(i, j) - S11(i, j)|)$$

Покладемо $f(x, y) = T_n(1 - x - y)$ (або $T_n(x + y)$, $T_n(2xy)$, $T_n(x^2 + y^2)$), де T_n - многочлен Чебишова, $n = 5; 6; 7; 8; 9; 10$ (див. рис.2.6).

$$T_n(x) = cos(n \cdot arccosx), \qquad n \in N$$

$$T_0(x) = 1, T_1(x) = x$$

Рекурентна ф-ма

$$T_{n+1}(x) = 2x \cdot T_1(x) - T_0(x) = 2x \cdot x - 1 = 2x^2 - 1$$

$$T_3(x) = 2xT_2(x) - T_1(x) = 2x(2x^2 - 1) - x = 4x^3 - 2x - x = 4x^3 - 3x$$

$$T_4(x) = 2x \cdot T_3(x) - T_2(x) = 8x^4 - 8x^2 + 1$$

$$\ldots$$

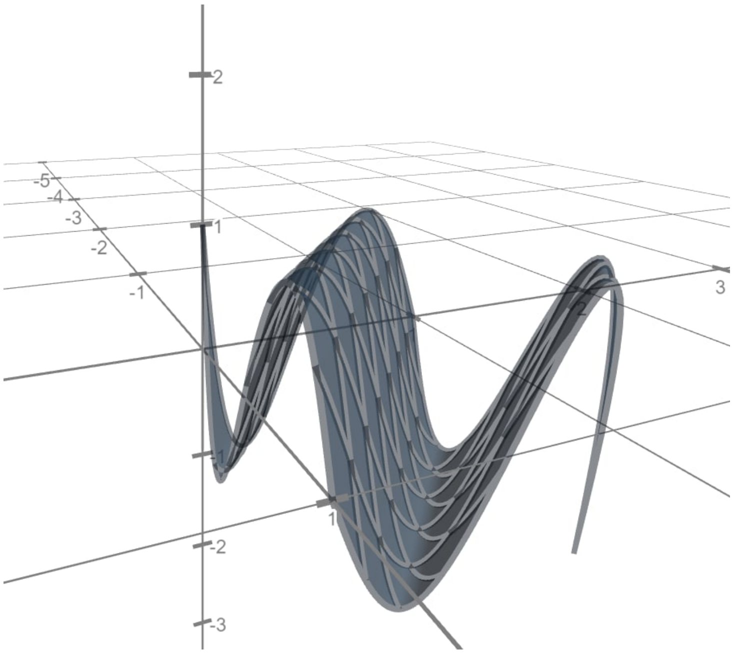


Рис. 2.6: Зображення $T_5(1 - x - y)$ *на квадраті* $[0,1] \times [0,1]$

В Додатку А наведена реалізація заданого алгоритму для пошуку асимптотично оптимальних вузлів сплайн-інтерполяції функцій двох змінних на Python та перевірка оптимальності алгоритму. На малюнку (2.7) зображена отримана за допомогою обчислень нерівномірна сітка.

2.3.1 Задача обрання оптимальної сітки з меншою кількістю вузлів

На вході дається значення функції y_k з досить густою рівномірною сіткою. Необхідно зробити певні перетворення, за допомогою яких, отримаємо близьку до оптимальної, нерівномірну сітку з більшими кроками, які беруться з умови, що інтеграл від цієї функції по таким інтервалам буде однаковий.

