Московский государственный университет

Имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики

Компьютерный практикум по курсу

«Введение в численные методы»

Задание №1

«РЕШЕНИЕ СЛАУ МЕТОДОМ ГАУССА И МЕТОДОМ ГАУССА С ВЫБОРОМ ГЛАВНОГО ЭЛЕМЕНТА»

Отчет о выполненном задании

Студента 203 учебной группы факультета ВМК МГУ

Чернышев Алексей Александрович

Москва, 2015 г.

# Постановка задачи.

Рассмотрим систему линейных алгебраических уравнений , где A  матрица n \* n, x =(x1, x2, … , xn)T  искомый вектор, f = (f1, f2, … , fn)T  заданный вектор. Будем предполагать, что определитель матрицы A отличен от нуля, т.е. решение системы существует. Написать программу решающую систему заданного пользователем размера m методом Гаусса и методом Гаусса с выбором главного элемента.

# Цели и задачи.

1. Решить заданную СЛАУ методом Гаусса и методом Гаусса с выбором главного элемента.
2. Вычислить определитель матрицы det(A).
3. Вычислить обратную матрицу A-1.
4. Определить число обусловленностей MA =
5. Исследовать вопрос об вычислительной устойчивости метода Гаусса (при больших значениях n.
6. Правильность работы программы подтвердить системой тестов.

# Описание метода

Запишем систему Ax *= f,* в развернутом виде

……………………………………………………………...

Метод Гаусса состоит в последовательном исключении неизвестных из этой системы. Предположим, что a11 ≠ 0. Последовательно умножая первое уравнение на и складывая с *i*-м уравнение, исключим X1 из всех уравнений кроме первого. Получим систему

………………………………………………………...

Где , , *I, j = 2, 3, … ,m*

Аналогичным образом из полученной системы исключим X2. Последовательно, исключая все неизвестные, получим систему треугольного вида



Описанная процедура называется прямым ходом метода Гаусса. Заметим, что ее выполнение было возможно при условии, что все ,  не равны нулю.

Выполняя последовательные подстановки в последней системе, (начиная с последнего уравнения) можно получить все значения неизвестных.

, .

Эта процедура называется обратным ходом метода Гаусса.

Один из основных недостатков метода Гаусса связан с тем, что при его реализации накапливается вычислительная погрешность. Показано, что для больших систем порядка *m* число действий умножений и делений близко к .

Для того, чтобы уменьшить рост вычислительной погрешности применяются различные модификации метода Гаусса. В данной работе рассмотрен метод Гаусса с выбором главного элемента по столбцам. Его идея состоит в том, что на каждом этапе прямого хода строки матрицы переставляются таким образом, чтобы диагональный угловой элемент был максимальным. При исключении соответствующего неизвестного из других строк деление будет произведено на наибольший из возможных коэффициентов и, следовательно, относительная погрешность будет наименьшей.

Выполняемые в методе Гаусса преобразования прямого хода, приведшие матрицу ***А*** системы к треугольному виду позволяют вычислить определитель матрицы:

.

Обратную матрицу будем искать методом Гаусса-Жордана. Для применения этого метода в одну матрицу запишем заданную матрицу A и единичную матрицу I, т.е. составим матрицу вида (A|I). После этого с помощью элементарных преобразований, выполняемых со строками расширенной матрицы, добиваются того, что матрица слева от черты станет единичной, причём расширенная матрица примет вид (I|A−1).

