**Лабораторная работа**

**«Метод Гаусса»**

**Арсений Анищенко**

**Группа 2**

**Задание 1.** Разработать программу численного решения СЛАУ методом Гаусса без выбора ведущего элемента.

Матрицу системы задать с диагональным преобладанием следующим образом:

* недиагональные элементы *ai,j*, *i≠j*, выбираются из чисел 0, –1, –2, –3, *–*4 произвольным образом;
* *ai,i=*, 2≤*i*≤*n*;
* *a*11*=*, *k*≥0.

Правую часть *b* задать умножением матрицы *A* на вектор *x=*(*m*, *m*+1, ... , *n*+*m*–1): *b=Ax*.

Для вычислений выбрать параметры:

* *m* – номер в списке студенческой группы;
* *n* – одно из чисел в пределах от 10 до 12;
* *k* – рассмотреть два случая: *k=*0, *k=*(номер студенческой группы); элементы *ai,j* при фиксированных *i* и *j* в обоих случаях одни и те же (матрицы отличаются только элементом *a*11).

**Решение**

Была реализована программа решения СЛАУ методом Гаусса без выбора ведущего элемента. Ниже приведена функция для решения СЛАУ методом Гаусса без выбора ведущего элемента. Входные параметры – матрица А и вектор b.

void gauss(float \*\*a, float \*b) {

cout << "Gaus method: \n";

for (int i = 0; i < n - 1; ++i) {

for (int j = i + 1; j < n; ++j) {

float coeff = a[j][i] / a[i][i];

for (int z = i; z < n; ++z) {

a[j][z] -= coeff \* a[i][z];

}

b[j] -= coeff \* b[i];

}

}

cout << "After the 1st step: " << endl;

for (int i = 0; i < n; ++i) {

cout << fixed << setprecision(4);

for (int j = 0; j < n; ++j) cout << setw(10) << a[i][j];

cout << setw(10) << b[i] << endl;

}

cout << endl;

for (int i = n - 1; i >= 1; --i) {

for (int j = i - 1; j >= 0; --j) {

float coeff = a[j][i] / a[i][i];

a[j][i] = 0;

b[j] -= coeff \* b[i];

}

}

float q = 0, w = 0;

for (int i = 0; i < n; ++i) {

\_x[i] = b[i] / a[i][i];

q += pow(\_x[i] - x[i], 2);

w += x[i] \* x[i];

cout << fixed << setprecision(10);

cout << i << "x = " << setw(14) << x[i] << " " << i << "\_x = " << \_x[i] << endl;

}

q = sqrt(q);

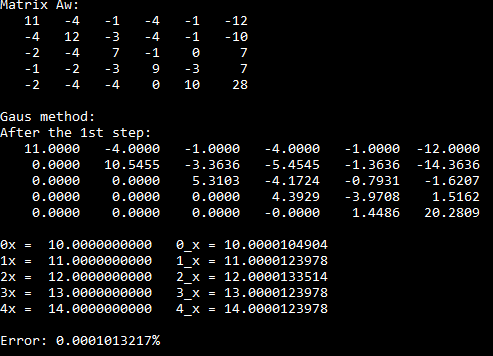
w = sqrt(w);

error = q / w \* 100;

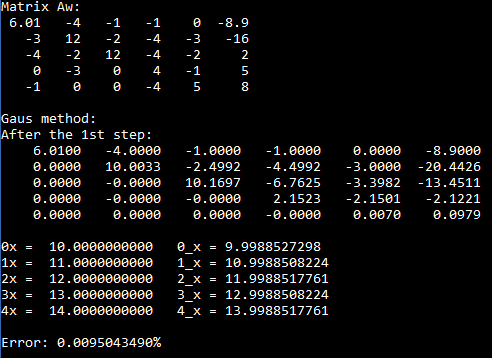
cout << endl << "Error: " << fixed << setprecision(10) << error << "%" << endl;

}

Пример выполнения программы при k = 0:



Пример выполнения программы при k = 2 (2 группа):



**Вывод**

Была создана программа для решения СЛАУ методом Гаусса без выбора ведущего элемента. При увеличении k была заметна потеря точности.

**Задание 2.** Разработать программу численного решения СЛАУ методом Гаусса с выбором ведущего элемента по столбцу.

Для заполнения матрицы *A* использовать случайные числа из диапазона от −100 до 100. Правую часть *b* задать умножением матрицы *A* на вектор *x=*(*m*, *m*+1, ... , *n*+*m*–1): *b=Ax*.

Для вычислений выбрать параметры:

* *m* – номер в списке студенческой группы;
* *n* – одно из чисел 7 или 8.

Программно реализовать вычисления для рассматриваемого примера методом Гаусса с выбором ведущего элемента и методом Гаусса без выбора ведущего элемента (система уравнений в обоих случаях одна и та же). Для вычислений использовать тип float.

