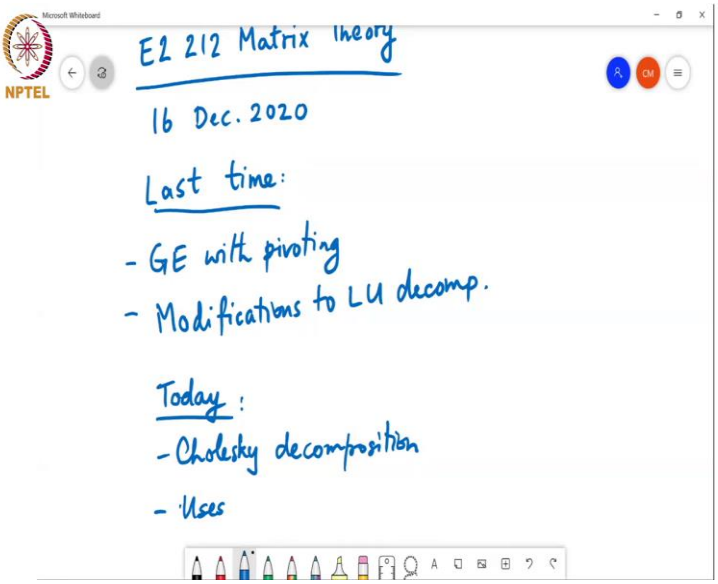
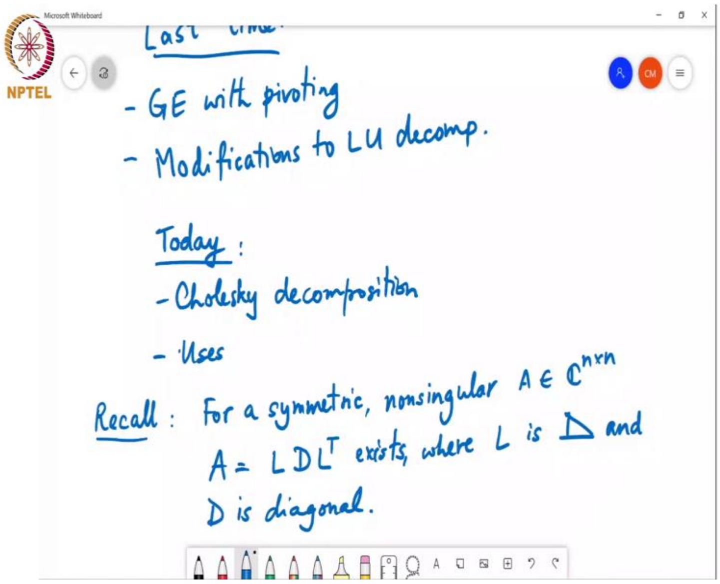
Matrix Theory
Professor Chandra R. Murthy
Department of Electrical Communication Engineering
Indian Institute of Science, Bangalore
Lecture 56
Cholesky decomposition and uses

(Refer Slide Time: 00:14)



So, good afternoon. Let us begin. So, last time we were looking at Gaussian Elimination with pivoting, we describe the process, which involves using these rotation matrices. So, today we will, and towards the end of the class, we started looking at other modifications to the LU decomposition. And we sort of working our way up to this decomposition we will discuss today which is known as Cholesky decomposition. And we will also discuss some uses of this Cholesky decomposition.

(Refer Slide Time: 00:48)