Описание работы программы

Пользователь сперва вводит sign (1 или 0), если хочет вводить данные из файла или если хочет вводить данные вручную соответственно. Далее вводится параметр n, указывающий на размер вводимой матрицы. Последний вводимый параметр это sign\_cop, указывающий на подвариант (1, в случае, если вводится сама матрица А и столбец f; и 2 в случае, если вводится параметр M, после чего по нему воспроизводится матрица, в этом случае далее пользователь также вводит параметр n - размера матрицы, и x – некая константа при вычислении матрицы)

После инициализации программа выводит систему линейных алгебраических уравнений, которую будет решать. Сначала выполняется прямой ход метода Гаусса, сразу же считается определитель матрицы А. Если определитель отличен от нуля, то затем выполняется обратный ход метода Гаусса, входе чего находится искомое решение СЛАУ. После программа аналогично решает исходную СЛАУ, только уже с выбором главного элемента по столбцам в методе Гаусса. В конце вычисляется обратная матрица к данной

Текст программы

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <limits.h>**

**#include <math.h>**

**#include <unistd.h>**

**#include <sys/file.h>**

**#include <fcntl.h>**

**#include <locale.h>**

**void**

**forward\_step\_without\_main\_elem(double \*\*mas, double \*f, int n, double \*determ)** \\ Прямой ход в методе Гаусса без

**{** \\ выбора главного элемента

**int i, j, k, s;**

**double det = 1, \*carry, cur;**

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

**s = i;**

**while (mas[s][i] == 0) {**

**s++;**

**if (s >= n) {**

**printf("Ошибка: матрица вырожденная");**

**exit(0);**

**}**

**}**

**if (s != i) {**

**det = -det;**

**}**

**carry = mas[i];**

**mas[i] = mas[s];**

**mas[s] = carry;**

**cur = mas[i][i];**

**det \*= mas[i][i];**

**for (j = i; j < n; j++) {**

**mas[i][j] /= cur;**

**}**

**f[i] /= cur;**

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

**cur = mas[j][i];**

**for(k = i; k < n; k++) {**

**mas[j][k] -= mas[i][k] \* cur;**

**}**

**f[j] -= f[i] \* cur;**

**}**

**}**

**\*determ = det;**

**}**

**void**

**forward\_step\_with\_main\_elem(double \*\*mas, double \*f, int n, int \*pos)** \\ прямой ход в методе Гаусса с выбором

**{** \\ главного элемента

**int i, j, jmax, k, icarry, q;**

**double cur, max;**

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

**max = 0;**

**for (j = i; j < n; j++) {**

**if (fabs(mas[i][j]) > max) {**

**max = fabs(mas[i][j]);**

**jmax = j;**

**}**

**}**

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

**cur = mas[j][i];**

**mas[j][i] = mas[j][jmax];**

**mas[j][jmax] = cur;**

**}**

**icarry = pos[i];**

**pos[i] = pos[jmax];**

**pos[jmax] = icarry;**

**cur = mas[i][i];**

**for (j = i; j < n; j++){**

**mas[i][j] /= cur;**

**}**

**f[i] /= cur;**

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

**cur = mas[j][i];**

**for(k = i; k < n; k++) {**

**mas[j][k] -= mas[i][k] \* cur;**

**}**

**f[j] -= f[i] \* cur;**

**}**

**}**

**}**

**void**

**back\_step(double \*\*mas, double \*f, int n, int \*pos, double \*x)** \\ обратный ход в методе Гаусса

**{**

**int i, j;**

**double \*initx;**

**initx = (double \*) calloc(n, sizeof(double));**

**for (i = n - 1; i >= 0; i--) {**

**x[i] = f[i];**

**for (j = n - 1; j > i; j--) {**

**x[i] -= mas[i][j] \* x[j];**

**}**

**}**

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

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

**if (i == pos[j]) {**

**initx[i] = x[j];**

**}**

**}**

**}**

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

**x[i] = initx[i];**

**}**

**free(initx);**

**}**

**void**

**print\_matrix\_with\_sol(FILE \*fw, double \*\*mas, double \*f, int n)** [\\](file:///\\печатаем) печатаем исходную матрицу

