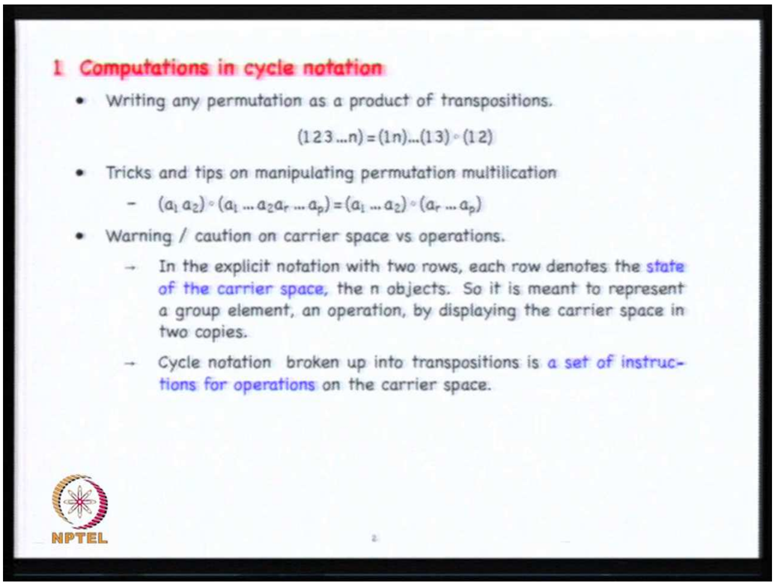
Theory of Group for Physics Applications

Prof. Urjit A. Yajnik
Department of Physics
Indian Institute of Technology, Bombay

Lecture - 11
Cycle Structures & Classification - I

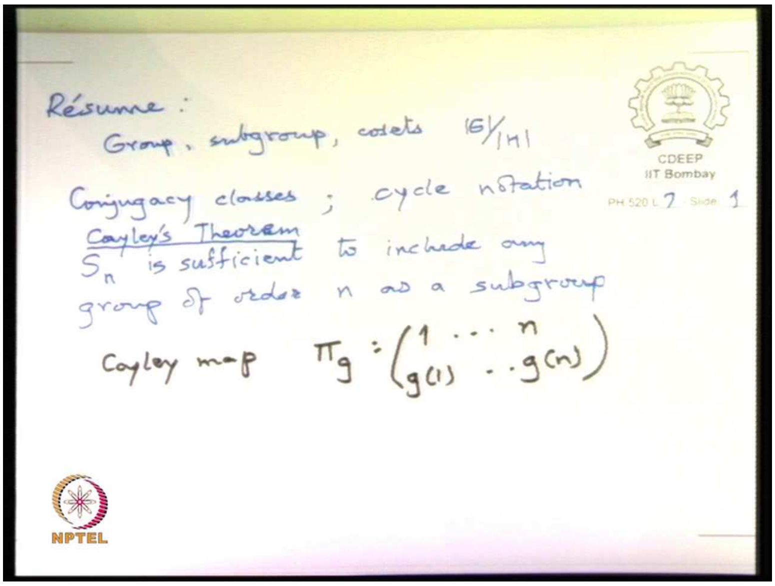
Well, I can tell you that if you liked it so far it is going to get more and more interesting, ok. It is really quite magical what comes out of group theory in the end. For that we have to understand this idea of conjugacy classes very well and then from next time we will start representations and that is where the all the power of it comes because.

(Refer Slide Time: 00:37)



In fact, chemists use the group theory all the time, but if you use chemistry books or chemistry lectures presentations they never teach any of this background. There is shortcut rules for figuring out the character classes of molecules and they just fill out tables, and then they quickly can see selection rules or multiplicities or degeneracies of state transitions and so on. But we being physicist you know we will labour through all the reasoning why those things work, but of course, not labour as much as mathematician. So, let us just try to gather all the things that we have been studying.

