10. Naive Gauss Elimination Method

Cheng Peng

Lecture Note for MAT325 Numerical Analysis

Contents

1 Introduction

We discuss numerical solutions to the linear system of equations. Consider the following general linear system of equations

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$$

$$\vdots \qquad\qquad\qquad \vdots$$

$$a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n$$

The goal is to find the solution to this system. There are different methods to solve the linear system. One way is to re-express the linear system into a matrix equation and then solve the matrix equation.

Example 1 Consider the following linear system of functions

$$
\begin{aligned}
3x_1 - x_2 + 2x_3 &= 8 \\
2x_1 - 2x_2 + 3x_3 &= 2 \\
4x_1 + x_2 - 4x_3 &= 9
\end{aligned}
$$

We can re-write the above system of linear equations into the following matrix form

$$\begin{bmatrix} 3 & -1 & 2 \\ 2 & -2 & 3 \\ 4 & 1 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 2 \\ 9 \end{bmatrix}$$

$$\mathbf{A} \qquad\qquad \mathbf{X} \qquad\qquad \mathbf{B}$$

We denote

$$\mathbf{A} = \begin{bmatrix} 3 & -1 & 2 \\ 2 & -2 & 3 \\ 4 & 1 & -4 \end{bmatrix}$$

be the coefficient matrix.

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

the vector of unknown, and

$$\mathbf{B} = \begin{bmatrix} 8 \\ 2 \\ 9 \end{bmatrix}$$

Then, the matrix form of the equation is equivalent to $\mathbf{AX} = \mathbf{B}$. If the inverse of $\mathbf{A}$, denoted by $\mathbf{A}^{-1}$, exists, we multiply $\mathbf{A}^{-1}$ to the left of the matrix equation to get the solution of the linear system in the following

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{B}$$

Finding the inverse of the matrix could be a challenge when the system is large. The R built-in function `solve()` to calculate the inverse of a matrix (if it exists). For example, the inverse of $\mathbf{A}$ can be found using the following code.

```
A = matrix(c(3, -1, 2, 2, -2, 3, 4, 1, -4), ncol = 3, byrow = TRUE)
A.inv = solve(A)
A.inv        # print out the

##            [,1]        [,2]        [,3]
## [1,] 0.3333333 -0.1333333  0.06666667
## [2,] 1.3333333 -1.3333333 -0.33333333
## [3,] 0.6666667 -0.4666667 -0.26666667

# matrix multiplication
A%*%A.inv    # expected to be the identity matrix

##      [,1]         [,2]         [,3]
## [1,]    1 -2.220446e-16 -2.220446e-16
## [2,]    0  1.000000e+00 -1.110223e-16
## [3,]    0  0.000000e+00  1.000000e+00
```

Two ways to find the solution to the system

```
B = c(8,2,9)
X.1 = solve(A)%*%B   # returns a matrix form solution
X.1
```

```
##      [,1]
## [1,]    3
## [2,]    5
## [3,]    2
```

```
X.2 = solve(A,B)    # returns a vector form solution
X.2
```

```
## [1] 3 5 2
```

2 Naive Gaussian Elimination Methods

Linear algebra texts discuss elimination methods to find the solution. The method has two processes: elimination and backward substitution. We use the equation in the above **Example 1** to review the elimination method.

Recall the linear system

$$
\begin{aligned}
3x_1 - x_2 + 2x_3 &= 8 \\
2x_1 - 2x_2 + 3x_3 &= 2 \\
4x_1 + x_2 - 4x_3 &= 9
\end{aligned}
$$

We define the following augmented matrix.

$$
\mathbf{A} = \begin{bmatrix} 3 & -1 & 2 & \vdots & 8 \\ 2 & -2 & 3 & \vdots & 2 \\ 4 & 1 & -4 & \vdots & 9 \end{bmatrix}
$$

Solution We show the detailed steps of forward elimination and backward substitution.

Forward Elimination

We use R_1, R_2, and R_3 to denote the three rows of the above-augmented matrix and then find the solution in the following $r = 3 - 1$ steps:

Step 1 The equation in the first row is called **pivot equation** and $a_{11} = 3$ is called **pivot element**. We first do the following row operations so that the elements in the first column of the resulting **new augmented** matrix are all zeros except for the one in the *pivot* element.

