Зміст

Вступ

1. Теоретична частина

1.1 Постановка задачі

1.2 Методи розв'язування задачі

1.3 Алгоритм розв'язку задачі

2. Практична частина

2.1 Архітектура програми

2.2 Опис програми

2.3 Контрольний приклад та результат машинного експерименту

Висновок

Список використаної літератури

Додатки

Вступ

Центральним поняттям програмування є, безперечно, поняття алгоритму. З нього починається робота над програмою і від якості алгоритму залежить її успішне створення. Тому вміння програмувати в значній мірі означає розробляти хороші алгоритми і застосовувати вже відомі.

На сьогодні існує велика кількість різноманітних мов програмування, кожна з яких має свої певні переваги та недоліки. В цьому розмаїтті не завжди легко зробити свій вибір на користь якоїсь певної мови програмування.

Для реалізації поставленої задачі вибрано середовище Turbo Pascal. Алгоритмічна мова Паскаль була створена Н.Віртом на початку 70-х років. Завдяки зусиллям розробників ця мова програмування стала потужним інструментом професійних програмістів‚ не втративши простоти і ясності, властивих цій мові від народження.

Розробник системи Turbo Pascal - фірма Borland International виникла в 1984 році і за порівняно короткий час неодноразово дивувала користувачів персональних ЕОМ своїми Turbo системами. Було випущено кілька версій Turbo Pascal: 3.0‚ 4.0‚ 5.0‚ 5.5‚ 6.0‚ 7.0‚ Pascal for Windows, Borland Pascal.

Головні особливості середовища Turbo Pascal:

широкий спектр типів даних‚ можливість обробки рядкових та структурних типів даних;

достатній набір операторів управління розгалуженнями та циклами;

добре розвинутий апарат підпрограм та зручні конструкції роботи з

файлами;

великі можливості управління усіма ресурсами ПЕОМ;

різноманітні варіанти стикування з мовою Асемблера;

підтримка ідей об'єктно-орієнтованого програмування (ООП).

Саме з огляду на ці особливості програмна реалізація курсового проекту було здійснено в середовищі Turbo Pascal.

Курсовий проект складається зі вступу, двох розділів, висновків, списку використаної літератури, графічної частини та додатків. Текст пояснювальної записки набрано та роздруковано з використанням текстового редактора Word. Графічна частина виконана з допомогою графічного редактора Visio.

1. Теоретична частина

1.1 Постановка задачі

Нехай дано систему п лінійних алгебраїчних рівнянь з п змінними

(I=1.2…..n) (1)



Систему (1) можна записати у вигляді одного матричного рівняння

AX=B, (2)

де



матриця коефіцієнтів (індекс і вказує рівняння, якому належить коефіцієнт, а індекс j – змінну, при якій він стоїть),



,



відповідно стовпець вільних членів і стовпець змінних.

Упорядкована сукупність п чисел , яка, будучи підставленою в систему (1) замість , перетворює всі рівняння в правильні числові рівності, називається розв’язком системи (1)



Методи розв’язування систем лінійних рівнянь можна поділити на дві групи: точні й ітераційні.

Точними називають такі методи, які дають змогу знайти точний розв’язок системи (1) за допомогою виконання скінченої кількості арифметичних операцій у припущенні, що всі обчислення виконуються точно (без округлень), а коефіцієнти системи і вільні члени – точні числа. Але на практиці всі обчислення виконуються з обмеженою кількістю десяткових розрядів, а ірраціональні коефіцієнти і вільні члени, якщо такі є, замінюються раціональними числами. Тому в процесі обчислення вдаються до округлень, а це означає, що розв’язки, які обчислюються за точними методами, фактично є наближеними числами з певними похибками (похибками округлень). До точних належать метод Гаусса, метод квадратних коренів, правило Крамера, сюди ж належить метод Жордана-Гаусса.

Інтераційними називають такі методи, які дають змогу знайти наближений розв’язок системи (1) із заздалегідь вказаною точністю шляхом виконання скінченої кількості арифметичних операцій, хоч самі обчислення можуть проводитись і без округлень, а коефіцієнти і вільні члени системи бути точними числами.