(Refer Slide Time: 01:26)

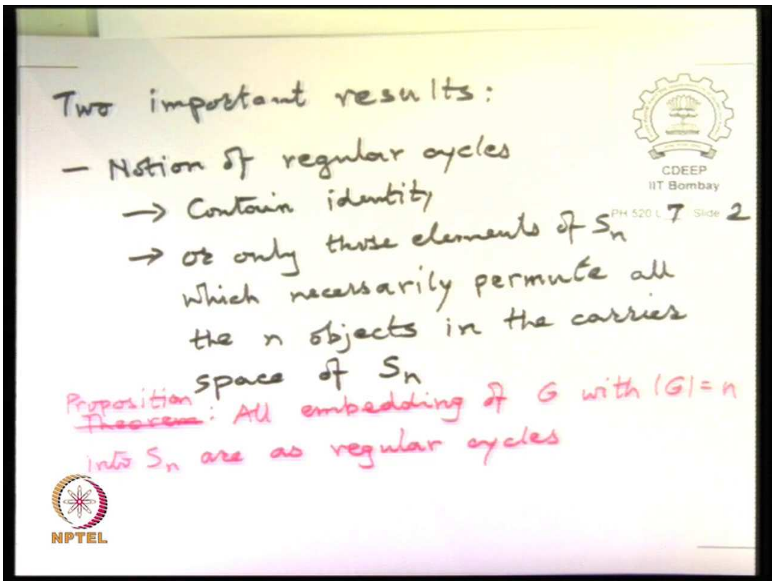


So, the early parts where of course, group subgroup and coset spaces and Lagrange's theorem has to be an integer; so now we are on to the idea of conjugacy classes and also the representation in cycle notation. So, currently what we are going through its conjugacy classes and of course, we are adopting cycle notation alongside.

So, today we will see some comprehensive picture of all of group theory in some sense ok, emerging from these things. But just to complete this ideas the thing to remember is the overarching theorem Cayley's theorem that S_n is sufficient to embed any group we know, ok. So, sufficient to include any group of order n as a subgroup and this is of course, Cayley's theorem.

And the proof these are called constructive proof where the proof is not just a clever argument, but it actually demonstrates by in detail why the thing works which is the I do not know whether it is called in literature like this, but let us call it Cayley map. The map is that for corresponding to any π_g where, $g \in G$ there is the permutation g which can be written out as 1 to n and then the fate of g(1)…. g(n) ok. What is the effect of the group operation on g(1) or group operation on g(n)? We will be shifting these definitions for convenience, but they will be defined precise clearly in the context in with they are being used. So, we have this map and we saw that this leads to some interesting results two of the important results.

(Refer Slide Time: 04:42)



One is that the groups G through this embedding are realized as groups consisting of regular cycles ok. So, there is notion of regular cycle which contain identity or only those elements of S_n which necessarily change everything shift all the elements permute all the elements. And let me remind you here a little bit permute all the n objects in the carrier space of S_n, ok.

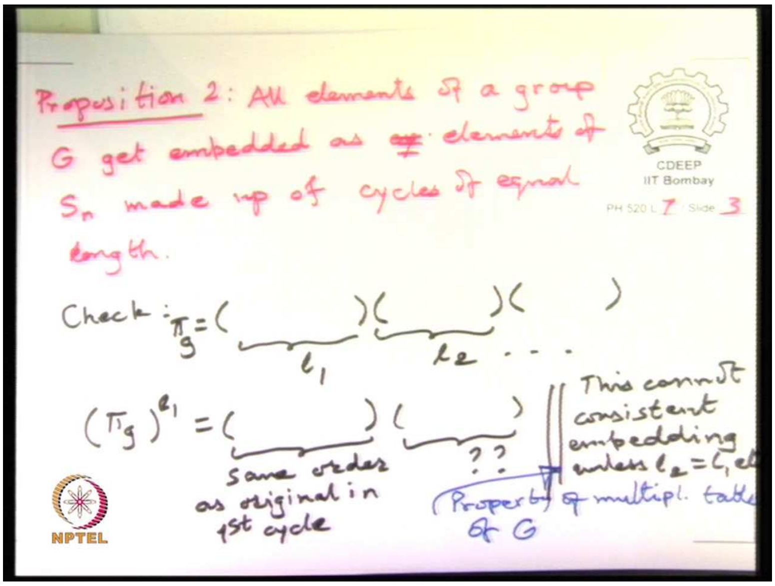
So, the regular cycles are those that will necessarily change everything the, so what is carrier space of S_n it is any said that contains some n objects. And it is called carrier space because by operations on this space we can realize the elements of the group S_n. So, these two, we have to add identity as a special case because somebody legally will say identity does not touch anything else anything at all. So, yes, so that is the only one that is an exception, but all everything else is necessarily something that shifts everything.

