**Міністерство освіти і науки України**

**Луцький національний технічний університет**
(повне найменування закладу вищої освіти)

**Факультет комп'ютерних та інформаційних технологій**
(повне найменування факультету)

**Кафедра комп'ютерної інженерії та кібербезпеки**
(повне найменування кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА
# ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

## ПРОГРАМНИЙ КОМПЛЕКС ЗАСОБАМИ MATLAB ДЛЯ ІНТЕРПОЛЯЦІЇ ТАБЛИЧНО ЗАДАНИХ ФУНКЦІЙ

## A SOFTWARE COMPLEX BY MEANS MATLAB FOR INTERPOLATION OF TABULAR FUNCTIONS

спеціальність 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія
(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІз-41
Слівінська Яніна Юріївна

________________________
(підпис)

Керівник:
к.т.н., доцент
Пех Петро Антонович

________________________
(підпис)

Кваліфікаційну роботу
допущено до захисту
« _____ » червня_____ 2023 р.
Гарант освітньої програми:
к.т.н., доцент
Лавренчук Світлана Василівна

________________________
(підпис)

Луцьк – 2023 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет *комп’ютерних та інформаційних технологій*
Кафедра *комп’ютерної інженерії та кібербезпеки*
Ступінь вищої освіти: *бакалавр*
Галузь знань: *12 Інформаційні технології*
Спеціальність: *123 Комп’ютерна інженерія*
Освітня програма: *«Комп’ютерна інженерія»*

ЗАТВЕРДЖУЮ
Завідувач кафедри
____________ проф. Н.Чернящук
«_______» _________ 2023 р.

# ЗАВДАННЯ
# НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

*Слівінській Яніні Юріївній*
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Програмний комплекс засобами Matlab для інтерполяції таблично заданих функцій*

Керівник роботи *к.т.н., доцент Пех Петро Антонович*

затверджені наказом закладу вищої освіти від «28» грудня 2022 року № 982/01-02
2. Строк подання здобувачем вищої освіти кваліфікаційної робо 01.06.2023р.

3. Вихідні дані до роботи *Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області та різні інтернет-ресурси технічного спрямування*

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):
*Вступ*
*Огляд математичних та програмних методів інтерполяції табличного заданих функцій*
*Методика інтерполяції таблично заданих функцій*
*Розробка програмного комплексу для інтерполяції таблично заданих функцій засобами Matlab*
*Висновки*

5. Перелік графічного (ілюстративного) матеріалу:
*Програма*
*Інтерфейс програми*
*Схема роботи програмного продукту*

6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис | |
|---|---|---|---|
| | | завдання видав | завдання прийняв |
| *Огляд математичних та програмних методів інтерполяції таблично заданих функцій* | *Пех П.А.* | | |
| *Методика інтерполяції таблично заданих функцій* | *Пех П.А.* | | |
| *Розробка програмного комплексу для інтерполяції таблично заданих функцій засобами* Matlab | *Пех П.А.* | | |
| *Висновки* | | | |

7. Дата видачі завдання 01.11.2022 р.

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
|---|---|---|---|
| 1. | *Обгрунтування теми* | До 15.11.2022 р. | Виконано |
| 2. | *Огляд літератури із досліджуваної проблеми* | До 15.12.2022 р. | Виконано |
| 3. | *Розділ 1.* | До 02.02.2023 р. | Виконано |
| 4. | *Розділ 2.* | До 02.03.2023 р. | Виконано |
| 5. | *Висновки та пропозиції* | До 02.04.2023 р. | Виконано |
| 6. | *Формування списку використаних джерел* | До 15.04.2023 р. | Виконано |
| 7. | *Формування додатків* | До 02.05.2023 р. | Виконано |
| 8. | *Оформлення ілюстративного матеріалу* | До 15.05.2023 р. | Виконано |
| 9. | *Нормоконтроль* | До 25.05.2023 р. | Виконано |
| 10. | *Інструментальна перевірка на академічний плагіат* | До 01.06.2023 р | Виконано |
| 11. | *Представлення кваліфікаційної роботи бакалавра до захисту* | До 07.06.2023 р. | Виконано |

**Здобувач вищої освіти** __________ Слівінська Я.Ю.
(підпис) (прізвище, ініціали)

**Керівник кваліфікаційної роботи** __________ Пех П.А.
(підпис) (прізвище, ініціали)

# АНОТАЦІЯ


Слівінська Я. Ю. Програмний комплекс засобами MatLab для інтерполяції табличнo заданих функцій. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп’ютерна інженерія» спеціальності 123 Комп’ютерна інженерія. Луцький національний технічний університет. Луцьк, 2023.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, списку використаних джерел.

Перший розділ присвяченно огляду різних методів інтерполяції, основні поняття інтерполяції та наведенно багато прикладів.

В другому розділі здійсненно огляд методики інтерполяції таблично заданих функцій, таких як методи Ньютона та метод Лагранжа.

Третій розділ описує розроблені програми для виконання інтерполяції таблично заданих функцій на графіках зростання, спадання, перегину.

Об'єкт – методи інтерполяції функції, що задані таблично.

Предмет – процес розробки програмного комплексу для виконання інтерполяції функції.

Метою роботи є розробка програмного комплексу, який виконує інтерполяцію таблично заданих функцій за допомогою MATLAB.



Ключові слова: інтерполяція, метод Ньютона, метод Лагранжа, функція, графік.

# ANNOTATION

Slivinska Y.Y. A software complex by means Matlab for interpolation of tabularfunctions. Manuscript.

Qualifying work of a bachelor of EP "Computer Engineering" specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2023.

Qualification work consists of an introduction, three sections, conclusions, references.

The first section  is devoted to the review of different methods of interpolation, the general concepts of interpolaton and many practical examples are given.

In the second section, an overview of the interpolation methodology of tabularfunctions is carried out such as the method of Newton and the method of Lagrange.

The third section is devoted to the description of the developed programs for performing interpolation of tabularfunctions on the growth graphs, decline graphs, inflection graphs.

The object – interpolation method of tabularfunctions.

The subject – the development process of software complex for perfoming interpolation function.

The aim of the work is to create a software complex that implementate the interpolation of tabularfunctions with the help of Matlab.

Keywords: interpolation, the method of Newton, the method of Lagrange, function, graph.

# ЗМІСТ

# ВСТУП

*Актуальність роботи.* Обчислення значень функцій, які задані таблично, завжди було і залишається актуальною задачею, оскільки вимагає вирішення ряду питань теоретичного та практичного характеру. Зокрема, значення функції часто отримуються експериментальним шляхом за певних дискретних значень вхідних факторів, і тоді гостро виникає питання, як визначити значення функції за інших значень цих факторів.

*Об'єкт дослідження* – програмні методи інтерполяції значень функції.

*Предмет дослідження* – розробка комплексу програм для інтерполяції значень функції засобами Matlab.

*Завдання, які розв'язані у кваліфікаційній роботі:*

- проаналізувати існуючі методи інтерполяції значень функції та їх програмної реалізації;
- розробити програмний комплекс засобами Matlab для вирішення задачі інтерполяції значень функції різними методами.

*Наукова новизна роботи.* У даній кваліфікаційній роботі буде розв'язана задача інтерполяції значень таблично заданих функцій за допомогою програмного комплексу, вперше розробленого здобувачем засобами Matlab.

*Практична цінність роботи.* Програмно-демонстраційний комплекс засобами Matlab має практичне спрямування і може бути використаний для виконання інтерполяції значень функції в інженерній практиці. Цей комплекс може бути використаний і навчальному процесі під час вивчення відповідної теми дисципліни «Методи обчислень та моделювання».

# РОЗДІЛ 1
# ОГЛЯД МЕТОДІВ ІНТЕРПОЛЯЦІЇ ТАБЛИЧНО ЗАДАНИХ ФУНКЦІЙ

## 1.1 Інтерполяція проміжних значень функції

Інтерполяція – це спосіб визначити значення функції для деяких проміжних значень аргумента, якщо сама функція задана множиною значень у табличному вигляді у певних точках, які називаються вузлами інтерполяції [1,2, 3]. Якщо проміжні значення аргумента знаходиться між вузлами інтерполяції (всередині таблиці), то говорять власне про інтерполяцію функції у вузькому сенсі цього слова; якщо ж проміжні значення аргумента є меншими від найменшого вузла або більші від найбільшого вузла інтерполяції (знаходяться поза таблицею), то говорять про екстраполяцію значень функції.

Як інтерполяцію, так і екстраполяцію значень функції можна виконати за допомогою різних методів, таких як метод Ньютона, метод Лагранжа, кубічна сплайн-інтерполяція тощо [4,5]. У загальному випадку задача інтерполяції полягає у виконанні наступних кроків [6,7]:

1. Задати множину значень функції у точках, які називаються вузлами інтерполяції, та множину проміжних значень аргумента, для яких мають бути знайдені проміжні значення функції.
2. Вибрати один із відомих математичних методів інтерполяції.
3. Знайти вигляд інтерполяційного многочлена для вибраного методу інтерполяції за заданих значень функції у вузлах інтерполяції.
4. Використовувати інтерполяційний многочлен, знайти значеня функції у проміжних точках, які не співпадають з вузлами інтерполяції.

Вибір методу інтерполяції залежить від типу функції та її властивостей, а також від точності, яку необхідно досягти. Для отримання найбільш точних результатів може бути виконана кубічна сплайн-інтерполяція, яка використовує кубічні функції для з'єднання точок і дозволяє забезпечити неперервність першої та другої похідних функції. Однак, залежно від вимог щодо швидкодії та ресурсів, можуть бути використані більш прості методи, такі як метод Ньютона або метод Лагранжа.

## 1.2 Метод Ньютона інтерполяції проміжних значень функції

Що таке метод Ньютона? Це один з методів інтерполяції функції, який дозволяє знайти апроксимуючу функцію на основі заданих табличних даних. Апроксимуюча функція будується у вигляді многочлена, який проходить через задані точки.

Метод Ньютона [1,8] є ефективним у випадку великої кількості точок таблиці і дозволяє знайти апроксимуючу функцію з досить високою точністю. Одна особливість цього методу є те, що він може бути застосований для інтерполяції функції в довільному порядку, починаючи з будь-якої точки. Тому він є більш гнучким, ніж інші методи інтерполяції, які вимагають, аби точки даних були упорядковані.

Інтерполяційний метод Ньютона широко застосовується у різних галузях науки, наприклад, в інженерії, математиці, фізиці, та інших.

Нехай невідому функцію $y = f(x)$ задано своїми числовими значеннями $y_0 = f(x_0)$, $y_1 = f(x_1)$, ..., $y_n = f(x_n)$; у точках $x_0, x_1, ..., x_n$; які називаються вузлами інтерполяції.

Метод Ньютона для інтерполяції значень функції базується на побудові інтерполяційного полінома, виходячи з такої базової формули:

$$\begin{aligned} P_n(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \\ &+ \ldots + a_n(x - x_0)(x - x_1)\ldots(x - x_{n-1}), \end{aligned} \tag{1.1}$$

де $(a_0, a_1, ..., a_n)$ - невідомі коефіцієнти;

$(x_0, x_1, ..., x_n)$ - заданні значення аргументу x (вузли інтерполяції);

Для обчислення многочлена Ньютона потрібно знати значення функції $f(x)$ в деяких вузлах $(x_0, x_1, ..., x_n)$. Ці вузли можуть бути рівномірно розташовані на відрізку $[a, b]$, або ж вони можуть бути розташовані довільним чином.

Метод Ньютона є ефективним і простим для інтерполяції. Але його недоліком є нестабільність при великій кількості вузлів, якщо вони є нерівновіддаленими, що призводить до меншої точності отримуваних результатів.

Ще одним недоліком методу Ньютона є те, що він використовує інформацію тільки з попередніх вузлів при обчисленні кожного нового коефіцієнту інтерполяційного полінома. Це може призводити до значної накопиченої похибки при обчисленні значень функції в точках, що лежать далеко від заданих вузлів.

Для розв'язання цих проблем можуть бути застосовані інші методи інтерполяції, наприклад, метод Лагранжа, кубічна інтерполяція, сплайн-інтерполяція тощо. Кожен з цих методів має свої переваги та недоліки і може бути більш ефективним у певних випадках.

Отже, метод Ньютона є одним з найпоширеніших інтерполяційних методів, але він має свої недоліки і не завжди може забезпечити потрібну точність результатів. Для досягнення кращих результатів можуть бути застосовані інші методи інтерполяції в залежності від конкретної задачі.

### 1.3 Перша інтерполяційна формула Ньютона

Формула Ньютона [9, 10] має дещо інакший вигляд, ніж многочлен (1.1), однак базуючись на виразі (1.1), за декілька перетворень ми можемо отримати першу інтерполяційну формулу Ньютона.

Як зазначалося раніше, результат інтерполяції функції буде знайдений з меншою похибкою, коли вузли $(x_0, x_1, ..., x_n)$, є рівновіддаленими, а відстань між ними буде постійною, і називається кроком функції, тобто:

$$h = x_{n+1} - x_n = const\ (n = 0,1,2...). \quad (1.2)$$

Далі для перетворень вихідного многочлена нам будуть потрібні скінченні різниці [11].

Нехай функцію $y = f(x)$ задано такими значеннями $y_0 = f(x_0)$ , $y_1 = f(x_1)$ , $y_2 = f(x_2)$ , …, $y_n = f(x_n)$ ; причому $x_1 = x_0 + h$ , $x_2 = x_0 + 2h$ , ..., $x_n = x_0 + nh$ ; та де $h{=}x_1{-}x_0{=}x_2{-}x_1{=}...{=}x_n{-}x_{n-1}$.

Скінченна різниця – це різниця між сусідніми значеннями функції інтерполяції, наприклад різниця першого порядку:

$$\Delta y_0 = y_1 - y_0,$$
$$\Delta y_1 = y_2 - y_1,$$
$$\Delta y_n = y_{n+1} - y_n \quad (n = 0,1,2,...).$$

Різниці другого порядку обчислюються з різниць першого порядку а саме:

$$\Delta^2 y_0 = \Delta y_1 - \Delta y_0,$$
$$\Delta^2 y_1 = \Delta y_2 - \Delta y_1,$$
$$\Delta^2 y_n = \Delta y_{n+1} - \Delta y_n (n = 0,1,2,...).$$

Отже, продовжуючи таким чином рахувати ці різниці, можна побудувати таблицю скінченних різниць довільного порядку, що досить зручно для використання. Таблиці скінченних різниць бувають діагональними (таблиця 1.1) та горизонтальними (таблиця 1.2).