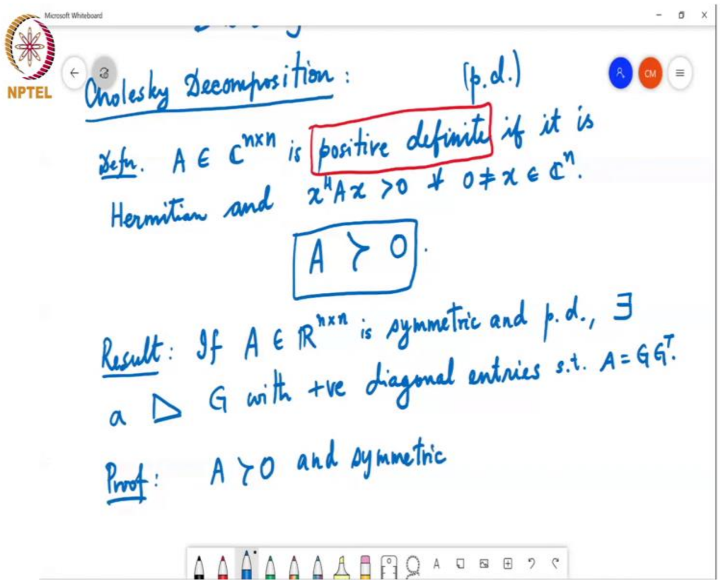


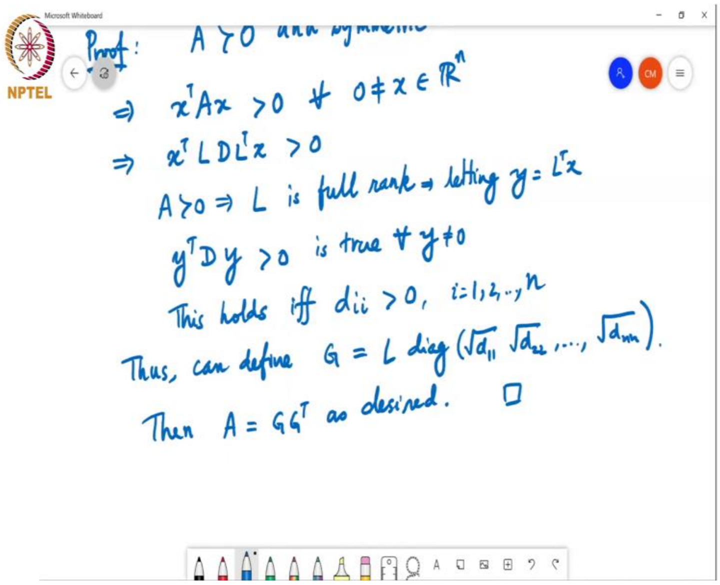
So, just to recall. In the last class, we saw that if matrix A, if matrix A is symmetric and non-singular, then this, this in the, in the decomposition, LDM transpose L is equal to M which means that we can write, we can find a lower triangular matrix L such that A is equal to LDL transpose exists, where L is lower triangular and D is diagonal. So, this kind of decomposition exists.

Now, do not confuse that the entries of D are the eigenvalues of the matrix, they are not, in general, they are not, D are some diagonal entries and this is the LDL transpose decomposition. One thing is that the eigenvalue decomposition of a matrix is not something that can be done in using a polynomial type, time procedure like we are describing here. Whereas, these kinds of LU decomposition or LDM transpose decomposition or LDL transpose decomposition these are you can write down a polynomial-time algorithm that can find this decomposition.

Whereas, to find the eigenvalue decomposition, you need to find eigenvalues and eigenvectors for which you need to typically use iterative procedures or some other techniques to find the eigenvalues and eigenvectors. So, the D is a diagonal matrix, but those D, the entries of D are not the eigenvalues of the matrix. So, now we will discuss about Cholesky decomposition.

(Refer Slide Time: 03:26)





So, I need one definition, which we are going to use very heavily in the coming classes and that is that of a positive definite matrix. So, A in C to the n cross n is positive definite, if it is Hermitian and x Hermitian Ax is strictly greater than 0 for all non-zero vectors in C to the n. And we will write this as A slanted greater than 0.

So, another way to define a positive definite matrix is a matrix is positive definite if it is Hermitian and all its eigenvalues are strictly greater than 0, but we will come to that. For the purposes of this discussion, this definition is enough. So, this is that of a positive definite matrix. So, now the first result we have is that if A is symmetric and positive definite.

Actually, this is for the real case, and positive definite, which I am going to abbreviate p.d. then there exists lower triangular matrix G with positive diagonal entries. Such that, A is equal to GG transpose. So, instead of an LDL transpose decomposition, we have a GG transpose decomposition. So, why is this true? It is because, so first of all, A is positive definite, and symmetric. So, that means that x transpose Ax is strictly positive for every non-zero x in R to the n.

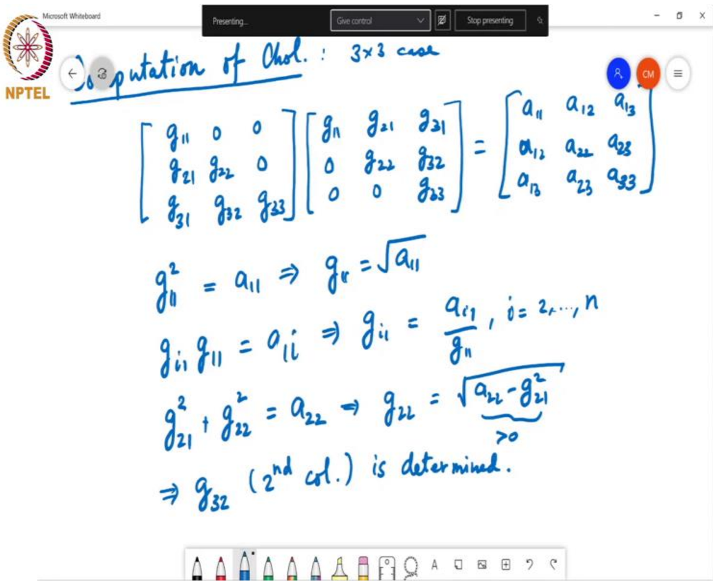
Which, since A is positive definite and x transpose x is strictly greater than 0 for all non-zero x, it means that the matrix X is non-singular. So, it admits a symmetric and non-singular. So, by the previous result, it admits an LDL transpose decomposition and so that means that x transpose LDL transpose x is strictly greater than 0. And since, A is positive definite, it means that the matrix L here is going to be non-singular rather full rank.