Now, one can check that, and this is where there was some ambiguity last time partly it arises because this book Hammermesh which contains everything, but it is scattered and not punctuated very well you know the concept where the definitions begin and where theorems begin and end things are not punctuated very well it is like a big rush of ideas. So, I got a little confused there, but if I consider what Hammermesh book says, then basically we could have directly stated the following theorem that all embeddings of G with |G| = n into S n are as regular cycles.

Need not call a theorem this is actually more like a proposition it is a fact which is sort of obvious from construction. So, this fact follows simply from the fact that the group multiplication is unique. So, that the multiplication table always has all the elements being permuted the n elements and therefore, when you embed them in the big permutation group they necessarily content cycles of this form ok, that is the statement. So, they are all necessarily regular cycles or identity. So, this is one general result.

The second result is that when you consider this embedding embed the elements represented in this way are necessarily ones with equal number of elements cycles of equal lengths. So, preposition to, so called this preposition 1 is that of a group G get embedded as cycles as elements of S_n made up of cycles of equal length.

(Refer Slide Time: 09:29)



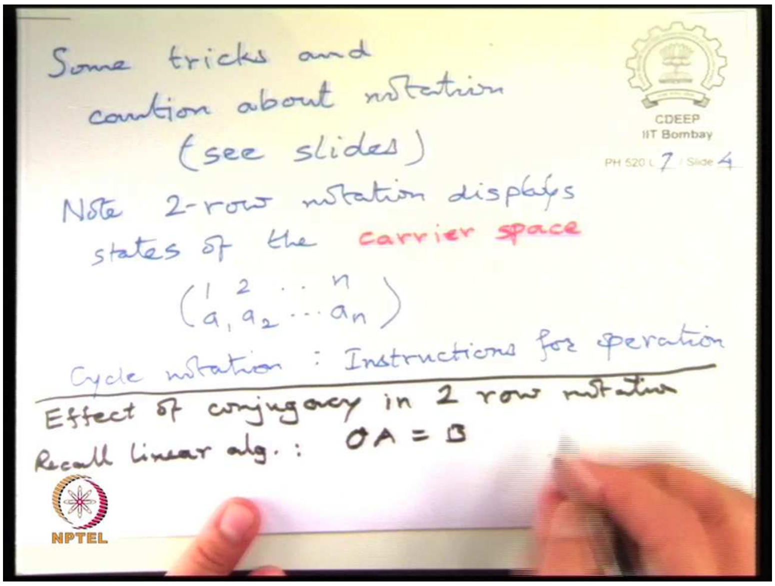
And this fact follows from the construction from preposition 1 that such embedding has to be necessarily regular. So, I will just say check rather than proof the check is that you will have some number of cycles representing G, π_g let us say, and let us say this cycle has length l_1 and this cycle has length l_2 and so on. Then we can see that $(\pi_g)^{l_1}$ will result in exact same order as original in this slot the first slot in first cycle can you see it.

Whereas here we do not know what happens if l_2 and l_1 are not compatible, if l_2 is a factor is a if l_2 is a much smaller number and the factor than it may be ok, but if l_2 is comparable and larger and not having a integer fraction which is l_1 then this l_1 cycling will leave l_2 in some unusual configuration which is not same as original. But we already concluded that if one piece is same then all others have to be same because otherwise you will have two different group elements having same for cycle which is also not possible because it means that their multi in their multiplication table the first l_1 entries are same this cannot happen for two different group elements, ok.