Таблиця 1.1 – Діагональна таблиця скінченних різниць.

| x | y | Δy | $\Delta^2 y$ | $\Delta^3 y$ |
|---|---|---|---|---|
| $x_0$ | $y_0$ | $\Delta y_0$ | $\Delta^2 y_0$ | $\Delta^3 y_0$ |
| $x_1$ | $y_1$ | $\Delta y_1$ | $\Delta^2 y_1$ | - |
| $x_2$ | $y_2$ | $\Delta y_2$ | - | - |

Таблиця 1.2 – Горизонтальна таблиця скінченних різниць.

| x | y | Δy | $\Delta^2 y$ | $\Delta^3 y$ |
|---|---|---|---|---|
| $x_0$ | $y_0$ | - | - | - |

| | | | | |
|---|---|---|---|---|
| - | - | Δy0 | - | - |
| x1 | y1 | - | Δ2y0 | - |
| - | - | Δy1 | - | Δ3y0 |
| x2 | y2 | - | Δ2y1 | - |
| - | - | Δy2 | - | - |
| x3 | y3 | - | - | - |

Скінченні різниці i – го порядку порядку визначаються так:

$$\Delta^i y_0 = \Delta^{i-1} y_1 - \Delta^{i-1} y_0,$$
$$\Delta^i y_1 = \Delta^{i-1} y_2 - \Delta^{i-1} y_1,$$
$$\Delta^i y_n = \Delta^{i-1} y_{n+1} - \Delta^{i-1} y_n \, (i = 1,2,...).$$

Різниці довільних порядків можуть бути виражені через значення функції:

$$\Delta y_n = y_{n+1} - y_n,$$
$$\Delta^2 y_n = \Delta y_{n+1} - \Delta y_n = (y_{n+2} - y_{n+1}) - (y_{n+1} - y_n) = y_{n+2} - 2y_{n+1} + y_n,$$
$$\Delta^3 y_n = \Delta^2 y_{n+1} - \Delta^2 y_n = (y_{n+3} - 2y_{n+2} + y_{n+1}) - (y_{n+1} - 2y_{n+1} + y_n) =$$
$$= y_{n+3} - 3y_{n+2} + 3y_{n+1} - y_n.$$

.

Тепер перейдемо до визначення невідомих коефіцієнтів $a_0, a_1, \ldots, a_n$ , многочлена (1.1).

Підставивши $x = x_0$ , у формулу (1.1), отримаємо $P_n\,(x_0) = a_0$ . Та ми знаємо що $P_n(x_i) = y_i$ , а точніше $P_n(x_0) = y_0$ , а це означає, що $a_0 = y_0$ .

Тепер зробимо те ж саме для коефіцієнта $a_1$ , позначаємо $x = x_1$ та підставляємо в формулу (1.1), і отримуємо:

$$P_n(x_1) = a_0 + a_1(x_1 - x_0).$$

Враховуючи, що $P_n(x_1) = y_1$ , та $a_0 = x_0$ з попередніх перетворень, а також, що $x_1 - x_0 = h$ , можемо записати таку формулу $y_1 = y_0 + a_1 h$ , звідки визначаємо $a_1$ і отримуємо $a_1 = \frac{y_1 - y_0}{h}$. Як згадували раніше, $y_1 - y_0 = \Delta y_0$ - скінченна різниця першого порядку, тому $a_1 = \frac{\Delta y_0}{h}$.

Вважаючи $x = x_2$ , і підставляючи в формулу (1.1), отримаємо:

$$P_n(x_2) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1).$$

Враховуючи, що $P_n(x_2) = y_2$, $a_0 = y_0$, $a_1 = \frac{\Delta y_0}{h}$, $x_2 - x_0 = 2h$, $x_1 - x_0 = h$, то

$$y_2 = y_0 + \frac{\Delta y_0}{h} 2h + a_2 2hh.$$

Звідси отримуємо:

$$a_2 2hh = y_2 - y_0 - \frac{\Delta y_0}{h} 2h,$$

$$a_2 = \frac{y_2 - y_0 - 2\Delta y_0}{2h^2}.$$

Як ми згадували раніше, скінченні різниці виражені через значення функції, то можна записати так:

$$y_2 - y_0 - 2\Delta y_0 = \Delta^2 y_0.$$

Отже:

$$a_2 = \frac{\Delta^2 y_0}{2h^2}.$$

Тому, продовжуючи такі перетворення, можна визначити наступні коефіцієнти:

$$a_3 = \frac{\Delta^3 y_0}{3!h^3},\ \dots,\ a_4 = \frac{\Delta^4 y_0}{4!h^4},\ \dots,\ a_k = \frac{\Delta^k y_0}{k!h^k},\ \dots,\ a_n = \frac{\Delta^n y_0}{n!h^n}.$$

Тепер ми знаємо усі коефіцієнти, і можемо їх підставити у формулу (1.1):

$$P_n(x) = y_0 + \frac{\Delta y_0}{h}(x - x_0) + \frac{\Delta^2 y_0}{2!h}(x - x_0)(x - x_1) + \frac{\Delta^3 y_0}{3!h^3}(x - x_0)(x - x_1)(x - x_2) +$$
$$+ \dots + \frac{\Delta^n y_0}{n!h^n}(x - x_0)(x - x_1)\dots(x - x_{n-1}).$$

Це і є перша інтерполяційна формула Ньютона. Ми можемо записати її ще простіше, позначивши

$$q = \frac{x - x_0}{h}. \tag{1.3}$$

що є числом кроків для досягнення точки x, виходячи з точки $x_0$.

Мають місце такі рівності:

$$\frac{(x - x_1)}{h} = \frac{x - (x_0 + h)}{h} = \frac{x - x_0 - h}{h} = q - 1,$$

$$\frac{x - x_2}{h} = q - 2,$$

$$\frac{x - x_{n-1}}{h} = q - n + 1.$$

І тепер перша інтерполяційна формула Ньютона набуває ось такого вигляду:

$$P_n(x) = y_0 + q\Delta y_0 + \frac{q(q-1)}{2!}\Delta^2 y_0 + \frac{q(q-1)(q-2)}{3!}\Delta^3 y_0 + \\ + \ldots + \frac{q(q-1)(q-2)\ldots(q-n+1)}{n!}\Delta^n y_0. \qquad (1.4)$$

## 1.4 Друга інтерполяційна формула Ньютона

Запишемо поліном Ньютона у вигляді многочлена:

$$P_n(x) = a_0 + a_1(x - x_n) + a_2(x - x_n)(x - x_{n-1}) + \\ + \ldots + a_n(x - x_n)(x - x_{n-1})\ldots(x - x_1). \qquad (1.5)$$

Тепер визначимо коефіцієнти $a_0$, $a_1$, $a_2$, …, $a_n$. дещо по іншому.Так як з початкової умови $P_n(x_i) = y_i\ (i = 0, 1, 2, \ldots, n)$, то, підставивши $x = x_n$, у многочлен (1.5) то отримуємо $a_0 = y_n$.

Тепер нехай $x = x_{n-1}$, та враховуючи $P_n(x_{n-1}) = y_{n-1}$, $a_0 = y_n$, а також $x_{n-1} - x_n = -h$, отримуємо $y_{n-1} = y_n - a_1 h$.

Визначаємо звідси коефіцієнт $a_1$:

$$a_1 = -\frac{y_{n-1} - y_n}{h} = \frac{y_n - y_{n-1}}{h} = \frac{\Delta y_{n-1}}{h}.$$

Нехай $x = x_{n-2}$ , та враховуючи, що $P_n(x_{n-1}) = y_{n-1}$ , $a_0 = y_n$ , та $a_1 = \frac{\Delta y_{n-1}}{h}$ , отримуємо

$$a_2 = \frac{y_n - 2y_{n-1} + y_{n-2}}{2h^2} = \frac{\Delta^2 y_{n-2}}{2!h^2}.$$

Продовжуючи такі перетворення, можемо знайти вирази і для інших коефіцієнтів:

$$a_3 = \frac{\Delta^3 y_{n-3}}{3!h^3}, \quad \ldots, \quad a_k = \frac{\Delta^k y_{n-k}}{k!h^k}, \quad \ldots, \quad a_n = \frac{\Delta^n y_0}{n!h^n}.$$

Тепер, маючи коефіцієнти, підставляємо їх у поліном (1.5):

$$P_n(x) = y_n + \frac{\Delta y_{n-1}}{h}(x - x_n) + \frac{\Delta^2 y_{n-2}}{2!h^2}(x - x_n)(x - x_{n-1}) +$$
$$+ \frac{\Delta^3 y_{n-3}}{3!h^3}(x - x_n)(x - x_{n-1})(x - x_{n-2}) + \ldots + \frac{\Delta^n y_0}{n!h^n}(x - x_n)(x - x_{n-1})\ldots(x - x_1).$$

Так ми отримали другу формулу Ньютона [8, 12].

Позначимо $q$ , для того щоб записати нашу формулу в більш зручному для обчислень вигляду:

$$q = \frac{x - x_n}{h}. \tag{1.6}$$

Мають місце рівності:

$$\frac{x - x_{n-1}}{h} = \frac{x - (x_n - h)}{h} = q + 1,$$
$$\frac{x - x_{n-2}}{h} = \frac{x - (x_n - 2h)}{h} = q + 2,$$
$$\ldots$$
$$\frac{x - x_1}{h} = \frac{x - [x_n - (n-1)h]}{h} = q + n - 1.$$

Підставивши у другу формулу Ньютона значення $q$ , отримаємо інший вигляд цієї формули:

$$P_n(x) = y_n + q\Delta\Delta_{n-1} + \frac{q(q+1)}{2!}\Delta^2 y_{n-2} + \frac{q(q+1)(q+2)}{3!}\Delta^3 y_{n-3} + \\ + \ldots + \frac{q(q+1)\ldots(q+n-1)}{n!}\Delta^n y_0. \quad (1.7)$$

## 1.5 Метод Лагранжа

Інтерполяція методом Лагранжа [13, 14, 15] в більшості випадків використовується для нерівновіддалених вузлів інтерполяції, тому будемо вважати, що:

$$h = x_{n+1} - x_n \neq const\,(n = 0,1,2\ldots).$$

Нехай на відрізку $[a,b]$ задано $(n+1)$ різних значень аргументів, $x_0, x_1, \ldots, x_n$, і відповідні їм значення функції $f(x_0) = y_0, f(x_1) = y_1, \ldots, f(x_n) = y_n$ .

Нам необхідно побудувати поліном так, щоб його значення у вузлах інтерполяції $x_0, x_1, \ldots, x_n$, співпадали зі значеннями функції $f(x)$, тобто

$$L(x_0) = y_0, L(x_1) = y_1, \ldots, L(x_n) = y_n.$$

Знайдемо інтерполяційний багаточлен $L_n(x)$ у такому вигляді:

$$L_n(x_i) = a_0(x - x_1)(x - x_2)\ldots(x - x_n) + a_1(x - x_0)(x - x_2)\ldots(x - x_n) + \\ + \ldots + a_i(x - x_0)(x - x_1)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n) + \\ + \ldots + a_n(x - x_0)(x - x_1)\ldots(x - x_{n-1}). \quad (1.8)$$

де $a_i\,(i = 1,2,\ldots,n)$ є невідомими коефіцієнтами.

Кожен доданок виразу є многочленом степення $n$, причому біля кожного коефіцієнта $a_i$ множника $x - x_i$ немає.

Тепер спробуємо знайти коефіцієнти $a_i$ , за даної нам вже умови $L(x_n) = y_n$, та підставивши $x = x_0$, у многочлен (1.8), отримаємо:

$$L_n(x_0) = y_0 = a_0(x_0 - x_1)(x_0 - x_2)\ldots(x_0 - x_n).$$

Звідси визначимо $a_0$:

$$a_0 = \frac{y_0}{(x_0 - x_1)(x_0 - x_2)\dots(x_0 - x_n)}.$$

Тепер спробуємо підставити $x = x_1$, у той самий многочлен (1.3):

$$L_n(x_1) = y_1 = a_1(x_1 - x_0)(x_1 - x_2)\dots(x_1 - x_n).$$

Визначимо $a_1$:

$$a_1 = \frac{y_1}{(x_1 - x_0)(x_1 - x_2)\dots(x_1 - x_n)}.$$

Аналогічно з'ясовуємо, що

$$a_n = \frac{y_n}{(x_n - x_0)(x_n - x_1)\dots(x_n - x_{n-1})(x_n - x_{n+1})\dots(x_n - x_{n-1})} (n = 0,1,\ \dots).$$

Підставивши знайденні значення коефіцієнтів у многочлен (1.8), ми отримаємо:

$$L_n(x) = y_0 \frac{(x - x_1)\dots(x - x_n)}{(x_0 - x_1)\dots(x_0 - x_n)} + y_1 \frac{(x - x_0)(x - x_2)\dots(x - x_n)}{(x_1 - x_0)(x_1 - x_2)\dots(x_1 - x_n)} +$$
$$\dots + y_n \frac{(x - x_0)(x - x_1)\dots(x - x_{n-1})}{(x_n - x_0)(x_n - x_1)\dots(x_n - x_{n-1})}.$$

Отримали інтерполяційну формулу Лагранжа. Тепер ми можемо записати неї у більш компактному вигляді:

$$L_n(x) = \sum_{i=0}^{n} \frac{(x - x_0)(x - x_1)\dots(x - x_{i-1})(x - x_{i+1})\dots(x - x_n)}{(x_i - x_0)(x_i - x_1)\dots(x_i - x_{i-1})(x_i - x_{i+1})\dots(x_i - x_n)} y_i. \qquad (1.9)$$

# РОЗДІЛ 2
# МЕТОДИКА ІНТЕРПОЛЯЦІЇ ТАБЛИЧНО ЗАДАНИХ ФУНКЦІЙ

## 2.1 Методика інтерполяції значень функції методом Ньютона

Розглянемо, як вирішується задача інтерполяції на конкретних прикладах.

Приклад 1. Обчислити значення таблично заданої функції (таблиця 2.1) для заданих значень аргументів (таблиця 2.2). При побудові таблиць різниць контролювати обчислення.

