**Solving Linear Equations**

Table of Contents

[Gaussian Elimination 2](#_Toc66212197)

[Permutations 8](#_Toc66212198)

[Inversion 9](#_Toc66212199)

[Gauss – Jordon Elimination 12](#_Toc66212200)

[Factorization into : 14](#_Toc66212201)

[Actually Using Factorization 16](#_Toc66212202)

[Why You Would Want to Use This Seemingly Long Process (Time Complexity) 18](#_Toc66212203)

[Permutation and Transposition 19](#_Toc66212204)

[Symmetric Matrices 21](#_Toc66212205)

## Gaussian Elimination

Consider the following set of equations:

The coefficient matrix is:

The idea behind elimination is to multiply one of the equations with the right number so as to eliminate one of the variables through subtraction. Here, we can multiply the first equation by 3 so that can be eliminated.

Now we change the matrix so that it looks like this:

If we multiply the first row by 3, we get the numbers 3, 6 and 3 respectively. The second row of the matrix is the result of subtracting these numbers from the corresponding numbers of the previous second row.

The first row is known as the pivot row and the top left number, 1, which we used as reference is known as a pivot, specifically, the first pivot.

We have eliminated the number in the 2-1 position of the matrix. The next step would be to eliminate the 3-1 position, but in our case, this is already 0.

Now we look at the second pivot, which is the centre number, 2. We are now trying to eliminate the number in the 3-2 position. Thus, the matrix becomes:

As we can see, the exact same procedure has been followed.

The third pivot is the bottom right number, 5. The matrix we have now is known as an upper triangular, or .

An important note is that none of the pivots can ever be 0. If a 0 is in a pivot position, then rows must be exchanged in such a way so as to prevent that from happening. This is allowed since the order of the equations is irrelevant. However, there might still be situations where even this does not solve the problem. In those cases, the Gaussian Elimination method cannot be used to solve this system.

The next step is backward substitution. First, we add an extra column for the right-hand side of the system, . The original matrix becomes:

This is known as the augmented matrix.

The augmented matrix underwent the following changes:

The final vector, is labelled . Thus, is converted to and is converted to .

Now, from the matrix, we have the equations

We can solve these from bottom to top. Thus, , and .

In the previous lecture we went over the theory of Gaussian Elimination. Here, we will look at how to actually perform the elimination using matrices.

Before starting, we need to be clear about how matrix multiplication works. Take the multiplication below:

This is the multiplication of a 3x3 matrix with a 3x1 matrix. The result should also be a 3x1 matrix. In matrices, we multiple a row of the left-hand side matrix, with a column of the right-hand side matrix, to obtain the result of a specific cell. For example, the result of the cell 1-1 is found by multiplying the first row of the left-hand side matrix with the first column of the right-hand side matrix.

Thus, , and .

That was very lengthy to think about, but is what is happening in the depth of the multiplication. We can think of this in a simpler way however. The above example was the multiplication of a 3x3 matrix with a column matrix. Next, look at the multiplication of a row matrix with a 3x3 matrix.

In the depths, the same procedure is being followed, with cell 1-1 being found by multiplying row 1 from the left, and column 1 from the right, cell 1-2 being found by multiplying row 1 from the left and column 2 from the right and so on. A really simple way of thinking about this is we multiply the 1st row on the right by 3, the 2nd row by 4 and the 3rd row by 5, and then add the numbers on the same column. This is actually the exact same process, with just a different view. Thus, after the multiplications we have

and adding everything in the same column, we get

The thinking process is very informal, but that doesn’t matter. Going back to the column matrix, we multiply column 1 by 3, column 2 by 4 and column 3 by 5, and then add everything on the same row. Thus,

We can now begin looking at Gaussian Elimination again. Our initial matrix was and we converted it to .

We need to find a matrix that is multiplied with our original matrix, to give us the resulting matrix. From the explanation on matrix multiplication above, it is clear the 1st row of the resulting matrix is related to the first row of our required transformation matrix. Since the first row remains unchanged, the first row of the transformation matrix must be , since we take the first row, and the other two rows. Similarly, the third row also remains unchanged, so the third row of the transformation matrix must be .

