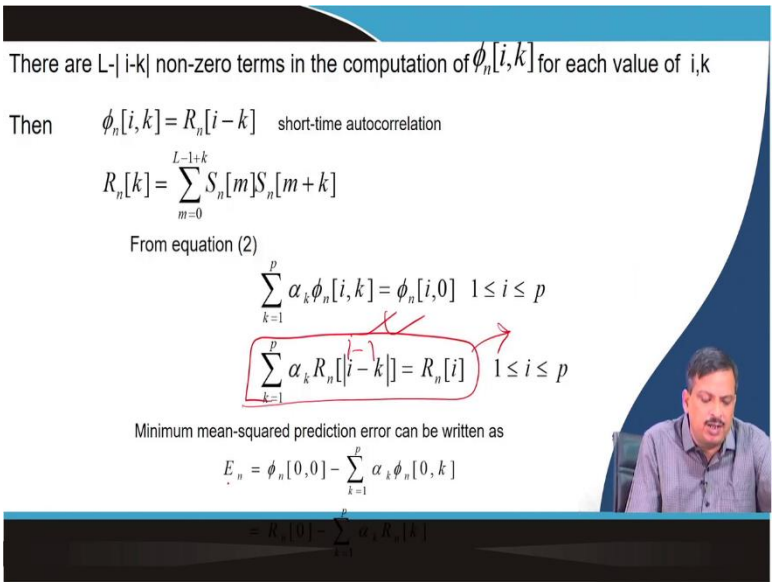
Signal Processing Techniques and Its Applications
Dr. Shyamal Kumar Das Mandal
Advanced Technology Development Centre
Indian Institute of Technology, Kharagpur

Lecture - 47
Covariance Method for Linear Prediction

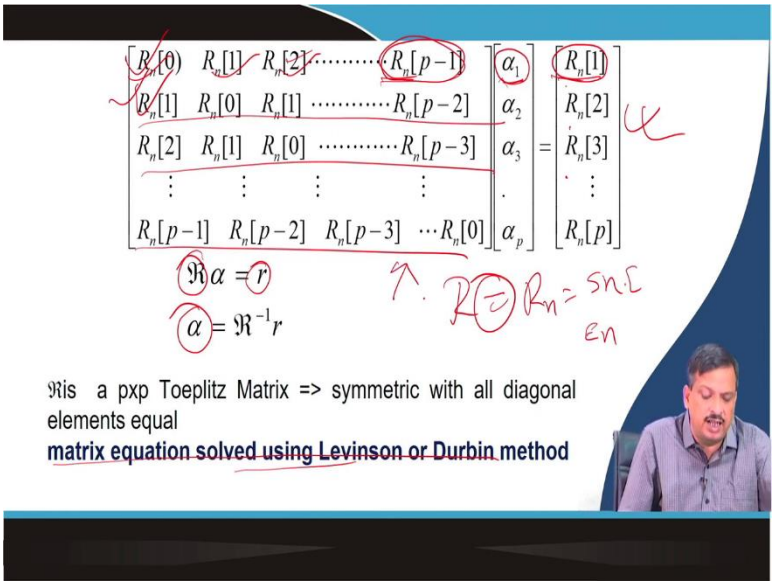
Ok. So, in the last class, we talk about the calculation of α value using the autocorrelation function. So, autocorrelation I have explained all those things in autocorrelation.

(Refer Slide Time: 00:30)



So, we are here to say that we have derived this equation. So, what is the equation? We said that the equation is this one α. This one is equal to Rn i. Now, if I want to represent this set of equations in matrix form, I can say that k varies from 1 to p, so let us say i equals 0. So, i equals 0 minus 0 minus k equals 1, so minus 1.

(Refer Slide Time: 00:56)

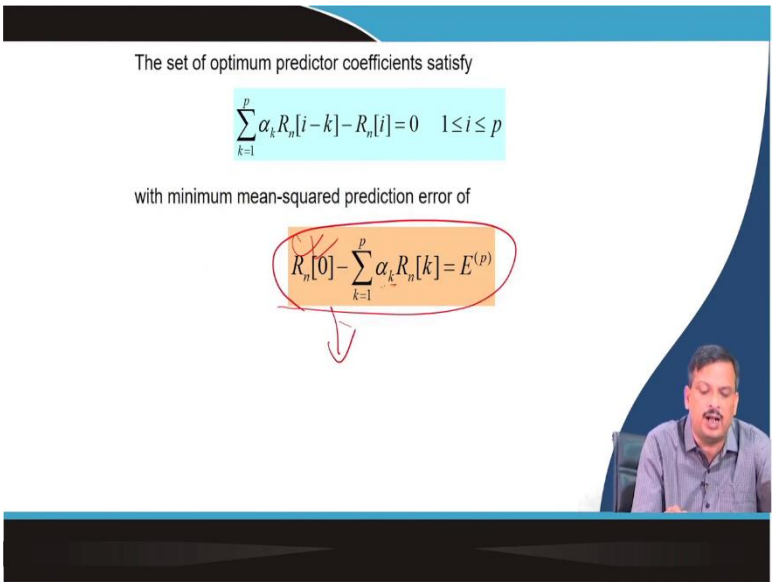


So, I can say that R0, R1, R2, Rn p minus 1, which is nothing but a multiplied by the α1 k equal to 1, means α1. So, α1, then it is nothing but a 1. So, I equal to I said I equal to 1. So, 1 minus 1 is equal to 0. So, it is R0, then say that 1 minus 2. So, mod of minus 1; that means mod means 1.

So, this is R1, then R2 Rp minus 1. And that is equal to multiplying by α1, and that is Rn 1 I equal to 1, then I equal to 2, I equal to 3, I equal to 4. I will get that, and that is the matrix form of this equation. So, I can say this matrix so, this is matrix let us say capital R; capital R into α equal to small r. What is capital R? It is nothing but an auto-correlation matrix.

