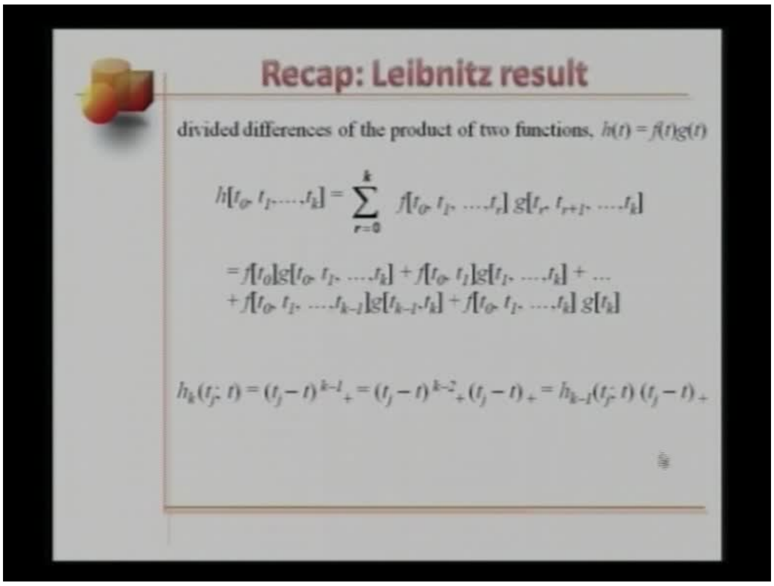
Computer Aided Engineering Design
Prof. Anupam Saxena
Department of Mechanical Engineering
Indian Institute of Technology, Kanpur

Lecture - 25

Good morning, we are continuing with our discussion on computer aided engineering design. We have been discussing splines, segments and curves. A quick recap on what we did last time. We used the Leibnitz result to derive the recursion relation for B spline basic function.

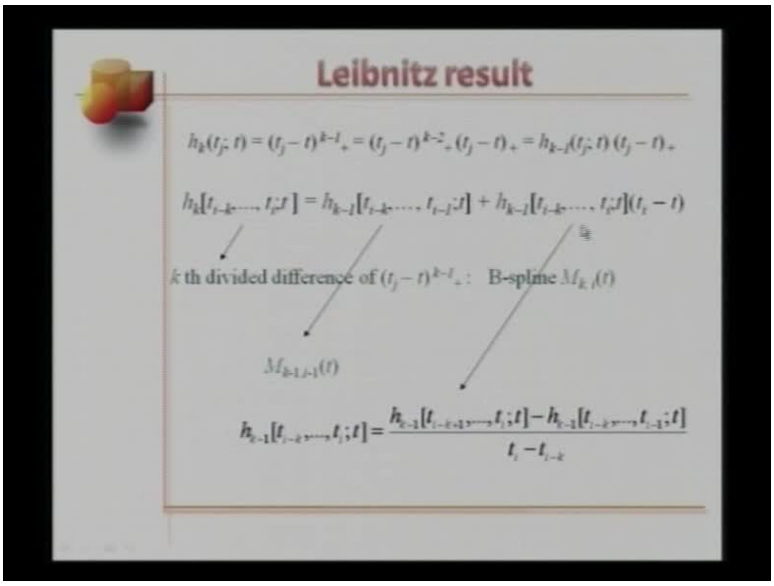
(Refer Slide Time: 00:51)



We work with the divided differences of the product of two functions h t equals f t times g t, the recap divide difference of h, which is h within square parentheses t 0, t 1 upto t k is given as the summation r going from 0 to k. The r divide difference of f, f t 0, t 1 upto t r times z within square parentheses t r, t r plus 1 up till t k. We can expand this the expression, will read like f of t 0 times g of t 0 t 1 upto t k plus f of t 0 t 1 times g t 1 upto t k plus for q more terms plus f of t 0 t 1 upto t k minus 1 times g t minus 1 z k plus f of t 0 t 1 upto t k times g t k.

We had model h t as the truncated for function of degree k minus 1 here. The subscript represented t order of the truncated power function. We had re written t j minus t truncated raise to k minus 1 as the product of two functions namely; t j minus t truncated raise to k minus 2 times t j minus t truncated. This expression here is h sub k minus 1 t j colon t. This expression here is simply t j minus t truncated, we computed the divided difference of h k t j colon t, using the Leibnitz relationship.