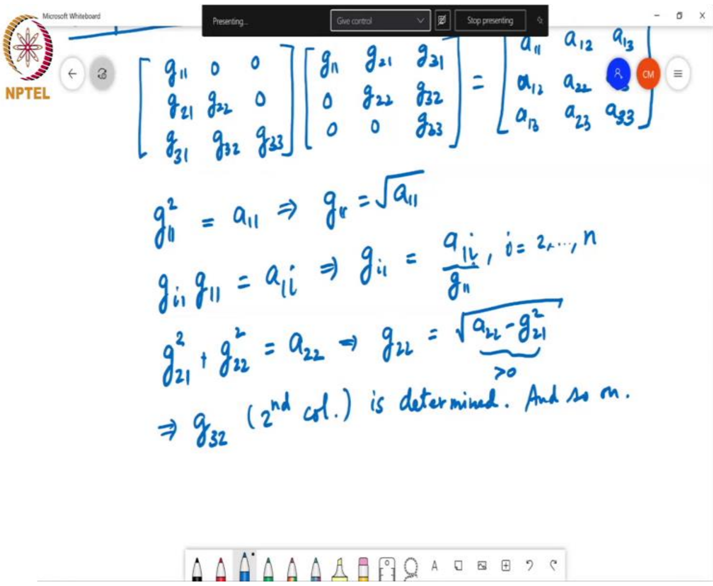
And so, if we let L is full rank which means that if we let y equal to L transpose x then what we have is Y transpose dy, just substituting, is greater than 0 and this is true for every y not equal to 0. And this, in turn, this holds if and only if dii, d is a diagonal matrix. So, each (())(09:00), you can, for example, choose y equal to A1 that will pull out d11 and that is greater than 0 then you can choose y equal to E2 that will pull out d22 and that is greater than 0. So, dii is greater than 0, for i equal to 1, 2, up to n.

And since dii is greater than 0, we can define G to be equal to L times this diagonal matrix containing square root of d11, square root of d22 up to square root of dnn. Then A equal to GG transpose and G is a lower triangular matrix with log with, with positive entries along the diagonal, this is as decided. So, one thing is that this Cholesky decomposition, where you are only looking for a G matrix such that equal to GG transpose.

Since A is positive definite and symmetric, it can be done with half the number of flops as compared to the vanilla LU decomposition. And the other thing is that because A is positive definite, it turns out that the computation of LU decomposition is stable without using pivoting. So, you do not need to do this pivoting step that we discussed in the previous class. So, here is how you compute the Cholesky decomposition, you can do it more efficiently than LU because of the following.

(Refer Slide Time: 10:53)





MATLAB has a built-in command CHOL of A and it will directly give you the Cholesky decomposition of A. So, we will illustrate this for the 3 cross 3 case. But you will see that the idea applies to any, any dimensional matrix. So, in the case of 3 cross 3 matrix, we are interested in, in solving for the matrix G, where it has the structure g11, g12, g22, g, sorry this g21, g31, g32, g33 and 0 is everywhere else.

This is G, G transpose will then be, g11, g21, g31, 0, g22, g21, sorry g0, g22, g32, and 0, 0, g33. This product should give me the matrix a11, a12, a13. So, then by following a proper order in trying to figure out these gij’s, you can actually compute each of them quite easily. So, for example, if you compare the 1 comma 1th element. So, the 1 comma 1th element here is g11 square that is equal to a11, and then taking the positive square root we have g11 equals square root of a11.

So, this, whether you take the positive or negative square root, gives you some flexibility in computing the Cholesky decomposition, but if you restrict yourself to taking the positive square root each time, then the computations are unique. So, the Cholesky decomposition is unique. And then what you do is you look at the first column of the product. So, that will be, the next entry will be g21 times g11. So, more generally, gi1 times g11 is equal to ai1, which implies that gi1 will be equal to ai1 divided by g11.

And essentially, the fact that the matrix A is positive definite implies that this a11 will be strictly positive because if you recall, a transpose Ax equal is greater than 0 for all non-zero x. So, if I take x to be equal to e1 that extracts the entry a11, and therefore a11 should be strictly greater than 0. So, I would not be dividing by 0 here. This is i equal to 2 to n. So, all elements in the first column of g are now already determined.

And similarly, if I look at the second column, the first entry will be this times this which will be g11, g21, but I already know what g11 and g22 are. So, that does not give me any new information. This matrix A is symmetric. So, actually these two will be the same. So, maybe just to avoid confusion, I will write this as, do the following, this is a12, this is a13, and this is a23. So, this times this gives me g11, g21 equals a12, which is the same as what I got over here g11, g21 equals a12, nothing new from that equation.

But if I take the 2 comma 2 times 3, I have g21 squared plus g22 squared equals a22, which is this entry here, which means that we already know what g21 is, sorry we already know, yeah, we already know what g21 is, gi1 we know i equal to 2 to n. And so, since we know this, we can then compute g22 to be equal to the square root of a22 minus g21 square. And the positive definiteness of a ensures that this quantity will be always positive.

This is something, these kinds of things are something we will see later. But it is a, it is indeed a consequence of the fact that the matrix A is positive definite. So, that ensures that this is always greater than 0. And so, since now, g22 is known, then the second column, which is essentially now g 32.

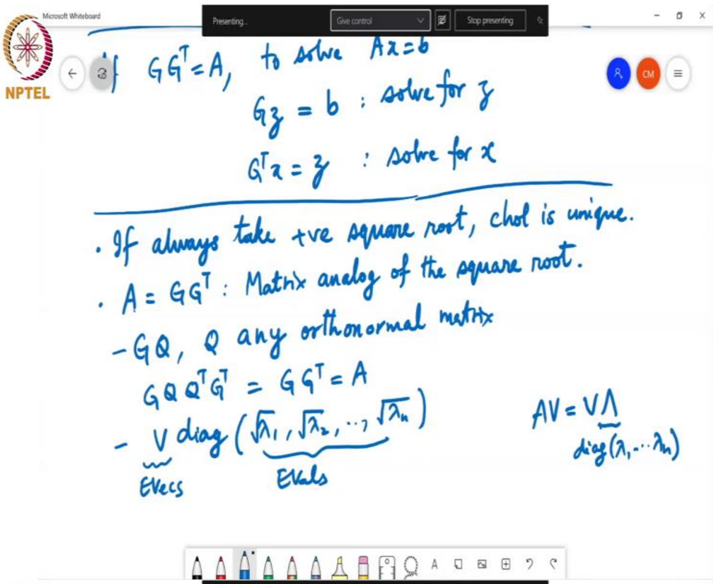
Student: Sir.