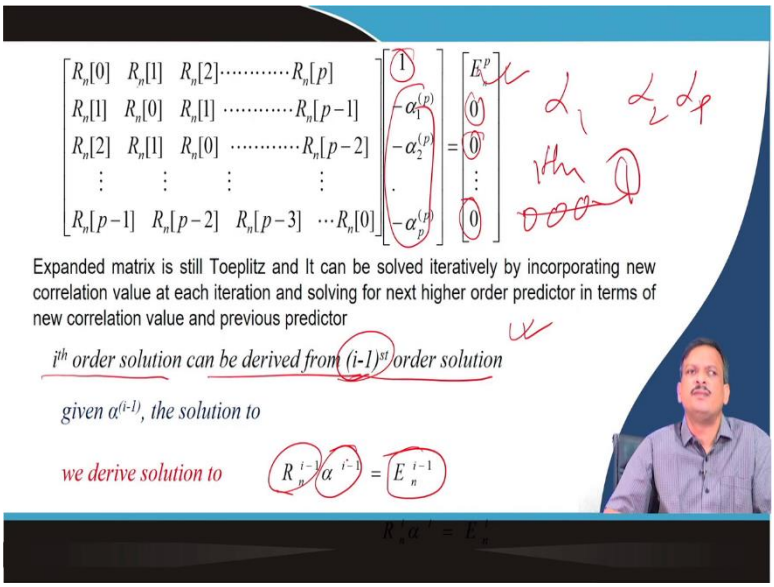
R n 0 is nothing but a signal multiplied by the same signal, which is nothing but the energy; Rn 1 just shifted by one sample. So, if I know all R, if I know this Rn 1, Rn 2, this one, then I can easily calculate the α value. So, this is the procedure and that procedure is called the Levinson Durbin method. So, this matrix is solved using Levinson Durbin methods.

(Refer Slide Time: 02:47)



Now, another one is that the minimum mean square error E n is equal to φ n 0 0. So, I can say that minimum mean square error can be expressed using the same procedure now if I want to represent this one in a matrix form.

(Refer Slide Time: 02:56)



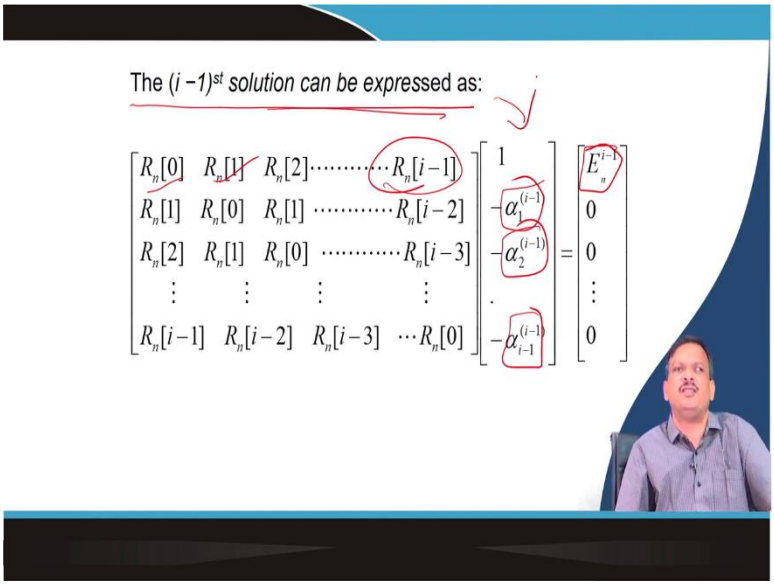
So, I can say this is nothing but R0, R1, and R2 multiplied by 1. I get R0, which is equal to E p. The Rest are minus α1, minus α2, minus α3 I multiplied and which are equal to 0.

That means when my predictor is 100 per cent correct, then the error is equal to 0 OKs. Now, I can solve this matrix iteratively. What are the iterative methods? That means the

ith order solution can be derived from^{i-1} order solution. So, physically what we are doing, let us say we are calculating the error for a first set of α; α1, α2, α p. Then I know that when I am predicting nothing to a signal, the prediction error is maximum.

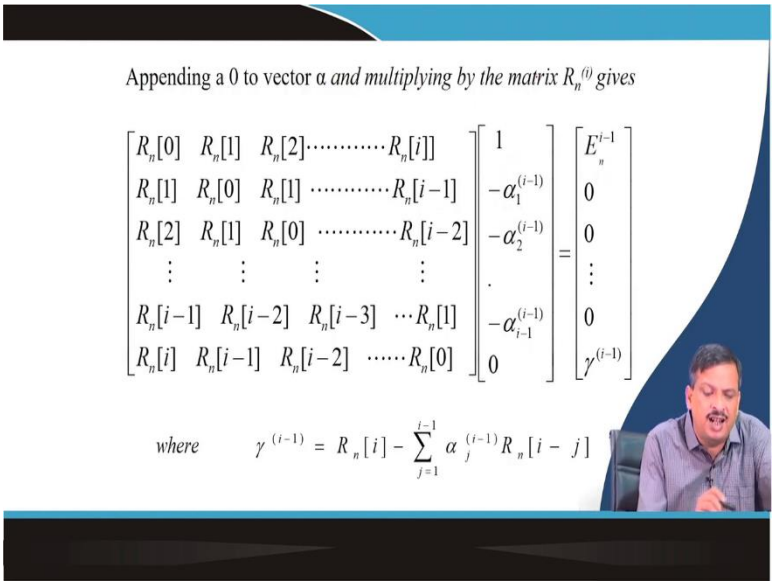
Now, when I go, the order increases. So, when I say the 1st order, I am predicting, that means I am predicting all 0th samples and the 1st sample, which will be an error, will be maximum, and when the order is gone, then the error will be minimum. So, I can say that I can say that I 2nd order prediction can be derived from 1st-order prediction. So, Rn^{i-1} αi minus 1 is equal to E n^{i-1} ok.