Only the second-row changes. To perform the change, remember that we took the first row, and subtracted it from the second row. Matrices don’t concern subtraction though, so we need to think of this as adding the 1st row to the second row. Thus, the second row of our transformation matrix must be since the third row of the original matrix has no effect in the change. Thus,

The calculation is . This transformation matrix is known as the matrix, since it eliminates the number at the 2-1 position.

The 3-1 position is already 0, so we move onto the 3-2 position. The 1st and 2nd rows remain unchanged, so they keep their identity values, and the third row is the result of the second row the third row. Thus,

Thus, . This transformation matrix is the matrix.

We have now converted to . In order to do it, we needed to multiply with , as , and then multiply the result with , thus giving .

The Associative Law of Matrix Multiplication states that the order of multiplication can be changed but the order of the matrices cannot. This means but .

Thus, our result can be rewritten as . The multiplication operation of will give us a single 3x3 matrix. This is essentially the combination of the elimination matrices, , the elementary (or elimination) matrix. Thus, .

From this, we can see the first and second rows of can be taken directly. The third row is the second row of the third row of . This gives us . Thus,

## Permutations

Sometimes, if one of the pivots becomes , we need to switch rows. Take the following matrix:

and have been performed on it. Before performing we need to change the rows, since the cell in position 2-2 cannot be 0. We could not perform this operation earlier, since cell 3-1 had a value and switching it would bring back a value for cell 2-1, which we just removed.

Here, there will be no change to row 1. Row 2 will be the result of adding row 1 row 2 row 3. Similarly, row 3 will be row 1 row 2 row 3. Thus,

The permutation matrix is called , since it switches row 2 and row 3. Our overall equation now stands:

We can also switch columns in the same way, though this will not be needed here. However, since we are switching columns, we need to switch the order of the matrices. Thus, switching column 2 and column 3 looks like this:

So, multiplying on the left is for row operations, and multiplying on the right is for column operations.

## Inversion

An inverse matrix will take us back to the original matrix. Remember the matrix :

This matrix tells us to subtract row 1 from row 2. To get back the original matrix, we would simply have to add row 1 to row 2. Thus,

and

.

Also notice that

Thus, multiplying a matrix by its inverse will give us the identity matrix.

Keep in mind however, that not all inverses can be found by simply flipping the sign on the number. Here, we actually figured out what the process would be to undo a certain effect (add row 1 to row 2) and created the inverse from it.

We know that only square matrices have inverses, but not all square matrices have inverses. For example, the matrix has no inverse. It is because the determinant is . We need to understand what this means though.

The job of a transformation matrix is to change an area. The matrix covers an area of 1, while the matrix covers an area of 6. The change in area is 6. The determinant is . Thus, the determinant is the scaling factor by which the area covered by a matrix is decreased or increased after linear transformation.

Not all matrices cause a change in area though. Consider the shear matrix . It has a determinant of . Thus, there is no change in area. There are also negative determinants, that indicate a change in orientation of one axis with respect to another. For example, the matrix causes the positive -axis to point to the left of the positive -axis, and the matrix has a determinant .

Matrices that have a determinant have columns that are linearly dependent. For example, the matrix is linearly dependent and has a determinant . Such matrices, in 2D space, squeeze the area of the matrix onto a single line or point, thus leaving it no area. In 3D space, they squeeze the area of the matrix onto a single line or plane or point.

The job of an inverse matrix is to undo an operation. If we use a matrix to stretch an object vertically to 3 times its height, the inverse matrix will edit the resultant matrix to 1/3rd its height, to bring it back to the original.

With matrices that squeeze objects onto a single point or line, that is, matrices with a zero determinant, it is not possible to bring back the original objects. There are an infinite number of ways we could stretch a point or line, so it is impossible to know what its original shape was.

Another way of looking at this is, for a matrix , if there is some combination of its columns that gives us a column vector , then that matrix cannot have an inverse. For this case, if we multiply the matrix by , we got the column vector . Thus, . The reason this cannot have an inverse is, if we try to find , we will always get the result , which is not true. Thus, the inverse matrix does not exist.

Since the inverse of a matrix undoes an operation, , the identity matrix.

## Gauss – Jordon Elimination

To find the inverse of a matrix, we can use the following method.

