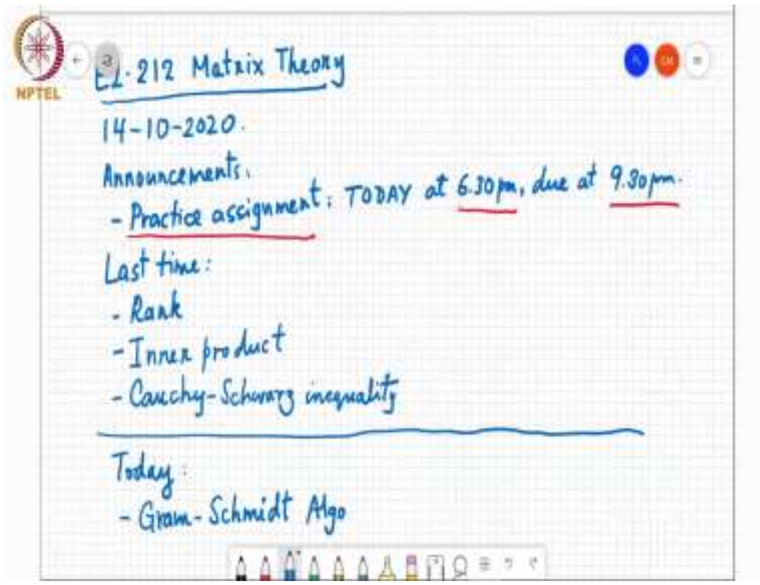
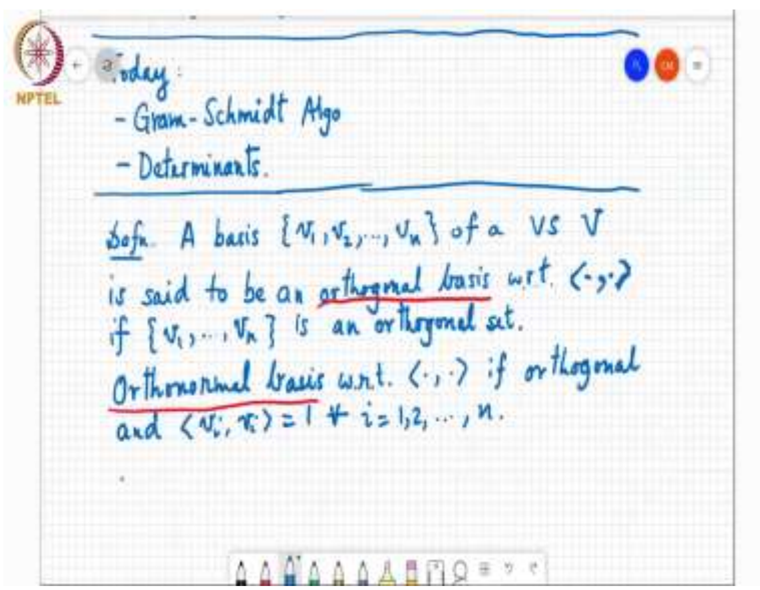
Matrix Theory

Professor. Chandra R. Murthy
Department of Electrical Communication Engineering
Indian Institute of Science, Bengaluru

Gram-schmidt algorithm

(Refer Slide Time: 0:14)