So, this means cannot be consistent embedding unless l_2 equal to l_1 etcetera due to the properties of multiplication table, ok. So, this consistency requirement comes from the property of the multiplication table from which you constructed this embedding. So, these two facts are interesting facts for embedding of n → S_n.

Now, let us go over some caution about the cycle notation and how to manipulate cycles.

(Refer Slide Time: 14:01)



So, here what I have tried to identify is the fact that one thing is that if you have a permutation which is an n cycle, this n need not be size of the group it could be any as could have written r ok, but if you have (1 2 3…. r) an r cycle if you want to write it up as a product of transpositions then it turns out to be (1...n) followed by (1….n-1), so transposition of (1 3) and transposition of (1 2).

So, you can check that you can read this from right to left because we want this to operate on something that it is this sequence of transpositions that makes up the cycle. In other words there are (n-1) transpositions in an n cycle, ok. So, whether a cycle is odd or even is decided by this list and it contains (n-1) the oddness or evenness has to do with whether (n-1) is odd or even, ok. So, a 2 cycle which is transposition itself is necessarily odd right because it makes one change. So, an even numbered two you remember it like this if it is a transposition with only two it actually results in an odd cycle. If you have a usual cyclic permutation as we say in physics (1 2 3) → (2 3 1) then that actually involves two permutations and therefore, it two transpositions and therefore, it is even. So, and that is a odd number, there are odd number of entries in that. So, there is (n-1) is the number that decides whether this n cycle is odd or even.

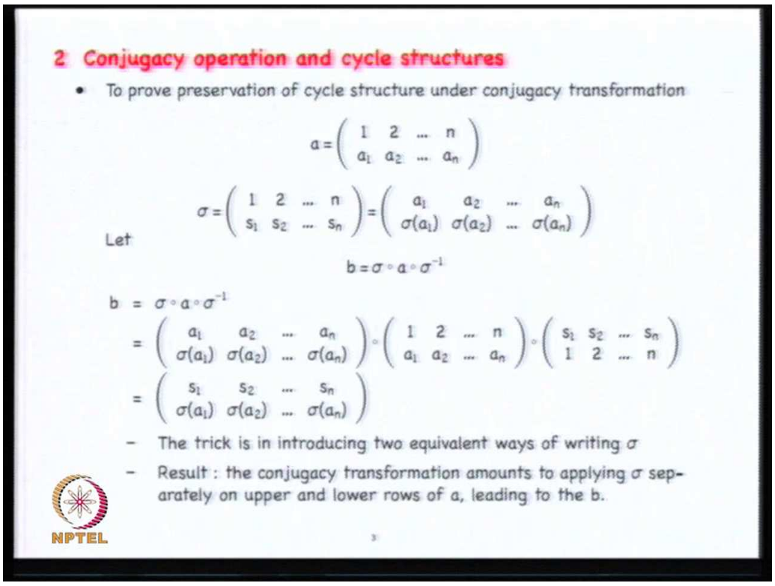
The other this is an interesting trick that I read in Hammermesh I do not know right now how to illustrate or I will probably not given a problem based on that, but it is interesting to know this that suppose I have one cycle which is (a_1 a_2) then some other element follows a_r and goes up to a_p. So, this is a cycle in the sense that a_1 goes to the next one next one up to a_2, then a_2 goes to a_r and then continues up to a_p. Suppose you take this cycle, but hit it on the left within (a_1 a_2) transposition ok, subsequent to doing the cycle you do (a_1 a_2) transposition. What does this do it? Actually breaks up this cycle because this instruction says that a_1 and a_2 are now exchanged.

Now, here what was happening was that a_2 was going into a_r, but now it that is no longer the case because a_2 is now suppose to go to a_1 so that account kind of closes this cycle. So, this becomes a cycle by itself and a_r has nothing originating it. So, it begins a new cycle by itself. So, there is this kind of a rule that operates for if you manipulate in terms of cycles then this is one way of that it will enter and in reading all of these I would say this general warning carrier space versus operations. So, the explicit notation with two rows that we have actually each row denotes state of the carrier space displays states of the carrier space right because it shows $\begin{bmatrix} 1 & 2 & .. & .. & n \\ a_1 & a_2 & .. & .. & a_n \end{bmatrix}$. So, this is the initial configuration of the carrier space and after the operation of this it will become this that is what it is showing.

