Naïve Gaussian Elimination Method
University of South Florida
United States of America
kaw@eng.usf.edu

NOTE: This worksheet demonstrates the use of Matlab to illustrate Naïve Gaussian Elimination, a numerical technique used in solving a system of simultaneous linear equations.

*************************************Introduction***************************************

One of the most popular numerical techniques for solving simultaneous linear equations is Naïve Gaussian Elimination method.
The approach is designed to solve a set of n equations with n unknowns,
[A][X]=[C], where [A]nxn is a square coefficient matrix, [X]nx1 is the solution vector, and [C]nx1 is the right hand side array.

Naïve Gauss consists of two steps:
1) Forward Elimination: In this step, the coefficient matrix [A] is reduced to an upper triangular matrix. This way, the equations are "reduced" to one equation and one unknown
in each equation.
2) Back Substitution: In this step, starting from the last equation, each of the unknowns is found.

A simulation of Naïve Gauss Elimination Method follows.

**************************************Input Data**
Below are the input parameters to begin the simulation.
Input Parameters:
n = number of equations
[A] = nxn coefficient matrix
[RHS] = nx1 right hand side array

These are the default parameters used in the simulation.
They can be changed in the top part of the M-file

n= 6

A =

12	7e-013	3	6.0007	5	6
1	5	1	9	7	8
13	12	4	8	4	6
5.6	3	7	1.003	7	4
1	2	3	4	5	6
6	7	5	6	7	5

RHS =

```
    22
7e-007
29.001
 5.301
     9
    90

---------------------------------------------------
With these inputs,to conduct Naïve Gauss Elimination, Matlab will combine the [A] and
[RHS] matrices into one augmented matrix, [C](n*(n+1)), that will facilitate the process
of forward elimination.

C =

 12    7e-013          3    6.0007         5         6        22
  1         5          1         9         7         8    7e-007
 13        12          4         8         4         6    29.001
5.6         3          7     1.003         7         4     5.301
  1         2          3         4         5         6         9
  6         7          5         6         7         5        90

*************************************Forward Elimination**********************************
Forward elimination consists of (n-1) steps. In each step k of forward elimination,
the coefficient element of the kth unknown will be zeroed from every
subsequent equation that follows the kth row. For example, in step 2 (i.e. k=2),
the coefficient of x2 will be zeroed from rows 3..n.
With each step that is conducted, a new matrix is generated until the coefficient matrix
is
transformed to an upper triangular matrix. Now, Matlab calculates the upper triangular
matrix while demonstrating the intermediate coefficient matrices that are produced for
each step k.

================== Step 1 =======================

C =

12    7e-013         3    6.0007         5         6        22
 0         5      0.75    8.4999    6.5833       7.5   -1.8333
 0        12      0.75    1.4992   -1.4167      -0.5    5.1677
 0         3       5.6   -1.7973    4.6667       1.2   -4.9657
 0         2      2.75    3.4999    4.5833       5.5    7.1667
 0         7       3.5    2.9996       4.5         2        79

The elements in column #1 below C[1,1] are zeroed

================== Step 2 =======================

C =

12    7e-013         3    6.0007         5         6        22
 0         5      0.75    8.4999    6.5833       7.5   -1.8333
 0         0     -1.05   -18.901   -17.217     -18.5    9.5677
```

```
 0         0      5.15   -6.8973   0.71667      -3.3   -3.8657
 0         0      2.45  0.099965      1.95       2.5       7.9
 0         0      2.45   -8.9003   -4.7167      -8.5    81.567

The elements in column #2 below C[2,2] are zeroed

================== Step 3 =======================

C =

12    7e-013         3    6.0007         5         6        22
 0         5      0.75    8.4999    6.5833       7.5   -1.8333
 0         0     -1.05   -18.901   -17.217     -18.5    9.5677
 0         0         0     -99.6   -83.727   -94.038    43.061
 0         0         0   -44.001   -38.222   -40.667    30.225
 0         0         0   -53.002   -44.889   -51.667    103.89

The elements in column #3 below C[3,3] are zeroed

================== Step 4 =======================

C =

12    7e-013         3    6.0007         5         6        22
 0         5      0.75    8.4999    6.5833       7.5   -1.8333
 0         0     -1.05   -18.901   -17.217     -18.5    9.5677
 0         0         0     -99.6   -83.727   -94.038    43.061
 0         0         0         0   -1.2333   0.87753    11.201
 0         0         0         0  -0.33408   -1.6249    80.976

The elements in column #4 below C[4,4] are zeroed

================== Step 5 =======================

C =

12    7e-013         3    6.0007         5         6        22
 0         5      0.75    8.4999    6.5833       7.5   -1.8333
 0         0     -1.05   -18.901   -17.217     -18.5    9.5677
 0         0         0     -99.6   -83.727   -94.038    43.061
 0         0         0         0   -1.2333   0.87753    11.201
 0         0         0         0         0   -1.8626    77.942

The elements in column #5 below C[5,5] are zeroed

This is the end of the forward elimination steps. The coefficient matrix
has been reduced to an upper triangular matrix

A =

12    7e-013         3    6.0007         5         6
 0         5      0.75    8.4999    6.5833       7.5
```

```
 0         0     -1.05   -18.901   -17.217     -18.5
 0         0         0     -99.6   -83.727   -94.038
 0         0         0         0   -1.2333   0.87753
 0         0         0         0         0   -1.8626


RHS =

     22
-1.8333
 9.5677
 43.061
 11.201
 77.942

*************************************Back substitution************************************

Back substitution begins with solving the last equation as it has only one unknown.
The remaining equations can be solved for using the following formula:

               x[i]=(C[i]-(sum{A[i,j]*X[j]}))/(A[i,i]


Using back substitution, we get the unknowns as:

X =

      -41.846
      -38.858
       71.742
       73.925
      -19.484
      -15.409

>>
```