We know that, for some matrices, . Combining the two matrices we know about, we can write:

We treat this entire thing as a single matrix and apply Gauss’s elimination method on it. Thus,

If we do not end up with a in the position (the one below here), and instead have , then we must divide the whole row by . Since both sides are being divided, this makes no difference.

Next, we turn to Jordon, who suggests we perform upward elimination like this:

The top row in the result is the bottom row the top row from the left-side matrix.

In the result, the square matrix on the right side of the separator is the inverse of the original matrix. We can check this.

The example below is for a matrix:

## Factorization into :

For the system looks like this:

Thus, for ,

This is a fairly simple inversion. The stands for upper triangular matrix. On the diagonal of are the pivots. stands for lower triangular matrix. Notice that when we switched sides, was put on the left-hand side of . This was because from the left-hand side, we perform row operations, which is what we are looking for here.

This equation can be further divided into ,

where only contains the pivot values, and is the result of dividing the first row by 2 and the second row by 3. This is just how this works. The rows of are divided by their corresponding pivot values.

For a matrix, , and . Notice the order in which the inverse elimination matrices were written. It is the reverse of the order in which the elimination matrices appeared, since we must undo the latest operation first and the first operation last. The order is very important.

Say we perform and then . The combination of this is . Notice that we have a in the position that isn’t immediately obvious. This is the effect of the first row on the third one. In , when we said row 2 row 3, we did not actually take row 2. We took the edited version of row 2, the result of , in which we performed row 1 row 2 on the actual row 2. Thus, actually said row 1 row 2 row 3. In simpler terms, if we consider the original matrix to be , then is the series of transformations.

So, the elimination matrix is a little ugly. The inverse matrix on the other hand is quite pretty. For the inversion . We simply took the columns corresponding to each inversion. If we know the original elimination matrices, we can find the overall inversion matrix easily, without even having to think about it. This only works if there are no row exchanges however.

## Actually Using Factorization

Consider the set of linear equations:

If we write this in the form ,

We want to transform the matrix into . In order to do this, we must find , and we already know that .

Performing Gaussian Elimination on will give us the following:

From here, we can easily see and that .

Since we know , we can now replace . Let where . The resulting equation, , easily gives us the values of :

, and

Going back further, since ,

, and .

## Why You Would Want to Use This Seemingly Long Process (Time Complexity)

Consider a matrix. To perform a single elimination, we must perform a multiplication on 100 elements (the top row) and 100 subtractions (row 2 – row 1). To get 0s on the first members of every row (except the first), we would need to do operations. We now have nearly 1 column of 0s. The next column of 0s will need roughly operations. Thus, we have . This gives us . Thus, elimination has a complexity O(n3). We also have an extra column though, . That adds a cost of n2 (remember how the first column worked).

In many systems however, does not change, only does. If we wanted to find the result due to that single column change purely through Gaussian Elimination, we would have to go over the entire process again. However, if we were to use factorization, it would be much faster. Since is fixed, we need to find and only one time. This would leave only the effect of the change in , which is O(n2), making the system much faster. (I know it still seems vague but properly studying factorization will make it clearer.)

## Permutation and Transposition

For a matrix, there are possible permutation matrices.

For an matrix, there are Permutation matrices.

The inverses of the matrices , and , are the matrices themselves. Thus, . However, the inverse of is and vice versa. The actual rule, for permutation matrices, is that the inverse of any permutation matrix is the transpose of that permutation matrix.

The transpose of a matrix is the matrix in which the rows are the columns of the original matrix, and the columns are the rows. Thus, for a matrix , .

Notice that for the matrices , and , they are their own transposes. is the transpose for and vice versa. Transposes being the inverse is a property unique to permutation matrices.

Although permutation matrices are not usually used when solving matrices, they are regularly used in mathematical programs. For example, in MATLAB, when we try to find the factorization for a matrix, if the program finds a pivot that is close to , it tries to change it. This is because values close to might become in the elimination process. The change is done through permutations. So, instead of , we actually get .

## Symmetric Matrices

When , i.e. the transpose of a matrix is the same as the original matrix, we say that the matrix is symmetric. For example, is a symmetric matrix.

For any matrix , if we multiply the matrix with its transpose, , we get a symmetric matrix. i.e. symmetric matrix. This can be proved in this way: