**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ЛАБОРАТОРНАЯ РАБОТА 1**

Тема: «Прямые методы решения СЛАУ»»

Выполнил студент 1 гр.

Рудь Андрей Владимирович

**Минск 2019**

**Задание №1**

**Постановка задачи:**

Написать программу, которая решает систему линейных алгебраических уравнений *Ax = f* с матрицей *A* порядка *n* методом Гаусса с выбором главного элемента по столбцу, а также вычисляет определитель матрицы *det A* , обратную матрицу A-1. Предусмотреть сообщения, предупреждающие о невозможности решения указанной задачи с заданной матрицей *A*.

Для проведения вычислительного эксперимента необходимо решить систему размерности *n* =10 . Матрицу *A* и вектор точного решения *x* заполнить случайными числами (сгенерировать) с двумя знаками после запятой из диапазона от -10 до 10. Правую часть задать умножением матрицы *A* на вектор *x* : *f = Ax* .

В результатах выполнения тестовой задачи необходимо привести следующую информацию:

Условие: матрица A (построчно), вектор f, точное решение x .

Полученное решение

Максимум-норма невязки ||*A* – *f ||*∞

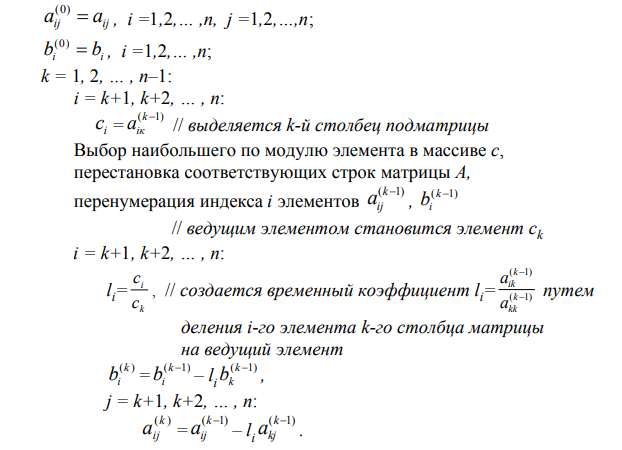
Максимум-норма погрешности || *x* - ||∞

Определитель матрицы *det A*

Обратную матрицу A-1 (построчно) и матрицу A-1A (построчно).

**Краткие теоретические сведения:**

Метод Гаусса с выбором главного элемента по столбцу:  
В качестве главного выбирается наибольший по модулю элемент. Данный метод эквивалентен применению обычного метода к системе, в которой на каждом шаге исключения проводится перенумерация уравнений. Алгоритм:



Если к системе с матрицей A применить метод Гаусса с выбором главного элемента (любой из вариантов выбора), то

  
где m – количество перестановок, осуществляемых в процессе исключения (другие преобразования, применяемые в процессе исключения, не изменяют определитель матрицы).

Задача нахождения матрицы, обратной матрице A, эквивалентна задаче решения матричного уравнения AX=E, где X – искомая матрица (если AX=E, то X=A-1E = A-1 ). Матричное уравнение AX=E с матрицей порядка n можно рассматривать как P систем с различными правыми частями и одной и той же матрицей, причем P=n, а правые части есть столбцы единичной матрицы.

Нормы векторов:

Кубическая норма (используется также название максимум-норма):

|| x ||I = max |xi|, 1 ≤ i ≤ n

Для этой нормы используется также обозначение || *x* ||∞

**Листинг программы:**

Matrix gauss(Matrix& A, Matrix& f, int& swapNumber)

{

int size = A.size\_x;

int size\_sol = f.size\_y;

//матрица решений:

Matrix solutions(size,size\_sol);

solutions.nullify();

//проход по диагональным:

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

//поиск главного элемента:

float maxElement = abs(A.matrix[i][i]);

int maxIndex = i;

int k = 0;

for (k = i + 1; k < size; ++k)

{

if (abs(A.matrix[k][i]) > maxElement)

{

maxIndex = k;

maxElement = abs(A.matrix[k][i]);

}

}

// обмен строк:

if (i != maxIndex)

{

A.matrix[i].swap(A.matrix[maxIndex]);

f.matrix[i].swap(f.matrix[maxIndex]);

++swapNumber;

}

//пересчет свободных членов, элементов матрицы

for (int k = i + 1; k < size; ++k)

{

float l = (float)(A.matrix[k][i] / A.matrix[i][i]);

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

{

A.matrix[k][j] -= (l \* A.matrix[i][j]);