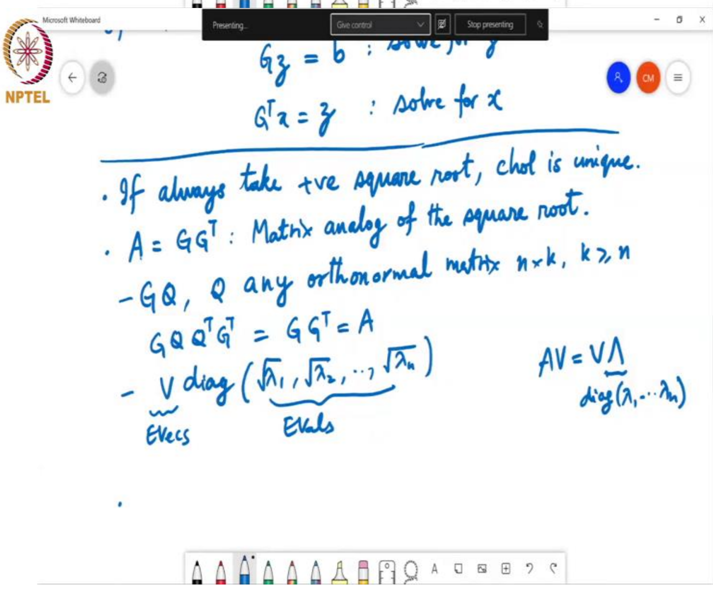
Professor Chandra R. Murthy: Yeah.

Student: Sir, as you change the entries, because of symmetry, you change the subscript things. The second line of your proof, like, it is not directly having any entry at A. You wrote ai1, so a21, a31 is not directly representing.

Professor Chandra R. Murthy: Okay, you can make that small correction, call this a1i., So, and so on. But you are right, this a21 between at the end. But the idea is that whatever entry I put here, you know it because I am just writing the entries out here. So, you can always determine these gi ones. So, basically this is how you can efficiently determine the Cholesky decomposition of a matrix A.

(Refer Slide Time: 17:58)





Now, if A equals GG transpose. Then, in order to solve Ax equals b, what we do is we first solve G times z equals b. So, basically G transpose x I am calling it z. And so, Gz equals P and you solve for z, and g is lower triangular. So, you can solve this efficiently using forward substitution. And then you solve G transpose x equal to z and solve for x. G transpose would be upper triangular, and so you can do backwards substitution and find x.

So, this Cholesky decomposition, like the Lu decomposition is useful to solve for, solve a system of linear equations. So, as I mentioned, if you take the positive square root at each step, by the way, we have developed it, it is clear that this Cholesky decomposition is unique. And yeah, So, this, so I will just make that note here. If always take the positive square root Cholesky decomposition is unique.

And the other thing is that since A equals GG transpose, this is like writing A as the product of something times its transpose. It is a, it is a matrix analogue of the square root operation. But the difference is that the square root of a matrix is not unique. So, in particular, if I take, if I consider G times Q, where Q is any orthonormal matrix. Then if I take GQ times GQ transpose, then that is also equal to GG transpose, which is equal to E. So, GQ is also a valid square root of this matrix K. So, it is not unique.

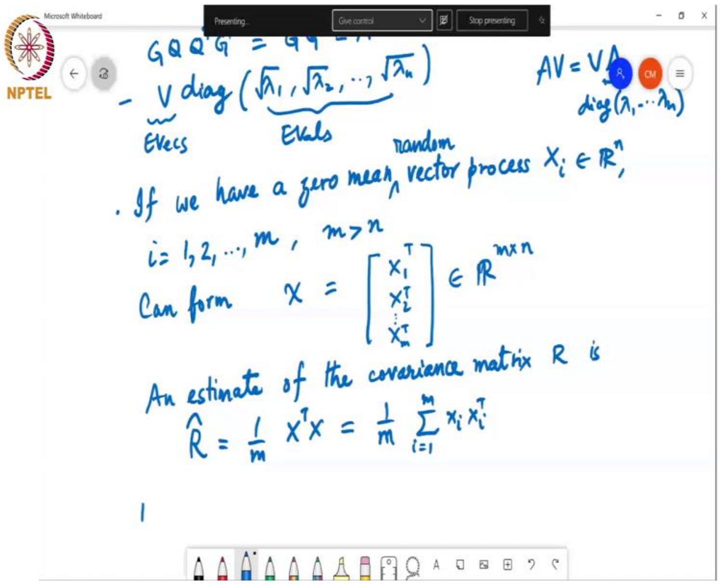
So, for example, another square root could be V times the diagonal matrix containing lambda 1, square root of lambda 1, square root of lambda 2, up to square root of lambda n. So, this also if I take these times its transpose it will give me V times lambda a diagonal matrix containing all the eigenvalues times V transpose and because so these are the eigenvectors and these are the eigenvalues.