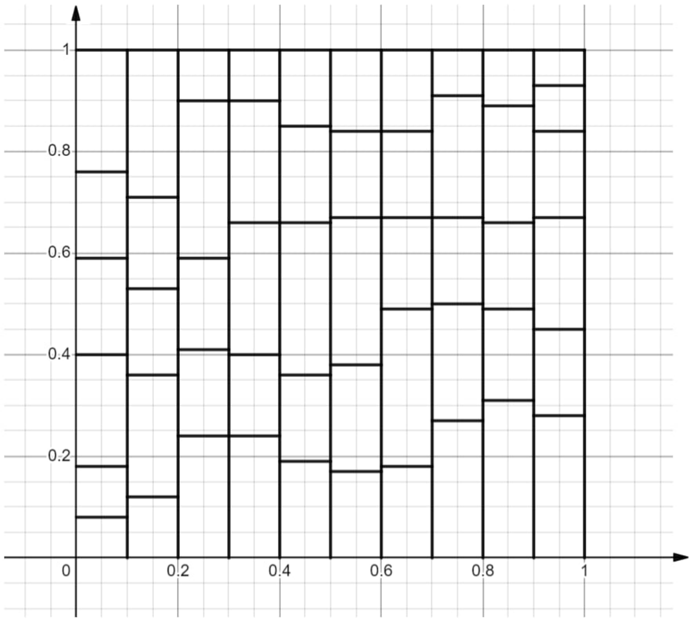


Рис. 2.7: Зображення отриманої асимптотично оптимальної сітки у вигляді прямокутників π_k, на які поділені смуги Π_r

Нехай задані значення функції $f(x)$ в вузлах рівномірної сітки.

$$\Delta_n(h) = \{x_k = kh, k = \overline{0, N}\}$$

$$f(x_k) = y_k$$

$$f \in C^4[0, b]$$

$$h = b/N$$

Значення $h = 0.01/0.05/0.1$.

$$f(x) = sin(\frac{\pi}{x + \alpha})$$

$$\alpha = 0.1/0.2$$

$$0 <= x <= 5$$

Визначаємо величини μ_k, які є наближеннями для $f''(x_k)$.

$$\mu_k = h^{-2}(-3y_k + \frac{5}{3}(y_{k+1} + y_{k-1})) - \frac{1}{6}(y_{k+2} + y_{k-2})$$