**{**

**int i, j;**

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

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

**fprintf(fw,"a%d%d = %-13.5f", i, j, mas[i][j]);**

**}**

**fprintf(fw, "%f", f[i]);**

**fprintf(fw, "\n");**

**}**

**}**

**void**

**print\_reverse\_matrix(FILE \*fw, double \*\*mas, int n)** [\\ выводим](file:///\\выводим) обратную матрицу

**{**

**int i, j;**

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

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

**fprintf(fw, "a%d%d = %-13.5f ", i, j, mas[i][j]);**

**}**

**fprintf(fw, "\n");**

**}**

**}**

**double**

**\*\*reverse\_matrix(FILE \*fw, double \*\*a, int n)** \\ Нахождение обратной матрицы

**{**

**int i, j, t, k;**

**double \*\*a\_rev, \*carry, cur;**

**a\_rev = (double \*\*) calloc(n, sizeof(double\*));**

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

**a\_rev[i] = (double \*) calloc( n, sizeof(double));**

**}**

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

**a\_rev[i][i] = 1;**

**}**

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

**t = i;**

**while (a[t][i] == 0) {**

**t++;**

**if (t > n){**

**fprintf(fw, "Ошибка: матрица вырожденная");**

**exit(0);**

**}**

**}**

**carry = a[i];**

**a[i] = a[t];**

**a[t] = carry;**

**carry = a\_rev[i];**

**a\_rev[i] = a\_rev[t];**

**a\_rev[t] = carry;**

**cur = a[i][i];**

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

**a[i][j] /= cur;**

**a\_rev[i][j] /= cur;**

**}**

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

**cur = a[j][i];**

**for(k = 0; k < n; k++) {**

**a[j][k] -= a[i][k] \* cur;**

**a\_rev[j][k] -= a\_rev[i][k] \* cur;**

**}**

**}**

**}**

**for (i = n - 1; i >= 0; i--) {**

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

**cur = a[j][i];**

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

**a[j][k] -= a[i][k] \* cur;**

**a\_rev[j][k] -= a\_rev[i][k] \* cur;**

**}**

**}**

**}**

**return a\_rev;**

**}**

**void**

**print\_solution(FILE \*fw, double \*x, int n)** \\ Выводим полученное решение

**{**

**int i;**

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

**fprintf(fw, "x%d = %-13.5f ", i, x[i]);**

**}**

**fprintf(fw, "\n");**

**}**

**int**

**main(void)**

**{**

**setlocale(LC\_ALL, "Russian");**

**FILE \*fw = fopen("output.txt", "w");**

**FILE \*fr = fopen("input.txt", "r");**

**int n, i, j, sign, sign\_con;**

**printf("Введите:\n0, если хотите вводить матрицу вручную\n1, если хотите считать матрицу из файла\n");**

**scanf("%d", &sign);**

**printf ("Введите размер матрицы\n");**

**scanf("%d", &n);**

**double \*\*mas1, \*\*mas2, \*\*mas3, \*f1, \*f2, \*x, \*\*arev, det, q, x\_p;**

**int m;**

**int \*pos;**

**mas1 = (double \*\*) calloc( n, sizeof(double));**

**mas2 = (double \*\*) calloc( n, sizeof(double));**

**mas3 = (double \*\*) calloc( n, sizeof(double));**

**f1 = (double \*) calloc( n, sizeof(double));**

**f2 = (double \*) calloc( n, sizeof(double));**

**x = (double \*) calloc( n, sizeof(double));**

**pos = (int \*) calloc( n, sizeof(int));**

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

**pos[i] = i;**

**}**

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

