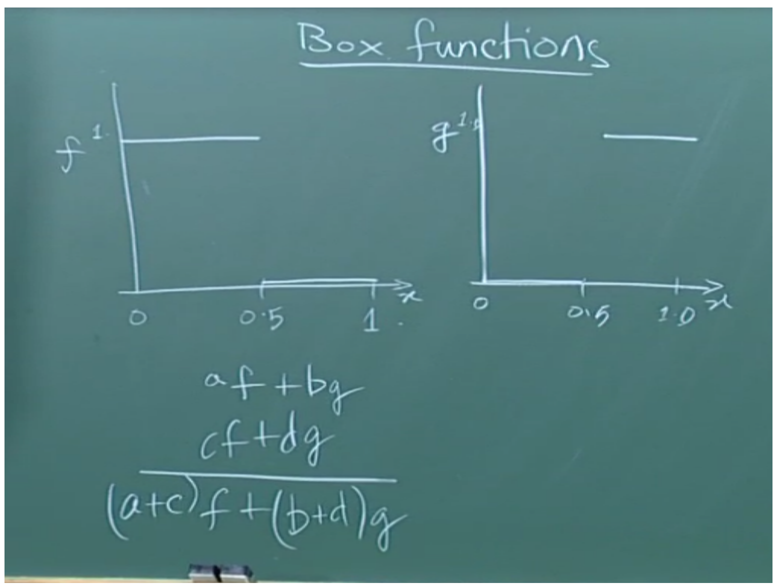
Introduction to Computational Fluid Dynamics
Prof. M. Ramakrishna
Department of Aerospace Engineering
Indian Institute of Technology – Madras

Lecture – 03
Representing functions - Box functions

So we have looked at representing that we can represent functions on to computer. So today we will look at specifically some mechanisms by which we do this. What we saw there was that we can represent functions, but we want to be able to organize functions the way we organized vectors which we have seen in the last class. Today I am going to talk about Box functions.

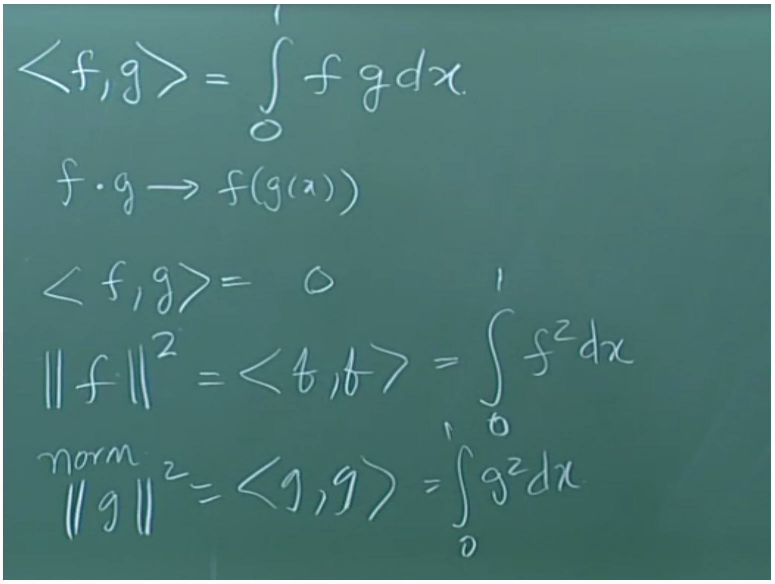
(Refer Slide Time: 00:43)



It is a special class of functions. I am going to define 2 functions to start with, one is a function f is defined on the interval 0,1. The value of f on the interval 0.50 to 0.5 is 1 and outside that it is 0. I define another function g it is also on the interval 0,1. Its value on the interval 0 to 0.5 is 0 and between 0.5 and 1 it is 1. So this value is 1 function value there is 1. If you consider a linear combination of these functions if you look at af + bg you will see that just like in the vectors in the case of vectors if I took another linear combination cf + dg an addition of these 2.

And I can add them out it is defined on the same domain an addition of these 2 will in fact give me a + c * f + b + d * g. So we see that just as we did the other usual letters that you are used to that we have something that looks like a summation and we are able to do the vector algebra and set it up in a systematic fashion once we define the dot product what we need to do is for functions we want to be able to define a dot product. I will define the dot product as follows.

(Refer Slide Time: 03:03)

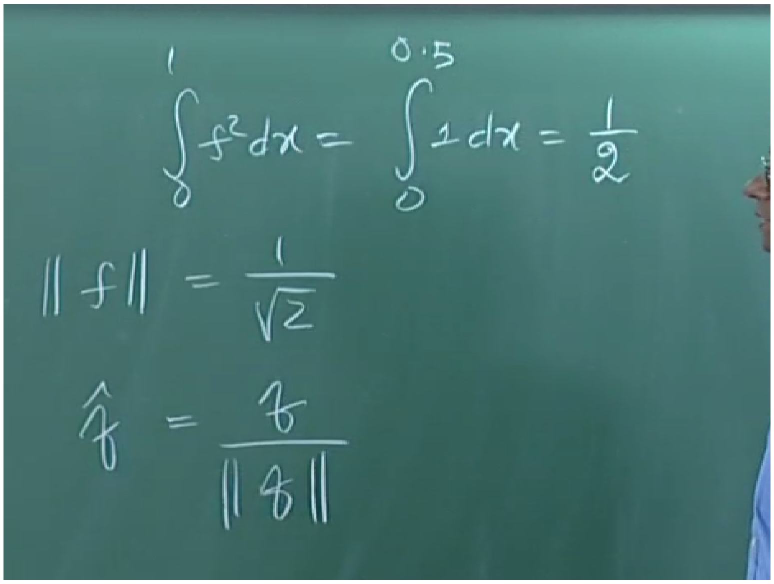


I will define it on for these 2 specific functions 0 to 1 as f * g * dx. I use the notation f, g right, because the dot is already used in the case of functions the dot is already used for composition of functions so we do not want any confusion. If you say mathematics if you say fg it is possible that you confuse it for f(g(x)) so that we do not confuse. We introduce a new notation for the dot product.