Таблиця 2.1 – Задана функція

| № | x | y |
|---|---|---|
| 1 | 2,180 | 5,61543 |
| 2 | 2,185 | 5,46693 |
| 3 | 2,190 | 5,32634 |
| 4 | 2,195 | 5,19304 |
| 5 | 2,200 | 5,06649 |
| 6 | 2,205 | 4,94619 |
| 7 | 2,210 | 4,83170 |
| 8 | 2,215 | 4,72261 |
| 9 | 2,220 | 4,61855 |
| 10 | 2,225 | 4,51919 |
| 11 | 2,230 | 4,42427 |

Таблиця 2.2 – Значення аргументів

| Значення аргументу |
|---|

Продовження таблиці 2.2

| | |
|---|---|
| $x_1$ | 2,179 |
| $x_2$ | 2,183 |
| $x_3$ | 2,227 |
| $x_4$ | 2,232 |

Спочатку будуємо таблицю скінченних різниць (таблиця 2.3)

Таблиця 2.3 – Скінченні різниці

| X | y | Δy | $\Delta^2$y | $\Delta^3$y | $\Delta^4$y |
|---|---|---|---|---|---|
| 2,180 | 5,61543 | -0,1485 | 0,00791 | -0,00062 | 0,00008 |
| 2,185 | 5,46693 | -0,14059 | 0,00729 | -0,00054 | 0,00004 |
| 2,190 | 5,32634 | -0,1333 | 0,00675 | -0,0005 | 0,00006 |
| 2,195 | 5,19304 | -0,12655 | 0,00625 | -0,00044 | 0,00003 |
| 2,200 | 5,06649 | -0,1203 | 0,00581 | -0,00041 | 0,00004 |
| 2,205 | 4,94619 | -0,11449 | 0,0054 | -0,00037 | 0,00004 |
| 2,210 | 4,83170 | -0,10909 | 0,00503 | -0,00033 | 0,00007 |
| 2,215 | 4,72261 | -0,10406 | 0,0047 | -0,00026 | - |
| 2,220 | 4,61855 | -0,09936 | 0,00444 | - | - |
| 2,225 | 4,51919 | -0,09492 | - | - | - |
| 2,230 | 4,42427 | - | - | - | - |

Визначимо крок таблиці за формулою 1.2:

$$h = x_1 - x_0 = 2{,}185 - 2{,}180 = 0.005.$$

Маючи крок таблиці, можемо обрахувати значення змінної q, яка є числом кроків, які потрібно пройти, щоб досягнути точки $x_1$, рухаючись від точки $x = 2.179$ будемо вважати $x_0 = 2.180$ за формулою 1.3:

$$q = \frac{x - x_0}{h} = \frac{2{,}179 - 2{,}180}{0.005} = 0{,}2.$$

Застосовуючи першу інтерполяційну формулу Ньютона (1.4), та підставляючи значення, що ми вже знайшли, обчислимо:

$$P_5(2{,}179) = 5{,}61543 + 0{,}2*(-0{,}1485) + \frac{0{,}2*(-0{,}8)}{2}*0{,}00791 + \frac{0{,}2*(-0{,}8)(-1{,}8)}{6}*$$
$$*(-0{,}00062) + \frac{0{,}2*(-0{,}8)(-1{,}8)(-2{,}8)}{24}*0{,}00008 = 5{,}61543 - 0{,}0297 - 0{,}08*0{,}00791 -$$
$$-0{,}048*0{,}00062 - 0{,}0336*0{,}00008 = 5{,}58573 - 0{,}0006328 - 0{,}00002976 -$$
$$-0{,}000002688 = 5{,}585064752$$

А тепер обчислимо за тією ж формулою значення змінної q (1.3) для $x = 2.183$, де $x_0 = 2.180$, крок таблиці такий ж самий:

$$q = \frac{x - x_0}{h} = \frac{2{,}183 - 2{,}180}{0.005} = 0{,}6.$$

Обчислюємо, підставляючи всі величини в формулу (1.4):

$$P_5(2.183) = 5.61543 + 0.6*(-0.1485) + \frac{0.6*(-0.4)}{2}*0.00791 + \frac{0.6*(-0.4)*(-1.4)}{6}*$$
$$*(-0.00062) + \frac{0.6*(-0.4)*(-1.4)*(-2.4)}{24}*0.00008 = 5.61543 - 0.0891 - 0.12*$$
$$*0.00791 - 0.056*0.00062 - 0.0336*0.00008 = 5.52633 - 0.0009492 - 0.00003472 -$$
$$-0.000002688 = 5.525343392.$$

Тепер обчислимо значення змінної q для $x = 2.227,$ для якого $x_0 = 2.225$, $h = 0.005$, підставляючи у формулу (1.6), тоді:

$$q = \frac{2.227 - 2.225}{0.005} = 0.4.$$

Підставляємо ці значення уже в другу інтерполяційну формулу Ньютона (1.7), отримаємо

$$P_5(2.227) = 4.51919 + 0.4*(-0.09936) + \frac{0.4*1.4}{2}*0.0047 + \frac{0.4*1.4*2.4}{6}*(-0.00033) +$$
$$+\frac{0.4*1.4*2.4*3.4}{24}*(-0.00004) = 4.51919 - 0.039744 + 0.28*0.0047 - 0.224*0.00033 -$$
$$-0.1904*0.00004 = 4.479446 + 0.001316 - 0.00007392 - 0.000007616 = 4.480680464$$

Теж саме ми повторюємо з $x = 2.232$, для якого $x_n = 2.230$, $h = 0.005$ підставивши у формулу (1.6):

$$q = \frac{2.232 - 2.230}{0.005} = 0.4.$$

Обчислюємо підставляючи ці всі значення у формулу (1.7):

$$P_5(2.232) = 4.42427 + 0.4*(-0.09492) + \frac{0.4*1.4}{2}*0.00444 + \frac{0.4*1.4*2.4}{6}*(-0.00026) +$$
$$+ \frac{0.4*1.4*2.4*3.4}{24}*0.00007 = 4.42427 - 0.037968 + 0.28*0.00444 - 0.224*0.00026 +$$
$$+ 0.1904*0.00007 = 4.386302 + 0.0012432 - 0.00005824 + 0.000013328 = 4.387500288.$$

## 2.2 Методика інтерполяції функції методом Лагранжа

Приклад 2. Побудуйте інтерполяційний многочлен Лагранжа другого степення для функції заданої таблично (таблиця 2.4), та знайдіть наближене значення функції в точці $x = 2.5$.

Таблиця 2.4 – Задана функція

| n | x | y |
|---|---|---|
| 0 | 1 | 15 |
| 1 | 2 | 17 |
| 2 | 3 | 7 |

Використаємо формулу (1.9), отриману нами раніше:

$$L_3(x) = 15*\frac{(x-2)(x-3)}{(1-2)(1-3)} + 17*\frac{(x-1)(x-3)}{(2-1)(2-3)} + 7*\frac{(x-1)(x-2)}{(3-1)(3-2)} = 15*\frac{x^2 - 3x - 2x + 6}{2} +$$
$$+ 17*\frac{x^2 - 3x - x + 3}{-1} + 7*\frac{x^2 - 2x - x + 2}{2} = \frac{15x^2 - 75x + 90}{2} - 17x^2 + 68x - 51 + \frac{7x^2 - 21x + 14}{2} =$$
$$= -17x^2 + 68x - 51 + \frac{22x^2 - 96x + 104}{2} = -17x^2 + 68x - 51 + \frac{2(11x^2 - 48x + 52)}{2} = -17x^2 +$$
$$+ 68x - 51 + 11x^2 - 48x + 52 = -6x^2 + 20x + 1.$$

Тепер, підставивши значення аргумента $x = 2.5,$ знайдемо наближене значення функції:

$$L_3(2,5) = -6*(2,5)^2 + 51 = -37,5 + 51 = 13,5.$$

# РОЗДІЛ 3
# РОЗРОБКА ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ ІНТЕРПОЛЯЦІЇ ТАБЛИЧНО ЗАДАНИХ ФУНКЦІЙ ЗАСОБАМИ MATLAB

## 3.1 Будова програмного комплексу

Розроблений у кваліфікаційній роботі програмно-демонстраційний комплекс складається з програм:

— інтерполяції таблично заданої функції на ділянці її монотонного зростання за допомогою формул Ньютона Newton_Formula;

— інтерполяції таблично заданої функції на ділянці її монотонного спадання за допомогою формул Ньютона Newton_Formula_1;

— інтерполяції таблично заданої функції на ділянці її перегину за допомогою формул Ньютона Newton_Formula_2;

— інтерполяції таблично заданої функції за допомогою полінома Лагранжа Lagrange_Polinom;

— інтерполяції таблично заданої функції за допомогою стандартної функції Nearest_Interp1;

— інтерполяції таблично заданої функції за допомогою стандартної функції Linear_Interp1;

— інтерполяції таблично заданої функції за допомогою стандартної функції Pchip_Interp1;

— інтерполяції таблично заданої функції за допомогою стандартної функції Cube_Spline;

— інтерполяції таблично заданої функції за допомогою стандартної функції Method_Interp1q;

## 3.2 Інтерполяція функції таблично заданої на ділянці її монотонно зростання за допомогою формул Ньютона

Розроблений у кваліфікаційній роботі програмно-демонстраційний комплекс складається з дев'яти окремих програм, кожна з яких вирішує задачу інтерполяції таблично заданої функції різними методами. У всіх цих програмах для досліджень вибрана одна і та ж функція сама функція y=sin(x), яка задана таблично на певному інтервалі числової осі. Добре відома елементарна функція вибрана для досліджень тому, що це дає змогу оцінити точність значень функції, які визначаються шляхом інтерполяції. Для цього достатньо порівняти значення функції, визначені за інтерполяційною програмою, з тими значеннями функції, які ми можемо легко обчислити безпосередньо за відомою формулою. А те, що у всіх програмах досліджується одна і та ж функція, дозволяє порівнювати ефективність різних програм у процесі вирішення задачі інтерполяції.

Звичайно, у якості досліджуваних функцій можна взяти і будь-яку іншу функцію. Більше того, можна взяти функцію, аналітичний вираз якої невідомий, але задана таблиця значень цієї функції, визначена, наприклад, експериментальним шляхом. Але тоді питання оцінки точності отриманих шляхом інтерполяції значень функції стає некоректним, оскільки невідома сама базова функція. Тобто, ми не можемо з певністю говорити, якими саме є точні значення функції.

Досліджувана функція може мати ділянки зростання, спадання або ж постійності значень. Варто з'ясувати, як працюють інтерполяційні методи на різних ділянках. З цією метою нами розроблені перші три програми комплексу. Програма Newton_Formula, код якої наведено на рис. 3.1, вирішує задачу інтерполяції таблично заданої на інтервалі монотонного зростання [0.2; 0.8] функції y=sin(x) за допомогою першої та другої інтерполяційних формул Ньютона. Програма побудована за технологією функціонального програмування. Це означає, що вона складається з головної функції Newton_Formula та низки підлеглих функцій.

```
function Newton_Formula
 % Функція y=sin(x) задана таблично у 7 (семи) вузлах інтерполяції:

 %  i   x(i)    y(i)
 %  1  0.200  0.19867
 %  2  0.300  0.29552
 %  3  0.400  0.38942
 %  4  0.500  0.47943
 %  5  0.600  0.56464
 %  6  0.700  0.64422
 %  7  0.800  0.71736

 % Необхідно:
 % Обчислити значення заданої функції у 4 (чотирьох) точках x0(i)
 % (i=1..4), що не співпадають з вузлами інтерполяції:

 %  i  x0(i)
 %  1  0.155
 %  2  0.255
 %  3  0.755
 %  4  0.855

 % Перші дві точки знаходяться ближче до початку таблиці.
 % Другі дві точки знаходяться ближче до кінця таблиці.
 % Для обчислень значень функції у перших двох точках
 % використовуємо першу інтерполяційну формулу Ньютона,
 % а для обчислень значень функції у других двох точках

 % використовуємо другу інтерполяційну формулу Ньютона,
 % Функція InitData() ініціалізує вхідні дані
 % і поертає їх значення у головну функцію Newton_Formula
 % x0 - вектор значень аргумента, для яких потрібно обчислити
значення функції
 % x - вектор значень вузлів інтерполяції
 % y - вектор значень функції у вузлах інтерполяції
 % n - кількість вузлів інтерполяції
 [x0,x,y,h,n]=InitData();

 % Функція будує графік y=sin(x) функції на заданому інтервалі
 FuncGraph(x,y);

 % Функція DifferenceTable(x,y,n) Формує значення табличних різниць
 [d1,d2,d3,d4,d5]=DifferenceTable(x,y,n);

 % Функція NewtonFormula(x0,x,y,d1,d2,d3,d4,d5,h,n)
 % обчислює значення функції у заданих точках
 % за першою або другою інтерполяційними формулами Ньютона
 % і повертає їх значення у головну функцію
 [p0,pn,y0]=NewtonFormula(x0,x,y,d1,d2,d3,d4,d5,h,n);

 % Функція CompareValues(x0,y0) прівнянює значеня функції,
 % знайдені за інтерполяційними формулами
 % з точними значеннями функції, обчисленими зо формулою y= sinx
 CompareValues(x0,y0);
end
```

```
% Ініціалізація вхідних даних
% x0 - вектор значень аргумента, для яких потрібно обчислити
значення функції
% x - вектор значень вузлів інтерполяції
% y - вектор значень функції у вузлах інтерполяції
% n - кількість вузлів інтерполяції
function [x0,x,y,h,n]=InitData()
 x0=[0.155; 0.255; 0.755;0.855];
 x(1)=0.2; % значення першого вузла інтерполяції
 x(2)=0.3; % значення другого вузла інтерполяції
 h=x(2)-x(1); % крок інтерполяційної таблиці
 n=7; % кількість вузлів інтерполяції
 for i=1:n % цикл формування значень вузлів інтерполяції (вектора x)
   x(i)=x(1)+(i-1)*h;
 end;
 x=x';
 y=[0.19867; 0.29552; 0.38942; 0.47943;0.56464; 0.64422; 0.71736];
end

% Побудова графіка функції
function FuncGraph(x,y);
 figure (1); % Створення вікна з номером 1
 bar(x,y);  grid on; % Побудова графіка функції y = f(x)
 title('y = sin(x)')
end

% Формування значень табличних різниць
function [d1,d2,d3,d4,d5]=DifferenceTable(x,y,n)
 % Цикл формування значень табличних різниць першого порядку
(вектора d1)
 for i=1:n-1
  d1(i)=y(i+1)-y(i);
 end
 d1=d1';
 % Цикл формування значень табличних різниць другого порядку
(вектора d2)
 for i=1:n-2
  d2(i)=d1(i+1)-d1(i);
 end
 d2=d2';
 % Цикл формування значень табличних різниць третього порядку
(вектора d3)
 for i=1:n-3
  d3(i)=d2(i+1)-d2(i);
 end
 d3=d3';
 % Цикл формування значень табличних різниць четвертого порядку
(вектора d4)
 for i=1:n-4
  d4(i)=d3(i+1)-d3(i);
 end
 d3=d3';
 % Цикл формування значень табличних різниць п'ятого порядку
```