У процесі вивчення різних питань економіки, природознавства, техніки тощо доводиться розв’язувати системи алгебраїчних рівнянь. Зокрема, до таких систем зводиться чисельне розв’язування лінійних, диференціальних та інтегральних рівнянь. У таких системає коефіцієнти і вільні члени рівнянь – числа наближені. А це веде до появи додаткових (так званих неусуваних) похибок.

Якщо систему рівнянь у пам’яті машини записати навіть точно, то в процесі її розв’язування ЕОМ обов’язково виникнуть похибки округлень, які не можуть не вплинути на точність розв’язку. Проте, якщо матриця А системи (2) майже вираджена то можна сподіватися, що малі зміни в коефіцієнтах і (або) вільних членах також призведуть до значних змін у її розв’язку.

Якщо малі збурення коефіцієнтів і (або) вільних членів системи (1) дуже збурюють її розв’язок, то таку систему рівнянь називають погано обумовленою. Наприклад, якщо малі збурення коефіцієнтів і (або) вільних членів системи (1) мало збурюють її розв’язок, то таку систему називають добре обумовленою. Прикладом погано обумовленої є, наприклад, система вигляду:

(3)



розв’язком якої є пара (1;0). Якщо число 6,1 у правій частині першого рівняння системи (3) змінити на 0,02, то система



матиме розв’язком пару (5,1;-7,35). Отже, мале збурення (меньше 0,33%) одного з вільних членів системи (3) зовсім змінило розв’язок системи.

На щастя, на практиці системи рівнянь, погано обумовлені, зустрічаються дуже рідко.

1.2 Методи розв’язування задачі

Метод Жордана-Гаусса був розроблений двома вченими Жорданом та Гаусом (ві яких і пішла назва методу). Цей метод вони помітили після довгої практики роботи з системами рівнянь. Це можна пояснити складністю розв’язку цим методом.

Суть методу заключається в тому, щоб послідовно вилучати один, за одним стовпці елементів квадратної матриці, які стоять біля відповідних невідомих. Це вилучення повинно відбуватися за посередництвом деякого елемента входящого у вилучаючий стовпець. Цей основний елемент, відповідно зв’язує вилучаючий стовпець з деяким рядком системи.

При кожному вилученні стовпців необхідно проводити такі зміни в системі, щоб в кінцевому рузультаті число, яке опиняється в деякому рядку відповідало невідомій вилученого стовпця, зв’язаного один раз з даним рядком через основний елемент.

Проводити ці зміни в системі вдається за формулою обчислення двомірного визначника



шляхом вибору із системи певних елементів для обчислення визначника. Кількість обчислень таких визначників при розв’язуванні певної системи дорівнює:



де n – розмірність квадратної матриці (або кількість невідомих квадратної матриці).

1.3 Алгоритм розв’язку задачі

Розглянемо систему m лінійних рівнянь з n невідомими. Для методу Жордана-Гауса її зручно зобразити у вигляді такої таблиці:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | … |  | … |  |  |
|  |  |  | … |  | … |  |  |
|  |  |  | … |  | … |  |  |
| … | … | … | … | … | … | … | … |
|  |  |  | … |  | … |  |  |
|  |  |  | … |  | … |  |  |

Нехай потрібно виразити змінну з і-го рфвняння системи, а потім потрібно підставити отриманий вираз в усі інші рівняння системи. Таке перетворення системи називають кроком Жорданових виключень з основним елементом .



Такі перетворення зручно виконувати користуючись таблицею (1), яка перейде в іншц таблицю за слідуючими правилами:

Усі вільні елементи реально заданої системи лінійних алгебраїчних рівнянь, тобто стовпець замінюють на протилежні;



Основний елемент замінюють на одиницю. Над основним стовпчиком записують , а біля рядка ;



Інші елементи основного стовпчика j залишають без змін;

Інші елементи основного рядка і-го змінюють лише свої знаки;

Елементи, які не належать розв’язуючому рядку або стовпчику обчислюють наступним чином. Створюється двовимірний визначник, який складається з таких елементів попередньої таблиці:

а) елемента з цими ж індексами, що й в обчислювальному елементі;

б) основний елемент;

в) елемент, який є спільним для стовпця з елементом (а) і стовпця з елементом (б);

г) елемент, який є спільним для стовпця з елементом (б) і рядка з елементом (а).

Шуканий елемент обчислюється як добуток елементів (а) та (б) мінус добуток залишившихся елементів визначника.

