РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теориии вероятностей

Лабораторная работа №4. Метод Гаусса и операции линейной алгебры.

Студент: Ухарова Софья

Группа: НПМ-02-22

Москва 2022

Цель работы

Ознакомиться с методом Гаусса для решения систем линейных уравнений и реализовать некоторые операции линейной алгебры, включая реализацию LU-разложения.

Задачи

1. На практике реализовать метод Гаусса для решения СЛУ
2. Изучить дополнительные операциии языка для операций по нахождению решений СЛУ
3. Изучить алгоритм LU-разложения
4. Реализоовать LU-разложения

Содержание

[Заключение][]

1. Метод Гаусса

Для решения системы линейных уравнений

$$ Ax = b $$

методом Гаусса, построим расширенную матрицу вида

$$ B = (A|b) $$

Для примера взята следующая расширенная матрица:

$$ B = \left(\begin{array}{rrr|r} 1&2&3&4\\ 0&-2&-4&6\\ 1&-1&0&0 \end{array}\right) $$

Создадим эту матрицу в оболочке GNU Octave:

```
>> B = [ 1 2 3 4 ; 0 -2 -4 6 ; 1 -1 0 0 ]
B =

   1   2   3   4
   0  -2  -4   6
   1  -1   0   0
```

Проведем над ней операции по поиску её отдельного элемента на строке 2, столбце 3; извлечем вектор 1 строки.

```
>> B(2,3)
ans = -4
>> B(1,:)
ans =

   1   2   3   4
```

Теперь явно реализуем метод Гаусса.

Сначала добавим к третьей строке первую строку, умноженную на -1:

```
>> B(3,:) = (-1)*B(1,:) + B(3,:)
B =

   1   2   3   4
   0  -2  -4   6
   0  -3  -3  -4
```

Далее добавим к третьей строке втору, умноженную на -1.5:

```
>> B(3,:) = -1.5 * B(2,:) + B(3,:)
B =

     1     2     3     4
     0    -2    -4     6
     0     0     3   -13
```

Теперь наша матрица имеет треугольный вид. Теперь мы точно знаем, что элемент x_3 вектора решений x будет равен $-\frac{13}{3}$. Чтобы найти остальные координаты x_1 и x_2:

- обнулим x_3 во второй строке;
- обнулим $x_{2,3}$ в первой строке;

```
>> B(2,:) = -3*B(2,:) - 4*B(3,:)
B =

     1     2     3     4
    -0     6     0    34
     0     0     3   -13

>> B(1,:) = 3*B(1,:) - B(2,:) - 3*B(3,:)
B =

     3     0     0    17
    -0     6     0    34
     0     0     3   -13
```

Теперь мы можем составить наш искомый вектор решений x:

```
>> x = [ B(1,4)/B(1,1) B(2,4)/B(2,2) B(3,4)/B(3,3) ]
x =

    5.6667    5.6667   -4.3333
```

Что и дает при переводе в обыкновенные дроби:

$$ x = (\frac{17}{3} \frac{17}{3} \frac{-13}{3}) $$

Воспользуемся встренным инструментом `rref` для поиска треугольной матрицы, а также изменим количество знаков после запятой:

```
>> rref(B)
ans =

   1.00000   0.00000   0.00000   5.66667
  -0.00000   1.00000   0.00000   5.66667
   0.00000   0.00000   1.00000  -4.33333

>> format long
>> rref(B)
ans =

   1.000000000000000e+00   0.000000000000000e+00   0.000000000000000e+00
  -0.000000000000000e+00   1.000000000000000e+00   0.000000000000000e+00
   0.000000000000000e+00   0.000000000000000e+00   1.000000000000000e+00

>> format short
```

2. Левое деление

Воспользуемся встроенной операцией для решения линейных систем вида \$Ax = b\$ - левым делением, в виде `A\b`.

