ГОУ ВПО «Донецкий национальный университет»

Физико-технический факультет

Кафедра компьютерных технологий

**ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА**

**ЛАБОРАТОРНАЯ РАБОТА №2**

**РЕШЕНИЕ СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ МЕТОДОМ ГАУССА С ВЫБОРОМ ГЛАВНОГО ЭЛЕМЕНТА**

Выполнил:

Ермоленко Евгений

Студент 2 уск.курса

группы ИВТ-4

Проверил:

Пшеничный К.А.

**Ход работы**

# Задание 1

Написать программу метода Гаусса для решения системы *n* линейных уравнений *Ax*=*f*. Продемонстрировать её работоспособность на примере решения системы линейных уравнений с матрицей Гильберта *A*, элементы которой *aij* =1/(*i*+*j*-1), *i*,*j*=1.. *n*. Свободный член системы уравнений задайте в виде *fi* = *n*/*i2*. Проведите расчёты без выбора и с выбором главного элемента матрицы. Необходимо найти число обусловленности рассматриваемой матрицы – Cond(*A*)=||*A*||·||*A*-1||.

Для решения всех заданий был использован язык программирования C#.

**Функция преобразования исходной матрицы в обратную:**

static void Inversion(double[,] A, int N)

{

double temp;

double[,] E = new double[N, N];

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

{

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

{

E[i, j] = 0.0;

if (i == j)

{

E[i, j] = 1.0;

}

}

}

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

{

temp = A[k, k];

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

{

A[k, j] /= temp;

E[k, j] /= temp;

}

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

{

temp = A[i, k];

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

{

A[i, j] -= A[k, j] \* temp;

E[i, j] -= E[k, j] \* temp;

}

}

}

for (int k = N - 1; k > 0; k--)

{

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

{

temp = A[i, k];

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

{

A[i, j] -= A[k, j] \* temp;

E[i, j] -= E[k, j] \* temp;

}

}

}

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

{

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

{

A[i, j] = E[i, j];

}

}

}

**Функция вывода матрицы:**

static void showMatrix(double[,] A, int N)

{

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

{

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

{

if (A[i, j] == -0)

{

A[i, j] = 0;

}

Console.Write(A[i, j].ToString() + "\t");

}

Console.WriteLine();

}

}

**Функция решения системы методом Гаусса без выбора главного элемента:**

static double[] GaussWithoutMainElement(double[,] a, double[] y, int n)

{

double[] x = new double[n];

double max;

int k, index;

double eps = 0.00001; // точность

k = 0;

while (k < n)

{

// Поиск строки с максимальным a[i][k]

max = Math.Abs(a[0, 0]);

index = 1;

// Перестановка строк

if (max < eps)

{

// нет ненулевых диагональных элементов

Console.WriteLine("Решение получить невозможно из-за нулевого столбца " + index + " матрицы A");

return null;

}

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

{

double tempValue = a[k, j];

a[k, j] = a[index, j];

a[index, j] = tempValue;

}

double temp = y[k];

y[k] = y[index];

y[index] = temp;

// Нормализация уравнений

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

{

double tempValue = a[i, k];

if (Math.Abs(tempValue) < eps)

{

continue;

} // для нулевого коэффициента пропустить

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

{

a[i, j] = a[i, j] / tempValue;

}

y[i] = y[i] / tempValue;

if (i == k)

{

continue;

} // уравнение не вычитать само из себя

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

{

a[i, j] = a[i, j] - a[k, j];

}

y[i] = y[i] - y[k];

}

k++;

}

// обратная подстановка

for (k = n - 1; k >= 0; k--)

{

x[k] = y[k];

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

{

y[i] = y[i] - a[i, k] \* x[k];

}

}

return x;

}

**Функция решения системы методом Гаусса c выбором главного элемента:**

static double[] GaussWithMainElement(double[,] a, double[] y, int n)