(Refer Slide Time: 03:09)

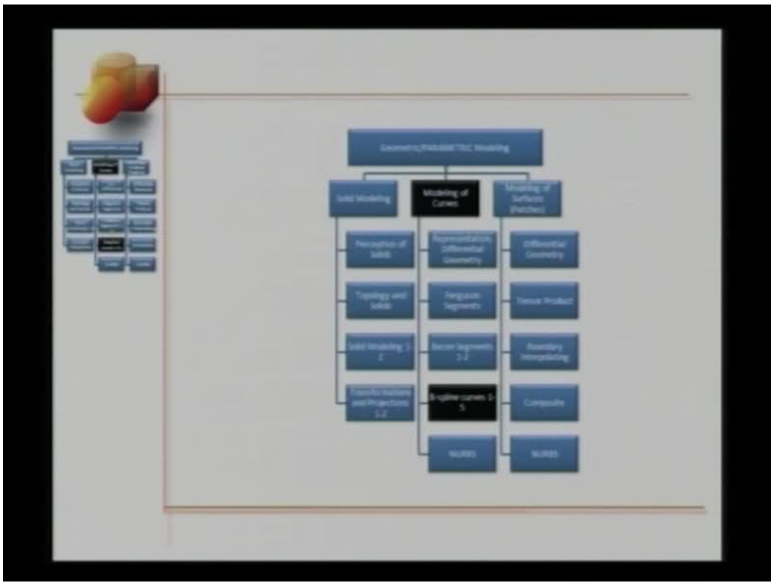


So, h k t i minus k, t i minus k plus 1 up till t i colon t. This is the k th divided difference is equal to h k minus 1 of t i minus k t i minus k plus 1 up to t i minus 1 colon t plus h k minus 1 of t i minus k t i minus k plus 1 up till t i colon t times t i minus t. This is a result that we have seen in previous lecture. So, there are only two terms that get retain in the k th divided difference of which k… Those terms at the last, but one term which is expand here and the last term which is this one. Since, we are interested in the value of t in between t i minus k and t I, we draw the truncated sign here, we have seen this before continuing from here.

If you look at this expression, this is the k th divided difference of truncated of function of k. So, this could be B spline M sub k sub i of t k is the order i is the index of the last number. How about this expression? This is the k minus 1 dived difference of truncated are function of order k minus 1, by similar logic. This could be M sub k minus 1 sub i minus 1 of t. k minus 1 is the order of this spline and i minus 1 is the last knot of this time. How will this expression look like? Now, notice here this k th divided difference of order k minus 1 truncated power of function. We do not know how this expression would look like?

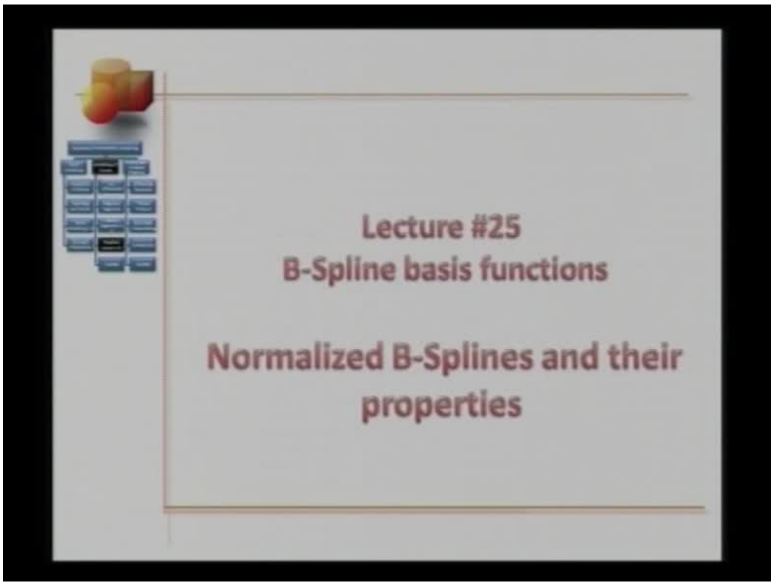
It is possible for us to express this relation in terms of two divided differences, each one would be the k minus 1 divided difference. So, h k minus 1 t i minus k after t i colon t, can be written as h of k minus 1 within square parenthesis t i minus k plus 1, going up till t i colon t minus h k minus 1 within parenthesis t i minus k, t i minus k plus 1 up to t i minus 1 colon t. Both of these are k minus 1 divided differences over t i minus t i minus k, this is this knot span here.

(Refer Slide Time: 06:47)



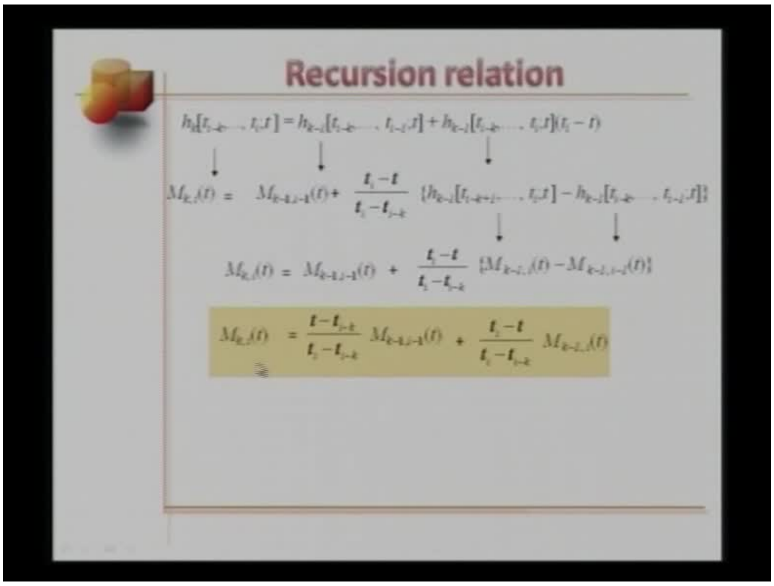
Now, we continue with our discussion today on B spline basis functions.

