Міністерство освіти, науки, молоді та спорту України
Національний технічний університет України
«Київський політехнічний інститут»
Фізико-технічний інститут

Варіант 4

Лабораторна робота №3

Методи обчислень
«РОЗВ'ЯЗАННЯ СЛАР
ІТЕРАЦІЙНИМИ МЕТОДАМИ»

Підготував:
студент 3 курсу
групи ФІ-84
Коломієць Андрій Юрійович

Викладач:
Стьопочкіна Ірина Валеріївна

Київ – 2021

Вхідна система

Матриця A

2,12	0,42	1,34	0,88
0,42	3,95	1,87	0,43
1,34	1,87	2,98	0,46
0,88	0,43	0,46	4,44

Вектор B

11,172

0,115

0,009

9,349

Маємо систему лінійних алгебраїчних рівнянь виду:

$$Ax=B.$$

Умова діагональної переваги **не виконується:**

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{4} |a_{ij}|, i = 1,2,3,4$$

Здійснимо певні перетворення з матрицею, для того щоб вихідна матриця стала матрицею з діагональною перевагою.

Домножимо **другий рядок на -0.1 та додамо до першого рядка**.

Домножимо **другий рядок на -0.4 та додаємо до третього**.

Отримуємо еквівалентну **СЛАР**, але тепер матриця задовільняє умові діагональної переваги:

$$A = \begin{pmatrix} 2.078 & 0.025 & 1.153 & 0.837 \\ 0.42 & 3.95 & 1.87 & 0.43 \\ 1.172 & 0.29 & 2.232 & 0.288 \\ 0.88 & 0.43 & 0.46 & 4.44 \end{pmatrix} \qquad B = \begin{pmatrix} 11.1605 \\ 0.115 \\ -0.037 \\ 9.349 \end{pmatrix}$$

Необхідно здійснити перетворення системи $Ax=B$ в систему $x^{(k+1)} = C*x^{(k)} + S$:

$$C = \begin{pmatrix} 0 & -0.0120308 & -0.55486 & -0.402791 \\ -0.106329 & 0 & -0.473418 & -0.108861 \\ -0.52509 & -0.129928 & 0 & -0.129032 \\ -0.198198 & -0.0968468 & -0.103604 & 0 \end{pmatrix} \qquad S = \begin{pmatrix} 5.37079 \\ 0.0291139 \\ -0.0165771 \\ 2.10563 \end{pmatrix}$$

За початкове наближення оберемо вектор стовпчик, який рівномірно віддалений від додатніх і від'ємних значень:

$$X^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Виконуємо ітераційний процес доки в нас справедлива умова:

$$\frac{qn}{1-q} \geq \epsilon$$

де $\boldsymbol{\varepsilon}=\mathbf{1}0^{-4}$

На елементи матриці **C** накладається умова:

$$\sum_{j=1}^{n} |c_{ij}| \leq q < 1 \text{ або } \sum_{i=1}^{n} |c_{ij}| \leq q < 1,$$

число q визначаємо наступним чином:

$$q = \max_{1 \leq i \leq 4} \sum_{j=1}^{4} |\alpha_{ij}| = 0.969682$$

Критерій закінчення ітераційного процесу:

$$\frac{1}{1-q} \max_{j} |x_j^{k+1} - x_j^k| < \varepsilon$$

Наша програма робить обчислення за **47** ітераційних процесів.

