Міністерство освіти і науки України
Державний вищий навчальний заклад
«Миколаївський політехнічний коледж»

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ

щодо виконання лабораторних робіт з дисципліни «Чисельні методи»

Укладач __Новожилова О.В., викладач , І категорія______
(П.І.Б., посада, категорія, пед. звання)

Розглянуто і затверджено на засіданні
предметної (циклової) комісії
Інженерія програмного забезпечення

Протокол № ______ від _______________ 20___ р.

Голова циклової комісії Плоткіна О.Г., викладач, вища категорія

Миколаїв, 2018

Укладач: Новожилова О.В., викладач, І категорія

Рецензент: Кузьма К.Т., доктор філософії в галузі інформаційних технологій, доцент кафедри комп'ютерної інженерії МНУ ім.В.О.Сухомлинського

Методичні рекомендації щодо виконання лабораторних робіт з дисципліни «Чисельні методи»/ Уклад. О.В. Новожилова. ДВНЗ Миколаївський політехнічний коледж, Миколаїв, 2018.- 75с.

Методична розробка призначена для здобувачів освіти спеціальності 121 «Інженерія програмного забезпечення» ДВНЗ «Миколаївський політехнічний коледж» для проведення лабораторних робіт з дисципліни "Чисельні методи".

Методична розробка призначена для внутрішнього використання при проведенні лабораторних робіт з дисципліни "Чисельні методи".

Розглянуто і схвалено
на засіданні методичної ради
інформатики та програмування

Протокол №___ від «__»______2018
Голова методоб'єднання викладачів
з інформатики та програмування
ВНЗ І-ІІ р.а. Миколаївської області

________________В.О.Круковська

Розглянуто і схвалено на засіданні методичної ради
Миколаївського політехнічного коледжу
Протокол №___від ________20___р.

ЗМІСТ

Вступ

Посібник призначений для студентів спеціальності 121 «Розробка програмного забезпечення» для проведення лабораторних робіт з дисципліни “Чисельні методи”.

Мета розробки – підготовка здобувачів освіти в області використання обчислювальної техніки для засвоєння методів, які найчастіше застосовуються на практиці при розв’язуванні стандартних задач обчислювальної математики, вивченні програмної реалізації їх, розширення досвіду практичної роботи на сучасних ЕОМ, розвиток у здобувачів освіти навичок логічного і алгоритмічного мислення, вивчення основ і методики рішення задач прикладної математики з наближеним обчисленням та числовими методами математичного аналізу.

В даній методичній розробці розглянуто чисельні методи розв’язування основних задач обчислювальної математики, а також питання проведення математичних розрахунків за допомогою мов програмування високого рівня. Подано методичні вказівки до використання цих методів, запитання для самоперевірки з метою кращого засвоєння теоретичного матеріалу, тексти підпрограм з прикладами їх використання.

Лабораторна робота

Алгебра матриць

М е т а   р о б о т и :   1. Практичне дослідження основних операцій над матрицями.

2. Отримань практичних навиків роботи з системою Turbo Pascal 7.0, закріплення знань щодо використання функцій.

Зміст звіту

1. Постановка завдання. Початкові дані.
2. Аналіз рішення задачі. Алгоритм рішення (блок – схема алгоритму).
3. Текст програми.
4. Результат виконання програми.
5. Висновки по роботі.

Короткі теоретичні відомості

Матрицею називається прямокутна таблиця з чисел, що містить деяку кількість *m* рядків і деяку кількість *n* стовпців. Числа *m* і *n* називаються порядками матриці. У випадку, якщо $m = n$, матриця називається квадратною, а число $m = n$ — її порядком.

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \qquad (1.1)$$

Основні операції над матрицями і їх властивості.

Складання матриць. Сумою двох матриць $A = \| a_{ij} \|$, де $(i = 1, 2 ..., m,\ j=1, 2 ..., n)$ і $B = \| b_{ij} \|$, де $(i = 1, 2 ..., m,\ j=1, 2 ..., n)$ одних і тих же порядків *m* і *n* називається матриця $C = \| c_{ij} \|$ $(i =1,2, ..., m;\ j = 1, 2 ...., n)$ тих же порядків *m* і *n*, елементи c_{ij} якою визначаються по формулі

$$c_{i\,j} = a_{i\,j} + b_{i\,j} \text{ де } (i = 1, 2 ..., m,\ j=1, 2 ..., n) \qquad (1.2)$$

Для позначення суми двох матриць використовується запис $C = A + B$. *Операція* складання суми матриць називається їх складанням. Отже, за визначенням:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} + \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{pmatrix} =$$

$$\begin{pmatrix} (a_{11}+b_{11}) & (a_{12}+b_{12}) & \dots & (a_{1n}+b_{1n}) \\ (a_{21}+b_{21}) & (a_{22}+b_{22}) & \dots & (a_{2n}+b_{2n}) \\ \dots & \dots & \dots & \dots \\ (a_{m1}+b_{m1}) & (a_{m2}+b_{m2}) & \dots & (a_{mn}+b_{mn}) \end{pmatrix}$$

Множення матриці на число. Множення матриці $A = \| a_{ij} \|$, де (i = 1, 2 ..., m, j=1, 2 ..., n) на дійсне число, називається матриця $C = \| c_{ij} \|$ *(i =1,2, ..., m; j = 1, 2 ...., n)*, елементи якої визначаються по формулі:

$$c_{ij} = \lambda\ a_{ij}, \text{ де } (i = 1, 2 \dots, m,\ j=1, 2 \dots, n) \qquad (1.3)$$

Для позначення множення матриці на число використовується запис *C = λ A або C = A λ*. Операція складання множення матриці на число називається множенням матриці на це число.

Зауваження. Різницею двох матриць *A* і *B* однакових порядків *m* і *n* природно назвати таку матрицю *C* тих же порядків *m* і *n,* яка в сумі з матрицею *B* дає матрицю А. Для позначення різниці двох матриць використовується природний запис: *C* = *A* — *B*.

Добуток матриць. Множенням матриці $A = \| a_{ij} \|$, *де (i = 1, 2 ..., m, j = 1, 2 ..., n)* що має порядки, відповідно рівні *m* і *n,* на матрицю $B = \| b_{ij} \|$, де *(i = 1, 2 ..., n, j=1, 2 ..., p),* що має порядки, відповідно рівні *n* і *p,* називається матриця $C = \| c_{ij} \|$ *(i =1,2, ..., m; j = 1, 2 ...., p)*, що має порядки, відповідно рівні *m* і *p* елементи якої визначаються по формулі:

$$c_{ij} = \sum_{k=1}^{n} a_{ik}\, b_{kj}, \text{ де } (i = 1, 2 \dots, m,\ j = 1, 2 \dots, p) \qquad (1.4)$$

Для позначення множення матриці A на матрицю B використовують запис $C = A \cdot B$.

Операція складання множення матриці A на матрицю B називається перемножуванням цих матриць.

Із сформульованого вище визначення витікає, що *матрицю A можна помножити не на всяку матрицю B,* необхідно, щоб число стовпців матриці A було рівне числу рядків матриці B.

Формула (1.4) є правилом складання елементів матриці C, матриці, що є множенням, A на матрицю B. Це правило можна сформулювати і словесно: *елемент $c_{i\ j}$ стоїть на перетині i-го рядка і j-го стовпця матриці C = A B, рівний сумі попарних множень відповідних елементів i-го рядка матриці A і j-го стовпця матриці B.*

Як приклад застосування вказаного правила приведемо формулу перемножування квадратних матриць другого порядку.

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} (a_{11}b_{11} + a_{12}b_{21}) & (a_{11}b_{12} + a_{12}b_{22}) \\ (a_{21}b_{11} + a_{22}b_{21}) & (a_{21}b_{12} + a_{22}b_{22}) \end{pmatrix}$$

Поняття визначника.

Розглянемо довільну квадратну матрицю будь-якого порядку *n:*

$$\mathrm{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \qquad (1.5)$$

З кожною такою матрицею зв'яжемо цілком певну чисельну характеристику, звану визначником, відповідним цій матриці.

Якщо порядок n матриці (1.5) рівний одиниці, то ця матриця складається з одного елемента a_{ij} визначником першого порядку відповідним такій матриці, ми назвемо величину цього елементу.

Якщо далі порядок n матриці (1.5) рівний двом, тобто якщо ця матриця має вигляд

$$\mathrm{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \qquad (1.6)$$

то визначником другого порядку, відповідним такій матриці, назвемо число, рівне *a11 a22, — a12 a21* і що позначається одним з символів:

$$\Delta = \det A = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$$

Отже, за визначенням

$$\Delta = \det A = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21} \qquad (1.7)$$

Формула (1.7) є правилом складання визначника другого порядку по елементах відповідної йому матриці. Словесне формулювання цього правила таке: визначник другого порядку, відповідний матриці (1.6), рівний різниці множення елементів, що стоять на головній діагоналі цієї матриці, і множення елементів, що стоять на побічній її діагоналі. Визначники другого і вищих порядків знаходять широке застосування при вирішенні систем лінійних рівнянь.

Мінором деякого елементу визначника називається визначник, отриманий з даного, викреслюванням рядка і стовпця, на яких знаходиться цей елемент.

Покажемо на прикладі, що є мінор визначника:

Алгебраїчним доповненням деякого елементу визначника називається його мінор із знаком плюс або мінус залежно від того, парна або непарна сума рядка і стовпця цього елементу.

Таким чином, мінор елементу a_{ij}, є наступний вираз:

$$A_{ij} = (-1)^{i+j} \cdot M_{ij}$$

Наприклад, доповнення алгебри елементу *a21* визначника, буде таким:

$$A_{21} = (-1)^{2+1} \cdot M_{21} = -M_{21}$$

Мінор і алгебраїчне доповнення визначників потрібні для того, щоб показати універсальний метод обчислення значень визначників будь-яких порядків.

Отже, визначник рівний сумі множень всіх елементів якого-небудь ряду (рядка або стовпця) на відповідні доповнення алгебри.

Наприклад, знайдемо визначника за допомогою розкладання другого стовпця:

$$\Delta = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{12}A_{12} + a_{22}A_{22} + a_{32}A_{32}$$

От так виглядатиме розкладання визначника по третьому рядку:

$$\Delta = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{31}A_{31} + a_{32}A_{32} + a_{33}A_{33}$$

У більшості визначників ми можемо спростити обчислення визначників, використовуючи показані вище властивості визначників.

Зворотною матрицею $B=A^{-1}$ по відношенню до матриці A, називається така матриця, коли $A \bullet B = B \bullet A = E$, де E – одинична матриця. Для того, щоб знайти зворотну матрицю даною, потрібно розділити алгебраїчні доповнення даного елементу на визначник даної матриці, і транспонувати отриману матрицю.

$$B = A^{-1} = \frac{A_{ij}}{|A|} = \begin{vmatrix} \frac{A_{11}}{|A|} & \frac{A_{21}}{|A|} & \cdots & \frac{A_{m1}}{|A|} \\ \frac{A_{12}}{|A|} & \frac{A_{22}}{|A|} & \cdots & \frac{A_{m2}}{|A|} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{A_{1n}}{|A|} & \frac{A_{2n}}{|A|} & \cdots & \frac{A_{mn}}{|A|} \end{vmatrix}$$

Aij - це алгебраїчне доповнення елементу aij матриці A,

$|A|$ - визначник матриці A.

Зворотну матрицю можна знайти тільки для невиродженої матриці. Для виродженої матриці це неможливо, тому як відбувається ділення на нуль.

Невиродженою матрицею називається така матриця, чисельне значення визначника якої не рівне нулю.

Завдання.

Розробити програмну реалізацію алгебри матриць, яка включає такі основні операції над матрицями, як сума, різниця, добуток матриць; задача знаходження алгебраїчних доповнень, визначників та обернених матриць

При запуску програми користувачеві повинно пропонуватися меню з 5-ти пунктів, що включають чотири основні операції алгебри матриць: складання, віднімання, множення матриць, отримання зворотної матриці; а також вихід з програми.

При складанні і відніманні матриць необхідно, щоб їх розмірності співпадали. При множенні матриць необхідною умовою є рівність стовпців матриці A і рядків матриці B. При обернені матриці необхідно, щоб матриця A була квадратною. Якщо дані умови не виконуються, програма повинна видати відповідне попередження і закінчити роботу.

Отримані результати звірити з тестовими розрахунками.

Розробка алгоритмів програми

1 Розробка алгоритмів основних операцій алгебри матриць

Блок-схеми алгоритмів основних операцій алгебри матриць зображені на рисунках 2.1 – 2.8.

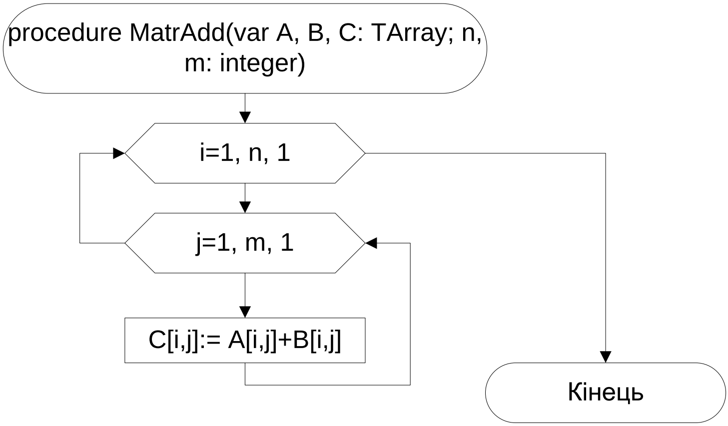


Рисунок 2.1 – Блок-схема алгоритму добутку двох матриць

(A, B – задані матриці; C – вихідна матриця; n, m – розмірність матриць)

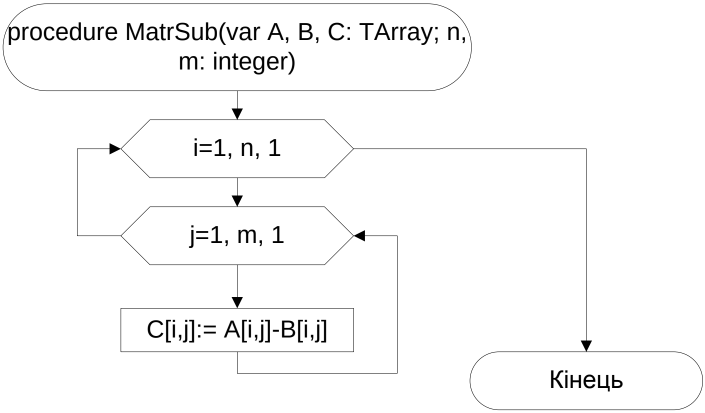


Рисунок 2.2 – Блок-схема алгоритму різниці двох матриць

(A, B – задані матриці; C – вихідна матриця; n, m – розмірність матриць)

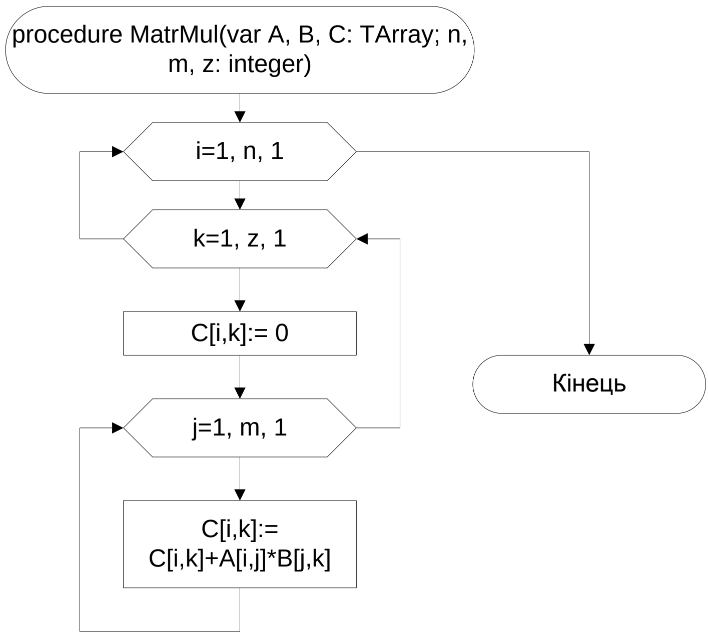


Рисунок 2.3 – Блок-схема алгоритму множення двох матриць (A, B – задані матриці; C – вихідна матриця; n, m – розмірність матриці A;

m, z – розмірність матриці B)

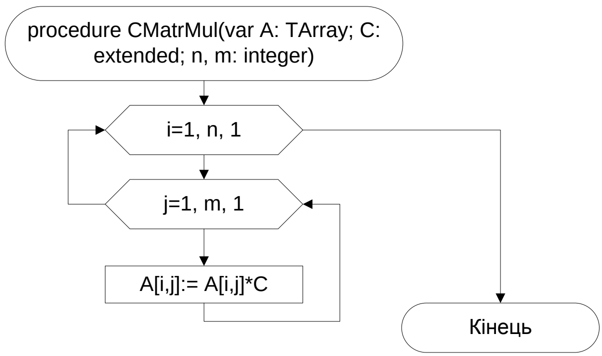


Рисунок 2.4 – Блок-схема алгоритму множення матриці на число

(A – задана матриця; C – число; n, m – розмірність матриці A)

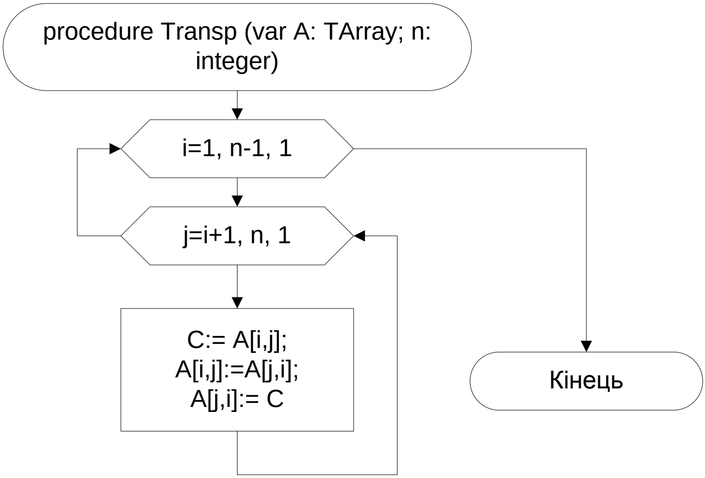


Рисунок 2.5 – Блок-схема алгоритму транспонування матриці
(A – задана матриця; n – розмірність матриці A)

