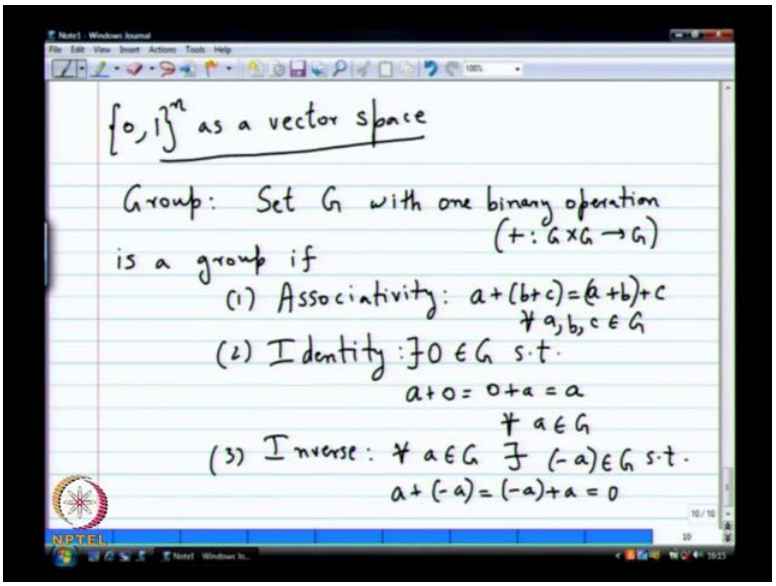
Coding Theory
Prof. Dr. Andrew Thangaraj
Department of Electronics and Communication Engineering
Indian Institute of Technology, Madras

Lecture - 2
Properties of Linear Block Codes

(Refer Slide Time: 00:27)



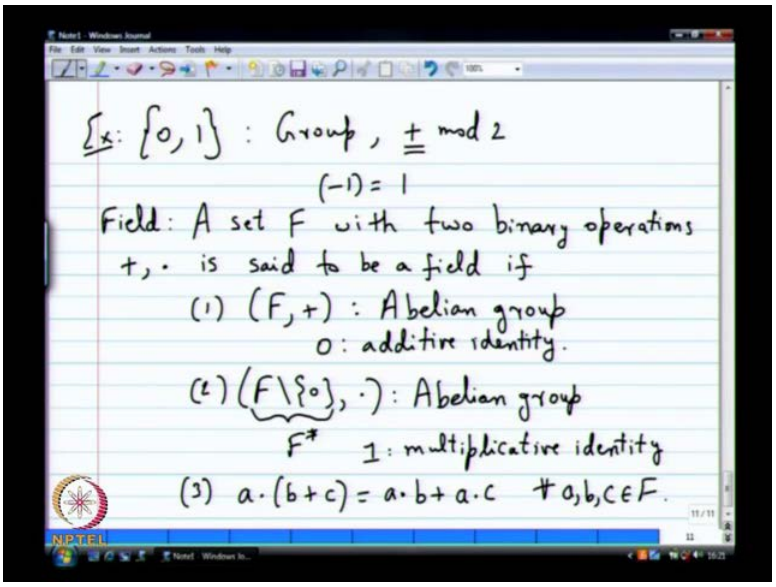
So, let us proceed. So let us see what I am going to do next is to formally talk about this 0, 1 n as a vector space. So, now we have seen informally what the 0, 1 n can be thought of this vector space by thinking about the linear code a as m times g extra, but it is good to formally see it, see this 0, 1 n as a vector space. Then, you will see some nice vector spaces arguments will come under casethese are coefficient elimination matrix operations and all that, but it is good to see as a vector space some formal definition. So, first thing you will need is a something about a group case, so group isthisset G case of finite case of always talk about finite, but it can be infinite also set G with one binary operation, what is a binary operation? A binary operation which we will denote as plus which is basically a mapping from G cross G to G in case what is the mapping from G cross G to G, take any two elements of G, it will do this operation and produce another element of G.

It is a binary operation which I will denote as plus said G with 1, 1 binary operation is a groupifsome three conditions is satisfied. So, how many conditions, three conditions, two

conditions, three conditions, first one would be associativity. Basically, what you need is a plus b plus c should be the same as a plus b plus c, so that is associatively, second property is you need an identity. There should be one special element in G which we will denote 0, there exist 0 in G such that what happens a plus 0 equals 0 plus a equals k.

So, this is for all a b c in G this is for alla and G, I am going a little bit fast because I am hoping you have seen this definition once before in your life. So, just a quick recap of what a group is any other property inverse, everybody seems to know group quite wellinversefor all a and G, there exists let us say what I call minus a in G such that a plus minus a equals minus a plus a equals. This is identity 0, so the group operation will usually take as plus, but it can be anything else. So, you can put any other shape,it is just a notation it does not change anything, so this is a group, so there are lots of examples I can give you for instance.

(Refer Slide Time: 03:50)



The most interesting examples for us is this 0, 1, this can be group with what operation addition modulo 2 what would be the identity 0inverse of 0 is inverse of 1 is 1 itself. So, that is the interesting part here, so minus 1 what is called minus 1 is the same as 1, so when you have addition modulo 2 is important is some examples for group. Of course, there are very more, I mean many more complicated examples of groupsand then the next thing we need is a field is a little bit more complicated.

So, let me just make sure I havesomewhere to be produced carefully a field is a set F with two binary operationswhich we will denote as plus and dot. So, plus is like the addition and dot is like the multiplication, so plus and dot this is set to be a field. Once again, we will think of this set as finite in most in throughout this course, but almost in other places, I mean there are there are also infinite fields. So, when is this set to be a field F comma plus should be what is called an Abelian group, so I forgot to define a Abelian group.

