ИНСТИТУТ ТРАНСПОРТА И СВЯЗИ

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ НАУК И ЭЛЕКТРОНИКИ

Лабораторная работа

По дисциплине

«Численные методы»

Тема: **Методы решения системы линейных уравнений. Число обусловленности матрицы.**

Студент: Виктор Выползов
Группа: 4102BD

Рига
2014 г.

1. Задание

В данной лабораторной работе рассматриваются прямые и итерационные методы. Ниже приведены системы линейных алгебраических уравнений (СЛАУ), которые следует решить, используя указанные методы из Таблицы 1. Предположим, по индивидуальному заданию требуется реализовать метод Краута. Данный метод относиться к прямым методам, поэтому для демонстрации результата реализованного алгоритма следует показать решение СЛАУ под номерами 1, 2 и 5. В результатах работы алгоритма указываются номера систем и, соответствующие им, полученные вектора неизвестных. Так же, в работе требуется произвести экспериментальный расчет числа обусловленности матрицы для двух систем: 3 и 5. Выполнить расчет можно как программным путем, так и ручным решением. Для вычислений требуется выбрать одну из двух норм: Манхетонская или Евклидова нормы.

Задачи для решения:

1) $$\begin{cases} x_1 - 2 \cdot x_2 + x_3 = 2 \\ 2 \cdot x_1 - 5 \cdot x_2 - x_3 = -1 \\ -7 \cdot x_1 + x_3 = -2 \end{cases}$$

2) $$\begin{cases} 5 \cdot x_1 - 5 \cdot x_2 - 3 \cdot x_3 + 4 \cdot x_4 = -11 \\ x_1 - 4 \cdot x_2 + 6 \cdot x_3 - 4 \cdot x_4 = -10 \\ -2 \cdot x_1 - 5 \cdot x_2 + 4 \cdot x_3 - 5 \cdot x_4 = -12 \\ -3 \cdot x_1 - 3 \cdot x_2 + 5 \cdot x_3 - 5 \cdot x_4 = 8 \end{cases}$$

3) $$\begin{cases} 2 \cdot x_1 - x_2 - x_3 = 5 \\ x_1 + 3 \cdot x_2 - 2 \cdot x_3 = 7 \\ x_1 + 2 \cdot x_2 + 3 \cdot x_3 = 10 \end{cases}$$

4) $$\begin{cases} 8 \cdot x_1 + 5 \cdot x_2 + 3 \cdot x_3 = 30 \\ -2 \cdot x_1 + 8 \cdot x_2 + x_3 = 15 \\ x_1 + 3 \cdot x_2 - 10 \cdot x_3 = 42 \end{cases}$$

5) $$\begin{cases} 0{,}78 \cdot x_1 + 0{,}563 \cdot x_2 = 0{,}217 \\ 0{,}913 \cdot x_1 + 0{,}659 \cdot x_2 = 0{,}254 \end{cases}$$

Таблица 1

Метод	Номер СЛАУ
Итерационные методы	1, 3, 4
Прямые методы	1, 2, 5

2. Метод исключения Гаусса с ведущим элементом

- **Листинг программной реализации алгоритма**

```
function X = gauss(A,B)
  C=[A, B];
  [m,n] = size(A);

  if(m == n && m == length(B) && det(A) ~= 0)

    % ведущий элемент
    for i=1:(n-1)
      max = abs(C(i,i));
      maxj = i;
      row = C(i,:);

      for j=(i+1):n
        if(max<abs(C(j,i)))
          max = abs(C(j,i));
          maxj = j;
        end
      end

      C(i,:) = C(maxj,:);
      C(maxj,:) = row;
    end

    % прямой ход
    for i=1:(n-1)
      a = C(i,i);

      for j=(i+1):n
        b = -C(j,i)/a;
        row = C(i,:) * b;
        C(j,:) = C(j,:) + row;
      end

    end

    X = zeros(n,1);
    [m,n] = size(C);

    % обратный ход
    for i=m:-1:1
      x = C(i,n);

      c = m;
      for j=i:m-1
        x = x - (C(i,c)*(X(c,1)));
        c = c - 1;
      end

      X(i,1) = x/C(i,i);
    end

    check = zeros(m,1);
    for i=1:m
      tmp = 0;
      for j=1:m
        tmp = tmp + A(i,j)*X(j,1);
      end
      check(i) = tmp;
```

```
        end

        X
        check

      else
        disp('invalid data. exit');
      end
    end
```