So what is f.g? f.g is this integral and as it turns out because these regions are non overlapping this dot product in fact turns out to be 0. You can work out that integral and check that the dot products in fact are note to be 0. We will come back to this point later. Right now what I am interested is in getting as we did earlier defining a magnitude something like a magnitude in the case of a function it is called a norm.

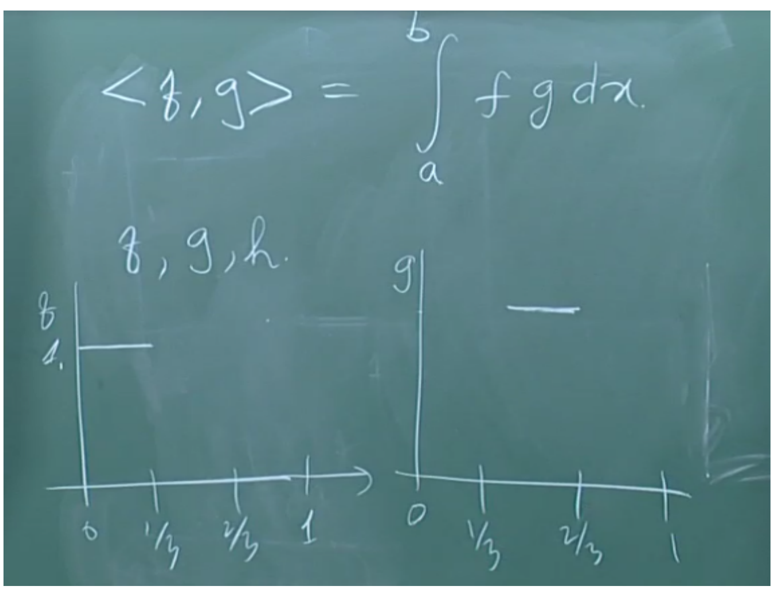
We define a norm that comes from this dot product just like we did with vectors. So I could define the square with a norm as f.f which turns out to be the integral 0 to 1 f square dx. Now in the similar fashion we can define norm g square is g.g. integral 0 to 1 g square dx. It is area into the curve. It is clear that they should both be the same. What does this work out to? So what is norm of f?

(Refer Slide Time: 05:18)



So it is integral 0 to 1 f square dx which is actually the integral 0 to 0.5 1 * dx which is one half. So norm of that in fact 1 over square root 2 and again I am just duplicating whatever work we did with the usual vectors that you have seen I can define a unit vector as f/norm of f and as I would expect its magnitude so to speak would be 1 is that fine. Any questions? Fine. What do we have? We will now manage to actually repeat the process that we did for vectors we can actually get.

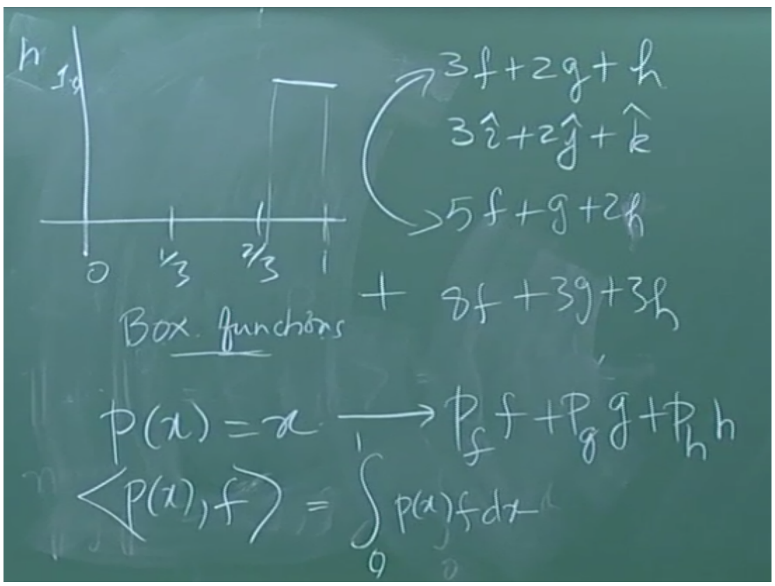
(Refer Slide Time: 06:31)



We can actually define a general dot product of 2 functions f and g. We will repeat the process, but in this case the functions could be defined between any 2 intervals a and b and you could actually define it as f g dx. f and g are functions that g from take some real number and return a real number. Is that fine? So in general it will not be 0 to 1. I will do everything here as from 0 to 1, but in general the definition is not restricted to 0 to 1 is that fine.

So why do not you try what do not you try something with 3 coordinates. So what if I define 3 functions f, g, and h I try something f, g, and h. f is defined in a similar fashion now the only difference is because I have got 3 functions I am going to break up the interval 0 to 1, one third. 2 third and that is my function f and I define my function g between one third, 2 third that is 1, 0 that is my function g and in a similar fashion I will define my function h.