A group is said to be Abelian if the addition operation is commutative, the operation is commutative means group comes Abelian. So, s comma plus should be an Abelian group, so field should have addition, so any you should have a plus operation should be defined let us say 0 is the additive identity. So, basically the idea is group something where you can do one operation, so either addition or multiplication or some other operation a field is something where you can do both addition as well as multiplication consistently. So, that is the idea behind fields for instancethe integers or a group and the addition of course and a multiplication, there would not be a group why inverse will not exist.

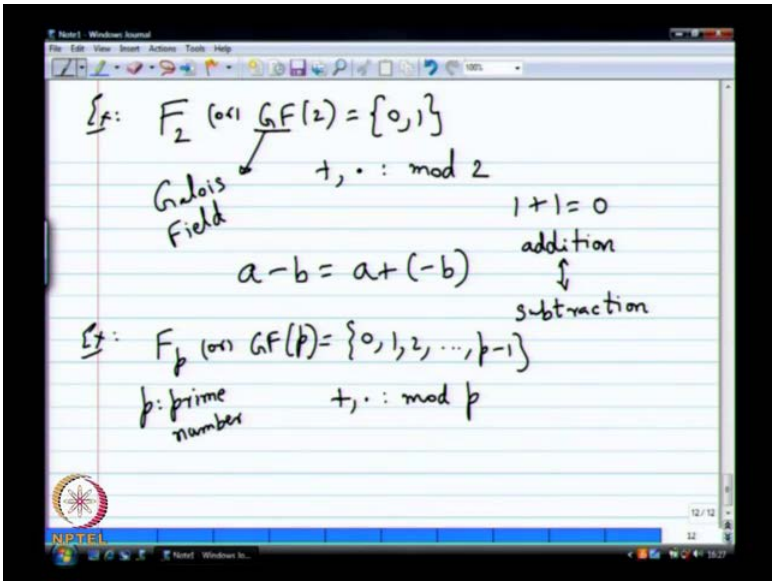
So, to all has no inverse in the other hand if you look at rational numbers, they are a group under addition under multiplication. They are not a quite a group why are they not quite a group under multiplication 0 will be a problem. So, you have to remove that 0, so it that is a second definition F minus the 0 which is the additive identitycomma dot should be a Abelian group again. So, that is the second condition for field, so this F minus this 0 is quickly denoted as F star k in a field, it is what is called the multiplicity a group of a field is F star it isit is the set F without the 0.

So, this is the second condition, this makes addition and multiplication possible in that field, so that is the basic idea. So, whenever you have a field or something the group the usually the problem is the inverse most things will naturally will be group you can extend them inverse is the problem. So, always that will show up in many cases the third property, so the third property usually you will denote by 1, the multiplicative identity. So, it is clear what I mean by multiplicative identity, so in this group F star comma dot there will be an identity element.

So, that identity element I will call as multiplicative identityare there any question at this point it is I hope it is clearthe additive identity and multiplicative identity have to be distinct for the field to be little bit meaning. So, otherwise it will not become meaningful at all, so 0 will be different from one that is the idea. So, the third point tells you how the addition and multiplication interact. So, that is the distributive property a dot b plus c should be equal to whata dot b plus b dot c. So, this is for alland people are so awake, so late in the evening have to be watchful c for all a b c in F. So, this is the distributive property, so you can see how the natural things that you take for granted in the rational field for instance are being abstracted are being made abstract in this definition.

So, you have field F which is which has an addition operation with respect to which it is in Abelian group. It has an additive identity 0 when you throw the additive identity 0 out of the set F what remains should be a multiplicative group the multiplicative identity 1. So, you can see it will show up even for the real numbers the same things hold the real field is significantly more complex. Anyway, you do not go anywhere near that, but this is an example, so in this codes will be mostly concerned with fields which are finite for instance the 0, 1 can be into a field I will show you how.

(Refer Slide Time: 10:04)



It is very simple, but never the less it is interesting to see how something so simple can be a field it is usually denoted F 2 or G F 2 when it is a field 0, 1 additive addition and multiplication are basically integer addition integer multiplication modular 2. So,

remember when I define a field what are the various things I should define? I should define the set and I should define the two operations plus and minus plus and dotaddition and multiplication.

So, I am defining the set first I am saying 0, 1, but I am calling it as F 2 or G F 2 when I say that implicitly, the operation is also definedwhen I say 0, 1 as F 2. What does it mean, I am talking about the set 0, 1 with addition and multiplication modular 2 that is what it means. So, that is the idea G F the expansion is Galois field, Galois is a very famous French mathematician who died very early, 32 or something. This is named after him, so you have to check all the field properties, but they are quite reveal to check it is easy to see that 0, 1 with plus mod 2 is a group.

So, you add it is very easy what about one itself is clearly multiplicative group, very little you can you can go wrong if you just have one. You just add you can do anything more to it. So, one curious thing in this field is see usually we think of also subtraction in a field.In a field, if you can add you can also subtract, how will you subtract how will you do a minus b in a field a minus b a plus minus bsee the additive inverse for b is defined so it is just the one operation plus you can also do subtraction.

Subtraction is just an artificial operation, so it is different way of calling addition, so you just do a plus minus b you get subtraction. So, what happens in this G F 2 what is the subtraction minus b is same as plus b, so that is the curious thing about this field addition and subtraction are the exact same operations. So, any minus you have can be made happily into a plus in any long drawn algebraic manipulation that you do with multiplication and all that.

