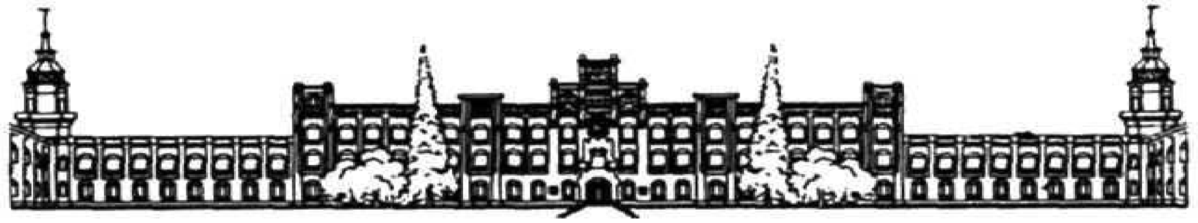


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

Лабораторна робота № 2

з дисципліни “Чисельні методи”
тема “Прямі методи розв’язання СЛАР та обчислення оберненої матриці”

Виконав(ла)
студент(ка) III курсу
групи КП-83
Матіюк Дарина Андріївна

Перевірив
“____” “____________” 20___ р.
Онай М. В.

варіант №12

Київ 2020

Мета роботи: набути вміння та навички використання прямих методів розв'язання систем лінійних алгебраїчних рівнянь і обчислення обернених матриць та застосування систем комп'ютерної математики, а також закріплення, поглиблення й узагальнення теоретичних знань щодо методів розв'язання систем лінійних алгебраїчних рівнянь і обчислення обернених матриць.

Постановка завдання:
Розробити програму на мові програмування C# у середовищі розробки Visual Studio 2013 (або вище), яка буде працювати у віконному режимі та дозволяти:

1. Розв'язати, першу задану за варіантом, систему лінійних алгебраїчних рівнянь двома заданими прямими методами та знайти матрицю A^{-1} і визначник матриці A, тобто $det\,A$ за допомогою формул, які випливають з відповідного метода.

2. Розв'язати, другу задану за варіантом, симетричну систему лінійних алгебраїчних рівнянь методом квадратних коренів (методом Холецького) і ще будь-яким одним прямим методом та знайти A^{-1} і $det\,A$ за допомогою формул, які випливають з метода квадратних коренів

3. Розв'язати, третю задану за варіантом, стрічкову систему лінійних алгебраїчних рівнянь методом прогонки і ще будь-яким одним прямим методом та знайти $det\,A$ за допомогою формули, яка випливає з метода прогонки.

4. Пункти 1-3 виконати двома способами: у Visual Studio 2005 (або вище) у віконному режимі на мові програмування C# та в одному з математичних пакетів (MatLab 7.0 / MathCAD 15.0 / Mathematica 7.0 / Maple 14.0 (або вище). Реалізуючи поставлені задачі на C# не дозволяється використовувати спеціальні функції для обробки матриць та векторів, тобто необхідно реалізувати розв'язання СЛАР шляхом покоординатних обчислень. Реалізуючи поставлені задачі у MatLab/MathCAD/Mathematica/ Maple дозволяється використовувати будь-які векторноматричні оператори, але таким чином, щоб завдання було виконане запропонованим за варіантом методом.

5. При виконанні завдань сформульованих в п.4 провести заміри часу роботи алгоритмів для кожного середовища розробки та звести отримані результати до таблиць. Обгрунтувати отримані результати та зробити висновки.

6. При вирішені задач поставлених в п.4 необхідно виводити на екран всі проміжні обчислення. Наприклад, для класичного методу Гаусса необхідно виводити на екран СЛАР кожного разу після виключення чергової змінної зі всіх рівнянь за індексом більших ніж поточне, тобто для СЛАР з дев’ятьма невідомими необхідно вивести 9 проміжних матриць. Аналогічним чином необхідно робити і для інших методів.

Варіант - 18.

СЛАР 1.

	-68	-18	95	-91	-55	67	-64	1	-63		-54
	60	35	-73	84	66	-5	87	24	-43		-98
	-96	15	-85	77	-91	-38	75	-31	50		-54
	-74	70	-42	-48	96	12	-41	-23	84		-43
A =	-42	-64	47	-61	12	66	15	95	-89	**b** =	88
	-65	-24	-33	63	-91	39	-22	-50	41		89
	40	-25	-32	-75	-93	67	-74	-21	-42		-52
	-87	-60	-22	25	51	-94	5	59	15		69
	-20	-72	88	79	92	52	8	-79	-36		-48

СЛАР 2.