1.1. Multiply $-a_{21}/a_{11}$ to R_1 and add it to R_2 to update R_2. **Row 2**: $i = 2$

$$
R_2 \leftarrow R_2 - (a_{21}/a_{11}) \times R_1 = (a_{21}, a_{22}, a_{23}, b_2) - \frac{a_{21}}{a_{11}}(a_{11}, a_{12}, a_{13}, b_1)
$$

$$
= \left(a_{21} - \frac{a_{21}}{a_{11}} \times a_{11}, a_{22} - \frac{a_{21}}{a_{11}} \times a_{12}, a_{23} - \frac{a_{21}}{a_{11}} \times a_{11}, b_2 - \frac{a_{21}}{a_{11}} \times a_{11}\right)
$$

$$
= (2, -2, 3, 2) - \frac{2}{3}(3, -1, 2, 8) = (0, -4/3, 5/3, -10/3)
$$

1.2 Multiply $-4/3$ to R_1 and add it to R_3 to update R_3. **Row 3**: $i = 3$

$$R_3 \leftarrow R_3 - (a_{31}/a_{11}) \times R1 = (a_{31}, a_{32}, a_{33}, b_3) - \frac{a_{31}}{a_{11}}(a_{11}, a_{12}, a_{13}, b_1)$$

$$= (4, 1, -4, 9) - \frac{4}{3}(3, -1, 2, 8) = \left(0, \frac{7}{3}, -\frac{20}{3}, -\frac{5}{3}\right)$$

After this step, we have the updated augmented matrix for the next step elimination.

$$\begin{bmatrix} 3 & -1 & 2 & \vdots & 8 \\ 0 & -4/3 & 5/3 & \vdots & -10/3 \\ 0 & 7/3 & -20/3 & \vdots & -5/3 \end{bmatrix}$$

Step 2: The equation in the first row is called *pivot equation* and $a_{22} = -4/3$ (**CAUTION**: *using the above updated augmented matrix!*)is called pivot element. We next do row operation: $R_3 - \frac{a_{32}}{a_{22}} \times R_2$ so that the resulting new augmented matrix. **Row 3**: $r = 3$.

$$R_3 \leftarrow R_3 - \frac{a_{32}}{a_{22}} \times R_2 = (a_{31}, a_{32}, a_{33}, b_3) - \frac{a_{32}}{a_{22}}(a_{21}, a_{22}, a_{23}, b_2)$$

$$= \left(0, \frac{7}{3}, -\frac{20}{3}, -\frac{5}{3}\right) - \frac{7/3}{-4/3}\left(0, -\frac{4}{3}, \frac{5}{3}, -\frac{10}{3}\right) = \left(0, \ 0, \ -\frac{15}{4}, -\frac{15}{2}\right)$$

With the above row operation, we have the following updated augmented matrix that has a triangular coefficient matrix.

$$\begin{bmatrix} 3 & -1 & 2 & \vdots & 8 \\ 0 & -4/3 & 5/3 & \vdots & -10/3 \\ 0 & 0 & -15/4 & \vdots & -15/2 \end{bmatrix} .$$

If we re-write the above-augmented matrix into the linear system of equations, we have

$$\begin{cases} 3x_1 & -x_2 & 2x_3 & = & 8 \\ & -\frac{4}{3}x_2 & \frac{5}{3}x_3 & = & -\frac{10}{3} \\ & & -\frac{15}{4}x_3 & = & -\frac{15}{2} \end{cases}$$

Backward Substitution

Next, we do backward substitution to find the solution to the system. The general backward substitution is given by

$$x_i = \frac{1}{a_{ii}}\left(b_i - \sum_{j=i+1}^{3} a_{ij}x_j\right),$$

where $i = 3 - 1, 3 - 2$. However $i = 3$

$$x_3 = \frac{b_3}{a_{33}} = \frac{-15/2}{-15/4} = 2.$$

We start from **row 3** of the last augmented matrix:

Row #3: $r = 3$:

$$x_3 = \frac{b_3}{a_{33}} = \frac{-15/2}{-15/4} = 2.$$

Row #2: $r = 2$

$$x_2 = \frac{1}{a_{22}}(b_2 - a_{23}x_3) = -\frac{1}{4/3}\left(-\frac{10}{3} - \frac{5}{3} \times 2\right) = \frac{3}{4}\frac{20}{3} = 5.$$

Row #1: $r = 1$

$$x_1 = \frac{1}{a_{11}}(b_1 - [a_{12}x_2 + a_{13}x_3]) = \frac{1}{3}(8 - [-1 \times 5 + 2 \times 2]) = \frac{1}{3}(8 - (-1)) = 3.$$

Therefore, the solution to the above system linear equation is $(x_1, x_2, x_3) = (3, 5, 2)$.