Any simplification you want to do if you know your field is G F 2, then all minuses become plus and 1 plus 1, and then is equal to what 0. So, that is the curious thing, so addition and subtraction become equal, so if you used to typing documents with spell check suddenly when you write something it does not look right and I have spelled addition correctly. So, you are my spell check, so if you tell me that I make a mistake, so this is curious thing about a field about this field in particular.

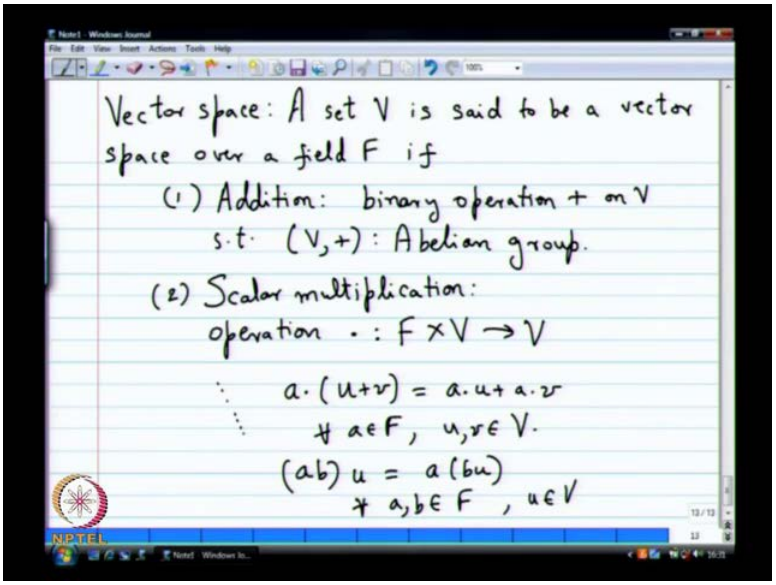
So, the other fields I will give another example I will give you a very generic example of what is called F p or again G F p 0, 1, 2 so on till p minus 1 and what is p, p should be a prime number. So, you should take it as a prime number clearly 2 is the special case, so

take p equal to 2, 2 is a prime number. So, that is one special case and addition and multiplication once again you can do modulo p, so one can show that this is a field that requires some work.

So, it is not very trivial to show that this is a fieldwhat is very trivial what are among the three properties which property is very trivial plus is very trivial. So, you add two things modulo p, I will obviously from 0 to p minus 1, so addition is its very trivial multiplication. So, it is a bit closure is, but inverse is bit non trivial, so you have to do some work to show that the inverse exists possible. So, this is a field, so clearly if p is not two addition and subtraction are different operations minus 1 and plus 1 are not the same in this field in general for p equals 2 it becomes something really simple.

There are also other fields, but we will see them later on, I do not want to go into them in more detail, we will come back and may be the this field also later on, so I just want to quickly show this, but for G of 2. It is very simple, so it is very easy to show or see that it is a field the addition and multiplication are very trivial. Multiplication is really nothing you multiply by 0 what happens get 0 multiply by 1, you get the same thing, so it is a very trivial multiplication.

(Refer Slide Time: 15:18)



So, the next thing vector spaces, it is you can askits trivial may be it is trivial require some work possible not saying it is possible. So, it is not a require some work vector space, so let us see so how do I define a vector space set V is said to be a vector space

over a field F. So, you need a lot of condition, so there should be first of all an additive operation in V, so you need v has there should be two operation the first one is addition operation. You should be able to add to vectors, there should be a binary operation plus on v such that v comma plus is an Abelian group.

So, you need to be a Abelian group, you could have started out by saying V is an Abelian group, for instance I just wanted to put it down another separate condition. So, you need to be able to add 2 vectors any 2 elements you have in a vector space, you should be able to add. So, that is the first condition which makes it addition the other thing is scalar multiplication that is where the field F comes in. So far, the field F has not come in the field F will come in a scalar multiplication, you need an operation which is dot or it may even be scripted.

So, never I said dot the product operation we usually do not write when a dot b, we right it as simply a b, so it is either the backwards operation or the dot dot which is a mapping from F cross V to V. So, that is the definition of a scalar multiplication what is F cross v you have a pair one belonging to F, the other belonging to the vector space and it takes you to the vector space. So, that is the scalar multiplication, so it is need an addition under which v becomes an Abelian group.

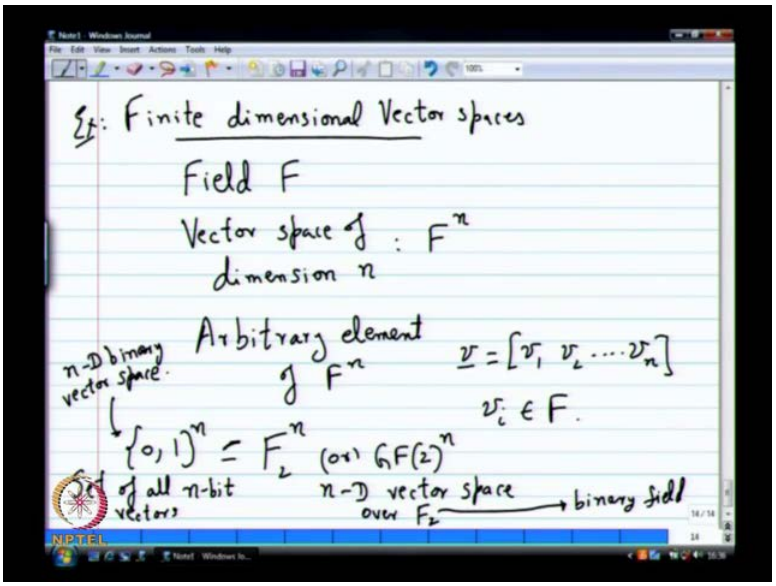
So, addition is particularly closed then it the vector space also needs a scalar multiplication, so if addition, there exist a scalar multiplication etcetera there should be a scalar multiplication operation which goes from F cross V to V. So, basically we usually think of this as a scalar F multiplying the vector V giving you another vector V, so these are the two conditions, then of course, there is the distributive condition.