**mas1[i] = (double \*) calloc(n, sizeof(double));**

**mas2[i] = (double \*) calloc(n, sizeof(double));**

**mas3[i] = (double \*) calloc(n, sizeof(double));**

**}**

**printf("Введите:\n1, если Приложение 1\n2, если Приложение 2\n");**

**scanf("%d", &sign\_con);**

**switch (sign\_con)**

**{**

**case 1:**

**{**

**if (sign) {**

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

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

**fscanf(fr, "%lf", &mas1[i][j]);**

**mas2[i][j] = mas1[i][j];**

**mas3[i][j] = mas1[i][j];**

**}**

**fscanf(fr, "%lf", &f1[i]);**

**f2[i] = f1[i];**

**}**

**} else {**

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

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

**scanf("%lf", &mas1[i][j]);**

**mas2[i][j] = mas1[i][j];**

**mas3[i][j] = mas1[i][j];**

**}**

**scanf("%lf", &f1[i]);**

**f2[i] = f1[i];**

**}**

**}**

**break;**

**}**

**case 2:**

**{**

**if (sign) {**

**fscanf(fr, "%d", &m);**

**fscanf(fr, "%lf", &x\_p);**

**q = 1.001 - 2 \* m \* 0.001;**

**for (i = 1; i <= n; i++) {**

**for (j = 1; j <= n; j++) {**

**if (i == j) {**

**mas1[i - 1][j - 1] = pow(q - 1, i + j);**

**} else {**

**mas1[i - 1][j - 1] = pow(q, i + j) + 0.1 \* (j - i);**

**}**

**mas2[i - 1][j - 1] = mas1[i - 1][j - 1];**

**mas3[i - 1][j - 1] = mas1[i - 1][j - 1];**

**}**

**f1[i - 1] = n \* exp(x\_p / (i + 0.0)) \* cos(x\_p);**

**f2[i - 1] = f1[i - 1];**

**}**

**} else {**

**scanf("%d", &m);**

**scanf("%lf", &x\_p);**

**q = 1.001 - 2 \* m \* 0.001;**

**for (i = 1; i <= n; i++){**

**for (j = 1; j <= n; j++){**

**if (i == j) {**

**mas1[i - 1][j - 1] = pow(q - 1, i + j);**

**} else {**

**mas1[i - 1][j - 1] = pow(q, i + j) + 0.1 \* (j - i);**

**}**

**mas2[i - 1][j - 1] = mas1[i - 1][j - 1];**

**mas3[i - 1][j - 1] = mas1[i - 1][j - 1];**

**}**

**f1[i - 1] = n \* exp((double) (x\_p / i)) \* cos(x\_p);**

**f2[i - 1] = f1[i - 1];**

**}**

**}**

**break;**

**}**

**default: break;**

**}**

**fclose(fr);**

**fprintf(fw, "Исходная матрица:\n");**

**print\_matrix\_with\_sol(fw, mas1, f1, n);**

**forward\_step\_without\_main\_elem(mas1, f1, n, &det);**

**back\_step(mas1, f1, n, pos, x);**

**fprintf(fw, "Решение методом Гаусса без выбора главного элемента:\n");**

**print\_solution(fw, x, n);**

**fprintf(fw, "Определитель = %f\n", det);**

**forward\_step\_with\_main\_elem(mas2, f2, n, pos);**

**back\_step(mas2, f2, n, pos, x);**

**fprintf(fw, "Решение методом Гаусса с выбором главного элемента:\n");**

**print\_solution(fw, x, n);**

**arev = reverse\_matrix(fw, mas3, n);**

**fprintf(fw, "Обратная матрциа:\n");**

**print\_matrix(fw, arev, n);**

**fclose(fw);**

**return 0;**

**}**

# Тесты, подтверждающие правильность работы программы

**Приложение 1. Вариант 12. Слау №1.**

2x1 – 2x2 + x4 = -3

2x1 +3x2 + x3 – 3x4 = -6

3x1 + 4x2 – x3 + 2x4 = 0

x1 + 3x2 + x3 – x4 = 2

Определитель матрицы равен -53.000000

Решение по методу Гаусса без выбора главного значения:

x0 = -2.00000 x1 = 1.00000 x2 = 4.00000 x3 = 3.00000

Решение по методу Гаусса с выбором главного значения:

x0 = -2.00000 x1 = 1.00000 x2 = 4.00000 x3 = 3.00000

Результат верный, проверено на сайте <http://ru.onlinemschool.com/>

Обратная матрица к исходной:

a00 = 0,26415 a01 = 0,16981 a02 = 0,07547 a03 = -0,09434

a10 = -0,16981 a11 = -0,03774 a12 = 0,09434 a13 = 0,13208

a20 = 0,37736 a21 = -0,47170 a22 = -0,32075 a23 = 1,15094

a30 = 0,13208 a31 = -0,41509 a32 = 0,03774 a33 = 0,45283

Результат верный, проверено вручную.