(Refer Slide Time: 04:35)



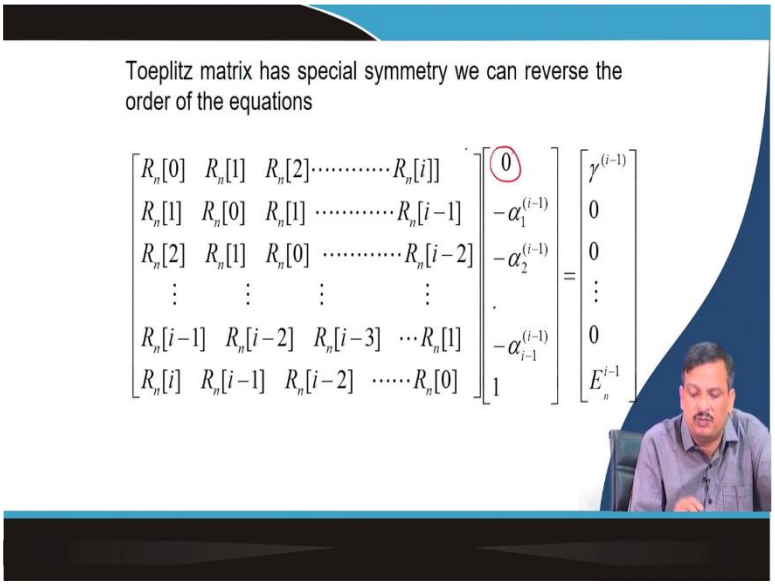
So, now, if I say that, then I can say, what is minus 1? What is a solution? R0, R1, R^{i-1}. So, it is nothing but an E n^{i-1} α1^{i-1} α2 i. So, 1st set of predictions, then 2nd iteration, then 3rd iteration. So, once the iteration is going on, that error will be minimized.

(Refer Slide Time: 05:07)



Now, I can say that this is the iterative method that I used to solve this matrix. So, if I solve this matrix, I just show you that because this is mathematics, you can see that matrix solution by going through the books. So, mathematics we can give. So, now, I append 0 to a vector α and multiply by Rn^{i-1}.

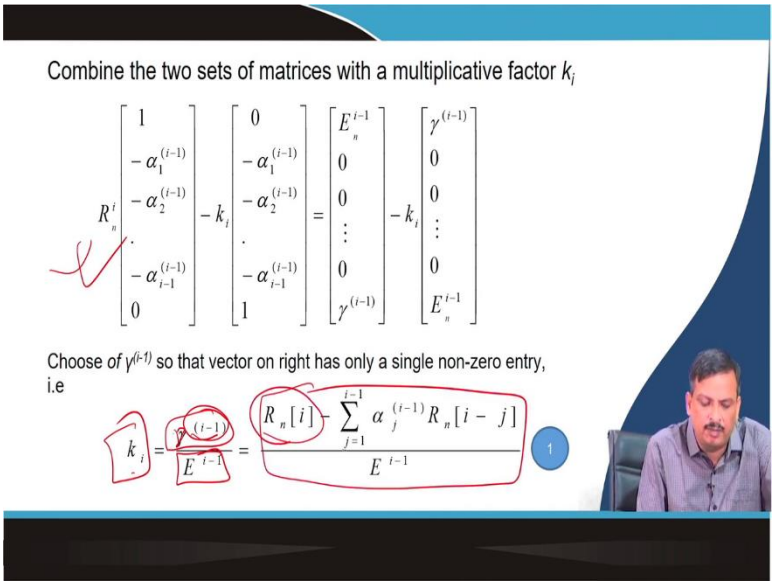
(Refer Slide Time: 05:24)



So, I am append 0 and multiplying by Rn^{i-1}. I get this one, sorry, so, 1 0. Now, appending 0 in here. So, if this is multiplied by column 0 0, nothing will happen. So, I can multiply this one and append a 0 vector here, which is nothing but a $sigma^{i-1}$; let us use this side.

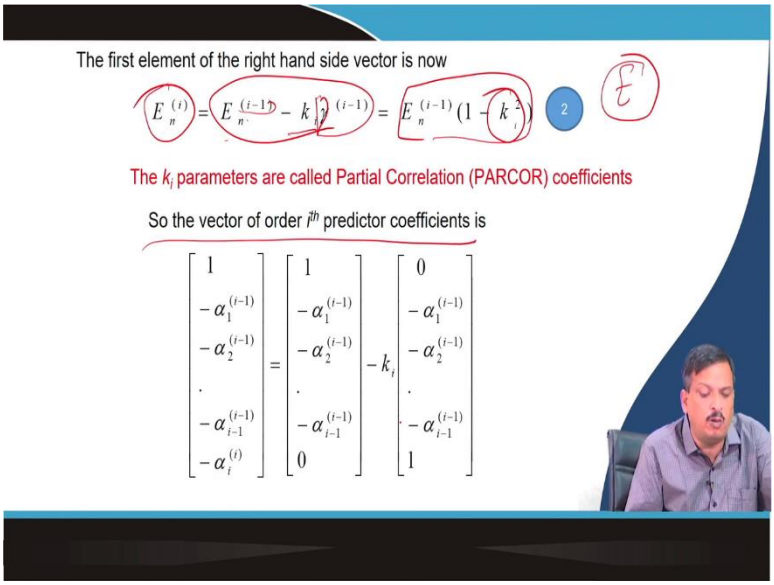
So, this can be solved from this matrix: sigma I minus psi^{i-1} is equal to Rn I minus j equal to 1 to^{i-1} this one, then I can reverse because it is a symmetry matrix.