So, if you have dot distributing over the plus and all that, so I am going to skip some of the other conditions. There are other few others, so it needs some work to write it down, for instance if you havea dot u plus v what should happen it should be the same a dot u plus a dot V, so for a in for a in F and u comma V in V. So, such conditions will be everything this is only other condition, there is also the other conditions for instance a btimes u should be the same as a times b u.

So, all the usual things that you associates, so this is for all a comma b and F and u in V, so there are several other simple things like what you expect for distribution and all that in scalar multiplication which you might have known. It is assumed in the definition, so

it is crucial you should remember there are two operations in anyvector space first one is an addition which makes it an Abelian group. The second one is a scalar multiplication which allows for multiplying any vector by a scalar from some field, so that is where the field will explicitly enter.

(Refer Slide Time: 20:18)



So, the most standard Example, I usually what I called finite dimensional vector space, so this finite dimensional vector space are the easiest and nicest examples and we will we willmore or less fully study only finite dimensional vector space. So, there is lot of theory behind this, you can you have to you can start carefully start from the axiom and show a lot of things, but it turns out all finite dimensional vector spaces have a very simple form, what is that form?

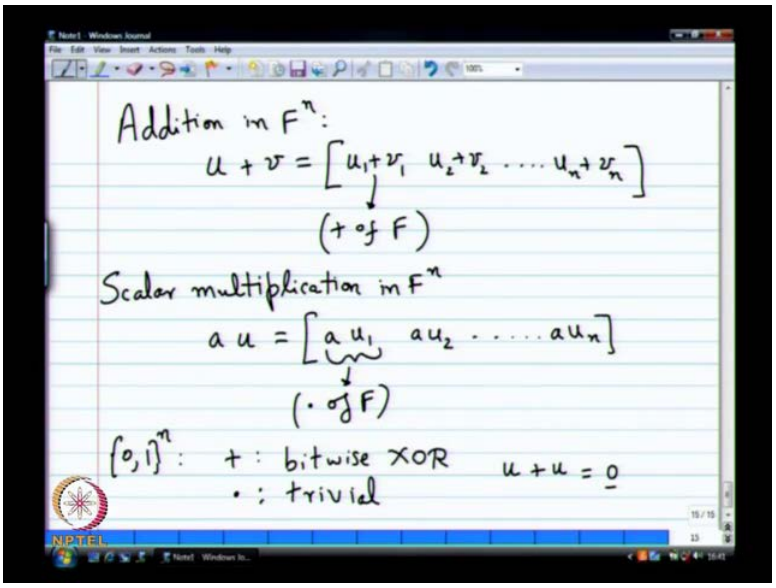
There will be one dimension number called which is usually denoted, let us say n and then there will be a field F and all finite dimensional vector space is will be of the form F n. So, it is this is form, so all finite dimensional vector spaces will have a field, you have a field F and the vector space of dimension n if you denoted it is simply F n what is F n, what is this notation F n. I used it before 0, 1 n or something, so basically you take n elements from F. So, an arbitrary element F n will look like what will be triple, so that is the basic idea, so it will be if you if you denote it, let me see usually you can denote it as let us say V. It will be triple V 1 V 2 v n and each v, I will belong to F, so you might be used to thinking of vector spaces like this.

So, immediately when I said a vector space, you might have thought about this, but this is not the way it is strictly defined, you have to define it automatically as some Abelian group with a scalar multiplication operation. There are there are vector spaces which may not look like this, so it looks strange also, but if you restrict to something called finite dimension which you can prove very carefully. You can show most vector spaces of interest at least in error control coding will have these form.

So, if you have dimension n it will be F n F is a field and all you have to do worry about is it triple over that field, so this gives us tools to view 0, 1 n as a vector space. I wanted to see 0, 1 n as a vector space, now you see how is may be quite obvious, how you can see 0, 1 n as a vector space. So, how would I do it 0, 1 n simply what is the same, sorry is a same as F 2 n or G F 2, so the set of all n bit vectors is the n dimensional vector space over the binary field F two G F 2. So, this is the set of all, so let me be careful here there are two things here on the left hand side, you have set of alln bit vectors which can be thought of as the n dimensional vector space over F 2.

So, this F 2 is also called the binary field, so this 0, 1 n is also called the n d binary vector space, now let us quickly check that this is indeed a vector space. In case if you have any questions, let me know is there any question at this point, it seems more or less. So, now I have to tell you how to do addition and scalar multiplication in this vector space, so it is quite easy might have you might have definitely seen it, but any way I will I will tell.

(Refer Slide Time: 24:58)



I will write it down addition in F n is what how will you add two things vector u plus vector v is simply u 1 plus v 1 u 2 plus v 2 so on till u n plus v n and what is this plus, the plus of F. Remember, I mean it is straight anything to say modulo p when I never said F is F p F is general abstract field, it has an addition operation defined on it. I take two vectors u and v and add them from F n, I can do component wise addition. This is what you do in r n, for instance you do the real addition component wise what is scalar multiplication, what is this operation the dot of F.

