

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. Сікорського

Кафедра
інформатики та програмної інженерії
(повна назва кафедри, циклової комісії)

КУРСОВА РОБОТА

з «Основи програмування»
(назва дисципліни)

на тему: Розв'язання СЛАР точними методами

Студента (ки, ів) 1 курсу, групи ІІІ-35
Адаменко Арсен Богданович

Спеціальності 121 «Інженерія програмного
забезпечення»

Керівник
Головченко М.М.
(посада, вчене звання, науковий
ступінь, прізвище та ініціали)

Кількість балів: _____

Національна оцінка _____

Члени комісії

(підпис)

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Київ- 202_ рік

Кафедра інформатики та програмної інженерії

Дисципліна Основи програмування

Напрям "ІПЗ"

Курс 1 Група ІП-35

Семестр 2

ЗАВДАННЯ

на курсову роботу студента

Адаменко Арсен Богданович

(прізвище, ім'я, по батькові)

1. Тема роботи Розв'язання СЛАР точними методами

2. Строк здачі студентом закінченої роботи !!!!!!!!!!!!!!

3. Вихідні дані до роботи Технічне завдання додаток А

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)
ВСТУП, ПОСТАНОВКА ЗАДАЧІ, ТЕОРЕТИЧНІ ВІДОМОСТІ, ОПИС АЛГОРИТМІВ,
ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ТЕСТУВАННЯ ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ, ВИСНОВКИ, ПЕРЕЛІК ПОСИЛАНЬ, ДОДАТОК А ТЕХНІЧНЕ
ЗАВДАННЯ, ДОДАТОК Б ТЕКСТИ ПРОГРАМНОГО КОДУ

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Дата видачі завдання 03 квітня 2024

КАЛЕНДАРНИЙ ПЛАН

| № п/п | Назва етапів курсової роботи | Термін виконання етапів роботи | Підписи керівника, студента |
|-------|---|--------------------------------|-----------------------------|
| 1. | Отримання теми курсової роботи | 03 квітня 2024 | |
| 2. | Підготовка ТЗ | 03 квітня 2024 | |
| 3. | Пошук та вивчення літератури з питань курсової роботи | 22 квітня 2024 | |
| 4. | Розробка сценарію роботи програми | 06 травня 2024 | |
| 6. | Узгодження сценарію роботи програми з керівником | 15 травня 2024 | |
| 5. | Розробка (вибір) алгоритму рішення задачі | 06 травня 2024 | |
| 6. | Узгодження алгоритму з керівником | 15 травня 2024 | |
| 7. | Узгодження з керівником інтерфейсу користувача | 8 травня 2024 | |
| 8. | Розробка програмного забезпечення | 01 травня 2024 | |
| 9. | Налагодження розрахункової частини програми | 15 травня 2024 | |
| 10. | Розробка та налагодження інтерфейсної частини програми | 15 травня 2024 | |
| 11. | Узгодження з керівником набору тестів для контрольного прикладу | | |
| 12. | Тестування програми | | |
| 13. | Підготовка пояснювальної записки | | |
| 14. | Здача курсової роботи на перевірку | | |
| 15. | Захист курсової роботи | | |

Студент

_____ (підпис)

Керівник

_____ (підпис)

Головченко Максим Миколайович

_____ (прізвище, ім'я, по батькові)

"__" _____ 20__ р.

АНОТАЦІЯ

Пояснювальна записка до курсової роботи: 55 сторінок, 24 рисунки, 28 таблиць, 4 посилання.

Мета роботи: Метою курсової роботи є забезпечення надійності та коректності програмного забезпечення для розв'язання СЛАР різними точними методами.

Вивчено методи: LUP-метод, метод Гауса-Холецького, метод обертання.

Виконана програмна реалізація алгоритму LUP-метода, метода Гауса-Холецького, метода обертання.

СИСТЕМА ЛІНІЙНИХ РІВНЯНЬ, ТОЧНІ МЕТОДИ ВИРІШЕННЯ, LUP-МЕТОД, МЕТОД ГАУСА-ХОЛЕЦЬКОГО, МЕТОД ОБЕРТАННЯ.

ЗМІСТ

| | |
|--|----|
| ВСТУП..... | 5 |
| 1 ПОСТАНОВКА ЗАДАЧІ..... | 6 |
| 2 ТЕОРЕТИЧНІ ВІДОМОСТІ..... | 7 |
| 3 ОПИС АЛГОРИТМІВ..... | 8 |
| 3.1. Загальний алгоритм..... | 8 |
| 3.2. Алгоритм LUP-методу..... | 8 |
| 3.3. Алгоритм методу Гауса-Холеского..... | 8 |
| 3.4. Алгоритм методу обертання..... | 8 |
| 4 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ..... | 9 |
| 4.1. Діаграма класів програмного забезпечення..... | 9 |
| 4.2. Опис методів частин програмного забезпечення..... | 9 |
| 4.2.1. Користувачські методи..... | 9 |
| 4.2.2. Стандартні методи..... | 9 |
| 5 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ..... | 10 |
| 5.1. План тестування..... | 10 |
| 5.2. Приклади тестування..... | 10 |
| 6 ІНСТРУКЦІЯ КОРИСТУВАЧА..... | 11 |
| 7 АНАЛІЗ РЕЗУЛЬТАТІВ..... | 12 |
| ВИСНОВКИ..... | 13 |
| ПЕРЕЛІК ПОСИЛАНЬ..... | 14 |
| ДОДАТОК А ТЕХНІЧНЕ ЗАВДАННЯ..... | 15 |
| ДОДАТОК Б ТЕКСТИ ПРОГРАМНОГО КОДУ..... | 18 |