(Refer Slide Time: 06:17)



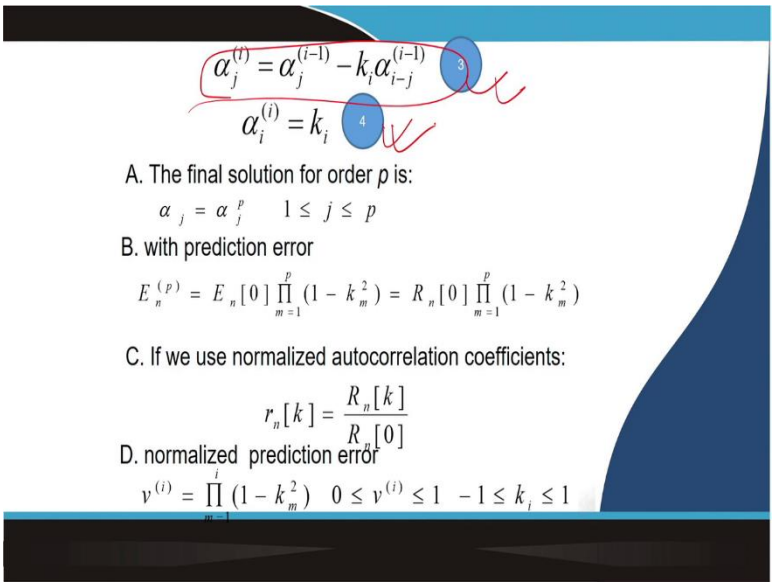
So, whether it is reversed, the order does not change it. So, I can change the reverse order, and then I can write in this combination. From this combination, I can say that k i is equal to sigma psi^{i-1}. This is, sorry, γ^{i-1} divided by E^{i-1}. So, γ^{i-1} I have already done that divided by E^{i-1}, I have done that. So, I can say k i is equal to Rn i j equal to 1 to^{i-1}. So, this is one equation I can get.

(Refer Slide Time: 06:50)



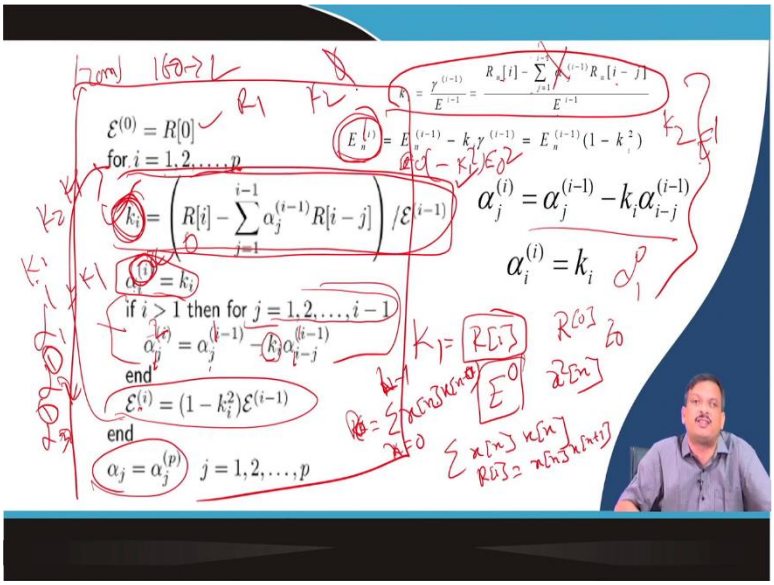
Now, once I get the k i, I can calculate that E n I, E n i is nothing but an E n^{i-1} minus k i this γ^{i-1}. So, it is nothing but an E n^{i-1} 1 minus k. So, I can say the energy once the prediction order is increased iteration increased energy will be minimized by multiplying by k I square k I square and that k i is called the partial correlation coefficient or PARCOR. So, k i is called the partial correlation coefficient, then I solve for ith predictor, and I can solve for αi j.

(Refer Slide Time: 07:34)



So, ultimately, I can say there are four equations I have to solve for calculating the α value 1st equation is this one. So, the 1st equation is this one, the k i is equal to this one, the 2nd equation is this one, the 3rd equation is this one and the 4th equation is this one.

(Refer Slide Time: 08:18)



So, let us say I want to implement it using a program. So, what can I do? I can calculate that. Let us say I copy these whole four equations on one page. Now, let us say I want to implement these four equations in a program. So, what is required if you see what is required from the 1st equation? From the 1st equation, what is the requirement? If I want to calculate k 1, let us say i is equal to 1. So, I require R1, I require E 0, I require R.

So, i equal to 1. So, i equal to 1 means j is equal to 1 to 1 minus 1. So, j is equal to so, 1 to 0. So, that term will not be there. So, I require k 1 to calculate k 1, and I require R1 and E 0. So, what is R1? What is R1? R1 is nothing but an autocorrelation coefficient shifted by 1. So, I can suppose I have given a signal. Let us say I give you a signal that I have a speech signal.

Let us say the speech signal is recorded with an 8-kilo hertz sampling frequency, and then I take a window size of 20 milliseconds. So, I have 160 samples in my speech signal. So, from 160 samples, I have to calculate this R0, R1, R2 autocorrelation coefficient. So, what is the equation? I know R is Rk is equal to k equal, so let us say n equal to 0 to n minus capital N minus 1 or capital L minus 1. This is L, x[n] multiplied by again x[n+k]. Let us say.

Outside the window L, the signal is 0. So, when I say R0, that means k is equal to 0; that means I am multiplying x[n] with x[n]. So, each sample will multiply its own sample. So,

it is nothing but x square of n, so that means nothing but the energy of the signal. So, it is nothing but an E 0. So, if R0 is equal to E 0, then what is R1?

R 1 is nothing but a shift of the signal by one sample x[n] multiplied by x[n] plus 1, R1 I can calculate. So, let us say I calculate R0 and R1. Then I know what k i? k is i is nothing but an R0; R0 I know my divided by so, R1 I know I divided by E 0. So, E 0, I know R0 is equal to E 0. So, k 1, I know. So, similarly, if I want to calculate k 2, then you can see I required R2, I required R1, and I required E 0.

So, I require R2, R1, and E 0, and then I will use the same equation I am implementing. So, I will vary from 1 to p, and k I will be k 1 and k 2 to k p. Now, once I calculate k 1, this for loop will be executed for i equal to 1 to p ok. Then I know α j i is equal to k i ok. Then, if I see this loop, then I can calculate E n i because when I say k 2, I require E 1. What is E 1?