(Refer Slide Time: 08:12)



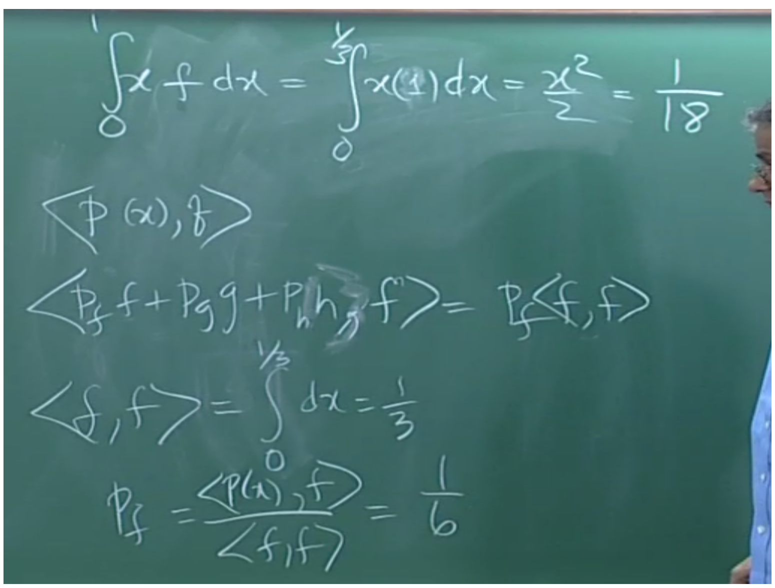
That is my function h. The value of h is 0 on the interval 0 to 2 third, 1 on the interval 2 third to 1. In a similar fashion g is 0 on the 2 intervals, 0, one third, 2 third, one third and it is 1 on the interval one third to 2 third. Here it is 1 on the interval 0 to one third and it is 0 on the rest of it. Now we go back to the example that I used in the earlier class. So I asked a question what is 3f + 2g + h. It looks very similar to 3i + 2j + k where i, j, k are standard unit vectors.

So it is almost as I have defined these you previously when you first learnt it you may wonder what they occur i, j, and k it is almost as though I actually told you what are i, j, and k define them in terms of functions and you can if you have 5f + g +h you can actually add up these 2 quantities you can perform the arithmetic just like you do normally and get 8f + 3g +3h enough that is fine.

All very nice! So it looks like we can use combinations of these coefficients and represent function how well does this work so that is the question that we have how well does this work? So what we have so far. We have defined functions in this form because it looks like somewhat like a box it is called a Box function. We have defined Box function and it is clear that at any interval I can define any number of Box functions that I want we will get to that in a little while and it looks like I can represent functions on the interval 0 1 using the Box functions.

So we will try to use it and see what happens. We will pick a simple function. We will pick a very simple function p(x) = x. So I would like to represent this function as p of f the f component pf * f + Pg * g + Ph * h. This is the f component, the g component, the h component of P and all I have to do now is take the dot product. So I asked the question what is P(x),f what is this dot product. This is the integral P(x) f dx integral 0 to 1. We will just evaluate that.

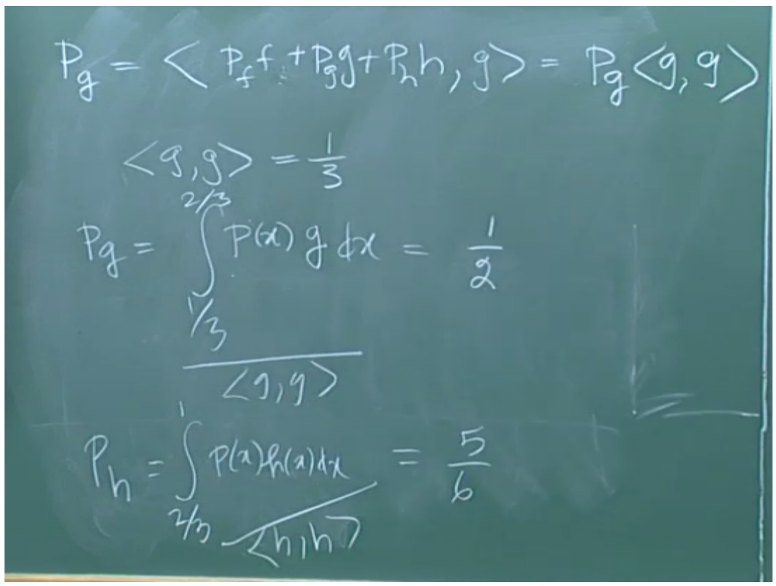
(Refer Slide Time: 11:32)



We will just evaluate that. So we will get x * f integral 0 to 1 dx which is well we know f is nonzero only between 0 and one third, so it is 0 to one third x * 1 in that domain dx which is x square/2 which gives me 1/18 is that fine? That is actually performing the integration. What is the dot product P(x), f without performing the integration from here? So what would we like it to be? Let me not say what is it what is this quantity dotted with f. Pf * f + Pg * g + Ph * h dotted with f what is this quantity.

It is just Pf(f,f) because f.g is 0, f.h is 0 and what is f.f? There be careful now. I changed the definition of my f. What is f.f. Well we have to calculate f.f for this case f.f is the integral 0 to and I will write it only till one third dx which gives me 1/3. So presumably if I have one of them it will give you 1 over n. So Pf in fact is I am going to get it from here. P(x) dotted with f/f.f which gives me 1/6. There are any questions? So what we are doing now.

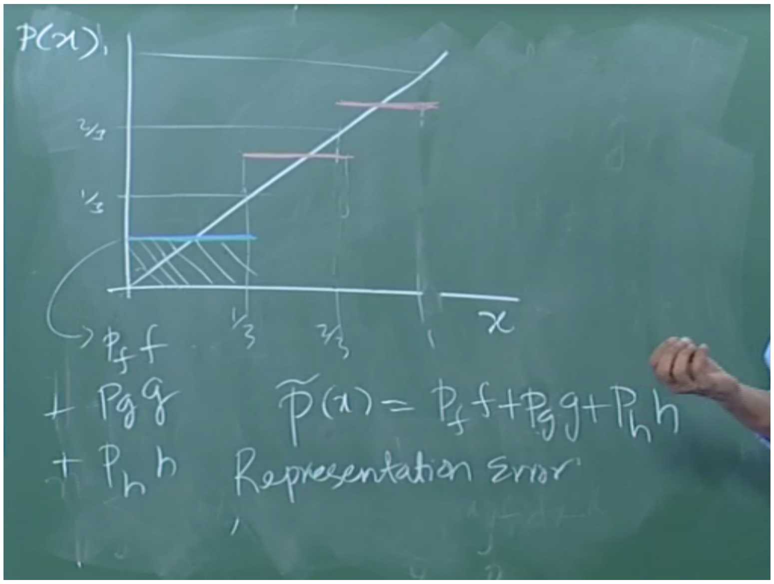
(Refer Slide Time: 14:24)