	106	59	8	70	6	19	49	6	13	44		154
	59	152	33	111	61	96	8	78	70	83		13
	8	33	104	20	21	66	45	6	54	81		64
	70	111	20	148	17	109	36	63	22	32		37
A =	6	61	21	17	49	40	7	25	58	49	**b** =	125
	19	96	66	109	40	133	47	59	60	55		27
	49	8	45	36	7	47	152	12	39	46		10
	6	78	6	63	25	59	12	123	52	30		91
	13	70	54	22	58	60	39	52	111	84		136
	44	83	81	32	49	55	46	30	84	144		60

СЛАР 3.

	150	-52	0	0	0	0	0	0	0	0		-832
	19	-87	33	0	0	0	0	0	0	0		-673
	0	-212	442	77	0	0	0	0	0	0		-352
	0	0	-8	-123	-47	0	0	0	0	0		-397
	0	0	0	-110	630	-160	0	0	0	0		-977
A =	0	0	0	0	-39	-315	15	0	0	0	**b** =	80
	0	0	0	0	0	-13	124	-13	0	0		-810
	0	0	0	0	0	0	5	31	14	0		-707
	0	0	0	0	0	0	0	-44	445	148		262
	0	0	0	0	0	0	0	0	-643	-732		719

Методи: класичний м. Гауса, м. відбиття на основі LQ-факторизації.

Математичне підгрунтя

У сучасному світі більшість прикладних задач можна звести до багатовимірних нелінійних рівнянь, що вирішуються шляхом лінеаризації. Під системою лінійних алгебраїчних рівнянь nxn мається на увазі система вигляду:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = a_{1n+1}, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = a_{2n+1}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = a_{nn+1} \end{cases}$$

Важливо.

1. Для вищеописаної СЛАР буде існувати єдине рішення, якщо детермінант матриці коефіцієнтів відмінний від нуля (невироджена матриця).
2. Обернена матриця існує лише для невироджених матриць (детермінант не може бути 0).

Обернена матриця.

Обернену матрицю можемо знайти використовуючи наступну схему

$$AA^{-1} = E \Longleftrightarrow \underbrace{\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}}_{A} \begin{pmatrix} a_{11}^{-1} & a_{12}^{-1} & \cdots & a_{1n}^{-1} \\ a_{21}^{-1} & a_{22}^{-1} & \cdots & a_{2n}^{-1} \\ \cdots & \cdots & \cdots & \cdots \\ \underbrace{a_{n1}^{-1}}_{A_1^{-1}} & \underbrace{a_{n2}^{-1}}_{A_2^{-1}} & \cdots & \underbrace{a_{nn}^{-1}}_{A_n^{-1}} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ \underbrace{0}_{E_1} & \underbrace{0}_{E_2} & \cdots & \underbrace{1}_{E_n} \end{pmatrix}$$

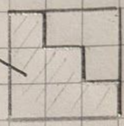
І розв'язуємо за допомогою обраного методу $AA_i^{-1} = E_i$, $i = 1...n$.
Розв'язками будуть відповідні стовпці оберненої матриці.

Класичний метод Гауса.

- Прямий хід:
    a. Опишемо доповнену матрицю $(A|b)$.
    b. Визначимо номер кроку k, $k = 1...n$.
    c. Переобчислимо значення матриці $(A|b)$, де $i = k + 1...n$, $j = k...n + 1$:
        - для першого рядка:
          $$a_{kj}^{(k)} = \frac{a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}}, \; k = 1, 2, ..., n.$$
        - для інших рядків:
          $$a_{ij}^{(k)} = a_{ij}^{(k-1)} - a_{kj}^{(k)} \bullet a_{ik}^{(k-i)}, \; k = 1, 2, ..., n.$$
    d. Якщо $k \, ! = n$ повертаємось до кроку *c* з $k = k + 1$.
- Зворотній хід:
  $$x_i = a_{tn+1}^{(t)} - \sum_{j=t+1}^{n} a_{tj}^{(t)} x_j .$$

Визначник. Визначник трикутної матриці дорівнює сумі всіх її діагональних елементів.

Метод відбиття на основі LQ-факторизації.