Рисунок 3.1 – Програма інтерполяції функції y=sin(x) на інтервалі монотонного зростання значень [0.2; 0.8] за допомогою формул Ньютона (початок)

```
(вектора d5)
 for i=1:n-5
  d5(i)=d4(i+1)-d4(i);
 end
 d5=d5';
 % Виведення значень функції та значень табличних різниць
 fprintf('\nЗначення аргумента, функції та табличних різниць у
вузлах інтерполяції:');
 fprintf('\n  i   x(i)    y(i)     d1(i)     d2(i)     d3(i)
d4(i)     d5(i)');
 for i=1:n-5
  fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f
%8.5f',i,x(i), y(i), d1(i), d2(i), d3(i), d4(i), d5(i));
 end
 i=n-4;
 fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f',i,x(i),
y(i), d1(i), d2(i), d3(i), d4(i));
 i=n-3;
 fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f',i,x(i),
y(i), d1(i), d2(i), d3(i));
 i=n-2;
 fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f',i,x(i),
y(i), d1(i), d2(i));
 i=n-1;
 fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f',i,x(i),
y(i), d1(i));

 i=n;
 fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f',i,x(i),
y(i));
 fprintf('\n');
end

% Обчислення значень функції у заданих точках
% за інтерполяційними формулами Ньютона
function [p0,pn,y0]=NewtonFormula(x0,x,y,d1,d2,d3,d4,d5,h,n)
 fprintf('\nЗначення функції у заданих точках, обчислені');
 fprintf('\nза першою (p0) або другою (pn) інтерполяційними
формулами Ньютона:');
 for i=1:4
  if (x0(i)<x(1)) | (x0(i)-x(1)<=x(n)-x0(i))
   q=(x0(i)-x(1))/h;
   q1=q-1;
   q2=q-2;
   q3=q-3;
   q4=q-4;

p0=y(1)+q*d1(1)+q*q1*d2(1)/2+q*q1*q2*d3(1)/6+q*q1*q2*q3*d4(1)/24+q*q
1*q2*q3*q4*d5(1)/120;
   y0(i)=p0;
   fprintf('\n x0=%6.3f p0=%8.5f',x0(i), p0);
  end
  if (x0(i)>x(n)) | (x0(i)-x(1)>x(n)-x0(i))
   q=(x0(i)-x(n))/h;
```

Рисунок 3.1 – Програма інтерполяції функції y=sin(x) на інтервалі монотонного зростання значень [0.2; 0.8] за допомогою формул Ньютона (продовження)

```
    q1=q-1;
    q2=q-2;
    q3=q-3;
    q4=q-4;

p0=y(1)+q*d1(1)+q*q1*d2(1)/2+q*q1*q2*d3(1)/6+q*q1*q2*q3*d4(1)/24+q*q
1*q2*q3*q4*d5(1)/120;
    y0(i)=p0;
    fprintf('\n x0=%6.3f p0=%8.5f',x0(i), p0);
   end
   if (x0(i)>x(n)) | (x0(i)-x(1)>x(n)-x0(i))
    q=(x0(i)-x(n))/h;
    q1=q+1;
    q2=q+2;
    q3=q+3;
    q4=q+4;
    pn=y(n)+q*d1(n-1)+q*q1*d2(n-2)/2+q*q1*q2*d3(n-
4)/6+q*q1*q2*q3*d4(n-5)/24+q*q1*q2*q3*q4*d5(n-6)/120;
    y0(i)=pn;
    fprintf('\n x0=%6.3f pn=%8.5f',x0(i), pn);
   end
  end
  fprintf('\n');
end

% Порівняння значень, знайдених за інтерполяційними формулами

% з точними значеннями функції, обчисленими зо формулою y= sinx
function CompareValues(x0,y0)
 % Цикл формування точних значень функції,обчислених зо формулою y=
sinx
 % та порівняння їх зі значеннями, обчисленими за інтерполяційнми
формулами
 fprintf('\n Інтерпольовані та точні значення функції у вузлах
інтерполяції та їх різниці:');
 fprintf('\n  i   x0(i)   y0(i)    yy(i)    ry(i)');
 for i=1:4
  yy(i)=sin(x0(i));
  ry(i)= abs(yy(i)-y0(i));
  fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f', i, x0(i),
y0(i), yy(i), ry(i) );
 end
  fprintf('\n');
end
```

Рисунок 3.1 – Програма інтерполяції функції y=sin(x) на інтервалі монотонного зростання значень [0.2; 0.8] за допомогою формул Ньютона (закінчення)

Головна функція по черзі викликає підлеглі функції, які реалізують частину операцій загального алгоритму розв’язку задачі. Кожна з підлеглих функцій може використовувати у якості своїх параметрів дані, які доступні в головній функції. Крім того, підлеглі функції самі можуть передавати у головну функцію певні дані, які були у цій підлеглій функції введені або обчислені.

Першою підлеглою функцією є функція [x0,x,y,h,n]=InitData. Як видно з формату цієї функції, вона не використовує жодних значень з головної функції, а передає в головну функцію дані, зазначені у квадратних дужках.

До таких даних відносяться:

— вектор значень аргументів **xo**, для яких потрібно знайти значення функції за першою чи другою інтерполяційними формулами Ньютона;

— вектор значень аргументів **x**, які є вузлами інтерполяції;

— вектор значень функції **y** у вузлах інтерполяції;

— крок інтерполяційної таблиці **h**, який визначається як різниця між двома сусідніми вузлами інтерполяції;

— кількість вузлів інтерполяції **n** у заданій інтерполяційній таблиці.

Друга підлегла функція FuncGraph(x,y) будує графік інтерполяційної функції (рис. 3.2), використовуючи для цього значення, задані в таблиці.

Підлегла функція [d1,d2,d3,d4,d5]=DifferenceTable(x,y,n) формує, виводить на екран та передає у головну функцію табличні різниці з першого по п’ятого порядку включно.

Наступна функція, яка фактично є основною функцією даної програми,

[p0,pn,y0]= NewtonFormula(x,y,d1,d2,d3,d4,d5,h,n), приймає з головної функції через формальні параметри x,y,d1,d2,d3,d4,d5,h,n усі раніше обчислені величини з єдиною метою – знайти значення (p0 чи pn) функції y=sin(x) відповідно за першою чи другою інтерполяційними формулами Ньютона, сформувати вектор усіх таких значень y0, і передати ці величини у головну функцію.

Результати тестування програми Newton_Formula.

```
Значення аргумента, функції та табличних різниць у вузлах інтерполяції:
    i    x(i)     y(i)      d1(i)      d2(i)      d3(i)      d4(i)      d5(i)
```

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0.200 | 0.19867 | 0.09685 | -0.00295 | -0.00094 | 0.00003 | 0.00005 |
| 2 | 0.300 | 0.29552 | 0.09390 | -0.00389 | -0.00091 | 0.00008 | -0.00006 |
| 3 | 0.400 | 0.38942 | 0.09001 | -0.00480 | -0.00083 | 0.00002 | |
| 4 | 0.500 | 0.47943 | 0.08521 | -0.00563 | -0.00081 | | |
| 5 | 0.600 | 0.56464 | 0.07958 | -0.00644 | | | |
| 6 | 0.700 | 0.64422 | 0.07314 | | | | |
| 7 | 0.800 | 0.71736 | | | | | |

Значення функції у заданих точках, обчислені за першою (p0) або другою (pn) інтерполяційними формулами Ньютона:

x0= 0.155 p0= 0.15437
x0= 0.255 p0= 0.25225
x0= 0.755 pn= 0.68529
x0= 0.855 pn= 0.75458

Інтерпольовані та точні значення функції у вузлах інтерполяції та їх різниці:

| i | x0(i) | y0(i) | yy(i) | ry(i) |
|---|---|---|---|---|
| 1 | 0.155 | 0.15437 | 0.15438 | 0.00001 |
| 2 | 0.255 | 0.25225 | 0.25225 | 0.00000 |
| 3 | 0.755 | 0.68529 | 0.68529 | 0.00000 |
| 4 | 0.855 | 0.75458 | 0.75457 | 0.00001 |

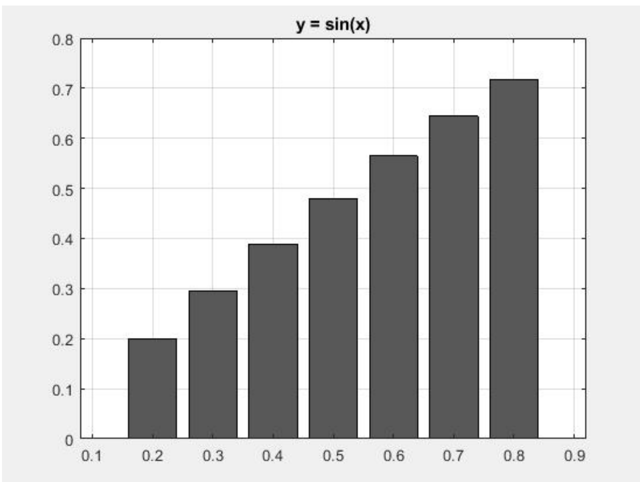


Рисунок 3.2 – Графік функції y=sin(x) на інтервалі монотонного зростання значень [0.2; 0.8]

## 3.3 Інтерполяція функції таблично заданої на ділянці її монотонно спадання за допомогою формул Ньютона

Код програми Newton_Formula_1 розробленої за тим же принципом, що й попередня програма, та результати її тестування наведені на рис. 3.3.

```
function Newton_Formula_1

% Функція y=sin(x) задана таблично у 7 (семи) вузлах інтерполяції:

 %  i   x(i)    y(i)
 %  1  2.500  0.59847
 %  2  2.600  0.51550
 %  3  2.700  0.42738
 %  4  2.800  0.33499
 %  5  2.900  0.23925
 %  6  3.000  0.14112
 %  7  3.100  0.04158

 % Необхідно:
 % Обчислити значення заданої функції у 4 (чотирьох) точках x0(i)
 % (i=1..4), що не співпадають з вузлами інтерполяції:

 %  i  x0(i)
 %  1  2.455
 %  2  2.555
 %  3  3.055
 %  4  3.155

 % Перші дві точки знаходяться ближче до початку таблиці.
 % Другі дві точки знаходяться ближче до кінця таблиці.
 % Для обчислень значень функції у перших двох точках
 % використовуємо першу інтерполяційну формулу Ньютона,
 % а для обчислень значень функції у других двох точках
```

Рисунок 3.3 – Програма інтерполяції функції y=sin(x) на інтервалі монотонного спадання значень [2.5; 3.1] за допомогою формул Ньютона (початок)

```
 % використовуємо другу інтерполяційну формулу Ньютона,
 % функція InitData() ініціалізує вхідні дані
 % і поертає їх значення у головну функцію Newton_Formula
 % x0 - вектор значень аргумента, для яких потрібно обчислити
значення функції
 % x - вектор значень вузлів інтерполяції
 % y - вектор значень функції у вузлах інтерполяції
 % n - кількість вузлів інтерполяції
 [x0,x,y,h,n]=InitData();

 % Функція будує графік y=sin(x) функції на заданому інтервалі
 FuncGraph(x,y);

 % Функція DifferenceTable(x,y,n) Формує значення табличних різниць
 [d1,d2,d3,d4,d5]=DifferenceTable(x,y,n);

 % Функція NewtonFormula(x0,x,y,d1,d2,d3,d4,d5,h,n)
 % обчислює значення функції у заданих точках
 % за першою або другою інтерполяційними формулами Ньютона
 % і повертає їх значення у головну функцію
 [p0,pn,y0]=NewtonFormula(x0,x,y,d1,d2,d3,d4,d5,h,n);

 % Функція CompareValues(x0,y0) прівнянює значеня функції,
 % знайдені за інтерполяційними формулами
 % з точними значеннями функції, обчисленими зо формулою y= sinx
 CompareValues(x0,y0);
end

% Ініціалізація вхідних даних
% x0 - вектор значень аргумента, для яких потрібно обчислити
значення функції
% x - вектор значень вузлів інтерполяції
% y - вектор значень функції у вузлах інтерполяції
% n - кількість вузлів інтерполяції
function [x0,x,y,h,n]=InitData()
 x0=[2.455; 2.555; 3.055; 3.155;];
 x(1)=2.5; % значення першого вузла інтерполяції
 x(2)=2.6; % значення другого вузла інтерполяції
 h=x(2)-x(1); % крок інтерполяційної таблиці
 n=7; % кількість вузлів інтерполяції
 for i=1:n % цикл формування значень вузлів інтерполяції (вектора x)
   x(i)=x(1)+(i-1)*h;
 end;
 x=x';
 y=[0.59847; 0.51550; 0.42738; 0.33499;
    0.23925; 0.14112; 0.04158];
end

% Побудова графіка функції
function FuncGraph(x,y);
 figure (1); % Створення вікна з номером 1
 bar(x,y);  grid on; % Побудова графіка функції y = f(x)
 title('y = sin(x)')
end
```

Рисунок 3.3 – Програма інтерполяції функції y=sin(x) на інтервалі монотонного спадання значень [2.5; 3.1] за допомогою формул Ньютона (продовження)

```
% Формування значень табличних різниць
function [d1,d2,d3,d4,d5]=DifferenceTable(x,y,n)
 % Цикл формування значень табличних різниць першого порядку
(вектора d1)
 for i=1:n-1
  d1(i)=y(i+1)-y(i);
 end
 d1=d1';
 % Цикл формування значень табличних різниць другого порядку
(вектора d2)
 for i=1:n-2
  d2(i)=d1(i+1)-d1(i);
 end
 d2=d2';
 % Цикл формування значень табличних різниць третього порядку
(вектора d3)
 for i=1:n-3
  d3(i)=d2(i+1)-d2(i);
 end
 d3=d3';
 % Цикл формування значень табличних різниць четвертого порядку
(вектора d4)
 for i=1:n-4
  d4(i)=d3(i+1)-d3(i);
 end
 d3=d3';
 % Цикл формування значень табличних різниць п'ятого порядку

(вектора d5)
 for i=1:n-5
  d5(i)=d4(i+1)-d4(i);
 end
 d5=d5';
 % Виведення значень функції та значень табличних різниць
 fprintf('\nЗначення аргумента, функції та табличних різниць у
вузлах інтерполяції:');
 fprintf('\n  i   x(i)    y(i)     d1(i)     d2(i)     d3(i)
d4(i)     d5(i)');
 for i=1:n-5
  fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f
%8.5f',i,x(i), y(i), d1(i), d2(i), d3(i), d4(i), d5(i));
 end
 i=n-4;
 fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f',i,x(i),
y(i), d1(i), d2(i), d3(i), d4(i));
 i=n-3;
 fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f',i,x(i),
y(i), d1(i), d2(i), d3(i));
 i=n-2;
 fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f',i,x(i),
y(i), d1(i), d2(i));
 i=n-1;
 fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f',i,x(i),
y(i), d1(i));
```