ВСТУП

Дана робота висвітлює методи розв'язання систем лінійних алгебраїчних рівнянь для вирішення великої кількості задач та проблем у напрямках галузі інформаційних технологій, які пов'язані з такими речами, алогритмами та методами, як графіка, штучний інтелект і машинне навчання.

Дана робота та програмне забезпечення є актуальним через використання нових підходів до розробки та супроводу програмного продукту, забезпечення додакового функціоналу для програмного забезпечення з цим призначенням.

Дане програмне забезпечення призначене для розв'язання систем лінійних алгебраїчних рівнянь різними методами з підрахунком практичної часової складності (кількість ітерацій), візуалізації розв'язків систем рівнянь, а також вивід розв'язків систем рівнянь до текстового файлу.

1 Постановка задачі

Розробити програмне забезпечення, що буде знаходити рішення для заданої СЛАР наступними методами:

- а) LUP-метод;
- б) метод Гауса-Холецького;
- в) метод обертання;

Вхідними даними для даної роботи є СЛАР, яка задана в матричному вигляді:

$$AX=B$$

, де A – матриця коефіцієнтів, X – вектор шуканих значень (рішення системи),

B – вектор вільних членів. Програмне забезпечення повинно обробляти матрицю коефіцієнтів та стовпець вільних членів для СЛАР розмірність яких знаходиться в межах від 1 до 10.

Вихідними даними для даної роботи являється сукупність дійсних чисел, що є розв'язками даної системи, які виводяться на екран. Програмне забезпечення повинно видавати розв'язок за умови, що для вхідних даних обраний метод сходиться. Якщо це не так, то програма повинна вивести відповідне повідомлення. Якщо розмірність системи рівне двом невідомим, то програмне забезпечення повинно виводити графік системи. Якщо система не має розв'язків або їх нескінченна кількість, то програма повинна видати відповідне повідомлення.

2 ТЕОРЕТИЧНІ ВІДОМОСТІ

Систему лінійних алгебраїчних рівнянь (далі — СЛАР) з n рівнянь можна задати наступним чином:

$$AX=B \quad (2.1)$$

де:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Тоді якщо , то система (2.1) має розв'язок та він є лише єдиним. Якщо система має єдиний розв'язок, то його можна знайти одним із наступних методів.

2.1. LUP-метод

Цей метод опирається на розкладання матриці коефіцієнтів A у вигляді добутку матриць L та U :

$$A=LU \quad (2.2)$$

де L — нижня трикутна матриця, а U — верхня трикутна матриця, де усі діагональні елементи дорівнюють 1:

$$L = \begin{bmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & 0 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & u_{12} & \dots & u_{1n} \\ 0 & 1 & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \quad (2.3)$$

Тоді з цього твердження випливає, що LUP-метод розв'язання СЛАР складається з двох головних етапів:

1. етапу факторизації матриці A ;
2. етапу отримання X .

Під час факторизації вектор B не змінюється, як ви бачите.

Із виразів (2.2) та (2.3) випливає, що значення всіх елементів матриці A можна записати наступним компактним чином:

$$\begin{aligned} a_{ij} &= \left(\sum_{k=1}^{j-1} l_{ik} u_{kj} \right) + l_{ij} \\ a_{ji} &= \left(\sum_{k=1}^{j-1} l_{jk} u_{ki} \right) + l_{jj} u_{ji} \end{aligned} \quad (2.4)$$

де $i = \overline{1, n}, j = \overline{1, i}$.

У виразі (2.4) записано, що i — номер рядка матриці L та номер стовпця матриці U , а j — вже номер стовпця матриці L та номер рядка матриці U .

З рівняння (2.4) випливає, що факторизація відбувається за n стадій. На кожній стадії j поточний елемент a_{ji} матриці L наступним чином:

$$l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj}, \quad i = \overline{j, n} \quad (2.5)$$

а елемент u_{ji} вже іншим:

$$u_{ji} = \frac{a_{ji} - \sum_{k=1}^{j-1} l_{jk} u_{ki}}{l_{jj}}, \quad i = \overline{j+1, n} \quad (2.6)$$

Елементи двох матриць обчислюються у шаховому порядку.

За деяких умов за формулами (2.5) та (2.6) значення елементів матриці L та U можна побати як $l_{i1} = a_{i1}, i = \overline{1, n}$, а $u_{1j} = \frac{a_{1j}}{l_{11}}, j = \overline{2, n}$.