Remark: We re-arrange the following linear system of linear equations

$$\begin{cases} 3x_1 & -x_2 & 2x_3 & = & 8 \\ & -\frac{4}{3}x_2 & \frac{5}{3}x_3 & = & -\frac{10}{3} \\ & & -\frac{15}{4}x_3 & = & -\frac{15}{2} \end{cases}$$

to get the following equivalent linear system

$$\begin{cases} -\frac{15}{4}x_3 & & & = & -\frac{15}{2} \\ \frac{5}{3}x_3 & -\frac{4}{3}x_2 & & = & -\frac{10}{3} \\ 2x_3 & -x_2 & +3x_1 & = & 8 \end{cases}$$

That can be re-written in the following matrix form

$$\begin{bmatrix} -\frac{15}{4} & 0 & 0 \\ \frac{5}{3} & -\frac{4}{3} & 0 \\ 2 & -1 & 3 \end{bmatrix} \begin{bmatrix} x_3 \\ x_2 \\ x_1 \end{bmatrix} = \begin{bmatrix} -\frac{15}{2} \\ -\frac{10}{3} \\ 8 \end{bmatrix}$$

The augmented matrix of the above linear system of equations can be written as

$$\begin{bmatrix} -\frac{15}{4} & 0 & 0 & \vdots & -\frac{15}{2} \\ \frac{5}{3} & -\frac{4}{3} & 0 & \vdots & -\frac{10}{3} \\ 2 & -1 & 3 & \vdots & 8 \end{bmatrix}$$

We can repeat the Gaussian **forward elimination** with the above-augmented matrix:

1. pivot element $a_{11} = -\frac{15}{4}$

$$R_2 \leftarrow R_2 - \frac{a_{21}}{a_{11}} R_1 = \left(\frac{5}{3}, -\frac{4}{3}, 0, -\frac{10}{3}\right) - \frac{5/3}{-15/4}\left(-\frac{15}{4}, 0, 0, -\frac{15}{2}\right) = \left(0, -\frac{4}{3}, 0, -\frac{20}{3}\right)$$

$$R_3 \leftarrow R_3 - \frac{a_{31}}{a_{11}} R_1 = (2, -1, 3, 8) - \frac{2}{-15/4}\left(-\frac{15}{4}, 0, 0, -\frac{15}{2}\right) = (0, -1, 3, 4)$$

We obtain the following updated augmented matrix

$$\begin{bmatrix} -\frac{15}{4} & 0 & 0 & \vdots & -\frac{15}{2} \\ 0 & -\frac{4}{3} & 0 & \vdots & -\frac{20}{3} \\ 0 & -1 & 3 & \vdots & 4 \end{bmatrix}$$

We need one more row operation on the above-augmented matrix

$$R_3 \leftarrow R_3 - \frac{a_{32}}{a_{22}} R_2 = (0, -1, 3, 4) - \frac{-1}{-4/3}\left(0, -\frac{4}{3}, 0, -\frac{20}{3}\right) = (0, 0, 3, 9)$$

The final augmented matrix has the following form

$$\begin{bmatrix} -\frac{15}{4} & 0 & 0 & \vdots & -\frac{15}{2} \\ 0 & -\frac{4}{3} & 0 & \vdots & -\frac{20}{3} \\ 0 & 0 & 3 & \vdots & 9 \end{bmatrix}.$$

Its equivalent linear system is given by

$$\begin{cases} -\frac{15}{4}x_3 &= -\frac{15}{2} \\ -\frac{4}{3}x_2 &= -\frac{20}{3} \\ 3x_1 &= 9 \end{cases}$$

Therefore, the solution to the original system of equation is (using vector operation - pair-wise division)

$$(x_3, x_2, x_1) = \frac{(-15/2, -20/3, 9)}{(-15/4, -4/3, 3)} = (2, 5, 3)$$

We will apply the logic used in this example to implement the **Gaussian Elimination method** (using *backward substitution*).