$$k = \overline{2, N-2}$$

$$p_k = \frac{\mu_{k+1} - 2\mu_k + \mu_{k-1}}{h^2}$$

$$k = \overline{3, N-3}$$

P_k - наближені значення $y^{IV}(x_k)$

$$\mu_k = |p_k|^{1/4}$$

$$k = \overline{3, N-3}$$

$$\mu_0 = \mu_1 = \mu_2 = \mu_3$$

$$\mu_{N-3} = \mu_{N-2} = \mu_{N-1} = \mu_N$$

$$\varphi_0 = 0, \varphi_1 = h\mu_1$$

$$\varphi_{K+1} = \varphi_K + h\mu_{K+1}$$

$$k = \overline{1, N}$$

$$d = \frac{\varphi_N}{m}, m << N$$

$(m+1)-$ кількість вузлів оптимальної сітки

$$\widetilde{\Delta_m} = \{z_0, z_1, z_2 ... z_m\} \in [0; b]$$

$$z_0 = 0$$

$$z_N = x_N$$

$$Y_0 = y_0, Y_m = y_N$$

$$Y_0' = \frac{4y_1 - y_2 - 3y_0}{2h}$$

$$Y_m' = \frac{y_{N-2} - 4y_{N-1} + y_N}{2h}$$

$$A = 0$$

2.3.2 Алгоритм знаходження вузлів оптимальної сітки z_j та похідних Y_j

DO $j = \overline{1, m-1}$ step 1

DO $k = \overline{1, N}$ step 1

$A = A + \mu_k$

IF $A < jd$ THEN $k = k + 1$ false $z_j = x_{k-1}$

$$u = \frac{4}{3}y_k - \frac{1}{6}(y_{k-1} + y_{k+1})$$

$$v = \frac{4}{3}y_{k-2} - \frac{1}{6}(y_{k-3} + y_{k-2})$$

$$Y'_j = \frac{u - v}{2h}$$

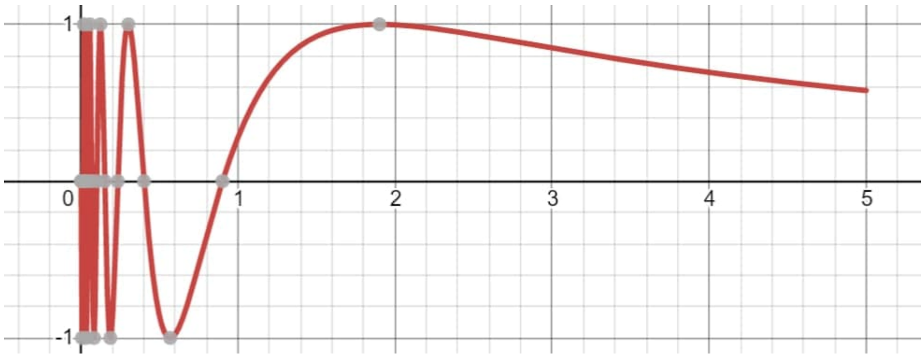


Рис. 2.8: Функція $f(x) = sin(\frac{\pi}{x+\alpha})$

Для перевірки правильності алгоритму був виконаний тест на правильність визначення вузлів z_j.

```
+ Код   + Текст

m = 16
d = 2
N = 1024
h=1

mu_k = np.zeros(N+1)
mu_k[0] = 0

for k in range (1, N):
    mu_k[k] = math.sqrt(k) - math.sqrt(k-1)

z = np.zeros(m+1)
z[0] = 0
z[m] = 1024

A = 0

for j in range (1, m):
  for k in range (1, N):
    A+=mu_k[k]
    if A>=j*d:
      #z[j] = (k-1)*h
      z[j] = pow(2*j, 2)-1
      break
```

Рис. 2.9: Тест на перевірку алгоритму

```
      break

print(z)

[   0.    3.   15.   35.   63.   99.  143.  195.  255.  323.  399.  483.
  575.  675.  783.  899. 1024.]
```

Рис. 2.10: Результати

В Додатку Б наведена реалізація заданого алгоритму на Python та знаходження кубічних сплайнів для кожного проміжку $[z_{j-1}, z_j]$ отриманої нерівномірної оптимальної сітки.

Висновки

1. Результати розрахунків підтвердили можливість використання методу Лигуна-Шумейка для оптимізації сплайн-інтерполяції функцій двох змінних.

2.В запропонованому методі слід розглянути обмеження на форму прямокутників $\boldsymbol{\pi}_k$, які більш точно адаптовані під поведінку інтерпольованої функції.

3. Окрім лінійних сплайнів двох змінних метод Лигуна-Шумейко можна узагальнити і на випадок кубічних сплайнів двох змінних.

Список літератури

[1] Лигун А.А., Шумейко А.А. Об оптимальном выборе узлов при приближении функций интерполяционными сплайнами//Журн. вычисл. матем. и матем. физ. – 1984.–24, №9. –С. 1283-1293

[2] Завьялов Ю. С., Квасов Б. И., Мирошниченко В. Л. Методы сплайн-функций. – М.: Наука, 1980. – 352 с.

[3] Стренг Г., Фикс Дж. Теория метода конечных элементов. Пер. с англ. - М., 1977, 352 с.

[4] Алберг Дж., Нильсон Э., Уолш Дж. (Ahlberg J. H., Nilson E. N., Walsh J. L.). Теория сплайнов и ее приложения. – М.: Мир, 1972. – 316 с.

[5] Кашпіровський О. І. Про локальні апроксимації обмежених розв'язків нескінченних систем різницевих рівнянь / О. І. Кашпіровський, О. В. Семенів, В. О. Яценко // Наукові записки Національного університету «Києво-Могилянська академія». — 2007. — Т. 61 : Фізико-математичні науки. — С. 17–22

[6] Городній М. Ф. Властивості розв'язків різницевих і диференціальних рівнянь та їх стохастичних аналогів у банаховому просторі. – Автореф, дис. ... докт. фіз. - мат. наук. – К., 2004. – 32 с.