Так як метод називається як LUP-методом, а не LU-методом, то він має ще один вбудований крок для факторизації матриці A . Так як при обрахуванні елемента u_{ji} значення l_{jj} може набувати від'ємного значення, що зробить розв'язання неможливим, або настільки близьким до нуля, що точність вирішення СЛАР може погіршитися.

Одним з методів вирішення цієї проблеми є знаходження такого рядка $k = \overline{j, n}$, для якого:

$$|a_{kj}| = |\max a_{ij}|, \quad i = \overline{j, n} \quad (2.7)$$

Після чого треба зберігти вектор P , де $P_i \in \overline{1, n}, i = \overline{1, n}$ зберігає оригінальний номер рядка матриці A . По замовчуванню матриця P має наступну початкову конфігурацію перед факторизацією: $P_i = i, i = \overline{1, n}$

Так як перед обрахуванням елементів матриці L та U в ітерації j елементи рядків $i=\overline{j,n}$ є незмінними до цього часу, то слід зробити перестановку перед поточною ітерацією j , а також обміняти місцями значення P_j та P_{p_j} , де j — поточний номер ітерації, а i — значення рядка, для якого формула (2.7) коректною.

В результаті факторизації ми маємо наступне рівняння СЛАР:

$$LUX=B \quad (2.8)$$

Далі рівняння (2.8) можна переписати як:

$$LY=B, Y=UX \quad (2.9)$$

Потім в нас з (2.9) з'являється наступне рівняння для розв'язку Y :

$$LY=B \quad (2.10)$$

А також (2.9) можна перетворити для розв'язку X :

$$UX=Y \quad (2.11)$$

Так як L є трикутною матрицею, то для розв'язку Y ми можемо перетворити рівняння (2.10) на наступне:

$$y_i = \frac{b_{p_{ii}} - \sum_{j=1}^{i-1} l_{ij} y_j}{l_{ii}}, \quad i=\overline{1,n}$$

де P — вектор перестановки для матриці A .

Так як U теж є трикутною матрицею, то вектор-стовпець розв'язок X можна переписати за допомогою формули (2.11) як:

$$x_i = y_i - \sum_{j=i+1}^n u_{ij} x_j, \quad i=\overline{n,1}$$

2.2. Метод Гауса-Холецького

Цей метод використовується для розв'язання СЛАР з матрицею коефіцієнтів A , яка має бути емітовою. Він ґрунтується на розкладанні матриці на добуток матриць L , D та L^+ :

$$A=LDL^+ \quad (2.12)$$

де L — нижня трикутна матриця, D — діагональна матриця, L^+ — комплексно спряженена матриця до матриці L та є верхньо трикутною до неї.

З виразу (2.12) випливає, що кожен елемент матриці A можна записати як:

$$a_{ij} = \sum_{k=1}^i l_{ik} d_{kk} \bar{l}_{jk}, \quad i \geq j \quad (2.13)$$

де \bar{l}_{jk} — елемент, комплексно-спряжений до l_{jk} .

У випадку, якщо $i=j$, то можна одержати наступне рівняння:

$$d_{jj} l_{jj}^2 = a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2 d_{kk} \quad (2.14)$$

Якщо $l_{ii}=1, i=\overline{1, n}$, то вираз (2.14) можна переписати вже ось так:

$$d_{jj} = a_{jj} - \sum_{k=1}^{j-1} (l_{jk}^2 d_{kk}), \quad j=\overline{1, n} \quad (2.15)$$

Тепер якщо $i>j$, то вираз (2.14) вже буде мати наступний вигляд:

$$l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} d_{kk} \bar{l}_{jk}}{d_{jj}}, \quad j=\overline{1, n}, i=\overline{j+1, n} \quad (2.16)$$

Таким чином, LDL факторизація матриці відбувається за n ітерацій. Всі елементи $l_{ii}, i=\overline{1, n}$ дорівнюють 1. На кожній ітерації j треба обчислити значення елементів d_{jj} за формулою (2.15), а потім — l_{ij} за (2.16).

Після факторизації матриці A ми отримаємо наступне рівняння:

$$LDL^+ X = B \quad (2.17)$$

Вираз (2.17) можна переписати як:

$$LY = B, \quad Y = DL^+ X \quad (2.18)$$

далі як:

$$DZ = Y, \quad Z = L^+ X \quad (2.19)$$

в кінці вже як:

$$L^+ X = Z \quad (2.20)$$

З рівності (2.18) випливає, що:

$$y_i = b_i - \sum_{j=1}^{i-1} l_{ij} y_j, \quad i=\overline{1, n}$$

З виразу (2.19):

$$z = \frac{y_i}{d_{ii}}, \quad i = \overline{1, n}$$

Далі з (2.20) впливає остаточною вираз обчислення вектора X :

$$x_i = z_i - \sum_{j=i+1}^n \bar{l}_{ji} x_j, \quad i = \overline{n, 1}$$

2.3. Метод обертання

Немає відповідної літератури

3 ОПИС АЛГОРИТМІВ

Перелік всіх основних змінних та їхнє призначення наведено в таблиці

3.1.

