Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа программной инженерии

О Т Ч Е Т

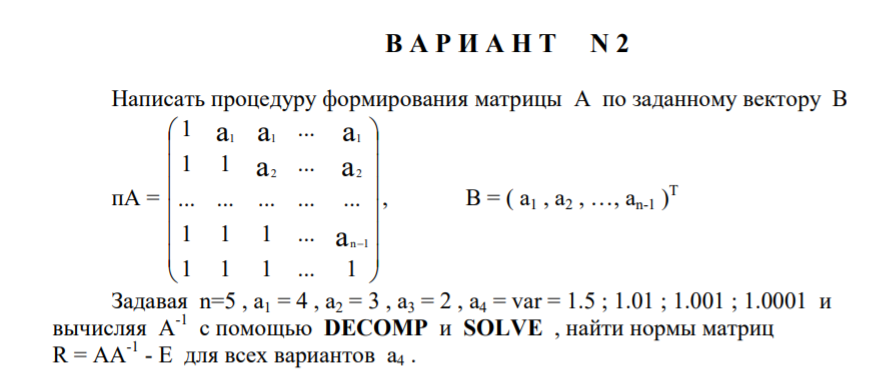
Лабораторная работа № 2

по дисциплине «Вычислительная математика»

Выполнила студент гр. 3530904/90002 Афанасьев Е. Д.

Преподаватель Воскобойников С. П.

**Постанова задачи**



**Код программы**

#include <iostream>

#include <vector>

#include <cmath>

#include <iomanip>

#include "cmath.h"

#include "decomp.c"

using Matrix = std::vector<double>;

void set\_matrix(Matrix& m, const Matrix v);

Matrix create\_inverse\_matrix(Matrix m);

Matrix create\_identity\_matrix();

Matrix create\_r\_matrix(Matrix m);

double get\_norm(Matrix m);

void show(Matrix m);

const Matrix b0 = {4, 3, 2, 1.5};

const Matrix b1 = {4, 3, 2, 1.01};

const Matrix b2 = {4, 3, 2, 1.001};

const Matrix b3 = {4, 3, 2, 1.0001};

std::vector<Matrix> b = {b0, b1, b2, b3};

const int size = 5;

double cond = 0;

int main()

{

Matrix m(size \* size);

for (int i = 0; i < 4; i++)

{

set\_matrix(m, b[i]);

Matrix r = create\_r\_matrix(m);

double norm = get\_norm(r);

std::cout << "Current fourth number is " << b[i][3] << '\n';

std::cout << "Cond = " << std::scientific << cond << '\n';

std::cout << "||R|| = " << std::scientific << norm << '\n';

std::cout << "Matrix R\n";

show(r);

std::cout << '\n';

}

}

void set\_matrix(Matrix& m, Matrix v)

{

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

{

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

{

m[i \* size + j] = (j <= i) ? 1 : v[i];

}

}

}

Matrix create\_inverse\_matrix(Matrix m)

{

Matrix matrix(size \* size);

int pivot[size];

int flag = 0;

decomp(size, size, m.data(), &cond, pivot, &flag);

Matrix e = create\_identity\_matrix();

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

{

Matrix b(size);

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

{

b[j] = e[j \* size + i];

}

solve(size, size, m.data(), b.data(), pivot);

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

{

matrix[k \* size + i] = b[k];

}

}

return matrix;

}

Matrix create\_identity\_matrix()

{

Matrix m(size \* size);

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

{

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

{

m[i \* size + j] = (i == j) ? 1 : 0;

}

}

return m;

}

Matrix create\_r\_matrix(Matrix m)

{

Matrix r(size \* size);

Matrix inv = create\_inverse\_matrix(m);

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

{

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

{

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

{

r[i \* size + j] += m[i \* size + k] \* inv[k \* size + j];

}

}

}

Matrix e = create\_identity\_matrix();

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

{

r[i] -= e[i];

}

return r;

}

double get\_norm(Matrix m)

{

double norm = 0;

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

{

double temp = 0;

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

{

temp += std::abs(m[i \* size + j]);

}

norm = (temp > norm) ? temp : norm;

}

return norm;

}

void show(Matrix m)

{

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

{

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

{

std::cout << m[i \* size + j] << ' ';

}

std::cout << '\n';

}

}

**Результат работы**

matrix is

1 4 4 4 4

1 1 3 3 3

1 1 1 2 2

1 1 1 1 1.5

1 1 1 1 1

cond is 5.175000e+01

matrix norm is 9.436896e-16

matrix R

0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00

5.551115e-17 0.000000e+00 0.000000e+00 0.000000e+00 8.881784e-16

5.551115e-17 0.000000e+00 0.000000e+00 0.000000e+00 8.881784e-16

5.551115e-17 0.000000e+00 0.000000e+00 0.000000e+00 8.881784e-16

5.551115e-17 0.000000e+00 0.000000e+00 0.000000e+00 8.881784e-16

matrix is

1.000000e+00 4.000000e+00 4.000000e+00 4.000000e+00 4.000000e+00

1.000000e+00 1.000000e+00 3.000000e+00 3.000000e+00 3.000000e+00

1.000000e+00 1.000000e+00 1.000000e+00 2.000000e+00 2.000000e+00

1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.010000e+00

1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00

cond is 2.207505e+03

matrix norm is 7.110978e-14

matrix R

0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00

5.551115e-17 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00

5.551115e-17 0.000000e+00 0.000000e+00 0.000000e+00 -2.842171e-14

5.551115e-17 0.000000e+00 0.000000e+00 2.842171e-14 -4.263256e-14

5.551115e-17 0.000000e+00 0.000000e+00 0.000000e+00 -1.421085e-14

matrix is

1.000000e+00 4.000000e+00 4.000000e+00 4.000000e+00 4.000000e+00

1.000000e+00 1.000000e+00 3.000000e+00 3.000000e+00 3.000000e+00

1.000000e+00 1.000000e+00 1.000000e+00 2.000000e+00 2.000000e+00

1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.001000e+00

1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00

cond is 2.200750e+04

matrix norm is 2.274292e-13

matrix R

0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00

5.551115e-17 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00

5.551115e-17 0.000000e+00 0.000000e+00 0.000000e+00 2.273737e-13

5.551115e-17 0.000000e+00 0.000000e+00 0.000000e+00 1.136868e-13

5.551115e-17 0.000000e+00 0.000000e+00 0.000000e+00 1.136868e-13

matrix is

1.000000e+00 4.000000e+00 4.000000e+00 4.000000e+00 4.000000e+00

1.000000e+00 1.000000e+00 3.000000e+00 3.000000e+00 3.000000e+00

1.000000e+00 1.000000e+00 1.000000e+00 2.000000e+00 2.000000e+00

1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000100e+00

1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00

cond is 2.200075e+05

matrix norm is 3.638034e-12

matrix R

0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00

5.551115e-17 0.000000e+00 0.000000e+00 0.000000e+00 3.637979e-12

5.551115e-17 0.000000e+00 0.000000e+00 0.000000e+00 3.637979e-12

5.551115e-17 0.000000e+00 0.000000e+00 1.818989e-12 0.000000e+00

5.551115e-17 0.000000e+00 0.000000e+00 0.000000e+00 1.818989e-12

**Вывод**

В данной работе были сформированы матрицы по заданным векторам и вычислены обратные матрицы с помощью decomp и solve. Найдены нормы матриц, причем, чем меньше 4 число в векторе, тем больше число обусловленности и тем больше норма полученной матрицы.