L 

$$A = LQ$$
$$L = A \cdot Q^T$$
$$L = A \cdot H_1 \cdot H_2 \cdot \ldots \cdot H_{n-2} \cdot H_{n-1}$$

$$H_i = I - 2\bar{\omega}_i \bar{\omega}_i^T$$
$$\bar{\omega}_i^T = \mu_i \left(0; \ldots; 0; a_{ii}^{(i-1)} - \beta_i; a_{i,i+1}^{(i-1)}; a_{in}^{(i-1)}\right)$$
$$\mu_i = \frac{1}{\sqrt{2\beta_i^2 - 2\beta_i a_{ii}^{(i-1)}}}$$
$$\beta_i = \operatorname{sign}\left(-a_{ii}^{(i-1)}\right)\sqrt{\sum_{k=i}^{n}\left(a_{ik}^{(i-1)}\right)^2}$$

$$A\bar{x} = \bar{b}$$
$$LQ\bar{x} = \bar{b}$$
$$L^{-1}LQ\bar{x} = \underbrace{L^{-1}\bar{b}}_{\bar{y}}$$
$$\begin{cases} \bar{y} = L^{-1}\bar{b} \\ Q\bar{x} = \bar{y} \end{cases} \Rightarrow \begin{cases} L\bar{y} = L \cdot L^{-1}\bar{b} \\ Q^T Q\bar{x} = Q^T y \end{cases} \Rightarrow$$
$$\Rightarrow \begin{cases} L\bar{y} = \bar{b} \\ Q^T\bar{y} = \bar{x} \end{cases}$$

Визначник.

$$A = LQ$$
$$\det A = \det L \cdot \det Q$$
$$I = QQ^T$$
$$\det I = \det Q \cdot \underbrace{\det Q^T}_{\det Q}$$
$$1 = (\det Q)^2 \Rightarrow \det Q = \pm 1$$
$$\det A = \det Q \det L = \pm \det L = \pm r_{11} r_{22} \ldots r_{nn}$$

Метод квадратних коренів (Холецького).

Матриця коефіцієнтів є симетричною. Для неї виконуємо розкладання:

$$A = U^T \cdot U \text{, где } U = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{pmatrix}, U^T = \begin{pmatrix} u_{11} & 0 & \cdots & 0 \\ u_{12} & u_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ u_{1n} & u_{2n} & \cdots & u_{nn} \end{pmatrix}.$$

Система матиме наступний вигляд:

$$\begin{cases} u_{11}^2 = a_{11}, \quad u_{11} \cdot u_{12} = a_{12}, \quad \ldots, \quad u_{11} \cdot u_{1n} = a_{1n}\,; \\ u_{12}^2 + u_{22}^2 = a_{22}, \quad \ldots, \quad u_{12} \cdot u_{1n} + u_{22} \cdot u_{2n} = a_{2n}\,; \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ u_{1n}^2 + u_{2n}^2 + \ldots + u_{nn}^2 = a_{nn}\,. \end{cases}$$

Трикутну матрицю знаходимо за формулами:

$$u_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{n-1} u_{ki}^2}\,, \quad i = 1, 2, \ldots, n;$$

$$u_{ij} = \frac{1}{u_{ii}} \left(a_{ij} - \sum_{k=1}^{i-1} u_{ki} u_{kj} \right), \quad j = 2, 3, \ldots, n;\ j > i; \quad u_{ij} = 0\ (j < i).$$

На основі розкладу можемо обчислити корені, обчисливши дві трикутні системи.

$$Ax = b \Rightarrow UU^T x = b \Rightarrow Uy = b; U^T x = y \Rightarrow x$$

Метод прогонки.

Метод застосовується для тридіагональних матриць

$$\begin{pmatrix} -\beta_1 & \gamma_1 & 0 & 0 & \cdots & 0 & 0 & \delta_1 \\ \alpha_2 & -\beta_2 & \gamma_2 & 0 & \cdots & 0 & 0 & \delta_2 \\ 0 & \alpha_3 & -\beta_3 & \gamma_3 & \cdots & 0 & 0 & \delta_3 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \cdots & \cdots & \alpha_n & -\beta_n & \delta_n \end{pmatrix}$$

При застосуванні прямого ходу метода Гауса будемо мати

$$\begin{pmatrix} 1 & -P_1 & 0 & 0 & \cdots & 0 & Q_1 \\ 0 & 1 & -P_2 & 0 & \cdots & 0 & Q_2 \\ 0 & 0 & 1 & -P_3 & \cdots & 0 & Q_3 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & Q_n \end{pmatrix}$$

