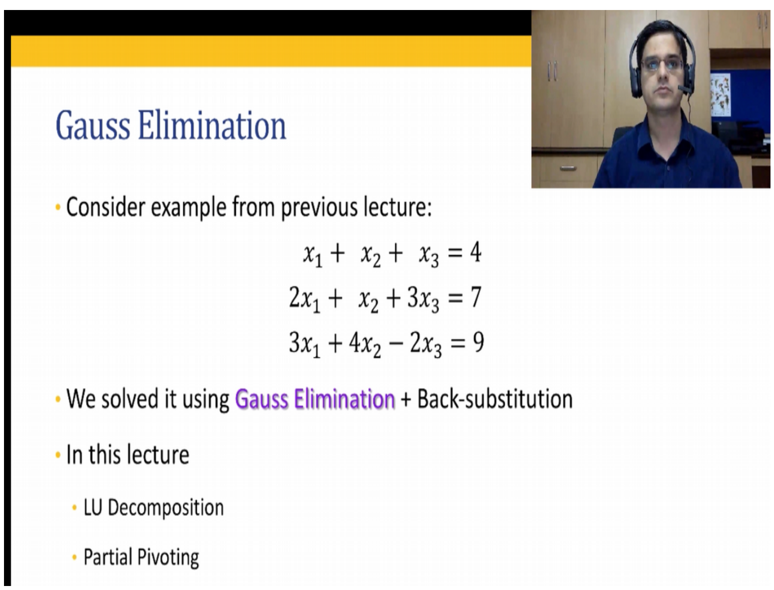
MATLAB Programming for Numerical Computation
Dr Niket Kaisare
Department of Chemical Engineering
Indian Institute of Technology, Madras

Module No. #04
Lecture No. #4.3
Linea Equations - LU decomposition; Partial Pivoting

Hello and welcome to MATLAB programming for numerical computations. We are in module 4, in this module, we are covering solving linear equations of the type ax = b where, a is an n /n matrix, x and b are n dimensional column vectors. In the previous lecture, what we had covered was a popular technique known as Gauss elimination followed by back substitution.

In this lecture, we will take the example, from the previous lecture, and in order to do two different things, one is to look at new method called LU decomposition and the second one is to look at partial pivoting in gauss elimination.

(Refer Slide Time: 00:52)



So, let us consider, the example from previous lecture, that was, a was 3 / 3 matrix as we all know x is a 3-dimensional vector containing x1, x2 and x3 where as b is 4, 7 and 9.

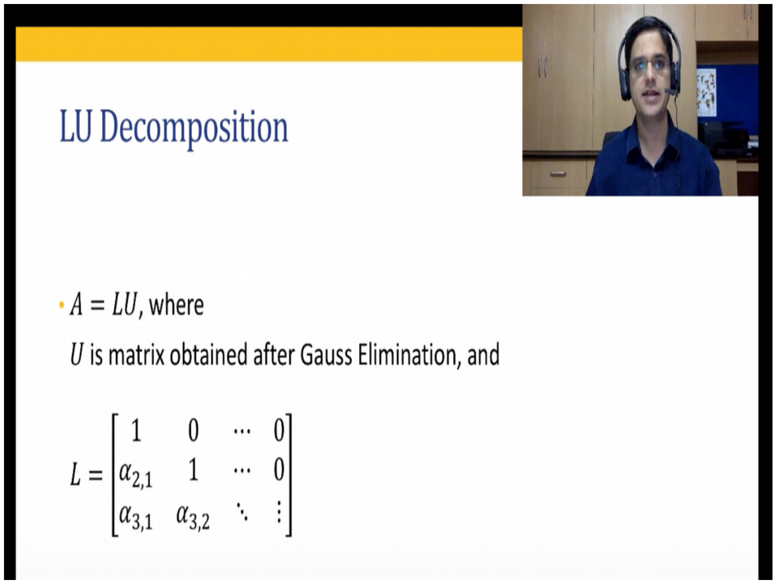
We solve this problem using gauss elimination and back substitution, and what we are going to do is, take the same code, from that we generated using MATLAB in the previous lecture, and use it to do two things, LU decomposition and partial pivoting. (Video Starts: 01:26) So, let us go to MATLAB, and edit gauss elm. So, this was the code that we wrote in our previous lecture. So, we will kind of, just modify the code little bit.

So, what we have done is, with A (1, 1) as the pivot element, we just enumerated for row 2 and as well as for row 3. So, let us just modify that and put in a, for loops for i=2 to 3, because we did this for row 2 over here. And we did this for row 3 over here. So, when i=2 okay, everything that we have 2, we can just replace it with i and see what we get. And let us see that we will replace Ab (i). Ab (i) and this is Ab1.

So, we do not have to change that, and we will just have to delete this, type end, and that should be good for us, and just indented as appropriate okay. And for this particular case, again what we are going to do is, something like this, we will just say i=3 because for i=3, 2, 3 in this particular case. And alpha is going to be Ab(i,2) just as above divided by Ab(2,2) and Ab (i, :) = Ab (i, :) α * Ab(2, :). So, you have done that. And now Let us run this and check if there are any errors.