3 Gaussian Elimination Algorithm

In the elimination step, the first row of the augmented matrix is the basis, and all other elements (cells) in the augmented matrix are updated iteratively using the algorithm using the recursive formula

$$R_i \leftarrow (a_{i1}, a_{i2}, \cdots, a_{in}, b_i) - \frac{a_{ik}}{a_{kk}}(a_{k1}, a_{k2}, \cdots, a_{kn}, b_k)$$

where

- $k = 1, 2, \cdots, n-1$, the iteration index for selecting the *pivot element*.

- $i = k+1, k+2, ...n$, the iterator of the row of the augmented matrix.

4 Gaussian Elimination Algorithm

We focus on Gaussian row elimination. The Gaussian forward elimination and backward substitution (Gauss-Jordan) are optional.

4.1 Gauss Forward Elimination

The algorithm of Gaussian is relatively simple compared with the iterative algorithms we learned earlier. We only need to convert the following recursive equation to an algorithm with double loops to accomplish the Gaussian forward elimination.

Gaussian Forward Elimination Algorithm

```
INPUT: A    (augmented matrix)

OUTPUT: AO  (row-echelon matrix)

STEP 1: find n = # of row

STEP 2: FOR i = 1 TO n
        STEP 3:  IF A[i,i] == 0 DO:
                    find j with A[j,i] != 0
                    swap row i and row j
                ENDIF

        STEP 4:  IF A[i,i] != 0 DO:
                    FOR j = i + 1 TO n DO:
                      A[j,] = A[j,]-(A[j,i]/A[i,i])*A[i,]
                    ENDFOR
                ENDIF
STEP 5: RETURN AO
```

The vector operation was used in **Step 4** of the pseudo-code. Row-swapping in **step 3** could also involve vector operation.

Next, we write R code to implement the pseudo-code.

```r
Gauss.Elimination = function(A){
  # Input A: augmented matrix. Make sure that he diagonal
  #          elements of the coefficient matrix are non-zero!
  AO = A
  n = dim(AO)[1]                   # number of rows
  for(i in 1:(n-1)){               # iterator fo pivot element AO[i,i]
                                   # i = 1, 2, ..., n-1
     #~~~~~~~~~~~~~~~~~~~~~~~  Make the function robust  ~~~~~~~~~~~~~~~~~~~~~~~~~#
     if(AO[i,i] == 0){
         non.0 = which(AO[,i] != 0)
             #cat("\n\n i =",i, ", non.0 =", non.0,". (i+1):n =", (i+1):n, ".")
             id = intersect(non.0, (i+1):n)[1] # which() equiv to find() in MATLAB
             if(!is.na(id)){
             tempi = AO[i,]                        # put the i-th row in a place-holder
```

7

```
            A0[i,] = A0[id,]                          # row swapping
            A0[id,] = tempi
         }
      }
      #~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
       if(A0[i,i] != 0){      # A0[i,i] is in the denominator of recursive equation
          for(j in (i+1):n){  # to make the augmented matrix in the row echelon form
            #cat("\n\n j =",j,".")
            A0[j,] = A0[j,] - (A0[j,i]/A0[i,i])*A0[i,]
         }
      }
   }
   A0
}
```

We now look at several examples using the above R function `Gauss.Elimination()`.

Example 2: Find the solution of the following system of linear equations.

$$
\begin{array}{rrrrrrrr}
x & - & 3y & + & 4z & = & 1 \\
2w & - & 2x & + & y & & = & -1 \\
2w & - & x & - & 2y & + & 4z & = & 0 \\
-6w & + & 4x & + & 3y & - & 8z & = & 1
\end{array}
$$