Таблиця 3.1 — Основні змінні та їхні призначення.

| Змінна | Призначення |
|------------|----------------------------------|
| A | Матриця коефіцієнтів |
| B | Вектор вільних коефіцієнтів |
| IsSolvable | Ознака можливості вирішення СЛАР |

3.1. Загальний алгоритм

1. ПОЧАТОК

2. Зчитати розмірність системи.

3. Зчитати матрицю системи та стовпець вільних членів:

3.1. Зчитати матрицю коефіцієнтів:

3.1.1. Цикл проходу по всіх рядках матриці системи (a_i — поточний рядок):

3.1.1.1. Цикл проходу всіх стовпцях матриці системи (a_{ij} — поточний рядок):

3.1.1.1.1. ЯКЩО поточний елемент матриці — вірно записане число, ТО записати його в відповідну комірку A. ІНАКШЕ видати повідомлення про помилку та перейти до пункту 8.

3.2. Зчитати вектор вільних коефіцієнтів:

3.2.1. Цикл проходу по всіх елементах стовпця вільного членів:

3.2.1.1. Якщо поточний елемент вектора вільних коефіцієнтів — вірно записане число, ТО записати його в відповідну комірку B. ІНАКШЕ видати повідомлення про помилку та перейти до пункту 8.

4. ЯКЩО обраний LUP-метод, ТО обробити дані згідно алгоритму методу Якобі (підрозділ 3.2).
5. ЯКЩО обраний метод Гауса-Зейделя, ТО обробити дані згідно алгоритму методу Якобі (підрозділ 3.3).
6. ЯКЩО обраний метод обертання, ТО обробити дані згідно алгоритму методу Якобі (підрозділ 3.4).
7. ЯКЩО IsSolvable дорівнює правді, ТО:
 - 7.1. ЯКЩО обрана система на дві невідомих, ТО побудувати та вивести графік системи.
 - 7.2. Вивести рішення системи.
 - 7.3. Записати систему та її рішення у файл.
8. ІНАКШЕ вивести повідомлення про неможливість розв'язання СЛАР вказаним методом.
9. КІНЕЦЬ

3.2. Алгоритм LUP-методу

1. ПОЧАТОК
2. Отримати розмірність матриці A як n .
3. Створити новий вектор P розміром n .
4. Заповнення вектора P початковими значеннями:
 - 4.1. ЦИКЛ по всім індексам нової матриці P лічильником i :
 - 4.1.1. Встановити значення p_i як i .
5. Створити нову матрицю NA розмірністю в n рядків та n стовпців.
6. Копіювання матриці A до матриці NA :
 - 6.1. ЦИКЛ по всім рядкам нової матриці NA лічильником i :
 - 6.1.1. ЦИКЛ по всім стовпцям нової матриці NA лічильником j :
 - 6.1.1.1. Встановити значення na_{ij} як у елемента a_{ij} .
7. Знайти факторизацію матриці NA :

7.1. Обміняти значення двох рядків матриці NA для забезпечення можливості розв'язання у випадку, коли $na_{jj}=0$:

7.1.1. ЦИКЛ для змінної j від 1 до n :

7.1.1.1. Знайти рядок з максимальним значенням i , де $|na_{ij}| = |\max na_{kj}|, j = \overline{1, n}, i = \overline{1, n}$:

7.1.1.1.1. Встановити значення для змінної $MaxColumnValue$ як a_{jj} .

7.1.1.1.2. Встановити значення для змінної $MaxColumnIndex$ як j .

7.1.1.1.3. ЦИКЛ для змінної i від j до n :

7.1.1.1.3.1. ЯКЩО $|na_{ij}| > |na_{MaxColumnIndex, j}|$, ТО

7.1.1.1.3.2. Встановити значення для змінної $MaxColumnValue$ як na_{ij} .

7.1.1.2. ЯКЩО $MaxColumnValue=0$ є правдою, ТО

7.1.1.2.1. Встановити змінну $IsSolvable$ як хибна.

7.1.1.2.2. Перейти до пункту 11.

7.1.2. ЦИКЛ для змінної i від 1 до n :

7.1.2.1. Обміняти елементи матриці NA $a_{j,i}$ та $na_{MaxColumnIndex, i}$ місцями.

7.1.3. Обміняти елементи вектора P p_j та $p_{MaxColumnIndex}$ місцями.

7.2. Ітерація факторизації матриці NA :

7.2.1. ЦИКЛ для змінної i від j до n :

7.2.1.1. Встановити змінну s як суму циклу з початковим значенням 0.

7.2.1.2. ЦИКЛ для змінної k від 1 до $j - 1$:

7.2.1.2.1. Збільшити значення змінної s як суми циклу на $na_{ik}a_{kj}$.

7.2.1.3. Встановити значення елементу матриці a_{ij} як $a_{ij} - s$.