I will click on run and I will go to the MATLAB command prompt, and type our x and also type Ab matrix, yes Ab matrix is indeed a lower triangular matrix and x, the solution is indeed what we expect 1 2 1 okay. So we are happy with this change. (Video Ends: 03:28). Let us go back to power point okay. So, what we did now is, we took the gauss elimination example from the previous lecture. And modified it so that we use basically loops in order to do the gauss elimination computation okay.

(Refer Slide Time: 03:47)



Let us go on to the next slide LU decomposition. So, what happens in an LU decomposition is as follows, interested readers can go on and look at computational techniques course. It is the sister course of this particular course, you can look at that course, module number 3, lecture 3 and there I have explained what LU decomposition is.

And basically, the idea behind, LU decomposition is, that any matrix A can be written as a product of the lower triangular matrix L and an upper triangular matrix U where, the U matrix is nothing but, the matrix that we obtained after gauss elimination. (Video Starts: 04:29) What that means is, our U matrix is nothing but, this guy 1, 1, 1; 0, -1, 1; 0, 0, -4. That is our U matrix. (Video Ends: 04:41) What is our L matrix? Our L matrix is nothing but, 1, 1, 1 as the diagonal elements and the alpha are those ratios of a A(2,1) / A(1,1) , A(3,1) / A(1,1) , A(3,2) / A(2,2) etc those ratios which we have already pre computed in gauss elimination form the l matrix okay.

So what do we do? So, what do we need to do is, U matrix is already obtained a gauss elimination. So, we do not really have to do anything about it. What we need to do is, to define that L matrix first as an identity matrix and afterwards as we do computations of gauss elimination, just populate the element alpha (2,1), then populate element alpha (3,1) and alpha (3,2) and so on and so forth.

As we do the computations, (Video Starts: 05:37) let us go on to MATLAB and what we will do is gauss elm. We will save as myLUcode okay. I will just change this, LU decomposition using naive gauss elimination. Everything remains the same, what we need is, the L matrix and the U matrix, we do not need back substitution. So, let us go, and delete our back substitution okay. Now that, we have deleted our back substitution. Our U matrix is nothing but, Ab from 1 to n, 1 to n. We do not want the b part of it. So, that is the reason, why we are writing U in this particular manner. Now the question is how to obtain L?

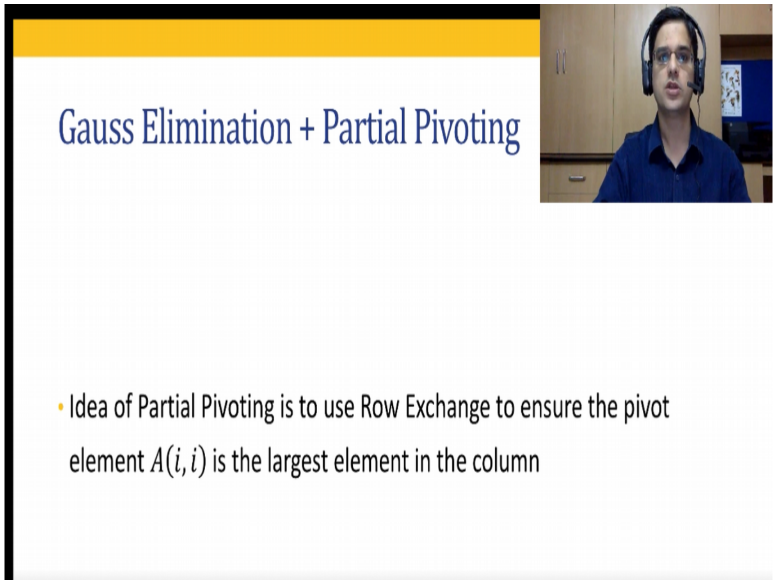
As I had said earlier, what we do with L is, initialize that, as an identity matrix eye n okay. (Video Ends: 06:41) So, when we initialize, it as an identity matrix, we are going to get once as the diagonal elements. The sub diagonal elements right now are all 0s and as we do the computation, we are going to populate that particular matrix as well okay. (Video Starts: 06:58) So, what we need to do, this alpha is, nothing but the first column row 2 and row 3.

So, L row i, that is row 2 and row 3 and the column number 1 is going to be nothing but the alpha okay. And we write that, and that should be good enough for us, and the same thing we are going to write for, our column number 2 as well. So we are going to write this and this is not going to be column number 1 but this is going to be column number 2. So, row number 3, column number 2, again is this alpha that was computed over here.