On the other hand the cycle notation directly contains instructions for operations, for the group operation. So, this is worth remembering if you ever get confused about what is happening which at least I got for some time. So, I have written here in the explicit notation of two rows each rows denote the state of the cycle space the n objects are listed out, it is meant to represent a group element and operation by displaying the carrier space in the two copies of the carrier space.

On the other hand the cycle notation is especially the assertion I made depends on you can always break it up into this transpositions and then that can be seen as a set of instructions for what to do on carrier space, transpose this, transpose this, transpose this pair etcetera ok. So, that is what the difference between the two is anyway. So, these are general comments for when you to keep in mind when you are trying to understand or calculate.

(Refer Slide Time: 20:14)



Now, we come to an interesting statement about conjugation of this conjugation operation. We did one example and I tried to see if there is any other reasoning, but turns out there is the kind of problem that I gave in which the a is, so looking look at this equation, $b = \sigma a \sigma^{-1}$ this is what we mean by conjugacy transformation.

The problem given, there gives you b and a and ask you to guess σ and this kind of a problem does not seem to have any simple ways of doing and at least the simpler problem that we gave it was possible to make guess work. But I could not see any generalization of that, but their it works what the solution we got in class is correct but I could not find any general way of checking if I find have you let you know.

For the time being I want to show you what happens when you carry out such a conjugacy operation. In the two row notation there is some very interesting thing that happens. So, consider a element $\begin{bmatrix} 1 & 2 & .. & .. & n \\ a_1 & a_2 & .. & .. & a_n \end{bmatrix}$ and we call this a. And consider the transforming element σ which is $\begin{bmatrix} 1 & 2 & .. & .. & n \\ s_1 & s_2 & .. & .. & s_n \end{bmatrix}$. But now we also play a trick and this is the crucial trick.

Note that we can always also write it by list some random elements a_1 …... a_n and always says that the second row consists of whatever happens to a_1 under operation σ . So, this is saying nothing it is a redundant statement actually if you see that σ, well σ if you wrote it into in fact, you can consider it as a prescription for how to write the two row notation for σ, the instruction is that given an element a_1 in the lower row you must write the fate of that element under the operation σ that is all it says ok. But certainly therefore, I can write out σ in this form. This is the sub commentary Hammermesh does not give.

So, the only clever thing we have done here is we have chosen this top row configuration to be whatever is going to appear in a in the bottom. Now, so this is how we write σ. Now, we proceed to compute this b which is $\sigma a \sigma^{-1}$ in which we write a of course, as defined here we write σ^{-1} by taking this thing upside down, clearly σ^{-1} this thing upside down right which is the it is the inverse operation of σ and the first factor σ we write in this format, ok.

Now, what is clever about this is that now we can start cancelling. We can cancel this 1…..n and get it to be (s_1 s_2 …..s_n) going to this right or we can see it $s_1 \rightarrow 1$ and after that $1 \rightarrow a_1$, $s_2 \rightarrow 2$ and $2 \rightarrow a_1$. So, after I complete this multiplication I will basically have

$$\begin{bmatrix} s_1 & s_2 & .. & .. & s_n \\ a_1 & a_2 & .. & .. & a_n \end{bmatrix}$$

but now I can do the second operation where I cancel of the (a_1 a_2 …..a_n) row. So, that I am left which an answer,

$$\begin{bmatrix} s_1 & s_2 & .. & .. & s_n \\ \sigma(a_1) & \sigma(a_2) & .. & .. & \sigma(a_n) \end{bmatrix}$$

