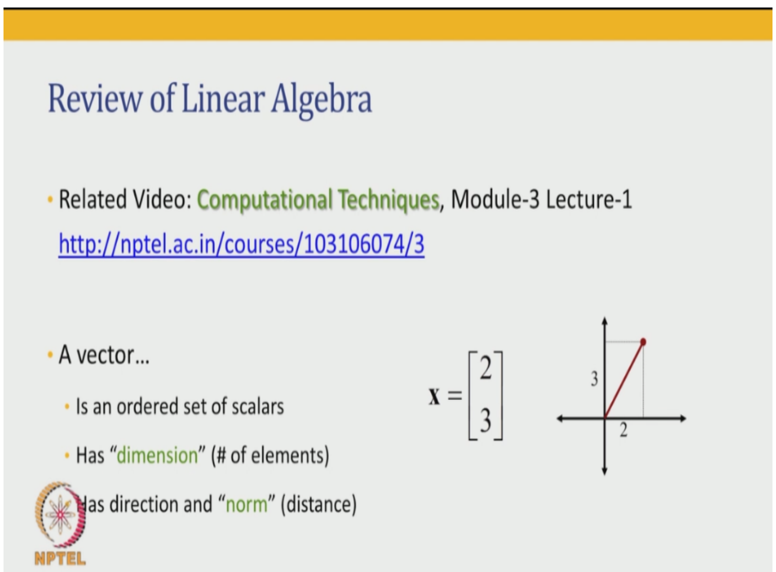
**MATLAB Programming for Numerical Computation**
**Dr. Niket Kaisare**
**Department of Chemical Engineering**
**Indian Institute of Technology, Madras**

**Module No. #04**
**Lecture No. #4.1**
**Linear Equations – Linear Algebra in MATLAB**

Hello and welcome to MATLAB programming for numerical computations. We are in now in week 4 of this course. In this week we are going to cover linear equation. So basically we are going to cover linear algebra in the first lecture 4.1 followed by several methods to use linear algebra to solve the problems of the nature ax equal to b to find the values of the vector x. Before we do that let us recap some of the results from linear algebra

(Refer Slide Time: 00:41)




For this the related videos are in the computational techniques course module 3 lecture 1.The link for which is given in front of you. So Let us review some of the results in linear algebra. So what is a vector is nothing but an ordered set of scalars an example is shown over here. So this vector has a dimension in this case the dimension of this vector is 2 as convenience what we are going to do in this entire lecture series.

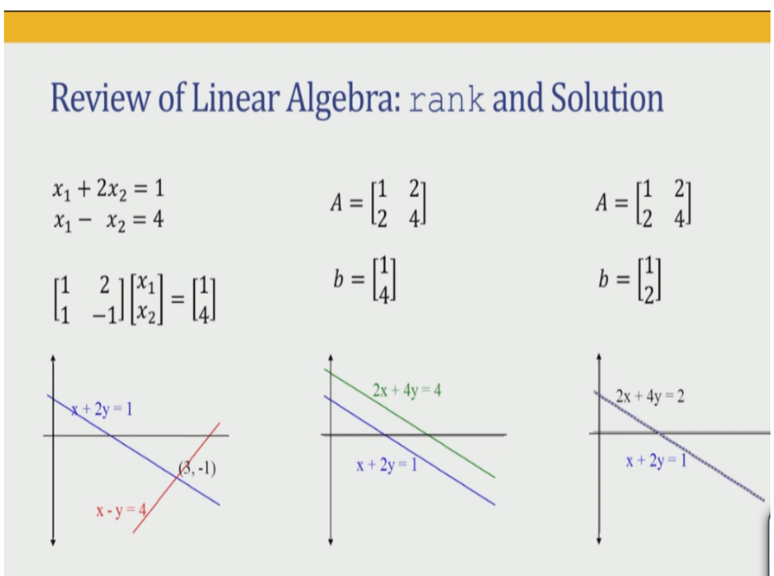
We are going to assume all the vectors to be column vectors which means that the vectors will have n rows but a single column. As you can see over here we have 2 rows but a single column now this column vector has a dimension which is nothing but the number of elements. If we want to look at the geometric interpretation.