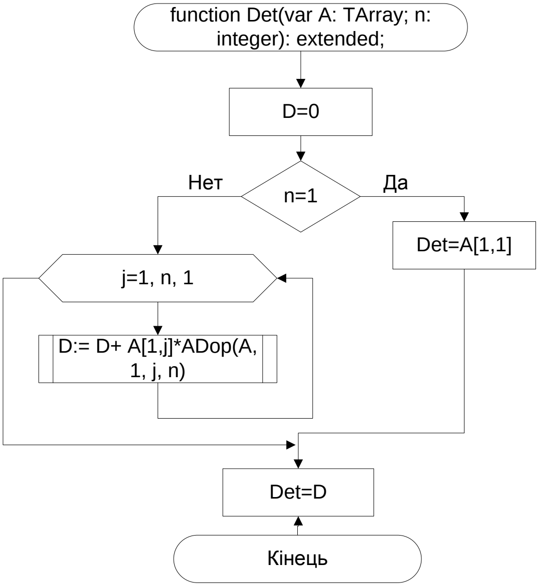


Рисунок 2.6 – Блок-схема знаходження визначника матриці
(A – задана матриця; n – розмірність матриці)

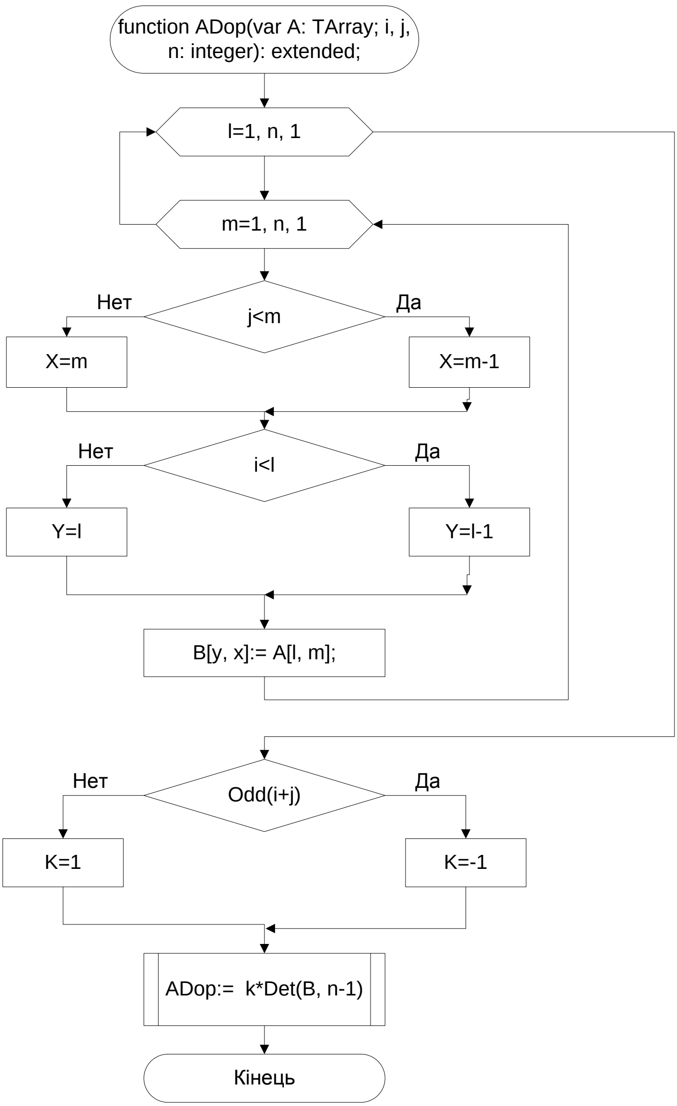


Рисунок 2.7 – Блок-схема алгоритму визначення алгебраїчного доповнення

(A – задана матриця; i, j – координати числа; n – розмірність матриці)

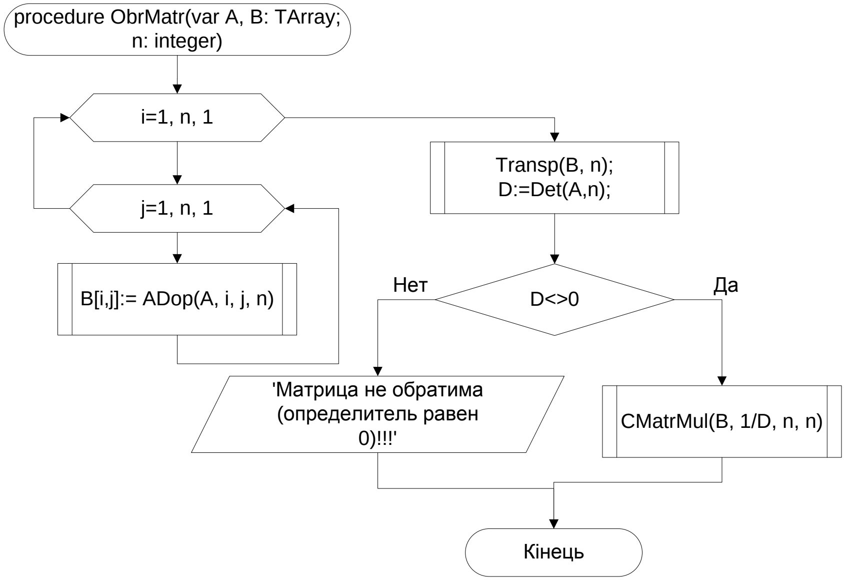


Рисунок 2.8 – Блок-схема алгоритму визначення оберненої матриці
(A – задана матриця; B – результуюча матриця; n – розмірність матриць)

2 Розробка допоміжних алгоритмів

Блок-схеми допоміжних алгоритмів приведені на рисунках 2.9 і 2.10.

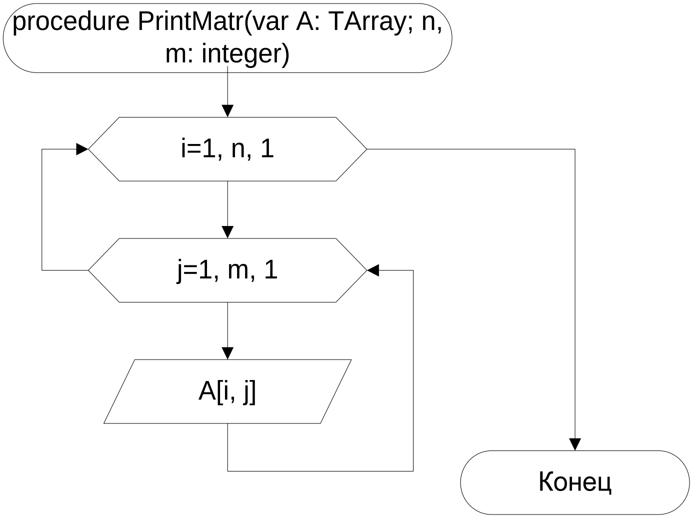


Рисунок 2.9 – Блок-схема алгоритму виводу елементів масиву A на екран

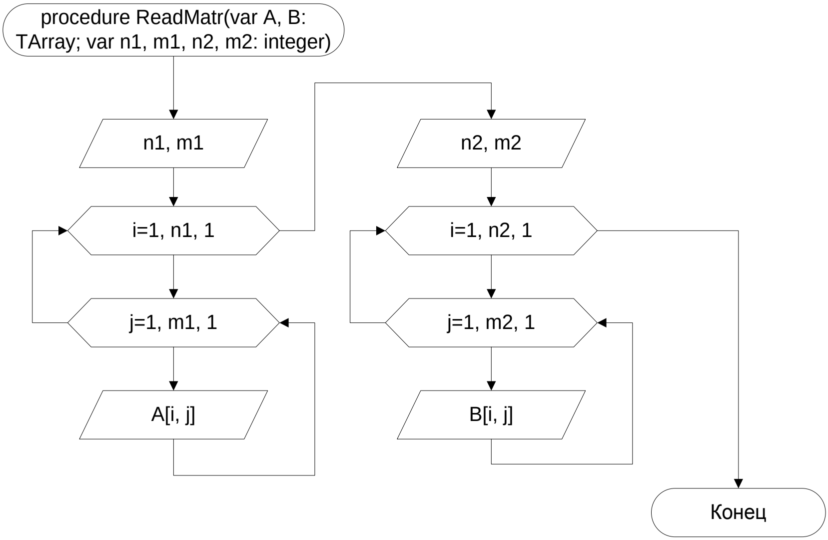


Рисунок 2.10 – Блок-схема алгоритму вводу елементів масивів з клавіатури

(A, B – масиви, що вводяться; n1, m1, n2, m2 – розмірності масивів A і B)

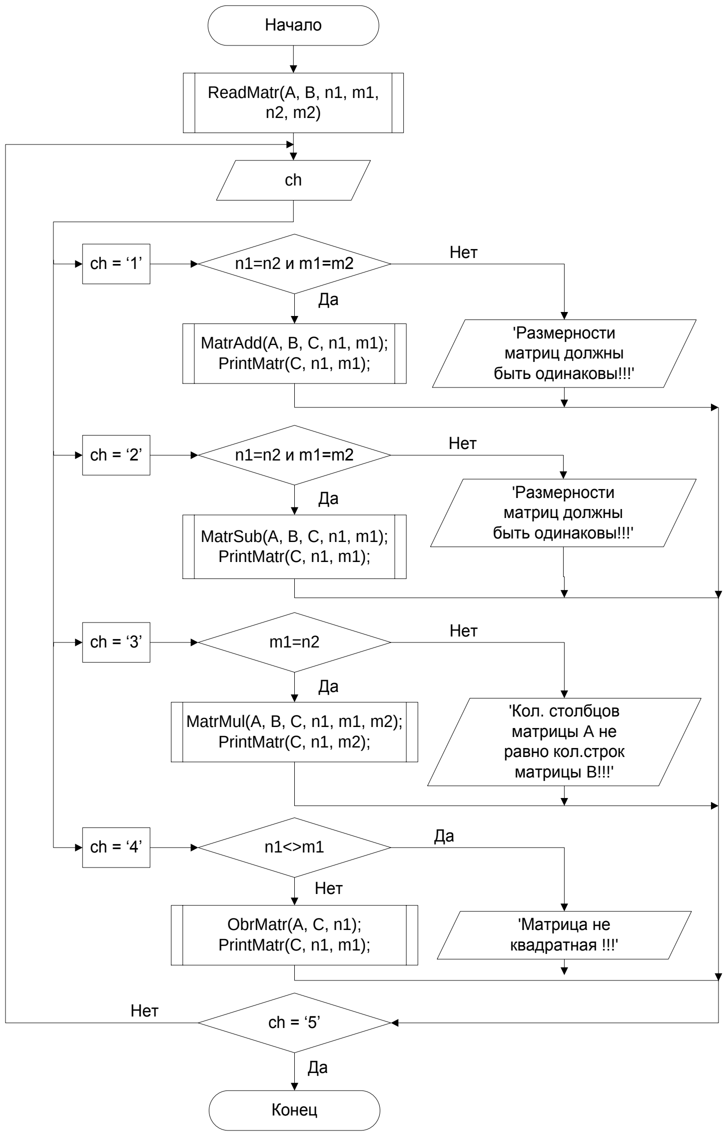


Рисунок 2.11 – Блок-схема основної програми

Контрольні питання

1. Основні операції над матрицями. Їх властивості
2. Дати визначення зворотної матриці та мінору.
3. Дати поняття визначника .
4. Дати визначення добутку матриць.

Лабораторна робота

Рішення систем лінійних рівнянь методом Гауса

Мета роботи: 1. Практичне дослідження методів чисельного вирішення лінійних рівнянь.
2. Отримань практичних навиків роботи з системою Turbo Pascal 7.0, закріплення знань щодо використання функцій.

Зміст звіту

1. Постановка завдання. Початкові дані.
2. Аналіз рішення задачі. Алгоритм рішення (блок – схема алгоритму).
3. Текст програми.
4. Результат виконання програми.
5. Висновки по роботі.

Постановка завдання

1. Використовуючи дані, знайти рішення СЛР методом Гауса з точністю до 0.0001.

2. Результатом роботи програми є значення коренів системи лінійних рівнянь.
3. За наслідками роботи зробити виводи про переваги і недоліки даного методу.

№ 1.

$$\begin{cases} 3{,}21x_1 - 4{,}25x_2 + 2{,}13x_3 = 5{,}06; \\ 7{,}09x_1 + 1{,}17x_2 - 2{,}23x_3 = 4{,}75; \\ 0{,}43x_1 - 1{,}4x_2 - 0{,}62x_3 = -1{,}05. \end{cases}$$

№ 2.

$$\begin{cases} 0{,}42x_1 - 1{,}13x_2 + 7{,}05x_3 = 6{,}15; \\ 1{,}14x_1 - 2{,}15x_2 + 5{,}11x_3 = -4{,}16; \\ -0{,}71x_1 + 0{,}81x_2 - 0{,}02x_3 = -0{,}17. \end{cases}$$

№ 3.

$$\begin{cases} 2{,}5x_1 - 3{,}12x_2 - 4{,}03x_3 = -7{,}5; \\ 0{,}61x_1 + 0{,}71x_2 - 0{,}05x_3 = 0{,}44; \\ -1{,}03x_1 - 2{,}05x_2 + 0{,}877x_3 = -1{,}16. \end{cases}$$

№ 4.

$$\begin{cases} 7{,}09x_1 + 1{,}17x_2 - 2{,}23x_3 = -4{,}75; \\ 0{,}43x_1 - 1{,}4x_2 - 0{,}62x_3 = -1{,}05; \\ 3{,}21x_1 - 4{,}25x_2 + 2{,}13x_3 = 5{,}06. \end{cases}$$

№ 5.

$$\begin{cases} 1{,}14x_1 - 2{,}15x_2 - 5{,}11x_3 = -4{,}16; \\ -0{,}71x_1 + 0{,}81x_2 - 0{,}02x_3 = -0{,}17; \\ 0{,}42x_1 - 1{,}13x_2 + 7{,}05x_3 = 6{,}15. \end{cases}$$

№ 6.

$$\begin{cases} 0{,}61x_1 + 0{,}71x_2 - 0{,}05x_3 = 0{,}44; \\ -1{,}03x_1 - 2{,}05x_2 + 0{,}87x_3 = -1{,}16; \\ 2{,}5x_1 - 3{,}12x_2 - 5{,}03x_3 = -7{,}5. \end{cases}$$

№ 7.

$$\begin{cases} 3{,}11x_1 - 1{,}66x_2 - 0{,}60x_3 = -0{,}92; \\ -1{,}65x_1 + 3{,}51x_2 - 0{,}78x_3 = 2{,}57; \\ 0{,}60x_1 + 0{,}78x_2 - 1{,}87x_3 = 1{,}65. \end{cases}$$

№ 8.

$$\begin{cases} 0{,}10x_1 + 12x_2 - 0{,}13x_3 = 0{,}10; \\ 0{,}12x_1 + 0{,}71x_2 + 0{,}15x_3 = 0{,}26; \\ -0{,}13x_1 + 0{,}15x_2 + 0{,}63x_3 = 0{,}38. \end{cases}$$

№ 9.

$$\begin{cases} 0{,}71x_1 + 0{,}10x_2 + 0{,}12x_3 = 0{,}29; \\ 0{,}10x_1 + 0{,}34x_2 - 0{,}04x_3 = 0{,}32; \\ 0{,}12x_1 - 0{,}04x_2 + 0{,}10x_3 = -0{,}10. \end{cases}$$

№ 10.

$$\begin{cases} 0{,}34x_1 - 0{,}04x_2 + 0{,}10x_3 = 0{,}33; \\ -0{,}04x_1 + 0{,}10x_2 + 0{,}12x_3 = -0{,}05; \\ 0{,}10x_1 + 0{,}12x_2 + 0{,}71x_3 = 0{,}28. \end{cases}$$

№ 11.

$$\begin{cases} 0{,}12x_1 - 0{,}43x_2 + 0{,}14x_3 = -0{,}17; \\ -0{,}07x_1 + 0{,}34x_2 + 0{,}72x_3 = 0{,}62; \\ 1{,}18x_1 - 0{,}08x_2 - 0{,}25x_3 = 1{,}12. \end{cases}$$

№ 12.

$$\begin{cases} 1{,}17x_1 + 0{,}53x_2 - 0{,}84x_3 = 1{,}15; \\ 0{,}64x_1 - 0{,}72x_2 - 0{,}43x_3 = 0{,}15; \\ 0{,}32x_1 + 0{,}43x_2 - 0{,}93x_3 = -0{,}48. \end{cases}$$

№ 13.

$$\begin{cases} 0{,}66x_1-1{,}44x_2-0{,}18x_3=1{,}83; \\ 0{,}48x_1-0{,}24x_2+0{,}37x_3=-0{,}84; \\ 0{,}86x_1+0{,}43x_2+0{,}64x_3=0{,}64. \end{cases}$$

№ 14.

$$\begin{cases} 0{,}82x_1+0{,}43x_2-0{,}57x_3=0{,}48; \\ -0{,}35x_1+1{,}12x_2-0{,}48x_3=0{,}52; \\ 0{,}48x_1+0{,}23x_2+0{,}37x_3=1{,}44. \end{cases}$$

№ 15.

$$\begin{cases} 1{,}6x_1+0{,}12x_2+0{,}57x_3=0{,}18; \\ 0{,}38x_1+0{,}25x_2-54x_3=0{,}63; \\ 0{,}28x_1+0{,}46x_2-1{,}12x_3=0{,}88. \end{cases}$$

№ 16.

$$\begin{cases} 1{,}16x_1+1{,}3x_2-1{,}14x_3=0{,}43; \\ 0{,}83x_1-0{,}48x_2-2{,}44x_3=-0{,}15; \\ 2x_1-0{,}16x_2+1{,}3x_3=1{,}5. \end{cases}$$

№ 17.

$$\begin{cases} 0{,}10x_1-0{,}04x_2-0{,}13x_3=-0{,}15; \\ -0{,}04x_1+0{,}34x_2+0{,}05x_3=0{,}31; \\ -0{,}13x_1+0{,}05x_2+0{,}63x_3=0{,}37. \end{cases}$$

№ 18.