Цю дію можна зобразити у вигляді формули:



Всі елементинової таблиці ділять на елемент . Тим самим створюють ще одну таблицю, але вже без стовпця з елементом , а рядок з елементом позначають так, як позначили вилучений стовпець.



Виконавши всі описані операції новостворена таблиця матиме вигляд:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | … |  |  |
|  |  |  | … |  |  |
|  |  |  | … |  |  |
| … | … | … | … | … | … |
|  |  |  | … |  |  |
| … | … | … | … | … | … |
|  |  |  | … |  |  |

Шукаючи невідомі системи лінійних алгебраїчних рівнянь продовжують виконувати операції 2...6, причому основним елементом вже не можна вибрати елемент jI рядка, в якому вже був при попередніх Жорданових виключеннях використаний елемент. Операції 2...6 продовжують мати, поки усі позначення рядків не будуть замінені позначеннями стовпців, тобто поки всі стовпці крім -го не будуть вилучені.



Шуканими елементами будуть елементи, які залишаться після всіх обчислень в рядках навпроти нових позначень даних рядків. Оскільки нові позначення рядків відповідають відповідним невідомим та елементи навпроти, будуть відповідати розв’язкам заданої системи. З обчислення випливає, що шукані невідомі опиняються в стовпці під позначенням стовпця вільних елементів.

З даного методу обчислення помітно, що для розв’язку система повинна мати одинакову кількість рядків і невідомих, бо в протилежному випадку невідомі буде важко чи навіть неможливо знайти (коли невідових більше чим рядків) даним методом.

Розв’язуючи систему вручну методом Жордана-Гауса, основним елементом зручно вибирати число на яке найменьше ділити і, зрозуміло, що неможливо вибирати 0. для полегшення розв’язку при обчисленні кожен рядок зокрема можна скорочувати.

Оскільки комп’ютер не має логічного мислення, то йому важко задати вибирати зручніший елемент. Тому йому в програмі можна задати, щоб він вибрав перший можливий елемент для основного елемента.

2. Практична частина

2.1 Архітектура програми

Реалізацію обчислення систем лінійних рівнянь з однаковою кількістю стовпців і невідомих методом Жордана-Гауса здійснює програма Kursova.pas.

Запустити дану програму можна наступними способами:

з головного меню середовища Turbo Pascal шляхом вибору опції Run/Run (спочатку програму потрібно завантажити в оперативну пам’ять);

з середовища Turbo Pascal нажиманням комбінації клавіш Ctrl+F9;

з середовища операційної оболонки Norton Commander шляхом запуску прграми Kursova.exe (попередньо програма повинна бути відкомпільована з опцією Destination to memory).

Програма Kursova.pas є продурноорієнтованою. До неї входять 7 наступних процедур:

ramka;

windo;

wikno;

wind;

vvidn;

obchuslennya.

Зробимо опис названих процедур:

Процедура Ramka призначена для виводу рамки в кожному, описаному у наступних процедурах вікні процедура виводить рамку у вигляді двох ліній. Біля верхнього лівого краю на верхній грані процедура виводить на екран 3 символи “[⏹]”. Процедура виконується для кращого оформлення вікна. Виклик цієї процедури здійснюється із наступних процедур.

Процедура Windo призначена для загального оформлення використовуваних у програмі вікон. Вона викликається із наступних процедур разом із вхідними, для даної процедури даними. Вхідні дані задають розміри виводжуваного вікна, основний та даний кольори, а також назву вікна.

Процедура Wikno. Призначення – виводить на екранвікно меню даної програми, та вікно опису даних про автора програми та дату створення даної програми. У вікні “Меню” процедура виводить на екран запрошення для натискання однієї із чотирьох клавіш F1-F4 та ESC відповідно при натисканні яких програма виводить вікно вводу розмірності матриці, вікно вводу елементів, вікно виводу результатів та здійснює вихід із програми. Процедура Wikno використовується процедурою Wind та із головного блоку програми.

Процедура Wind. Використовується для виводу на екран робочих вікон та вікон опису програми. Параметри виводу робочих вікон задаються при кожному виклику цієї процедури, параметри вікон опису програми є стандартними і описані вони у процедурі Wikno, яка викликається процедурою Wind. Розміри робочих вікон виводяться залежно від кількості вхідних чи вихідних даних. Процедура виконується при вводі розмірності системи, вводу елементів, та виводу результатів.