(Refer Slide Time: 06:54)



This is lecture number 25, we also be discussing normalized B spline and their properties. But first let us revise derivation of the B spline M k i as a function of t. So, I have copied this expression from the previous slide.

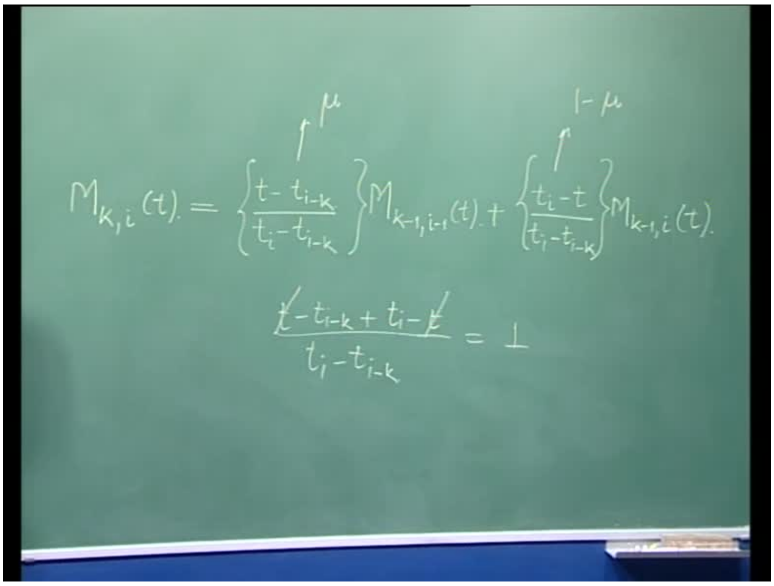
(Refer Slide Time: 07:17)



You know what this is? This is M k i of t and that is equal to M k minus 1 i minus 1 of t plus t i minus t over t i minus t i minus k times. We have seen what this is before. This is h k minus 1 square parentheses t i minus k plus 1 up to t i colon t minus h k minus 1 square parentheses t i minus k going up to t i minus 1 colon t. So, M k i t equals m k minus 1 i minus 1 t plus t i minus t over t i minus t i minus k times… Now, let us be careful here. This is the k minus 1 order this spline M k minus 1, the last knot is t I, so we have an index i here corresponding to it, minus again this is the k minus 1 order this line with the last knot t i minus 1.

So, the index here is i minus 1 and both of these are functions t. I can rearrange the right hand side of this equation to club M k minus 1 i minus 1 t and M k minus 1 i t. So, this is how the final recursion relation looks like, M k i t equal t minus t i minus k over t i minus t i minus k times M k minus 1 i minus 1 of t plus t i minus t over t i minus t i minus k times M k minus 1 i t. This is an important expression that you are going to be using as basis to study B spline basis functions further.

(Refer Slide Time: 10:49)



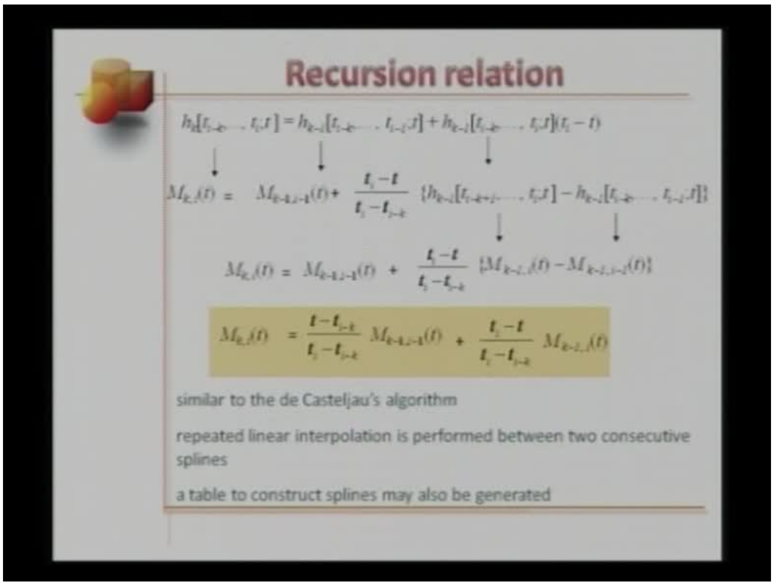
Many books will start this recursion relation or similar one to this discussion, the segment and curves. I have to gone quality books and I always wonder as to where this expression comes from? The previous few lectures derived this expression. So, we know where this relationship comes from. This looks a little complex in first curve but there is a very easy way for us to remember to this expression. Let me explain this on the board. This is the way I remember to write the recursion relation of B spline basis function. Say I have to write recursion relation for M sub k i of t.