Результати роботи програми

```
Start of work program!

Coefitient of iterating matrix:
        0       -0.0120308      -0.55486        -0.402791
        -0.106329       0       -0.473418       -0.108861
        -0.52509        -0.129928       0       -0.129032
        -0.198198       -0.0968468      -0.103604       0

Coefitient of iterating vector:
        5.37079
        0.0291139
        -0.0165771
        2.10563

Coefitient of sum element by module: 0.969682

Iteration 1:

        Step root: [ 5.37079; 0.0291139; -0.0165771; 2.10563; ]

        Difference: 5.37079

        Inconspicuous: [ 1.74403; 3.13015; 6.90943; 4.73119; ]

Iteration 2:

        Step root: [ 4.53151; -0.76333; -3.1122; 1.04005; ]

        Difference: 3.09562

        Inconspicuous: [ 4.48096; 6.59951; 1.52033; 2.50331; ]

Iteration 3:

        Step root: [ 6.68789; 0.907433; -2.43105; 1.60386; ]

        Difference: 2.15638

        Inconspicuous: [ 1.29905; 2.42187; 3.17417; 2.92937; ]

Iteration 4:

        Step root: [ 6.06274; 0.2943; -3.85317; 0.944087; ]

        Difference: 1.42212

        Inconspicuous: [ 2.20726; 3.20563; 1.10049; 1.46795; ]

Iteration 5:

        Step root: [ 7.12495; 1.10585; -3.36012; 1.27471; ]

        Difference: 1.0622

        Inconspicuous: [ 0.865504; 1.5103; 1.57547; 1.51051; ]
```

```
Iteration 42:
        Step root: [ 7.22006; 1.08331; -4.07652; 0.992051; ]
        Difference: 1.08078e-05
        Inconspicuous: [ 1.66035e-05; 2.60048e-05; 1.68339e-05; 1.80392e-05; ]

Iteration 43:
        Step root: [ 7.22007; 1.08331; -4.07651; 0.992055; ]
        Difference: 7.99013e-06
        Inconspicuous: [ 1.22612e-05; 1.92065e-05; 1.24438e-05; 1.33316e-05; ]

Iteration 44:
        Step root: [ 7.22006; 1.08331; -4.07652; 0.992052; ]
        Difference: 5.90048e-06
        Inconspicuous: [ 9.0629e-06; 1.41949e-05; 9.19021e-06; 9.84784e-06; ]

Iteration 45:
        Step root: [ 7.22007; 1.08331; -4.07652; 0.992055; ]
        Difference: 4.36136e-06
        Inconspicuous: [ 6.69375e-06; 1.04852e-05; 6.79244e-06; 7.2773e-06; ]

Iteration 46:
        Step root: [ 7.22006; 1.08331; -4.07652; 0.992053; ]
        Difference: 3.22124e-06
        Inconspicuous: [ 4.94705e-06; 7.74851e-06; 5.01714e-06; 5.376e-06; ]

Iteration 47:
        Step root: [ 7.22006; 1.08331; -4.07652; 0.992054; ]
        Difference: 2.38068e-06
        Inconspicuous: [ 3.65423e-06; 5.72396e-06; 3.70775e-06; 3.9725e-06; ]

End of work program!
```