{

double[] x = new double[n];

double max;

int k, index;

const double eps = 0.00001; // точность

k = 0;

while (k < n)

{

// Поиск строки с максимальным a[i][k]

max = Math.Abs(a[k, k]);

index = k;

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

{

if (Math.Abs(a[i, k]) > max)

{

max = Math.Abs(a[i, k]);

index = i;

}

}

// Перестановка строк

if (max < eps)

{

// нет ненулевых диагональных элементов

Console.WriteLine("Решение получить невозможно из-за нулевого столбца " + index + " матрицы A");

return null;

}

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

{

double tempValue = a[k, j];

a[k, j] = a[index, j];

a[index, j] = tempValue;

}

double temp = y[k];

y[k] = y[index];

y[index] = temp;

// Нормализация уравнений

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

{

double tempValue = a[i, k];

if (Math.Abs(tempValue) < eps)

{

continue;

} // для нулевого коэффициента пропустить

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

{

a[i, j] = a[i, j] / tempValue;

}

y[i] = y[i] / tempValue;

if (i == k)

{

continue;

} // уравнение не вычитать само из себя

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

{

a[i, j] = a[i, j] - a[k, j];

}

y[i] = y[i] - y[k];

}

k++;

}

// обратная подстановка

for (k = n - 1; k >= 0; k--)

{

x[k] = y[k];

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

{

y[i] = y[i] - a[i, k] \* x[k];

}

}

return x;

}

**Код программы:**

unsafe static void Main(string[] args)

{

double[,] a;

double[,] inv\_a;

double[,] tmp;

double[,] inv\_tmp;

double[] y;

double[] x;

double[] obr;

int n;

double temp;

Console.Write("Введите количество уравнений: ");

n = Convert.ToInt32(Console.ReadLine());

a = new double[n, n];

inv\_a = new double[n, n];

tmp = new double[n, n];

inv\_tmp = new double[n, n];

y = new double[n];

// Матрица Гильберта

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

{

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

{

temp = 1.0 / (i + 1 + j + 1 - 1); // (i+1),(i+1) т.к. матрицы начинаются с 1, а не с 0

Console.Write(Math.Round(temp,5).ToString() + "\t\t");

a[i, j] = temp;

tmp[i, j] = 0;

}

Console.WriteLine();

}

Console.WriteLine();

Console.WriteLine("Свободные члены системы:");

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

{

temp = n / Math.Pow(i + 1, 2); // Свободный член системы fi = n/i^2

Console.Write("f[" + i + "]= " + temp);

y[i] = temp;

Console.WriteLine();

}

matrixCopy(a, tmp, n);

Console.WriteLine();

ShowSystem(a, y, n);

Console.WriteLine();

x = GaussWithMainElement(a, y, n);

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

{

Console.WriteLine("x[" + i + "]=" + x[i].ToString());

}

Console.WriteLine();

matrixCopy(a, inv\_a, n);

obr = GaussWithoutMainElement(tmp, y, n);

Console.WriteLine("Решение СЛАУ методом Гаусса (без выбора главного эл.)");

showMatrix(tmp, n);

Console.WriteLine("");

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

{

Console.WriteLine("x[" + i + "]=" + obr[i].ToString());

}

Console.WriteLine();

matrixCopy(tmp, inv\_tmp, n);

Console.WriteLine("Обратная матрица");

Inversion(inv\_tmp, n);

showMatrix(inv\_tmp, n);

Console.WriteLine();

Console.WriteLine();

Console.WriteLine("Решение СЛАУ методом Гаусса (с выбором главного эл.)");

showMatrix(a, n);

Console.WriteLine();

Console.WriteLine();

Console.WriteLine("Обратная матрица");

Inversion(inv\_a, n);

showMatrix(inv\_a, n);

Console.WriteLine();

Console.WriteLine();

Console.ReadLine();