So, all the linear algebra that you might have learnt for instance collision elimination particularly things about linear independence what is linearly dependent. Basis vectors sub spaces everything applies to what an arbitrary vector space defined over a field any field F. So, you might have learnt only over the reals and complexes you might have seen a lot of exercises only using real and complexes, but all the theory of dimension any independence spaces.

How do you how do you collision elimination row reduced column form everything applies to an arbitrary vector space defined over a finite field. The only things you need are the axioms, the only things you use again and again are the axioms, and nothing else is needed. So, this is sometimes of surprised to people if you not seen it before, buthopefully it is you will accept it like I said we do not time to go over this in too much detail. Hopefully, it is clear, so maybe I should mention a few of the few of the things

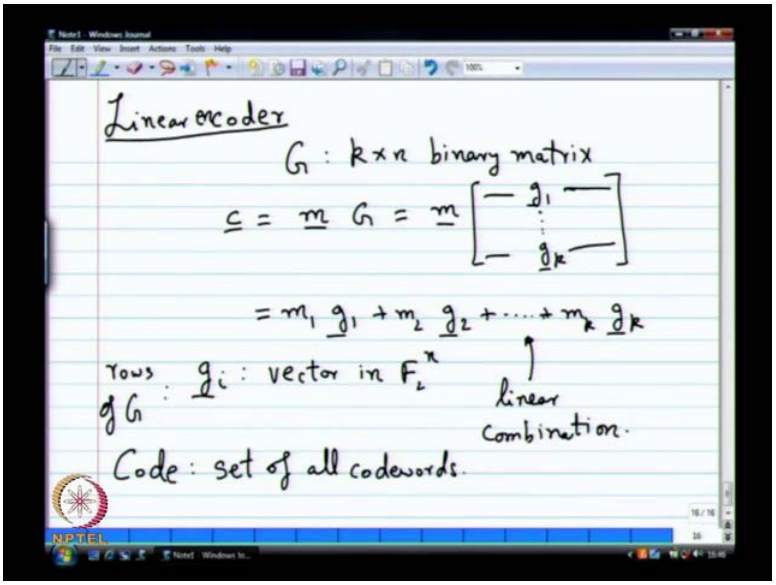
which are important, but I will mention when as we go along, I think at this point I should not continue it.

So, a particularly in the binary vector space 0, 1 n addition of two vectors will be what addition is you can think of it as component wise mod 2 addition or if you want a very much very much digital systems terminology it is bit wise x. So, that is what addition is, it is a simply a bit wise x orwhat is multiplication it is actually trivial multiplication. Scalar multiplication is trivial in what way it is trivial, if you multiply by 0 what you get 0 if you multiply by one the same thing.

So, the multiplication is very trivial in this G F 2 vector space in particular if you do this, what will happen if I take one vector u, and then added to itself what will happen it is 0. So, that is something you will get in G F 2 which is non zero vector added to itself, you get 0, this can happen in this vector space.

So, it is a bit curious vector space it is not might be surprising to you, so let me move ahead. So, we wanted to see 0, 1 n as a vector space because of because of some reasons, we will come to that soon enough. So, that is hopefully clear why 0, 1 n is vector space, why it is set of all n bit vectors can be seen as a vector space, now let us go back to our linear code definition.

(Refer Slide Time: 29:46)



Linear encoder definition, sorry what was the definition of a linear encoder, we have g which is a k cross n binary matrix which was the generator matrix in systematic form, butwhatever, then the code word was formed times G. So, what did I write down this operation as if you think of the rows of G as G 1 through g k, now may be the linear term linear combination term will make sense. So, you can think of each of these g 1 through g k's as vectors in 0, 1 n, so then what you do is simply linear combination.

Now, I am going to say these g i 's this is a vector in F 2 n, now I am going to think of g i as a vector in F 2 n and clearly what is happening here is a linear combination. So, that is a nice motivation and interpretation in the 0, 1 n vector space, so this brings back and hopefully tries up something. So, we will see this in more detail as we go along, so this g i 's which are the rows of G, remember these are rows of G. These are all vectors in F 2 n, the binary vectors is and when I form a code word, what am I doing? I am taking linear combinations of a set of vectors from a vector space all possible linear combinations.

So, that is the strange thing for people to get used to, so for instance, so for you might have dealt with only R nand when you say all possible linear combinations in R n. How many possible linear combinations will there be infinite remaining, but in F 2 n what happens you have only finite remaining linear combinations in F 2 n. If I take k vectors, I form all possible linear combinations, how many possible linear combinations are there only 1 power k, but go back the definition of what is called the sub space what is a sub space?