$$\begin{cases} 0{,}63x_1+0{,}05x_2+0{,}15x_3=0{,}34; \\ 0{,}05x_1+0{,}34x_2+0{,}10x_3=0{,}32; \\ 0{,}15x_1+0{,}10x_2+0{,}71x_3=0{,}42. \end{cases}$$

№ 19.

$$\begin{cases} 1{,}20x_1-0{,}20x_2+0{,}30x_3=-0{,}60; \\ -0{,}20x_1+1{,}60x_2-0{,}10x_3=0{,}30; \\ -0{,}30x_1+0{,}10x_2-1{,}5x_3=0{,}40. \end{cases}$$

№ 20.

$$\begin{cases} 0{,}30x_1+1{,}20x_2-0{,}20x_3=-0{,}60; \\ -0{,}10x_1-0{,}20x_2+1{,}60x_3=0{,}30; \\ -1{,}50x_1-0{,}30x_2+0{,}10x_3=40. \end{cases}$$

№ 21.

$$\begin{cases} 0{,}20x_1+0{,}44x_2+0{,}81x_3=0{,}74; \\ 0{,}58x_1-0{,}29x_2+0{,}05x_3=0{,}02; \\ 0{,}05x_1+0{,}34x_2+0{,}10x_3=0{,}32. \end{cases}$$

№ 22.

$$\begin{cases} 6{,}36x_1+11{,}75x_2+10x_3=-41{,}70; \\ 7{,}42x_1+19{,}03x_2+11{,}75x_3=-49{,}49; \\ 5{,}77x_1+7{,}42x_2+6{,}36x_3=-27{,}67. \end{cases}$$

№ 23.

$$\begin{cases} 0{,}40x_1+0{,}11x_2+0{,}18x_3=0{,}47; \\ 0{,}28x_1-0{,}59x_2+0{,}03x_3=0{,}01; \\ 0{,}02x_1+0{,}24x_2+0{,}10x_3=0{,}22. \end{cases}$$

№ 24.

$$\begin{cases} 0{,}18x_1+0{,}25x_2-0{,}44x_3=1{,}15; \\ 0{,}42x_1-0{,}35x_2+1{,}12x_3=0{,}86; \\ 1{,}14x_1+0{,}12x_2-0{,}83x_3=0{,}68. \end{cases}$$

№ 25.

$$\begin{cases} 1{,}2x_1+0{,}18x_2-0{,}42x_3=1{,}5; \\ 0{,}44x_1+0{,}36x_2+0{,}12x_3=1{,}16; \\ 0{,}36x_1-0{,}42x_2-0{,}22x_3=0{,}15. \end{cases}$$

№ 26.

$$\begin{cases} 0{,}64x_1-0{,}43x_2+0{,}57x_3=0{,}43; \\ 0{,}56x_1+0{,}12x_2-0{,}27x_3=0{,}88; \\ 0{,}63x_1-0{,}83x_2+0{,}43x_3=-0{,}12. \end{cases}$$

№ 27.

$$\begin{cases} 1{,}60x_1+2{,}18x_2-0{,}72x_3=1{,}15; \\ 0{,}43x_1-0{,}16x_2+0{,}53x_3=0{,}83; \\ 0{,}34x_1+0{,}57x_2-0{,}83x_3=-0{,}42. \end{cases}$$

№ 28.

$$\begin{cases} 0{,}8x_1-0{,}13x_2+0{,}63x_3=1{,}15; \\ 0{,}42x_1+0{,}57x_2+0{,}32x_3=0{,}84; \\ 0{,}54x_1+0{,}62x_2-0{,}32x_3=0{,}25. \end{cases}$$

№ 29.

$$\begin{cases} 1{,}06x_1-0{,}28x_2+0{,}84x_3=0{,}57; \\ 0{,}43x_1+0{,}62x_2-0{,}35x_3=0{,}66; \\ 0{,}37x_1-0{,}75x_2-0{,}64x_3=-0{,}38. \end{cases}$$

№ 30.

$$\begin{cases} 0{,}75x_1-0{,}84x_2+1{,}11x_3=0{,}66; \\ 1{,}12x_1-0{,}14x_2+0{,}45x_3=0{,}83; \\ 0{,}32x_1+0{,}23x_2-0{,}48x_3=0{,}14. \end{cases}$$

Короткі відомості з теорії

У основі методу лежить ідея послідовного виключення невідомих. Процес знаходження коріння методом Гауса складається з двох етапів. На першому етапі (прямий хід) система за допомогою послідовного виключення невідомих приводиться до трикутного вигляду. На другому етапі (зворотний хід) з системи трикутного вигляду послідовно в зворотному порядку знаходиться коріння.

Для простоти розглянемо систему 4-го порядку.

$$\begin{cases} a_{11}^{(0)}x_1 + a_{12}^{(0)}x_2 + a_{13}^{(0)}x_3 + a_{14}^{(0)}x_4 = b_1^{(0)} \\ a_{21}^{(0)}x_1 + a_{22}^{(0)}x_2 + a_{23}^{(0)}x_3 + a_{24}^{(0)}x_4 = b_2^{(0)} \\ a_{31}^{(0)}x_1 + a_{32}^{(0)}x_2 + a_{33}^{(0)}x_3 + a_{34}^{(0)}x_4 = b_3^{(0)} \\ a_{41}^{(0)}x_1 + a_{42}^{(0)}x_2 + a_{43}^{(0)}x_3 + a_{44}^{(0)}x_4 = b_4^{(0)} \end{cases} \qquad (1)$$

Прямий хід (1-й крок, до k = 1). Знаходимо з 1-го рівняння системи (1)

$x_1 * a^{(0)}_{11}$ - ведучий елемент першого рядка. Якщо $a^{(0)}_{11} \neq 0$, то

$$x_1 = -\underbrace{a_{12}^{(0)}/a_{11}^{(0)}}_{a_{12}^{(1)}} x_2 - \underbrace{a_{13}^{(0)}/a_{11}^{(0)}}_{a_{13}^{(1)}} x_3 - \underbrace{a_{14}^{(0)}/a_{11}^{(0)}}_{a_{14}^{(1)}} x_4 + \underbrace{b_1^{(0)}/a_{11}^{(0)}}_{b_1^{(1)}}$$

Обозначив $a_{12}^{(1)}$ $a_{13}^{(1)}$ $a_{14}^{(1)}$ $b_1^{(1)}$,

получим $x_1 = -a_{12}^{(1)}x_2 - a_{13}^{(1)}x_3 - a_{14}^{(1)}x_4 + b_1^{(1)}$, (2)

где b_1 **и** $a_{ij}^{(1)}$ **вычисляются по рекуррентным формулам:**

$$a_{kj}^{(k)} = a_{kj}^{(k-1)}/a_{kk}^{(k-1)}$$

$$b_k^{(k)} = b_k^{(k-1)}/a_{kk}^{(k-1)}$$

Підставимо вираз (2) для x_1 в 2-е, 3-е і 4-е рівняння системи (1).
В результаті отримаємо:

$$\begin{cases} a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + a_{24}^{(1)} = b_2^{(1)} \\ a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 + a_{34}^{(1)} = b_3^{(1)} \\ a_{42}^{(1)}x_2 + a_{43}^{(1)}x_3 + a_{44}^{(1)} = b_4^{(1)} \end{cases} \qquad (4)$$

$a_{ij}^{(1)}, b_i^{(1)}$ **вычисляются по рекуррентным формулам:**

$$\begin{aligned} a_{ij}^{(k)} &= a_{ij}^{(k-1)} - a_{ik}^{(k-1)}a_{kj}^{(k)} \\ b_i^{(k)} &= b_i^{(k-1)} - a_{ik}^{(k-1)}b_k^{(k)} \end{aligned} \qquad i = \overline{(k+1), n} \qquad (5)$$

В результаті виконання всіх кроків прямого ходу система (1) приводиться до системи трикутного вигляду, яка виходить об'єднанням всіх виразів для
X_1 X_2 ..., X_n.
Наша система приймає вигляд:

$$\begin{cases} x_1 = -a_{12}^{(1)}x_2 - a_{13}^{(1)}x_3 - a_{14}^{(1)}x_4 + b_1^{(1)} \\ x_2 = -a_{23}^{(2)}x_3 - a_{24}^{(2)}x_4 + b_2^{(2)} \\ x_3 = -a_{34}^{(3)}x_4 + b_3^{(3)} \\ x_4 = b_4^{(4)} \end{cases} \qquad (6)$$

где $b_4^{(4)} = b_4^{(3)} / a_{44}^{(3)}$

Обчислення по всіх кроках прямого ходу при побудові алгоритму організовуємо в циклі $\overline{\text{по } k = }$ 1(n-1). Останній, n-й крок прямого ходу виводиться з циклу, оскільки на цьому кроці не потрібні обчислення щ і Ь*, а реалізується тільки одне вираз xn=bn/aim. На кожному кроці прямого ходу всі перетворення коефіцієнтів і вільних членів проводяться по рекурентних формулах (3) і (5).

Зворотний хід.

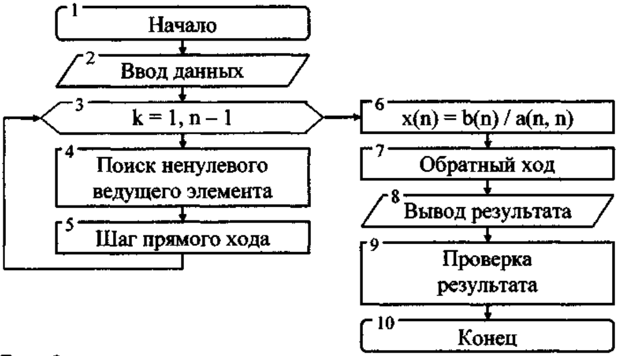
В процесі зворотного ходу з системи (6) невідомі знаходяться в зворотному порядку (цикл по i = (n- 1),1,-1).
Використовуємо рекурентну формулу:

$$b_i = b_i - x_j a_{ij}$$
$$x_i = b_i.$$

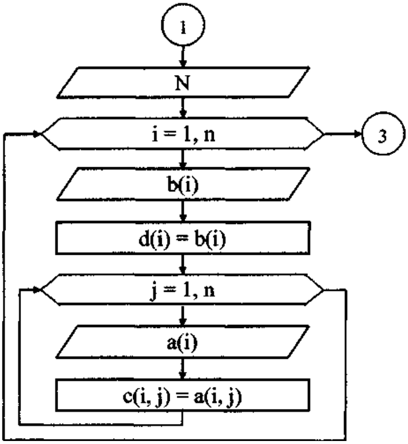
Пошук ненульового провідного елементу.

Розглянутий простий метод Гауса називається схемою єдиного ділення. Він володіє наступним недоліком: якщо провідний елемент якого-небудь рядка (a_{ij}) виявиться рівним 0, то цей метод формально непридатний, хоча система може мати рішення. Тому в схемі алгоритму необхідно додати блок пошуку ненульового провідного елементу. Суть пошуку наступна: якщо в якому-небудь рядку є нульовий провідний елемент a_{kk}, то в цьому ж k-м стовпці в циклі по
$k = \overline{(k+1),n}$ здійснюється пошук ненульового елементу. Якщо в рядку kn цього стовпця такий елемент знайдений, то рядки до і kn міняються місцями. Для цього використовуємо робочий осередок R.

Блок-схема алгоритма.



Блок 2 – ввод данных



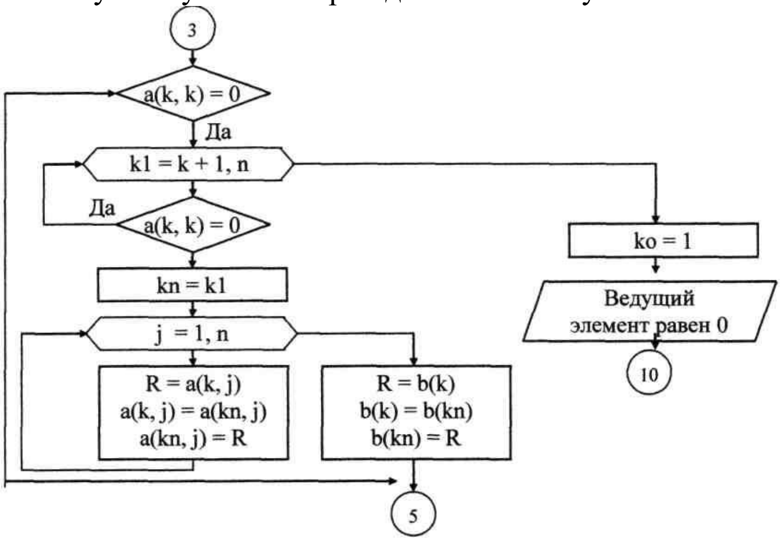
Блок 3. Організовуємо в циклі по k, де k = 1, (n - 1), обчислення по всіх кроках прямого ходу. Останній n-й крок прямого ходу виводиться з циклу.

За допомогою двох вкладених циклів з керівниками змінними i і j організовуємо введення коефіцієнтів a_i і вільних членів b_i початкової системи. Для того, щоб надалі можна було виконати в блоці 9 перевірку результату, в алгоритмі передбачається збереження значень a_{ij} і b_i початкової системи за допомогою тих, що переприсвоїли:

$$c_{ij} = a_{ij}$$

$$d_i = b_i.$$

Блок 4. Пошук ненульового провідного елементу.



Пошук ненульового провідного елементу ведеться в наступному порядку:

а) на кожному кроці прямого ходу, в к-й рядку, елемент a(k, k) дорівнюється з нулем.

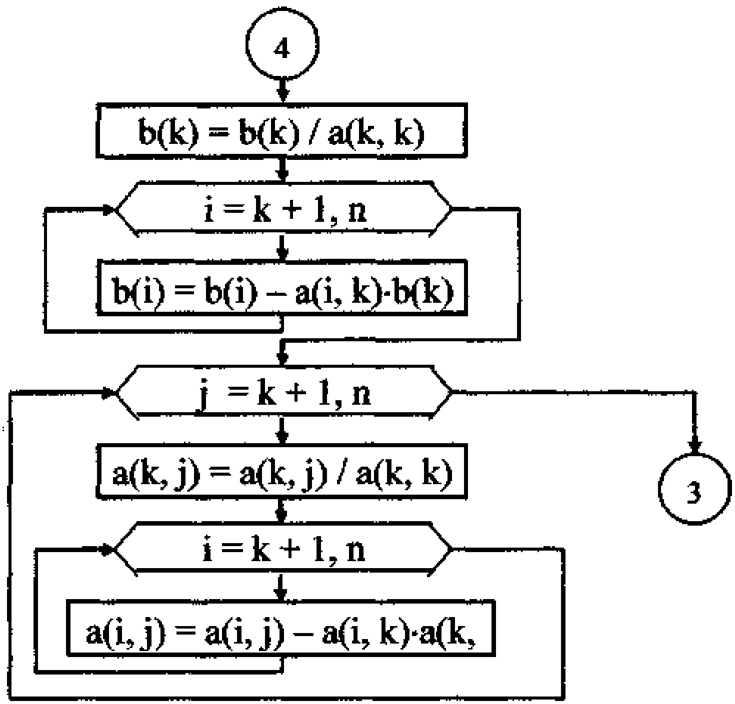
б) якщо в к-й рядку є нульовий провідний елемент а, те в к-м стовпці,
починаючи з наступного рядка, в циклі по kl = (k + 1, n) здійснюється пошук ненульового провідного елементу.

в) якщо в рядку kn такий ненульовий елемент a(kn, k) знайдений, то рядки k i kn поелементно, в циклі по j = (1, n), міняємо місцями. Для перестановки елементів рядків використовується робочий осередок R.

г) якщо ненульовий провідний елемент не знайдений, то коду помилок КО привласнюється одиниця і розрахунок припиняється.

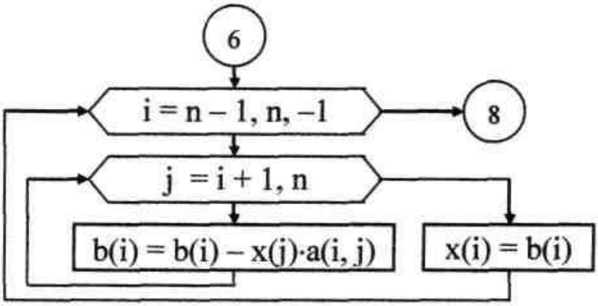
Блок 5 - крок прямого ходу.

Проводимо прямий хід виключення невідомих шляхом перетворення коефіцієнтів і вільних членів системи по отриманих раніше рекурентним формулам:



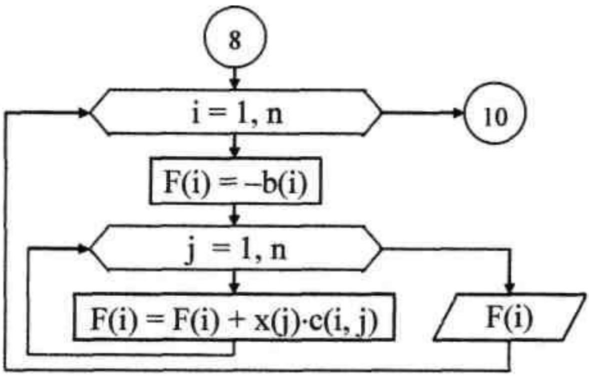
$k = 1, n - 1$
$i = k + 1, n$
$j = k + 1, n$

Блок 6. У цьому блоці виконується останній крок прямого ходу.
Блок 7.



В процесі зворотного ходу методу Гауса з системи трикутного вигляду послідовно в зворотному порядку в циклі i = n - 1, 1 знаходимо невідомі системи по рекурентній формулі.

Блок 8. Коріння системи можна вивести у вигляді масиву x, або поелементно в циклі.



Блок 9. Цей блок в алгоритмі рекомендується використовувати в процесі відладки методу (в цьому випадку можна об'єднати блоки 8 і 9). Надалі при використанні методу Гауса, при вирішенні інших прикладних завдань блок 9 можна опустити. Тоді в блоці те, що 2 переприсвоїло не потрібне.

Контрольні питання

1. Обґрунтувати необхідність застосування чисельних методів для вирішення систем лінійних рівнянь.
2. У чому полягає сенс застосування метода Гауса?
3. У чому переваги і недоліки застосування даного метода?