7.2.1.4. ЯКЩО $i > j$, ТО

- 7.2.1.4.1. Встановити змінну s як суму циклу з початковим значенням 0.
- 7.2.1.4.2. ЦИКЛ для змінної k від 0 до $j - 1$:
 - 7.2.1.4.2.1. Збільшити значення змінної s як суми циклу на $a_{jk} a_{ki}$.
- 7.2.1.4.3. Встановити значення елементу матриці a_{ji} як $\frac{a_{ji} - s}{a_{jj}}$.
- 7.3. Створити нову матрицю L розмірністю в n рядків та n стовпців.
- 7.4. Створити нову матрицю U розмірністю в n рядків та n стовпців.
- 7.5. Заповнити матрицю L матрицею NA :
 - 7.5.1. ЦИКЛ по всіх рядках нової матриці L лічильником j :
 - 7.5.1.1. ЦИКЛ для змінної i від 1 до j :
 - 7.5.1.1.1. Встановити значення l_{ji} як a_{ji} .
- 7.6. Заповнити матрицю L матрицею NA :
 - 7.6.1. ЦИКЛ по всіх рядках нової матриці U лічильником j :
 - 7.6.1.1. ЦИКЛ для змінної i від j до n :
 - 7.6.1.1.1. ЯКЩО $i > j$, ТО
 - 7.6.1.1.1.1. Встановити значення u_{ji} як a_{ji} .
 - 7.6.1.1.2. ІНАКШЕ
 - 7.6.1.1.2.1. Встановити значення u_{ji} як 1.
8. Обчислити значення вектора Y :
 - 8.1. Створити новий вектор Y розміру n .
 - 8.2. ЦИКЛ для змінної i від 1 до n :
 - 8.2.1. Встановити змінну s як суму циклу з початковим значенням 0.
 - 8.2.2. ЦИКЛ для змінної k від 0 до $i - 1$:
 - 8.2.2.1. Збільшити значення змінної s як суми циклу на $l_{ik} y_k$.
 - 8.2.3. Встановити значення вектора y_i як $\frac{b_{pi} - s}{l_{ii}}$.

9. Обчислити значення вектора X :

9.1. Створити новий вектор Y розміру n .

9.2. ЦИКЛ для змінної i від n до 1 донизу:

9.2.1. Встановити змінну s як суму циклу з початковим значенням 0 .

9.2.2. ЦИКЛ для змінної k від $i + 1$ до n :

9.2.2.1. Збільшити значення змінної s як суми циклу на $u_{ik}x_k$.

9.2.3. Встановити значення розв'язка x_i як $y_i - s$.

10. Встановити змінну $IsSolvable$ як правда.

11. КІНЦЕЬ

3.3. Алгоритм методу Гауса-Холецького

3.4. Загальний методу обертання

1. ПОЧАТОК

2. Отримати розмірність матриці A як n .

3. Створити нову матрицю NA розмірністю в n рядків та $n + 1$ стовпців.

4. Заповити нову матрицю NA значеннями з матриці A та B .

4.1. ЦИКЛ по всім рядкам нової матриці лічильником i :

4.1.1. ЦИКЛ по всім стовпцям нової матриці лічильником j :

4.1.1.1. ЯКЩО вираз $j \leq n$ є правдивим, ТО

4.1.1.1.1. Встановити значення na_{ij} як у елемента a_{ij} .

4.1.1.2. ІНАКШЕ

4.1.1.2.1. Встановити значення na_{ij} як у елемента b_i .

5. Обчислити розкладання нової матриці NA :

5.1. ЦИКЛ для змінної i від 1 до $n - 1$:

5.1.1. ЦИКЛ для змінної j від $i + 1$ до n :

5.1.1.1. Встановити значення для змінної b як na_{ji} .

5.1.1.2. Встановити значення для змінної a як na_{ii} .

5.1.1.3. ЯКЩО значення $a^2 + b^2$ не є позитивним, ТО

5.1.1.3.1. Встановити змінну *IsSolvable* як хиба.

5.1.1.3.2. Перейти до пункту 8.

5.1.1.4. Встановити значення для змінної c як $\frac{a}{\sqrt{(a^2 + b^2)}}$.

5.1.1.5. Встановити значення для змінної b як $\frac{b}{\sqrt{(a^2 + b^2)}}$.

5.1.1.6. ЦИКЛ для змінної k від i до $n + 1$:

5.1.1.6.1. Встановити значення для змінної t як na_{ik} .

5.1.1.6.2. Встановити значення елемента матриці na_{ik} як $ca_{ik} + sa_{jk}$.

5.1.1.6.3. Встановити значення елемента матриці na_{jk} як $-st + ca_{jk}$.

6. Обчислити значення вектора X довжиною n :

6.1. ЦИКЛ для змінної i від n до 1 донизу:

6.1.1. Встановити змінну s сумми цикла як 0.

6.1.2. ЦИКЛ для змінної j від $i + 1$ до n :

6.1.2.1. Збільшити значення s змінної суми цикла на $a_{ij}x_j$.

6.1.2.2. ЯКЩО значення a_{ii} є нулем, ТО

6.1.2.2.1. Встановити змінну *IsSolvable* як хиба.

6.1.2.2.2. Перейти до пункту 8.