Процедура Widn викликається при натисканні клавіші F1. Вона виводить вікно вводу розмірності квідратної системи лінійних рівнянь. В разі введення розмірності більше 20-ти програма виведе повідомлення “Розмірність повинна бути не більше 20-ти”. Це означає, що дана програма не розв’яже систему з кількістю невідомих більше 20.

Процедура Vvid призначена для вводу елементів системи та її вільних елементів. Процедура виводить на екран вікно з розмірами, які залежать від кількості вводжуваних даних. Коли потрібно розв’язати систему з кількістю невідомих більше 4, виводиться вікно на 20 ввідних елементів (стільки ж скільки містить всіх елементів 4-х мірна система рівнянь). При заповненні вікна здійснюється перехід до наступного ж які попереднє вікно і так доти поки не будуть введені всі елементи. Дана процедура викликається із головного блоку програми при натисканні клавіші F2.

Процедура Obchuslennya обчислює невідомі на основі остатньо введених даних. Обчислення виконуються за методом Жордана-Гауса. Невідомі виводяться у спеціальному вікні, розміри якого розраховуються так, щоб помістились всі невідомі. Всі результати виводяться послідовно в стовпчику зверху вниз з такою ж послідовністю як розміщені невідомі в кожному рядку розвязуючої системи в напрямку зліва на право. У випадку, коли обчислювана система рівнянь не має розв’язків програма виводить вікно “Повідомлення” з повідомленням “Система не має розв’язків”.

Головний блок програми, при входженні в програму, виводить на екран вікно “Меню” із запрошенням натиснути одну з 4-х клавіш: F1, F2, F3, Esc та вікно “Про автора”. Для загального опису даних про створення пргграми. В подальшій роботі крім цих вікон на екран виводяться вікна, які залежать від поточної роботи з програмою.

При натисканні однієї з 4-х даних клавіш активізується відповідна процедура. При натисканні клавіші ESC закінчується роюота програми і здійснюється повернення в середовище системи програмування Turbo Pascal.

2.2 Опис програми

На початку програми міститься вступний коментар до програми. Інші оператори мають наступне призначення:

Заголовок програми

Підключення зовнішнього модуля управління дисплеєм (Crt)

Опис константи (m), якій присвоюється число 21, для обмеження розмірності обчислювальних рівнянь

Надання змінній m\_1 типу двомірного масиву дійсних чисел з обмеженням m\*m

Надання змінній m\_2 типу одномірного масиву дійсних чисел з обмеженням m

Надання змінній m\_3типу одномірного масиву цілих чисел з обмеженням m

Опис, використаної у головному блоці програми мітки m1

{008} - {011} Опис використовуваних у програмі масивів та змінних

Заголовок процедури Ramka

Початок процедури

Перехід у позицію з координатами (х1, y1) і вивід у цих позиціях символа “╔”, як верхнього лівого кута рамки

Цикл виводу символів “═”, для обмеження рамки зверху

Вивід символа “╗”, як верхнього правого кута рамки

{017} - {020} Цикл виводу символа “║” для обмеження рамки з боків

Прехід у позицію з координатами (x1,y2) і вивід символа “╚” (нижній лівий кут рамки)

Цикл виводу символа “═”для обмеження рамки знизу

Вивід символа “╝” (правого нижнього кутка рамки)

Вивід у позицію (x1+3,y1) символів “[⏹]”

Закінчення процедури Ramka

Заголовок процедури Windo та опис вхідних даних

Початок процедури

{028} - {029} Встановлення основного та фонного кольору

Опис вікна з параметрами, які задаються при виклику процедури Windo

Очистка екрану

Виклик процедури Ramka із вхідними для неї розмірами:2,1,x2-x1,y2-y1+1

{033} - {034} Опис виводу на верхній грані рамки аосередині назви вікна

Закінчення процедури

Заголовок процедури Wikno

Початок процедури

Задання кольору фону та основного кольору

{039} - {042} Цикл виводу по всій величині екрану пропусків

{043} - {047} Виклик процедури Windo із заданнням назви вікна “Меню” та вихід у вікні головного меню

{048} – {053} Виклик процедури Windo із заданням інших координат виводячого вікна та задання назви вікна “Про автора” а також вивід у цьому вікні відомостей про автора

Кінець прцедури Vikno

Заголовок процедури Wind та опис вхідних даних

Початок процедури

Виклик процедури Vikno

Виклик процедури Windo

Закінчення процедури Wind

Заголовок процедури Vvidn

Початок процедури

Виклик процедури Wind із заданням назви виводячого вікна “Ввід розмірності квадратної матриці”.