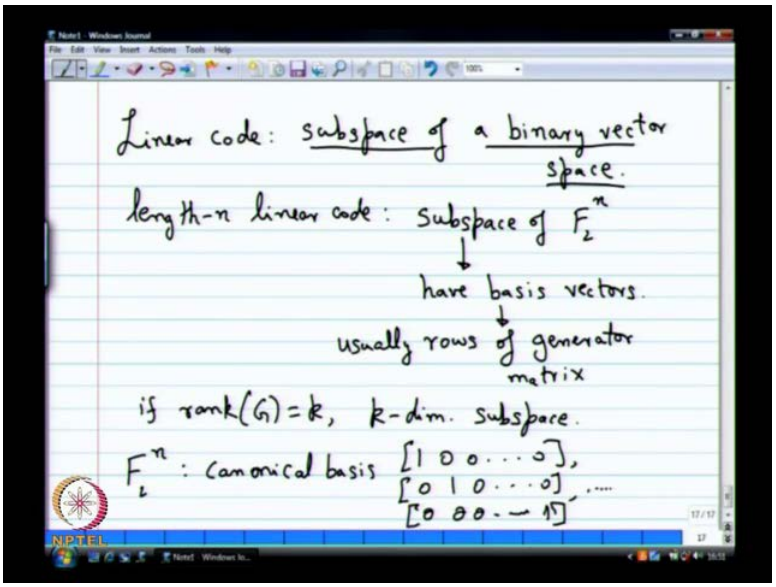
You take a subsidiary; you take a set of vectors and form all possible linear combinations of them you get what is called a sub spacethe same definition holds in F 2 n also. Nothing changes there, even though you have only a finite number that is all the linear combination. Now, you have may be striking new interpretations for this m times g operation what is m times g actually doing. It is defining a sub space of 0, 1, so that is the abstract way of viewing this operation. So far, we just viewed it in a very functional way we started with the vectors I said you define this vectors and take all possible linear combinations to get code words the abstract way to view this is these vectors g.

I define a or span a sub space of the binary vector space 0, 1 n, so that is another way of defining the linear encoder or the linear code. So, by the way one more definition which

I did not make before is what is a code is set of all code words. So, this is the code, so simple definition I guess encoded is more crucial set of all code words is the code. So, if I define a linear encoder, the code that I get is from it is in fact a sub space of the binary vector space F 2 n what dimensional sub spaceless than or equal to k when is it equal to kwhen g i's are linearly independent.

Now, I can comfortably use the word linearly independent because I know it is a vector space, I know what that I know the linear independence is properly defined is not anything difficult for me to define. So, if you are not used to reading a board vector space is an abstract way this might be a little bit surprising. Anything you read about r n pretty much holds in F 2 n also linear independence is the same how would you define linear independence all 0's. So, if you have you have an abstract way of defining linear independence go back and check that you can define in that way. So, all those things can be done, you can do quick Gaussian elimination to figure out how the whole everything works, so everything can be done without too much struggle.

(Refer Slide Time: 35:35)



Like I said, I am not going to back and review in this in too much detail, so let me just summarize what I said in the next slide. So, a linear code is usually defined as a sub space of the binary vectors space of a binary vector space. So, if I want a length n linear codewhat should I be looking forI should be looking for a sub space of what of F 2 n. So, how can I define a sub space of F 2 n usually it is defined by a set of basis vectors, so

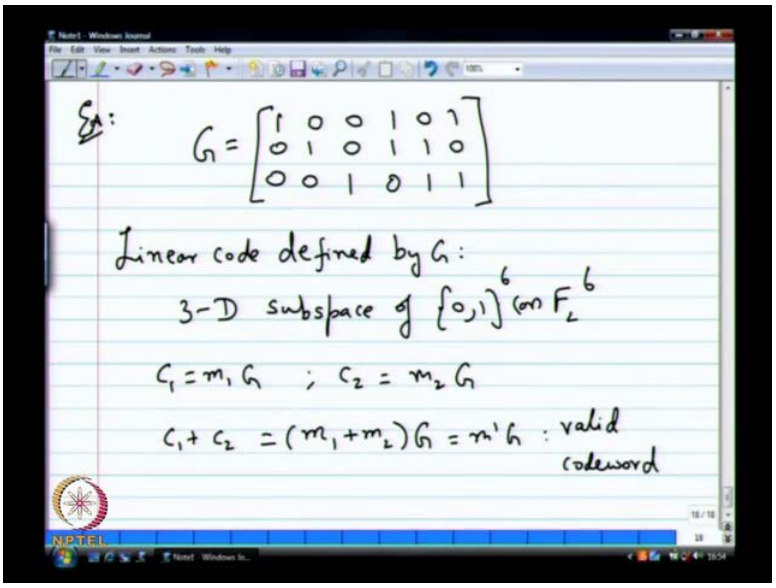
that is where the basis vectors will come, so sub spaces have basis vectors. So, these basis vectors are basically what so far we been thinking of the generated matrix. So, this basis vectors are usually the rows of generator matrix, so of course you have to assume full rank here usually a rows of generator matrix.

So, that is the way you have to think about the linear code in terms of a vector space,I find the code words as m times g, but as a sub space of F 2 n the rows of g are defining a space. Usually, you also called the row space of G, so this linear code that I am defining with the generator matrix g is in fact a sub space of F 2 n and if the rank of g is equal to k, and then I have a k dimensional sub space of F 2. So, if rank of g is equal to k you have a k dimension, so this view is very critical, it gives you a lot of intuition on top of what you might get from the simple matrix multiplication definition.

So, m times G, but you get a sub space of a sub space is as better structure, so next question I am going to ask is quiet easy, but let us get it out of the way. So, if I think F 2 n, so simple questions, so fix some ideas, if you think of F 2 n it is n dimensional vector space over F 2 what is the basis for F 2 n. So, you usually take this canonical basis what is called a canonical basis for F 2 n, so we will always use this canonical basiswhich is basically what is called e i.

So, one followed by bunch of 0's and then you take 0, 1 followed by 0's and finally you will end up with what 0, 0 followed by 1. So, my writing is going very bad when I come to the bottom of the tablet, hopefully it is clear. So, this is a canonical basis for the entire vector space, so for the sub space you will have different basis which is g 1 g 2 g k etcetera. So, let us go back to thisexample that we had may be see some sub space ideas.