}

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

{

f.matrix[k][j] -= (l \* f.matrix[i][j]);

}

}

}

//если матрица вырожденная:

bool b = false;

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

{

if (A.matrix[i][i] == 0)

b == true;

}

if (b)

{

throw Bad("Вырожденная матрица");

}

//подсчет решений:

for (int j = 0; j < size\_sol; ++j) {

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

float y = 0;

for (int k = i + 1; k < size; ++k) {

y += A.matrix[i][k] \* solutions.matrix[k][j];

}

solutions.matrix[i][j] = (f.matrix[i][j] - y) / A.matrix[i][i];

}

}

return solutions;

}

float maximum\_norm(Matrix const v)

{

if (v.size\_y != 1)

{

throw Bad("not a vector!");

return 0.;

}

else

{

float max = v.matrix[0][0];

for (int i = 1; i < v.size\_x; ++i)

{

if (abs(v.matrix[i][0]) > abs(max))

max = v.matrix[i][0];

}

return abs(max);

}

}

//для верхне-треугольной:

float DetA(Matrix const& m, int numberOfSwaps) {

float det = pow(-1, numberOfSwaps);

for (int i = 0; i < m.size\_x; ++i) {

det \*= m.matrix[i][i];

}

return det;

}

**Результаты:**

Сгенерированная матрица А:



|  |  |
| --- | --- |
| Сгенерированный вектор x: | Полученная правая часть f = Ax: |
|  |  |

Полученное решение

|  |  |
| --- | --- |
|  |  |

Максимум-норма невязки ||*A* – *f ||*∞



Максимум-норма погрешности || *x* - ||∞

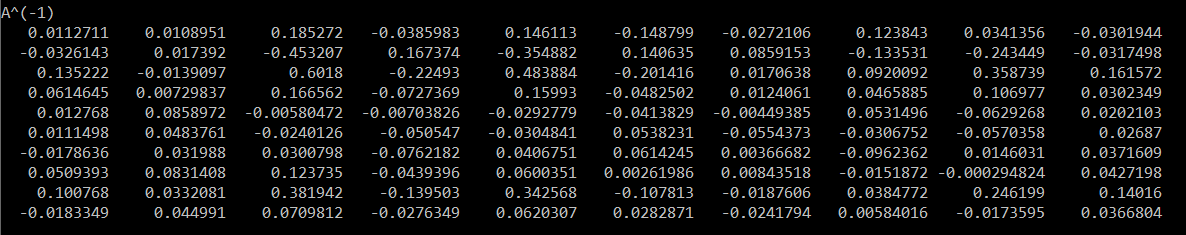


Определитель матрицы *det A*

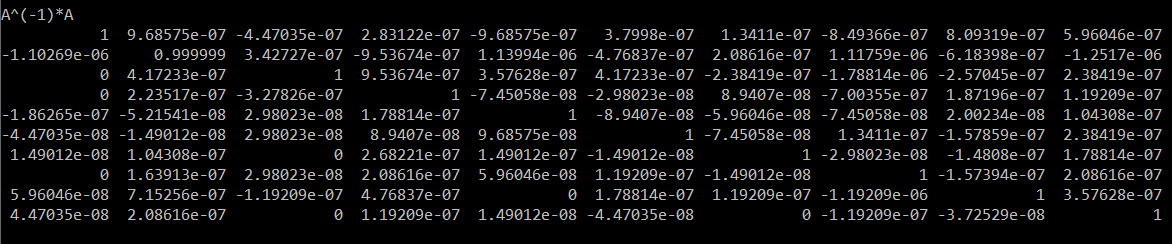


Обратную матрицу A-1 (построчно) и матрицу A-1A (построчно).

Обратная матрица A-1:



Матрица A-1A:



**Вывод:**

Cхемы выбора главного элемента не гарантируют устойчивость (т.е. низкую чувствительностью к погрешностям вычислений) метода Гаусса, но могут в значительной степени уменьшить влияние вычислительной погрешности на отклонение полученного результата от настоящего решения задачи.

**Задание №2**

**Постановка задачи:**

Написать программу, которая решает систему линейных алгебраических уравнений *Ax = f* для трехдиагональной матрицы A порядка n методом прогонки. Предусмотреть сообщения, предупреждающие о невозможности решения указанной задачи с заданной матрицей A.

Для проведения вычислительного эксперимента необходимо сгенерировать случайную трехдиагональную матрицу *A* с диагональным преобладанием размерности *n* =10 и вектор точного решения *x* . Правую часть задать умножением матрицы *A* на вектор *x* : *f = Ax.* Затем необходимо решить полученную систему с помощью вашей программы и занести в отчет результаты.