}

**Результаты вычислений:**

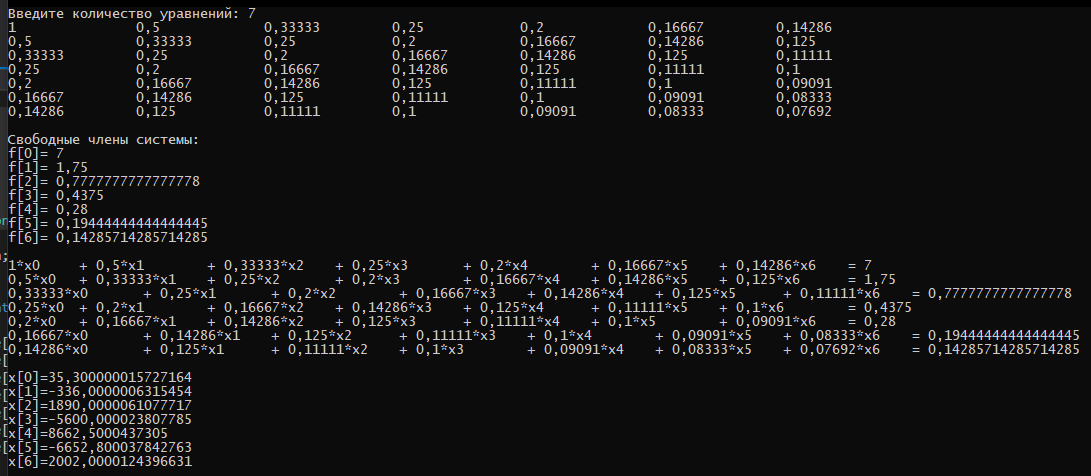


Рисунок 1.1 – Результаты вычислений



Рисунок 1.2 – Результаты вычислений

Выполнил проверку полученных результатов путем подстановки в систему значений x0…xn.

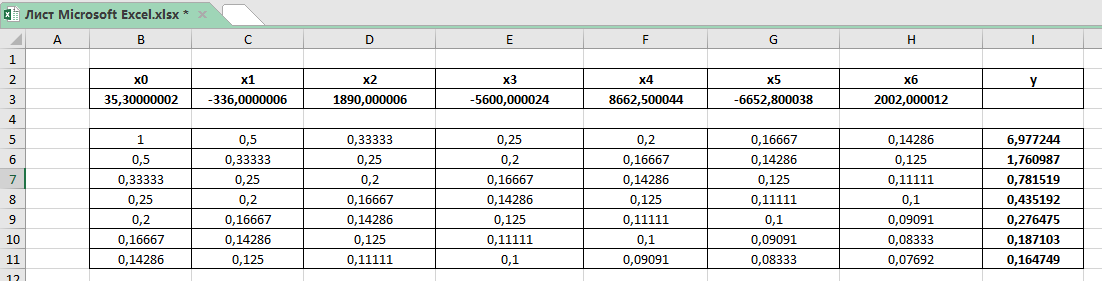


Рисунок 1.3 – Результат подстановки значений в систему уравнений

Произвел расчет числа обусловленности для матрицы в еквклидовой и бесконечной форме полученной при вычислении методом Гаусса с выбором главного элемента (рис. 1.4 – 1.5) в программе PTC Mathcad Express Prime 4.0.