E 1 is nothing but an E 0 minus k 1 square 1 minus k 1 square into E 0. E 0 I know k 1 I know. So, I can calculate E 1; E 1 is required for k 2 calculation αi i. So, α1 1 is equal to k 1 α1. So, i equal to 1 2. So, α1 2, α3 1. So, iteration is 1 first iteration. When I said, then I equal to 1 to p. So, in the first iteration, I get α1 1 α1 2. So, α2 1 α3 1.

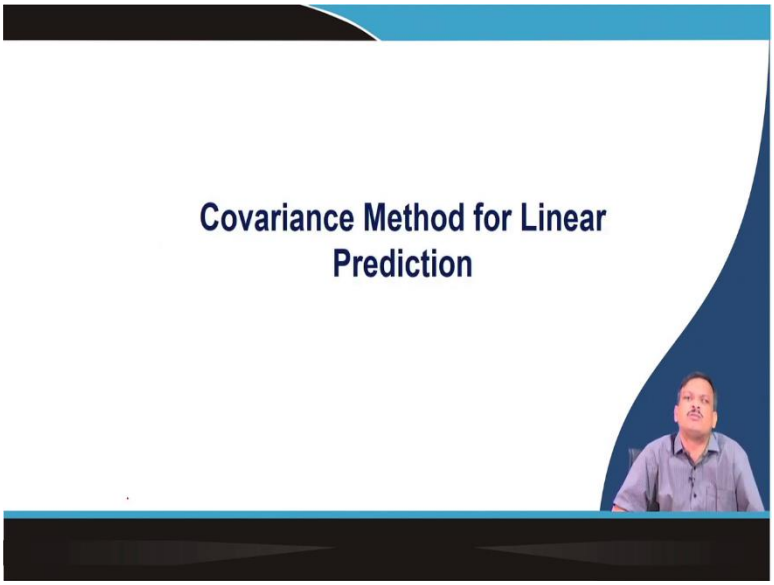
So, the upper one is the iteration, and the lower one is the α different α value. Now, once I get that, then I know if I am equal to 1, then nothing is there; if I am equal to 1, but if the because it is α1 0. So, more than 1. So, j is equal to 1 to $^{i-1}$ when I say, then I can say. So, α1 I cannot iterate. So, from the second stage, I can iterate.

So, I can say the j is equal to 1 α2 let us say. So, j equal to 1 α2 is nothing but a j equal to 1 α1 minus k 1 into α1 1 understand and modify the energy. So, when I equal to 1, this portion will not enter. So, it modifies the energy and goes back to here, and for i equal to 2, the k 2 will be calculated, then the α1 1 will be 1 2 will be calculated.

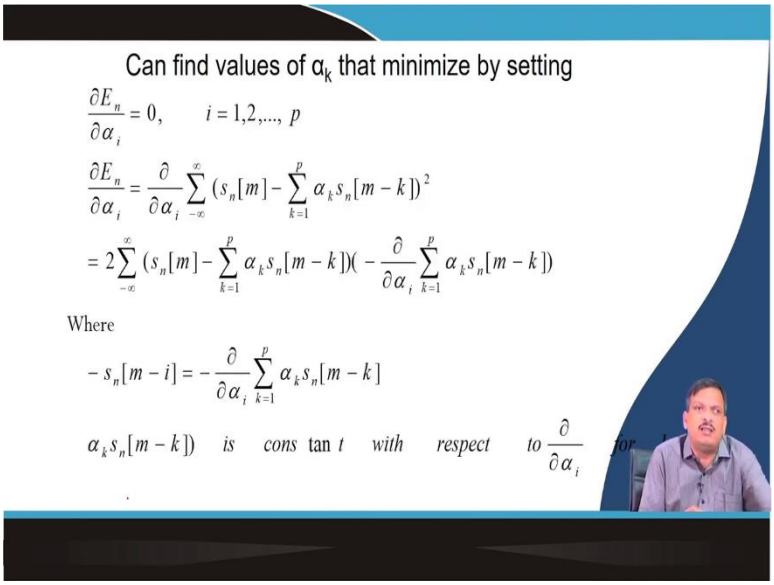
That way, this whole equation will be implemented. So, I ultimately get α1 p. So, what I am saying is that I am calculating α value based on the iteration value. So, if you use this for loop implement in a C program and check for any recording of sound, let us say record your name and implement it take a frame and then implement it to calculate that α value. Outside the window, your signal is 0.

First, you have to calculate all autocorrelation coefficients, and then you have to do it. So, that is the one method for the calculation of the linear prediction coefficient using the autocorrelation method. What is the assumption? Outside the window, the signal is 0, and the error is maximum at the beginning of the window; at the end, the error is maximum ok.

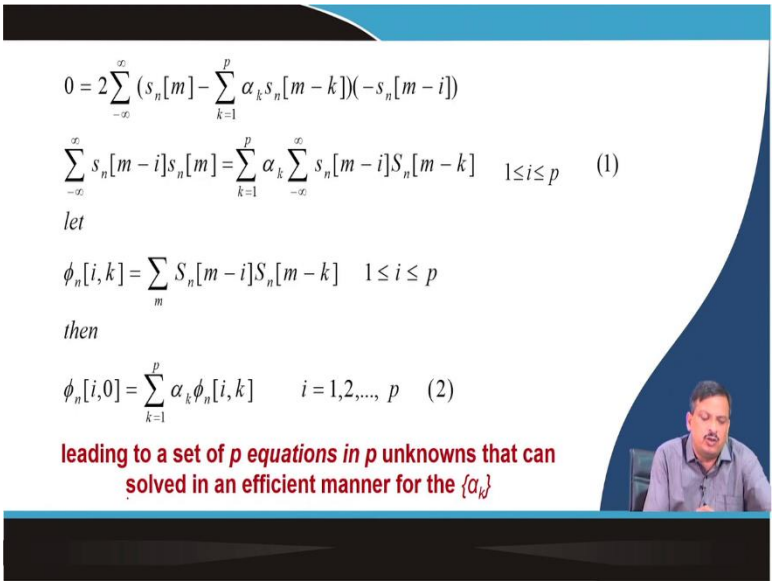
(Refer Slide Time: 15:14)