This could be nothing but a point in this space in a 2 dimensional space which is the distance 2 on the x axis and the distance 3 on the y axis or more appropriately distance 2 on axis on the first axis and distance 3 on the second axis. Now this particular vector has a direction which is basically given by the value 2, 3 as well as it has a norm.

Now the most common norm that we are used to so far is what is known as the 2 norm of this vector which is nothing but the square of the distance of that particular point from origin. So square root of 2 square + 3 squared is nothing but the 2 norm of this particular vector. Likewise we have the 1 norm of vector and n norm of the vector or the infinitive norm of the vector something we are not going to go into.

But essentially just the way we are associate a distance with a physical point in space a vector can be associated with a norm which is nothing but an indication of the size or the distance of that particular vector.

(Refer Slide Time: 02:52)




Okay, now in next case we have seen is what does it mean when we want to solve 2 equations into unknown. So here I am showing 3 different cases of 2 equations and 2 unknowns. The unknowns are x1 and x2 and the 2 equations can actually have a single solution can have 0 solution and can have infinite number of solution. So when do we get a single solution, 0 solutions and infinite number of solutions we are showing graphically on this slide okay.

So if you look at the first example, we have the equation x1+ 2 x2 = 1, x1- x2 = 4 these represents the blue and the red line over here and these 2 lines intersect at a point (3-1). So (3-1) is a unique solution okay. Next is if these 2 equations instead of having x- x1- x 2 as the second equation, if we were to have 2x1+ 4x2 which is nothing but the double of the left hand side, as the left hand side of the next equation which is 2x1+ 4x2 that is the left hand side of the next equation.

And the right hand side if we take that value as 4, we can see that these now represents 2 parallel lines. These 2 parallel lines are not going to intersect and we are going to have 0 solutions in this case. And now in this case if the second equation was nothing but double of the first equation so 2x1+ 4 x2=2, which is nothing but double of this first equation.

We are going to have these 2 lines lying on top of each other and we are going to have infinite number of solutions. So the question is when do we get single solution? When we do get infinite number of solutions? And when we do get 0 solutions? Okay and answer to that is from finding out the rank of the matrix.

So if we look at the rank of matrix a and if the rank of matrix in this particular case was equal to 2, we are going to have a unique solution if rank of a square matrix of an n by n dimensional square matrix is equal to n, then we will have a unique solution for the problem ax=b okay.

But what if the rank is not equal to 2? That is the case that we see in the middle column as well to the right column. In that case we need to look at the rank of the matrix a, b if rank of matrix a, b is equal to the rank of a, then we are going to have infinite number of solutions but if rank of a, b is not going to be equal to rank of a, then we are going to have 0 solutions okay.

(Video Starts: 05:45) Let us go on over to MATLAB and look at these 3 examples that we have seen so let us look at the matrix a, matrix a is 1 2; 1 -1. b is the right hand side that is 1, 4 okay. So these are the 2 values that we have, when we want to solve ax = b. We know that the solution x is going to be inv, inverse of a*b okay. And x as you can see over here is 3, -1 something that we had seen in our power point slide a minute earlier okay.

Now MATLAB has a more efficient way of solving these equations and that is using slash in this case backslash. Because we are pre multiplying inverse of a with b, so we write this as a backslash b and when we press enter we are going to get the solution 3, -1.  A backslash b for and a which has a rank = n is exactly same as this particular command okay.

Let us do help slash c and see what we get okay. If you recall, what we did in the first module we had looked at element by element division and element by element operations and this is what the later part of this help slash showed us. But what we are more interested in is to look at what the left division and right division actually mean.