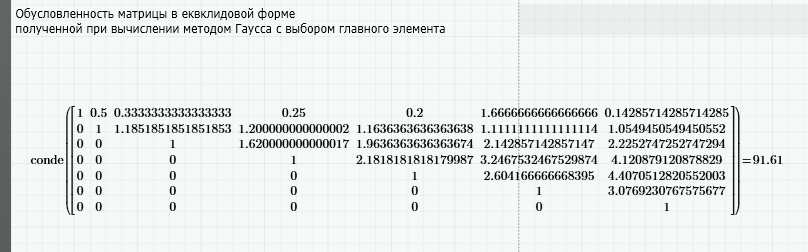


Рисунок 1.4 – Расчет числа обусловленности для матрицы в еквклидовой форме

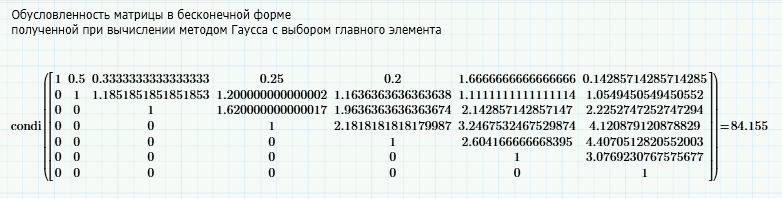


Рисунок 1.5 – Расчет числа обусловленности для матрицы в бесконечной форме

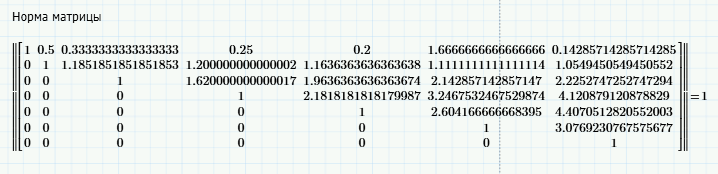


Рисунок 1.5 – Расчет нормы матрицы

Произвел расчет числа обусловленности для матрицы Гильберта для 7 уравнений в еквклидовой и бесконечной форме (рис. 1.6) в программе PTC Mathcad Express Prime 4.0.

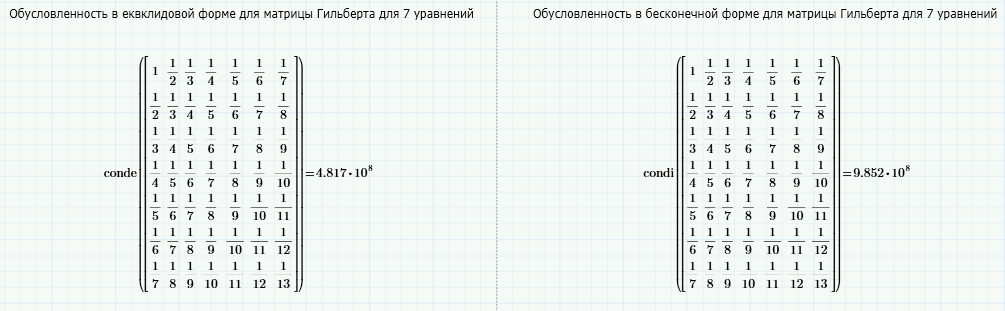


Рисунок 1.6 – Расчет числа обусловленности для матрицы Гильберта для 7 уравнений

**Вывод:**

Число обусловленности матрицы Гильберта далеко не равное 1 и, следовательно, матрица Гильберта является плохо обусловленной матрицей. Так как, если число обусловленности матрицы близко к 1, то это указывает на то, что матрица является хорошо обусловленной.

Методы Гаусса с частичным и полным выбором главного элемента отличаются друг от друга только в способе выбора главного элемента. Процессы пересчета элементов матрицы СЛАУ и вектора правой части в ходе исключения, а также обратный ход у этих двух методов абсолютно одинаковы. Поэтому все отличия, которые возникают у этих двух методов обязаны лишь отличиям в главных элементах.

Вычислительная сложность методов Гаусса с частичным и полным выбором главного элемента определяется в соответствии с формулами и равна https://www.ok-t.ru/studopediaru/baza3/351589208893.files/image086.gif для СЛАУ размером https://www.ok-t.ru/studopediaru/baza3/351589208893.files/image047.gif, но несмотря на то, что в обоих методах общее количество операций определяется как многочлен третьей степени от размера СЛАУ.