**Приложение 1. Вариант 12. Слау №2.**

1x1 + 3x2 + 2x3 + x4 = 0

2x1 - x2 + 3x3 – 2x4 = 0

3x1 - 5x2 + 4x3 - 3x4 = 0

x1 + 17x2 + 4x3 – 23x4 = 0

Определитель матрицы равен 0.

Результат верный, проверено на сайте <http://ru.onlinemschool.com/>

**Приложение 1. Вариант 2. Слау №3.**

45x1 - 28x2 + 34x3 - 52x4 = 9

36x1 - 23x2 + 29x3 – 43x4 = 3

47x1 - 32x2 + 36x3 - 48x4 = -17

27x1  - 19x2 + 22x3 – 35x4 = 6

Определитель матрицы равен -1440.000000

Результат верный, проверено на сайте <http://ru.onlinemschool.com/>

Решение по методу Гаусса без выбора главного элемента:

x0 = 1.00000 x1 = 2.00000 x2 = -4.00000 x3 = -3.00000

Решение по методу Гаусса с выбором главного элемента:

x0 = 1.00000 x1 = 2.00000 x2 = -4.00000 x3 = -3.00000

Обратная матрица к исходной:

a00 = 0,33333 a01 = -0,22222 a02 = -0,00000 a03 = -0,22222

a10 = 0,07708 a11 = 0,37361 a12 = -0,14375 a13 = -0,37639

a20 = -0,49167 a21 = 0,76111 a22 = 0,02500 a23 = -0,23889

a30 = -0,09375 a31 = 0,10417 a32 = 0,09375 a33 = -0,14583

Результат верный, проверено вручную.

**Приложение 2. Вариант 4.**

Решение по методу Гаусса и решение по методу Гаусса с выбором главного элемента на данных тестах дали одинаковый результат (m = 4, n = 100, x = m):

Матрица:

a00 = 0,00005 a01 = 1,07915 a02 = 1,17229 a03 = 1,26549

a04 = 1,35873 a05 = 1,45202 a06 = 1,54535 a07 = 1,63874

a08 = 1,73216 ………………………………………………………………………..

………………………………………… a9998 = 0,14712 a9999 = 0,00000

x0 = 3425,29895 x1 = 301,28093 x2 = 61,86547 x3 = -10,13609 x4 = -43,32282

x5 = -62,15298 x6 = -74,22374 x7 = -82,60167 x8 = -88,74666 x9 = -93,43757

x10 = -97,1260 x11 = -100,09124 x12 = -102,51446 x13 = -104,5183 x14 = -106,188

x15 = -107,58718 x16 = -108,7588 x17 = -109,7377 x18 = -110,549 x19 = -111,21443

x20 = -111,74758 x21 = -112,16132 x22 = -112,46528 x23 = -112,66715 x24 = -112,77303

x25 = -112,78778 x26 = -112,71527 x27 = -112,55852 x28 = -112,31990 x29 = -112,00121

x30 = -111,60378 x31 = -111,12853 x32 = -110,57606 x33 = -109,94665 x34 = -109,24035

x35 = -108,45698 x36 = -107,59615 x37 = -106,65730 x38 = -105,63974 x39 = -104,54260

x40 = -103,36490 x41 = -102,10555 x42 = -100,76333 x43 = -99,33693 x44 = -97,82495