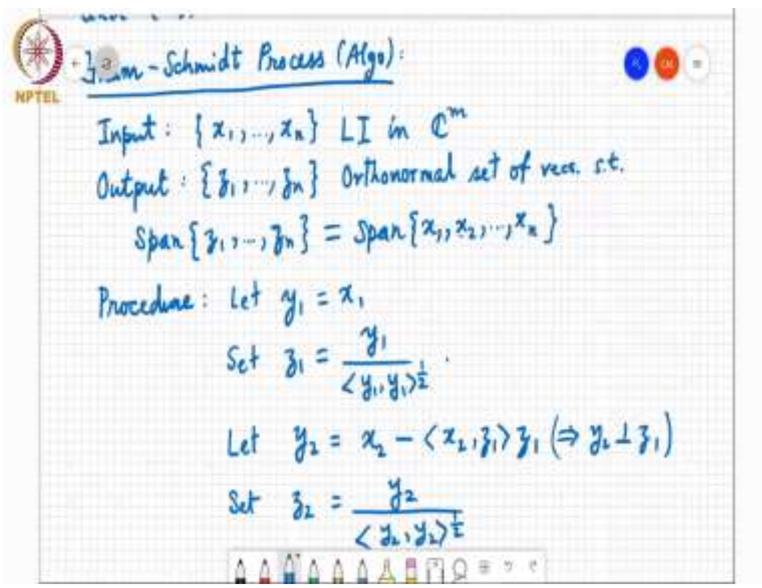


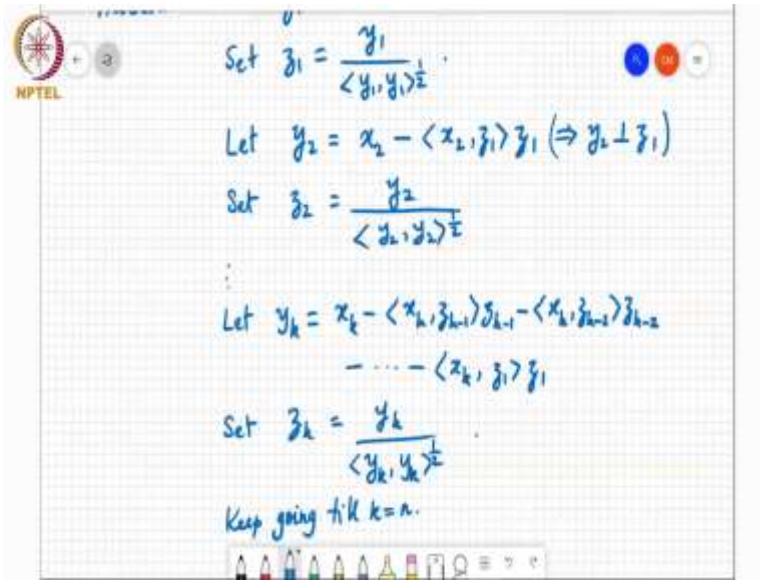
That was the only announcement I made. So, today we will continue this discussion in particular, I am going to talk about the Graham Smith orthonormalization process. So I, there is one definition I want to put down before I begin this is the definition of an orthonormal or orthogonal basis. So, if we are given a basis consisting of vectors v1 through vn of a vector space, vector space V is said to be an orthogonal basis with respect to an inner product.

So, in order to say whether a pair of vectors is orthogonal or not, you need to have some notion of an inner product and I denote that inner product by these braces with 2 arguments here and a comma between them. If this set v1 to vn is an orthogonal set, what is an orthogonal set we defined it in the previous class, a set of vectors are set to be orthogonal or mutually orthogonal if the inner product between any pair of vectors in that set is equal to 0. And we say that it, this basis is an orthonormal basis if it is orthogonal and vi in a product with vi equals 1 for all i.

That is 1 definition. So, 2 keywords that have come here orthogonal basis and an orthonormal basis. So, it turns out that any vector space V and in a product defined like this has an orthonormal basis. So, the question is how do you find such an orthonormal basis? So, if you are given a basis for a vector space, then you can use the Gram Schmidt process or the Gram Schmidt algorithm to find an orthonormal basis for the vector space V. And why are we interested in orthonormal basis? We will be clear as we go further in the discussion.

(Refer Slide Time: 4:16)





So, the Gram Schmidt process or Algo. So, basically this is the input. So, it is a set of vectors say x1 through xn. And these are linearly independent vectors in C to the m. And what is going to be the output of this algorithm is a set of vectors z1 through zn is an orthonormal set of vectors such that span of this first set of vectors or z1 through zn is equal to the span of x1 through xn.

So, by the way, actually the correct way of writing it, writing a sequence is always like this, you have to always write out the first two elements and then separated by a comma, then a comma and exactly 3 dots, then a comma and then the last element of the set accent, you should not put 4 dots, 5 dots and so on. And they should be always these commas in between them. And you should always list the first two but sometimes as usual, I get lazy and I just write z1 to zn, where there is no confusion that I am not skipping anything in between, its just z1 z2 up to zn.