Выделим матрицу \$A\$ и векторм \$b\$ из расширенной матрицы \$B\$, вернув её в изначальное состояние.

```
>> B = [ 1 2 3 4 ; 0 -2 -4 6 ; 1 -1 0 0 ]
B =

   1   2   3   4
   0  -2  -4   6
   1  -1   0   0

>> A = B(:,1:3)
A =

   1   2   3
   0  -2  -4
   1  -1   0

>> b = B (:,4)
b =

   4
   6
   0
```

Теперь непосредственно найдем вектор x:

```
>> A\b
ans =

   5.6667
   5.6667
  -4.3333
```

3. LU-разложение

Нам необходимо представить матрицу A в виде: $$ A = LU, $$

где L - нижняя треугольная матрица вида:

$$ L = \begin{pmatrix} 1&0&0\\ l_{21}&1&0\\ l_{31}&l_{32}&1 \end{pmatrix}, $$

а U - верхняя треугольная матрица вида:

$$ U = \begin{pmatrix} u_{11}&u_{12}&u_{13}\\ 0&u_{22}&u_{23}\\ 0&0&u_{33} \end{pmatrix}, $$

Необходимы следующие условия для существования LU-разложения:

- матрица A обратима: $\exists A^{-1}$
- все главные миноры матрицы A невырождены: $|\Delta_{(1,..,n)}| \neq 0$

Если мы проверим все эти условия для данной нам матрицы

$$ A = \left(\begin{array}{rrr} 1&2&3\\ 0&-2&-4\\ 1&-1&0 \end{array}\right), $$

то убедимся, что все они выполняются для неё.

Алгоритм LU-разложения

Пусть

$U = A$

$L = I$, где I матрица порядка, эквивалентного порядку n матрицы A.

Далее, повторяем следующие последовательные действия:

$$ \left.\begin{array}{l} l_{ij}=\frac{u_{ij}}{u_{jj}} \\ u_i = u_i-u_j\cdot l_{ij} \end{array} \right\} \left.\begin{array}{c} i=2, ..., n \\ j=1, ..., i \end{array}\right. $$

где u_{i} и u_{j} - i-ая строка и j-ый столбец матрицы U.

Реализация в коде

Итоговый скрипт `lu.m` выглядит следующим образом:

```
A = input("Enter the matrix A: ")
disp("----------------")
L = eye(3)
disp("Making U = A")
U = A
disp("----------------")

for i=2:3
    disp("Step "), disp(i-1)
    for j=1:i-1
        L(i,j) = U(i,j)/U(j,j)
        U(i,:) = U(i,:) - U(j,:)*L(i,j)
    end
    disp("----------------")
end

disp("Check the answer of L*U: "), disp(L*U)
```

Мы самостоятельно вводим любую матрицу, за исключением того что мы должны заранее проверить на ранее описанные условия, т.к. скриптом предусмотрено, что мы уверены в вверности вводимой нами матрицы.

Далее, мы создаем матрицы \$L\$ и \$U\$, приравнивая их к соответствующим матрицам, после чего исполняем цикл.

Для проверки результата, произведение матриц \$LU\$ должно равняться матрице \$A\$.

Вывод скрипта `lu.m`:

```
Enter the matrix A: [1 2 3; 0 -2 -4; 1 -1 0]
A =

   1   2   3
   0  -2  -4
   1  -1   0

----------------
L =

Diagonal Matrix

   1   0   0
```

```
   0   1   0
   0   0   1

Making U = A
U =

   1   2   3
   0  -2  -4
   1  -1   0

----------------
Step
 1
L =

   1   0   0
   0   1   0
   0   0   1

U =

   1   2   3
   0  -2  -4
   1  -1   0

----------------
Step
 2
L =

   1   0   0
   0   1   0
   1   0   1

U =

   1   2   3
   0  -2  -4
   0  -3  -3

L =

   1.00000   0.00000   0.00000
   0.00000   1.00000   0.00000
   1.00000   1.50000   1.00000

U =

```

```
   1   2   3
   0  -2  -4
   0   0   3

----------------
Check the answer of L*U:
   1   2   3
   0  -2  -4
   1  -1   0
```

Видим, что произведение матриц совпадает с нашей изначальной матрицей. Теперь, проверим свойства, вытекающие из этого произведения (произведение определителей, решение СЛУ):

```
>> detofLU = det(L)*det(U)
detofLU = -6
>> detofA = det(A)
detofA = -6
>> detofA == detofLU
ans = 1
>> y = L\b
y =

    4
    6
  -13

>> x = U\y
x =

   5.6667
   5.6667
  -4.3333
```

Найденные нами матрицы абсолютно верны.

Заключение

Мы успешно реализовали метод Гаусса для решения систем линейных уравнений и реализовали некоторые операции линейной алгебры, включая LU-разложение, в IDE языка Octave.