Solution: Performing Gaussian eliminations (i.e., the series of row operations), we have

$$
\begin{bmatrix}
0 & 1 & -3 & 4 & 1 \\
2 & -2 & 1 & 0 & -1 \\
2 & -1 & -2 & 4 & 0 \\
-6 & 4 & 3 & -8 & 1
\end{bmatrix}
\xrightarrow{r_1 \leftrightarrow r_2}
\begin{bmatrix}
2 & -2 & 1 & 0 & -1 \\
0 & 1 & -3 & 4 & 1 \\
2 & -1 & -2 & 4 & 0 \\
-6 & 4 & 3 & -8 & 1
\end{bmatrix}
$$

$$
\xrightarrow[3r_1 \text{ added to } r_4]{-r_1 \text{ added to } r_3}
\begin{bmatrix}
2 & -2 & 1 & 0 & -1 \\
0 & 1 & -3 & 4 & 1 \\
0 & 1 & -3 & 4 & 1 \\
0 & -2 & 6 & -8 & -2
\end{bmatrix}
$$

$$
\xrightarrow[2r_2 \text{ added to } r_4]{-r_2 \text{ added to } r_3}
\begin{bmatrix}
2 & -2 & 1 & 0 & -1 \\
0 & 1 & -3 & 4 & 1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

```
##
A1 = matrix(c(0,1,-3,4,1,2,-2,1,0,-1,2,-1,-2,4,0,-6,4,3,-8,1), ncol =5, byrow = TRUE)
Gauss.Elimination(A = A1)

##      [,1] [,2] [,3] [,4] [,5]
## [1,]    2   -2    1    0   -1
## [2,]    0    1   -3    4    1
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
```

Example 3 Solve the following system of linear equations.

$$
\begin{aligned}
x + y - 3z &= 4 \\
2x + y - z &= 2 \\
3x + 2y - 4z &= 7
\end{aligned}
$$

Solution: Performing Gaussian eliminations (i.e., the series of row operations), we have

$$
\begin{bmatrix} 1 & 1 & -3 & 4 \\ 2 & 1 & -1 & 2 \\ 3 & 2 & -4 & 7 \end{bmatrix}
\xrightarrow[\substack{-3r_1 \text{ added to } r_3}]{-2r_1 \text{ added to } r_2}
\begin{bmatrix} 1 & 1 & -3 & 4 \\ 0 & -1 & 5 & -6 \\ 0 & -1 & 5 & -5 \end{bmatrix}
$$

$$
\xrightarrow{-r_2 \text{ added to } r_3}
\begin{bmatrix} 1 & 1 & -3 & 4 \\ 0 & -1 & 5 & -6 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

```
A1 = matrix(c(1,1,-3,4,2,1,-1,2,3,2,-4,7), ncol = 4, byrow = TRUE)
Gauss.Elimination(A = A1)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    1   -3    4
## [2,]    0   -1    5   -6
## [3,]    0    0    0    1
```

Example 4 Solve the following system of linear equations.

$$
\begin{aligned}
\tfrac{1}{4}a + \tfrac{1}{2}b + c &= \tfrac{23}{4} \\
a + b + c &= 7 \\
4a + 2b + c &= 2
\end{aligned}
$$

Solution: Performing Gaussian eliminations (i.e., the series of row operations), we have

$$
\begin{bmatrix} \tfrac{1}{4} & \tfrac{1}{2} & 1 & \tfrac{23}{4} \\ 1 & 1 & 1 & 7 \\ 4 & 2 & 1 & 2 \end{bmatrix}
\xrightarrow{4r_1}
\begin{bmatrix} 1 & 2 & 4 & 23 \\ 1 & 1 & 1 & 7 \\ 4 & 2 & 1 & 2 \end{bmatrix}
$$

$$
\xrightarrow[\substack{-4r_1 \text{ added to } r_3}]{-r_1 \text{ added to } r_2}
\begin{bmatrix} 1 & 2 & 4 & 23 \\ 0 & -1 & -3 & -16 \\ 0 & -6 & -15 & -90 \end{bmatrix}
$$

$$
\xrightarrow[\substack{-r_2}]{-6r_2 \text{ added to } r_3}
\begin{bmatrix} 1 & 2 & 4 & 23 \\ 0 & 1 & 3 & 16 \\ 0 & 0 & 3 & 6 \end{bmatrix}
$$

```
A1 = matrix(c(1/4, 1/2, 1, 23/4, 1,1,1,7,4,2,1,2), ncol = 4, byrow = TRUE)
Gauss.Elimination(A = A1)
```

```
##      [,1] [,2] [,3]    [,4]
## [1,] 0.25  0.5    1    5.75
## [2,] 0.00 -1.0   -3  -16.00
## [3,] 0.00  0.0    3    6.00
```

