Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра ЕОМ



**Лабораторна робота №1**

З дисципліни:”Паралельні та розподілені обчислення ”

На тему:” ВИКОРИСТАННЯ ФУНКЦІОНАЛЬНОЇ ДЕКОМПОЗИЦІЇ ДЛЯ РОЗВ’ЯЗКУ ОБЧИСЛЮВАЛЬНИХ ЗАДАЧ”

Виконав: ст.гр. КІ-34

Заяць А.М. Прийняв: Козак Н.Б.

Львів 2020

МЕТА РОБОТИ.

Вивчити методи декомпозицій задач. Набути навиків розв’язування задач з використанням функціональної декомпозиції.

**Варіант 7**

|  |  |  |  |
| --- | --- | --- | --- |
| 7 | число | | |
| bi=7i | A1(b1+c1) | A2(B2-C2)  Cij=1/(i3+j2) |

Правила знаходження елементів виразу.

1).Задати\* квадратну матрицю А порядку n. Отримати вектор(стовпець) , де b – вектор-стовпець, елементи якого обраховуються за формулою, згідно варіанту.

2).Задати квадратну матрицю А1 порядку n та вектори-стовпці b1 та c1 з n елементами кожен. Отримати вектор  згідно формули, що задається варіантом.

3).Задати квадратні матриці А2 та B2 порядку n. Отримати матрицю , яка залежить від А2, B2 та додатково визначеної матриці С2, елементи якої знаходяться за формулою, вказаною варіантом.