So, this is how the procedure works. And I will first put down the procedure and then maybe make some remarks as to why this procedure will yield a set of orthonormal vectors such that the span of these vectors is exactly the same as the span of x1 through xn. So, I expect that you also seen this procedure previously. So, this is more of a recap.

So, you start out by saying let y1 be equal to x1 itself and then you normalize it. So, you set z1 so I have already found out what z1 is the first vector is going to be y1 divided by the inner product of y1 with itself power half. So, this means that z1 is now normalized, I already mentioned this in the previous class, if you take any vector divided by the square root of the inner product of that vector with itself, you will get a unit norm vector pointing in the same direction as the original vector. So, z1 is now normalized.

Now, I want to find z2, which is orthogonal to z1, but span of z1 z2 will be span of x1 x2. So, how do we do that, so, we let y2 be equal to x2 minus the, minus some scaled version of z1 and the scaling factor is the inner product between x2 and z1. So, now, what this means is that this y2 is perpendicular to z1, why is that true? So, if I take the inner product between y2 and z1 that will be the inner product between x2 and z1, inner products are linear.

So, it is the inner product between x2 and z1 minus this coefficient times the inner product of z1 with itself, but z1 is already unit now. And therefore, the inner product of z1 with itself is one. So, it is the inner product of x2 with z1 minus the inner product of a x2 with z1 which is 0. So y2 is a vector that is perpendicular to z1. And then I normalize this and I set z2 equal to y2 divided by the inner product between y2 and itself power half, so now z2 is unit now and so on, so I proceed like this.

And so just to kind of make it clear in the kth (())(10:01) I let yk equal to xk minus the inner product between xk zk minus 1 times zk minus 1 minus the inner product between xk and zk minus 2, zk minus 2 and so on inner product between xk and z1 times z1. So, what this means then is that yk will be perpendicular to all the previous vectors z1 z2 up to zk minus 1 and so, we can set zk equal to yk divided by inner product of yk with itself power half. So, this is the overall algorithm and you keep going up to k equal to n.

Student: Sir.

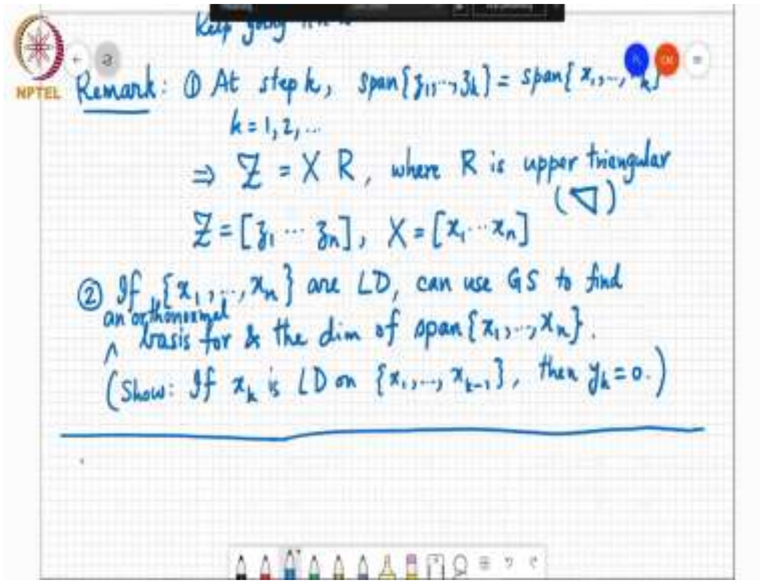
Professor: Yeah.

Student: So, can we say that orthogonal vectors are linearly independent?

Professor: So, if you remember in one of the previous classes, I did say that any set of non-zero orthogonal vectors are linearly independent. And I said you should try to write out a little proof for why that should be the case. So, obviously, if I take the 0 vector, the 0 vector is orthogonal to every other vector, you take the inner product of the 0 vector with any other vector, you will always get 0. So, orthogonal vectors are not necessarily linearly independent but non-zero orthogonal vectors are linearly independent.

Student: Yes Sir.

(Refer Slide Time: 12:37)



Professor: So just to make some remarks about this just by construction. At step k we have that span of z1 through zk zk equals span of x1 through xk. And this is true for k equal to 1, 2 and so on. So, at every k this is actually true. So, what this means is that if I want to write a linear relationship between x and z, then I can write this matrix z whose columns are z1, z2 and so on. This is equal to the matrix X times R, where R is an upper triangular matrix.