(Refer Slide Time: 39:47)



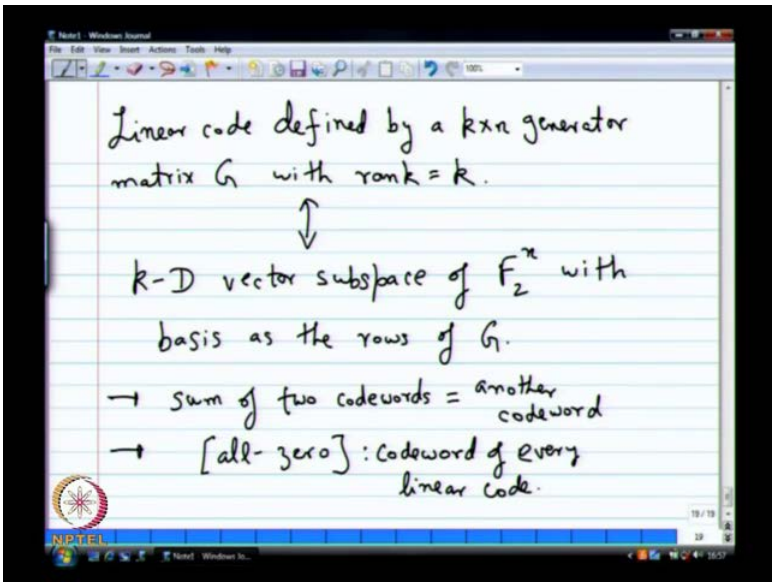
So, we had g to be equal to 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, this was one of our examples. So, the linear code defined by Gis what a linear code, so this is a 3 dimensional sub space of 0 1 what 6 or F 2 6, so F 2 six is a little bit better because when you know more about the little bit more percent. So, it is a 3 dimensional sub space of F 2 6andso there are there are several properties, so once you say sub space what happens if I take 2 vectors from a sub space and add them you will still remain in the same sub space. What happens if I take a vector in a sub space and multiply with the scalar, it remains in the same sub space.

So, those properties which are easy to prove simply with c equals m G also you can prove it with that. Also, if you define the code as simply c equals m times G, you can show that m 1 G plus m 2 G is the same as m 1 plus m 2 into G. What happens if you have c 1 equal s m 1 G and thenc 2 equals m 2 G what is c 1 plus c 2 m 1 G plus m 2 G which is m 1 plus m 2 times G this is from basic matrix co operations.

So, what is m 1 plus m 2 into G, so this is some prime into G which is also a valid code word, so just by defining it as a m times G, I can prove this properties, but once you know it is a sub space intuitively all these things are much more clear. It is a close sub space, so add two vectors you should get another vector in the same sub space. So, if you multiply a vector by a scalar, you should get another vector in the same sub space, so this linear code defined by G has a set of code words. I can also think of each code word as a

vector in F 2 n and all these vectors together make a k dimensional sub space, so let me summarize all these things in the next slide.

(Refer Slide Time: 43:00)



So, if you have a linear code defined by a k by n generator matrix G with rank, it is a equal to k this is equivalent to following description have a k dimensional vector sub space of F 2 n with basis as what the rows of G. So, in particular what happens is some of two code words, two or more of course, if I said two code words, then 2 or more also same thing applies equals another code word. So, it is closed under addition that is the linear property and then well in F 2 the scalar multiplication is really trivial.

So, there is nothing really to do if you multiply by 0 what happens you get 0 multiplied by one you get the code word itself. So, I am going to write that down, but another interesting thing is the all 0 vector, what will be a code word,this is code word of every linear code. So, even with the m times g definition is obvious, I mean how you get the all 0 code wordtake the entire 0 message. So, whatever you define is g, you will get the all 0 code, but from a sub space point of view, also it is pleasing every sub space will definitely have the all 0 vector in it.

So, it will pass through the all 0 vectors you have, so this gives, I think is important totake n no it is an important view of a linear code. So, from an implementation point of view at the n coder, the view that is interesting is m time's g may be g in systematic form. You produce a message and then compute a sub set of parities to get what you

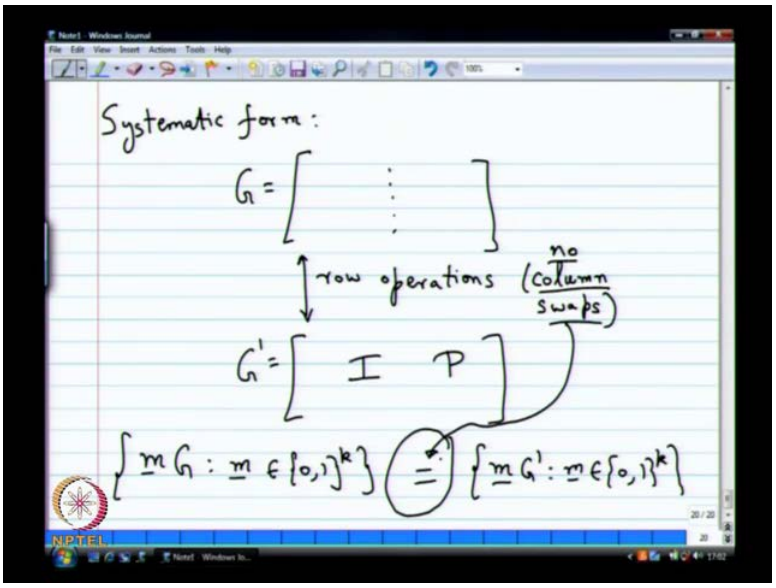
want, what is interesting from an abstract theoretical point of view which has a lot of practical utilities soon enough. Particularly, when we moved to the decoder, I will talk more about it is this vector space view; you view the set of all n bitvectorsn bit numbers.