Then remember that the eigenvalue decomposition is of the form AV equals V lambda, where this is a diagonal matrix containing lambda 1 through lambda n. And so, A is equal to V lambda V transpose. So, this is also a valid square root. So, there are many square roots that are possible. But the Cholesky decomposition is a unique square root in the sense that we have taken the positive square roots at each step and what you are getting is a matrix G, which is lower triangular and has positive diagonal entries.

So, if you restrict, to restrict to square root matrices that have these two properties, that it should be lower triangular and it should have positive diagonal entries, then the square root of the matrix is unique. Now, if we have a one small point here is that what I need here is that QQ transpose must be the identity matrix.

So, Q need not even be a square orthonormal matrix, it can be any matrix as that QQ transpose equals the identity matrix. So, it could be of size n cross k with k greater than or equal to n and still this will work. So, the square root of a matrix in this sense may not even be a square matrix because GQ will then be of size n cross k.

(Refer Slide Time: 23:47)

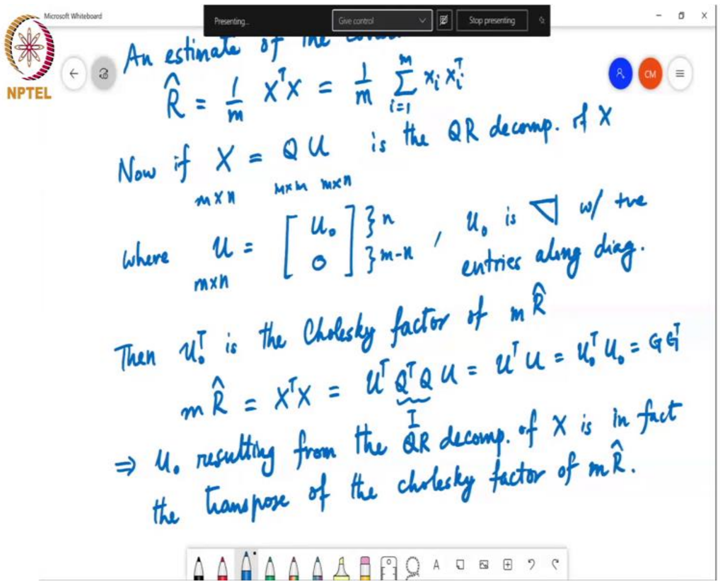


Now, another point is that, if we consider, if we have a zero mean vector process xi in R to the n. So, this is a random process. So, maybe I will write it as capital Xi. And suppose we have m samples of this random process and suppose that m is greater than n. So, you have more number of samples then you have, then the dimension of each of the vectors. Then we can form the matrix X which is a concatenation of all these like this X, X1 transpose, X2 transpose, Xm transpose.

And this is going to have m rows and n columns, then each row here is a sample Xi transpose, then an estimate of the covariance matrix R is equal to 1 over m times X transpose X. So, you might have seen this for example, in this form 1 over m sigma i equal to 1 to m X i X i transpose.

So, this is, this is a way to estimate the covariance of this may, this vector process x and this is, this has some properties that you will see possibly in your detection estimation class next term. But for now, just say that this is the estimate of the covariance of X. Now, if we have then call it R hat, the true covariance matrix R and this is the estimate obtained using m samples. So, we call it R hat.

(Refer Slide Time: 26:52)



Now, if X is equal to Q times u, where this is the QR decomposition. So, this is, remember that this is m by n. So, this matrix would be m by m and this u would be an upper triangular matrix of size m by n. And this is the QR decomposition x. I do not want to write R because I said the two covariances R. So, I write it as u. Where…

Student: Sir.

Professor Chandra R. Murthy: Yes.

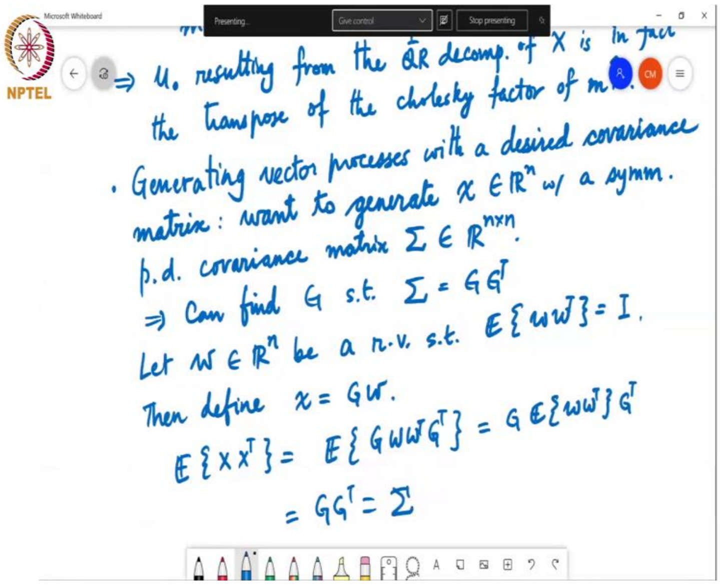
Student: Sir, if u is not squared metric, so how can we create like a triangular matrix, sir, upper triangular?