{063} - {064} Ввід розмірності

{065} - {069} Перевірка умови чи часом не введена розмірність більша за 20, якщо так то виводиться на екран повідомлення “Розмірність повинна бути не більшою за 20”

Кінець процедури Vvidn

Заголовок процедури Vid

Опис використаної у процедурі Vvidn мітки (m8)

Початок процедури

Присвоєння змінній k введеної розмірності системи

{075} - {076} Перевірка умови чи k>4 якщо так то змінній k присвоюється 4, що буде використане для обмеження розмірності вікна.

{077} - {079} Присвоєння змінній позиціонування курсора та змінній організації початкових значень циклів початкового значення j

{080} Виклик процедури Window із вхідними параметрами та назвою виводячого вікна “Ввід елементів”

{081} - {082} Початок циклів нумерації ввідних елементів

{083} - {086} Перехід у позицію з координатами (3,t+1) та вивід на екран “Елемент а[i,j]=” або “вільний елемент [i,j]=” залежно від поточного вводу елемента та очікування вводу цього елемента

{087} Перевірка умови чи вікно вводу повністю заповнене, якщо так то початок складного оператора

{088} – {089} Присвоєння змінній позиціонування курсора та змінній початку циклу по рядках відповідно значень 1 і значення початкового номера рядка

{090} – {094} Перевірка чи остатньовведене число є елементом чи вільним членом. Якщо елементом, то змінній початку циклу по стовпцях присвоюється номер наступного стовпця. Якщо вільний елемент то змінній h стає одиницею, змінна початку циклу по рядкає дорівнює номеру наступного стовпця, якщо вільний елемент то змінна h стає одиницею.

{095} Очистка заповненого елементами вікна.

{096} Передача на мітку M8

{097} Закінчення складного оператора

{098}-{102} Перевірка чи в поточне вікно ще вмітиться наступний елемент: якщо так, то змінна переміщування курсора збільшується на одиницю, а змінній h і z присвоюється початкове значення один.

{103} Закінчення циклу вводу елементів

{104} Закінчення процедури Vvid

{105} Заголовок процедури Obchuslennia

{106} Опис використаних у процедурі міток

{107} Початок процедури

{108} Присвоєння змінній j номера стовпця вільних елементів

{109}-{110} Цикл заміни вільних елементів на протилежні

{111}-{112} Обнуління змінної t, яка буде використовуватись для обчислення в циклі кількості виконуваних стовпців та присвоєння змінній e розмірності розв’язуючої системи

{113}-{114} Цикл обнуління масиву E, який буде використовуватись для запам’ятовування кількості зсувів вліво кожного стовпця після кожного наступного викидування стопця.

{115} Початок циклу обчислення невідомих

{116} Обнуління змінної I

{117}-{118} Перевірка чи I дорівнює кількості рядків системи. Якщо так то перехід на мітку.

{119} Присвоєння нуля змінні j

{120}-{123} Перевірка чи з даного перевіряючогорядка, вже був використаний елемент для основного елемента чи ні. Якщо так, то перехід на мітку m4

{124} Надання змінній j нульового значення

{125} Збільшення змінної j на одиницю

{126} Перевірка умови: чи дійшли до останнього рядка, чи перевіряючий елемент стоїть перед вільним елементом даного рядка і чи даний елемент дорівнює нулю. Якщо все так то перехід на мітку m6

{127} Перевірка умов: чи перевіряючий елемент є передостатнім в рядку I чи наний елемент дорівнює нулю, якщо так то перехід на мітку m2

{128}-{132} Якщо перевіряючий елемент дорівнює нулю то перехід на мітку m5 і запам’ятовування індексів даного перевіряю чого елементу

{133} Присвоєння змінній k остатнього перевіряю чого елементу

{134}-{135} Початок циклів по стовпцю і рядку для обчислення елементів залишившиїся системи після попередніх обчислень.