In a similar fashion Pg will be Pf * f + Pg * g + Ph * h dotted with g which will give me Pg * g.g and you can check that g.g is going to be one third, p.g is going to one third and therefore what is Pg. So if you take P(x) * g integrate from one third to 2 third because that is where g is nonzero. Of course do not forget that you have to divide by g.g. What does this give me? This should give me one half.

This will give me one half and Ph in a similar fashion Ph you will have to integrate it between 2 thirds and 1 P(x) * h(x) * dx/h, h which will be you can verify that it is 5/6. If you are wondering how I know the answer it is just basically comes from asymmetry. So this is about a 6 way from 1 that is basically how that comes. So what is a function? What is the function that we want to plot?

(Refer Slide Time: 16:49)



Let us plot this function. So this is x, this is P(x) and the function itself is a 45 degree line is that fine. The function itself is 45 degree line. We have representation of a function on those 3 intervals represented by I use 3 different colors here represented by values which are so this is one third, 2 third, 1. It is a 45 degree line, one third, 2 third, 1. So where is one sixth. One sixth is somewhere in between.

What is that? That is Pf * f to which I want to add that is + Pg * g and the last one not that different in colour, but that does not matter. The last one is Ph * h. is that fine. So look at the 2 functions. I have the 45 degree line that I am trying to approximate and I had this other thing made up of blue, and red lines which are nothing but our representation on the computer If you have to use Box functions to represent the functions in the computer this is what we would get.

So clearly just like we have errors in representing numbers, we have round off error. We have a similar error here if you try to represent the function. If you try to represent the function using Box functions, you do not get the original function actually it should be obvious when using constant functions to represent something that is linear. But the fact that matter is that I projected it, I went through the formal process.

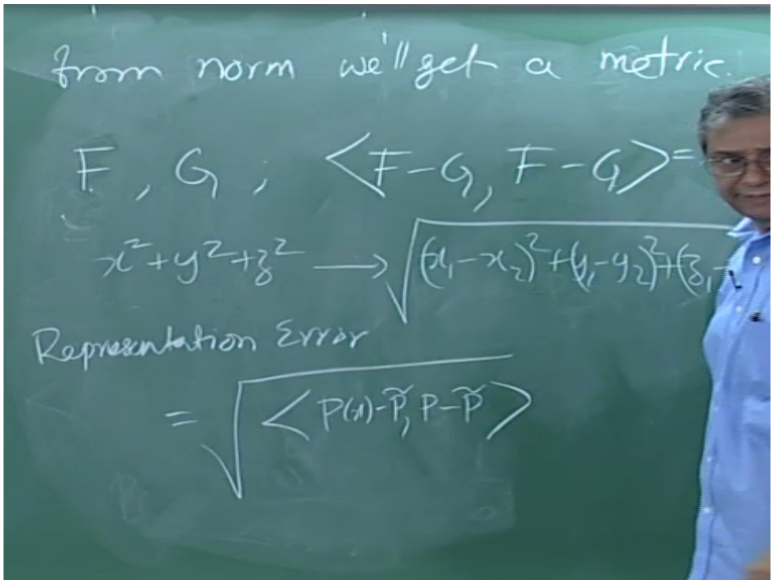
I projected the functions on to the Box functions, I went through the formal process and what do I have. I basically have a function here. I have a function here which supposedly represents this properly. Take a closer look. In some integral sense if you look at this you notice the area under this curve is indeed the area under that little triangle there. We will manage somehow we have captured that integral being defined as a dot product have come through and it is the same thing here.

So we are somehow capturing that area properly, but the function value itself is not right. So I no longer want to call this, I do not want to call this P of x I want to call our representation give a representation a name. Right now I will call it P tilde. Right now I will call it P tilde. We will introduce more formal notation later. Right now I will call it P tilde. So P tilde of x = Pf * f + Pg* g + Ph * h. So what we got so far.

If we give me a function it is possible for me to define Box functions. It is possible for me to use Box functions on the same interval and represent get components, project the function on to the Box functions and get components. I am aware that the resulting function that I have will not be identical. It is unlikely to be identical through original function. There is an error. In the case of numbers, we called it round off error. In this case I will just call it representation error for now. So this is a representation error and the representation error.

We have to now figure out how to quantify the representation error. So having found a way by which we can find a representation and have discovered that there is an error in the representation we want to now quantify how we get that representation error. So how can we do this? Suggestions? So we have the dot product. We use the dot product to do it. So normally what we do is we will reuse so that is the reason why this dot product is extremely powerful that is the reason why we have used the dot product. We use the dot product here.