So a backslash b is left division which is same as inv a*b, if a is an n by n matrix with rank = m okay. It is something different if the mat rank of a is not equal to n we will not go into that right now. And then we have an equivalent called a right division right division is b multiplied by inverse of a okay.

For the problem ax equal to b, we needed to have a left inverse. So x was inverse of a multiplied by b not b multiplied by inverse of a, and therefore we had left inverse which is indicated by backslash, a backslash small b over here not capital b and that is what the solution that we got okay. So that was the first thing that we saw how to solve these equations.

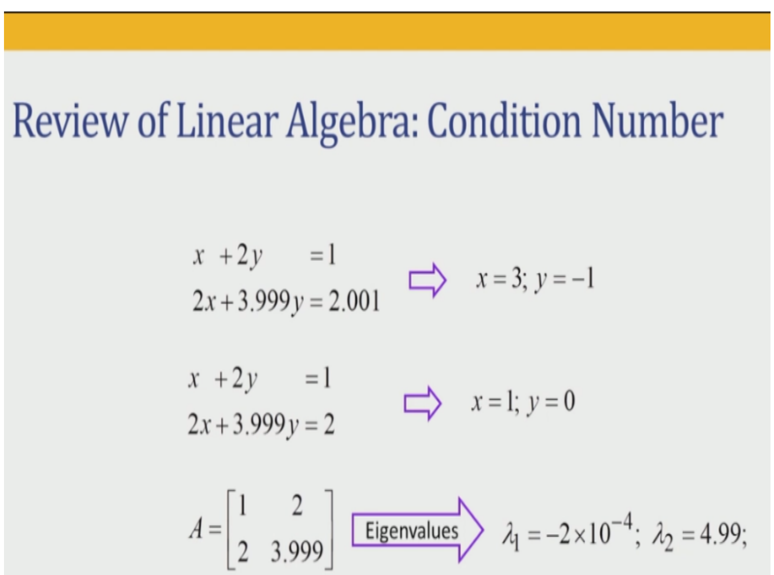
Let us look at how to find out using MATLAB whether or not your equations are going to have a unique solutions or infinite number of solutions or 0 solutions. And we can do that by using rank a and rank of matrix a is equal to 2, so a we say is a full rank matrix. That means a, the rank of a is equal to the number of rows or the number of columns of matrix a, okay.

We can also find the size of matrix a by saying m, n = size a. m is going to be number of rows, n is going to be number of columns we have already seen that in module 1. (Video Ends: 09:25) Let us now look at the second example and the second example, our b remain the same we changed our a to 1 2 2 4. (Video Starts: 09:33) Let us make that change a equal to 1, 2; 2, 4. Let me echo this by skipping the semicolon okay, so our a is 1 2 2 4. Let us get rank of a and rank of a = 1 okay.

And therefore a will not have a unique solution sorry, ax= b we will not have a unique solution. It can either have infinite number of solution or 0 solutions. And to do that we need to find the rank of matrix a, b. And the rank of matrix a, b = 2 which means that the rank of a, b is not the same as the rank of a. (Video Ends: 10:19)

So we are going to have 0 solutions that is this particular case. Now Let us change b to 1 2, (Video Starts: 10:26) b = 1; 2 note that we have used that semicolon because we want the column vector. So that our b. Rank of a is not going to be change because we have not changed rank sorry we have not change the matrix a. Let us now look at rank of a. And rank of a, b in this case equals rank of a in which case (Video Ends: 10:52)

(Refer Slide Time: 10:53)



We are going to have infinite number of solutions okay. Let us now look at the next concept in linear algebra and that concept is what is known as condition number. In these 2 examples we had equation x1+ 2x2 = 1 and x1+ 4x2 = 2. Instead of the second equation being in x1+ 4x2, if we have to change this to 3.999 x2 okay. We are going to this particular equation okay and our matrix a is going to be 1 2; 2 3.999

If you solve this equation we are going to get the solution as 3, -1. Now if we make a very small change in our b that means instead of 1 and 2.001 if we change it to 1 and 2.0, we are going to have a very large change in the solution. And the reason for this is what is known as condition number or what is known as a poorly condition matrix. Condition number is nothing but the ratio of Eigen values of the matrix.