Матриця буде мати наступні розв’язки

$$x_i = P_i x_{i+1} + Q_i\,,\ i = 1, 2, ..., n-1$$

Далі на основі формул прямого ходу метода Гауса можемо вивести наступні рекурентні формули пошуку необхідних елементів

$$P_i = \frac{\gamma_i}{\beta_i - \alpha_i P_{i-1}}, \quad Q_i = \frac{\alpha_i Q_{i-1} - \delta_i}{\beta_i - \alpha_i P_{i-1}}, \quad i = 1, 2, \ldots, n-1.$$

Значення коренів

СЛАР 1.

Метод	C#	MatLab
Гаус	0,5289303878900076 -1,0077316927167819 0,7426730955936555 0,8939511211267733 -0,36731396310218434 1,057488115479492 -0,4531186881345075 2,267449722363134 2,3444018035231773	0.528930387890012 -1.007731692716795 0.742673095593643 0.8939511211267535 -0.3673139631021816 1.057488115479499 -0.4531186881344929 2.267449722363111 2.344401803523175
LQ	0,5289303878900126 -1,0077316927167903 0,742673095593647 0,8939511211267577 -0,367313963102183 1,0574881154794988 -0,45311868813449685 2,2674497223631187 2,3444018035231773	0.5289303878900129 -1.007731692716793 0.7426730955936468 0.8939511211267572 -0.3673139631021831 1.0574881154795 -0.4531186881344957 2.267449722363118 2.344401803523178

СЛАР 2.

Метод	C#	MatLab
Квадратних коренів (Холецького)	-521,2707267052261 -1200,8788918845328 934,4346273044324 2089,799799309536 2210,111024211383 -1860,1928530664227 3,763936836595779 239,78210342895972 -246,10030324225787 -87,31215542239613	-521.2707267051953 -1200.878891884462 934.4346273043776 2089.799799309413 2210.111024211253 -1860.192853066314 3.763936836595331 239.7821034289455 -246.1003032422428 -87.31215542239097
Гаус	-521,270726704817 -1200,8788918836149 934,4346273037156 2089,799799307924 2210,1110242096925 -1860,1928530649868 3,76393683659019 239,78210342877938 -246,10030324207227 -87,31215542233102	-521.2707267048706 -1200.878891883734 934.4346273038103 2089.799799308134 2210.111024209915 -1860.192853065175 3.763936836591228 239.7821034288033 -246.1003032420971 -87.31215542234041

СЛАР 3.

Метод	C#	MatLab
Прогонки	-2,7483917851659942 8,071946773559633 2,469054945995211 3,479615972269602 -1,0796851948323993 -0,5372464355879242 -8,756023320577313 -20,674052929918748 -1,594588754973731 0,4184707232897665	-2.748392344661471 8.071945159630371 2.46905101322455 3.47963410384919 -1.079731976154187 -0.5374431025034282 -8.760274957239544 -20.71441033578146 -1.503707486041215 0.1332147061753824
LQ	-2,748391785165994 8,071946773559635 2,469054945995209 3,4796159722696016 -1,079685194832399 -0,5372464355879243 -8,756023320577313 -20,67405292991875 -1,594588754973731 0,41847072328976753	-2.748391785165993 8.071946773559633 2.469054945995209 3.479615972269603 -1.079685194832399 -0.5372464355879242 -8.756023320577313 -20.67405292991875 -1.594588754973731 0.418470723289768

Час роботи алгоритмів

СЛАР 1.

Метод	C#	MatLab
Гаус	5 ms	2.36583 ms
LQ	9 ms	5.21302 ms

СЛАР 2.

Метод	C#	MatLab
Квадратних коренів (Холецького)	<1ms	0.0779629 ms
Гаус	7 ms	3.03388 ms

СЛАР 3.

Метод	C#	MatLab
Прогонки	<1ms	0.274181 ms
LQ	5 ms	2.36893 ms

Висновки

Методи, що були реалізовані в MatLab, дали більш точний результат, аніж реалізовані методи іншою мовою програмування в лабораторній.

Вимірами часу підтвердили думку, що спеціалізовані методи для деяких типів матриць швидше виконують обрахунки, аніж запрограмовані методи на мові C#.