CSE 190 Lab 2:
Elimination

In this lab, you will practice using the elimination algorithm to solve systems of linear equations.

1 Matrix form of linear equations

We’ll be writing systems of linear equation as \( Ax = b \), where \( A \) is the coefficient matrix, \( x \) is the vector of variables, and \( b \) is the RHS vector. For example:

\[
A = \begin{bmatrix}
2 & 1 & 1 \\
4 & -6 & 0 \\
-2 & 7 & 2
\end{bmatrix},
\]
\[
x = \begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix},
\]
\[
b = \begin{bmatrix}
5 \\
-2 \\
9
\end{bmatrix}.
\] (1)

2 Elementary row operations

The elementary row operations of elimination are of the form

Subtract \( c \) times row \( j \) from row \( i \)

where \( c \) is a scalar (i.e. a real number), and \( i \) and \( j \) specify rows in a matrix. These operations are performed on the rows of \( A \) and \( b \) simultaneously (i.e. on both sides of the equal sign). You’ve undoubtedly used these operations to solve systems of linear equations by hand; here we’ll see them in the context of the elimination algorithm.

Elimination is the process of using elementary row operations to transform the coefficient matrix to another matrix with a particular form called upper triangular. A matrix is upper triangular if the first column has only zeros below the first row, the second column has only zeros below the second row, and so on. For example, the following matrices are upper triangular:

\[
\begin{bmatrix}
2 & 1 & 1 \\
0 & -8 & -2 \\
0 & 0 & 1
\end{bmatrix}, \begin{bmatrix}
2 & 1 & 1 & 5 \\
0 & -8 & -2 & -12 \\
0 & 0 & 1 & 2 \\
0 & 0 & 0 & 0
\end{bmatrix}.
\] (2)
The non-square-ness of the last two matrices should not alarm you; they still meet the criteria we set for being upper triangular.

2.1 Applying elementary operations in Matlab

We’ve prepared a Matlab program called `Elimination` that lets you interactively apply elementary row operations to a matrix.

```
A = [2,1,1;4,-6,0;-2,7,2] % Coefficient matrix
b = [5;-2;9] % RHS vector
M = [A,b] % Form augmented matrix
Elimination(M) % Start the program
```

Using the augmented matrix $M = [A, b]$ is a trick to force `Elimination` to apply the row operations to $A$ and $b$ simultaneously; the RHS vector is always the last column of the augmented matrix. After starting the `Elimination` program, you’ll see the current input matrix and be prompted for a command. Suppose we want to subtract 2 times the first row from the second row. Type `s` and `Enter` to issue the “Subtract $c$ times row $j$ from row $i$” command. You’ll be prompted for the values $c$, $j$, and $i$; after supplying these (2, 1, and 2), the program will print something called an elementary matrix (ignore this for now) followed by the resulting matrix $M$ after applying the elementary operation. You should see that the desired operation has been applied. You can now issue more commands to further transform the matrix. (Tip: when supplying the value $c$, you can enter a fraction like $3/2$.) To quit the program, type `q` and `Enter`; you’ll be returned to the usual Matlab prompt.

In the next few exercises, we won’t worry about applying the row operations to $b$; instead we’ll just work on transforming the matrix $A$ to upper triangular form. (So calling `Elimination(A)` is enough.)

Exercise 1. Transform the following matrices to upper triangular form. To do this, you may use the `Elimination` program to manually carry out the
steps of the elimination algorithm.

\[ A = \begin{bmatrix}
1 & 2 & 3 \\
4 & 12 & 17 \\
10 & 32 & 51
\end{bmatrix} \]

\[ A = \begin{bmatrix}
1 & 0 & 0 \\
2 & 1 & 0 \\
-3 & 4 & 1
\end{bmatrix} \]

\[ A = \begin{bmatrix}
4 & 3 & 2 & 1 \\
-16 & -9 & -6 & -3 \\
12 & 3 & 4 & 2 \\
-8 & -3 & -10 & -4
\end{bmatrix} \]

\[ A = \begin{bmatrix}
1 & -3 & 0 & -4 & 0 \\
2 & -1 & -4 & 0 & -1 \\
-5 & 0 & 0 & 1 & -3
\end{bmatrix} \]

Write down the operations used (e.g. “Subtract \(-2\) times row 1 from row 2; subtract \(1\) times row 1 from row 3”) as well as the resulting upper triangular matrix.

In Exercise 1, you should have only needed to use the “Subtract …” command to transform the matrices to upper triangular form. But sometimes we are not so lucky:

\[ \begin{bmatrix}
1 & 1 & 1 \\
2 & 2 & 5 \\
4 & 6 & 8
\end{bmatrix} \rightarrow \begin{bmatrix}
1 & 1 & 1 \\
0 & 0 & 3 \\
4 & 6 & 8
\end{bmatrix} \rightarrow \begin{bmatrix}
1 & 1 & 1 \\
0 & 0 & 3 \\
0 & 0 & 2
\end{bmatrix} \quad (3) \]

(the first operation is “Subtract \(2\) times row 1 from row 2”; the second operation is “Subtract \(4\) times row 1 from row 3”). We seem stuck; subtracting any multiple of the second row from the third row will not be able to eliminate the \(2\) in the third row. But the second row already has a zero in the same column; if the second and third rows were swapped, we would have the upper triangular form! This is easily accomplished in the Elimination program. Type \(\text{Exchange row } i \text{ and row } j\) to issue the

Exchange row \(i\) and row \(j\)

command; you will be prompted to supply \(i\) and \(j\).