I know this would be related to 2 B spline basis functions of order k minus 1 each. Also, one of those basis functions will have the last knot as i and the other one will have last knot as i minus 1. So, let me write them down now, M k minus 1 comma i minus 1 t and M k minus 1 comma i t. Of course, there will be associated constant of which with these basis functions, I will fill them up in a while. Now, first the denominator, I come back to my parenthesis function here.

The last knot is the t i and the first knot is t i minus k. I remember this and I use the entire knot span, t i minus t i minus k here as well as here, t i minus t i minus k. Now, the numerators i like do here is replace t i by t t minus t i minus k. So, copy this expression here and let go of this index, how about the numerical here? I do the other way, I will copy this expression here and let curve of the index here. This will be t i minus t, if we take little bit for you practice to understand and remember this, but this is not very difficult.

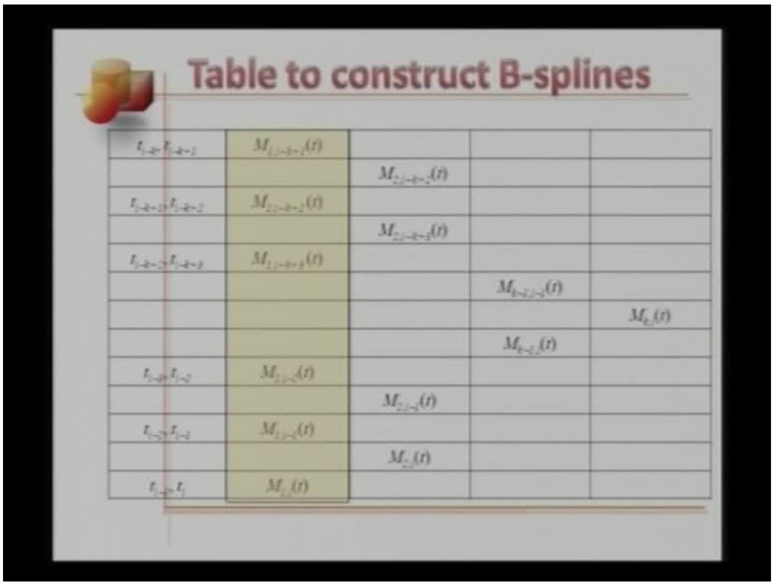
There is another interesting feature of this recursion relation now. Say this plate is mu, it could not be very difficult for you to figure that this plate here will be 1 minus mu. Easy way to check this is art raise together. So, we have the common denominator t i minus t i minus k will be have t minus t i minus k plus t i minus t. These 2 t s cancel out, so we have in the numerator t i minus t i minus k, which cancels with the denominator to get result as 1. What we notice here? This expression is very similar to the linear combination or the waited linear combination. We had seen in case of the De Casteljau, I will get… Remember we have studied that when discussing Bezier segment.

(Refer Slide Time: 15:34)



So, I said this before this expression is very similar to that that you have seen in the De Casteljau algorithm. This like repeated in your interpolation between two consecutive splines. There is however a minor difference. In the De Casteljau algorithm, these weights were fixed for the given parameter. However, here defending on these notes t i and t i minus k, these weights get changed. We will see this later, but because of this similarity a table like construction of B spline basis functions is very much reasonable.

(Refer Slide Time: 16:35)



Let us look at that, so in the first column, we have the knot span the first knot span t i minus k t i minus k plus 1, the second knot span t i minus k plus 1 k iminus k plus 2, third one, t i minus k plus 2 k i minus k plus 3, the fourth, therefore and so on and so forth. After this 1 here t i minus 3 to t i minus 2, then t i minus 2 t i minus 1. Finally, t i minus 1 to t i. We have to start with order 1 B splines at this time. M 1 i minus k plus 1. Now, concentrate on this not here, this correspond to the last knot in this interval.