If you want anything as a vector space which is F 2 nthe addition is bit wise x or multiplication is it trivial multiplication the scalar field is F 201. When you view it that way, then there is m times G the rows of g have a special meaning, they in fact span the rows span a space which is sub space and that becomes your code. So, code itself is a collection of code words which is actually thought of as a vector sub space. All these things are vectors in this vector space you havevector sub space of a particular dimension.

So, what happens when the rank of g is less than k, sorry you have a lower dimensional vector spaceit may be k minus 1 or k minus 2 or whateverhow do you find the basis in that case. In general, you do row deduction, eventually whatever linearly dependent will become 0, that is one thing and then now you see how see one more thing you know is when you do row deduction or Gaussian elimination, what does not change the span. All the vectors do not change, you take a row, a matrix and do row operations and do row deduction what does not change is the row space, the row space remains the same.

Now, you see the statement I made that any code can be made into systematic code systematic form is very obvious, why is it obvious? So, it is a linear vector space, I can always reduce into row reduced the canola form and I will get I P. So, you do the Gaussian elimination reduce it to systematic form, so we saw an example where something in nonsystematic form can be reduced to a systematic form. You might wonder if m G defined with the non systematic form is the same as m G defined with the systematic form. So, it turns out the code is the same, so we will see that is very easily seen from these kind of vector space viewpoints. So, I think I have two minutes left, so let me see let me rate the systematic form a little bit more.

(Refer Slide Time: 48:26)

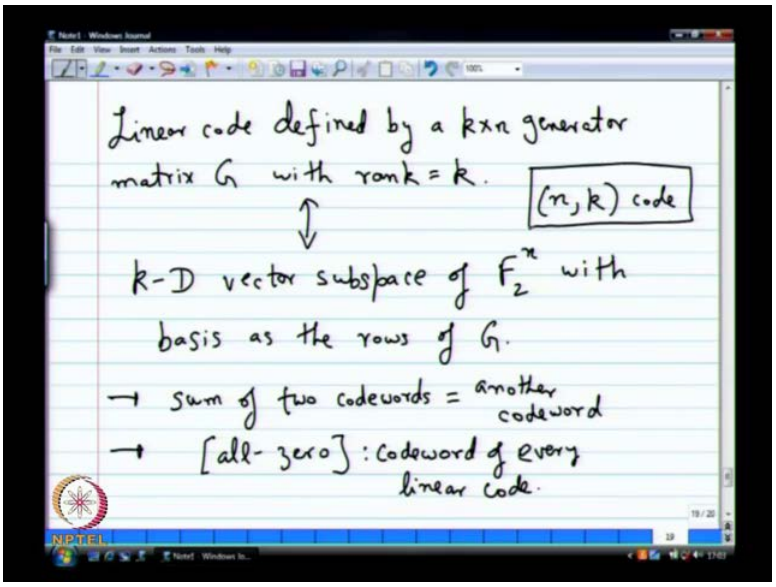


Let us go back to this systematic formwhat are we doing in this systematic form you start with the g which we say arbitraryvectors you do a series of row operations. So, let us see we also allow columns swap, so this is a bit ofcrumbling thing in linear algebra you would never allow a column swaps. I mean it is not allowed, so we will allow column swaps why do we allow column swaps first from an operational point of view. It is the same you know may be you have some column of wearing here or later only thing it changes is the sequences of which the bits it go out the code word from operational point of view for us it is equivalent. So, column swaps will allow if you do this you get systematic form which isI P, so once you do this, you can now define m G1 set for all m in 0, 1 k 2 k and then m G prime for all m n 01 k.

So, let us say, I mean for simplicity we will assume that no column swaps, we need it, so for now just to keep it simple we will assume no columns swaps needed column swaps are needed. You will have to undo the column swaps here for this statement to be true, so what do you think this is the sign that I can put in the middle here. You can in general say definitely equivalent is there are no column swaps what can I say equal. So, it will be equal under this condition if there are columns swaps what is the only thing that can happen the order of the bridges in which they appear will changeyou can do a permutation and change it to equal side.

So, that is the meaning of saying any code any linear code can be put into systematic form, so you get the same is that. So, you do all these you know row operations are just invertible multiplications, so you can undo every single row operation it is no problem. So, I guess this is not the main, last point this is the main thing that you should take away from these two lectures.

(Refer Slide Time: 51:11)



So, what is a linear code a linear code is a vector sub space of a binary vector space n dimensional binary vector space. So, such a code is also denoted as an n k code, so this is a very popular notation n comma k code. When I say an n comma k linear code what I mean is a k dimensional sub space of F 2 n how do I specify that sub space, I have to specify k linearly independent vectors as basis vectors for my sub spaces. So, what will see in a next class is there is an equivalent way of defining sub space you might have learnt in basic 3 d algebra.

So, if you do not if you want to define a plane you either define two vectors in a plane or what do you do you define a normal. So, what is the property of the normal is octagonal to every vector, so to define orthogonally what do you need dot product. So, we have not talked about a dot product, so what will do next is define a dot product and then we will see how to define vector spaces or vector sub spaces using normal vectors. So, to speak they are called something else dual vectors, we will dual vectors to define vector spaces

and that is a very powerful tool, so more than a generator matrix you should understand the dual space that is it.

Thanks.