Professor Chandra R. Murthy: Upper triangular just means that there is nothing below the main diagonal, I mean, it is all zeros below the main diagonal. So, that is what I am writing here. So, this is of size m by n, it has more rows than columns, it is a tall matrix and this is the first n rows and this is the remaining m minus n rows. They will all be zeros. So, u0 is upper triangular with positive entries along the diagonal.

So, we will choose the QR to ensure that, then u0 transpose is, the Cholesky factor of mR hat that is mR hat is equal to X transpose X just from this itself and that is equal to u transpose Q transpose QU and Q transpose Q is the identity matrix. And so, this is equal to u transpose u, where u has the structure which is equal to u0 transpose u0 transpose considering the block matrix multiplication, sorry u0 transpose u0, which is equal to I can write, think of this as GG transpose because it has the structure that u0 is upper triangular with n positive entries along the diagonal.

So, this is like my G transpose matrix and this is my G matrix. So, u0 transpose is the same as G. So, what this means is that the u0 resulting from the QR decomposition, from the QR decomposition of X, is, in fact, the Cholesky, in fact, the Cholesky, in fact, the transpose of the Cholesky factor of mR hat, m is just a scalar. So, in fact, it is a scaled version of it, 1 over square root of m times u0 would be the transpose of the Cholesky factor of R hat itself.

(Refer Slide Time: 31:05)



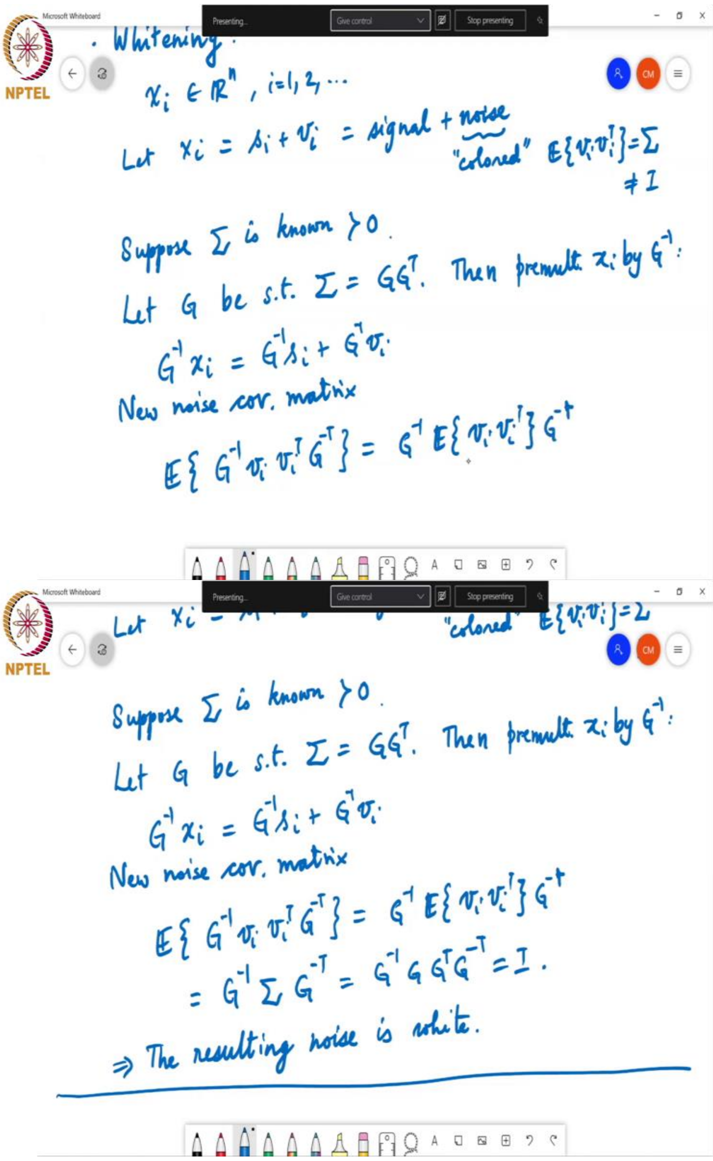
So, continuing with these points, this Cholesky decomposition is useful for, for example, if you want to generate a vector process with a desired covariance matrix. It is useful for simulation purposes. So, suppose, we want to generate an x which is in R to the n with a desired covariance matrix and let us call the desired covariance matrix sigma, symmetric.

Any covariance matrix is by definition symmetric, positive definite covariance matrix. Then what we can do is, so, we first find, since sigma is symmetric and positive definite, we can find G such that sigma equals GG transpose. Then what we do is, we say we start with the vector w which is in R to the n be a random variable, such that its covariance matrix expected value of ww transpose equals the identity matrix.

So, generating such, these are called isotropic, isotopically distributed random vectors is easy. So, you just generate vectors with independent and identically distributed entries, then the expected value of ww transpose will be 1 if the variance of each of those entries is equal to 1. Then, then we define x equal to G times w, then the expected value of XX transpose will be equal to the expected value of Gww transpose G transpose, which is equal to, G is just a linear operator and expectation is also a linear operator.

So, I can exchange them and pull the expectation inside and pull the G transpose out from the other side. So, this is G times the expected value of ww transpose times G transpose which is equal to GG transpose is equal to sigma. So, we have generated random vectors that have this desired covariance matrix sigma. This is useful as I said for computer simulations to creating vector processes with a desired covariance matrix.