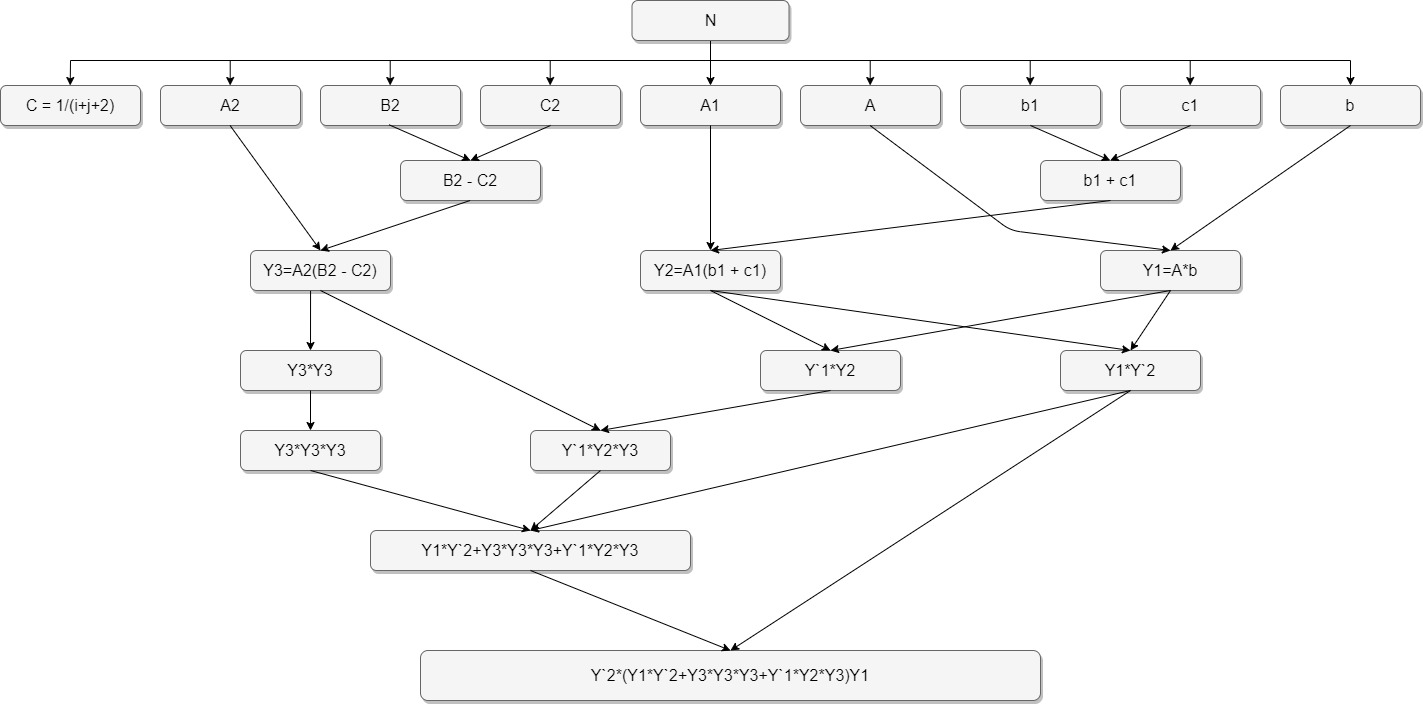


Рис.1. Схема декомпозиції поставленої задачі

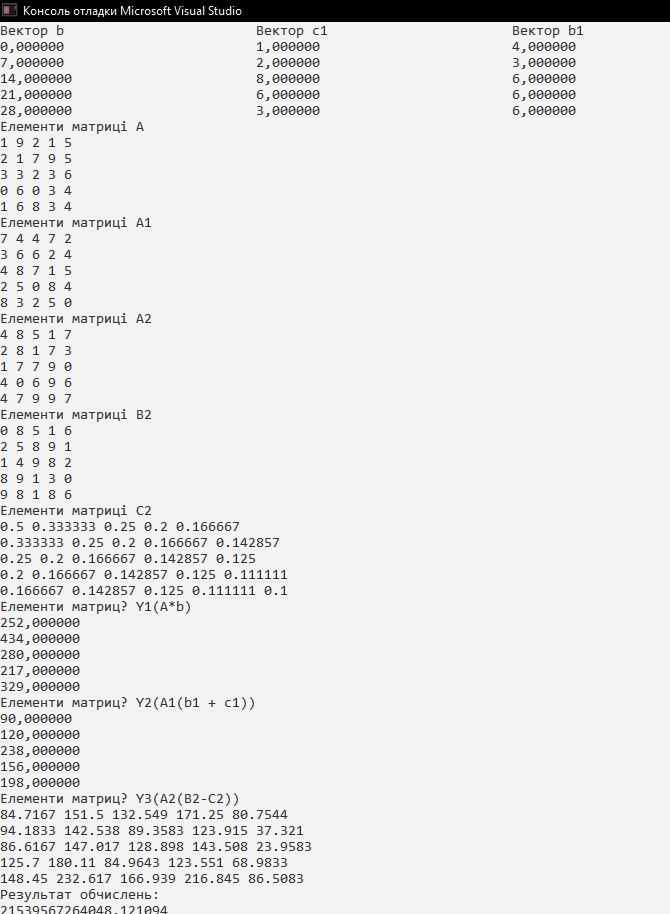


Рис.2. Результат обчислень виразу

Висновок: виконуючи дану лабораторну роботу я вивчив метод декомпозицій задач. Набув навиків розв’язування задач з використанням функціональної декомпозиції.

Лістинги програми:

#include "pch.h"

#include "pch.h"

#include <iostream>

#include <Windows.h>

using namespace std;

void PM(double \*\*mas, int size)

{

for (int i = 0; i < size; i++)

{

for (int j = 0; j < size; j++)

{

cout << mas[i][j] << " ";

}

cout << endl;

}

}

void PV(double \*mas, int size)

{

for (int i = 0; i < size; i++)

{

cout << mas[i] << " ";

cout << endl;

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

int N;

int REZ;

cout << "Кiлькiсть елементiв" << endl;

cin >> N;

/////////////////////////

double \*tmp\_v = new double[N];

double \*\*tmp\_m = new double\*[N];

double \*b1 = new double[N];

double \*c1 = new double[N];

double \*b = new double[N];

/////////////////////////

double \*\*A = new double \*[N];

double \*\*A1 = new double \*[N];

double \*\*A2 = new double \*[N];

double \*\*B2 = new double \*[N];

double \*\*C2 = new double \*[N];

double \*Y1 = new double [N];

double \*Y2 = new double [N];

double \*\*Y3 = new double \*[N];

/////////////////////////

for (int i = 0; i < N; i++)