**Решение**

Была реализована программа для решения СЛАУ методом Гаусса с выбором ведущего элемента. Так же, как и в случае с методом Гаусса без выбора, программа представляет собой функцию, принимающую в качестве параметров матрицу A и вектор b.

void gauss\_with\_choice(float \*\*a, float \* b) {

cout << "\nGaus method with choice: \n";

for (int i = 0; i < n - 1; ++i) {

float mx = a[i][i];

int pos = i;

for (int j = i + 1; j < n; ++j) {

if (fabs(a[j][i]) > fabs(mx)) {

mx = a[j][i];

pos = j;

}

}

for (int z = 0; z < n; ++z) {

swap(a[i][z], a[pos][z]);

}

swap(b[i], b[pos]);

for (int j = i + 1; j < n; ++j) {

float coeff = a[j][i] / a[i][i];

for (int z = i; z < n; ++z) {

a[j][z] -= coeff \* a[i][z];

}

b[j] -= coeff \* b[i];

}

}

cout << "After the 1st step: " << endl;

for (int i = 0; i < n; ++i) {

cout << fixed << setprecision(4);

for (int j = 0; j < n; j++) cout << setw(10) << a[i][j];

cout << setw(10) << b[i] << endl;

}

cout << endl;

for (int i = n - 1; i >= 1; --i) {

for (int j = i - 1; j >= 0; --j) {

float coeff = a[j][i] / a[i][i];

a[j][i] = 0;

b[j] -= coeff \* b[i];

}

}

float q = 0, w = 0;

for (int i = 0; i < n; ++i) {

\_x[i] = b[i] / a[i][i];

q += pow(\_x[i] - x[i], 2);

w += x[i] \* x[i];

cout << fixed << setprecision(10);

cout << i << "x = " << setw(14) << x[i] << " " << i << "\_x = " << \_x[i] << endl;

}

q = sqrt(q);

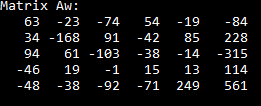
w = sqrt(w);

error = q / w \* 100;

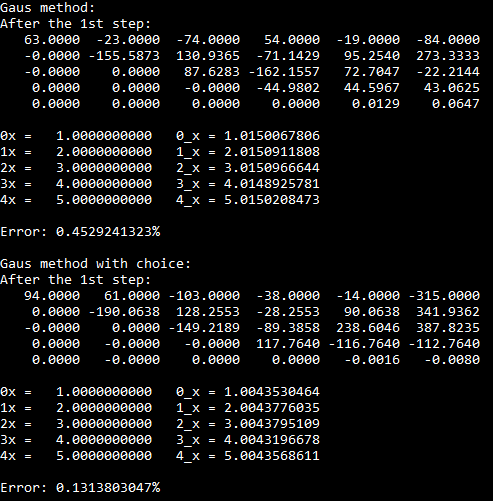
cout << endl << "Error: " << fixed << setprecision(10) << error << "%" << endl;

}

Рассмотренная матрица:



Решение:



int main() {

srand(time(0));

a = new float\*[SIZE];

b = new float[SIZE];

for (int i = 0; i < n; ++i) {

a[i] = new float[SIZE];

s[i] = 0;

for (int j = 0; j < n; ++j) {

if (i != j) {

a[i][j] = rand() % 200 - 100;

s[i] += a[i][j];

}

}

a[i][i] = -s[i];

}

a[0][0] += pow(10, -k);

for (int i = 0; i < n; ++i) {

b[i] = 0;

x[i] = m + i;

}

for (int i = 0; i < n; ++i){

for (int j = 0; j < n; ++j){

b[i] += a[i][j] \* x[j];

}

}

float\*\* a1 = new float\*[SIZE];

float\* b1 = new float[SIZE];

float\*\* a2 = new float\*[SIZE];

float\* b2 = new float[SIZE];

cout << "Matrix Aw: " << endl;

for (int i = 0; i < n; ++i) {

a1[i] = new float[SIZE];

a2[i] = new float[SIZE];

for (int j = 0; j < n; ++j) {

cout << setw(5) << a[i][j];

a1[i][j] = a2[i][j] = a[i][j];

}

cout << setw(6) << b[i] << endl;

b1[i] = b2[i] = b[i];

}

cout << endl;

gauss(a1, b1);

gauss\_with\_choice(a2, b2);

return 0;

}

**Вывод**

Была составлены программа численного решения СЛАУ методом Гаусса с и без выбора ведущего элемента с использованием для вычислений тип float. При этом была получена относительная погрешность 0.1313803047% для первого случая и для второго 0.4529241323%. Таким образом мы получили что, для конкретно данного теста метод Гаусса с выбором ведущего элемента был значительно точнее, чем без выбора.