Лабораторна робота

Рішення систем лінійних рівнянь методом Крамера

Мета роботи: 1. Практичне дослідження методів чисельного вирішення лінійних рівнянь.
2. Отримань практичних навиків роботи з системою Turbo Pascal 7.0, закріплення знань щодо використання функцій.

Зміст звіту

1. Постановка завдання. Початкові дані.
2. Аналіз рішення задачі. Алгоритм рішення (блок – схема алгоритму).
3. Текст програми.
4. Результат виконання програми.
5. Висновки по роботі.

Постановка завдання

1. Використовуючи дані, знайти рішення СЛР методом Крамера.

$$1)\begin{cases} 8x_1 + 4x_2 - 6x_3 = -18, \\ -2x_1 - 4x_3 - 6x_4 = -2, \\ 6x_1 + 4x_2 + 4x_3 + 6x_4 = -14, \\ 4x_1 + 6x_2 + 8x_3 + 8x_4 = -6; \end{cases} \qquad 2)\begin{cases} -8x_1 + 2x_2 - 2x_4 = 34, \\ -6x_1 - 4x_2 - 2x_3 - 2x_4 = 24, \\ -10x_1 + 2x_2 + 4x_4 = 68, \\ -2x_1 - 6x_2 + 8x_3 - 4x_4 = -36; \end{cases}$$

$$3)\begin{cases}6x_1-4x_3-4x_4=-34,\\-10x_1+10x_3=20,\\-8x_1-4x_2+2x_4=44,\\-2x_1-10x_2+6x_3+4x_4=-2;\end{cases}$$

$$4)\begin{cases}8x_1+2x_2-2x_3=-32,\\2x_1+4x_3+2x_4=-14,\\2x_1-8x_2-8x_3=6,\\-10x_1-4x_2+10x_3+2x_4=24;\end{cases}$$

$$5)\begin{cases}2x_1+6x_2+4x_3=-16,\\-6x_1+8x_2+4x_3+2x_4=34,\\-2x_2+6x_3-10x_4=-60,\\6x_1-10x_2+2x_3-81x_4=-78;\end{cases}$$

$$6)\begin{cases}6x_1-2x_2+10x_3+4x_4=-46,\\-6x_1-4x_2+10x_3+10x_4=36,\\x_3-4x_4=-19,\\8x_2-4x_3+10x_4=60;\end{cases}$$

$$7)\begin{cases}6x_1+8x_3-6x_4=-2,\\10x_1-10x_2-2x_3-8x_4=42,\\4x_1-2x_2-2x_3+10x_4=12,\\-4x_1-2x_2-2x_3=4;\end{cases}$$

$$8)\begin{cases}-4x_1+6x_2-4x_3-6x_4=-18,\\4x_1+10x_2-8x_3+2x_4=-18,\\2x_2-6x_3+6x_4=0,\\-2x_3-2x_4=2;\end{cases}$$

$$9)\begin{cases}4x_1+4x_2+4x_3+8x_4=-12,\\-8x_2-2x_3+6x_4=26,\\-2x_1+2x_2-8x_3-8x_4=0,\\-8x_2+2x_3-6x_4=22;\end{cases}$$

$$10)\begin{cases}2x_1+8x_2+6x_3=-28,\\-4x_2+6x_3+8x_4=6,\\-8x_1+4x_2+10x_4=-20,\\-6x_1-2x_2-4x_3+2x_4=4;\end{cases}$$

$$11)\begin{cases}-4x_1-8x_3-4x_4=4,\\6x_1-2x_2-6x_3-6x_4=18,\\-4x_1+2x_2-8x_3-8x_4=-2,\\-8x_2-6x_3-8x_4=30;\end{cases}$$

$$12)\begin{cases}-2x_1-2x_2+2x_4=4,\\-8x_2-6x_3-8x_4=30,\\-4x_1-10x_2-10x_3+10x_4=36,\\10x_1+4x_2+4x_3-4x_4=-6;\end{cases}$$

$$13)\begin{cases}-10x_1+10x_4-20=0,\\6x_1+2x_2-6x_3+2x_4-24=0,\\2x_1+6x_2+2x_3+10x_4-28=0,\\4x_1+6x_2+4x_3-4x_4+16=0;\end{cases}$$

$$14)\begin{cases}-9x_1-9x_2-5x_3+10x_4=31,\\-4x_1+7x_2+5x_3=-14,\\9x_1-5x_2+x_3=7,\\-11x_2-13x_3+2x_4=32;\end{cases}$$

$$15)\begin{cases}-5x_1+x_2-7x_3+8x_4=33,\\9x_2-3x_3-4x_4=-6,\\-3x_1+7x_2+5x_3=-13,\\-7x_2-11x_3-4x_4=10;\end{cases}\qquad 16)\begin{cases}3x_1-11x_2+5x_3+4x_4=5,\\-8x_1-5x_2-3x_3+10x_4=28,\\3x_1-x_2+5x_3=-7,\\-6x_1-11x_2+3x_3=-12;\end{cases}$$

$$17)\begin{cases}5x_1-7x_2-11x_3+8x_4=51,\\-6x_1-9x_2-3x_3-2x_4=-6,\\3x_1+5x_2-5x_3-6x_4=-5,\\-5x_2-13x_3+4x_4=38;\end{cases}\qquad 18)\begin{cases}x_1+7x_2+3x_3+4x_4=7,\\5x_2-9x_3-4x_4=6,\\9x_1-9x_2-9x_3+8x_4=51,\\-10x_1-7x_2-7x_3-6x_4=-14;\end{cases}$$

$$19)\begin{cases}7x_1+x_2+3x_3+4x_4=11,\\3x_2+5x_3-6x_4=-6,\\x_1+9x_2+5x_3=1,\\5x_2-3x_3+10x_4=10;\end{cases}\qquad 20)\begin{cases}-3x_1-11x_2-13x_3+12x_4=9,\\7x_2-9x_3+6x_4=6,\\5x_1-3x_2-x_3+10x_4=15,\\-12x_1-x_2-11x_3-2x_4=-14;\end{cases}$$

$$21)\begin{cases}-3x_1+x_2-11x_3+4x_4=1,\\-4x_1-3x_2-3x_3=-4,\\x_1-7x_2-13x_3-2x_4=-1,\\4x_1-5x_2+3x_3-4x_4=0;\end{cases}\qquad 22)\begin{cases}-5x_1-7x_2-5x_3+6x_4=1,\\5x_2+7x_3=0,\\-3x_1+7x_2+x_3+12x_4=9,\\-12x_1+5x_2-11x_3-2x_4=-14;\end{cases}$$

$$23)\begin{cases}-5x_1-3x_2+5x_3-2x_4=-7,\\4x_1+7x_2-3x_3+8x_4=12,\\x_1-11x_2-x_3+6x_4=7,\\3x_2+x_3-4x_4=-4;\end{cases}\qquad 24)\begin{cases}9x_1-5x_2+5x_3=9,\\-2x_1+9x_2-3x_3+2x_4=0,\\x_1+3x_2+3x_3-2x_4=-1,\\-14x_1-3x_2+3x_3-2x_4=-16;\end{cases}$$

$$25)\begin{cases}3x_1-3x_2+x_3-2x_4=-2,\\-9x_2-5x_3+8x_4=-14,\\-x_1-3x_2-3x_3-6x_4=-6,\\-3x_2-9x_3-4x_4=-12;\end{cases}\qquad 26)\begin{cases}-3x_1+9x_2+7x_3-4x_4=16,\\-6x_1+5x_2-9x_3+8x_4=-4,\\-7x_1-9x_2-7x_3+4x_4=-16,\\7x_2+x_3=8;\end{cases}$$

2. Результатом роботи програми є значення коренів системи лінійних рівнянь.
3. За наслідками роботи зробити виводи про переваги і недоліки даного методу.

Короткі відомості з теорії

Хай ми маємо квадратну систему лінійних рівнянь:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases}.$$

Її можна записати в матричній формі:

$$\mathbf{AX} = \mathbf{B}$$

де

$$\mathbf{A} = (a_{ij})i, j = 1,2,\ldots,n; \mathbf{X} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}; \mathbf{B} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}.$$

Якщо визначник матриці А не рівний нулю, то система має єдине рішення, визначуване формулами:

$$\begin{cases} x_1 = \dfrac{\Delta_1}{\Delta} \\ x_2 = \dfrac{\Delta_2}{\Delta} \\ \ldots \quad \ldots \\ x_n = \dfrac{\Delta_n}{\Delta} \end{cases}.$$

Тут $_i$ – визначник *n-го* порядку, що виходить з визначника Δ матриці **А** коефіцієнтів системи заміною *і-го* стовпця стовпцем вільних членів.

Наприклад

$$\begin{cases} 3x_1 + 2x_2 - \quad x_3 = 2 \\ 2x_1 - \quad x_2 + 2x_3 = 3 \\ 4x_1 + \quad x_2 + 3x_3 = 5 \end{cases};$$

$$\Delta = \begin{vmatrix} 3 & 2 & -1 \\ 2 & -1 & 2 \\ 4 & 1 & 3 \end{vmatrix} = -17; \qquad \Delta_1 = \begin{vmatrix} 2 & 2 & -1 \\ 3 & -1 & 2 \\ 5 & 1 & 3 \end{vmatrix} = -16;$$

$$\Delta_2 = \begin{vmatrix} 3 & 2 & -1 \\ 2 & 3 & 2 \\ 4 & 5 & 3 \end{vmatrix} = 3; \qquad \Delta_3 = \begin{vmatrix} 3 & 2 & 2 \\ 2 & -1 & 3 \\ 4 & 1 & 5 \end{vmatrix} = -8;$$

$$x_1 = \frac{16}{17}; \quad x_2 = -\frac{3}{17}; \quad x_3 = \frac{8}{17}.$$

Відзначимо, що якщо визначник матриці А коефіцієнтів квадратної системи лінійних рівнянь рівний нулю, то можливий один з двох випадків: або система несумісна, або вона сумісна і невизначена.

Контрольні питання

1. Обґрунтувати необхідність застосування чисельних методів для вирішення систем лінійних рівнянь.
2. У чому полягає сенс застосування метода Крамера?
3. У чому переваги і недоліки застосування даного метода?

Лабораторна робота

Метод Зейделя.

М е т а р о б о т и : 1. Практичне дослідження методу простих ітерацій.
2. Отримання практичних навиків роботи з системою Turbo Pascal 7.0, закріплення знань щодо використання функцій.

Зміст звіту

1. Постановка завдання. Початкові дані.
2. Аналіз рішення задачі. Алгоритм рішення (блок – схема алгоритму).
3. Текст програми.
4. Результат виконання програми.
5. Висновки по роботі.

Теоретичні відомості

Є система лінійних алгебраїчних рівнянь, що зведена до нормального вигляду

$$\begin{cases} x_1 = \beta_1 + \alpha_{11}x_1 + \alpha_{12}x_2 + \alpha_{13}x_3 + ... + \alpha_{1n}x_n \\ x_2 = \beta_2 + \alpha_{21}x_1 + \alpha_{22}x_2 + \alpha_{23}x_3 + ... + \alpha_{2n}x_n \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ x_n = \beta_n + \alpha_{n1}x_1 + \alpha_{n2}x_2 + \alpha_{n3}x_3 + ... + \alpha_{nn}x_n \end{cases}.$$

Тоді за методом Зейделя, вибираючи вектор початкових наближень

$$X^{(0)} = \left\{x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, ..., x_n^{(0)}\right\}$$

(як правило, це стовпець вільних членів $\beta = \{\beta_1, \beta_2, \beta_3, ..., \beta_n\}$), уточнення значень невідомих проводять наступним чином:

1) перше наближення:

$$\begin{cases} x_1^{(1)} = \beta_1 + \alpha_{11}x_1^{(0)} + \alpha_{12}x_2^{(0)} + \alpha_{13}x_3^{(0)} + ... + \alpha_{1n}x_n^{(0)} \\ x_2^{(1)} = \beta_2 + \alpha_{21}x_1^{(1)} + \alpha_{22}x_2^{(0)} + \alpha_{23}x_3^{(0)} + ... + \alpha_{2n}x_n^{(0)} \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ x_n^{(1)} = \beta_n + \alpha_{n1}x_1^{(1)} + \alpha_{n2}x_2^{(1)} + \alpha_{n3}x_3^{(1)} + ... + \alpha_{n,n-1}x_n^{(1)} + \alpha_{nn}x_n^{(0)} \end{cases}$$

2) $k + 1$ наближення

$$\begin{cases} x_1^{(k+1)} = \beta_1 + \alpha_{11}x_1^{(k)} + \alpha_{12}x_2^{(k)} + \alpha_{13}x_3^{(k)} + ... + \alpha_{1n}x_n^{(k)} \\ x_2^{(k+1)} = \beta_2 + \alpha_{21}x_1^{(k+1)} + \alpha_{22}x_2^{(k)} + ... + \alpha_{2n}x_n^{(k)} \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ x_i^{(k+1)} = \beta_i + \alpha_{i1}x_1^{(k+1)} + \alpha_{i2}x_2^{(k+1)} + ... + \alpha_{i,i-1}x_{i-1}^{(k+1)} + \alpha_{ii}x_i^{(k)} + ... + \alpha_{in}x_n^{(k)} \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ x_n^{(k+1)} = \beta_n + \alpha_{n1}x_1^{(k+1)} + \alpha_{n2}x_2^{(k+1)} + ... + \alpha_{n,n-1}x_{n-1}^{(k+1)} + \alpha_{nn}x_n^{(k)} \end{cases}$$

$k = 0, 1, 2, \dots$.

Таким чином ітераційний процес подібний до методу простих ітерацій, однак уточнені значення $x_i^{(k+1)}$ одразу ж підставляються в наступні рівняння:

$$\boxed{x_i^{(k+1)} = \beta_i + \sum_{j=1}^{i-1} \alpha_{ij} \cdot x_j^{(k+1)} + \sum_{j=i}^{n} \alpha_{ij} \cdot x_j^{(k)}} \; -$$

метод Зейделя.

Іншими словами, метод Зейделя відрізняється від методу простої ітерації тим, що при обчисленні $x_i^{(k+1)}$ на “k + 1”-му кроці враховуються значення $x_1^{(k+1)}$, $x_2^{(k+1)}$, $x_{i\text{-}1}^{(k+1)}$, обчислені на цьому самому кроці.

Слід сподіватись, що ітерації за методом Зейделя дадуть при тому ж числі кроків більш точні результати, ніж за методом простої ітерації. Або така ж точність буде досягнута за менше число кроків, оскільки чергові значення невідомих визначаються тут більш точно ітераційний процес припиняється.

Завдання

Розв’язати систему лінійних рівнянь методом ітерації та методом Зейделя з точністю до 0,001.

№1 $\begin{cases} 2{,}7x_1+3{,}3x_2+1{,}3x_3=2{,}1; \\ 3{,}5x_1-1{,}7x_2+2{,}8x_3=1{,}7; \\ 4{,}1x_1+5{,}8x_2-1{,}7x_3=0{,}8. \end{cases}$ №2 $\begin{cases} 1{,}7x_1+2{,}8x_2+1{,}9x_3=0{,}7; \\ 2{,}1x_1+3{,}4x_2+1{,}8x_3=1{,}1; \\ 4{,}2x_1-1{,}7x_2+1{,}3x_3=2{,}8. \end{cases}$

№3 $\begin{cases} 3{,}1x_1+2{,}8x_2+1{,}9x_3=0{,}2; \\ 1{,}9x_1+3{,}1x_2+2{,}1x_3=2{,}1; \\ 7{,}5x_1+3{,}8x_2+4{,}8x_3=5{,}6. \end{cases}$ №4 $\begin{cases} 9{,}1x_1+5{,}6x_2+7{,}8x_3=9{,}8; \\ 3{,}8x_1+5{,}1x_2+2{,}8x_3=6{,}7; \\ 4{,}1x_1+5{,}7x_2+1{,}2x_3=5{,}8. \end{cases}$

№5 $\begin{cases} 3{,}3x_1+2{,}1x_2+2{,}8x_3=0{,}8; \\ 4{,}1x_1+3{,}7x_2+4{,}8x_3=5{,}7; \\ 2{,}7x_1+1{,}8x_2+1{,}1x_3=3{,}2. \end{cases}$ №6 $\begin{cases} 7{,}6x_1+5{,}8x_2+4{,}7x_3=10{,}1; \\ 3{,}8x_1+4{,}1x_2+2{,}7x_3=9{,}7; \\ 2{,}9x_1+2{,}1x_2+3{,}8x_3=7{,}8. \end{cases}$

№7 $\begin{cases} 3{,}2x_1-2{,}5x_2+3{,}7x_3=6{,}5; \\ 0{,}5x_1+0{,}34x_2+1{,}7x_3=-0{,}24; \\ 1{,}6x_1+2{,}3x_2-1{,}5x_3=4{,}3. \end{cases}$ №8 $\begin{cases} 5{,}4x_1-2{,}3x_2+3{,}4x_3=-3{,}5; \\ 4{,}2x_1+1{,}7x_2-2{,}3x_3=2{,}7; \\ 3{,}4x_1+2{,}4x_2+7{,}4x_3=1{,}9. \end{cases}$

№9 $\begin{cases} 3{,}6x_1 + 1{,}8x_2 + 1{,}3x_3 = 3{,}8; \\ 2{,}7x_1 - 3{,}6x_2 + 2{,}8x_3 = 0{,}4; \\ 1{,}5x_1 + 4{,}5x_2 + 3{,}3x_3 = -1{,}6. \end{cases}$ №10 $\begin{cases} 5{,}6x_1 + 2{,}7x_2 - 1{,}7x_3 = 1{,}9; \\ 3{,}4x_1 - 3{,}6x_2 - 6{,}7x_3 = -2{,}4; \\ 0{,}8x_1 + 1{,}3x_2 + 3{,}7x_3 = 1{,}2. \end{cases}$

№11 $\begin{cases} 2{,}7x_1 + 0{,}9x_2 - 1{,}5x_3 = 3{,}5; \\ 4{,}5x_1 - 2{,}8x_2 + 6{,}7x_3 = 2{,}6; \\ 5{,}1x_1 + 3{,}7x_2 - 1{,}4x_3 = -0{,}14. \end{cases}$ №12 $\begin{cases} 4{,}5x_1 - 3{,}5x_2 + 7{,}4x_3 = 2{,}5; \\ 3{,}1x_1 - 0{,}6x_2 - 2{,}3x_3 = -1{,}6; \\ 0{,}8x_1 + 7{,}4x_2 - 0{,}5x_3 = 6{,}4. \end{cases}$