Рисунок 3.3 – Програма інтерполяції функції y=sin(x) на інтервалі монотонного спадання значень [2.5; 3.1] за допомогою формул Ньютона (продовження)

```
 i=n;
 fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f',i,x(i),
y(i));
 fprintf('\n');
end

% Обчислення значень функції у заданих точках
% за інтерполяційними формулами Ньютона
function [p0,pn,y0]=NewtonFormula(x0,x,y,d1,d2,d3,d4,d5,h,n)
 fprintf('\nЗначення функції у заданих точках, обчислені');
 fprintf('\nза першою (p0) або другою (pn) інтерполяційними
формулами Ньютона:');
 for i=1:4
  if (x0(i)<x(1)) | (x0(i)-x(1)<=x(n)-x0(i))
   q=(x0(i)-x(1))/h;
   q1=q-1;
   q2=q-2;
   q3=q-3;
   q4=q-4;

p0=y(1)+q*d1(1)+q*q1*d2(1)/2+q*q1*q2*d3(1)/6+q*q1*q2*q3*d4(1)/24+q*q
1*q2*q3*q4*d5(1)/120;
   y0(i)=p0;
   fprintf('\n x0=%6.3f p0=%8.5f',x0(i), p0);
  end
  if (x0(i)>x(n)) | (x0(i)-x(1)>x(n)-x0(i))
   q=(x0(i)-x(n))/h;

   q1=q-1;
   q2=q-2;
   q3=q-3;
   q4=q-4;

p0=y(1)+q*d1(1)+q*q1*d2(1)/2+q*q1*q2*d3(1)/6+q*q1*q2*q3*d4(1)/24+q*q
1*q2*q3*q4*d5(1)/120;
   y0(i)=p0;
   fprintf('\n x0=%6.3f p0=%8.5f',x0(i), p0);
  end
  if (x0(i)>x(n)) | (x0(i)-x(1)>x(n)-x0(i))
   q=(x0(i)-x(n))/h;
   q1=q+1;
   q2=q+2;
   q3=q+3;
   q4=q+4;
   pn=y(n)+q*d1(n-1)+q*q1*d2(n-2)/2+q*q1*q2*d3(n-
4)/6+q*q1*q2*q3*d4(n-5)/24+q*q1*q2*q3*q4*d5(n-6)/120;
   y0(i)=pn;
   fprintf('\n x0=%6.3f pn=%8.5f',x0(i), pn);
  end
 end
 fprintf('\n');
end

% Порівняння значень, знайдених за інтерполяційними формулами
```

Рисунок 3.3 – Програма інтерполяції функції y=sin(x) на інтервалі монотонного спадання значень [2.5; 3.1] за допомогою формул Ньютона (продовження)

```
% з точними значеннями функції, обчисленими зо формулою y= sinx
function CompareValues(x0,y0)
 % Цикл формування точних значень функції,обчислених зо формулою y=
sinx
 % та порівняння їх зі значеннями, обчисленими за інтерполяційнми
формулами
 fprintf('\n Інтерпольовані та точні значення функції у вузлах
інтерполяції та їх різниці:');
 fprintf('\n  i   x0(i)   y0(i)    yy(i)    ry(i)');
 for i=1:4
  yy(i)=sin(x0(i));
  ry(i)= abs(yy(i)-y0(i));
  fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f', i, x0(i),
y0(i), yy(i), ry(i) );
 end
  fprintf('\n');
end
```

Рисунок 3.3 – Програма інтерполяції функції y=sin(x) на інтервалі монотонного спадання значень [2.5; 3.1] за допомогою формул Ньютона (закінчення)

Результати тестування програми Newton_Formula.

```
Значення аргумента, функції та табличних різниць у вузлах інтерполяції:
    i   x(i)    y(i)     d1(i)     d2(i)     d3(i)     d4(i)     d5(i)
    1  2.500  0.59847 -0.08297  -0.00515   0.00088   0.00004   0.00000
    2  2.600  0.51550 -0.08812  -0.00427   0.00092   0.00004  -0.00002
    3  2.700  0.42738 -0.09239  -0.00335   0.00096   0.00002
    4  2.800  0.33499 -0.09574  -0.00239   0.00098
    5  2.900  0.23925 -0.09813  -0.00141
    6  3.000  0.14112 -0.09954
    7  3.100  0.04158

  Значення функції у заданих точках, обчислені
  за першою (p0) або другою (pn) інтерполяційними формулами Ньютона:
   x0= 2.455 p0= 0.63390
   x0= 2.555 p0= 0.55352
   x0= 3.055 pn= 0.08648
   x0= 3.155 pn=-0.01341

Інтерпольовані та точні значення функції у вузлах інтерполяції та їх
різниця:
    i   x0(i)   y0(i)    yy(i)    ry(i)
    1  2.455  0.63390  0.63391   0.00000
```

```
2  2.555  0.55352  0.55353   0.00000
3  3.055  0.08648  0.08648   0.00000
4  3.155 -0.01341 -0.01341   0.00000
```

Графік відповідної інтерполяційної функції зображений на рис. 3.4.

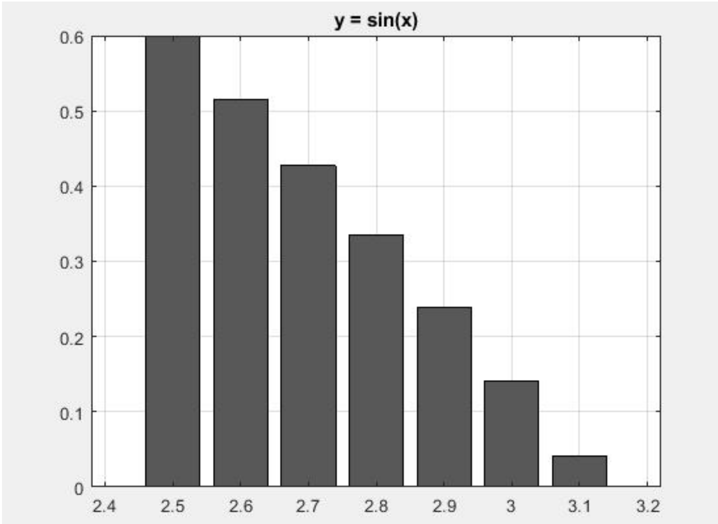


Рисунок 3.4 – Графік функції y=sin(x) на інтервалі монотонного спадання значень [2.5; 3.1]

## 3.4 Інтерполяція таблично заданої функції на ділянці перегину її графіка за допомогою формул Ньютона

Наводимо код програми Newton_Formula_2 (рис. 3.5). Як і дві попередні програми комплексу, програма Newton_Formula_2 розроблена за технологією функціонального програмування, складається з головної та п’яти підлеглих функцій.

```
function Newton_Formula_2

% Функція y=sin(x) задана таблично у 7 (семи) вузлах інтерполяції:

 %  i  x(i)  y(i)
 %  1  1.3  0.96356
 %  2  1.4  0.98545
 %  3  1.5  0.99749
 %  4  1.6  0.99957
 %  5  1.7  0.99166
 %  6  1.8  0.97385
 %  7  1.9  0.94630

 % Необхідно:
 % Обчислити значення заданої функції у 4 (чотирьох) точках x0(i)
 % (i=1..4), що не співпадають з вузлами інтерполяції:

 %  i  x0(i)
 %  1  1.255
 %  2  1.355
 %  3  1.855
 %  4  1.955

 % Перші дві точки знаходяться ближче до початку таблиці.
 % Другі дві точки знаходяться ближче до кінця таблиці.
 % Для обчислень значень функції у перших двох точках
 % використовуємо першу інтерполяційну формулу Ньютона,
 % а для обчислень значень функції у других двох точках
 % використовуємо другу інтерполяційну формулу Ньютона,
 % Функція InitData() ініціалізує вхідні дані
 % і поертає їх значення у головну функцію Newton_Formula
 % x0 - вектор значень аргумента, для яких потрібно обчислити
значення функції
 % x - вектор значень вузлів інтерполяції
 % y - вектор значень функції у вузлах інтерполяції
 % n - кількість вузлів інтерполяції
 [x0,x,y,h,n]=InitData();

 % Функція будує графік y=sin(x) функції на заданому інтервалі
 FuncGraph(x,y);

 % Функція DifferenceTable(x,y,n) Формує значення табличних різниць
 [d1,d2,d3,d4,d5]=DifferenceTable(x,y,n);

 % Функція NewtonFormula(x0,x,y,d1,d2,d3,d4,d5,h,n)
 % обчислює значення функції у заданих точках
 % за першою або другою інтерполяційними формулами Ньютона
 % і повертає їх значення у головну функцію
 [p0,pn,y0]=NewtonFormula(x0,x,y,d1,d2,d3,d4,d5,h,n);

 % Функція CompareValues(x0,y0) прівнянює значеня функції,
 % знайдені за інтерполяційними формулами
 % з точними значеннями функції, обчисленими зо формулою y= sinx
 CompareValues(x0,y0);
end
```

Рисунок 3.5 – Програма інтерполяції функції y=sin(x) на інтервалі перегину [1.3; 1.9] за допомогою формул Ньютона (початок)

```
% Ініціалізація вхідних даних
% x0 - вектор значень аргумента, для яких потрібно обчислити
значення функції
% x - вектор значень вузлів інтерполяції
% y - вектор значень функції у вузлах інтерполяції
% n - кількість вузлів інтерполяції
function [x0,x,y,h,n]=InitData()
 x0=[1.255; 1.355; 1.855; 1.955;];
 x(1)=1.3; % значення першого вузла інтерполяції
 x(2)=1.4; % значення другого вузла інтерполяції
 h=x(2)-x(1); % крок інтерполяційної таблиці
 n=7; % кількість вузлів інтерполяції
 for i=1:n % цикл формування значень вузлів інтерполяції (вектора x)
   x(i)=x(1)+(i-1)*h;
 end;
 x=x';
 y=[0.96356; 0.98545; 0.99749; 0.99957;
    0.99166; 0.97385; 0.94630];
end

% Побудова графіка функції
function FuncGraph(x,y);
 figure (1); % Створення вікна з номером 1
 plot(x,y,'r -*');  grid on; % Побудова графіка функції y=sin(x)
 title('y = sin(x)')
end

% Формування значень табличних різниць
function [d1,d2,d3,d4,d5]=DifferenceTable(x,y,n)
 % Цикл формування значень табличних різниць першого порядку
(вектора d1)
 for i=1:n-1
  d1(i)=y(i+1)-y(i);
 end
 d1=d1';
 % Цикл формування значень табличних різниць другого порядку
(вектора d2)
 for i=1:n-2
  d2(i)=d1(i+1)-d1(i);
 end
 d2=d2';
 % Цикл формування значень табличних різниць третього порядку
(вектора d3)
 for i=1:n-3
  d3(i)=d2(i+1)-d2(i);
 end
 d3=d3';
 % Цикл формування значень табличних різниць четвертого порядку
(вектора d4)
 for i=1:n-4
  d4(i)=d3(i+1)-d3(i);
 end
 d3=d3';
 % Цикл формування значень табличних різниць п'ятого порядку
```

Рисунок 3.5 – Програма інтерполяції функції y=sin(x) на інтервалі перегину [1.3; 1.9] за допомогою формул Ньютона (продовження)

```
(вектора d5)
 for i=1:n-5
  d5(i)=d4(i+1)-d4(i);
 end
 d5=d5';
 % Виведення значень функції та значень табличних різниць
 fprintf('\nЗначення аргумента, функції та табличних різниць у
вузлах інтерполяції:');
 fprintf('\n  i   x(i)    y(i)     d1(i)     d2(i)     d3(i)
d4(i)     d5(i)');
 for i=1:n-5
  fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f
%8.5f',i,x(i), y(i), d1(i), d2(i), d3(i), d4(i), d5(i));
 end
 i=n-4;
 fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f',i,x(i),
y(i), d1(i), d2(i), d3(i), d4(i));
 i=n-3;
 fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f',i,x(i),
y(i), d1(i), d2(i), d3(i));
 i=n-2;
 fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f',i,x(i),
y(i), d1(i), d2(i));
 i=n-1;
 fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f',i,x(i),
y(i), d1(i));

 i=n;
 fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f',i,x(i),
y(i));
 fprintf('\n');
end

% Обчислення значень функції у заданих точках
% за інтерполяційними формулами Ньютона
function [p0,pn,y0]=NewtonFormula(x0,x,y,d1,d2,d3,d4,d5,h,n)
 fprintf('\nЗначення функції у заданих точках, обчислені');
 fprintf('\nза першою (p0) або другою (pn) інтерполяційними
формулами Ньютона:');
 for i=1:4
  if (x0(i)<x(1)) | (x0(i)-x(1)<=x(n)-x0(i))
   q=(x0(i)-x(1))/h;
   q1=q-1;
   q2=q-2;
   q3=q-3;
   q4=q-4;

p0=y(1)+q*d1(1)+q*q1*d2(1)/2+q*q1*q2*d3(1)/6+q*q1*q2*q3*d4(1)/24+q*q
1*q2*q3*q4*d5(1)/120;
   y0(i)=p0;
   fprintf('\n x0=%6.3f p0=%8.5f',x0(i), p0);
  end
  if (x0(i)>x(n)) | (x0(i)-x(1)>x(n)-x0(i))
   q=(x0(i)-x(n))/h;
```

Рисунок 3.5 – Програма інтерполяції функції y=sin(x) на інтервалі перегину [1.3; 1.9] за допомогою формул Ньютона (продовження)

```
    q1=q-1;
    q2=q-2;
    q3=q-3;
    q4=q-4;

p0=y(1)+q*d1(1)+q*q1*d2(1)/2+q*q1*q2*d3(1)/6+q*q1*q2*q3*d4(1)/24+q*q
1*q2*q3*q4*d5(1)/120;
    y0(i)=p0;
    fprintf('\n x0=%6.3f p0=%8.5f',x0(i), p0);
   end
   if (x0(i)>x(n)) | (x0(i)-x(1)>x(n)-x0(i))
    q=(x0(i)-x(n))/h;
    q1=q+1;
    q2=q+2;
    q3=q+3;
    q4=q+4;
    pn=y(n)+q*d1(n-1)+q*q1*d2(n-2)/2+q*q1*q2*d3(n-
4)/6+q*q1*q2*q3*d4(n-5)/24+q*q1*q2*q3*q4*d5(n-6)/120;
    y0(i)=pn;
    fprintf('\n x0=%6.3f pn=%8.5f',x0(i), pn);
   end
  end
  fprintf('\n');
end

% Порівняння значень, знайдених за інтерполяційними формулами

% з точними значеннями функції, обчисленими зо формулою y= sinx
function CompareValues(x0,y0)
 % Цикл формування точних значень функції,обчислених зо формулою y=
sinx
 % та порівняння їх зі значеннями, обчисленими за інтерполяційнми
формулами
 fprintf('\n Інтерпольовані та точні значення функції у вузлах
інтерполяції та їх різниці:');
 fprintf('\n  i   x0(i)   y0(i)    yy(i)    ry(i)');
 for i=1:4
  yy(i)=sin(x0(i));
  ry(i)= abs(yy(i)-y0(i));
  fprintf('\n %2d %6.3f %8.5f %8.5f  %8.5f  %8.5f  %8.5f', i, x0(i),
y0(i), yy(i), ry(i) );
 end
  fprintf('\n');
end
```

Рисунок 3.5 – Програма інтерполяції функції y=sin(x) на інтервалі перегину [1.3; 1.9] за допомогою формул Ньютона (закінчення)

Графік заданої функції та результати тестування програми (рис. 3.6).