(Refer Slide Time: 34:46)

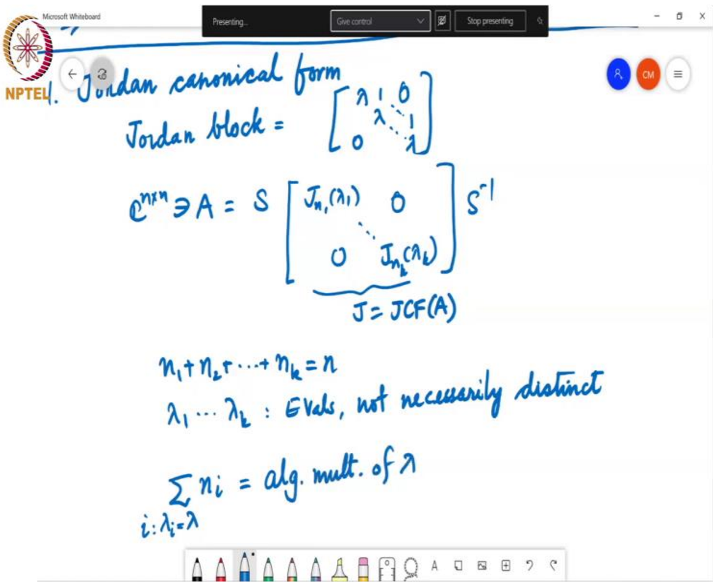


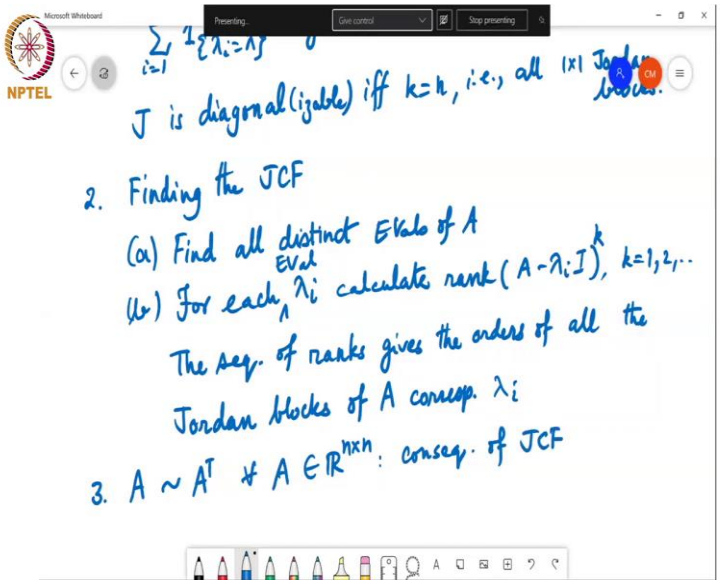
The converse of this is what is called whitening, which is also very useful because, when we have, for example, a stationary random process xi, which is R to the n and i equal to 1, 2, etcetera. And suppose, we get to observe or we get to observe this xi, with xi being equal to some si plus vI, where si is the signal plus vi is the noise and this vi being a noise is coloured. Meaning the expected value of vi vi transpose is not the identity matrix, but some of the matrix sigma is not equal to the identity matrix.

So, suppose the sigma is known, somehow you have access to independent noise samples from which you are able to estimate the noise covariance matrix and suppose sigma is known and is positive definite then, then there is the G that sigma equal to GG transpose. So, let G be such that sigma equals GG transpose, then what we will do is we pre multiply xi by G inverse, by G inverse what that gives us is G inverse times xi is equal to G inverse times si plus G inverse times vi.

Then, the new noise covariance matrix is the expected value of G inverse vi vi transpose G inverse transpose which is equal to G inverse expected value of vi vi transpose G inverse transpose and this is just sigma. So, that is equal to G inverse sigma G inverse transpose, which is equal to G inverse GG transpose G inverse transpose, which is equal to the identity matrix. So, we whiten in the noise. So, the resulting noise is white. So, basically this Cholesky decomposition is very useful and noise whitening, which is a very important tool in (())(28:26) processing. And in particular, Cholesky is used because it is stable and it is easy to compute.

(Refer Slide Time: 38:38)





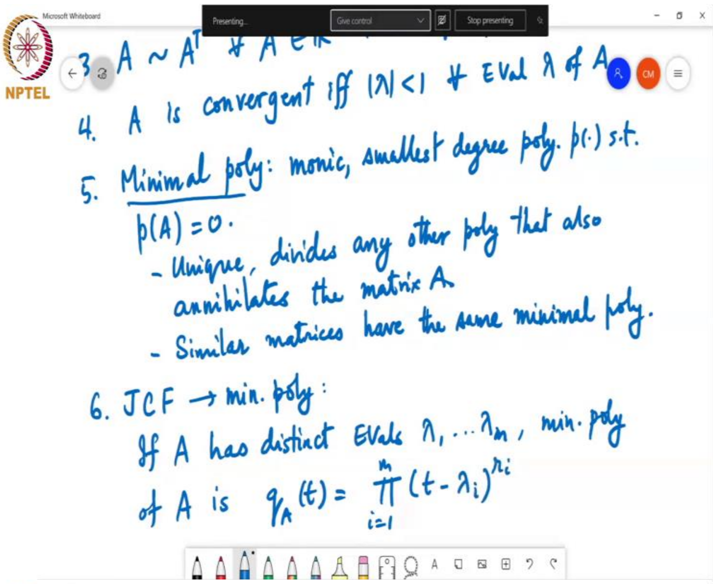
So, just to summarize what we have seen in this, seen so far in this chapter. Is that we looked at the Jordan canonical form, where the main working block was this Jordan block which has the farm lambda along the diagonal and once on the first super diagonal and 0 is everywhere else. And we saw that in the Jordan Canonical Form any A can be written as S times a matrix that contains some Jordan blocks call it, Jn1 of lambda 1, Jnk of lambda k, and 0 is everywhere else, times S inverse.