{136}-{137} Перевірка умови: чи обчислювальний елемент не стоїть в рядку або стовпці до якого належить основний елемент. Якщо умова виконується то поточний елемент обчислюється за правилом обчислення двомірного визначника і результат запам’ятовується в інший масив

{138}-{139} Якщо обчислювальний елемент стоїть в рядку де є останьо обраний основний елемент але не є основним елементом то обчислювальний елемент запам’ятовується в масив В з протилежним знаком

{140} Закінчення циклу обчислення елементів

{141}-{142} Початок циклу по викиданню стовпця з основним елементом

{143}-{144} Якщо індекс стовпця менший за цей же індекс останнього елемента то в масив А буде записане число, яке одержується шляхом ділення елемента з масиву В з цими ж індексами на основний елемент

{145}-{148} якщо індекс стовпця більший або рівний цьому ж індексу основного елемента то в масив А запам’ятовується число отримане шляхом ділення наступного елемента поточного рядка з масиву В на основний елемент

{149} Закінчення циклу викидання стовпця

{150} Збільшення змінної t на одиницю

{151} Запам’ятовування в масив С з яких рядків уже були використані елементи для основного елемента

{152} Запам’ятовування в масив D навпроти кожного рядка остаточний результат якій змінній має відповідати

{153}-{158} Цикл запам’ятовування кількості зсувів вліво після кожного наступного викидування стовпця

{159} Зменшення змінної z на одиницю

{160} закінчення циклу обчислення невідомих

{161} присвоєння змінній z одиниці

{162}-{165} Цикл впорядкування результатів

{166} Присвоєння змінній t нуля

{167} Виклик процедури Wind і надання розмірів та заголовку вікна, яке виводиться цією процедурою

{168}-{171} Цикл виводу результатів

{172} Перехід на мітку m7

{173} Виклик процедури Wind і задання заголовку

{174} Перехід у позицію (10,3) і вивід повідомлення “Система немає розв’язку”

{175} Очікування натискання клавіші на клавіатурі

{176} Закінчення процедури Obchusennja

{177} Початок головного блоку програми

{178} Виклик процедури Vikno

{179} Очікування натискання однієї з ф-них клавіш: F1, F2, F3 або ESC

{180} Заголовок оператора вибору, який аналізує код натиснутої клавіші

{181} Якщо нажато клавішу F1 то здійснюється виклик процедури Widn

{182} Якщо нажато клавішу F2, викликається процедура Vvid

{183} Якщо нажато клавішу F3 виклик процедури Obchuslenja

{184} Кли нажато ESC – вихід з програми

{185} Закінчення оператору вибору

{186} Прехід на мітку m1

{187} Кінець програми

2.3 Контрольний приклад та результат машинного експерименту

Найбільш відповідальною, найбільш трудомісткою і з найбільшими витратами часу є випробовування (створення програм та усунення помилок. Не можна бути впевненим, що програма працює правильно коли вона виводить якісь результати. Найчастіше ці результати перші) бувають помилковими. Це означає, що багато помилок. Тому для перевірки правильності програми необхідно розробити тестові дані з наперед визначеними (обчисленими) результатами. Тестові дані повинні бути розроблені так, щоб перевірити всі оператори програми без виключення.

Для перевірки на правильність даної програми розроблені наступні тести:

Тест 1.

Система з трьома невідомими



Результати:



Порівнявши результати обчислені вручну до тестових даних і результати обчислені за допомогою програми Kursova.pas, фкі подані у додатку бачимо що все співпадає, тому можна стверджувати, що програма дає правильні результати і саме не містить помилок. Однак остаточним висновок про правильність програми можна зробити тільки після тривалої експлуатації.

Висновок

Розв’язування математичних задач за допомогою персональних комп’ютерів має велике значення, оскільки кожен розв’язок є оперативним та точним. Обчислюючи математичну задачу за допомогою сучасного комп’ютера, зрозуміло, що ймовірність помилкового введення даних в пам’ять ПК є набагато більшою чим ймовірність помилкового обчислення комп’ютером, чим тут практично можна знехтувати. Тоді що вже можна говорити про порівняня ручного обчислення і машинного. Тому надзвичайно важливо самостійно складати програми для виникаючих задач, які потрібно часто рішати. Звичайно головним в написанні програми є правильність її складання, оскільки правильність результатів обчислення найперш залежить від правильності програми.