6.1.2.3. Встановити значення розв'язку x_i як $\frac{a_{in} - s}{a_{ii}}$.

7. Встановити змінну *IsSolvable* як правда.

8. КІНЕЦЬ

4 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1. Діаграма класів програмного забезпечення

Немає

4.2. Опис методів частин програмного забезпечення

4.2.1. Користувацькі методи

| № п/п | Назва класу | Назва методу | Призначення методу | Опис вхідних параметрів | Опис вихідних параметрів | Заголовочний файл |
|-------|-------------|--------------|--|-------------------------|--------------------------|-------------------|
| 1 | Vector | Vector | Створити новий пустий вектор | | | Vector.hpp |
| 2 | Vector | Vector | Створити новий вектор довжини size | size | | Vector.hpp |
| 3 | Vector | operator[] | Отримати посилання на елемент вектора | index | | Vector.hpp |
| 4 | Vector | Size | Отримати довжину вектора | | | |
| 5 | Matrix | Matrix | Створити нову пусту матрицю | | | Matrix.hpp |
| 6 | Matrix | Matrix | Створити матрицю розмірністю width на height | height, width | | Matrix.hpp |
| 7 | Matrix | At | Отримати посилання на елемент | y, x | | Matrix.hpp |

| № п/п | Назва класу | Назва методу | Призначення методу | Опис вхідних параметрів | Опис вихідних параметр ів | Заголово чний файл |
|----------|-----------------|-----------------|---|-------------------------------|------------------------------------|------------------------------|
| | | | матриці | | | |
| 8 | Matrix | Width | Отримати кількість стовпців матриці | | | Matrix.hp p |
| 9 | Matrix | Height | Отримати кількість рядків матриці | | | Matrix.hp p |
| 10 | Matrix | TryGetEdgeSize | Спробувати отримати розмірність квадратної матриці по її ширині | | | Matrix.hp p |
| 11 | Matrix | IsSquare | Чи є матриця квадратною | | | Matrix.hp p |
| 12 | AllocArray2D<T> | AllocArray2D | Створити пустий новий двовимірний масив | | | AllocArra y2D.inc.h pp |
| 13 | AllocArray2D<T> | AllocArray2D | Створити новий двовимірний масив шириною width та висотою height | height, width | | AllocArra y2D.inc.h pp |
| 14 | AllocArray2D<T> | At | Отримати посилання на елемент двовимірного масиву по ширині x та висоті y | y, x | | AllocArra y2D.inc.h pp |

| № п/п | Назва класу | Назва методу | Призначення методу | Опис вхідних параметрів | Опис вихідних параметр ів | Заголово чний файл |
|----------|---------------------|-------------------|--|-------------------------------|------------------------------------|------------------------------|
| 15 | AllocArray2 D<T> | Width | Отримати ширину двовимірний масиву | | | AllocArra y2D.inc.h pp |
| 16 | AllocArray2 D<T> | Height | Отримати висоту двовимірний масиву | | | AllocArra y2D.inc.h pp |
| 17 | SolvingRes ult | Successful | Отримати об'єкт типу класу цього методу. Об'єкт репрезентує, що статус значення є успішним при обробці | | | LSESolve r.hpp |
| 18 | SolvingRes ult | Error | Отримати об'єкт типу класу цього методу. Об'єкт репрезентує, що статус значення не є успішним при обробці | | | LSESolve r.hpp |
| 19 | SolvingRes ult | SetItersCou nt | Встановити кількість ітерацій при виконанні алгоритмів та вернути оригінальний об'єкт | itersCount | | LSESolve r.hpp |

| № п/п | Назва класу | Назва методу | Призначення методу | Опис вхідних параметрів | Опис вихідних параметр ів | Заголово чний файл |
|----------|----------------|--------------------------|---|-------------------------------|------------------------------------|--------------------------|
| 20 | LSESolver | LSESolver | Конструктор стану об'єкта цього типу | | | LSESolver r.hpp |
| 21 | LSESolver | ~LSESolver | Віртуальний деструктор для класів- нащадків | | | LSESolver r.hpp |
| 22 | LSESolver | SetEquationsCount | Встановити кількість лінійних рівнянь єдиної | eqsCount | | LSESolver r.hpp |
| 23 | LSESolver | SetVariablesCoefficients | Встановити матрицю коефіцієнтів | | | LSESolver r.hpp |
| 24 | LSESolver | SetFreeCoefficients | Встановити вектор вільних членів | | | LSESolver r.hpp |
| 25 | LSESolver | SolveLSE | Почати процес розв'язку встановленої ЛСАР | | | LSESolver r.hpp |
| 26 | LSESolver | IsLSESolvedSuccessfully | Отримати статус виконання алгоритму розв'язку | | | LSESolver r.hpp |
| 27 | LSESolver | GetSolvedValuesOnce | Отримати розв'язок встановленої СЛАР єдиної | | | LSESolver r.hpp |
| 28 | LSESolver | GetTotalIterations | Отримати | | | LSESolver |