So in this particular case a being n / n matrix has 2 Eigen values and the ratio of 2 Eigen values in this case is approximately 25000 which is a fairly large number and therefore we see that the results change quite significantly for a reasonably small change in b matrix okay. (Video Starts: 12:28) So Let us look at the condition number of a when a was defined as 1 2; 2 3.999 that was our a matrix.

We will use the command cond in order to find the condition number of a and cond a is 2.5 was approximately 2.5*10 ^ 4 this is approximately 25000. Now we can find Eigen values of a using the command eig and eig will give you 2 Eigen values and we can also verify that the condition number is the ratio of the 2 Eigen values okay. And the absolute value of that. (Video Ends: 13:30).

Yeah and this is exactly what our condition number is that we obtained using the command cond. So the MATLAB command cond is finding condition number of any matrix. The MATLAB command rank is to find out the rank of that particular matrix and MATLAB command eig is the command for finding Eigen values of that matrix.

(Video Starts: 13:45) Now Let us go back and look at what else our Eigen values command can do. So I will type help eig okay. And so we can see that Eigen values. So if we give a command eig with just 1 output argument, we are going to get a column vector containing Eigen values. However if we give 2 arguments v and d, v is going to be of a matrix containing the Eigen vectors and d is going to be matrix whose diagonal elements are the Eigen values.

So Let us look at that in MATLAB v, d equal to eig okay. So as remember just a few minutes back we had said that will define vectors as column vectors. So this is the Eigen vector 1 which is -.89, 0.45 that is the first Eigen vector corresponding to this Eigen value. The second eigenvector is this guy which corresponds to this Eigen value.

Okay Let us look at whether this satisfies the equation rather a multiplied by v = lambda multiplied by v whether this is satisfied or not okay. Let us try to look at that okay. So Let us say l was nothing but d1 okay. And our v was nothing but the first column. The first column basically means all the rows and column number 1. This l is the first Eigen value, this v is the first Eigen vector. Let us say a*v- lambda * v what we get is, we get the value as 10 to the power- 15.

Look, Let us look at the value of the a * v and a * v is of the order of 10 ^ - 3. So for all practical purposes this particular guy a * lambda is up almost equal to lambda * v okay. Again keep in mind what he had said earlier in the previous 2 modules is, computer is a finite precision machine. (Video Ends: 16:17)

What that means is that computer has a least count. So when we are going to do problems using floating point numbers using real numbers, we are not always going to get when we do (a-b). We are not always going to get the value equal to 0 if a and b are equal. We are going to get some times value of (a-b) as a very small number compared to value of both a and b.

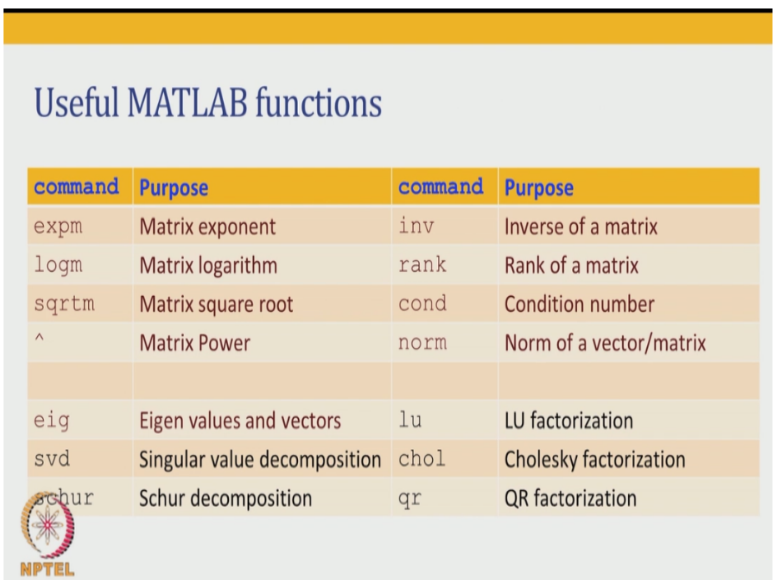
(Video Starts: 16:44) And that is what essentially we are looking for is whether or not a multiple by v is close enough to a lambda * v. So a multiple by v is this value and lambda * v is also the same value. Let us look at the other example where, we take the lambda as this second Eigen value.