{

A1[i] = new double[N];

A[i] = new double[N];

A2[i] = new double[N];

B2[i] = new double[N];

C2[i] = new double[N];

Y3[i] = new double[N];

tmp\_m[i] = new double[N];

b[i] = 7 \* i;

}

cout << "Ручне введеня елементiв матрицi - 1\nРандомне заповнення - 2" << endl;

cin >> REZ;

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

C2[i][j] = 1 / ((double)i + (double)j + 2);

}

}

if (REZ == 2)

{

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

A[i][j] = rand() % 10;

A1[i][j] = rand() % 10;

A2[i][j] = rand() % 10;

B2[i][j] = rand() % 10;

}

c1[i] = rand() % 10;

b1[i] = rand() % 10;

}

}

else

{

cout << "Матриця А: " << endl;

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

cin >> A[i][j];

}

}

cout << "Матриця А1: " << endl;

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

cin >> A1[i][j];

}

}

cout << "Матриця А2: " << endl;

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

cin >> A2[i][j];

}

}

cout << "Матриця B2: " << endl;

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

cin >> B2[i][j];

}

}

cout << "Матриця b1: " << endl;

for (int i = 0; i < N; i++)

{

cin >> b1[i];

}

cout << "Матриця c1: " << endl;

for (int i = 0; i < N; i++)

{

cin >> c1[i];

}

}

system("cls");

cout << "Вектор b\t\t\tВектор с1\t\t\tВектор b1" << endl;

for (int i = 0; i < N; i++)

{

printf("%2f\t\t\t%2f\t\t\t%2f\n", b[i],c1[i],b1[i]);

}

cout << "Елементи матрицi A" << endl;

PM(A, N);

cout << "Елементи матрицi A1" << endl;

PM(A1, N);

cout << "Елементи матрицi A2" << endl;

PM(A2, N);

cout << "Елементи матрицi B2" << endl;

PM(B2, N);

cout << "Елементи матрицi C2" << endl;

PM(C2, N);

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

tmp\_m[i][j] = (B2[i][j]) - C2[i][j];

}

tmp\_v[i] = b1[i] + c1[i];

}

for (int i = 0; i < N; i++)

{

Y2[i] = 0;

Y1[i] = 0;

for (int j = 0; j < N; j++)

{

Y3[i][j] = 0;

for (int k = 0; k < N; k++)

{

Y3[i][j] += A2[i][k] \* tmp\_m[k][j]; // A2(B2 - C2)

}

Y1[i] += A[i][j] \* b[j];

Y2[i] += A[i][j] \* tmp\_v[i];

}

}

cout << "Елементи матриці Y1(A\*b)" << endl;

for (int i = 0; i < N; i++)

{

printf("%f\n", Y1[i]);

}

cout << "Елементи матриці Y2(A1(b1 + c1))" << endl;

for (int i = 0; i < N; i++)

{

printf("%f\n", Y2[i]);

}

cout << "Елементи матриці Y3(A2(B2-C2))" << endl;

PM(Y3, N);

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

tmp\_m[i][j] = 0;

for (int k = 0; k < N; k++)

{

tmp\_m[i][j] += Y3[i][k] \* Y3[k][j]; // tmp\_m = Y3 \* Y3

}

}

}

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

B2[i][j] = 0;

A[i][j] = 0;

for (int k = 0; k < N; k++)

{

B2[i][j] += tmp\_m[i][k] \* Y3[k][j]; // B2 = Y3 \* Y3 \* Y3

A[i][j] += A1[i][k] \* Y3[k][j]; // A = Y1\*Y2\*Y3

}

}

}

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

tmp\_m[i][j] = 0;

for (int k = 0; k < N; k++)

{

tmp\_m[i][k] += B2[i][k] + A[i][k] + A1[i][k];

}

b1[i] = Y1[i] \* Y2[i]; // A1 = Y1 \* Y2

}

}

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

B2[i][j] = 0;

for (int k = 0; k < N; k++)

{

B2[i][k] += tmp\_m[i][k] \* b1[i];

}

}

}

printf("Результат обчислень: \n");

printf("%f \n", B2[0][0]);

}