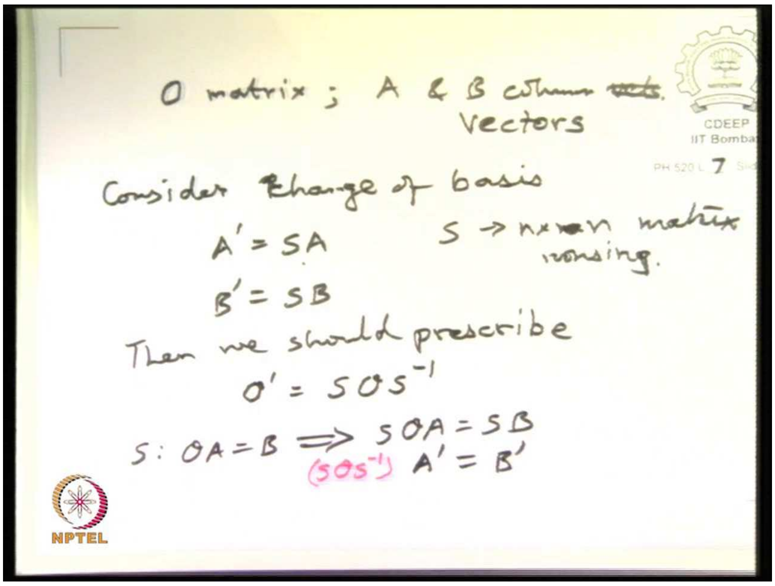
So, b is of this particular form.

Now, as I said the trick is in introducing the two equivalent ways of writing σ which were to ever the result is now very interesting I should have highlighted this. So, if somebody hands you a, and then hands you are σ you can proceed to write out b without having to suffer all this by just putting the first row of b to be whatever is the second row of σ, ok; and by putting in the second row whatever is the effect of operation σ on the second row of a. Another way of saying it is once you understand what this instruction $\sigma(r_1) \rightarrow s_1 \rightarrow s_2$ you take these instructions apply them to the top row by itself that will produce this row right because it sends $1 \rightarrow s_n$ and secondly, apply the same rules to second row independently.

So, whatever σ is suppose to do to a_1 put it here and whatever σ is supposed to do a_2 put here. The set of instructions are all here what happens to each a_1 value is written here. In other words the transformation amounts to applying the σ operation separately on upper and lower rows of a that is what it boils down to. And why does this work? Why does this magical looking thing working? To find the conjugacy transformation all I do is apply conjugacy individually on the top row and bottom row. It is the one of the comments I made earlier about the two row representation. What does two row representation have? Shows the state of the carrier space, and the conjugacy transformation is something that, so you remember the vector space example on a vector space remember linear algebra here if I have suppose I have OA = B, ok.

So, where A and B are column vectors.

(Refer Slide Time: 27:31)



So, O is the matrix and A and B column vectors. Now, you know that if you think of this abstractions the column vector is after all some components and if it is suppose to be something physical then it has a meaning of its own. The individual numbers do not mean anything the whole collection, but this array has an, has the independent meaning as an array only if the numbers are arbitrary because they are only components with respect to some basis, right. So, you can change the basis.

So, consider basis change, change of basis transformation $A' = SA$, where S is again $n \times n$ matrix, $n \times n$ matrix should be non singular and B' should not naturally be shifted to SB the components will change because you rotated your basis or even stress and strain does not matter this is the general linear transformations. Then that same operation OA = B is reproduced in the new basis provided we should prescribe $O' = SOS^{-1}$. Then I can make this whole equation OA = B read in the new language by applying S to the left, S on $OA = B \Longrightarrow SOA = SB$, but $SB = B'$ and by inserting a SS^{-1} here right. So, then, we have to prescribe here SOS^{-1} then I can put A' here by putting a SS^{-1} there, right.

So, the kind of transformation we are talking about conjugacy transformation is well known in linear algebra as a similarity transformation. So, when similarity transformation seems to a quadratically $\sigma a \sigma^{-1}$ on the operators. On the carrier space there is only a linear transformation and that is exactly what is happening in our two row notation because the conjugacy transformation amounts to doing conjugacy transformation on S as a carrier space list and on the second one also as carrier space list that is what is happening, because these are carrier space the vector space on which the groups act will eventually act as matrices, in fact, will be using that as a representation, ok.

So, I think once you understand it is quite trivial, but it is interesting to note. This is about regular cycles this we did we went over first theorems of equality of lengths of cycles, for regular cycles which form order and subgroup of S_n.