В результатах выполнения тестовой задачи необходимо привести следующую информацию:

Условие: матрица A (построчно), вектор f, точное решение x.

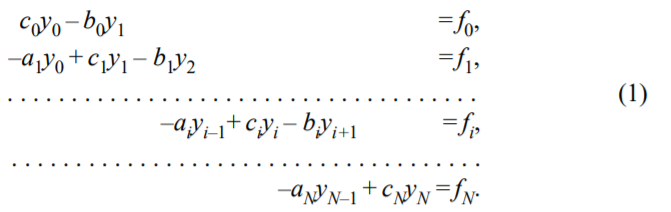
Полученное решение

Максимум-норма невязки ||*A* – *f ||*∞

Максимум-норма погрешности || *x* - ||∞

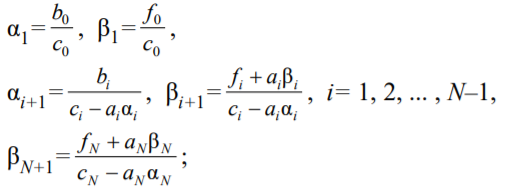
**Краткие теоретические сведения**

Алгоритм метода правой прогонки для решения системы (1)



имеет следующий вид:

прямая прогонка – вычисление прогоночных коэффициентов по формулам



обратная прогонка – вычисление решения по формулам



**Листинг программы:**

Matrix thomas\_algorithm(vector<float>& a, vector<float>& b, vector<float>& c, vector<float>& f, int n)

{

vector<float>beta;

for (int i = 0; i < n + 2; ++i)

beta.push\_back(0.);

vector<float>alpha;

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

alpha.push\_back(0.);

alpha[1] = b[0] / c[0];

beta[1] = f[0] / c[0];

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

{

alpha[i + 1] = b[i] / (c[i] - a[i] \* alpha[i]);

beta[i + 1] = (f[i] + a[i] \* beta[i]) / (c[i] - a[i] \* alpha[i]);

}

beta[n + 1] = (f[n] + a[n] \* beta[n]) / (c[n] - a[n] \* alpha[n]);

vector<float> y(n + 1);

y[n] = beta[n + 1];

for (int i = n - 1; i >= 0; --i)

{

y[i] = alpha[i + 1] \* y[i + 1] + beta[i + 1];

}

Matrix m(n + 1, 1);

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

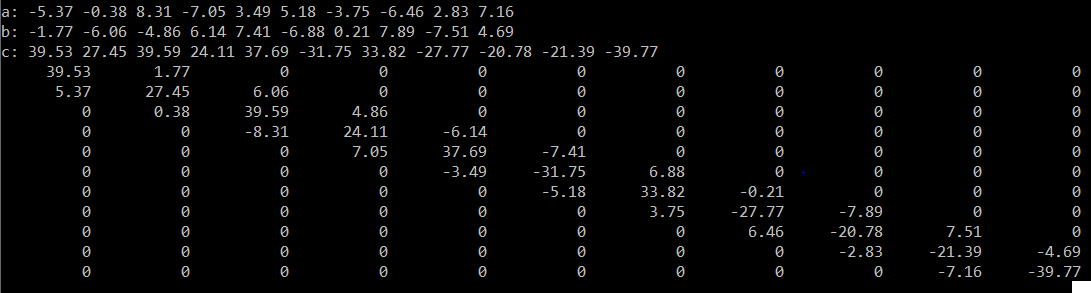
m.matrix[i][0] = y[i];

return m;

}

**Результаты:**

Сгенерированная матрица А:



|  |  |
| --- | --- |
| Сгенерированный вектор x: | Полученная правая часть f = Ax: |
|  |  |

Полученное решение

|  |  |
| --- | --- |
|  |  |

Максимум-норма невязки ||*A* – *f ||*∞



Максимум-норма погрешности || *x* - ||∞



* **Вывод**

Метод Гаусса является универсальным прямым методом решения систем линейных алгебраических уравнений. В то же время, учет специфики задачи позволяет построить алгоритмы с меньшими, по сравнению с универсальными, вычислительными затратами. Подсчет количества арифметических операций показывает, что для реализации метода прогонки требуется примерно *2N* делений, примерно *3N* умножений, примерно *3N* сложений и вычитаний. Таким образом, при классификации по вычислительной сложности, прогонка относится к алгоритмам с линейной сложностью.