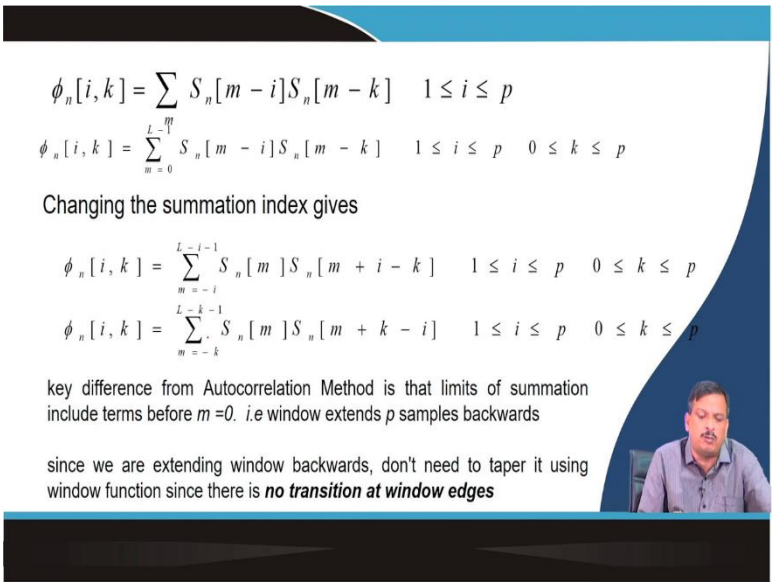
(Refer Slide Time: 15:19)



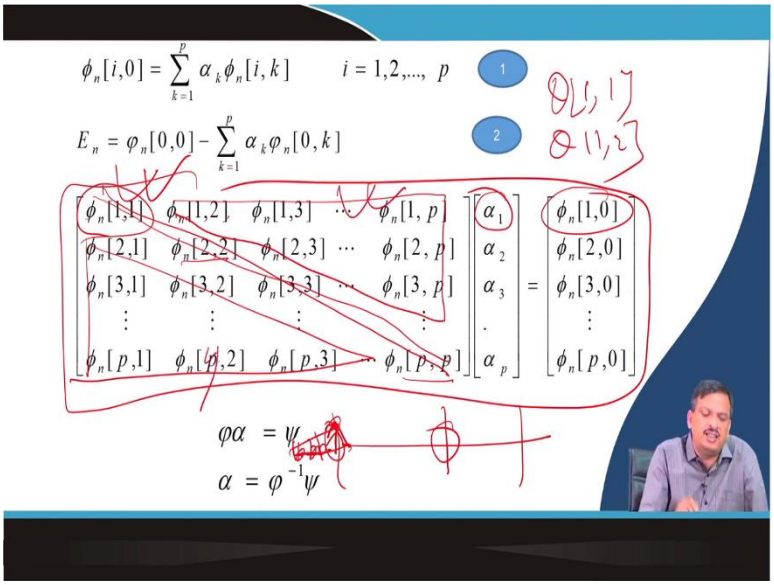
(Refer Slide Time: 15:22)



(Refer Slide Time: 15:27)



(Refer Slide Time: 15:28)

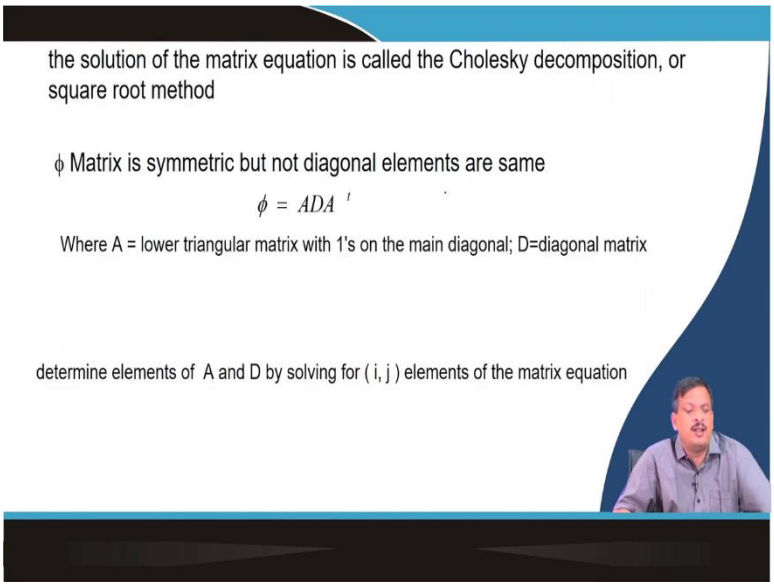


Now, there is another method called the covariance method. instead of going to autocorrelation, I can directly calculate. So, the φ matrix is the same. So, this is the derivation of the φ matrix, which is the same here also. So, once I get this φ matrix. What is φ matrix? It is nothing but a covariance.

So, instead of replacing the covariance with an autocorrelation coefficient, I can directly calculate φ[1, 1], I can directly calculate φ[1, 2]. I am saying the signal is not 0 outside the window. So, I am not taking the signal as 0 outside the window.

I am taking. I am saying I have a signal. I take a window. So, outside the window, the signal is not 0. So, when I calculate the error prediction error for this sample, I do not predict from 0; I predict from the original sample value. So, that error is not maxima at the boundary edge. So, I directly calculate all the covariance. So, then, that means I know the φ matrix. I so, once I know the φ matrix, I can calculate the α matrix by simple matrix decomposition method.