So l is the d2, 2 and v is nothing but v all the rows and second columns okay. Let us look at a multiplied by v, this is the value of a*v. And let us look at lambda multiplied by v and that is the same value that we get as lambda * v okay. To recap what we did. We did use the command eig to find Eigen values and Eigen vectors, when you run the command with 2 arguments. (Video Ends: 17:37)

The first argument will give you a matrix containing the Eigen vectors. The first column of that matrix is the first Eigen vector, the second column of that matrix a second Eigen vector and so on. The second argument from eig is a matrix whose diagonal elements are the various Eigen values.

The first a11 is the first Eigen value, a22 is the second Eigen value and so on to ann is the nth Eigen value.

(Refer Slide Time: 18:18)


Useful MATLAB functions

| command | Purpose | command | Purpose |
|---|---|---|---|
| expm | Matrix exponent | inv | Inverse of a matrix |
| logm | Matrix logarithm | rank | Rank of a matrix |
| sqrtm | Matrix square root | cond | Condition number |
| ^ | Matrix Power | norm | Norm of a vector/matrix |
| | | | |
| eig | Eigen values and vectors | lu | LU factorization |
| svd | Singular value decomposition | chol | Cholesky factorization |
| schur | Schur decomposition | qr | QR factorization |



So that is the review of the linear algebra that is going to be useful for rest of this course and let us now recap all the useful MATLAB functions that we have done so far. In the previous lectures in module 1 primarily. We have looked at matrix exponent, matrix logarithm, matrix square root, matrix power and today we also looked at matrix division, left division as well as right division. So these were the commands that we have covered before in like in modules before.

Today we covered the command inv to find the inverse of a matrix, rank to find rank of matrix, cond to find condition number of matrix. We will also in a minute use the command norm to find norm of a vector or a matrix. We will just take ourselves to vectors for now but the norm exists for a matrix as well okay.

We also used eig to find Eigen values and Eigen vectors, lu cholesky qr factorization, schur decomposition these are something that a lot of engineers use quite a bit and I have put them in this particular table just for the sake of completeness. Lud composition is something that we are going to cover in the fourth lecture of this module okay.

However you do not have to worry about the remaining thing singular value decomposition or schur decomposition or cholesky factorization and so on for the purpose of this course okay. Okay so Let us finally, now let us go and find out the norm. (Video Starts: 19:49) So Let us say, Let us look at our vector x. Now norm of this vector is nothing but $3^2 + 1^2$ square root of that, which square of n.

So norm of x is going to be square root of 10 which is nothing but 3.1623 okay. So norm is an indicator of the size of any vector. (Video Ends: 20:16) So with that we come to end of lecture 1 of module 4. So what we have done in lecture 1 of this particular module is, to cover the basics of linear algebra.

So primarily we have looked at trying to solve equations of the form ax = b, look at Eigen values and Eigen vectors of a what does the condition number mean, what does rank means and how we can use rank in order to find out whether we will get unique solutions, infinite number solutions or 0 solutions to an equation of the form ax =b so that ends this lecture for recapping linear algebra. And see you in the next lecture thank you.