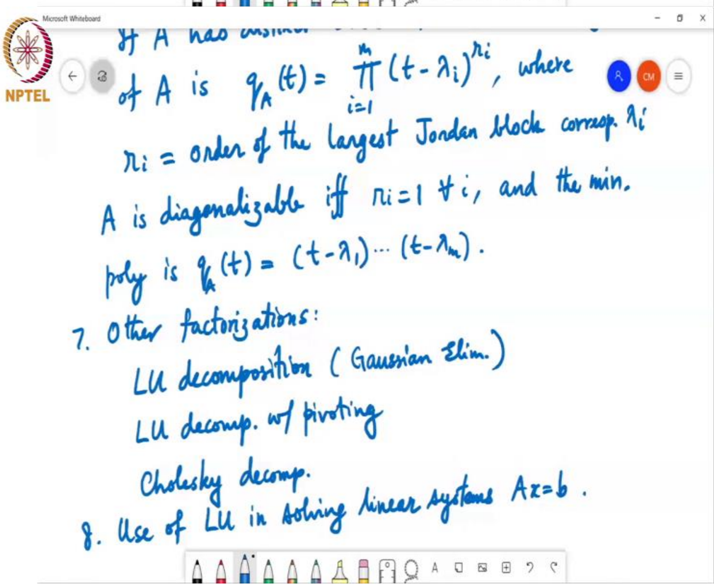
And we call this matrix J which is the Jordan canonical form of A. And here these n1, n2 their block sizes are such that n1 plus n2 plus, etcetera plus nk is equal to n and lambda 1 to lambda k are the eigenvalues of the matrix and these eigenvalues are not necessarily distinct. And further, if I look at the summation of ni overall all i such that lambda i equal some particular value lambda this gives me the algebraic multiplicity of lambda and if I look at the sum of i equal to 1 to k. I just add one each time lambda i equals lambda.

This counts the number of blocks in which this eigenvalue lambda appears in the Jordan Canonical Form and this is equal to the geometric multiplicity of lambda. And of course, see from this itself you can see that the algebraic multiplicity is greater than or equal to geometric multiplicity. If all these blocks are of size 1, then all these ni's are equal to 1 and these two will be equal. And so, the matrix J or A is diagonalizable, if and only if k equal to n, that is all are 1 cross 1 blocks.

And the other thing we saw, the next thing we saw was how to find the JCF Jordan Canonical Form, what we do is the recipe we wrote out, where the first step is to find all distinct eigenvalues of A and then for each lambda i, eigenvalue lambda i we calculate the rank of A minus lambda i times the identity matrix power k for k equal to 1, 2, etcetera. And we study the sequence of ranks and this sequence gives the orders of all the Jordan blocks of A corresponding to the eigenvalue lambda i. Let me just do this, let me just write here for each eigenvalue. And one very interesting consequence of the Jordan Canonical Form is that A is similar to A transpose for every A.

(Refer Slide Time: 44:40)





And we saw that A, matrix A is convergent if and only if mod lambda is less than 1 for every eigenvalue lambda of A, this is also true for non- diagonalizable matrices. And then we discussed a bit about the minimal polynomial, which is a monic polynomial that is the leading coefficient equals 1 and smallest degree polynomial that annihilates A, that is if i p such that p of A equals 0. And this minimal polynomial is unique and divides any other polynomial that also annihilates A.

And the other thing we saw is that similar matrices have the same minimal polynomial. And other thing is that the JCF can be used to find the minimal polynomial, although it may not be the best way to do it. But what you do is that, if A has eigenvalues, the distinct eigenvalues lambda 1 to lambda m, then the minimal polynomial is of the form to A of t equal to the product i equal to 1 to m t minus lambda i power ri, where ri is at least equal to 1, but it is equal to and in fact, ri is the size of the or the order of the largest Jordan block corresponding to lambda i.

And, of course, A is diagnoseable if and only if ri equals 1 for all i. Let us comes back to the point that all the Jordan blocks are 1 cross 1 if ri is equal to 1. And the minimal polynomial is of the form, say t minus lambda 1 to t minus lambda m. Then, we looked at other factorizations which is the Lu decomposition and the Lu decomposition with this is just nothing but a Gaussian Elimination and Lu decomposition with pivoting which is a numerically stable way of computing the Lu decomposition, and then we looked at Cholesky decomposition.

And we briefly discussed about the use of Lu in solving linear systems. So, this just sort of summarizes what we saw in this chapter so far. So, with this, I conclude what I wanted to say about these matrix decompositions.