Прямой ход метода Гаусса с полным выбором главного элемента (а значит и метод Гаусса с полным выбором главного элемента в целом) требует бóльшего количества операций (за счет операций сравнения), чем прямой ход метода Гаусса с частичным выбором главного элемента (а значит и метод Гаусса с частичным выбором главного элемента в целом).

Если выбор главного элемента вообще не производится, то операции сравнения здесь будут отсутствовать, но пересчет элементов матрицы СЛАУ и вектора правой части будет осуществляться также, как и в методах Гаусса с выбором главного элемента и потребует для этого https://www.ok-t.ru/studopediaru/baza3/351589208893.files/image086.gif арифметических операций, то есть вычислительная сложность метода Гаусса без выбора главного элемента также составляет https://www.ok-t.ru/studopediaru/baza3/351589208893.files/image086.gif.

**Контрольные вопросы**

1. Каков порядок количества вычислений с вещественными числами в методах решения линейных систем *n* уравнений: методом Крамера, методом Гаусса, методом прогонки?

Метод Крамера требует вычисления n!  
Метод Гаусса менее трудоемкий, количество арифметических операций, необходимых для решения линейной системы *n*-го порядка методом Гаусса, составляет примерно 2n3/3

Методы прогонки очень эффективны. Например, при решении линейной системы *n*-го порядка с трехдиагональной матрицей методом правой прогонки требуется выполнить около 10n арифметических операций. При количество операций составит около, то есть почти столько же, сколько требуется для решения системы сотого порядка по методу Гаусса.

1. Что такое главный (ведущий) элемент в методе Гаусса? Как он вычисляется?

Основная идея метода Гаусса с выбором главного элемента состоит в том, чтобы на очередном шаге исключать не следующее по номеру неизвестное, а то неизвестное, коэффициент при котором является наибольшим по модулю. Таким образом, в качестве ведущего элемента здесь выбирается главный, т.е. наибольший по модулю элемент. Тем самым, если , то в процессе вычислений не будет происходить деление на нуль.

1. Как методом Гаусса находят обратные матрицы?

Существуют альтернативные методы нахождения обратной матрицы, например, **метод Гаусса - Жордана.**

Суть метода Гаусса-Жордана заключается в том, что если с единичной матрицей Е провести элементарные преобразования, которыми невырожденная квадратная матрица А приводится к Е, то получится обратная матрица .

1. Что такое хорошо обусловленная матрица? Каковы пределы обусловленности?

Хорошо обусловленной называют матрицу A, для которой матрица устойчива (если ее собственные значения имеют отрицательные действительные части). Такая матрица при погрешности во входных данных будет давать малую погрешность в решении.

Как правило, если число обусловленности {\displaystyle \mu (f)=10^{k}}u(f) = 10k, то вы можете потерять до k цифр точности сверх того, что будет потеряно для числового значения из-за потери точности из арифметических методов.

1. Какой матрицей должна характеризоваться система линейных уравнений для решения её методом прогонки?

Метод прогонки или алгоритм Томаса используется для решения систем линейных уравнений вида, где A — трёхдиагональная матрица. Представляет собой вариант метода последовательного исключения неизвестных.

1. В чём суть итерационных методов решения систем линейных уравнений?

Метод итерации — это численный и приближенный метод решения СЛАУ.

Суть: нахождение по приближённому значению величины следующего приближения, которое является более точным. Метод позволяет получить значения корней системы с заданной точностью в виде предела последовательности некоторых векторов (итерационный процесс). Характер сходимости и сам факт сходимости метода зависит от выбора начального приближения корня x0.

Их сущность состоит в том, что сначала записывается некоторая последовательность столбцов матрицы, после чего производится поочередное вычисление каждого столбца. Каждый новый столбец вычисляется на основе вычисленных предыдущих, при этом с каждым вычислением получается всё более точное приближение искомого решения. Когда достигнута необходимая точность, процесс вычисления прерывают и в качестве решения используют последний вычисленный столбец.