Результати тестування програми Newton_Formula.

```
Значення аргумента, функції та табличних різниць у вузлах інтерполяції:
```

| i | x(i) | y(i) | d1(i) | d2(i) | d3(i) | d4(i) | d5(i) |
|---|---|---|---|---|---|---|---|
| 1 | 1.300 | 0.96356 | 0.02189 | -0.00985 | -0.00011 | 0.00008 | 0.00004 |
| 2 | 1.400 | 0.98545 | 0.01204 | -0.00996 | -0.00003 | 0.00012 | -0.00005 |
| 3 | 1.500 | 0.99749 | 0.00208 | -0.00999 | 0.00009 | 0.00007 | |
| 4 | 1.600 | 0.99957 | -0.00791 | -0.00990 | 0.00016 | | |
| 5 | 1.700 | 0.99166 | -0.01781 | -0.00974 | | | |
| 6 | 1.800 | 0.97385 | -0.02755 | | | | |
| 7 | 1.900 | 0.94630 | | | | | |

```
Значення функції у заданих точках, обчислені
за першою (p0) або другою (pn) інтерполяційними формулами Ньютона:
 x0= 1.255 p0= 0.95054
 x0= 1.355 p0= 0.97681
 x0= 1.855 pn= 0.95989
 x0= 1.955 pn= 0.92708
```

Інтерпольовані та точні значення функції у вузлах інтерполяції та їх різниця:

| i | x0(i) | y0(i) | yy(i) | ry(i) |
|---|---|---|---|---|
| 1 | 1.255 | 0.95053545 | 0.95054936 | 0.00001391 |
| 2 | 1.355 | 0.97680994 | 0.97680619 | 0.00000375 |
| 3 | 1.855 | 0.95989102 | 0.95988524 | 0.00000578 |
| 4 | 1.955 | 0.92707872 | 0.92709721 | 0.00001848 |

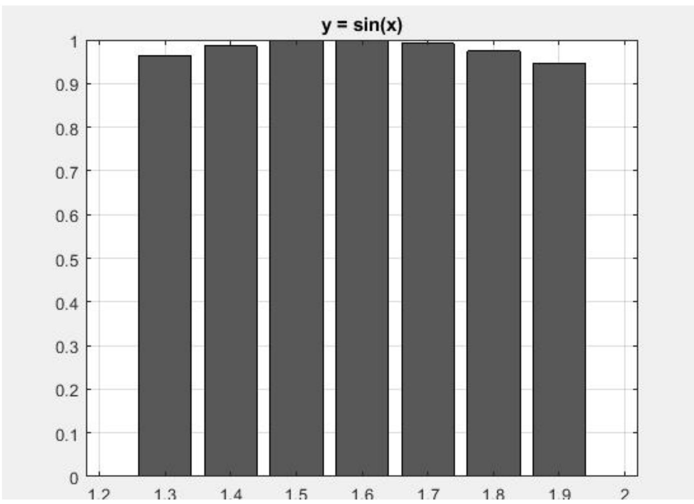


Рисунок 3.6 – Графік функції y=sin(x) на інтервалі перегину [1.3; 1.9]

### 3.5 Інтерполяція таблично заданої функції за допомогою полінома Лагранжа

Наводимо код і результати тестування програми Lagrange_Polinom (рис. 3.7). Як і дві попередні програми комплексу, програма Lagrange_Polinom розроблена за технологією функціонального програмування, складається з головної та чотирьох підлеглих функцій. Підпрограма [L] = Lagrange_Polinom(x0,x,y,n) обчислює та повертає в головну функцію значення полінома Лагранжа L для значень функції x0, використовуючи для цього параметри x0,x,y,n, про які йшлося раніше.

```
function Lagrange_Polinom
 % функція y=sin(x) задана таблично у 7 (семи) вузлах інтерполяції:

 %  i  x(i)  y(i)
 %  0  0.2  0.19867
 %  1  0.3  0.29552
 %  2  0.4  0.38942
 %  3  0.5  0.47943
 %  4  0.6  0,56464
 %  5  0.7  0.64422
 %  6  0.8  0.71736

 % Необхідно:
 % Обчислити значення цієї функції у 4 (чотирьох) точках x0(i):
 % що не співпадають з вузлами інтерполяції,
 % використовуючи поліном Лагранжа.

 %  i   x0(i)
 %  1  0.155
 %  2  0.255
 %  3  0.755
 %  4  0.855

 % Функція InitData(), яка ініціалізує вхідні дані
 % і передає їх у головну функцію.
 % x0 - вектор значень аргумента, для яких потрібно обчислити
значення функції
 % x - вектор значень вузлів інтерполяції
 % y - вектор значень функції у вузлах інтерполяції
 % n - кількість вузлів інтерполяції
 [x0,x,y,h,n]=InitData();
```

Рисунок 3.7 – Програма інтерполяції функції y=sin(x) на інтервалі [0.2; 0.8] за допомогою полінома Лагранжа (початок)

```
 % Побудова графіка функції
 FuncGraph(x,y);

 % Функція LagrangePolinom(x0,x,y,n) обчислює значення функції
 % у заданих точках за інтерполяційним поліномом Лагранжа
 % і передає їх у головну функцію.
 [L]=LagrangePolinom(x0,x,y,n);

 % Функція LagrangePolinom(x0,x,y,n) порівнює значення,
 % изнайдені за інтерполяційни поліномом Лагранжа
 % з точними значеннями функції, обчисленими зо формулою y= sinx
 CompareValues(x0,L);

end

% Ініціалізація вхідних даних
% x0 - вектор значень аргумента, для яких потрібно обчислити
значення функції
% x - вектор значень вузлів інтерполяції
% y - вектор значень функції у вузлах інтерполяції
% n - кількість вузлів інтерполяції
function [x0,x,y,h,n]=InitData()
 x0=[0.155; 0.255; 0.755; 0.855];
 x(1)=0.2; % значення першого вузла інтерполяції
 x(2)=0.3; % значення другого вузла інтерполяції

 h=x(2)-x(1); % крок інтерполяційної таблиці
 n=7; % кількість вузлів інтерполяції
 for i=1:n % цикл формування значень вузлів інтерполяції (вектора x)
   x(i)=x(1)+(i-1)*h;
 end;
 x=x';
 y=[ 0.19867; 0.29552; 0.38942; 0.47943; 0.56464; 0.64422; 0.71736];
 fprintf('\n Задана інтерполяційна таблиця:');
 fprintf('\n  i     x(i)        y(i)');
 for i=1:n
   fprintf('\n %2d   %6.5f    %8.5f',i,x(i),y(i));
 end;
 fprintf('\n');
 fprintf('\n Заданий вектор значень аргумента,');
 fprintf('\n для яких потрібно обчислити значення функції:');
 fprintf('\n  i     x(i)');
 for i=1:4
   fprintf('\n %2d   %6.5f',i,x0(i));
 end;
 fprintf('\n');
end

 % Побудова графіка функції
 function FuncGraph(x,y);
  figure (1); % Створення вікна з номером 1
  plot(x,y,'r -*');  grid on; % Побудова графіка функції y = f(x)
  title('y = sin(x)')
 end
```

Рисунок 3.7 – Програма інтерполяції функції y=sin(x) на інтервалі [0.2; 0.8] за допомогою полінома Лагранжа (продовження)

```
% Обчислення значень функції у заданих точках
% за інтерполяційни поліномом Лагранжа
function [L]=LagrangePolinom(x0,x,y,n)
 fprintf('\nЗначення функції у заданих точках, обчислені');
 fprintf('\nза інтерполяційни поліномом Лагранжа:');
 for k=1:4
  L(k)=0;
  for i=1:n
    a(i)=1;
    for j=1:n
      if not(i == j)
       a(i)=a(i)*(x0(k)-x(j))/(x(i)-x(j));
      end
    end
    L(k)=L(k)+a(i)*y(i);
  end
  fprintf('\n x0(%2d)=%6.5f L(%2d)=%8.5f',k,x0(k),k,L(k));
 end
  fprintf('\n');
end

% Порівняння значень, знайдені за інтерполяційним поліномом Лагранжа
% з точними значеннями функції, обчисленими зо формулою y= sinx
function CompareValues(x0,L)
 % Цикл формування точних значень функції,обчислених зо формулою y=
sinx
 % та порівняння їх зі значеннями, обчисленими за інтерполяційнми
формулами

 fprintf('\n Інтерпольовані та точні значення функції у вузлах
інтерполяції та їх різниця:');
 fprintf('\n  i   x0(i)   y0(i)     yy(i)      ry(i)');
 for i=1:4
  yy(i)=sin(x0(i));
  ry(i)= abs(yy(i)-L(i));
  fprintf('\n %2d %6.3f %8.5f %10.8f  %10.8f  %10.8f', i, x0(i),
L(i), yy(i), ry(i) );
 end
  fprintf('\n');
end
```

Рисунок 3.7 – Програма інтерполяції функції y=sin(x) на інтервалі [0.2; 0.8] за допомогою полінома Лагранжа (закінчення)

Результати тестування програми:

```
Задана інтерполяційна таблиця:
  i     x(i)        y(i)
  1   0.20000     0.19867
```

```
  2   0.30000     0.29552
  3   0.40000     0.38942
  4   0.50000     0.47943
  5   0.60000     0.56464
  6   0.70000     0.64422
  7   0.80000     0.71736
```

Заданий вектор значень аргумента,
для яких потрібно обчислити значення функції:

```
  i     x(i)
  1   0.15500
  2   0.25500
  3   0.75500
  4   0.85500
```

Значення функції у заданих точках, обчислені
за інтерполяційни поліномом Лагранжа:

```
 x0( 1)=0.15500 L( 1)= 0.15435
 x0( 2)=0.25500 L( 2)= 0.25225
 x0( 3)=0.75500 L( 3)= 0.68530
 x0( 4)=0.85500 L( 4)= 0.75451
```

Інтерпольовані та точні значення функції у вузлах інтерполяції та їх різниця:

| i | x0(i) | y0(i) | yy(i) | ry(i) |
|---|---|---|---|---|
| 1 | 0.155 | 0.15435 | 0.15438010 | 0.00002835 |
| 2 | 0.255 | 0.25225 | 0.25224541 | 0.00000316 |
| 3 | 0.755 | 0.68530 | 0.68528867 | 0.00001041 |
| 4 | 0.855 | 0.75451 | 0.75457092 | 0.00006334 |

Графік заданої функції (рис. 3.8)

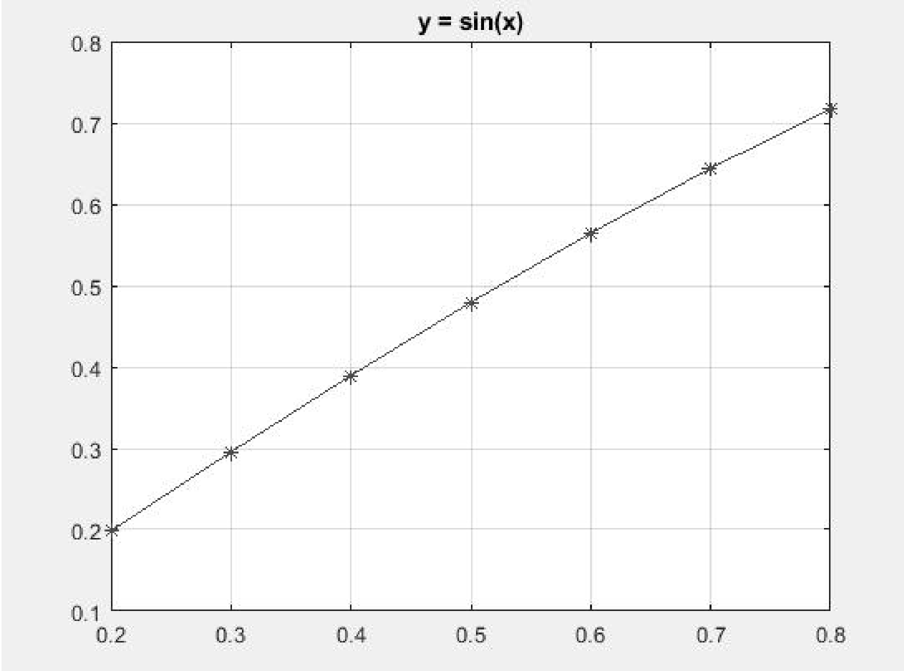


Рисунок 3.8 – Графік таблично заданої функції на інтервалі $x \in [0.2;\ 0.8]$.

## 3.6 Ступінчата інтерполяція функції y=sin(x) за допомогою стандартної функції Nearest_Interp1

Система комп'ютерної математики Matlab має у своєму розпорядженні широкий арсенал власних (стандартних) програм (або ж інакше - функцій) для обчислення значень таблично заданих математичних функцій шляхом інтерполяції. До таких програм відносяться:

— програма ступінчатої інтерполяції interp1() з параметром nearest;

— програма лінійної інтерполяції interp1() з параметром linear;

— програма інтерполяції многочленами Ерміта interp1() з параметром pchip;

— програма інтерполяції кубічними сплайнами spline();

— модифікована програма лінійної інтерполяції interp1q().

Розглянемо використання кожної з цих функцій детальніше. Код і результати тестування програми Nearest_Interp1, яка реалізує метод ступінчатої інтерполяції, наведені на рис. 3.9. Як бачимо з результатів тестування, ця програма дає досить неточні результати і може бути використана хіба що для грубої оцінки значення функції.

```
function Nearest_Interp1
 % Ступінчата інтерполяція таблично заданої функції y=sin(x).
 % Функція y=sin(x) задана таблично у 7 (семи) вузлах інтерполяції:

 %  i   x(i)    y(i)
 %  1  0.200  0.19867
 %  2  0.300  0.29552
 %  3  0.400  0.38942
 %  4  0.500  0.47943
 %  5  0.600  0.56464
 %  6  0.700  0.64422
 %  7  0.800  0.71736

 % Необхідно:
 % використовуючи стандартну функцію interp1(x,y,'nearest','pp')
 % обчислити значення заданої функції у 4 (чотирьох) точках x0(i)
 % (i=1..4), що не співпадають з вузлами інтерполяції:

 %  i  x0(i)
 %  1  0.155
 %  2  0.255
 %  3  0.755
 %  4  0.855

 x=0.2:0.1:0.8;
 y=sin(x);
 pp=interp1(x,y,'nearest','pp'); % формування вектора pp
 xx=0.2:0.05:0.8;
 yy=ppval(pp, xx); % формування вектора yy
 plot(x,y,'ko'); grid on;
 plot(xx,yy,'r-'); grid on;
 title('y=sin(x)');

 X=[0.155; 0.255; 0.755; 0.855];
 % Формуємо масив значень функції для заданих значень аргументів,
 % визначених шляхом ступінчатої інтерполяції):
 Y=ppval(pp,X); % ступінчата інтерполяція функції


 % Порівняння значень, знайдених шляхом ступінчатої інтерполяції
 % за допомогою стандартної функції interp1(x,y,'nearest','pp'),
 % з точними значеннями функції, обчисленими зо формулою y= sinx.
 % Цикл формування точних значень функції,обчислених зо формулою y=
sinx
 % та порівняння їх зі значеннями, обчисленими за допомогою
 % стандартної функції interp1(x,y,'nearest','pp'),
 fprintf('\n Інтерпольовані та точні значення функції у вузлах
інтерполяції та їх різниці:');
 fprintf('\n  i   X(i)   Y(i)     yt(i)      ry(i)');
 for i=1:4
  yt(i)=sin(X(i));
  ry(i)= abs(yy(i)-yt(i));
  fprintf('\n %2d %6.3f %8.5f %10.8f  %10.8f  %10.8f', i, X(i),
Y(i), yt(i), ry(i) );
 end
 fprintf('\n');
end
```

Рисунок 3.9 – Програма ступінчатої інтерполяції функції y=sin(x) за допомогою стандартної функції Nearest_Interp1

Результати тестування програми:

```
Інтерпольовані та точні значення функції у вузлах інтерполяції та
їх різниці:
    i   X(i)    Y(i)      yt(i)       ry(i)
    1  0.155  0.19867 0.15438010  0.04428923
    2  0.255  0.29552 0.25224541  0.04327480
    3  0.755  0.71736 0.68528867  0.38976846
    4  0.855  0.71736 0.75457092  0.36515257
```

Графік заданої функції зображенні на рис. 3.10.