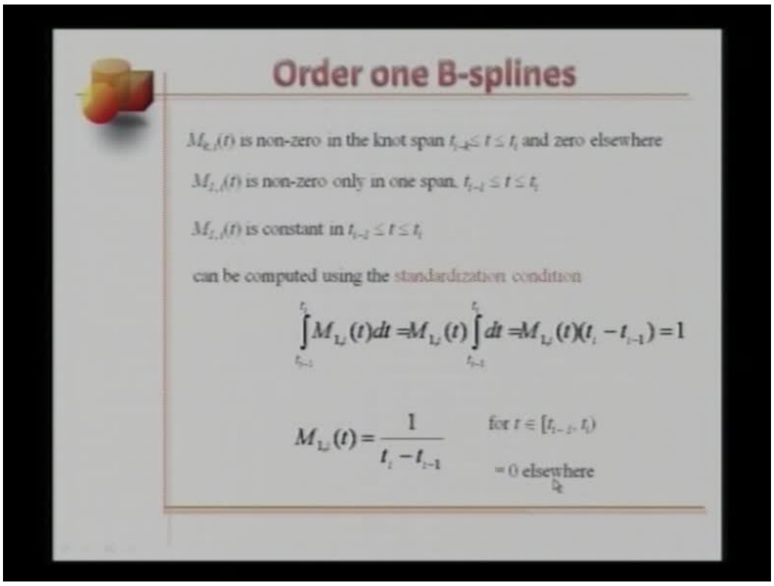
Likewise the second one will be M 1 i minus k plus 2, this knot here correspond to this knot here, on the right. Then M 1 i minus k plus 3, likewise M 1 i minus 2, notice the correspond here, M 1 i minus 1 and finally, M 1 i. We can start performing lenient tribulations from here on. Combined these two to get, what? Of course, you will get the spline of order 1 plus 1. How about the last knot over which the second order spline will stand this time? Take look at this and this come back take look at these knot spans of here. This is the first knot span, this is the second knot span.

So, in two something is expected to be styling over these two knot span. The last knot will clearly be i minus k plus 2 and 2 i minus k plus 2 of t. What happened, to be combined these 2 order one basis functions? Again there is increasing order and the last knot is i minus k plus 3. We can keep on moving, we combined these two guys together to get M 2 i minus 1, which is this knot here and M 2 i is obtain will be combined M 1 i minus 1 and M 1 i. There is another way to figure this last knot here, if we go back to first column, we will have to merge these two knot spans and the last knot would correspond to knot under write of this merge knot span.

We can keep working on the repeated lenient tribulation and eventually we get to the last, but one stage where will have 2 basis splines M k minus 1 with the last knot i minus 1 and M k minus 1 with the last knot i. To look at this basic function, you would see that it could stand with the knot span given by these individual splines, t i minus k 2 t i minus 1. I would to take this simply subtract k minus 1 from i minus 1 to get the first knot i minus k, to right index over here will automatically denote right most. Now, i minus 1, likewise for M k minus I 1, this would be not standing on the north spline given by this. The first knot will be t i minus k plus 1, the last not will be t i.

So, as we move towards the right constructing the splines of higher order, these knot spans will get merged 1 by 1. Eventually, we will get to M k i that would stand in the north spline t i minus k 2 t i, not very difficult once you have little bit of practice. Now, we have this recursion relation that, allow us to construct M 2, M 3, M 4 and so on up to M k. We will have to start however with the M one’s. Do we have the recursion relation for M 1, now how to be then find, what M one’s are? Let us suppress this. What you think the order 1 basis spline will look like?

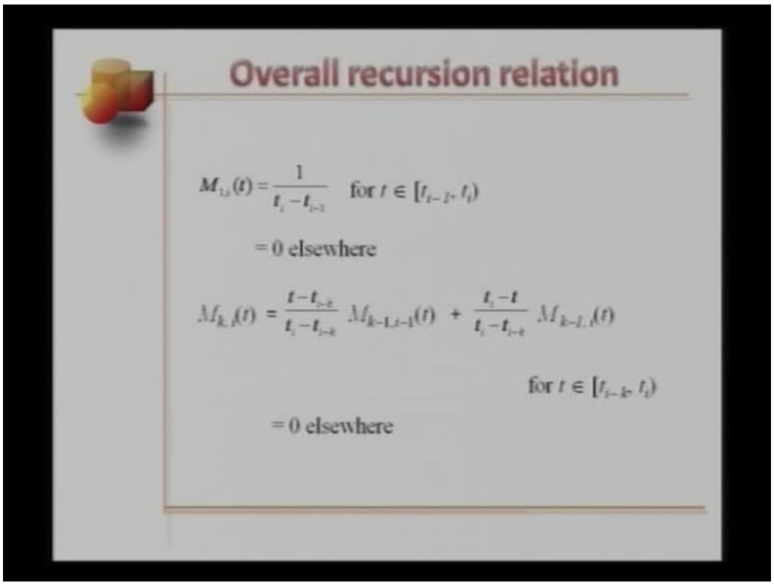
(Refer Slide Time: 22:44)



This use this observation that M k i of t is non zero in the knot span t i minus k to t i and 0, elsewhere the order 1 B spline will be following the same root or the same observation. So, M 1 i is non zero only in one span, which is t i minus 1 to t i and M 1 i is 0 for all the other values of t. Also, sense the order of M 1 is 1, the degree is 0. That would mean the M 1 i should be a constant in this knot interval, t i minus 1 to t I, what would that constant be? We will be seen this standardization condition report and we can use that compute M 1 i. Integral going from t i minus 1 to t i of M 1 i of function of t d t should be equal to 1 over 1.