(Refer Slide Time: 22:01)

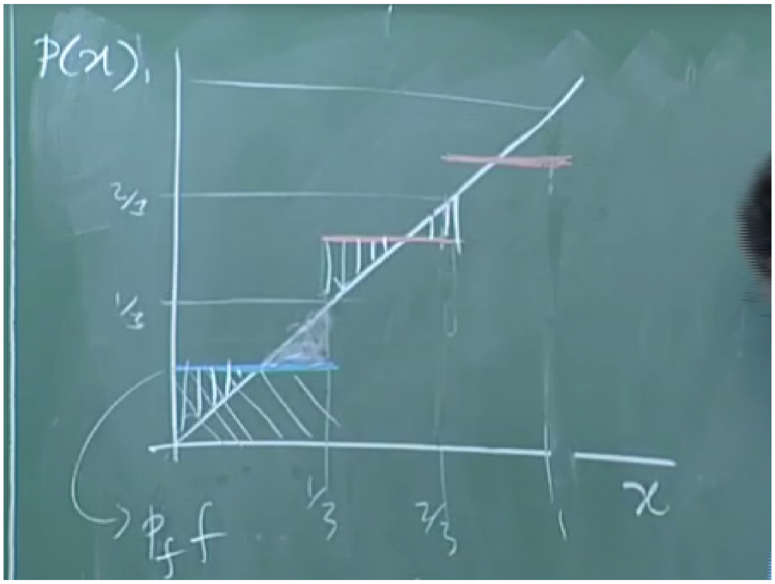


To define something called a metric. So from the dot product, from the norm we will get a metric. Metric is basically a distance. We will get a distance now. Right now we only have magnitude. So think about it normally you can have magnitudes of vectors which is what we normally we have defined so far, but you can also get the distance between 2 position vectors. The end points of 2 position vectors. So we will do the same thing.

So we can use the dot product. So if I have 2 functions F and G and I want to know what is the distance between those 2 functions then I can look at F - G and take that dot product with itself and that should give me a measure of the distance. So this should be some distance between F and G square. In Cartesian coordinates if your magnitude of a vector is x square + y square + z square and then the distance function between the magnitude square is x square + y square + z square.

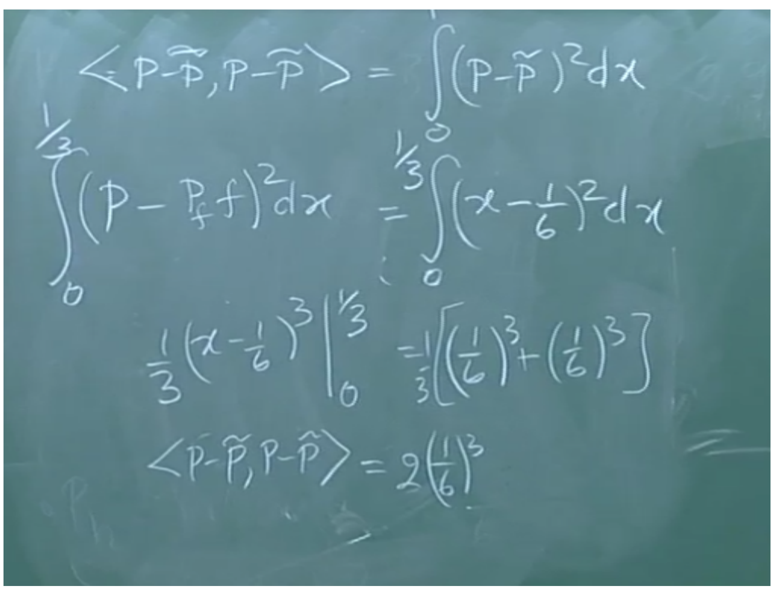
Then you would say x1 - x2 square + y1 - y2 square and so on. So I am just basically mimicking that + z1 - z2 square and the distance itself would be the square root of that. I am just basically mimicking this. I am just repeating whatever we have done in the standard vectors. I am just repeating them. It works there, it should work here. So what is the difference? So now I can ask the question representation error. I can define it as square root of the dot product of P of x - P tilde, P - P tilde is that fine?

(Refer Slide Time: 24:22)



P - P tilde. Let us get back to this P - P tilde. So P - P tilde is here. This is P - P tilde. That distance is P - P tilde. In fact, if you look at it these 3 intervals P - P tilde is actually the same in this particular case. P - P tilde happens to be the same. P - P tilde happens to be same. So if I find P - P tilde for the first one then I do not have to really repeat it for all 3 of them. I will just do it for the first one. So find out what is the representation error.

(Refer Slide Time: 25:02)

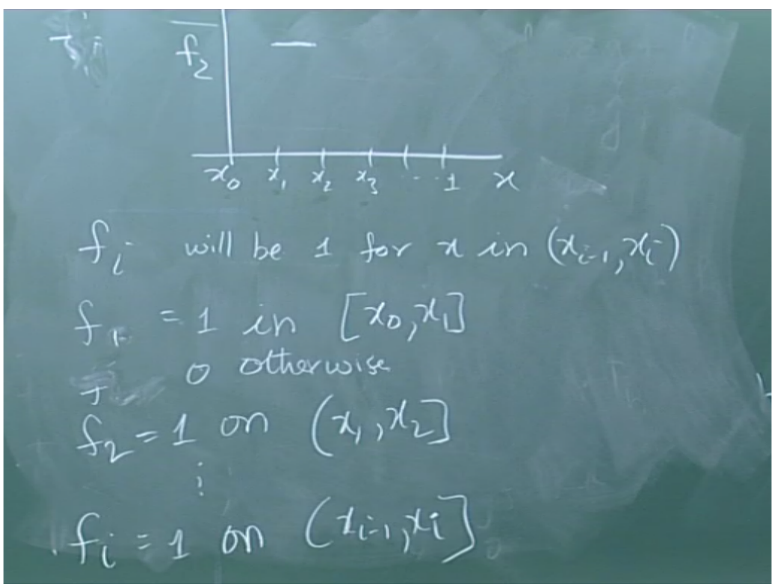


Representation error is P - P tilde a dot product = dx from 0 to 1 and I will find only the first component. So I am going to find P - Pf * f squared dx integral 0 to one third is that fine everyone? These 3, the differences between the representation and the function in this particular problem they happen to be same. I am just making use of that fact that is all. So what is this? x - 1/6 squared integral 0 to one third dx this is the only way by which we could do this.