[7] Andrii Cheredarchuk, Oleksii Kashpirovskyi, Galyna Kriukova, Maksym Sarana. Towards Image Processing Based on System of Difference

Equations. 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS).

[8] Пашковский С. Вычислительные применения многочленов и рядов Чебышева. – М.: Наука, 1983. – 384 с.

Додаток А

Реалізація алгоритму для пошуку асимптотично оптимальних вузлів сплайн-інтерполяції функцій двох змінних.

```
[ ]  #Імпорт бібліотек
     from pylab import figure, cm
     import matplotlib.pyplot as plt
     import numpy as np

     #Визначення многочлена Чебишова.

     cheb = np.polynomial.chebyshev.Chebyshev((0,0,0,0,0,1))
     coef = np.polynomial.chebyshev.cheb2poly(cheb.coef)

     print(coef)
     # [  0.,   5.,   0., -20.,   0.,  16.]

     poly = np.polynomial.Polynomial(coef)
     print(poly)
     # 0.0 + 5.0·x¹ + 0.0·x² - 20.0·x³ + 0.0·x⁴ + 16.0·x⁵

     [  0.   5.   0. -20.   0.  16.]
     0.0 + 5.0·x¹ + 0.0·x² - 20.0·x³ + 0.0·x⁴ + 16.0·x⁵
```

```
[ ]  #Покладемо f(x,y) = T_n(1-x-y), де T_n - многочлен
     #Чебишова степеня n. В даному випадку n=5.
     def f(x1,x2):
         return poly(1-x1-x2)

     #Покладемо розмірність 101
     dim = 101
     h = 1
     x = np.arange(start=0, stop=101, step=h)
     y = np.arange(start=0, stop=101, step=h)

     x_i = np.array([0.01*(element) for element in x])
     y_j = np.array([0.01*(element) for element in y])

     #Визначаємо F[i][j] - двовимірний масив значень
     #функції f(x,y) в точках (xi,yj), i,j від 0 до 100
     F = [ [0]*dim for i in range(dim)]
     for i in range (0,dim):
       for j in range (0,dim):
         F[i][j] = f(x_i[i],y_j[j]);
```

```
[ ]  #Визначення двовимірного масиву G
     G = [ [0]*dim for i in range(dim)]

     for i in range (1,dim-1):
       for j in range (1,dim-1):
         D_xx = F[i+1][j]-2*F[i][j]+F[i-1][j]
         D_yy = F[i][j+1]-2*F[i][j]+F[i][j-1]
         D_xy = 0.25 * (F[i+1][j+1]+F[i-1][j-1]-F[i+1][j-1]-F[i-1][j+1])
         R = np.max([np.abs(D_xx),np.abs(D_yy), np.abs(D_xy)])
         G[i][j]=np.sqrt(R)

     #Додатково визначимо
     G[100][100] = G[99][99]

     for i in range (1, dim-1):
         G[i][100] = G[i][99]
         G[100][i] = G[99][i]
```

```
[ ] #Визначимо двовимірний масив S(r,j)
    rowSize = 11
    S = [ [0]*dim for i in range(rowSize)]

    for r in range (1, rowSize):
      S[r][0] = 0
      for j in range (1,dim):
        sumG = 0
        for i in range (10*(r-1)+1, 10*r+1):
          sumG+=G[i][j]
        S[r][j] = S[r][j-1] + sumG
```

```
[ ] sumT = 0
    for r in range (1, rowSize):
      sumT+=S[r][100]
    T=0.02*sumT
    print(T)
    #8.69805939298281

    8.69805939298281
```

```
[ ] Y = [0 for i in range(53)]
    z = [0 for i in range(53)]
    r1 = [0 for i in range(53)]

    B=0
    k=0
    for r in range (1, 11):
      k=k+1
      A=T
      for i in range (1, 100):
        if(S[r][i]>=A):
          Y[k] = 0.01*i
          r1[k] = r
          z[k] = B
          B = Y[k]
          k=k+1
          A=A+T
      Y[k] = 1
      z[k]=B
      B=0
```

```
[ ] print(Y)
    #[0, 0.08, 0.18, 0.4, 0.59, 0.76, 1, 0.12, 0.36, 0.53, 0.71,

    [0, 0.08, 0.18, 0.4, 0.59, 0.76, 1, 0.12, 0.36, 0.53, 0.71, :
```

```
[ ] #Перевірка точності асимптотично оптимального наближення
    E = [0 for i in range(53)]
    S11 = [[0]*dim for i in range(dim)]

    for k in range (1, 53):
      E[k]=0
      i0=round(10*(r1[k]-1))
      i1=round(10*r1[k])
      j0=round(10*z[k])
      j1=round(10*Y[k])
      H=1/(j1-j0)
      A=F[i0][j0]
      B=F[i1][j0]
      A1=F[i0][j1]
      B1=F[i1][j1]
```

```
[ ]   if(i0>0 and i1>0):
        for i in range (i0, i1):
         u = 0.1*(i-i0)
         for j in range (j0, j1):
           v=(j-j0)*H
           S11[i][j]=(1-v)*((1-u)*A+u*B)+v*((1-u)*A1+u*B1)
           print(np.abs(F[i][j]-S11[i][j]))
           E[k]=np.max([E[k],np.abs(F[i][j]-S11[i][j])])
    #Тож результати відхилення досить малі,
    #тож алгоритм асимптотично оптимально наближує функцію

    0.008099260799999952
    0.010664035199999922
    0.009996307199999832
    0.011737881600000388
    0.010836825600000322
    0.011755200000000243
    0.010733856000000208
    0.010829030400000217
    0.009801974400000302
    0.009073948800000176
    0.008157100800000178
    0.006605875200000139
    0.005916307200000109
```