Now, armed with the “Subtract …” and “Exchange …” commands, you should be able to transform any matrix into upper triangular form.
Exercise 2. Same as Exercise 1, but now with the following matrices:

\[
A = \begin{bmatrix}
1 & 1 & 1 \\
3 & 3 & -1 \\
1 & -1 & 1
\end{bmatrix}
\]

\[
A = \begin{bmatrix}
3 & 2 & 4 & 1 \\
0 & 0 & 1 & 1 \\
-6 & -1 & -7 & 2 \\
9 & 9 & 13 & 9
\end{bmatrix}
\]

Again, write down the operations used (e.g. “Subtract $-2$ times row 1 from row 2; subtract 1 times row 1 from row 3; exchange rows 2 and 3”) as well as the resulting upper triangular matrix.

Exercise 3. Here’s a generic system of three linear equations with three unknown:

\[
\begin{bmatrix}
A_{11} & A_{12} & A_{13} \\
A_{21} & A_{22} & A_{23} \\
A_{31} & A_{32} & A_{33}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix}
= \begin{bmatrix}
b_1 \\
b_2 \\
b_3
\end{bmatrix}
\]

(4)

Suppose elimination is run on this system. True or false:

1. If the third equation starts with a zero coefficient (i.e. $A_{31} = 0$), then no multiple of the first equation will be subtracted from the third equation.

2. If the third equation has a zero as its second coefficient (i.e. $A_{32} = 0$), then no multiple of the second equation will be subtracted from the third equation.

3. If $A_{31} = A_{32} = 0$, then no multiple of either the first or second equation will be subtracted from the third equation.

Briefly justify your answers.

3 Elementary matrices

It turns out that every elementary row operation “Subtract $c$ times row $j$ from row $i$” applied to a matrix $A$ can be written as a product $EA$. For example, if $A$ is $3 \times 3$, then multiplying $A$ on the left by

\[
E = \begin{bmatrix}
1 & 0 & 0 \\
-2 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

(5)
will subtract 2 times the first row from the second row. In general, to subtract \( c \) times row \( j \) from row \( i \) (for \( i \neq j \)), if \( A \) is \( m \times n \), then \( E \) is an \( m \times m \) matrix with 1s along the diagonal, \( -c \) in the \((i,j)\)th position (i.e. row \( i \), column \( j \)), and 0s everywhere else.

The Elimination program produces an elementary matrix for each row operation. They are displayed after each operation, and can also be retrieved after quitting the Elimination program by assigning the outputs of the program:

\[
[M,E] = \text{Elimination}([2,1,1;4,-6,0;-2,7,2]);
\]

\( M \) % the result after applying row operations

\( E \) % the elementary matrices

\( E(:,:,1) \) % the first elementary matrix

\( E(:,:,2) \) % the second elementary matrix

Here, \( E \) is a three-dimensional array (in the sense that a vector is a one-dimensional array, and a matrix is a two-dimensional array). This is just a convenient way for Matlab to store a bunch of matrices in a single variable.

**Exercise 4.** For each of the four matrices \( A \) in Exercise 1, write down the product of elementary matrices and \( A \) that results in a upper triangular matrix. For example:

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 4 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 1 & 0 \\
0 & 1 & 0 \\
3 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 \\
2 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 1 & 1 \\
-2 & -1 & -1 \\
-3 & -7 & -6
\end{bmatrix}
= \begin{bmatrix}
1 & 1 & 1 \\
0 & 1 & 1 \\
0 & 0 & 1
\end{bmatrix}.
\]

Remember, matrix multiplication is not commutative (i.e. \( AB \neq BA \) in general), so it is important to write the matrices in the correct order.

Note that it isn’t possible to repeat the previous exercise for the matrices in Exercise 2. This is because those matrices required row exchanges to complete the process of transforming \( A \) to upper triangular. But it is possible if one is allowed to re-order the rows of \( A \) beforehand.

**Exercise 5.** Repeat Exercise 4 for the matrices \( A \) in Exercise 2, except instead of necessarily multiplying the elementary matrices by \( A \), you may multiply them by a matrix the same as \( A \) except with possibly re-ordered rows. For example, if

\[
A = \begin{bmatrix}
0 & -1 & -1 \\
1 & 1 & 1 \\
-3 & -7 & -6
\end{bmatrix}
\]
then you may write

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 4 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
3 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 1 & 1 \\
0 & -1 & -1 \\
-3 & -7 & -6
\end{bmatrix} =
\begin{bmatrix}
1 & 1 & 1 \\
0 & 1 & 1 \\
0 & 0 & 1
\end{bmatrix}
\]

(we swapped the first and second rows of \(A\)).

**Exercise 6.** What are the \(3 \times 3\) matrices that:

1. Adds 3 times the first row to the second row?

2. Adds 3 times the first row to the second row, and adds 3 times the first row to the third row?

3. Adds 3 times the first row to the third row, and at the same time adds 3 times the third row to the first row?

4. Adds 3 times the first row to the third row, and then adds 3 times the third row to the first row?

5. Exchanges the first and second rows?

6. Exchanges the first and second rows, and then exchanges the first and third rows?

Verify your answers by multiplying these matrices by \(x = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}\).