So this is x - 1/6 cube one third between the limits 0 and one third. What does this give me? So one is going to be one third - one sixth which is one sixth cube and the other is going to be + one sixth cube. Is that fine? And there are 3 of them. So to get this, to get the representation error I have 3 of these so in fact it turns out to be 2 times one sixth cubed. Is that okay everyone. So this would be the representation error. So as I said please remember this.

This is equivalent for our functions like similar to round off error. Are there any questions? Now we are going to do can be do better. This function on the interval 0 and 1 we define 2 functions f and g and we define 3 different functions f, g, and h why not see if we can define n functions. So we will define whole host of function. So we will define a whole host of function and we will use the subscript fi to indicate the ith function.

(Refer Slide Time: 28:28)

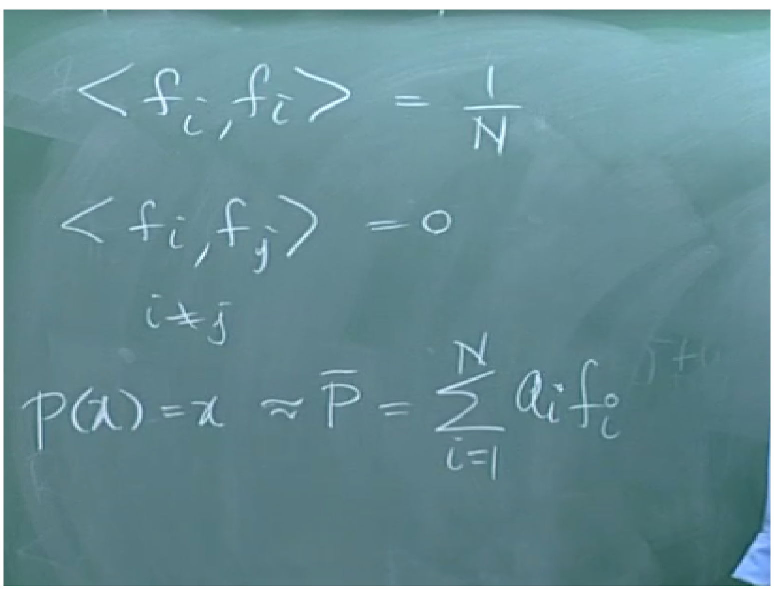


And what is the ith function I am going to look like. The ith function so if I have n intervals, if I have n intervals and number these x0, x1, x2, x3 and so on. All the x's are equally spaced. This goes to 1 my ith function so the second function for instance will go from will be nonzero from x1 to x2. The ith function that would be f2. The ith function will be nonzero will be 1 for x belonging to xi - 1, xi. Is that fine?

So I had to make it be a little more precise about the way I do it. So what I will do is I will say f0 = 1 in the close intervals x0, x1. f1 and it is 0 otherwise. f1 = 1 on open interval it is open on 1 side x2 closed on the right side. So we will keep everything open thereon open on the left closed on the right. fi = 1 on x1 - 1 xi. I make a mistake. So f2 would go from 1 to 2 and f3 would go from 2 to 3. f1 will go from 1 to 2. I started with f0. I am sorry. Yeah fi.

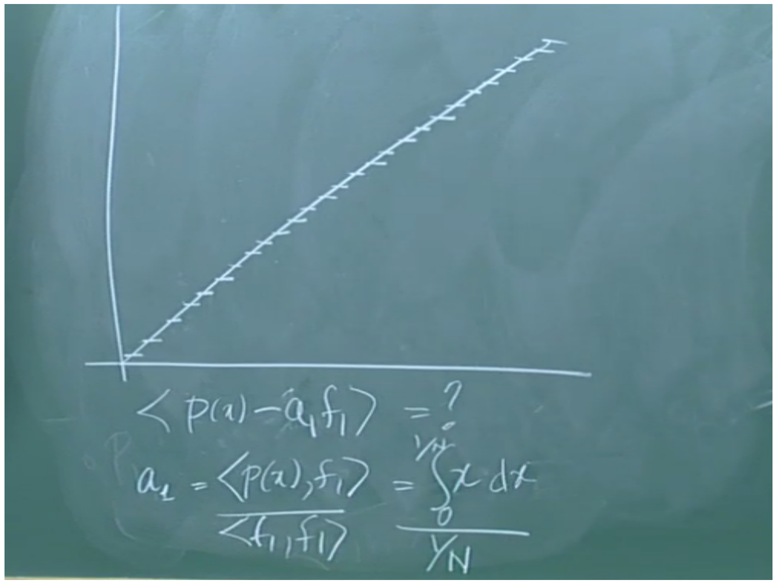
That is the programming. When we program we start the count at 0 actually I started the count there at 0. You will have to keep your eyes open for that. Are there any questions? The intervals are equally spaced in all these discussions, but you have to know see why these functions are orthogonal, why does this work, why are we able to do this that is where we are going to end this class, but right now let me see if I can use this fi and then we will summarize what we have got. So what do I have?

(Refer Slide Time: 31:35)



If I take fi.fi the length of any interval is 1/n fi.fi is 1/n Please check to make sure that works fi.fi is in fact 1/N. fi.fj if I is != j = 0 so any function in our case right now the function that we are using can be represented as is approximately some P tilde can be represented as summation over I going from 1 through N some ai fi. Let us get back we will try to draw a figure for this and see what happens and again we will find the representation error. Draw a figure for it.