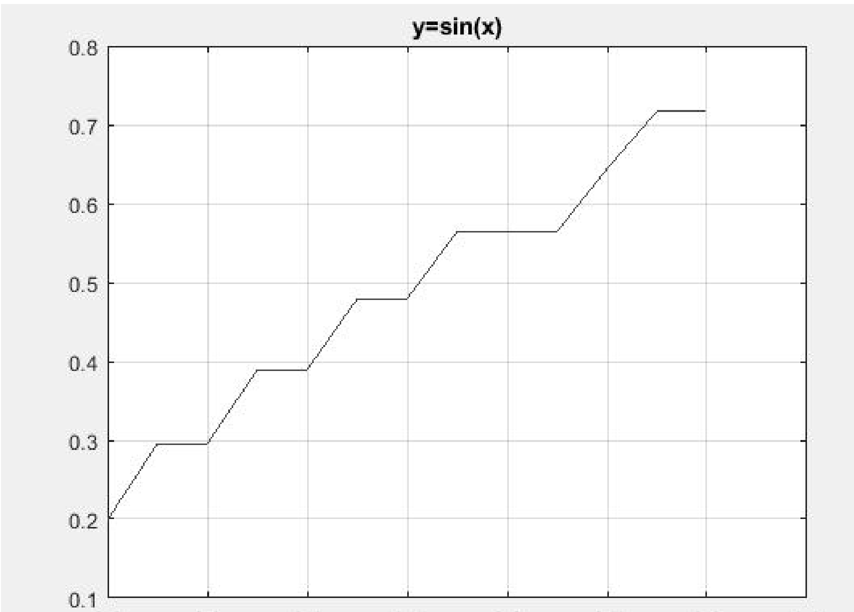


Рисунок 3.10 – Графічна інтерпретація ступінчатої інтерполяції таблично заданої функції на інтервалі $x \in [0.2;\ 0.8]$.

## 3.7 Лінійна інтерполяція функції y=sin(x) за допомогою стандартної функції Linear_Interp1

Код і результати тестування програми Linear_Interp1, яка реалізує метод лінійної інтерполяції, наведені на рис. 3.11. На відміну від попередньої програми, дана програма дає значно точніші результати. Однак не слід очікувати у даному разі дуже високої точності, оскільки ми замінюємо криволінійні ланки функції прямими лініями.

```
function Linear_Interp1
 % Лінійна інтерполяція таблично заданої функції y=sin(x).
 % Функція y=sin(x) задана таблично у 7 (семи) вузлах інтерполяції:
 %  i   x(i)    y(i)
 %  1  0.200  0.19867
 %  2  0.300  0.29552
 %  3  0.400  0.38942
 %  4  0.500  0.47943
 %  5  0.600  0.56464
 %  6  0.700  0.64422
 %  7  0.800  0.71736

 % Необхідно:
 % використовуючи стандартну функцію interp1(x,y,'linear','pp')
 % обчислити значення заданої функції у 4 (чотирьох) точках x0(i)
 % (i=1..4), що не співпадають з вузлами інтерполяції:

 %  i  x0(i)
 %  1  0.155
 %  2  0.255
 %  3  0.755
 %  4  0.855

 x=0.2:0.1:0.8;
 y=sin(x);
 pp=interp1(x,y,'linear','pp'); % формування вектора pp
 xx=0.2:0.05:0.8;
 yy=ppval(pp, xx); % формування вектора yy
 subplot (2,1,1); % Ділимо вікно на 2 рядки і один стовпець
 plot(x,y,'k*'); grid on;
```

Рисунок 3.11 – Програма лінійної інтерполяції функції y=sin(x) за допомогою стандартної функції Linear_Interp1(початок)

```
title('y=sin(x)');
subplot (2,1,2); % Ділимо вікно на 2 рядки і один стовпець
plot(xx,yy,'r-'); grid on;
title('y=sin(x)');
X=[0.155; 0.255; 0.755; 0.855];
% Формуємо масив значень функції для заданих значень аргументів,
% визначених шляхом ступінчатої інтерполяції):
Y=ppval(pp,X); % лінійна інтерполяція функції
% Порівняння значень, знайдених шляхом лінійної інтерполяції
% за допомогою стандартної функції interp1(x,y,'linear','pp'),
% з точними значеннями функції, обчисленими зо формулою y= sinx.
% Цикл формування точних значень функції,обчислених зо формулою y=
sinx
% та порівняння їх зі значеннями, обчисленими за допомогою
% стандартної функції interp1(x,y,'linear','pp').
fprintf('\n Інтерпольовані та точні значення функції у вузлах
інтерполяції та їх різниці:');
fprintf('\n  i   X(i)   Y(i)     yt(i)      ry(i)');
for i=1:4
 yt(i)=sin(X(i));
 ry(i)= abs(yy(i)-yt(i));
 fprintf('\n %2d %6.3f %8.5f %10.8f  %10.8f  %10.8f', i, X(i),
Y(i), yt(i), ry(i) );
end
fprintf('\n');
end
```

Рисунок 3.11 – Програма лінійної інтерполяції функції y=sin(x) за допомогою стандартної функції Linear_Interp1(закінчення).

Результати тестування програми:

```
Інтерпольовані та точні значення функції у вузлах інтерполяції та
їх різниці:
  i   X(i)   Y(i)     yt(i)      ry(i)
  1  0.155  0.15509 0.15438010  0.04428923
  2  0.255  0.25194 0.25224541  0.00515064
  3  0.755  0.68444 0.68528867  0.38976846
  4  0.855  0.75758 0.75457092  0.41210164
```

Графік заданої функції зображенні на рис. 3.12

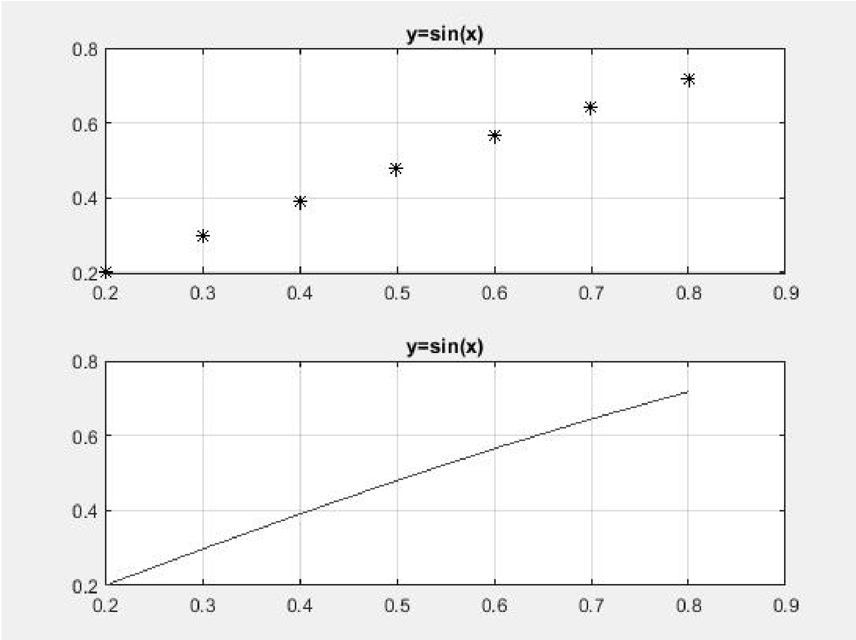


Рисунок 3.12 – Графічна інтерпретація лінійної інтерполяції таблично заданої функції на інтервалі $x \in [0.2;\ 0.8]$.

## 3.8 Інтерполяція функції y=sin(x) за допомогою стандартної функції Pchip_Interp1 многочленами Ерміта

Код та результати тестування програми Pchip_Interp1, яка реалізує інтерполяцію функцій многочленами Ерміта, наведені на рис. 3.13. Дана програма дала результати з високою точністю. Оскільки ми замінюємо криволінійні ланки функції іншими криволінійними ланками за допомогою многочленів Ерміта.

```
function Pchip_Interp1
 % Інтерполяція таблично заданої функції y=sin(x) многочленами
Ерміта.
 % Функція y=sin(x) задана таблично у 7 (семи) вузлах інтерполяції:
 %  i    x(i)     y(i)
 %  1  0.200  0.19867
 %  2  0.300  0.29552
 %  3  0.400  0.38942
 %  4  0.500  0.47943
 %  5  0.600  0.56464
 %  6  0.700  0.64422
 %  7  0.800  0.71736
 % Необхідно:
 % використовуючи стандартну функцію interp1(x,y,'pchip','pp')
 % обчислити значення заданої функції у 4 (чотирьох) точках x0(i)
 % (i=1..4), що не співпадають з вузлами інтерполяції:
 %  i  x0(i)
 %  1  0.155
 %  2  0.255
 %  3  0.755
 %  4  0.855

 x=0.2:0.1:0.8;
 y=sin(x);
 pp=interp1(x,y,'pchip','pp'); % формування вектора pp
 xx=0.2:0.05:0.8;
 yy=ppval(pp, xx); % формування вектора yy
 subplot (2,1,1); % Ділимо вікно на 2 рядки і один стовпець
 plot(x,y,'k*'); grid on;
 title('y=sin(x)');
 subplot (2,1,2); % Ділимо вікно на 2 рядки і один стовпець
 plot(xx,yy,'r-'); grid on;
 title('y=sin(x)');

 X=[0.155; 0.255; 0.755; 0.855];
 % Формуємо масив значень функції для заданих значень аргументів,
 % визначених шляхом інтерполяції)многочленами Ерміта:
 Y=ppval(pp,X); % інтерполяція функції многочленами Ерміта
 % Порівняння значень, знайдених шляхом інтерполяції многочленами
Ерміта
 % за допомогою стандартної функції interp1(x,y,'pchip','pp'),
 % з точними значеннями функції, обчисленими зо формулою y= sinx.
 % Цикл формування точних значень функції,обчислених зо формулою y=
sinx
 % та порівняння їх зі значеннями, обчисленими за допомогою
 % стандартної функції interp1(x,y,'pchip','pp').
 fprintf('\n Інтерпольовані та точні значення функції у вузлах
інтерполяції та їх різниці:');
 fprintf('\n  i   X(i)   Y(i)     yt(i)      ry(i)');
 for i=1:4
  yt(i)=sin(X(i));
  ry(i)= abs(yy(i)-yt(i));
  fprintf('\n %2d %6.3f %8.5f %10.8f  %10.8f  %10.8f', i, X(i),
Y(i), yt(i), ry(i) );
 end
 fprintf('\n');
end
```

Рисунок 3.13 – Програма інтерполяції функції y=sin(x) многочленами Ерміта за допомогою стандартної функції Pchip_Interp1

Результати тестування програми:

```
Інтерпольовані та точні значення функції у вузлах інтерполяції та
їх різниці:

    i   X(i)    Y(i)      yt(i)       ry(i)
    1  0.155  0.15413 0.15438010  0.04428923
    2  0.255  0.25231 0.25224541  0.00477869
    3  0.755  0.68523 0.68528867  0.38976846
    4  0.855  0.75477 0.75457092  0.41167162
```

Графік заданої функції на рис. 3.14.

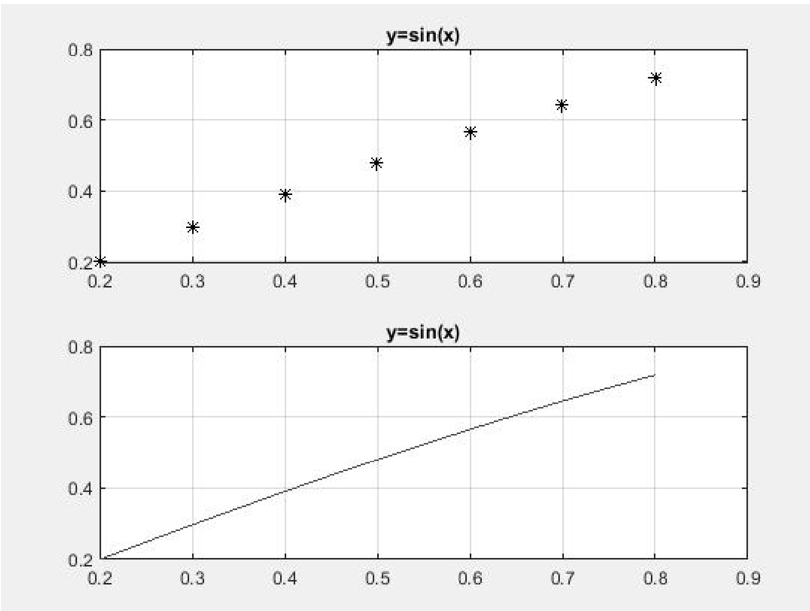


Рисунок 3.14 – Графічна інтерпретація інтерполяції таблично заданої функції на інтервалі $x \in [0.2;\ 0.8]$ многочленами Ерміта.