- **Результаты работы алгоритма на указанных СЛАУ**
  - Пример 1 — gauss([1 -2 1; 2 -5 -1; -7 0 1],[2; -1; -2]);
    Результат — X=[0.5200; 0.0800; 1.6400]
    Проверка — check=[2;-1;-2]
  - Пример 2 — gauss([5 -5 -3 4; 1 -4 6 -4; -2 -5 4 -5; -3 -3 5 -5],[-11;-10;-12;8]);
    Результат — X=[-12.8235;-2.2941,11.7647;19.2353]
    Проверка — check=[-11.0000;-10.0000;-12.0000,8.0000]
  - Пример 5 — gauss([0.78 0.563; 0.913 0.659],[0.217; 0.254]);
    Результат — X=[1; -1]
    Проверка — check=[0.2170;0.2540]

3. Метод простых итераций

- **Листинг программной реализации алгоритма**

```
function X = iter(A,B,e)
   [m,n]=size(A);

   if(m == n && m == length(B) && det(A) ~= 0)
      convergence=true;

      for i=1:n
         a = abs(A(i,i));
         row_sum = sum(abs(A(i,:))) - a;
         col_sum = sum(abs(A(:,i))) - a;
         if (a < row_sum) || (a < col_sum)
            convergence = false;
            break;
         end
      end

      if convergence == true
         alpha = zeros(m,n);
         beta = zeros(m,1);
         X0 = beta;

         for i=1:n
           beta(i) = B(i)/A(i,i);
           for j=1:n
             if i~=j
               alpha(i,j) = -A(i,j)/A(i,i);
             end
           end
         end

         iter_count = 1;
         Xj = alpha*X0 + beta;
         while norm(Xj - X0) > e
            X0 = Xj;
            Xj = alpha*X0 + beta;
            iter_count = iter_count + 1;
         end

         X = Xj;

         check = zeros(m,1);
         for i=1:n
            tmp = 0;
            for j=1:n
               tmp = tmp + A(i,j)*X(j,1);
            end
            check(i) = tmp;
         end

         X
         check
         iter_count

      else
        disp('convergence == false. exit');
      end

   else
```

```
        disp('invalid data. exit');
    end
end
```

- **Результаты работы алгоритма на указанных СЛАУ**
    - Пример 1 — iter([1 -2 1; 2 -5 -1; 0 -7 1],[2; -1; -2],10^-5);
      Результат — convergence == false. exit. (нет сходимости)
    - Пример 3 — iter([2 -1 -1; 1 3 -2; 1 2 3],[5; 7; 10],10^-5);
      Результат — X=[3.8125; 1.6875; 0.9375]
      Проверка — check=[5.0000;7.0000;10.0000]
      Количество итераций — 105
    - Пример 4 — iter([0.78 0.563; 0.913 0.659],[0.217; 0.254],10^-5);
      Результат — X=[3;3;-3]
      Проверка — check=[30.0001;15.0000;42.0000]
      Количество итераций — 19

4. Экспериментальное определение числа обусловленности матрицы

Формула:

$$\frac{\|\Delta X\|}{\|X\|} \leqslant Cond(A) * \frac{\|\Delta B\|}{\|B\|}$$

Найдем экспериментальную обусловленность для первого примера, прибавим к коэффициентам матрицы B по 0.02 и решим уравнение.

Результат: X=[0.5256; 0.0824; 1.6592];

$$\Delta X = \begin{bmatrix} 0.0056 \\ 0.0024 \\ 0.0192 \end{bmatrix}$$

$$\|\Delta X\| = 0.0201$$

$$\|X\| = 1.7223$$

$$\Delta B = \begin{bmatrix} -0.02 \\ 0.02 \\ 0.02 \end{bmatrix}$$

$$\|\Delta B\| = 0.0346$$

$$\|B\| = 3$$

$$Cond(A) = 5.9104$$

0.0116 <= 5.9104*0.0115 → 0.0116 <= 0.0679

5. Выводы

Реализовав два метода, метод исключения Гаусса и метод простых итераций, для решения СЛАУ я с уверенностью могу сказать, что мне больше понравился метод исключения Гаусса. Это два простых метода и реализовать их не составила труда, но метод простых итераций хуже по тем причинам, что мы сначала должны провести проверку на сходимость, а так же должны указывать точность и соответственно на это влияет кол-во итераций. То есть метод исключения Гаусса решает все простые СЛАУ, а в методе простых итераций это не всегда возможно из-за сходимости, а так же дополнительное не удобство в виде точности добавляется. Мой выбор остается за метод исключения Гаусса.