№13 $\begin{cases} 3{,}8x_1 + 6{,}7x_2 - 1{,}2x_3 = 5{,}2; \\ 6{,}4x_1 + 1{,}3x_2 - 2{,}7x_3 = 3{,}8; \\ 2{,}4x_1 - 4{,}5x_2 + 3{,}5x_3 = -0{,}6. \end{cases}$ №14 $\begin{cases} 5{,}4x_1 - 6{,}2x_2 - 0{,}5x_3 = 0{,}52; \\ 3{,}4x_1 + 2{,}3x_2 + 0{,}8x_3 = -0{,}8; \\ 2{,}4x_1 - 1{,}1x_2 + 3{,}8x_3 = 1{,}8. \end{cases}$

№15 $\begin{cases} 7{,}8x_1 + 5{,}3x_2 + 4{,}8x_3 = 1{,}8; \\ 3{,}3x_1 + 1{,}1x_2 + 1{,}8x_3 = 2{,}3; \\ 4{,}5x_1 + 3{,}3x_2 + 2{,}8x_3 = 3{,}4. \end{cases}$ №16 $\begin{cases} 3{,}8x_1 + 4{,}1x_2 - 2{,}3x_3 = 4{,}8; \\ -2{,}1x_1 + 3{,}9x_2 - 5{,}8x_3 = 3{,}3; \\ 1{,}8x_1 + 1{,}1x_2 - 2{,}1x_3 = 5{,}8. \end{cases}$

№17 $\begin{cases} 1{,}7x_1 - 2{,}2x_2 + 3{,}0x_3 = 1{,}8; \\ 2{,}1x_1 + 1{,}9x_2 - 2{,}3x_3 = 2{,}8; \\ 4{,}2x_1 + 3{,}9x_2 - 3{,}1x_3 = 5{,}1. \end{cases}$ №18 $\begin{cases} 2{,}8x_1 + 3{,}8x_2 - 3{,}2x_3 = 4{,}5; \\ 2{,}5x_1 - 2{,}8x_2 + 3{,}3x_3 = 7{,}1; \\ 6{,}5x_1 - 7{,}1x_2 + 4{,}8x_3 = 6{,}3. \end{cases}$

№19 $\begin{cases} 3{,}3x_1 + 3{,}7x_2 + 4{,}2x_3 = 5{,}8; \\ 2{,}7x_1 - 2{,}3x_2 - 2{,}9x_3 = 6{,}1; \\ 4{,}1x_1 + 4{,}8x_2 - 5{,}0x_3 = 7{,}0. \end{cases}$ №20 $\begin{cases} 7{,}1x_1 + 6{,}8x_2 + 6{,}1x_3 = 7{,}0; \\ 5{,}0x_1 + 4{,}8x_2 + 5{,}3x_3 = 6{,}1; \\ 8{,}2x_1 + 7{,}8x_2 + 7{,}1x_3 = 5{,}8. \end{cases}$

№21 $\begin{cases} 3{,}7x_1 + 3{,}1x_2 + 4{,}0x_3 = 5{,}0; \\ 4{,}1x_1 + 4{,}5x_2 - 4{,}8x_3 = 4{,}9; \\ -2{,}1x_1 - 3{,}7x_2 + 1{,}8x_3 = 2{,}7. \end{cases}$ №22 $\begin{cases} 4{,}1x_1 + 5{,}2x_2 - 5{,}8x_3 = 7{,}0; \\ 3{,}8x_1 - 3{,}1x_2 + 4{,}0x_3 = 5{,}3; \\ 7{,}8x_1 + 5{,}3x_2 - 6{,}3x_3 = 5{,}8. \end{cases}$

Зразок виконання завдання в паперовому вигляді

$$\begin{cases} 4{,}5x_1 - 1{,}8x_2 + 3{,}6x_3 = -1{,}7; & \text{(I)} \\ 3{,}1x_1 + 2{,}3x_2 - 1{,}2x_3 = 3{,}6; & \text{(II)} \\ 1{,}8x_1 + 2{,}5x_2 + 4{,}6x_3 = 2{,}2. & \text{(III)} \end{cases}$$

Зведемо систему до вигляду, в якому елементи головної діагоналі перевищували б інші елементи рядків:

$$\begin{cases} 7{,}6x_1 + 0{,}5x_2 + 2{,}4x_3 = 1{,}9; & (\mathrm{I} + \mathrm{II}) \\ 2{,}2x_1 + 9{,}1x_2 + 4{,}4x_3 = 9{,}7; & (2\mathrm{III} + \mathrm{II} - \mathrm{I}) \\ -1{,}3x_1 + 0{,}2x_2 + 5{,}8x_3 = -1{,}4. & (\mathrm{III} - \mathrm{II}) \end{cases}$$

$$\begin{cases} 10x_1 = 2{,}4x_1 - 0{,}5x_2 - 2{,}4x_3 + 1{,}9; \\ 10x_2 = -2{,}2x_1 + 0{,}9x_2 - 4{,}4x_3 + 9{,}7; \\ 10x_3 = 1{,}3x_1 - 0{,}2x_2 - 4{,}2x_3 - 1{,}4. \end{cases}$$

$$\begin{cases} x_1 = 0{,}24x_1 - 0{,}05x_2 - 0{,}24x_3 + 0{,}19; \\ x_2 = -0{,}22x_1 + 0{,}09x_2 - 0{,}44x_3 + 0{,}97; \\ x_3 = 0{,}13x_1 - 0{,}02x_2 - 0{,}42x_3 - 0{,}14. \end{cases}$$

Норма $\|A\|_1$ матриці, яка складається з коефіцієнтів при невідомих в правих частинах рівнянь, рівна $\{0{,}53; 0{,}77; 0{,}57\} = 0{,}77 < 1;$ отже, процес Зейделя збігається.

Розрахунки розташовуємо в таблиці:

N	x_1	x_2	x_3	N	x_1	x_2	x_3
0	0,19	0,97	-0,14	5	0,2467	1,1138	-0,2237
1	0,2207	1,0703	-0,1915	6	0,2472	1,1143	-0,2241
2	0,2354	1,0988	-0,2118	7	0,2474	1,1145	-0,2243
3	0,2424	1,1088	-0,2196	8	0,2475	1,1145	-0,2243
4	0,2454	1,1124	-0,2226				

Відповідь: $x_1 \approx 0,248$; $x_2 \approx 1,115$; $x_3 \approx -0,224$.

Лабораторна робота

Тема: Метод простих ітерацій.

Мета роботи: 1. Практичне дослідження методу простих ітерацій.
2. Отримання практичних навиків роботи з системою Turbo Pascal 7.0, закріплення знань щодо використання функцій.

Зміст звіту

1. Постановка завдання. Початкові дані.
2. Аналіз рішення задачі. Алгоритм рішення (блок – схема алгоритму).
3. Текст програми.
4. Результат виконання програми.
5. Висновки по роботі.

Нехай дано лінійну систему

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases} \qquad (1)$$

Розглянемо матриці

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix}; \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix}; \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \end{bmatrix}$$

Тоді систему (1) можна записати у вигляді матричного рівняння

$$AX = b \tag{2}$$

Будемо вважати, що діагональні коефіцієнти $a_{ii} \neq 0$ (i = 1, 2,…, n).

Розв'яжемо перше рівняння системи (1) відносно x_1, друге відносно x_2 і т.д. Тоді одержимо еквівалентну систему

$$\begin{cases} x_1 = \beta_1 + \alpha_{12}x_2 + \alpha_{13}x_3 + ... + \alpha_{1n}x_n \\ x_2 = \beta_2 + \alpha_{21}x_1 + \alpha_{23}x_3 + ... + \alpha_{2n}x_n \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ x_n = \beta_n + \alpha_{n1}x_1 + \alpha_{n2}x_2 + ... + \alpha_{n,n-1}x_{n-1} \end{cases}, \tag{3}$$

де $\beta_i = \dfrac{b_i}{a_{ii}}$; $\alpha_{ij} = -\dfrac{a_{ij}}{a_{ii}}$, при $i \neq j$; $\alpha_{ij} = 0$, при $i = j$;

$$x_i = -\sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{a_{ij}}{a_{ii}} x_j + \frac{b_i}{a_{ii}}\ ;\ a_{ii} \neq 0\ ;\ i = \overline{1,n}$$

Іноді кажуть, що система (3) зведена до нормального вигляду.

Введемо матриці α та β

$$\alpha = \begin{bmatrix} \alpha_{11} & \alpha_{12} & ... & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & ... & \alpha_{2n} \\ ... & ... & ... & ... \\ \alpha_{n1} & \alpha_{n2} & ... & \alpha_{nn} \end{bmatrix}; \quad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ ... \\ \beta_n \end{bmatrix}$$

Систему (3) запишемо у вигляді

$$x = \beta + \alpha \cdot x \tag{4}$$

Систему (3) будемо розв'язувати методом послідовних наближень. За нульове наближення позначимо, наприклад, стовпчик вільних членів $x^{(0)} = \beta$. Далі послідовно будуємо матриці-стовпці:

$$x^{(1)} = \beta + \alpha x^{(0)} \text{ – перше наближення}$$

$$x^{(2)} = \beta + \alpha x^{(1)} \text{ – друге наближення і т.д.}$$

Будь-яке (k + 1)-е наближення обчислюється за формулою:

$$x^{(k+1)} = \beta + \alpha x^{(k)}, \ (k = 0, 1, 2, \ldots) \qquad (5)$$

В розгорнутому вигляді $\quad x_i^{(k+1)} = \beta_i + \sum_{j=1}^{n} \alpha_{ij} x_j^{(k)}$.

Якщо послідовність наближень $x^{(0)}, x^{(1)}, \ldots, x^k$ має границю

$$x = \lim_{k \to \infty} x^{(k)}, \qquad (6)$$

то ця границя є розв’язком системи (3).

На практиці ітераційний процес припиняють, коли $\left| x_i^{(k)} - x_i^{(k-1)} \right| < \varepsilon$, де ε – гранична абсолютна похибка.

Умови збіжності ітераційного процесу

Нехай задана зведена до нормального вигляду система лінійних рівнянь

$$X = \beta + \alpha X$$

Умова збіжності : якщо сума модулів елементів рядків або модулів елементів стовпців матриці α менша ніж 1, то процес ітерації для даної системи збігається до єдиного розв’язку незалежно від вибору вектора початкових наближень.

Для системи

$$\begin{cases} 8x_1 + x_2 + x_3 = 26 \\ x_1 + 5x_2 - x_3 = 7 \\ x_1 - x_2 + 5x_3 = 7 \end{cases} \quad \text{або}$$

$$\begin{cases} x_1 = 3{,}25 - 0{,}125x_2 - 0{,}125x_3 \\ x_2 = 1{,}4 - 0{,}2x_1 + 0{,}2x_3 \\ x_3 = 1{,}4 - 0{,}2x_1 + 0{,}2x_2 \end{cases},$$

де $\alpha = \begin{Vmatrix} 0 & -0{,}125 & -0{,}125 \\ -0{,}2 & 0 & 0{,}2 \\ -0{,}2 & 0{,}2 & 0 \end{Vmatrix}$

– сума модулів по стовпцях

$$|\alpha_{11}| + |\alpha_{21}| + |\alpha_{31}| = 0{,}2 + 0{,}2 = 0{,}4 < 1$$

$$|\alpha_{12}| + |\alpha_{22}| + |\alpha_{32}| = 0{,}125 + 0{,}2 = 0{,}325 < 1$$

$$|\alpha_{13}| + |\alpha_{23}| + |\alpha_{33}| = 0{,}125 + 0{,}2 = 0{,}325 < 1$$

Аналогічно можна було б перевірити виконання умови збіжності, беручи суми модулів елементів рядків.

Наведена вище умова являється достатньою, але не необхідною. Це означає, що якщо умова виконується, то процес буде збіжним. Коли ж умова не виконується, то це ще не означає, що процес буде розбіжним.

Для системи лінійних рівнянь, заданих у вигляді

$$A*X = b,$$

метод простої ітерації збігається, якщо модулі діагональних коефіцієнтів a_{ii} для кожного рівняння системи більші, ніж суми модулів всієї решти коефіцієнтів (не враховуючи вільних членів).

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|, (i = \overline{1, n}) \tag{10}$$

Завдання

Розв'язати систему лінійних рівнянь методом ітерації з точністю до 0,001.

№ 1

$$\begin{cases} x_1 = 0{,}23x_1 - 0{,}04x_2 + 0{,}21x_3 - 0{,}18x_4 + 1{,}24 \\ x_2 = 0{,}45x_1 - 0{,}23x_2 + 0{,}06x_3 - 0{,}88 \\ x_3 = 0{,}26x_1 + 0{,}34x_2 - 0{,}11x_3 + 0{,}62 \\ x_4 = 0{,}05x_1 - 0{,}26x_2 + 0{,}34x_3 - 0{,}12x_4 - 1{,}17. \end{cases}$$

№ 2

$$\begin{cases} x_1 = 0{,}21x_1 + 0{,}12x_2 - 0{,}34x_3 - 0{,}16x_4 + 0{,}64 \\ x_2 = 0{,}34x_1 - 0{,}08x_2 + 0{,}17x_3 - 0{,}18x_4 + 1{,}42 \\ x_3 = 0{,}16x_1 + 0{,}34x_2 + 0{,}15x_3 - 0{,}31x_4 - 0{,}42 \\ x_4 = 0{,}12x_1 - 0{,}26x_2 - 0{,}08x_3 + 0{,}25x_4 + 0{,}83. \end{cases}$$

№ 3

$$\begin{cases} x_1 = 0{,}32x_1 - 0{,}18x_2 + 0{,}02x_3 + 0{,}21x_4 + 1{,}83 \\ x_2 = 0{,}16x_1 + 0{,}12x_2 - 0{,}14x_3 + 0{,}27x_4 - 0{,}65 \\ x_3 = 0{,}37x_1 + 0{,}27x_2 - 0{,}02x_3 - 0{,}24x_4 + 2{,}23 \\ x_4 = 0{,}12x_1 + 0{,}21x_2 - 0{,}18x_3 + 0{,}25x_4 - 1{,}13. \end{cases}$$

№4

$$\begin{cases} x_1 = 0{,}42x_1 - 0{,}32x_2 + 0{,}03x_3 + 0{,}44 \\ x_2 = 0{,}11x_1 - 0{,}26x_2 - 0{,}36x_3 + 1{,}42 \\ x_3 = 0{,}12x_1 + 0{,}08x_2 - 0{,}14x_3 - 0{,}24x_4 - 0{,}8320, \\ x_4 = 0{,}15x_1 - 0{,}35x_2 - 0{,}18x_3 - 1{,}42. \end{cases}$$

№ 5

$$\begin{cases} x_1 = 0{,}18x_1 - 0{,}34x_2 - 0{,}12x_3 + 0{,}15x_4 - 1{,}33 \\ x_2 = 0{,}11x_1 + 0{,}23x_2 - 0{,}15x_3 + 0{,}32x_4 + 0{,}84 \\ x_3 = 0{,}05x_1 - 0{,}12x_2 + 0{,}14x_3 - 0{,}18x_4 - 1{,}16 \\ x_4 = 0{,}12x_1 + 0{,}08x_2 + 0{,}06x_3 + x0{,}57. \end{cases}$$

№ 6

$$\begin{cases} x_1 = 0{,}13x_1 + 0{,}23x_2 - 0{,}44x_3 - 0{,}05x_4 + 2{,}13 \\ x_2 = 0{,}24x_1 - 0{,}31x_3 + 0{,}15x_4 - 0{,}18 \\ x_3 = 0{,}06x_1 + 0{,}15x_2 - 0{,}23x_4 + 1{,}44 \\ x_4 = 0{,}72x_1 - 0{,}08x_2 - 0{,}05x_3 + 2{,}42. \end{cases}$$

№ 7

$$\begin{cases} x_1 = 0{,}17x_1 + 0{,}31x_2 - 0{,}18x_3 + 0{,}22x_4 - 1{,}71 \\ x_2 = -0{,}21x_1 + 0{,}33 + 0{,}22x_4 + 0{,}62 \\ x_3 = 0{,}32x_1 - 0{,}18x_2 + 0{,}05x_3 - 0{,}19x_4 - 0{,}89 \\ x_4 = 0{,}12x_1 + 0{,}28x_2 - 0{,}14x_3 + 0{,}94. \end{cases}$$

№ 8

$$\begin{cases} x_1 = 0{,}13x_1 + 0{,}27x_2 - 0{,}22x_3 - 0{,}18x_4 + 1{,}21 \\ x_2 = -0{,}21x_1 - 0{,}45x_3 + 0{,}18x_4 - 0{,}48 \\ x_3 = 0{,}12x_1 + 0{,}13x_2 - 0{,}33x_3 + 0{,}18x_4 - 0{,}48 \\ x_4 = 0{,}33x_1 - 0{,}05x_2 + 0{,}06x_3 - 0{,}28x_4 - 0{,}17. \end{cases}$$

№ 9

$$\begin{cases} x_1 = 0{,}19x_1 - 0{,}07x_2 + 0{,}38x_3 - 0{,}21x_4 - 0{,}81 \\ x_2 = -0{,}22x_1 + 0{,}08x_2 + 0{,}11x_3 + 0{,}33x_4 - 0{,}64 \\ x_3 = 0{,}51x_1 - 0{,}07x_2 + 0{,}09x_3 - 0{,}11x_4 + 1{,}71 \\ x_4 = 0{,}33x_1 - 0{,}41x_2 - 1{,}21. \end{cases}$$

№ 10

$$\begin{cases} x_1 = 0{,}22x_1 - 0{,}11x_2 + 0{,}31x_4 + 2{,}7 \\ x_2 = 0{,}38x_1 - 0{,}12x_3 + 0{,}22x_4 - 1{,}5 \\ x_3 = 0{,}11x_1 + 0{,}23x_2 - 0{,}51x_4 + 1{,}2 \\ x_4 = 0{,}17x_1 - 0{,}21x_2 + 0{,}31x_3 - 0{,}17. \end{cases}$$