Since, M 1 i is a constant, we can take this thing out of the interval sign, so this is M 1 i of t integration t i minus t i d t. This is M 1 I t times t i minus t i minus 1 and this is equal to 1. So, M 1 i of t is equal to 1 over t i minus t i minus 1. With this information here, we tabular construction, that we have seen in the previous slide will work very well. Of course, M 1 i of t equals to 1 over t i minus t i minus 1 for values of t belongs to t i minus 1 and t i. Notice here that I am using the closed bracket on the left and open bracket on the right, that will imply that t i with take the value t i minus 1, but not t i. Of course, M 1 i will be 0 for all other values of t.

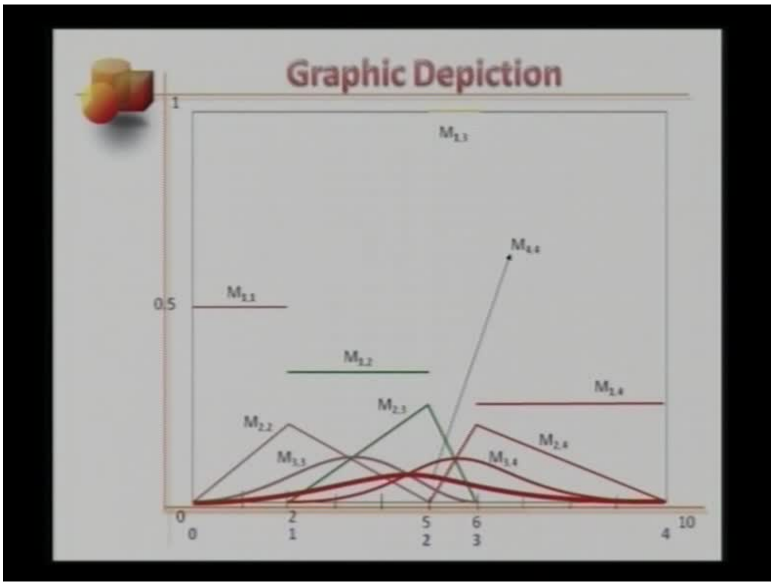
(Refer Slide Time: 25:50)



So, the overall recursion relation looks like M 1 I t equals 1 over t i minus t i minus 1. For value of t in between t i minus 1 and t i close bracket from the left and open bracket from the right. This is 0 for other values of t and then M k i t equals and you know how to remember this expression? t minus t i minus k over t i minus t i minus k times M k minus 1 i minus 1 of t plus t i minus t over t i minus t i minus k times m k minus 1 i t.

This is for values of t in between closed t i minus k comma t i open and 0 for all other values of t and this index k go from 2 to any value you want.

(Refer Slide Time: 27:13)



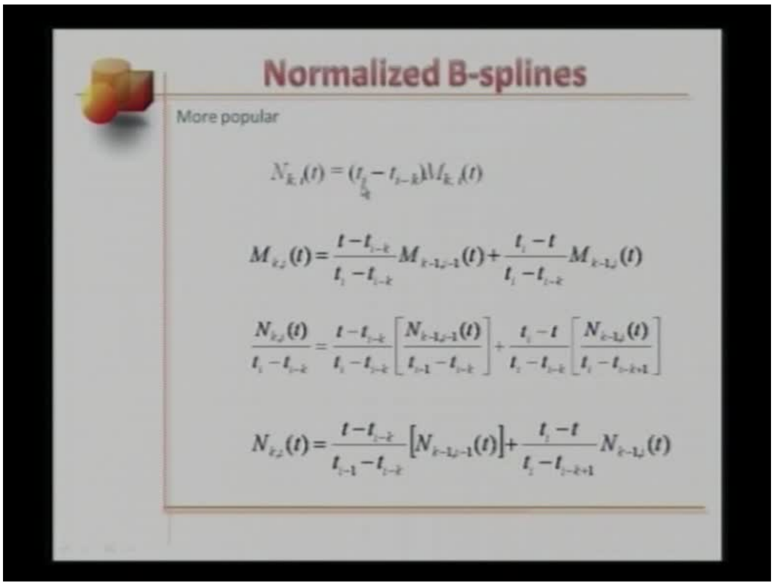
Let us look up how this individual basic functions look like graphic? Let us take an example. Let say we have the marks 0, 2, 5, 6 and 10, how many knot spans will they be 1, 2, 3 and 4? What is the highest order of B spline basis function that can stand over this entire knot span? I would want you to guess that. Anyhow let us start constructing order 1 B splines, let us number the knots t 0, t 1, t 2, t 3 and t 4 or simply 0, 1, 2, 3 and 4. This is how M 1 1 will look like, the order M 1 and the last knot is the second index. This value is 0.5, how is that so? M 1 1 is 1 over 2 minus 0, how about the second first order spline?

This will be M 1 2, it is over 5 minus 2, there is 0.3333 recurring about the third first order B spline. This is M 1 3 1 over 1 this i s M 1 4, which is 1 over 10 minus 6 1 over 4.25. Notice that all this first order B splines are constants, in the respective t intervals. With these this sub construct order 2 B spline or degree one B splines. Of course, geometrically they are going to be looking like straight lines, inclined that is. This curve here is M 2, what is the second index 2? Clearly it stand over the knot span 0 to 2, including the first two knot spans.