В даній курсовій роботі складено програму по обчисленню різних систем рівнянь з квадратною матрицею. Дану програму розроблено в середовищі Turbo Pascal. Її налагоджено з використанням різноманітних тестових даних. Прикладом для перевірки роботоздатності програми було розроблено вручну тест. Результати обчислення вручну та за допомогою комп’ютера повністю співпали. Тому можна вважати цей програмний продукт готовим для практичного застосування.

Список використаної літератури

Turbo pascal(учебник) / С.А. Немнюгин – СПБ: Издательство “Питер”, 2000.

М.Я. Ляшенко, М.С. Головань. Чисельні методи. К.:”Либідь”, 1996 – 285 с.

Математика для техникумов – Алгебра и начала анализа.

Додаток 1

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)

(\* програма по розвязуванню \*)

(\* систем лiнiйних рiвнянь \*)

(\* методом Жордана-Гауса,написана до курсового проекту. \*)

(\* Програму склав Добромильський Iван \*)

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)

{001} program metod\_Zhordana\_Hausa;

{002} uses crt;

{003} const m!;

{004} type m\_1=array[1..m,1..m] of real;

{005} m\_2=array[1..m] of real;

{006} m\_3=array[1..m] of integer;

{007} label m1;

{008} var A,B:m\_1; X:m\_2; C,D,E:m\_3;

{009} R:real;

{010} i,j,k,t,l,z,h,n,p:integer;

{011} s:char;

(\* Процедура виводу рамки \*)

{012} procedure ramka(x1,y1,x2,y2:integer);

{013} begin

{014} gotoxy(x1,y1);write('Й');

{015} for i:=1 to x2-x1-1 do write('Н');

{016} write('»');

{017} for i:=1 to y2-y1-1 do begin

{018} gotoxy(x2,y1+i);write('є');

{019} gotoxy(x1,y1+i);write('є');

{020} end;

{021} gotoxy(x1,y2);write('И');

{022} for i:=1 to x2-x1-1 do write('Н');

{023} write('ј');

{024} gotoxy(x1+3,y1);write('[ю]');

{025} end;

(\* Процедура опису вiкна \*)

{026} procedure windo(x1,y1,x2,y2:integer;Nazva:string);

{027} begin

{028} TextColor(15);

{029} TextBackGround(1);

{030} Window(x1,y1,x2,y2);

{031} clrscr;

{032} ramka(2,1,x2-x1,y2-y1+1);

{033} l:=x2-x1+1-length(nazva);

{034} gotoxy((l div 2)+3,1);write(Nazva);

{035} end;

(\* Процедура виводу загального опису \*)

{036} procedure vikno;

{037} begin

{038} TextBackGround(15);TextColor(1);

{039} for i:=1 to 80 do

{040} for j:=1 to 25 do begin

{041} gotoxy(i,j);write(' ');

{042} end;

{043} windo(50,2,75,7,' Меню ');

{044} gotoxy(3,2);write('F1-Ввiд розмiрностi');

{045} gotoxy(3,3);write('F2-Ввiд даних');

{046} gotoxy(3,4);write('F3-Вивiд результатiв');

{047} gotoxy(3,5);write('ESC-Вихiд');

{048} windo(50,10,78,16,' Про автора ');

{049} gotoxy(3,2);write('Програма курсово» роботи.');

{050} gotoxy(3,3);write('Складена студентом групи');

{051} gotoxy(3,4);write('39-П Добромильським I.');

{052} gotoxy(3,5);write('Дата складення програми');

{053} gotoxy(3,6);write('травень 2003р');

{054} end;

(\* Процедура виводу вiкна \*)

{055} procedure wind(x1,y1,x2,y2:integer;Nazva:string);

{056} begin

{057} vikno;

{058} windo(x1,y1,x2,y2,nazva);

{059} end;

(\* Пороцедуга вводу розмiрностi квадратно» матрицi \*)

{060} procedure vvidn;

{061} begin

{062} wind(2,2,47,5,'=Ввiд розмiрностi квадратно» матрицi');

{063} gotoxy(3,2); write('n=');

{064} readln(n);

{065} if n>=m then begin

{066} wind(2,2,40,6,'Повiдомлeння');

{067} gotoxy(4,3);writeln('Розмiрнiсть повинна не бiльша ',m-1,'');

{068} S:= readkey;

{069} end;

{070} end;

(\* Процедура вводу елементiв \*)

{071} procedure vvid;

{072} label m8;