Додаток Б

Реалізація алгоритму визначення оптимальних вузлів інтерполяції функції кубічним сплайном.

```
[ ]  import numpy as np
     import math
     import matplotlib.pyplot as plt
```

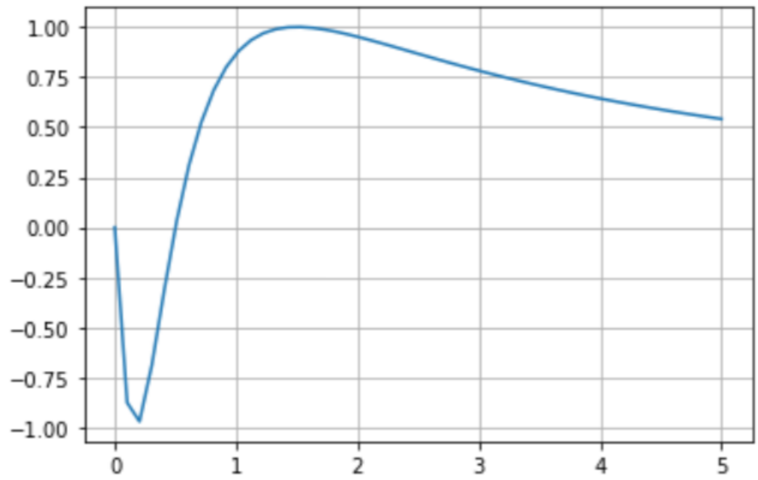
```
[ ]  alpha  = 0.5
     def f(x):
       return np.sin (math.pi / (x + alpha))

     a = 0
     b = 5
     h = 0.001
     N = 5000
```

```
[ ]  x_ = np.zeros(N+1)
     for k in range (0, N+1):
       x_[k] = k*h
```

```
[ ] x_ = np.zeros(N+1)
    for k in range (0, N+1):
      x_[k] = k*h
```

```
#Представлення значень функції на графіку від [0,5]
x1 = np.linspace(a,b)
y1 = [f(element) for element in x1]
plt.plot(x1,y1)
plt.grid(True)
```




```
[73] # k = [2, N-2]
     def mu_k(k):
       return (-3*f(k*h)+ 5/3 * ( f((k+1)*h) + f((k-1)*h) ) - 1/6 * ( f((k+2)*h) + f((k-2)*h) ) ) / pow(h,2)
```

```
[74] # k = [3, N-3]
     def p_k(k):
       return (mu_k(k+1) - 2*mu_k(k) + mu_k(k-1))/ pow(h,2)
```