As I said before, this is linear. Let me ask this question with this function here satisfy definition of a spline, do you remember that? Well the answer of course, is yes. This is an order two spline, a degree one spline and it has to be position continues everywhere. Even those slope here is not unique, we can see position continuous throughout. Why position continues? Its c n minus 2 where n is the art. So, sense the order 2 it would be c 0 continues. Coming back to the second, second order B spline. It spans between the knot span 1 2 and 2 3 again linear and c 0 continues, this is given by M 2 3, this is the third order 2 B spline, M 2 4 order t and x of the last number. Now, let us start constructing order 3 B spline. The first one will be the combination of M 2 2 and M 2 3. Really the order is going to be raise by 1 and the last knot will be 3. In this case, these two knot spans will get merged. M 3 3, which is displaying here will be quadratic. c 1 are slope continuance prove up. Notice that the slope of M 3 3 3 and here are both 0.

Coming to the second order 3 B spline that would be acumination of M 2 3 and M 2 4, in knot spans getting merged here, will be from 2 to 6 and from 5 to 10. So, M 3 4, which is red curve here is stand over knot t 1 and t 4. Finally, one would combined M 3 3 and M 3 41 together to have order 4 B spline, which is this bold red curve here. Of course, this B spline could be standing over entire knot span t 0 to t 4. This is interesting, any one would able to practice this construction technique and understand, how B spline basis functions get constructed recursively?

(Refer Slide Time: 34:14)

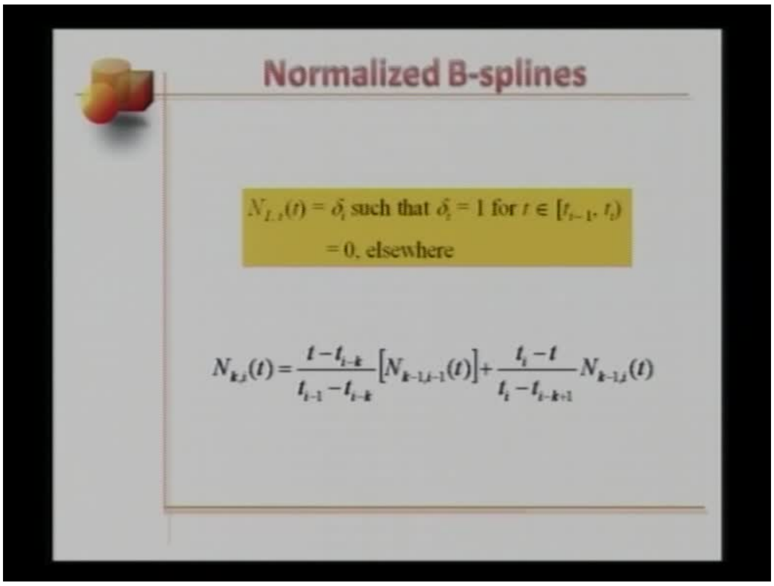


Now, let us move on normalized basis plans of B splines. Of course, they happen to be more popular than M k i of t. The only difference between normalized B splines N k i of t and M k i of t is this factor here. In other words you get normalized B splines by multiplying M k i t with the entire knot span over which this basic functions stands, which is t i minus t i minus k. Now, you have seen this recursion relation reports in terms of M's order k, order k minus 1, order k minus 1. The last knot I, the last knot i minus 1, the last knot i.

You have also seen a little technique to this remember this expression. Using this definition here, I can substitute for M k i M k minus 1 i minus 1 and M k minus 1 i. So, M k i t is M k i of t over t i minus t i minus k. Use this relation, how about M k minus 1 i minus 1, that is N k minus 1 i minus 1 of t over last knot t i minus 1 minus the first knot t i minus 1 minus k minus 1. It is t and t minus k plus t i minus t over t i minus t i minus k times N k minus 1 i over t i minus t i minus k plus 1.

We can rearrange this relation as N k i t equals t minus t i minus k overt i minus 1 minus t i minus k times N k minus 1 i minus 1 of t plus t i minus t over t I minus t i minus k plus 1 times N k minus 1 i t. Now, I find it personally to little difficult to remember to this expression, if you can it is wonderful just that I find it little difficult, I find It more convenient to remember this expression here and to compute normalized B spline N k i all I will do is I multiply N k i by t i minus t i minus k.