{073} begin

{074} k:=n;

{075} if k>4 then

{076} k:=4;

{077} t:=1;

{078} z:=1;

{079} h:=1;

{080} m8:wind(2,2,35,k\*k+k+3,'Ввiд елементiв');

{081} for i:=z to n do

{082} for j:=h to n+1 do begin

{082} gotoxy(3,t+1);

{084} if j<=n then write ('Елемент a[',i,j,']=')

{085} else write ('Вiльний елемент a[',i,j,']=');

{086} readln(A[i,j]);

{087} if t>19 then begin

{088} t:=1;

{089} z:=i;

{090} if j<(n+1) then h:=j+1

{091} else begin

{092} z:=i+1;

{193} h:=1;

{194} end;

{195} clrscr;

{196} goto m8;

{197} end;

{198} if t<\_ then begin

{199} t:=t+1;

{100} z:=1;

{101} h:=1;

{102} end;

{103} end;

{104} end;

(\* Процедура обчислення невiдомих \*)

{105} procedure obchuslennja;

{106} label m2,m3,m4,m5,m6,m7;

{107} begin

{108} j:=n+1;

{109} for i:=1 to n do

{110} A[i,j]:=-A[i,j];

{111} t:=0;

{112} l:=n;

{113} for i:=1 to l+1 do

{114} E[i]:=0;

{115} for p:=1 to n do begin

{116} i:=0;

{117} m2:if i=n then goto m6

{118} else i:=i+1;

{119} j:=0;

{120} m3:if j=t then goto m4

{121} else j:=j+1;

{122} if i=C[j] then goto m2

{123} else goto m3;

{124} m4:j:=0;

{125} m5:j:=j+1;

{126} if(i=n)and(j=l)and(A[i,j]=0) then goto m6;

{127} if (j=l)and(A[i,j]=0) then goto m2;

{128} if A[i,j]=0 then goto m5

{129} else begin

{130} k:=i;

{131} h:=j;

{132} end;

{133} R:=A[k,h];

{134} for i:=1 to n do

{135} for j:=1 to l+1 do begin

{136} if(i<>k)and(j<>h) then

{137} B[i,j]:=R\*A[i,j]-A[i,h]\*A[k,j];

{138} if (i=k)and(j<>h) then

{139} B[i,j]:=-A[i,j];

{140} End;

{141} for i:=1 to n do

{142} for j:=1 to l do begin

{143} if j<h then

{144} A[i,j]:=B[i,j]/R;

{145} if j>=h then begin

{146} z:=j+1;

{147} A[i,j]:=B[i,z]/R;

{148} end;

{149} end;

{150} t:=t+1;

{151} C[t]:=k;

{152} D[k]:=E[h]+h;

{153} for i:=1 to l+1 do begin

{154} if i>h then begin

{155} z:=i-1;

{156} E[z]:=E[i]+1;

{157} end;

{158} end;

{159} l:=l-1;

{160} END;

{161} z:=1;

{162} for i:=1 to n do begin

{163} j:=D[i];

{164} X[i]:=A[j,z];

{165} end;

{166} t:=0;

{167} wind(2,2,45,n+3,'Результати');

{168} for i:=1 to n do begin

{169} gotoxy(5,i+1);writeln('x[',i,']=',X[i]:8:3);

{170} t:=t+1;

{171} end;

{172} goto m7;

{173} m6:wind(2,2,45,6,'Повiдомлення');

{174} gotoxy(10,3); write('Система не ма№ розвязкiв');

{175} m7: s:=readkey;

{176} end;

(\* Головний блок програми \*)

{177} BEGIN

{178} m1:vikno;

{179} s:=readkey;

{180} case s of

{181} #59:vvidn;

{182} #60:vvid;

{183} #61: obchuslennja;

{184} #27:halt;

{185} end;

{186} goto m1;

{187} END.

Додаток 4

Екранна форма виводу результатів обчислення

Мею

Результати

|  |  |  |
| --- | --- | --- |
| x[1]= 4.000 x[2]= -1.000  x[3]= 2.000 |  | F1-Ввiд розмiрностi  F2-Ввiд даних  F3-Вивiд результатiв  ESC-Вихiд |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | Про автора |
|  |  | Програма курсової роботи.  Складена студентом групи  39-П Добромильським I.  Дата складення програми  травень 2003р |