Keep in mind, that I am not writing this L before the alpha. I am writing this L after the alpha and myLUcode and I will press enter. Hopefully we will not get an error. No, we did not, so we look at our L matrix and we look at our U matrix. So, this is our L and U matrix. And let us, take the product L * U and let us see what do we get L and when we say a L * U, we will get exact same a matrix back. So, that is what LU decomposition is. (Video Ends: 08:11)

(Video Starts: 08:13) And we go back once again to the code, and see really what changes have we really made. The only changes, that we have made is, this particular command L (i,1) has been added in this overall computation and we initialized our L as identity matrix and finally we compute our U in this particular. So, those are the only changes that we have to make you save this and we quit. (Video Ends: 08:44)

(Refer Slide Time: 08:46)



So, the idea of partial pivoting is as follows, so what we are go doing in gauss elimination is, we use row exchange and row operations. In order to do to get 0s, in the sub diagonal elements, in the elements below the diagonal okay. In the naive gauss elimination, we just blindly use whatever so, overall results that we have, we just keep blindly using that the matrix A. We blindly use that, in case of gauss elimination with partial pivoting.

What we are going to do is, we are going to use row exchange, in order to exchange it, in such a way. Exchange the rows in such a way, that the pivot element is the largest value, the absolute value of pivot element is the largest value, in the pivot column. So in the first operation, when A(1,1) is the pivot element at that point, we look at all the coefficients in the first column okay. And the largest coefficient is going to be swapped with row 1.

In this case, the large coefficient happens to be 3. So, we will swap these 2 equations, and then do the first step of gauss elimination okay. (Video Starts: 10:03) So, let us go back to MATLAB, and open our gauss elimination code okay. And I will save this as gauss elimination with partial pivoting, using gauss elimination, okay. So what I need to do is now A(1, 1) is the pivot element change, to ensure A(1, 1) is the largest in column 1 okay.

So, what that means, is that, we want to look at the column 1. So, col 1=Ab, the entire row, all the rows and in the first column. So, that is the column 1, we want to find which one is the largest, and to do that, we use the command called max. How do we exchange row 1 and 3, and we remember Ab (3, :) is nothing but the third row right. So if we were to write Ab (3, :) and assign it to Ab (1, :) okay.

That is going to actually, give us the third row and it will put the third row instead of the first row okay. There is going to be a problem with this. I will show you, what that problem is by clicking enter okay. Can you can see what the problem was, compared with our Ab matrix? Can you actually, see what the problem was? Okay.

The problem is this, by doing this what is happening is, we have now lost our first row okay. We now no longer know what our Ab (1, :) was. So, we need a way to store that Ab (1, :) and then use this particular command. So let us just copy this and go onto our editor and do this again. Before, we do that, we need the max also. And we want to store Ab (1, :) before, we change the value of (1, :). So let us just reuse that variable dummy because anyway that variable was a dummy variable.

So dummy is nothing but Ab (1, :). It is a temporary place holder for the first row. And Ab (3, :) is nothing but dummy okay. So, let us save this, and what we can do is, this is okay. And I will just right click and click on evaluate selection, and see what we get. If we type Ab, we have seen that row number 3, has now been exchanged with what earlier was row number 1, so row exchange has happened. So, this is how partial pivoting will work. So, this is partial pivoting.

What we have done over here is, absolutely what partial pivoting is. We did that with the first row when A (1, 1) was the pivot element okay. Computation in the pivot column okay. And we repeat this, row exchange to, okay. And again we need to look at only the sub diagonal elements, so col2 is going to be=Ab ((2 to end), 2) okay.

And as before, just copy this and dummy, idx okay, is this, and dummy=Ab (2, :). Ab (2, :) =Ab (idx) and Ab (idx) =dummy, and the same thing we can do over here as well. Ab (1, :) is going to be dummy, okay. Ab (1, :) is nothing but Ab (idx, :) and Ab (idx, :) is going to be nothing but dummy okay. So we have this and we continue with our gauss elimination, so okay. So save this and let us hope there are no errors.

We have run this and go back and check. Our Ab, is now it is a upper triangular matrix again over here okay. But we have done this partial pivoting, which has resulted in exchange of row number 1 and 2 in the first case okay. And let us see, what we get as the solution x, after back substitution. And after back substitution, we notice that our x is exactly the same as before 1, 2, 1. So yes, it does work and we want to find out.

Let us say if the idx is over here has changed or not. So I can just type, I think there was a mistake over here. I think this should be col2. I do not, I am not sure why, this actually run, probably I did because, let us see how, what Ab is different this time okay. As you can see Ab is different but x you are going to get as the correct solution 1, 2, 1, and you will see that in this particular case. In this case of partial pivoting, in this particular case, a row exchange between 2 and 3 had happened in the second, for the second pivot element.

In this case, the particular row exchange had not happened. So, that is the only difference between these 2 examples okay. And we do get our x solution as 1 2 1 using partial pivoting okay. So, that pretty much brings me to the end of this particular lecture okay. (Video Ends: 17:41)

What we have covered in this lecture, two concepts, one is that of LU decomposition where the matrix A can be written as a product of lower and an upper triangular matrix. The upper triangular matrix was nothing but the solution obtained after the gauss elimination, and the lower triangular matrix have been the matrix containing the alphas okay.

And the idea behind gauss elimination plus partial pivoting, was to use row exchanges. In order to ensure that the A (i, i) is the largest element in that particular column, when we are comparing with sub diagonal elements only okay. So, with that we come to end, of this particular lecture. Thank you for listening and see you in the next lecture.