| № п/п | Назва класу | Назва методу | Призначення методу | Опис вхідних параметрів | Опис вихідних параметрів | Заголовочний файл |
|-------|--------------------|---------------------|---|-------------------------|--------------------------|-------------------|
| | | rationsCount | кількість ітерацій при виконанні алгоритму | | | r.hpp |
| 29 | GUISession | GUISession | Створити об'єкт типу сесії графічного інтерфейсу | | | GUI.hpp |
| 30 | GUISession | Init | Ініціалізувати вікно графічної сесії аргументами командного рядка | argc, argv | | GUI.hpp |
| 31 | GUISession | GetWindow | Отримати посилання на вікно програми | | | GUI.hpp |
| 32 | Application Data | GetLSEInputDataRef | Отримати посилання на вхідні дані СЛАР | | | GUI.hpp |
| 33 | Application Data | GetLSEOutputDataRef | Отримати посилання на дані розв'язку СЛАР | | | GUI.hpp |
| 34 | Application Window | SetApplicationData | Встановити посилання на об'єкт, що зберігає дані програми | appData | | GUI.hpp |
| 35 | Application Window | ReadyWindow | Встановити стан вікна як | | | GUI.hpp |

| № п/п | Назва класу | Назва методу | Призначення методу | Опис вхідних параметрів | Опис вихідних параметр ів | Заголово чний файл |
|----------|---------------------|--------------------------------|---|-------------------------------|------------------------------------|--------------------------|
| | | | готового для опрацювання | | | |
| 36 | LSEConfigu rator | SetLSEInp utData | Встановити місце для запису конфігурації СЛАР | lseInputData | | GUI.hpp |
| 37 | LSESolveO utput | SetLSEOut putDataRef | Встановити місце для зчитання даних СЛАР | lseSolveData | | GUI.hpp |
| 38 | LSESolveO utput | SetLSEInp utDataRef | Встановити місце для зчитання даних про розв'язок СЛАР | lseInputData | | |
| 39 | LSESolveO utput | OutputSolv e | Показати розв'язки СЛАР | | | |
| 40 | LSESolveO utput | ClearSolve | Приховати розв'язки СЛАР | | | |
| 41 | LSESolver UI | SetLSEInp utData | Встановити місце для зчитання даних СЛАР | lseInputData | | |
| 42 | LSESolver UI | SetLSESol vesPlace | Встановити місце для запису даних про розв'язок СЛАР | lseSolvesPla ce | | |
| 43 | LSESolver UI | SetLSESol veOutputI ORef | Встановити віджет для відображення | lseSolveOut put | | |

| № п/п | Назва класу | Назва методу | Призначення методу | Опис вхідних параметрів | Опис вихідних параметр ів | Заголово чний файл |
|----------|----------------|-----------------|-----------------------|-------------------------------|------------------------------------|--------------------------|
| | | | розв'язків СЛАР | | | |
| 44 | | | | | | |

4.2.2. Стандартні методи

| № п/п | Назва класу | Назва методу | Призначення методу | Опис вхідних параметрів | Опис вихідних параметр ів | Заголово чний файл |
|----------|----------------|-----------------|-----------------------|-------------------------------|------------------------------------|--------------------------|
| | | | | | | |

5 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1. План тестування

Немає

5.2. Приклади тестування

Немає

6 ІНСТРУКЦІЯ КОРИСТУВАЧА

6.1. Робота з програмою

Після запуску виконавчого файлу з розширенням *.exe, відкривається головне вікно програми:

Рисунок 6.1 — Головне вікно програми

Далі треба встановити розмірність системи лінійних алгебраїчних рівнянь (далі — СЛАР) біля таблички «???», потім натиснути на клавішу «???» щоб встановити нову СЛАР, яка буде оброблятися програмою (рисунок 6.2).

Рисунок 6.2 — Вибір розміру системи

Після чого треба встановити всі коефіцієнти матриці та вільні коефіцієнти до всіх комірок форми введення (рисунок 6.3).

Рисунок 6.3 — Форма введення СЛАР

Потім треба натиснути на клавішу «???», що ви згодні встановити дану введену СЛАР до програми для подальшої обробки програмою (рисунок 6.3).

Якщо значення комірок форми введення СЛАР не є числами, то буде виведена помилка про неможливість підтвердити дану СЛАР. Якщо СЛАР введена правильно користувачем, то буде відображено час підтвердження СЛАР.

Далі користувач має вказати один з методів розв'язання введеної СЛАР до програми за допомогою вибору одного з методів у списку (рисунок 6.4) у підменю «???». Завжди перед кожною спробою розв'язання СЛАР, треба її підтвердити.

Рисунок 6.4 — Список вибору методу розв'язання СЛАР

Потім треба натиснути на клавішу «???» для запуску процесу розв'язання введеної СЛАР користувачем. Якщо СЛАР була розв'язана успішно, то програма відобразить час завершення розв'язання СЛАР та час можливості отримати розв'язок СЛАР графічно або виводом у файл в залежності від її розмірності.

Якщо СЛАР не була розв'язана вдало, то буде виведено повідомлення про неможливість розв'язку СЛАР.

Якщо СЛАР була розв'язана вдало та її розмірність дорівнює 2, то буде виведено її графічний розв'язок у вікні програми.

6.2. Формат вхідних даних

Користувач на вхід програми подається СЛАР у матричному вигляді, тобто задається за допомогою матриці системи та стовпця вільних членів, числа яких є дійсними.

Результатом виконання програми є розв'язок встановленої СЛАР. У вікні програми СЛАР буде відображатися вертикальною таблицею значень розв'язків СЛАР (рисунок 6.5). Якщо розв'язок буде виведено до файлу, то його змістом буде список дійсних чисел, де числа розділені одним пробілом.

Рисунок 6.5 — Табличний метод відображення розв'язків СЛАР

1.3. Системні вимоги

| | Мінімальні | Рекомендовані |
|----------------------------------|---|-------------------------------------|
| Операційна система | Linux (з останніми оновленнями) | ArchLinux (з останніми оновленнями) |
| Процесор | Intel i7-6700HQ (8) @ 3.500GHz | |
| Оперативна пам'ять | 4 GB RAM | 8 GB RAM |
| Відеоадаптер | Intel HD Graphics 530 | |
| Дисплей | 1920x1080 | 1920x1080 та вище |
| Прилади введення | Клавіатура, комп'ютерна миша | |
| Додаткове програмне забезпечення | Базові пакети («base», «base-devel»); бібліотека для GTKMM-3.0 («gtkmm3») | |

7 АНАЛІЗ РЕЗУЛЬТАТІВ

Немає

ВИСНОВКИ

Немає

ПЕРЕЛІК ПОСИЛАНЬ

- [1] Прокопенко, Ю. В., Татарчук, Д. Д., & Казміренко, В. А. (2003). Обчислювальна математика.
- [2] Шахно, С. М., Дудикевич, А. Т., & Левицька, С. М. (2009). ББК В 162.73 я73 Ш81.

КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. Сікорського

Кафедра
інформатики та програмної інженерії

Затвердив

Керівник Головченко М.М.

« 03 » квітня _____ 2024 р.

Виконавець:

Студент Адаменко А.Б.

« 03 » квітня _____ 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання курсової роботи

на тему: «Розв'язання СЛАР точними методами»

з дисципліни:

«Основи програмування»

Київ 2024

- 7.1. *Мета:* Метою курсової роботи є забезпечення надійності та коректності програмного забезпечення для розв'язання СЛАР різними точними методами.
- 7.2. *Дата початку роботи:* «_03»_квітня_2024 р.
- 7.3. *Дата закінчення роботи:* «___»_____2024 р.
- 7.4. *Вимоги до програмного забезпечення.*

1) Функціональні вимоги:

- Можливість ввести розмірність СЛАР.
- Можливість задавати вільні коефіцієнти та коефіцієнти невідомих змінних СЛАР.
- Можливість обирати один з методів розв'язання СЛАР.
- Розв'язати СЛАР обраним методом (LUP-методом, метод обертання, метод Гауса-Холеского (квадратного кореня)).
- Можливість зберігати результати розв'язання СЛАР у текстовий файл.
- Можливість графічного розв'язання СЛАР у випадку розмірності СЛАР у два рівняння.
- Відображати значення практичної складності алгоритмів розв'язання СЛАР.
- Нормальна обробка та реакція на некоректні дії користувача.
- Можливість відображення практичної складності алгоритму.

2) Нефункціональні вимоги:

- Можливість запускати та працювати з програмним забезпеченням на операційній системі «Linux».
- Все програмне забезпечення та супроводжувача технічна документація повинні задовольняти наступним ДЕСТам:

ГОСТ 29.401 - 78 - Текст програми. Вимоги до змісту та оформлення.

ГОСТ 19.106 - 78 - Вимоги до програмної документації.

ГОСТ 7.1 - 84 та ДСТУ 3008 - 2015 - Розробка технічної документації.

7.5. *Стадії та етапи розробки:*

- 1) Об'єктно-орієнтований аналіз предметної області задачі (до __.__.2024 р.)
- 2) Об'єктно-орієнтоване проектування архітектури програмної системи (до __.__.2024 р.)
- 3) Розробка програмного забезпечення (до __.__.2024 р.)
- 4) Тестування розробленої програми (до __.__.2024 р.)
- 5) Розробка пояснювальної записки (до __.__.2024 р.).
- 6) Захист курсової роботи (до __.__.2024 р.).

7.6. *Порядок контролю та приймання.* Поточні результати роботи над КР регулярно демонструються викладачу. Своєчасність виконання основних етапів графіку підготовки роботи впливає на оцінку за КР відповідно до критеріїв оцінювання.

ДОДАТОК Б ТЕКСТИ ПРОГРАМНОГО КОДУ

*Тексти програмного коду програмного забезпечення вирішення задачі
знаходження розв'язків системи лінійних рівнянь*

(Найменування програми (документа))

CD-RW

(Вид носія даних)

182 арк, 124 Кб

(Обсяг програми (документа), арк., Кб)

*студента групи ІІІ-35 І курсу
Адаменко А.Б.*