№ 11

$$\begin{cases} x_1 = 0{,}07x_1 - 0{,}08x_2 + 0{,}11x_3 - 0{,}18x_4 + 0{,}51 \\ x_2 = 0{,}18x_1 + 0{,}52x_2 + 0{,}21x_4 + 1{,}17 \\ x_3 = 0{,}13x_1 + 0{,}31x_2 - 0{,}21x_4 - 1{,}02 \\ x_4 = 0{,}08x_1 - 0{,}33x_3 + 0{,}28x_4 + 0{,}28. \end{cases}$$

№ 12

$$\begin{cases} x_1 = 0{,}05x_1 - 0{,}06x_2 - 0{,}12x_3 + 0{,}14x_4 - 2{,}17 \\ x_2 = 0{,}04x_1 - 0{,}12x_2 + 0{,}08x_3 + 0{,}11x_4 + 1{,}4 \\ x_3 = 0{,}34x_1 + 0{,}08x_2 - 0{,}06x_3 + 0{,}14x_4 - 2{,}1 \\ x_4 = 0{,}11x_1 + 0{,}12x_2 - 0{,}03x_4 - 0{,}8. \end{cases}$$

№ 13

$$\begin{cases} x_1 = 0{,}08x_1 - 0{,}03x_2 - 0{,}04x_4 - 1{,}2 \\ x_2 = 0{,}31x_1 + 0{,}27x_3 - 0{,}08x_4 + 0{,}81 \\ x_3 = 0{,}33x_1 - 0{,}07x_3 + 0{,}21x_4 - 0{,}92 \\ x_4 = 0{,}11x_1 + 0{,}03x_3 + 0{,}58x_4 + 0{,}17. \end{cases}$$

№ 14

$$\begin{cases} x_1 = 0{,}12x_1 - 0{,}23x_2 + 0{,}25x_3 - 0{,}16x_4 + 1{,}24 \\ x_2 = 0{,}14x_1 + 0{,}34x_2 - 0{,}18x_3 + 0{,}24x_4 - 0{,}89 \\ x_3 = 0{,}33x_1 + 0{,}03x_2 + 0{,}16x_3 - 0{,}32x_4 + 1{,}15 \\ x_4 = 0{,}12x_1 - 0{,}05x_2 + 0{,}15x_4 - 0{,}57. \end{cases}$$

№ 15

$$\begin{cases} x_1 = 0{,}23x_1 - 0{,}14x_2 + 0{,}06x_3 - 0{,}12x_4 + 1{,}23 \\ x_2 = 0{,}12x_1 + 0{,}32x_3 - 0{,}18x_4 + 0{,}72 \\ x_3 = 0{,}08x_1 - 0{,}12x_2 + 0{,}23x_3 + 0{,}32x_4 - 0{,}58 \\ x_4 = 0{,}25x_1 + 0{,}22x_2 + 0{,}14x_3 + 1{,}56. \end{cases}$$

№ 16

$$\begin{cases} x_1 = 0{,}14x_1 + 0{,}23x_2 + 0{,}18x_3 + 0{,}17x_4 - 1{,}42 \\ x_2 = 0{,}12x_1 - 0{,}14x_2 + 0{,}08x_3 + 0{,}09x_4 - 0{,}83 \\ x_3 = 0{,}16x_1 + 0{,}24x_2 - 0{,}35x_4 + 1{,}21 \\ x_4 = 0{,}23x_1 - 0{,}08x_2 + 0{,}05x_3 + 0{,}25x_4 + 0{,}65. \end{cases}$$

№ 17

$$\begin{cases} x_1 = 0{,}24x_1 + 0{,}21x_2 + 0{,}06x_3 - 0{,}34x_4 + 1{,}42 \\ x_2 = 0{,}05x_1 + 0{,}32x_2 + 0{,}12x_4 + 0{,}57 \\ x_3 = 0{,}35x_1 - 0{,}27x_2 - 0{,}05x_4 + 0{,}68 \\ x_4 = 0{,}12x_1 - 0{,}43x_2 + 0{,}04x_3 - 0{,}21x_4 - 2{,}14. \end{cases}$$

№ 18

$$\begin{cases} x_1 = 0{,}17x_1 + 0{,}27x_2 - 0{,}13x_3 - 0{,}11x_4 - 1{,}42 \\ x_2 = 0{,}13x_1 - 0{,}12x_2 + 0{,}09x_3 - 0{,}06x_4 + 0{,}48 \\ x_3 = 0{,}11x_1 + 0{,}05x_2 - 0{,}02x_3 + 0{,}12x_4 - 2{,}34 \\ x_4 = 0{,}13x_1 + 0{,}18x_2 + 0{,}24x_3 + 0{,}43x_4 + 0{,}72. \end{cases}$$

№ 19

$$\begin{cases} x_1 = 0{,}15x_1 + 0{,}05x_2 - 0{,}08x_3 + 0{,}14x_4 - 0{,}48 \\ x_2 = 0{,}32x_1 - 0{,}13x_2 - 0{,}12x_3 + 0{,}11x_4 + 1{,}24 \\ x_3 = 0{,}17x_1 + 0{,}06x_2 - 0{,}08x_3 + 0{,}12x_4 + 1{,}15 \\ x_4 = 0{,}21x_1 - 0{,}16x_2 + 0{,}36x_3 - 0{,}88. \end{cases}$$

№ 20

$$\begin{cases} x_1 = 0{,}28x_1 - 0{,}17x_2 + 0{,}06x_4 + 0{,}21 \\ x_2 = 0{,}52x_1 + 0{,}12x_2 + 0{,}17x_4 - 1{,}17 \\ x_3 = 0{,}17x_1 - 0{,}18x_2 + 0{,}21x_3 - 0{,}81 \\ x_4 = 0{,}11x_1 + 0{,}22x_2 + 0{,}03x_3 + 0{,}05x_4 + 0{,}72. \end{cases}$$

Лабораторна робота

Чисельне вирішення нелінійних рівнянь

Мета роботи: 1. Практичне дослідження методів чисельного вирішення нелінійних рівнянь.
2. Отримань практичних навиків роботи з системою Turbo Pascal 7.0, закріплення знань щодо використання функцій.

Зміст звіту

1. Постановка завдання. Початкові дані.
2. Аналіз рішення задачі. Алгоритм рішення (блок – схема алгоритму).
3. Текст програми.
4. Результат виконання програми.
5. Висновки по роботі.

Постановка завдання

1. Використовуючи початкові дані, а також дані таблиці 2.1, уточнити корінь рівняння f(x)=0 на заданому інтервалі [a; b] з точністю 0.00001.

Таблиця 2.1

Варіант	**Метод знаходження кореня**	**Варіант**	**Метод знаходження кореня**
1 – 3	Половинного ділення, метод дотичних	16 – 18	Половинного ділення, метод хорд
4 – 6	Метод хорд, комбінований метод	19 – 21	Метод хорд, метод дотичних
7 – 9	Половинного ділення, метод хорд	22 – 24	Половинного ділення, метод дотичних

10 – 12	Метод хорд, метод дотичних	25 – 27	Метод хорд, комбінований метод
13 – 15	Половинного ділення, комбінований метод	28 – 30	Половинного ділення, комбінований метод

2. Результатом роботи програми є:

    1) Значення кореня рівняння f(x)=0.

    2) Кількість проведених ітерацій для кожного з вказаних методів.

3. За наслідками роботи зробити виводи про переваги і недоліки кожного з використовуваних методів.

Короткі відомості з теорії

Для визначення коріння рівнянь алгебри і трансцендентних розроблені чисельні методи, засновані на уточненні значення кореня в припущенні, що на відрізку [A..B] функція Y=F(x) безперервна і має тільки один корінь. В цьому випадку значення функції на кінцях відрізання мають різні знаки.

Метод половинного ділення

Одним з поширених і простих методів є метод половинного ділення (дихотомії). Він полягає в послідовному діленні навпіл відрізання, де знаходиться корінь. При цьому аналізується зміна знаку функції на половинах відрізання, і одна з меж відрізання [A..B] переноситься в його середину. Переноситься та межа, з боку якої функція на половині відрізання знаку не міняє. Процес повторюється до тих пір, поки довжина інтервалу [A..B] не стане менше заданої погрішності E знаходження кореня.

Початковими даними для програми є:

1) Задана функція F(X) рівняння F(x)= 0.
2) Відрізок [A..B] на якому знаходиться єдиний корінь.
3) Похибка знаходження кореня E.

Алгоритм програми знаходження кореня рівняння методом половинного ділення має наступний вигляд:

1. Привласнити початкові значення проміжку: a:= .; b:= .
2. Знайти наближене значення кореня: X:= (a + b) /2

3. Якщо F(a)*F(X)>0 те a:= x інакше b:= x
4. Якщо (b – a) > E те перейти до п.2
5. Вивести значення X.

Перевагою даного методу є простота алгоритму і мінімальна кількість математичних розрахунків. Разом з цим, використання методу не завжди доцільно у зв'язку з великою кількістю ітерацій (циклів уточнення кореня). В цьому випадку користуються іншими методиками.

Метод хорд

Одним з поширених методів вирішення рівнянь алгебри і трансцендентних є метод хорд («метод помилкового положення», «метод лінійної інтерполяції»).

Ідея методу полягає в тому, що на достатньо малому проміжку [a; b] дуга кривої y=f(x) замінюється хордою, що стягує її. Як наближене значення кореня приймається точка перетину хорди з віссю OX.

Шукане значення x знаходиться по формулі:

$$x = a - \frac{f(a)\cdot(b-a)}{f(b)-f(a)}.$$

Ця формула носить назву формули методу хорд. Якщо значення кореня x нас не влаштовує, то його можна уточнити. Для цього необхідно зменшити відрізок [a; b] по наступному алгоритму: якщо знак функції в крапці a співпадає із знаком в точці x, тобто f(a)*f(x)>0, то зрушуємо крапку a, інакше зрушуємо точку b (b=x). Потім застосовуємо метод хорд до нової ділянки [a; b] і так далі Процес необхідно продовжувати до тих пір, поки отримане значення кореня не задовольнятиме заданій точності.

Для оцінки точності можна використовувати формулу ?<=$|X_n-X_{n-1}|$, де Xn і Xn-1 наближення кореня, отримані на n і n-1 кроках.

Початкові дані для програми ті ж, що для попереднього завдання.

Алгоритм програми знаходження кореня рівняння методом хорд має наступний вигляд:

1. Привласнити початкові значення проміжку: a:= .; b:= .
2. Прийняти початкове значення кореня: X:= a

3. Зберегти старе значення кореня: Xo:= X
4. Знайти нове наближене значення кореня: X:= a-F(a)*(b-a)/(F(b) -F(a))
5. Якщо F(a)*F(X)>0 те a:= x інакше b:= x
6. Якщо |X – Xo| > E те перейти до п.3
7. Вивести значення X.

Перевагою методу хорд є порівняно менша кількість ітерацій, чим в методі половинного ділення.

Метод Ньютона (метод дотичних)

Геометричний сенс методу Ньютона полягає в тому, що дуга кривої y=f(x) замінюється дотичною до цієї кривої (звідси і друга назва методу). В цьому випадку, уточнене значення кореня розраховується по формулі:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

При виборі початкового наближення необхідно керуватися правилом: за початкову точку слід вибирати той кінець відрізання [a, b], в якому знак функції співпадає із знаком другої похідної:

$$x_0 = \begin{cases} a, f(a) \cdot f''(a) > 0 \\ b, f(b) \cdot f''(b) > 0 \end{cases}$$

Уточнення проводяться до тих пір, поки не виконається умова $|X_n - X_{n-1}| <= ?$.

Алгоритм програми знаходження кореня рівняння методом дотичних має наступний вигляд:

1. Привласнити початкові значення проміжку: a:= .; b:= .
2. Якщо F(a)*F``(a)>0 те X:= a інакше X:= b
3. Зберегти старе значення кореня: Xo:= X
4. Знайти нове наближене значення кореня: X:= XO-F(Xo) /F`(Xo)
5. Якщо |X – Xo| > E те перейти до п.3
6. Вивести значення X.

Метод Ньютона дає мала кількість ітерацій, але для його роботи необхідне знаходження 1 і 2-й похідних заданої функції.

Комбінований метод хорд і дотичних

Методи хорд і дотичних дають наближення кореня з різних сторін. Тому їх часто застосовують в поєднанні один з одним, і уточнення кореня відбувається швидше.

Сенс методу полягає в тому, що ділянка [a; b] з кожним кроком ітерації зменшується з кожного краю: один кінець розраховується методом хорд, інший – методом дотичних. Вибір методу залежить від знаків функції і другої її похідної на кінцях відрізання [a; b] і визначається таким чином:

$$f(a)\cdot f''(a)>0:\begin{cases} a=a-\dfrac{f(a)}{f'(a)} \\ b=b-f(b)\cdot\dfrac{b-a}{f(b)-f(a)} \end{cases};$$

$$f(a)\cdot f''(a)<0:\begin{cases} a=a-f(a)\cdot\dfrac{b-a}{f(b)-f(a)} \\ b=b-\dfrac{f(b)}{f'(b)} \end{cases}.$$

Розрахунок ведеться до тих пір, поки не виконатися умова: |a-b|<=?.
Алгоритм програми знаходження кореня рівняння комбінованим методом має наступний вигляд:

1. Привласнити початкові значення проміжку: a:= .; b:= .
2. Якщо F(a)*F``(a) < 0 те поміняти місцями значення a і b
3. Знайти нове значення крапки a: a:= a - F(a) /F`(a)
4. Знайти нове значення точки b: b:= b - F(b)*(b-a)/(F(b) - F(a))
5. Якщо |b – a| > E те перейти до п.3
6. Знайти значення кореня: X:= (a+b) /2
7. Вивести значення X.

Контрольні питання

1. Обґрунтувати необхідність застосування чисельних методів для вирішення нелінійних рівнянь.
2. У чому полягає геометричний сенс застосування кожного з методів?
3. У чому переваги і недоліки застосування кожного з методів?

Варіанти завдання

№	Функція F(x)	Параметри			№	Функція F(x)	Параметри		
		A	B	M			A	B	M
1	$F(x) := x - \sin(x) - 0.25$	1	1.5	5	14	$F(x) := 1.8 \cdot x^2 - \sin(10 \cdot x)$	-0.4	-0.3	5
2	$F(x) := \tan(0.58 \cdot x + 0.1) - x^2$	-0.3	0.2	5	15	$F(x) := \cot(x) - \frac{x}{4}$	-1.5	-1	5
3	$F(x) := \sqrt{x} - \cos(0.387 \cdot x)$	0.5	1.5	5	16	$F(x) := \tan(0.3 \cdot x + 0.4) - x^2$	-0.6	-0.4	5
4	$F(x) := \tan(0.4 \cdot x + 0.4) - x^2$	-1	0	5	17	$F(x) := x^2 - 20 \cdot \sin(x)$	-0.3	0.3	5
5	$F(x) := \log(x) - \frac{7}{2 \cdot x + 6}$	3	4	5	18	$F(x) := \cot(x) - \frac{x}{3}$	-1.5	-1	5
6	$F(x) := \tan(0.5 \cdot x + 0.2) - x^2$	-0.3	-0.2	5	19	$F(x) := \tan(0.47 \cdot x + 0.2) - x^2$	-0.3	-0.2	5
7	$F(x) := 3 \cdot x - \cos(x) - 1$	0.4	0.7	5	20	$F(x) := x^2 + 4 \cdot \sin(x)$	-0.1	0.1	5
8	$F(x) := x + \log(x) - 0.5$	0.5	0.8	5	21	$F(x) := \cot(x) - \frac{x}{2}$	-1.5	-1	5
9	$F(x) := \tan(0.5 \cdot x + 0.1) - x^2$	-0.3	0	5	22	$F(x) := 2 \cdot x - \log(x) - 7$	3	4	5
10	$F(x) := x^2 + 4 \cdot \sin(x)$	-0.2	0.2	5	23	$F(x) := \tan(0.44 \cdot x + 0.3) - x^2$	-0.4	-0.2	5
11	$F(x) := \cot(1.05 \cdot x) - x^2$	0.5	1	5	24	$F(x) := 3 \cdot x - \cos(x) - 1$	0.5	0.7	5
12	$F(x) := \tan(0.4 \cdot x + 0.3) - x^2$	-0.5	-0.2	5	25	$F(x) := \cot(x) - \frac{x}{10}$	-1.5	-1	5
13	$F(x) := x \cdot \log(x) - 1.2$	1	3	5					

Лабораторна робота

Чисельна інтеграція функцій

Мета роботи: Навчитися знаходити певний інтеграл функції методами прямокутників, трапецій і парабол.

Зміст звіту

1. Постановка завдання. Початкові дані.
2. Аналіз рішення задачі. Алгоритм рішення (блок – схема алгоритму).
3. Текст програми.
4. Результат виконання програми.
5. Висновки по роботі.

Постановка завдання

1. Знайти певний інтеграл функції f(x) на відрізок [a; b] використовуючи метод прямокутників, трапецій і парабол (метод Сімпсона). Кількість елементарних відрізань прийняти N=100.
2. Отримані дані вивести в наступному вигляді:

```
Аналітичний розрахунок:  ***
Розрахунок методом прямокутників:*** Похибка:  ***
Розрахунок методом трапецій:     ***  Похибка:  ***
Розрахунок методом Сімпсона:     ***  Похибка:  ***
```

3. Аналізуючи набутих значень, зробити висновки про точність кожного методу.

Короткі теоретичні відомості

Завдання чисельної інтеграції полягає в наступному: знайти певний інтеграл на заданому відрізку [a; b], якщо відома підінтегральна функція. Формули наближеної інтеграції називають квадратурними формулами. Простими з них є: метод прямокутників, метод трапецій, метод парабол (метод Сімпсона). Існують і складніші методики, що забезпечують вищу точність інтеграції (квадратурні формули Чебишева, Гауса і так далі). Початковими даними для завдання є:

1. Підінтегральна функція y = f(x)
2. Відрізок [a, b]
3. Кількість елементарних ділянок N (або крок інтеграції h).

Метод прямокутників

Суть методу прямокутників полягає в наступному. Хай задана деяка функція f(x) для якої необхідно знайти певний інтеграл на ділянці [a, b]. Розіб'ємо дану ділянку на n рівних проміжків. На кожній ділянці приймемо значення функції постійним і рівним її значенню на початку, середині або в кінці проміжку. Залежно від наближення, методи носять назви лівих, середніх або правих прямокутників.

Знаходження інтеграла в даному випадку зводиться до знаходження площі отриманих прямокутників. Наприклад, для методу лівих прямокутників, отримаємо формулу:

$$\int_a^b f(x)dx = (x_1 - x_0)\cdot y_0 + (x_2 - x_1)\cdot y_1 + \ldots + (x_n - x_{n-1})\cdot y_{n-1} = \frac{b-a}{n}(y_0 + y_1 + \ldots + y_{n-1}) = h\cdot\sum_{i=0}^{n-1} y_i$$

де h – крок інтеграції.

Алгоритм знаходження інтеграла методом прямокутників:

1. Привласнити початкові значення: a:= .; b:= .; N:= .;
2. Розрахувати крок інтеграції: h:= (b-a) /N
3. Для I від 0 до N-1 виконати S:= S + F(a + h*i)
4. Розрахувати інтеграл: S:= S*h
5. Вивести результат S

Метод трапецій

Замінимо на відрізку [a, b] дугу графіка інтегральної функції f(x) хордою, що стягує її, і обчислимо площу трапеції, що вийшла. Приймемо значення певного інтеграла чисельно рівним площі цієї трапеції:

$$\int_a^b f(x)dx \approx (b-a)\cdot\frac{f(a)+f(b)}{2}$$

Точність обчислень зростає, якщо відрізок [a, b] розділити на декілька частин і застосувати формулу трапецій для кожного відрізання. Для простоти обчислень зручно ділити на рівні відрізки. В цьому випадку чисельне значення інтеграла визначатиметься:

$$\int_a^b f(x)dx = \frac{b-a}{n}\cdot\sum_{i=0}^{n-1}\frac{y_i + y_{i+1}}{2} = h\cdot\left(\frac{y_0}{2} + y_1 + y_2 + \ldots + y_{n-1} + \frac{y_n}{2}\right)$$

де h = (b-a)/n – крок інтеграції.

Алгоритм знаходження інтеграла методом трапецій:

1. Привласнити початкові значення: a:= .; b:= .; N:= .;
2. Розрахувати крок інтеграції: h:= (b-a) /N
3. Розрахувати початкове значення S: S:= (f(a)+ f(b)) /2
4. Для I від 1 до N-1 виконати S:= S + F(a + h*i)
5. Розрахувати інтеграл: S:= S*h
6. Вивести результат S

Метод Сімпсона

Даний метод дає помітно вищу точність в порівнянні з методами прямокутників і трапецій. Крім того, є досить простим і зручним для програмування.

Суть методу полягає в тому, що підінтегральна функція y = f(x) на відрізку [a; b] замінюється квадратичною функцією. Як інтерполяційний многочлен використовується многочлен Ньютона. Виходячи з цього, отримаємо вираз:

$$\int_a^b f(x)dx = \frac{h}{6} \cdot \left(f(a) + 4 \cdot \frac{f(a) + f(b)}{2} + f(b) \right).$$

інтеграла на кожній ділянці довжини 2h:

$$\int_{x_0}^{x_2} f(x)dx \approx \frac{h}{3} \cdot (y_0 + 4 \cdot y_1 + y_2),$$

$$\int_{x_2}^{x_4} f(x)dx \approx \frac{h}{3} \cdot (y_2 + 4 \cdot y_3 + y_4),$$

. . . .

$$\int_{x_{2n-2}}^{x_{2n}} f(x)dx \approx \frac{h}{3} \cdot (y_{2n-2} + 4 \cdot y_{2n-1} + y_{2n}).$$

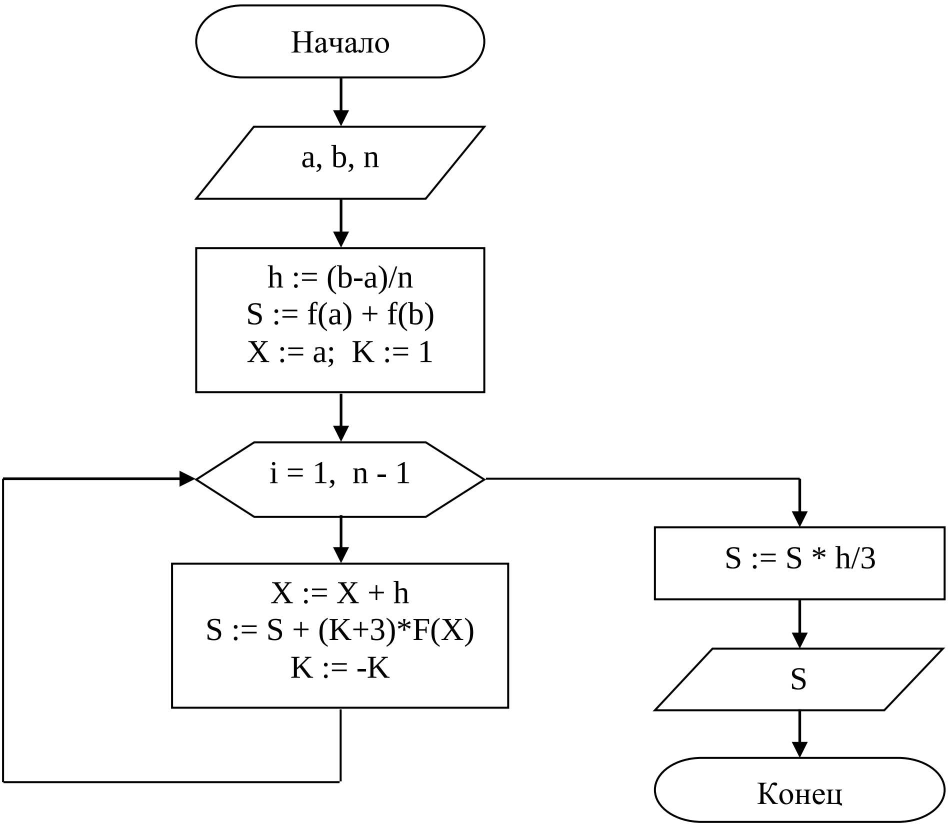
Тоді чисельне значення певного інтеграла на відрізку [a; b] буде рівне сумі інтегралів, тобто

$$\int_a^b f(x)dx \approx \frac{h}{3} \cdot (y_0 + 4 \cdot y_1 + 2 \cdot y_2 + 4 \cdot y_3 + 2 \cdot y_4 + ... + 4 \cdot y_{2n-1} + y_{2n}),$$

де $h = \frac{b-a}{2n}$.

Дане співвідношення називається загальною формулою Сімпсона.

Блок-схема алгоритму знаходження інтеграла методом Сімпсона:



Методичні вказівки

1. Зручно оформити розрахунки за допомогою кожного методу у вигляді окремої функції. В цьому випадку заголовок такої функції матиме приблизно такий вигляд:

   ```
   Function Integral(a, b: real; N: integer):
   real;
   ```
   де a, b – діапазон інтеграції

   N – кількість відрізань.
2. Для розрахунку погрішності кожного методу необхідно заздалегідь провести аналітичний розрахунок певного інтеграла і задати його значення в програмі у вигляді константи. Погрішність знаходиться таким чином: E =

|Ian - Iras|, де Ian – аналітичне значення інтеграла, Iras – розрахункове.

Контрольні питання

1. У чому полягає математична постановка завдання інтеграції функції?
2. Геометричний сенс різних методів інтеграції.
3. Переваги і недоліки кожного методу інтеграції.
4. Від чого залежить точність розрахунку інтеграла?
5. Як на практиці оцінюється погрішність кожного з методів?

Лабораторна робота

Інтерполяція функцій

Мета роботи: Практичне дослідження інтерполяції функцій за допомогою інтерполяційного многочлена Лагранжа.

Зміст звіту

1. Постановка завдання. Початкові дані.
2. Аналіз рішення задачі. Алгоритм рішення (блок – схема алгоритму).
3. Текст програми.
4. Результат виконання програми.
5. Висновки по роботі.

Постановка завдання

1. Використовуючи таблицю функції f(x), отриману в лабораторній роботі №1, побудувати інтерполяційний многочлен Лагранжа L(x).
2. Побудувати таблицю функції L(x), розбивши відрізок [a; b] на 2M відрізань (дані див. в роботі №1).
3. Отримані дані вивести у вигляді наступної таблиці:

X	F(x)	L(x)	E
a	F(a)	L(a)	|F(a) – L(a)|
.	.	.	.
b	F(b)	L(b)	|F(b) – L(b)|

4. Аналізуючи значення E зробити виводи про точність методу.

Методичні вказівки

1. Для розрахунку значень f(x) і L(x) в програмі зручно користуватися підпрограмами – функціями.
2. Многочлен Лагранжа розраховується по формулі:

$$L(x) = y_0 \frac{(x-x_1)\dots(x-x_n)}{(x_0-x_1)\dots(x_0-x_n)} + y_1 \frac{(x-x_0)(x-x_2)\dots(x-x_n)}{(x_1-x_0)(x_1-x_2)\dots(x_1-x_n)} + \dots$$

$$\dots + y_n \frac{(x-x_0)(x-x_1)\dots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\dots(x_n-x_{n-1})}$$

3. Табличні значення функції $f(x_i)$ і значення аргументу xi зручно задати у вигляді масивів-констант, а не підставляти готові значення у формулу 6.1.

Короткі теоретичні відомості

У економіці і техніці постійно доводиться стикатися з необхідністю обчислення значень функції y=f(x) в точках x, відмінних від значень аргументу, фіксованих в таблиці. Крім того, в деяких випадках, не дивлячись на те, що аналітичне значення функції y=f(x) відоме, воно є дуже складним і незручним для подальших математичних перетворень. Подібні завдання практики формалізуються як математичні завдання інтерполяції.

Хай на відрізку [a; b] задана функція y=f(x) своїми n+1 значеннями:

$$y_0 = f(x_0);\ y_1 = f(x_1);\ \dots;\ y_n = f(x_n)$$

в точках $x_0, x_1, \dots, x_n$, які називають вузлами інтерполяції. Потрібно знайти аналітичний вираз F(x) табульованої функції співпадаючої у вузлах інтерполяції із значеннями заданої функції, тобто

$$y_0 = F(x_0) = f(x_0),\ y_1 = F(x_1) = f(x_1),\ \dots,\ y_n = F(x_n) = f(x_n).$$

Процес обчислення значень функції в точках x, відмінних від вузлів інтерполяції, називають інтерполяцією функції f(x).

Геометрично завдання інтерполяції для функції однієї змінної y=f(x) означає побудову кривої, площині, що проходить через крапки, з координатами $(x_0;\ y_0)$, $(x_1;\ y_1)$, …, $(x_n;\ y_n)$. Це завдання стає однозначним, якщо як інтерполююча функція F(x) для функції y=f(x), заданою n+1 своїми значеннями, вибрати многочлен Fn(x) ступеня не вище n, такий, що

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

Многочлен Fn(x), що задовольняє цим умовам, називають інтерполяційним многочленом, а відповідні формули – інтерполяційними формулами. У разі, коли F(x) вибирається в класі статечних функцій, інтерполяція називається параболічною.

Найбільш загальною формулою параболічної інтерполяції є інтерполяційна формула Лагранжа. В цьому випадку інтерполяційний многочлен має загальний вигляд:

$L(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$,

де n – кількість вузлів інтерполяції;

a_i – (i = 0, 1 ., n) невідомі постійні коефіцієнти, які треба знайти.

Остаточно многочлен Лагранжа розраховується по формулі:

$$L(x) = y_0 \frac{(x-x_1)\ldots(x-x_n)}{(x_0-x_1)\ldots(x_0-x_n)} + y_1 \frac{(x-x_0)(x-x_2)\ldots(x-x_n)}{(x_1-x_0)(x_1-x_2)\ldots(x_1-x_n)} + \ldots$$

$$\ldots + y_n \frac{(x-x_0)(x-x_1)\ldots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\ldots(x_n-x_{n-1})}$$

де $x_0 - x_n$, $y_0 - y_n$ – значення таблиці функції, отримані заздалегідь.

Контрольні питання

1. У чому полягає математична постановка завдання інтерполяції функції?
2. Види інтерполяцій. Доцільність застосування різних видів інтерполяції.
3. Поняття кроку інтерполяції.

Лабораторна робота

Чисельне диференціювання функцій

Мета роботи: Отримати практичні навики диференціювання функцій з використанням інтерполяційних формул Ньютона.

Зміст звіту

1. Постановка завдання. Початкові дані.
2. Аналіз рішення задачі. Алгоритм рішення (блок – схема алгоритму).
3. Текст програми.
4. Результат виконання програми.
5. Висновки по роботі.

Постановка завдання

1. Знайти першу і другу похідну функції f(x) для початкової і кінцевої точки таблиці.
2. Отримані дані вивести в наступному вигляді:

x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$
.	.	.	.	.

```
Аналітичний розрахунок   f´(x0) = …f´´(x0) = …
                         f´(xn) = …f´´(xn) = …


Чисельний метод          f´(x0) = …f´´(x0) = …
                         f´(xn) = …f´´(xn) = …
```

3. Аналізуючи набутих значень зробити виводи про точність чисельного методу розрахунку для першої і другої похідної.

Короткі теоретичні відомості

При вирішенні завдань часто доводиться обчислювати похідну, проте в багатьох практичних завданнях функції задаються табличний і методи диференціального числення до дослідження таких функцій застосувати не можна. В цьому випадку удаються до чисельного диференціювання.

Завдання чисельного диференціювання ставиться таким чином. Хай функція y = f(x) на відрізку [a, b] задана табличний своїми n + 1 значеннями. Потрібно знайти аналітичний вираз похідній.

Як апроксимуюча функція виберемо інтерполяційний многочлен. Якщо вузли інтерполяції в початковому завданні рівновіддалені, тобто $x_{i+1}-x_i = h$ (де i = 0, 1 ..., n), то в цьому випадку для заміни функції f(x) можна скористатися інтерполяційними формулами Ньютона. Тоді

$$f(x) \approx y_0 + q\Delta y_0 + \frac{q^2-q}{2}\Delta^2 y_0 + \frac{q^3-3q^2+2q}{6}\Delta^3 y_0 + \frac{q^4-6q^3+11q^2-6q}{24}\Delta^4 y_0 + \ldots ,$$

де $q = (x-x_0)/h$ і $h = x_{i+1}-x_i$; Δy_i, Δy^1_i, Δy^2_i, Δy^3_i – кінцеві різниці (див. далі).

З урахуванням того, що $\frac{df(x)}{dx} = \frac{df(x)}{dq}\cdot\frac{dq}{dx} = \frac{1}{h}\frac{df(x)}{dq}$,

набудемо значень похідних:

$$f'(x) \approx \frac{1}{h}\left[\Delta y_0 + \frac{2q-1}{2}\Delta^2 y_0 + \frac{3q^2-6q+2}{6}\Delta^3 y_0 + \frac{2q^3-9q^2+11q-3}{12}\Delta^4 y_0 + \ldots\right],$$

$$f''(x) \approx \frac{1}{h^2}\left[\Delta^2 y_0 + (q-1)\Delta^3 y_0 + \frac{6q^2-18q+11}{12}\Delta^4 y_0 + \ldots\right].$$

Формули наближеного диференціювання для визначення похідних у вузлах інтерполяції значно спрощуються. Вважаючи q = 0, отримаємо

$$f'(x) \approx \frac{1}{h}\left[\Delta y_0 - \frac{\Delta^2 y_0}{2} + \frac{\Delta^3 y_0}{3} - \frac{\Delta^4 y_0}{4} + \frac{\Delta^5 y_0}{5} \ldots\right],$$

$$f''(x) \approx \frac{1}{h}\left[\Delta^2 y_0 - \Delta^3 y_0 + \frac{11}{12}\Delta^4 y_0 - \frac{5}{6}\Delta^5 y_0 + \ldots\right].$$

Щоб набути значення похідної в крапці, лежачої в кінці таблиці, слід скористатися другою інтерполяційною формулою Ньютона. Тоді наближене значення для похідної є:

$$f'(x) \approx \frac{1}{h}\left[\Delta y_{n-1} + \frac{2q+1}{2}\Delta^2 y_{n-2} + \frac{3q^2+6q+2}{6}\Delta^3 y_{n-3} + \frac{2q^3+9q^2+11q+3}{12}\Delta^4 y_{n-1} + \ldots\right]$$

$$f''(x) \approx \frac{1}{h^2}\left[\Delta^2 y_{n-2} + (q+1)\Delta^3 y_{n-3} + \frac{6q^2+18q+11}{12}\Delta^4 y_{n-4} + \ldots\right].$$

Формули наближеного диференціювання для визначення похідних у вузлах інтерполяції:

$$f'(x) \approx \frac{1}{h}\left[\Delta y_{n-1} + \frac{\Delta^2 y_{n-2}}{2} + \frac{\Delta^3 y_{n-3}}{3} + \frac{\Delta^4 y_{n-4}}{4} + \frac{\Delta^5 y_{n-5}}{5} \ldots\right],$$

$$f''(x) \approx \frac{1}{h}\left[\Delta^2 y_{n-2} + \Delta^3 y_{n-3} + \frac{11}{12}\Delta^4 y_{n-4} + \frac{5}{6}\Delta^5 y_{n-5} + \ldots\right].$$

Кінцеві різниці знаходяться по формулах:

$\Delta y_i = y_{i+1} - y_i$,

$\Delta y^1_i = \Delta y_{i+1}, - \Delta y_i$,

$\Delta y^2_i = \Delta y^1_{i+1} - \Delta y^1_i$,

...

Методичні вказівки

1. Для отримання похідної точки, лежачої на початку таблиці необхідно скористатися першою інтерполяційною формулою Ньютона, а в кінці таблиці – другою формулою.
2. Кількість кінцевих різниць вибрати n=M-1 (Δyi, Δ^{2yi}., Δ^{nyi}), де M – кількість точок таблиці.
3. Для розрахунку погрішності кожного методу необхідно заздалегідь провести аналітичний розрахунок похідних

заданої функції на краях відрізання [a; b] і задати значення в програмі у вигляді констант.