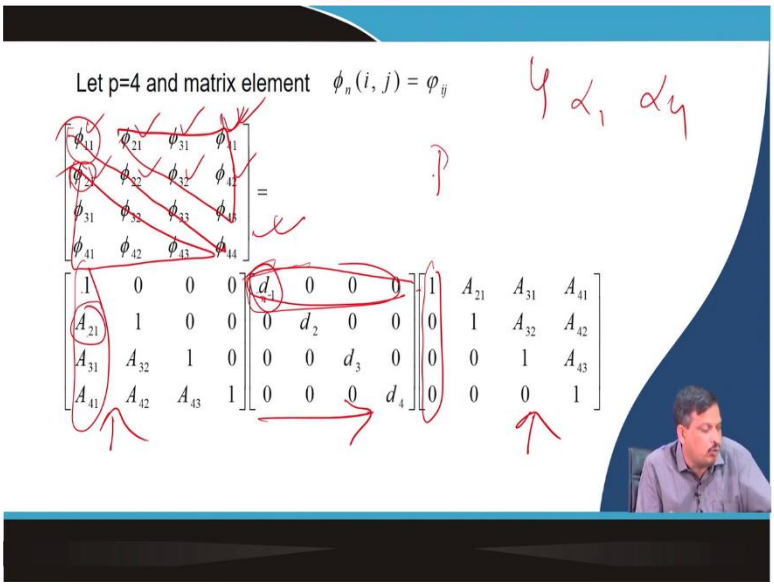
(Refer Slide Time: 16:37)



What is the matrix decomposition method? Let us say I have a matrix φ is a matrix symmetric matrix because if you see φ[1, 1], φ[1, 2]. So, all diagonal elements φ[1, 1], φ[2, 2], φ[p, p] ok. So, all diagonal elements are the same; that means φ[1, 1] is not the same, but there are diagonal elements are there.

And those are the, but if you see this portion of the matrix and this portion of the matrix is same. So this is called the upper triangular matrix; this is called the lower triangular matrix.

(Refer Slide Time: 17:24)

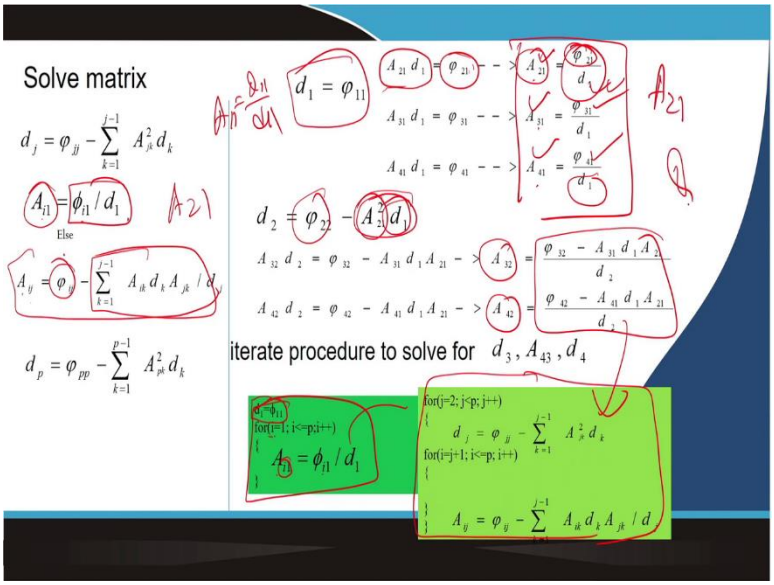


So, I can say that let us say, for example, let us say I have a 4th order 1 2 prediction 4th L α1 to α4 α1 2 3 4 I want to predict. So, what do I know? I know φ[1, 1], φ[2, 1], φ[3, 1], φ[4, 1], φ[2, 1], φ[2, 2], φ[3, 2], φ[4, 2] understand.

So, now, I know all those things. So, I can say that all one is a diagonal element, one is an upper triangular matrix, and one is a lower triangular matrix. So, let us say I decomposed this matrix into a lower triangular matrix, diagonal matrix and upper triangular matrix. If I multiply these three matrices, I will get this one. Now, if you see that if I do that, if I solve this matrix. So, I can say that d 1. So, first, you multiply this one by this column.

So, d 1 then this the I can say that this column with this row. So, I can say d 1 is equal to φ[1, 1] only else are 0.

(Refer Slide Time: 18:47)



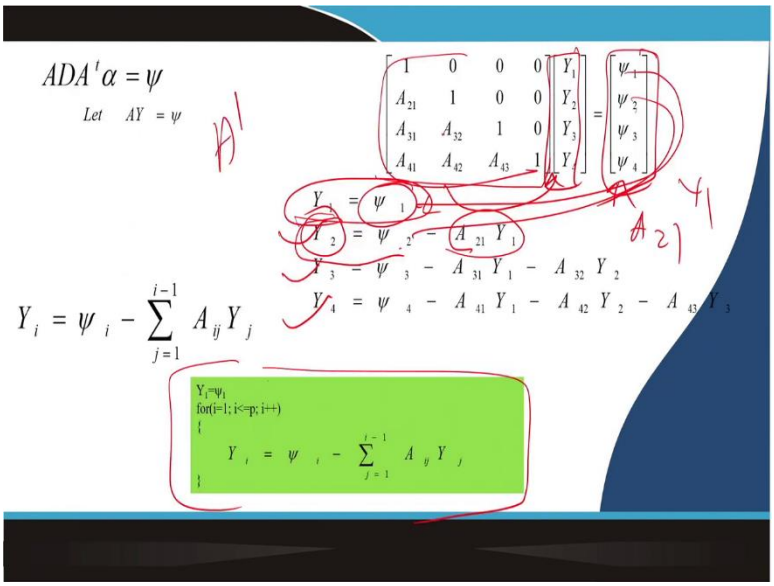
So, φ[1, 1] is equal to d 1. Then what is d 2? d 2 is nothing but a φ[2, 2], d 2 is nothing but a φ[2, 2] minus A21 square into d 1. And if you see A21 d 1 is equal to φ[2, 1] A21 d 1 come in here. A21 multiplied by d 1 is equal to φ[2, 1] ok. So, I can say that φ[2, 1].