4.2   Gauss-Jordan Elimination (`Optional`)

```
####
Gauss.Jordan = function(A){
  # A is the augmented matrix
  A1 = Gauss.Elimination(A)              # perform Gauss elimination
  n = length(A1[,1])
  id.0 = rep(1,n)                        # indicator of zero A[i,i]
  for (i in 1:n){
    if(A1[i,i] == 0) id.0[i] = 0
  }
  if(sum(id.0) == n){                    # A[i,i] != 0
      k = n
      A2 = A1[k:1, c(k:1,(n+1))]         # re-arrange rows and columns
      A3 = Gauss.Elimination(A2)         # perform Gauss elimination
      A4 = A3[k:1, c(k:1,(n+1))]         # backward re-arrangement
    } else{                              # if exists A[i,i] =0
      k = which(id.0==0)[1]-1   # select rows and columns for re-arrangement
      A2 = A1[c(k:1,(k+1):n), c(k:1,(k+1):(n+1))]
      A3 = Gauss.Elimination(A2)
      A4 = A3[c(k:1,(k+1):n), c(k:1,(k+1):(n+1))]
    }
  A4
}
```

Example 5 (continuation of **Example 4**)

We can perform additional row operations (Gaussian-Jordan) to get

$$
\begin{bmatrix} 1 & 2 & 4 & 23 \\ 0 & 1 & 3 & 16 \\ 0 & 0 & 3 & 6 \end{bmatrix} \xrightarrow[\frac{1}{3}r_3]{-r_3 \text{ added to } r_2} \begin{bmatrix} 1 & 2 & 4 & 23 \\ 0 & 1 & 0 & 10 \\ 0 & 0 & 1 & 2 \end{bmatrix}
$$

$$
\xrightarrow{-4r_3 \text{ added to } r_1} \begin{bmatrix} 1 & 2 & 0 & 15 \\ 0 & 1 & 0 & 10 \\ 0 & 0 & 1 & 2 \end{bmatrix}
$$

$$
\xrightarrow{-2r_2 \text{ added to } r_1} \begin{bmatrix} 1 & 0 & 0 & -5 \\ 0 & 1 & 0 & 10 \\ 0 & 0 & 1 & 2 \end{bmatrix}
$$

```
A1 = matrix(c(1/4, 1/2, 1, 23/4, 1,1,1,7,4,2,1,2), ncol = 4, byrow = TRUE)
Gauss.Jordan(A = A1)
```

```
##      [,1] [,2] [,3]    [,4]
## [1,] 0.25    0    0   -1.25
## [2,] 0.00   -1    0  -10.00
```

```
## [3,] 0.00    0    3   6.00
```

Example 6 (continuation of **Example 2**) Gauss-Jordan elimination

```
A6 = matrix(c(0,1,-3,4,1,2,-2,1,0,-1,2,-1,-2,4,0,-6,4,3,-8,1), ncol =5, byrow = TRUE)
Gauss.Jordan(A = A6)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    2    0   -5    8    1
## [2,]    0    1   -3    4    1
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
```

Example 7 (continuation of **Example 3**) Gauss-Jordan form

```
A7 = matrix(c(1,1,-3,4,2,1,-1,2,3,2,-4,7), ncol = 4, byrow = TRUE)
Gauss.Jordan(A = A7)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    2   -2
## [2,]    0   -1    5   -6
## [3,]    0    0    0    1
```

Example 8 (continuation of **Example 4**) Gauss-Jordan form

```
A8 = matrix(c(1/4, 1/2, 1, 23/4, 1,1,1,7,4,2,1,2), ncol = 4, byrow = TRUE)
Gauss.Jordan(A = A8)
```

```
##      [,1] [,2] [,3]    [,4]
## [1,] 0.25    0    0   -1.25
## [2,] 0.00   -1    0  -10.00
## [3,] 0.00    0    3    6.00
```

5   Chapter 6 HW Part I

6.1.8. Use the code in the note to solve the linear systems.

6.2.10 (d). **optional**, need to modify the code by using three-digit chopping approach in each row operation.