Приклад виконання завдання

```
uses CRT;
const N = 4; {M=N+1=5}

var   D: array [1..N, 0..N-1] of real; {Delta}
      i, j: integer;
      B1, B2, E1, E2, h: real;
{Таблиця функції}
const X: array [0..N] of real = (0, 0.39270, 0.78540,
0.17810, 1.57080);
      Y: array [0..N] of real = (0, 0.38268, 0.70711,
0.92388, 1);
begin
  ClrScr;
  h:= X[2]-X[1];   {Розраховуємо крок}
    {Заповнюємо таблицю кінцевих різниць}
  for i:= 0 to N-1 do D[1, i]:= Y[i+1]-Y[i];
  for i:= 2 to N do
  for j:= 0 to N-i do D[i, j]:= D[i-1, j+1] - D[i-1, j];
  { Розраховуємо 1-у і 2-у похідні для першої крапки
функції}
    { використовуючи першу формулу Ньютона}
  B1:= 1/h*(D[1,0]-0.5*D[2,0]+1/3*D[3,0]-0.25*D[4,0]);
  B2:= 1/h/h*(D[2,0]-D[3,0]+11/12*D[4,0]);
    { Розраховуємо 1-у і 2-у похідні для останньої крапки
функції}
    { використовуючи другу формулу Ньютона}
  E1:= 1/h*(D[1,3]+0.5*D[2,2]+1/3*D[3,1]+0.25*D[4,0]);
  E2:= 1/h/h*(D[2,2]+D[3,1]+11/12*D[4,0]);
    { Виводимо результат на екран}
 writeln('Xi':10, 'Yi':10, 'D':10, 'D1':10, 'D2':10,
'D3':10);
  writeln('-----------------------------------------------
--------------');
  for i:= 0 to n do
  begin
    write(X[i]:10:5, Y[i]:10:5);
    for j:=1 to N-i do write(D[j, i]:10:5);
    writeln;
  end;
  writeln;
```

```
  writeln('Аналітичний розрахунок F1(a)=',1.:10:5, '
F2(a)=',0.:10:5);
  writeln('                     F1(b)=',0.:10:5, '
F2(b)=',-1.:10:5);
  writeln;
  writeln('Чисельний розрахунок     F1(a)=',B1:10:5, '
F2(a)=',B2:10:5);
  writeln('                     F1(b)=',E1:10:5, '
F2(b)=',E2:10:5);
end.
```

Протокол виконання програми

```
        Xi         Yi          D         D1         D2
D3
-----------------------------------------------------------
---
   0.00000   0.00000   0.38268  -0.05825  -0.04941
0.01642
   0.39270   0.38268   0.32443  -0.10766  -0.03299
   0.78540   0.70711   0.21677  -0.14065
   0.17810   0.92388   0.07612
   1.57080   1.00000

Аналітичний розрахунок F1(a)=   1.00000  F2(a)=   0.00000
                       F1(b)=   0.00000  F2(b)=  -1.00000

Чисельний розрахунок   F1(a)=   0.99626  F2(a)=   0.04028
                       F1(b)=  -0.00279  F2(b)=  -1.02837
```

Контрольні питання

1. У чому полягає математична постановка завдання диференціювання функції?
2. Як розраховуються кінцеві різниці?
3. Методика розрахунку похідних в початкових і кінцевих точках таблиці функції.

Лабораторна робота

Наближені методи вирішення звичайних диференційних рівнянь

М е т а   р о б о т и: Отримати практичні навики вирішення звичайних диференціальних рівнянь методами Ейлера і Рунге-Кутта.

Зміст звіту

1. Постановка завдання. Початкові дані.
2. Аналіз рішення задачі. Алгоритм рішення (блок – схема алгоритму).
3. Текст програми.
4. Результат виконання програми.
5. Висновки по роботі.

Постановка завдання

1. Використовуючи метод Ейлера і метод Рунге-Кутта, скласти таблицю наближених значень інтеграла диференціального рівняння y\`=f(x, y), що задовольняє початковим умовам y(x0)=y_0 на відрізку [a; b]; крок h=0,1. Всі обчислення вести з чотирма десятковими знаками. Початкові дані для розрахунків узяти з таблиці 6.1.
2. Вивести отримані дані на екран в наступному вигляді:

x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$
.	.	.	.	.

де Y1 – рішення, отримане методом Ейлера;
Y2 – рішення, отримане методом Рунге-Кутта;

E – погрішність розрахунку методом Ейлера.

3. Аналізуючи набутих значень зробити виводи про точність чисельного методу розрахунку методом Ейлера залежно від номера кроку інтеграції.

Короткі теоретичні відомості

Вирішити диференціальне рівняння виду $y' = f(x,y)$ чисельним методом – це означає для заданої послідовності аргументів $x_0, x_1, \ldots, x_n$ і числа y_0, не визначаючи функцію $y=F(x)$, знайти такі значення $y_1, y_2, \ldots, y_n$, що $y_i = F(x_i)$ $(i=1,2,\ldots,n)$ і $F(x_0)=y_0$.

Таким чином, чисельні методи дозволяють замість знаходження функції $y=F(x)$ отримати таблицю значень цієї функції для заданої послідовності аргументів. Величина $h=x_k-x_{k-1}$ називається кроком інтеграції.

Розглянемо деякі з чисельних методів.

Метод Ейлера

Даний метод є порівняно грубим і застосовується в основному для орієнтовних розрахунків. Проте ідеї, покладені в основу методу Ейлера, є початковими для ряду інших методів.

Хай дано диференціальне рівняння першого порядку

$$y' = f(x,y)$$

з початковою умовою

$$x = x_0,\ y(x_0) = y_0.$$

Розіб'ємо ділянку $[a,\ b]$ на n рівних частин і отримаємо послідовність $x_0, x_1, \ldots, x_n$,

де $x_i = x_o + ih$ $(i=0,1,2,\ldots,n)$, а $h=(b-a)/n$ – крок інтеграції.

Виберемо к-й ділянку $[x_k,\ x_{k+1}]$ і проінтегруємо рівняння (1):

$$\int_{x_k}^{x_{k+1}} f(x,y)dx = \int_{x_k}^{x_{k+1}} y'dx = y(x)\Big|_{x_k}^{x_{k+1}} = y(x_{k+1}) - y(x_k) = y_{k+1} - y_k\ ,$$

тобто

$$y_{k+1} = y_k + \int_{x_k}^{x_{k+1}} f(x, y)dx.$$

Якщо в останній ділянці інтегральну функцію на ділянці [x_k, x_{k+1}] прийняти постійною і рівною початковому значенню в точці $x=x_k$, то отримаємо

$$\int_{x_k}^{x_{k+1}} f(x_k, y_k)dx = f(x_k, y_k)\cdot x \Big|_{x_k}^{x_{k+1}} = f(x_k, y_k)\cdot(x_{k+1} - x_k) = y'_k \cdot h.$$

Тоді формула (3) прийме вигляд

$$y_{k+1} = y_k + y'_k \cdot h$$

Позначивши $y_{k+1} - y_k = \Delta y_k$, тобто $y'_k\, h = \Delta y_k$, отримаємо

$$y_{k+1} = y_k + \Delta y_k.$$

Продовжуючи цей процес і кожного разу приймаючи інтегральну функцію на відповідній ділянці постійною і рівною її значенню на початку ділянки, отримаємо таблицю вирішень диференціального рівняння на заданій ділянці [*a, b*].

Для оцінки погрішності на практиці, як правило, використовують метод «подвійного прорахунку». Спочатку розрахунок ведеться з кроком h, потім крок дроблять і повторний розрахунок ведеться з кроком h/2. погрішність точнішого значення y^*_n оцінюється формулою

$$\left|y_n^* - y(x_n)\right| \approx \left|y_n^* - y_n\right|.$$

Метод Рунге-Кутта

Метод Рунге-Кутта є одним з методів підвищеної точності. Він має багато загального з методом Ейлера.

Хай на відрізку [*a, b*] потрібно знайти чисельне вирішення рівняння (1) з початковою умовою (2).

Розіб'ємо ділянку [*a, b*] на n рівних частин точками $x_i = x_o + ih$ $(i = 0, 1, 2, \ldots, n)$, де $h = (b-a)/n$ – крок інтеграції. У методі Рунге-Кутта, так само і в методі Ейлера, послідовні значення *yi* шуканої функції *y* визначають по формулі (4).

Якщо розкласти функцію в ряд Тейлора і обмежитися членами до h^4 включно, то приріст функції sy можна представити у вигляді:

$$\Delta y = y(x+h) - y(x) = hy'(x) + \frac{h^2}{2} y''(x) + \frac{h^3}{6} y'''(x) + \frac{h^4}{24} y^{IV}(x),$$

де $y''(x)$, $y'''(x)$, $y^{IV}(x)$) – визначаються послідовним диференціюванням з рівняння (1).

Замість безпосередніх обчислень за формулою (6) в методі Рунге-Кутта визначаються чотири числа:

$$k_1 = h \cdot f(x, y) \qquad k_2 = h \cdot f(x + \frac{h}{2}, y + \frac{k_1}{2})$$

$$k_3 = h \cdot f(x + \frac{h}{2}, y + \frac{k_2}{2}) \quad k_4 = h \cdot f(x + h, y + k_3)$$

При цьому, значення Δy, обчислене по формулі (6), з точністю до четвертих ступенів буде рівне:

$$\Delta y = \frac{1}{6}(k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4)$$

Таким чином, для кожної пари поточних значень xi, yi по формулі (7) визначаються значення:

$$k_1^{(i)} = h \cdot f(x_i, y_i) \qquad k_2^{(i)} = h \cdot f(x_i + \frac{h}{2}, y_i + \frac{k_1^{(i)}}{2})$$

$$k_3^{(i)} = h \cdot f(x_i + \frac{h}{2}, y_i + \frac{k_2^{(i)}}{2}) \quad k_4^{(i)} = h \cdot f(x_i + h, y_i + k_3^{(i)})$$

по формулі (8) знаходиться, а потім $y_{i+1} = y_i + \Delta y_i$.

Даний метод має порядок точності h^4 на всьому відрізку [a, b]. Оцінка точності методу дуже скрутна. Грубу оцінку погрішності можна отримати за допомогою «подвійного прорахунку» по формулі

$$\varepsilon \approx \frac{y_i^* - y_i}{15},$$

де y^*_i і y_i – наближені значення, отримані з кроком $h/2$ і h. Якщо ε – задана точність рішення, те число n (число ділень) вибирається так, щоб $h^4 < \varepsilon$, тобто

$$n > \frac{b-a}{\sqrt[4]{\varepsilon}}$$

Методичні вказівки

1. Погрішність методу Ейлера розраховується з припущення, що точність методу Рунге-Кутта на багато вище, і з точністю до четвертого десяткового знаку погрішність розрахунку рівна нулю. Виходячи з цього припущення, $Ei = |Y_2(xi)-Y_1(xi)|$.
2. Вирішення диференціальних рівнянь зручно оформити у вигляді відповідних функцій.

Приклад написання програм

Завдання: Вирішити диференціальне рівняння на проміжку [1,6; 2,6] з кроком h=0,25 методом Ейлера і методом Рунге-Кутта. Початкові умови Xo=1,6; Yo=-4,6.

```
Program Eiler;
Uses CRT;
var   X, Y: array [0..20] of real;
      i, N: integer;
const A  = 1.6;  B  = 2.6; { Діаппазон інтеграції X=a,b] }
      Xo = A;    Yo = -4.6; { Початкові умови Yo = F(Xo) }
   h  = 0.25;            { Крок інтеграції h             }

function F1(x,y: real): real; { Визначаємо функцію Y' =
F(X, Y)}
begin
  F1:=x+cos(x/3)
end;
begin
 ClrScr;
 X[0]:=Xo; Y[0]:=Yo;  { Установка початкових значень (крок
i = 0) }
 N:= round((b-a)/h);
  for i:=1 to N do
  begin
    X[i]:=X[i-1]+h;
    Y[i]:=Y[i-1]+h*F1(X[i-1], Y[i-1]);
  end;
 writeln('i':5,'X':10,'Y':10);
 writeln('---------------------------------');
 for i:=0 to round((B-A)/h) do
```

```
     writeln(i:5,X[i]:10:4,Y[i]:10:4);
 writeln;
end.
```

Протокол виконання програми

```
    i         X         Y
--------------------------------
    0    1.6000   -4.6000
    1    1.8500   -3.9847
    2    2.1000   -3.3183
    3    2.3500   -2.6021
    4    2.6000   -1.8374
```

```
Program Metod_Runge_Kutta; { Вирішення диференціального
рівняння методом Рунге-Кутта}
Uses CRT;
var   X, Y: array [0..20] of real;
      i: integer;

const A  = 1.6;  B  = 2.6;   { Діаппазон інтеграції X =
[a,b]    }
   Xo = A;    Yo = -4.6;  { Початкові умови Yo = F(Xo)
}
   h  = 0.25;             { Крок інтеграції h
}

function F1(x,y: real): real; far; { Визначаємо функцію Y'
= F(X, Y)}
begin
  F1:=x+cos(x/3)
end;
function DY(X,Y,h: real): real;  { Допоміжна функція
обчислення  }
var; K1, K2, K3, K4: real;       { елементарного приросту
функції DY }
begin                            { на i-м кроці інтеграції
методом  }
 K1:= h*F1(X,     Y);                       { Рунге-Кутта}
 K2:= h*F1(X+h/2, Y+K1/2);
  K3:= h*F1(X+h/2, Y+K2/2);
  K4:= h*F1(X+h,   Y+K3);
  DY:= (K1+2*K2+2*K3+K4) /6;
end;
begin
  ClrScr;    { Очищення екрану }
```

```
 X[0]:=Xo; Y[0]:=Yo;          { Установка початкових
значень (крок i = 0) }
  i:=1;                        { Перехід на наступний крок
(i = 1) }
  while i<=round((B-A)/h) do   { Поки не пройдені всі
кроки ... }
  begin
    X[i]:=X[i-1]+h;                    { Збільшуємо X на
величину кроку h }
  Y[i]:=Y[i-1]+DY(X[i-1], Y[i-1], h);{ Обчислюваний Yi }
  inc(i)                             { Збільшуємо номер
кроку на 1 }
 end;
  writeln('i':5,'X':10,'Y':10);
  writeln('--------------------------------');
  for i:=0 to round((B-A)/h) do
     writeln(i:5,X[i]:10:4,Y[i]:10:4);
end.
```

Протокол виконання програми

```
    i         X         Y
-------------------------------
    0    1.6000   -4.6000
    1    1.8500   -3.9590
    2    2.1000   -3.2676
    3    2.3500   -2.5270
    4    2.6000   -1.7387
```

Контрольні питання

1. Що є вирішенням звичайного диф. рівняння?
2. У чому полягають переваги і недоліки кожного методу вирішення диференціальних рівнянь?
3. Як точність розрахунків методом Ейлера залежить від кроку інтеграції?
4. Яким чином визначається погрішність розрахунків в даній роботі?
5. Як на практиці знаходять погрішність розрахунків?
6. Від чого залежить точність вирішення диференціальних рівнянь?

Таблиця – Початкові дані

№	Рівняння	y0(a)	x∈[a; b]	№	Рівняння	y0(a)	x∈[a; b]
1	$y` = x + \cos\frac{y}{\sqrt{5}}$	2,6	[1,8; 2,8]	16	$y` = x + \sin\frac{y}{\sqrt{2,8}}$	2,2	[1,4; 2,4]
2	$y` = x + \cos\frac{y}{3}$	4,6	[1,6; 2,6]	17	$y` = x + \sin\frac{y}{e}$	2,5	[1,4; 2,4]
3	$y` = x + \cos\frac{y}{\sqrt{10}}$	0,8	[0,6; 1,6]	18	$y` = x + \sin\frac{y}{\sqrt{2}}$	1,3	[0,8; 1,8]
4	$y` = x + \cos\frac{y}{\sqrt{7}}$	0,6	[0,5; 1,5]	19	$y` = x + \sin\frac{y}{\sqrt{3}}$	1,5	[1,1; 2,1]
5	$y` = x + \cos\frac{y}{\pi}$	5,3	[1,7; 2,7]	20	$y` = x + \sin\frac{y}{\sqrt{11}}$	1,2	[0,6; 1,6]
6	$y` = x + \cos\frac{y}{2,25}$	2,2	[1,4; 2,4]	21	$y` = x + \sin\frac{y}{1,25}$	1,8	[0,5; 1,5]
7	$y` = x + \cos\frac{y}{e}$	2,5	[1,4; 2,4]	22	$y` = x + \sin\frac{y}{\sqrt{15}}$	1,1	[0,2; 1,2]
8	$y` = x + \cos\frac{y}{\sqrt{2}}$	1,4	[0,8; 1,8]	23	$y` = x + \sin\frac{y}{\sqrt{1,3}}$	0,8	[0,1; 1,1]
9	$y` = x + \cos\frac{y}{\sqrt{3}}$	2,1	[1,2; 2,2]	24	$y` = x + \sin\frac{y}{\sqrt{0,3}}$	0,6	[0,5; 1,5]
10	$y` = x + \cos\frac{y}{\sqrt{11}}$	2,5	[2,1; 3,1]	25	$y` = x + \sin\frac{y}{\sqrt{0,7}}$	1,4	[1,2; 2,2]
11	$y` = x + \sin\frac{y}{\sqrt{5}}$	2,6	[2,8; 2,8]	26	$y` = x + \cos\frac{y}{1,25}$	0,8	[0,4; 1,4]
12	$y` = x + \sin\frac{y}{\sqrt{3}}$	4,6	[1,6; 2,6]	27	$y` = x + \cos\frac{y}{\sqrt{1,5}}$	0,9	[0,3; 1,3]
13	$y` = x + \sin\frac{y}{\sqrt{10}}$	0,8	[0,6; 1,6]	28	$y` = x + \cos\frac{y}{\sqrt{1,3}}$	1,8	[1,2; 2,2]
14	$y` = x + \sin\frac{y}{\sqrt{7}}$	0,6	[0,5; 1,5]	29	$y` = x + \cos\frac{y}{\sqrt{0,3}}$	2,1	[0,7; 1,7]
15	$y` = x + \sin\frac{y}{\pi}$	5,3	[1,7; 2,7]	30	$y` = x + \cos\frac{y}{\sqrt{0,7}}$	1,7	[0,9; 1,9]

Список використаних джерел

1. Н.И. Данилина и др. Численные методы. Учебник для техникумов. М., «Высш. школа», 1976.
2. Калиткин Н.Н. Численные методы/Под ред. А.А. Самарского/ -М.: Наука, 1978.
3. Фельдман Л.П. та ін. Числові методи в інформатиці. Підручник. За ред. М.З. Згуровського.
4. Воробьева Г.Н., Данилова А.Н. Практикум по численным методам: Учеб. Пособие для техникумов. – М.: Высш. школа, 1979. – 148 с.