So, I can say A21 is nothing but a φ[2, 1] divided by d 1. So, I know d 1 I know φ[2, 1]. So, I can calculate A21, I can calculate A31, I can calculate A41. Since a d 1 is known and all φ is known. So, first step, what is the first step? The first one, d 1, is equal to φ[1, 1]. This is the generalization.

Rest cases I can say that A. So, if I generalize this one, A21, so, 2 1 means, let us say, A i 1. So, this is 1, which is fixed; only say the 2, 3, and 4 are varying. So, i from A11 is equal to d 1. So, diagonal, I can say that φ i is equal to 1 2. So, A φ[1, 1] divided by d 1. So, I can say that A21. So, I can say that φi1 is equal to φi1 divided by d 1 else. Only the case of A11 is equal to φ[1, 1] divided by d 1.

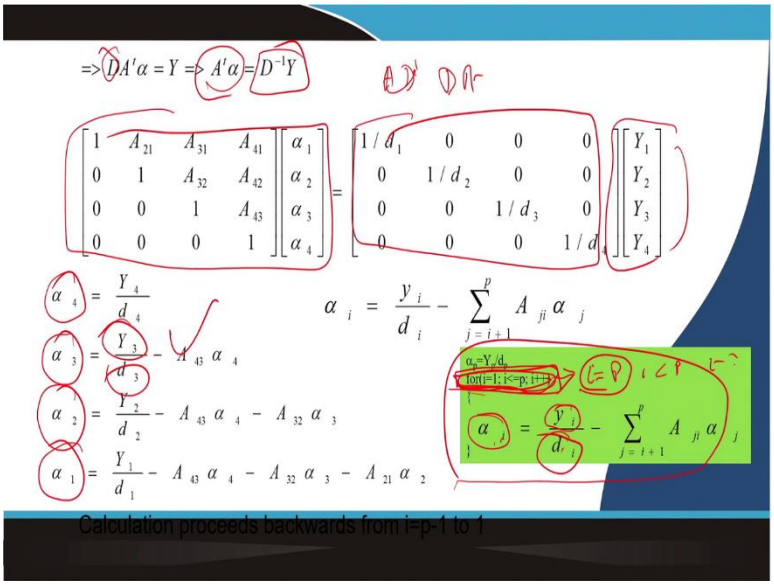
Else I can say that A i j A i j A21 is nothing but a φ i j minus this one. So, first you just do for this one and then you go for if you see that A32, A42 I can calculate like this one. So, I just implemented that in C programming. So, once I know the d p A matrix and the d matrix I can, I can calculate it from there from a given φ matrix.

(Refer Slide Time: 21:34)



So, A and d I know. Let us say this: A matrix multiplied by Y is equal to this one, ok? So, now let us say this one multiplied by this one is equal to this one. Then you generalized it if I multiplied it: Y 1 is equal to this one, Y 2 is equal to this one, Y 2 this one minus this one, Y 1. So, what is Y 1? I know what it is Y, then I can calculate Y 2 because A21 I already know. So, A21 is known as Y 1. So, I can calculate Y 2, Y 3, Y 4. So, I write down the program.

(Refer Slide Time: 22:22)



Now, like once I know that, what can I do? I can say the A D, D A transpose because if you see that if the A this one is A, what is A transpose? Nothing but an upper triangular matrix. So, A transpose α is equal to D inverse Y. So, what is D? D is nothing but a diagonal matrix. So, what is the inverse of the diagonal matrix? This one multiplied by Y is equal to A transpose α. Now, I calculate for α value α4 α3.

So, if you see that order, I get α4 first, α3 second, α2 third and α1 first. So, it is in reverse order. So I can write down the program in reverse order. Maybe the programming will be reverse-ordered. Now, instead of forwarding the order, it will be reversed. Not forward order. This program is wrong. This is in forward order. It will be I equal to p 2 I less than p plus and p I minus.

So, then I get α4; α4 is equal to Y 4 divided by d 4 Y 4 I knew from the previous slide and d 4 I can calculate here d 4; d 4 I know. So, I can calculate α4 then α3, α2, α1 this is known, this is known, this is known. So, this is a mistake; here is a mistake: if you see this for loop, it will be p i equal to p 2 less than p i minus less than equal to p i minus minus.

So, this is the matrix decomposition form. I can calculate that α value using matrix decomposition form, but what do I have to calculate? I have to calculate covariance. Is that okay or not? But, if you see in the case of autocorrelation, what do I have to calculate? I have to calculate the R matrix and the autocorrelation coefficient matrix. In the case of

autocorrelation, I have to calculate the R matrix. In the case of covariance, I have to calculate the φ matrix.

So, I can say the autocorrelation and covariance methods are two-stage method; in the first stage, I have to calculate autocorrelation case of the first stage, I have to calculate covariance. In the case of autocorrelation, I have to learn the Levinson recursion to find out the α value. In the case of covariance, I have to do the matrix decomposition to compute that α value.

So, in both cases, the second stage is iterative, and the first step is a calculation of the R-value and φ value. So, in both cases, what is the computational complexity in the case of autocorrelation methods and covariance methods? For the autocorrelation method in the first stage, I have to compute the autocorrelation, which requires computational complexity. Similarly, in covariance methods, I have to compute that covariance. So, that requires a computation.

So, if the L is equal to 160 samples, I have to calculate R0 R. So, if I want to calculate the 16th order LPC, then L will be there every time, and p will be 1. So, 16 into L number of multiplication I have to do. In covariance, we also have to window length by that time. So, in both cases, computational complexity is in two stages: the first stage is the calculation of autocorrelation, the second stage is the calculation of the covariance, and the second stage is the iteration.

So, now, people are thinking, can we have a solution to find out the single-stage solution; that means, let us say auto correlation iterative methods will not be there that I do not have to compute that autocorrelation. So, can I directly compute that α value, or can I directly compute the k value? If I know the k value, I can compute the α value

So, that method is called lattice formulation. So, for part of the next lecture, I will cover that lattice formulation for linear prediction.

Thank you.