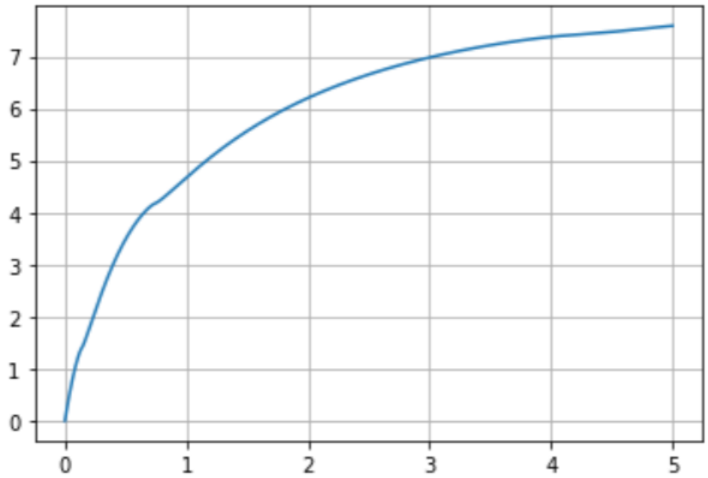
```
mu_ = np.zeros(N+1)
for k in range (3, N-3):
  mu_[k] = abs(p_k(k))**(1/float(4))
mu_[0] = mu_[3]
mu_[1] = mu_[3]
mu_[2] = mu_[3]
mu_[N-2] = mu_[N-3]
mu_[N-1] = mu_[N-3]
mu_[N] = mu_[N-3]
```

```
phi_ = np.zeros(N+1)
phi_[0] = 0
phi_[1] = h*mu_[1]
for k in range (1, N):
    phi_[k+1] = phi_[k] + h*mu_[k+1]

plt.plot(x_,phi_)
plt.grid(True)

print(phi_[N])
```

```
7.586435407681773
```




```
m = 20;
d = phi_[N]/m

z = np.zeros(m+1)
z[0] = a
z[m] = b

A = 0
Y = np.zeros(m+1)
Y_der = np.zeros(m+1)
Y_der[0] = (4*f(h) - f(2*h) -3*f(0))/2*h
Y_der[m] = (f((N-2)*h) - 4*f((N-1)*h) + f(N*h))/2*h
```

```
A = 0
j = 1
for k in range (3, N):
    if j>(m-1):
        break;
    A+=h*mu_[k]
    if A>=j*d:
      z[j] = (k-1)*h
      j+=1;
      u = 4/3*f(k*h) - 1/6*( f((k-1)*h) + f((k+1)*h) )
      v = 4/3*f((k-2)*h) - 1/6*( f((k-3)*h) + f((k-2)*h) )
      Y = f(z[j])
      Y_der[j] = (u-v)/2*h

print(z)
```

```
[0.    0.029 0.061 0.101 0.167 0.223 0.279 0.34  0.408 0.489 0.59  0.764
 0.956 1.145 1.357 1.605 1.906 2.287 2.795 3.558 5.   ]
```

```
def S(x,j):
    h = z[j] - z[j-1]
    t = (x-z[j-1])/h
    P = (f(z[j]) - f(z[j-1]))/h
    A = -2*P+( Y_der[j-1]+ Y_der[j])
    B = -A+P - Y_der[j-1]
    return f(z[j-1])+t*h*(Y_der[j-1]+t*(B+t*A))
```

```
plt.grid(True)
```