### 3.9 Інтерполяція функції y=sin(x) за допомогою стандартної функції Spline кубічними сплайнами

Код і результати тестування програми Cube_Spline, яка реалізує інтерполяцію функцій кубічними сплайнами, наведені на рис. 3.15. Дана програма забезпечує отримання результатів з високою точністю. оскільки ми замінюємо криволінійні ланки функції кубічними параболами за допомогою кубічних сплайнів.

```
function Cube_Spline()
 % Інтерполяція кубічними сплайнами заданої таблично функції
y=sin(x).
 % Функція y=sin(x) задана таблично у 7 (семи) вузлах інтерполяції:

 %  i   x(i)    y(i)
 %  1  0.200  0.19867
 %  2  0.300  0.29552
 %  3  0.400  0.38942
 %  4  0.500  0.47943
 %  5  0.600  0.56464
 %  6  0.700  0.64422
 %  7  0.800  0.71736

 % Необхідно:
 % використовуючи стандартну функцію spline()
 % обчислити значення заданої функції у 4 (чотирьох) точках x0(i)
 % (i=1..4), що не співпадають з вузлами інтерполяції:

 %  i  x0(i)
 %  1  0.155
 %  2  0.255
 %  3  0.755
 %  4  0.855

 clear;
```

Рисунок 3.15 – Програма інтерполяції функції y=sin(x) кубічними сплайнами за допомогою стандартної функції Spline (початок)

```
clear;
x=0.2:0.1:0.8; % формування вектора абсцис
% Формування вектора ординат
y=[0.19867; 0.29552; 0.38942; 0.47943; 0.56464; 0.64422; 0.71736];
yy=spline(x,y,x);
plot(x,y,'ro', x, yy,'b-');
grid on;
X=[0.155; 0.255; 0.755; 0.855];
Y=spline(x,y,X); % Інтерполяція кубічними сплайнами

% Порівняння значень, знайдених шляхом інтерполяції кубічними
сплайнами
% за допомогою стандартної функції spline(x,y,X),
% з точними значеннями функції, обчисленими зо формулою y= sinx.
% Цикл формування точних значень функції,обчислених зо формулою y=
sinx
% та порівняння їх зі значеннями, обчисленими за допомогою
% стандартної функції spline(x,y,X).
fprintf('\n Інтерпольовані та точні значення функції у вузлах
інтерполяції та їх різниці:');
fprintf('\n  i   X(i)   Y(i)     yt(i)      ry(i)');
for i=1:4
  yt(i)=sin(X(i));
  ry(i)= abs(yy(i)-yt(i));
  fprintf('\n %2d %6.3f %8.5f %10.8f  %10.8f  %10.8f', i, X(i),
Y(i), yt(i), ry(i) );
end
fprintf('\n');
end
```

Рисунок 3.15 – Програма інтерполяції функції y=sin(x) кубічними сплайнами за допомогою стандартної функції Spline (закінчення)

Результати тестування програми:

```
Інтерпольовані та точні значення функції у вузлах інтерполяції та
їх різниці:
    i   X(i)    Y(i)      yt(i)       ry(i)
    1  0.155  0.15438 0.15438010  0.04428990
    2  0.255  0.25225 0.25224541  0.04327459
    3  0.755  0.68530 0.68528867  0.29586867
    4  0.855  0.75455 0.75457092  0.27514092
```

Графік заданої функції зображений на рис. 3.16.

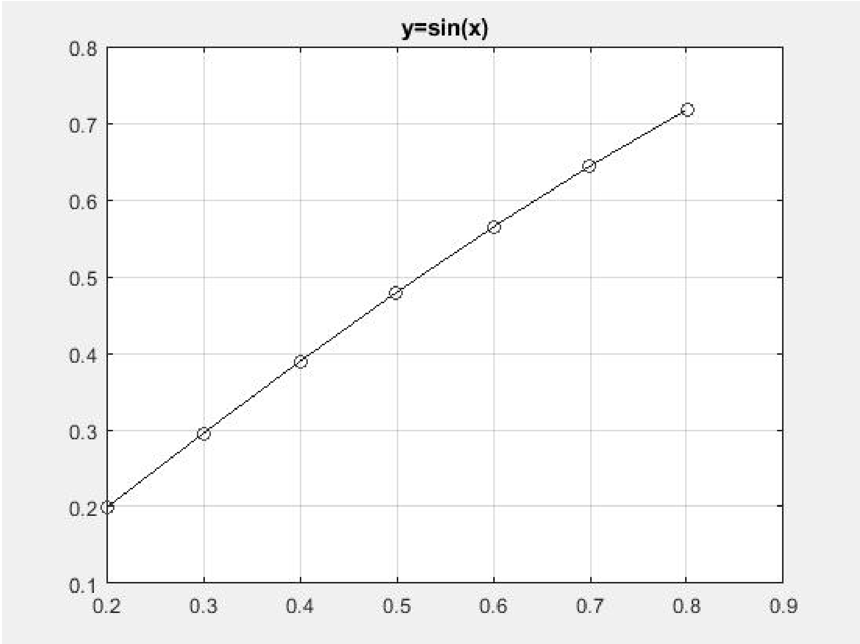


Рисунок 3.16 – Графічна інтерпретація інтерполяції таблично заданої функції на інтервалі $x \in [0.2;\ 0.8]$ кубічними сплайнами

## 3.10 Інтерполяція таблично заданої функції y=sin(x) за допомогою стандартної функції Interp1q

Код і результати тестування програми Method_Interp1q, яка є модифікацією програми лінійної інтерполяції, наведені на рис. 3.17. Як бачимо з результатів тестування, дану програму варто використовувати лише для інтерполяції значень функції лише для тих значень аргументів, які знаходяться всередині інтерполяційної таблиці.

```
function Method_Interpq1()
 % Інтерполяція таблично заданої функції y=sin(x) методом
Interp1q().
 % функція y=sin(x) задана таблично у 7 (семи) вузлах інтерполяції:
 %  i   x(i)    y(i)
 %  1  0.200  0.19867
 %  2  0.300  0.29552
 %  3  0.400  0.38942
 %  4  0.500  0.47943
 %  5  0.600  0.56464
 %  6  0.700  0.64422
 %  7  0.800  0.71736

 % Необхідно:
 % використовуючи стандартну функцію interp1q(x,y,X)
 % обчислити значення заданої функції у 4 (чотирьох) точках x0(i)
 % (i=1..4), що не співпадають з вузлами інтерполяції:

 %  i  x0(i)
 %  1  0.155
 %  2  0.255
 %  3  0.755
 %  4  0.855

 x=(0.2:0.1:0.8)';
 y=sin(x);
 xx=(0.2:0.05:0.8)';
 yy=interp1q(x,y,xx); % формування вектора yy
 subplot (2,1,1); % Ділимо вікно на 2 рядки і один стовпець
 plot(x,y,'k*'); grid on;
  title('y=sin(x)');
  subplot (2,1,2); % Ділимо вікно на 2 рядки і один стовпець
  plot(xx,yy,'r-'); grid on;
  title('y=sin(x)');
  X=[0.155; 0.255; 0.755; 0.855];
  % Формуємо масив значень функції для заданих значень аргументів,
  % визначених шляхом інтерполяції)многочленами Ерміта:
  Y=interp1q(x,y,X); % лінійна інтерполяція функції
  % Порівняння значень, знайдених шляхом інтерполяції многочленами
 Ерміта
  % за допомогою стандартної функції interp1l(x,y,X),
  % з точними значеннями функції, обчисленими зо формулою y= sinx.
  % Цикл формування точних значень функції,обчислених зо формулою y=
 sinx
  % та порівняння їх зі значеннями, обчисленими за допомогою
  % стандартної функції interp1q(x,y,X).
  fprintf('\n Інтерпольовані та точні значення функції у вузлах
 інтерполяції та їх різниці:');
  fprintf('\n  i   X(i)   Y(i)     yt(i)      ry(i)');
  for i=1:4
   yt(i)=sin(X(i));
   ry(i)= abs(yy(i)-yt(i));
   fprintf('\n %2d %6.3f %8.5f %10.8f  %10.8f  %10.8f', i, X(i),
 Y(i), yt(i), ry(i) );
  end
  fprintf('\n');
 end
```

Рисунок 3.17 – Програма інтерполяції функції y=sin(x) за допомогою стандартної функції Interp1q

Результати тестування програми:

```
  Інтерпольовані та точні значення функції у вузлах інтерполяції
та їх різниці:
    i   X(i)   Y(i)      yt(i)       ry(i)
    1  0.155       NaN 0.15438010  0.04428923
    2  0.255  0.25194 0.25224541  0.00515064
    3  0.755  0.68444 0.68528867  0.38976846
    4  0.855       NaN 0.75457092  0.41210164
```

Графік заданої функції зображений на рис. 3.18.

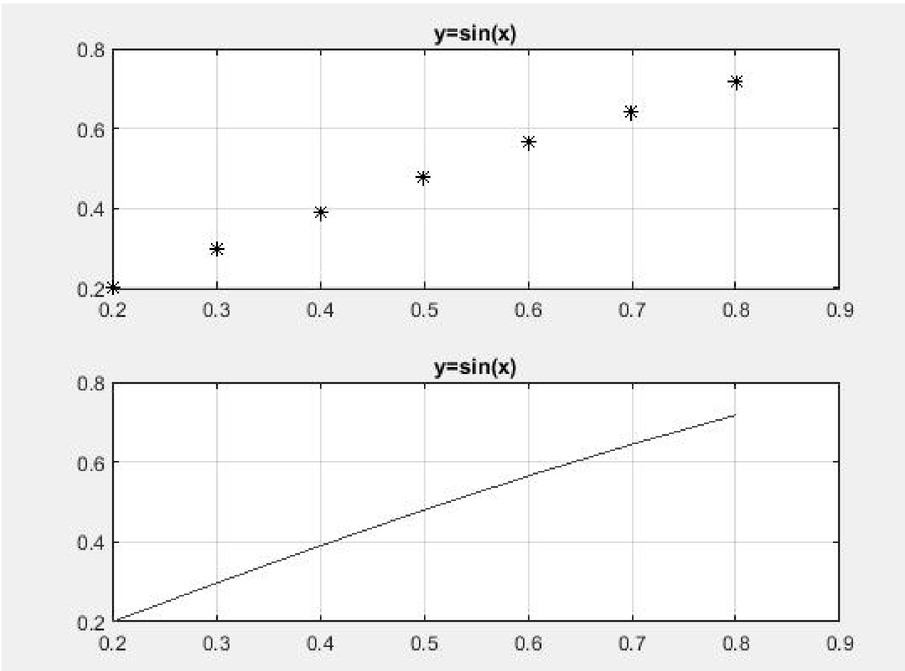


Рисунок 3.18 – Графічна інтерпретація інтерполяції таблично заданої функції на інтервалі $x \in [0.2;\ 0.8]$ за допомогою стандартної функції Interp1q

# ВИСНОВКИ

У першому розділі кваліфікаційної роботи виконано огляд математичних та програмних методів інтерполяції значень таблично заданих функцій такі як перший інтерполяційний метод Ньютона, другий інтерполяційний метод Ньютона та метод Лагранжа.

У другому розділі кваліфікаційної роботи розглянуто методику інтерполяції значень таблично заданих функцій.

У кваліфікаційній роботі розроблено і детально описано у третьому розділі програмно-демонстраційний комплекс у такому складі:

- програма інтерполяції таблично заданої функції на ділянці її монотонного зростання за допомогою формул Ньютона Newton_Formula;
- програма інтерполяції таблично заданої функції на ділянці її монотонного спадання за допомогою формул Ньютона Newton_Formula_1;
- програма інтерполяції таблично заданої функції на ділянці її перегину за допомогою формул Ньютона Newton_Formula_2;
- програма інтерполяції таблично заданої функції за допомогою полінома Лагранжа Lagrange_Polinom;
- програма інтерполяції таблично заданої функції за допомогою стандартної функції Nearest_Interp1;
- програма інтерполяції таблично заданої функції за допомогою стандартної функції Linear_Interp1;
- програма інтерполяції таблично заданої функції за допомогою стандартної функції Pchip_Interp1;
- програма інтерполяції таблично заданої функції за допомогою стандартної функції Cube_Spline;
- програма інтерполяції таблично заданої функції за допомогою стандартної функції Method_Interp1q.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ


1. Фокіна Д. В. Перша інтерполяційна формула Ньютона для рівновіддалених вузлів інтерполяції. *Всеосвіта*. URL: https://vseosvita.ua/library/persa-interpolacijna-formula-nutona-dla-rivnoviddalenih-vuzliv-interpolacii-43378.html (дата звернення: 25.02.2023).
2. Перша інтерполяційна формула Ньютона для рівновіддалених вузлів інтерполяції. *www.mathros.net.ua - Сайт для студентів спеціальності інформатика*. URL: https://www.mathros.net.ua/persha-interpoljacijna-formula-njutona-dlja-rivnoviddalenyh-vuzliv-interpoljacii.html (дата звернення: 28.02.2023).
3. Регресія. *Презентації українською мовою*. URL: https://svitppt.com.ua/algebra/regresiya.html (дата звернення: 28.02.2023).
4. Горда О. В. Чисельні методи. *Київський національний університет будівництва і архітектури*. URL: https://org2.knuba.edu.ua/pluginfile.php/34509/mod_resource/content/1/конспект_2009.pdf (дата звернення: 04.03.2023).
5. Інтерполяційний многочлен Лагранжа. *StudFiles*. URL: https://studfile.net/preview/7370221/page:2/ (дата звернення: 13.03.2023).
6. Поліном лагранжа. побудова в maple. *Контрольна робота - вища математика, теорія ймовірностей, диференціальні рівняння*. URL: https://yukhym.com/uk/matematika/polinom-lagranzha-v-maple.html (дата звернення: 15.03.2023).
7. Інтерполяція і наближення функцій. *Павлоградський Фаховий Коледж*. URL: http://amc.ptngu.com/rozdil6.html (дата звернення: 18.03.2023).
8. Комп'ютерне моделювання систем та процесів. Методи обчислень / Р. Н. Квєтний та ін. *Вінницький Національний Технічний Університет*. URL: https://web.posibnyky.vntu.edu.ua/fksa/2kvetnyj_komp'yuterne_modelyuvannya_system_procesiv/t1/612..htm (дата звернення: 21.03.2023).

9. 1-D data interpolation (table lookup) - MATLAB interp1. *MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink*. URL: https://www.mathworks.com/help/matlab/ref/interp1.html (date of access: 28.03.2023).
10. 2-D and 3-D plots- MATLAB & simulink. *MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink*. URL: https://www.mathworks.com/help/matlab/learn_matlab/plots.html (date of access: 07.03.2023).
11. Create 2-D line plot- MATLAB & simulink. *MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink*. URL: https://www.mathworks.com/help/matlab/creating_plots/using-high-level-plotting-functions.html (date of access: 15.03.2023).
12. Houcque D. Introduction to matlab for engineering students. *Northwestern's McCormick School of Engineering*. URL: https://www.mccormick.northwestern.edu/documents/students/undergraduate/introduction-to-matlab.pdf (date of access: 06.03.2023).
13. Interpolation of multiple 1-D value sets- MATLAB & simulink. *MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink*. URL: https://www.mathworks.com/help/matlab/math/interpolation-of-multiple-1-d-value-sets.html (date of access: 21.03.2023).
14. Kiss P. D. E. M. MATLAB Tutorial - An introduction for beginners. *Tutorials, quizzes, learning apps - Prof. E. Kiss, Hochschule Kaiserslautern*. URL: https://www.evamariakiss.de/tutorial/matlab/ (date of access: 02.03.2023).
15. MATLAB tutorial. *ETH Zürich - Homepage | ETH Zürich*. URL: https://ethz.ch/content/dam/ethz/special-interest/bsse/cobi-dam/documents/Matlab_Tutorial/matlab_tutorial.pdf (date of access: 22.02.2023).