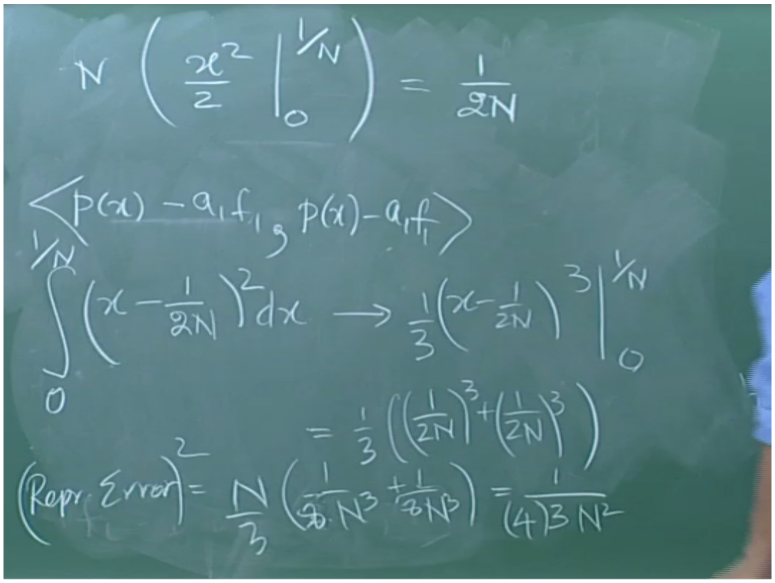
(Refer Slide Time: 32:50)



I am not going to draw an exact figure now this is a little more difficult because there are N of them so it is little more difficult. That is the 45 degree line. So what would you expect and as I said you can check this out what you will get is a bunch of steps in this fashion over each interval it is going to be a constant. You cannot represent the linear it may be a constant however looking at this I would guess that the representation error is quite small.

Looking at this I guess that the representation error is quite small and just as I did earlier I am going to find the representation error only in the first interval and just multiply it by n. So what is the representation error in the first interval? For p(x) - a1* f1 is what we want. What is this representation error? In order to do this, we need to find a1. What is a1? a1 = dot product of P(x) dotted with f1/f1.f = x dx integral 0 to 1 over n/1 over n.

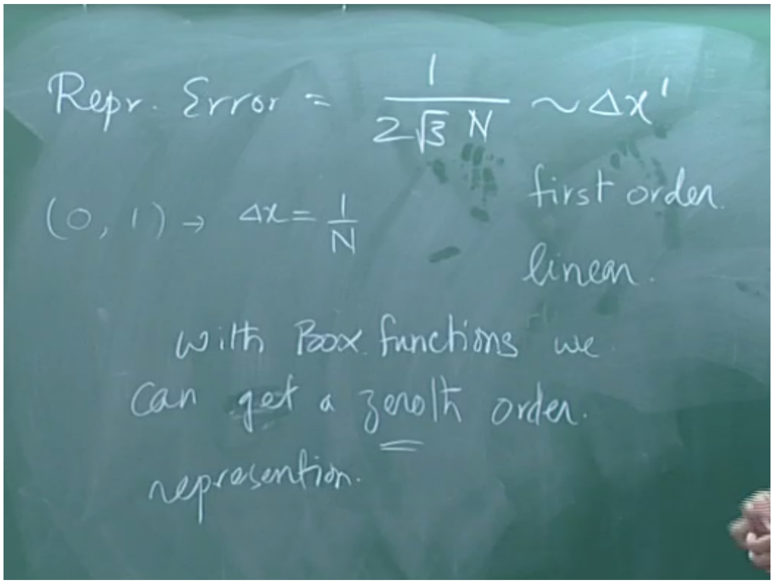
(Refer Slide Time: 35:18)



X square/2 0 to 1/N * N gives me 1/2N because we have general expression and 1/6 everything works. We have a general expression 1/2N. So by a1 therefore is P(x) by error therefore representation error P(x) - a1f1 and you guys are letting me get away with a mistake here. It is better. P(x) - a1f1 gives me x - 1/2N squared the integral between 0 to 1/N is that right. Earlier we had 0 to 1/3rd x - 1/6 just verify that it is okay and this in fact will be x - 1/2N cube one third between the limits 0 and 1/N and that should turn about to be fine.

So the whole interval I will have N of these. The total representation error we just say representation error is do I take this square root first or do I so at the end of these what do I get. N of these N/3 * 1/8N cubed + 1/8N cubed that gives me 1/4 I am deliberately leaving the 4 because I planned to take a square root 3 and squared. This is representation error squared and therefore the square has a representation error is that.

(Refer Slide Time: 38:51)



Our representation error is 1/2 cube root 3 * N. now what we are going to do is we are going to look at what we have managed to do so far and define a few terms. I just gone through and done all the relevant regulation just a repetition of what we did earlier. Of course in the case of vectors we did not really have a representation error. We had representation error in the form of round off error only at numbers.

Now the first thing to note is it gets better as N gets larger that is what our instance tells us that we get closer and closer to the straight line as N gets larger. So that part is good as N gets larger it gets better, but there is a bad part to it. What is the bad part? Number of jumps that you have in your representation will also increase which I am not comfortable. So we may be able to live with it. There are by the way the ways by which you can live with it.

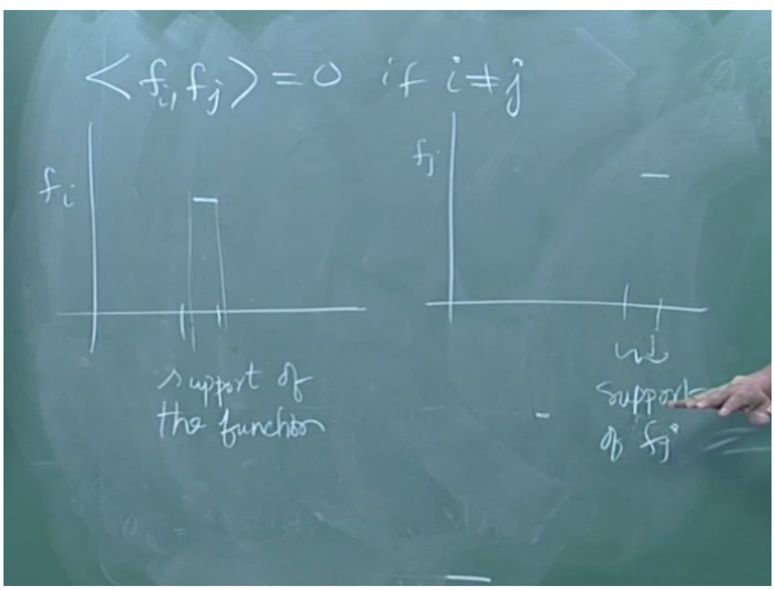
There are things that you can do to get around it, but it is not we asked the question can we do better. So now what have we managed with Box function what have we managed. We have managed to get some way by which we can represent functions. We have managed to figure out some way by which we can estimate the error in that function. So the error in the function is of the order 1/N which basically means that if I am dividing the interval 0, 1 into N parts then I have a delta x which is of the order of 1/N.