So, going forward I will use this notation to write upper triangular in short, so, if you look at this, if R is upper triangular, what it means is that when I multiply so, just for the sake of completeness, I will say here, Z is this matrix formed by z1 through zn. And X is the matrix formed by x1 through xn. I do not want to put commas here. It is just stacking these next to each other.

So, if R is upper triangular. It means that if I look at the first column, remember I talked about how to view matrix multiplication. If I look at the first column of V, it is a linear combination involving only the first column of X. So basically that that is what is captured by this equation here. If I take the second column of Z, it is a linear combination of the first two columns of x and so it goes.

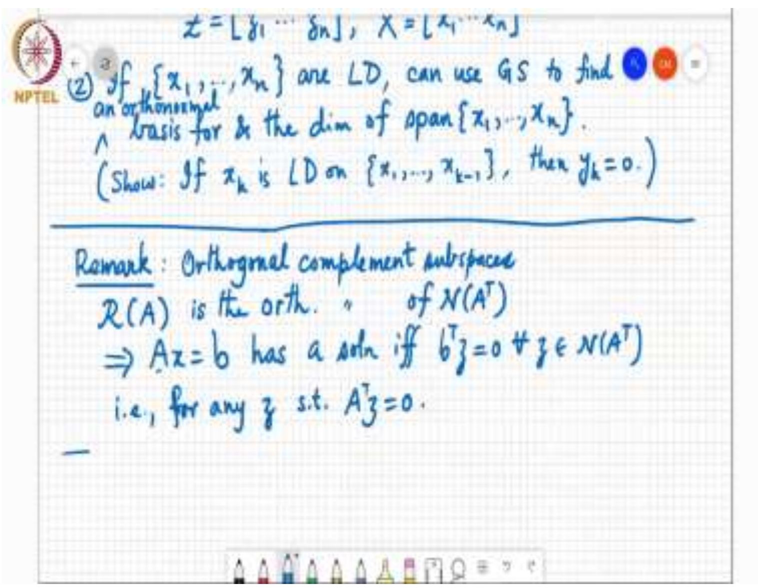
And the second remark I want to make is that if x1 through xn are linearly dependent, remember I presented the algorithm by assuming that x1 to xn are linearly independent vectors in C to the

M, but I can actually run this algorithm even if these vectors are linearly dependent. So, if x1 through xn linearly dependent if they are linearly dependent, we can still use Graham Schmidt to find basis and the dimension of span x1 through xn.

So, basically, how this happens is that if for example, xk is linearly dependent on x1 through xk minus 1, then when you run this Gram Schmidt process, and if you compute this yk here, it will turn out that yk will become equal to 0. So, that is something you should convince yourself about, I will write it down for you. And you should again try to write out a small proof for that then what you can do is to skip xk and then proceed, so you just do not compute a zk, you skip the kth z and then you proceed with yk plus 1 and then you run the algorithm.

So, basically, this is something you should show if xk is linearly dependent on x1 through xk minus 1 then yk equals 0. So, the end of the Gram Schmidt process, you will have an orthonormal basis for a span of x1 to xn. So, to be more clear, these are two remarks. Any questions about the Gram Schmidt process?

(Refer Slide Time: 18:41)



So, I make one other small remark, this is it relates back to some things we discussed in the previous class, actually two classes ago. So, this is about, its a note about orthogonal complement subspaces. So, recall that the range space of A is the orthogonal complement of null space, of the null space of A transpose. So, that means that A x equals b, if I take this system of linear equations, this has a solution if and only if. So, this will have a solution only if and only if be lies in the range space of A, if b is not in the rain space of A, then it will not have a solution.

So, if b lies in the range space of A it means that b must be orthogonal to any vector that lies in the null space of A transpose because the null space of A transpose is the orthogonal complement of the range space of A. So, that you know, so, to put that in words the way to write it is b transpose z must be equal to 0 for any vector belonging to the null space of any A transpose that is to say that for any z such that A transpose z equals 0. So, that is just a side remark I wanted to make about orthogonal complement subspaces.