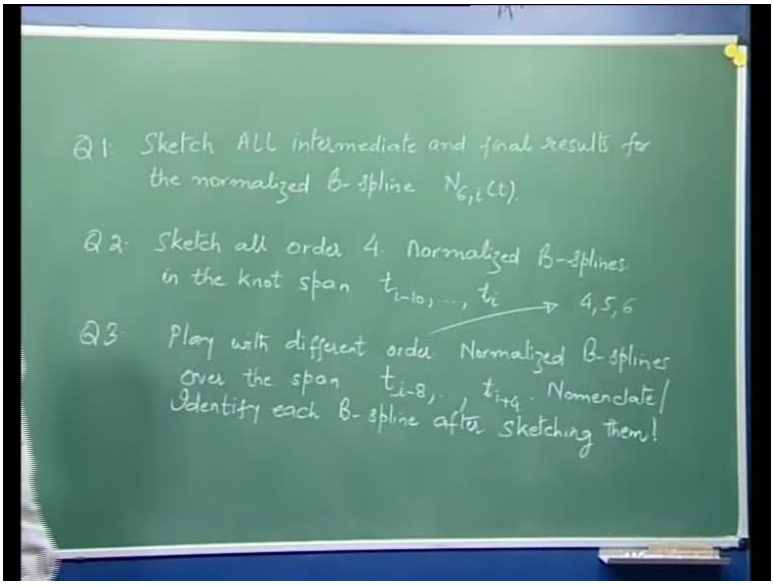
(Refer Slide Time: 37:50)



So, the overall recursion relation for normalized B splines, the first order normalized splines N 1 i t is equal to delta i such that delta i is equal to 1 for values of t belonging to closed t i minus 1, open t i. N 1 i t is 0 elsewhere for all other values of t. Then for any other order N k i t is equals t minus t i minus k over t i minus 1 minus t i minus k times N k minus 1 minus i minus 1 t plus t i minus t over t i minus t i minus k plus 1 times N k minus 1 i t. Before we proceed, a few commons on this recursion relation to summarize. We have obtained to this recursion relation using truncated power series function using Newton's divide differences and the associated Leibnitz result.

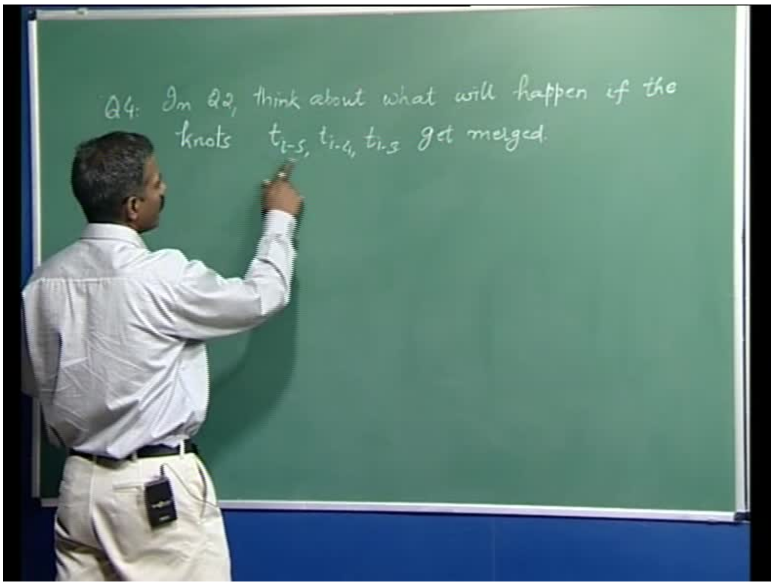
The second observation is with regard computer coding. If you recall we had gone through very serious procedure to compute polynomial splines. In contrast this is much easier relation to work with and trying to code to compute B splines basis functions, whether normalized or not normalized. I have always been emphasizing that is nice idea for us to be comfortable to constructing or sketching these B spline basis functions geometrically. The reason is very simple, it will help us to understand the properties of normalized B spline basis function better, if we are adopt in sketching these functions.

(Refer Slide Time: 40:35)



In that regard, let me leave you with a few questions for you to (( )). Question 1, sketch all intermediate and final results for the normalized B spline N 6 i t. Remember that is sketching procedure for N 6 sub i t for any other normalized B spline basis functions is identical to that corresponding N k i t. You do not want to worry about the magnitude while you are sketching these B spline basis functions. Question 2, sketch all order 4 normalized B splines in the knot span t i minus 10 to t i. We would only need final result here, we do not want to worry about the intermediate B spline basis functions. Question 3, play with different order normalized B splines over the span t minus i minus 8 to t i plus 4, normal plate or identify each B spline. After sketching them to make matters simple, try order 4, 5 and 6.

(Refer Slide Time: 44:57)



Question 4, in question 2 think about, what will happen if the knots t i minus 5 t i minus 4 t i minus 3 get merged? So, remember here that not all order four normalized spans will get effected when a few knots get merged. Also, we can stimulate the merging either from the right or from the left, but one means you can do so in steps, you can first say t i minus 4 equals t i minus 3. Then t i minus 5 is equal to t i minus 3 or you can merged these knots from right i. Saying this t i minus 4 equals t i minus 5 and then t i minus 3 equals t i minus 5. In the next lecture, we are going to be studying the properties of normalized B spline basis functions.