So the representation error is of the order of delta x and if I make by delta x smaller and smaller the representation error also gets smaller and smaller in a linear fashion. So the error is of first order. The exponent is 1. The error is a first order and rather casually introducing language here the error is of first order. The convergence meaning as I increase N how close do I get close to solution or to the function that I want how close is the representation get to the original function.

So it converges to the original function because the error goes to 0 and the rate at which it does it is linear. So the convergence is linear. Error is what I want, the convergence is linear. The representation, how well what is the polynomial that you are able to represent exactly. What polynomial can you represent exactly constant. So the representation itself is 0th order. The error is first order, the representation itself.

So we represent with Box functions we can get a 0th order representation. 0th order representation, first order error, linear convergence, 3 different things. Let us look at the orthogonality business. We need to from this in order to talk to converse about these functions we also need to define some terms. It is where did this orthogonality come from?

(Refer Slide Time: 42:40)



Where did we get this? Fifj = 0 if I != j. why did that work? How did that happen? Why does that work? So that works simply because if you have 1 function which is nonzero on the ith interval and you have another function which is nonzero on the jth interval then if you have to multiply the 2 this is nonzero where this is 0, this is nonzero or that is 0. So this business where is it 0, where is it not 0 we need. We need some language for this.

So the interval or the region where the function is nonzero is called the support of the function. So the support of the function is that part of the domain definition where the function is nonzero from the support of the function of fj. Right it is the support of the function. So what we have done is we got orthogonality fi.fj has been 0 when i != j from the fact that the support of the 2 functions are non-overlapping in the sense we support to sort of orthogonal.

Supports for non-overlapping you understand. This contrast with if you seen Fourier series before sin x and cosine x on the interval 0 to 2 pi are orthogonal to each other, but they define, they are both nonzero almost everywhere on the interval 0 to 2 pi is that clear. Whereas here we have achieved orthogonality by basically saying that the support of this function is different from the support of that function. They are non-overlapping that is important.

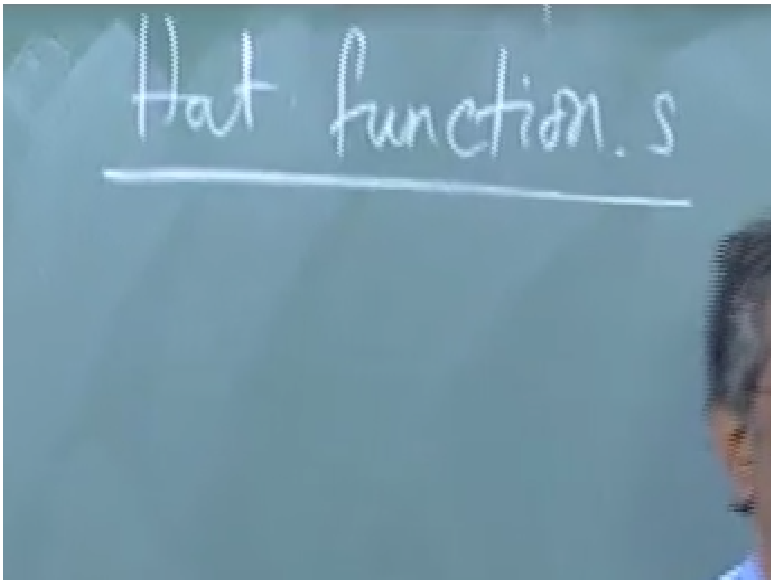
They are non-overlapping and therefore we have orthogonality that is 1 thing that 1 critical thing that we have to notice from here. The second thing is though the function gets closer it gets jumpier. So we have to ask ourselves the question is there a way by which can we come up with something that will give us smoother functions. We have constructed these functions Box functions now, but is there a mechanism by which we can get smoother functions.

So, non-overlapping will give us orthogonality. In some sense and, if we go for smoother functions. How do I get smoother the representation here is 0th order. I would like to represent at least the first order which means that I should use some kind of linear interpolate that is the possibly. The other question that you could ask is why bother with all these little bits and pieces f1, f2, f3, f4 and so on why not we use polynomials directly would it be just smoother to use just polynomials directly.

So there are 2 possible parts that we can take having mentioned Fourier series why get the orthogonality from non-overlapping domains, non-overlapping supports right the domain is overlapped, but non-overlapping supports. Why get orthogonality from that? Why not just get orthogonality from somehow to construct it using the Gram-Schmidt process or something of that sort why not just get orthogonality directly from the functions.

So there are 2 possible ways by which we can go. What we will do is we will try to follow this linear process right now. If we have the Box function, we will now try to get go to a linear mechanism we will try to use this non-overlapping we will try to use these non-overlapping functions supports for the function and see if it is possible for us to generate linear interpolates. Is that fine? Are there any questions? So in tomorrow's class we are basically going to look at we will start with Hat functions.

(Refer Slide Time: 47:10)



What are called Hat functions where we will use linear interpolates as the basic mechanism.