x45 = -96,22590 x46 = -94,53819 x47 = -92,76018 x48 = -90,89011 x49 = -88,92619

x50 = -86,86652 x51 = -84,70915 x52 = -82,45207 x53 = -80,09318 x54 = -77,63033

x55 = -75,06130 x56 = -72,38379 x57 = -69,59547 x58 = -66,69392 x59 = -63,67665

x60 = -60,54114 x61 = -57,28477 x62 = -53,90487 x63 = -50,39871 x64 = -46,76350

x65 = -42,99636 x66 = -39,09436 x67 = -35,05452 x68 = -30,87377 x69 = -26,54898

x70 = -22,07694 x71 = -17,45439 x72 = -12,67799 x73 = -7,74433 x74 = -2,64993

x75 = 2,60878 x76 = 8,03541 x77 = 13,63368 x78 = 19,40736 x79 = 25,36032

x80 = 31,49649 x81 = 37,81989 x82 = 44,33462 x83 = 51,04486 x84 = 57,95488

x85 = 65,06904 x86 = 72,39175 x87 = 79,92757 x88 = 87,68109 x89 = 95,65702

x90 = 103,86018 x91 = 112,29544 x92 = 120,96780 x93 = 129,88235 x94 = 139,04427

x95 = 148,45885 x96 = 158,13148 x97 = 168,06765 x98 = 178,27297 x99 = 188,75315

Определитель матрицы равен 0.000000

Результат верный, проверено на сайте <http://ru.onlinemschool.com/>

Так же программа тестировалась на собственных тестах, ниже приведены СЛАУ.

**Собственные тесты.**

1.Исходная матрица:

a00 = 0.78000 a01 = 0.56300 f0 = 0.21700

a10 = 0.91300 a11 = 0.65900 f1 = 0.25400

Определитель матрицы равен 0.000001

Решение по методу Гаусса без выбора главного элемента:

x0 = 0.99999 x1 = -0.99999

Решение по методу Гаусса с выбором главного элемента:

x0 = 0.99999 x1 = -0.99999

2. Исходная матрица:

x1 + 2x2 + 6x3 + 6x4 + 4x5 = 1

2x1 + 3x2 + 8x3 + 3x5 = 3

3x1 + 4x2 + 4x3 + 5x4 + 2x5 = 8

4x1 + 5x2 + 3x3 + 4x4 = 3

5x1 + 7x2 + 2x3 + 2x4 + 2x5 = 4

Определитель матрицы равен -119.00000

Решение по методу Гаусса без выбора главного элемента:

x0 = 40,29412 x1 = -30,04202 x2 = -1,74790 x3 = -0,68067 x4 = 8,84034

Решение по методу Гаусса с выбором главного элемента:

x0 = 40,29412 x1 = -30,04202 x2 = -1,74790 x3 = -0,68067 x4 = 8,84034

3. Исходная матрица:

3x1 + x2 + 2x3 + 3x4 + 3x5 = 4

2x1 + 2x2 + 2x3 + 4x4 + 4x5 = 0

6x1 + 4x2 + 3x3 + 6x4 + 5x5 = 6

5x1 + 4x3 + 8x4  + 5x5= 6

4x1 + 4x2 + 5x4 + 4x5 = 4

Определитель матрицы равен -116,000000

Решение по методу Гаусса без выбора главного элемента:

x0 = 2,58621 x1 = -1,24138 x2 = -1,00000 x3 = -0,68966 x4 = 0,51724

Решение по методу Гаусса с выбором главного элемента:

x0 = 2,58621 x1 = -1,24138 x2 = -1,00000 x3 = -0,68966 x4 = 0,51724

**Вывод**

В ходе работы было рассмотрено два метода: метод Гаусса и метод Гаусса с выбором главного элемента. Был изучен принцип метода Гаусса решения системы линейных алгебраических уравнений. Был написан алгоритм вычисления определителя и обратной матрицы.

Метод Гаусса для плохо обусловленных матриц коэффициентов является вычислительно неустойчивым. Так для матриц Гильберта метод приводит к очень большим ошибкам даже при небольшой размерности этих матриц. Уменьшить вычислительную ошибку можно с помощью метода Гаусса с выделением главного элемента, который является условно устойчивым.