Код програми

```
#include <iostream>
#include <string>
#include <cmath>
using namespace std;

#define type long double

#define line
cout<<endl<<"_____________________________________________________________________
_____________"<<endl;

void output_matrix(type**matrix, int size_matrix)
{

	//cout << endl << "View of matrix:" << endl;

	for (int i = 0; i < size_matrix; i++)
	{
		for (int j = 0; j < size_matrix; j++)
		{
			cout << "\t"<<matrix[i][j];
		}

		cout << endl;
	}
}

void output_vector(type* vector, int size_matrix)
{

	//cout << endl << "View of vector:" << endl;

	for (int i = 0; i < size_matrix; i++)
	{
		cout <<"\t"<< vector[i];
		cout << endl;
	}

}

type** count_coefitient_function_for_iterating_matrix(type**iterating_matrix,
type** matrix,int size_matrix)
{
	for (int i = 0; i < size_matrix; i++)
	{
		iterating_matrix[i][i] = 0;

		for (int j = i + 1; j < size_matrix; j++)
		{
			iterating_matrix[i][j] = -matrix[i][j] / matrix[i][i];
			iterating_matrix[j][i] = -matrix[j][i] / matrix[j][j];
		}
	}

	return iterating_matrix;
}

type* count_coefitient_function_for_iterating_vector(type*iterating_vector, type**
matrix, type* vector, int size_matrix)
{
	for (int i = 0; i < size_matrix; i++)
	{
		iterating_vector[i] = vector[i] / matrix[i][i];
	}
```

```
	return iterating_vector;
}

type count_coefitient_sum_by_module(type** iterating_matrix, type
sum_of_element,type any_coefitient,int size_matrix)
{
	for (int i = 0; i < size_matrix; i++)
	{
		for (int j = 0; j < size_matrix; j++)
		{
			sum_of_element = sum_of_element + abs(iterating_matrix[i][j]);
		}

		if (any_coefitient < sum_of_element)
		{
			any_coefitient = sum_of_element;
		}

		sum_of_element = 0;
	}

	return any_coefitient;
}

type* initialization_function_initial_approximation(type*initial_approximation,
int size_matrix)
{
	for (int i = 0; i < size_matrix; i++)
	{
		initial_approximation[i] = 0;
	}

	return initial_approximation;
}

type* step_root_function(type**iterating_matrix,type*
step_root,type*initial_approximation,type*iterating_vector,type sum_of_element,int
size_matrix)
{
	for (int i = 0; i < size_matrix; i++)
	{
		for (int j = 0; j < size_matrix; j++)
		{
			sum_of_element += iterating_matrix[i][j] *
initial_approximation[j];
		}

		step_root[i] = iterating_vector[i] + sum_of_element;

		sum_of_element = 0;
	}

	return step_root;
}

type condition_of_exit(type *step_root, type *initial_approximation, int
size_matrix)
{
	type max = abs(step_root[0] - initial_approximation[0]);

	for (int i = 1; i < size_matrix; i++)
	{
		if (max < abs(step_root[i] - initial_approximation[i]))
		{
			max = abs(step_root[i] - initial_approximation[i]);
		}
	}
```

```
	return max;
}

type* iterating_process(type**matrix, type*vector, type**iterating_matrix,
type*step_root, type*initial_approximation,
type*iterating_vector,type*inconspicuous,type any_coefitient,type temp, type
sum_of_element,type epsilon, int size_matrix)
{
	int iteration_number = 1;

	do {
		line
		cout << endl << "Iteration " << iteration_number << ":" << endl;
step_root_function(iterating_matrix, step_root, initial_approximation,
iterating_vector, sum_of_element, size_matrix);

		cout << endl << "\t Step root: [ ";

		for (int i = 0; i < size_matrix; i++)
		{
			cout << step_root[i] << "; ";
		}
		cout << "]" << endl;

		temp = condition_of_exit(step_root, initial_approximation,
size_matrix);

		cout << endl << "\t Difference: " << temp << endl;

		for (int i = 0; i < size_matrix; i++)
		{
			for (int j = 0; j < size_matrix; j++)
			{
				sum_of_element += matrix[i][j] * step_root[j];
			}
			inconspicuous[i] = abs(vector[i] - sum_of_element);
			sum_of_element = 0;
		}

		cout << endl << "\t Inconspicuous: = [ ";

		for (int i = 0; i < size_matrix; i++)
		{
			cout << inconspicuous[i] << "; ";
		}
		cout << "]" << endl;

		for (int i = 0; i < size_matrix; i++)
		{
			initial_approximation[i] = step_root[i];
		}

		iteration_number += 1;

	} while (any_coefitient*temp / (1 - any_coefitient) > epsilon);

	return step_root;
}

int main()
{
	line
	cout << endl << "Start of work program!" << endl;
	line

	//common data block -start
```

```
	int size_matrix = 4;

	type temp=0;
	type epsilon = 0.0001;
	type any_coefitient = 0;
	type sum_of_element = 0;

	//common data block -end


	// creating block-start
	type **matrix = new type*[size_matrix];

	for (int i = 0; i < size_matrix; i++)
	{
		matrix[i] = new type[size_matrix];
	}

	type **iterating_matrix = new type*[size_matrix];

	for (int i = 0; i < size_matrix; i++)
	{
		iterating_matrix[i] = new type[size_matrix];
	}

	type *vector = new type[size_matrix];
	type *iterating_vector = new type[size_matrix];
	type *initial_approximation = new type[size_matrix];
	type *step_root = new type[size_matrix];
	type *inconspicuous = new type[size_matrix];

	// creating block-end


	//initialization block -start

	// already counted matrix with diagonal priority

	matrix[0][0] = 2.078; matrix[0][1] = 0.025; matrix[0][2] = 1.153;
matrix[0][3] = 0.837;
	matrix[1][0] = 0.42;  matrix[1][1] = 3.95;  matrix[1][2] = 1.87;
matrix[1][3] = 0.43;
	matrix[2][0] = 1.172; matrix[2][1] = 0.29; matrix[2][2] = 2.232;
matrix[2][3] = 0.288;
	matrix[3][0] = 0.88;  matrix[3][1] = 0.43;  matrix[3][2] = 0.46;
matrix[3][3] = 4.44;


	// already counted vector with diagonal priority of matrix
	vector[0] = 11.1605;
	vector[1] = 0.115;
	vector[2] = -0.037;
	vector[3] = 9.349;

	//initialization block -end


	//count of coefitient of iterating matrix block-start


	count_coefitient_function_for_iterating_matrix(iterating_matrix,
matrix,size_matrix);

	cout << endl << "Coefitient of iterating matrix:" << endl;
```

```
	output_matrix(iterating_matrix, size_matrix);


	count_coefitient_function_for_iterating_vector(iterating_vector, matrix,
vector, size_matrix);

	cout << endl << "Coefitient of iterating vector:" << endl;

	output_vector(iterating_vector, size_matrix);


	//count of coefitient of iterating matrix and vector block-end


	//iterating process -start

	any_coefitient=count_coefitient_sum_by_module(iterating_matrix,sum_of_eleme
nt,any_coefitient,size_matrix);

	cout <<endl<< "Coefitient of sum element by module: " << any_coefitient <<
endl;

	initialization_function_initial_approximation(initial_approximation,
size_matrix);

	iterating_process(matrix,vector,
iterating_matrix,step_root,initial_approximation,iterating_vector,inconspicuous,an
y_coefitient,temp,sum_of_element,epsilon,size_matrix);

	//iterating process -end


	line
	cout << endl << "End of work program!" << endl;
	line


	//deliting block start

	delete[] inconspicuous;
	delete[] step_root;
	delete[] initial_approximation;
	delete[] iterating_vector;
	delete[] vector;

	for (int i = 0; i < size_matrix; i++)
	{
		delete iterating_matrix[i];
	}

	delete[] iterating_matrix;

	for (int i = 0; i < size_matrix; i++)
	{
		delete matrix[i];
	}

	delete[] matrix;

	//deliting block-end

	return 0;
}
```