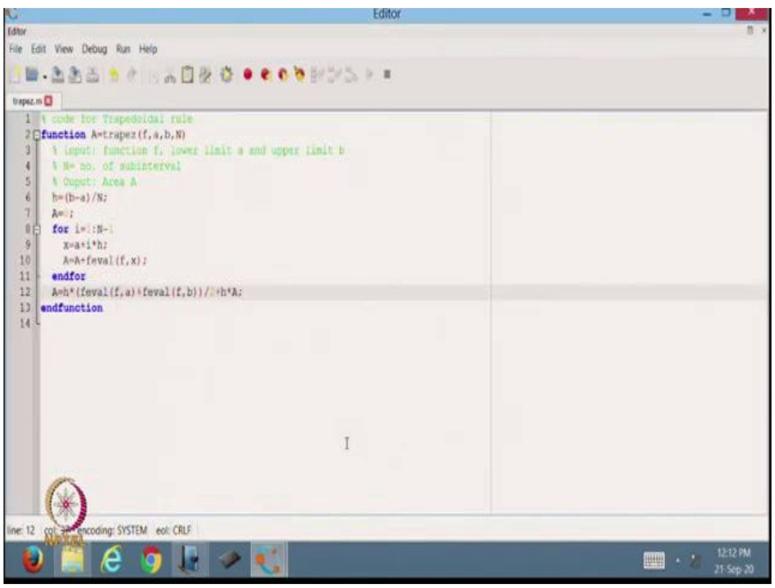
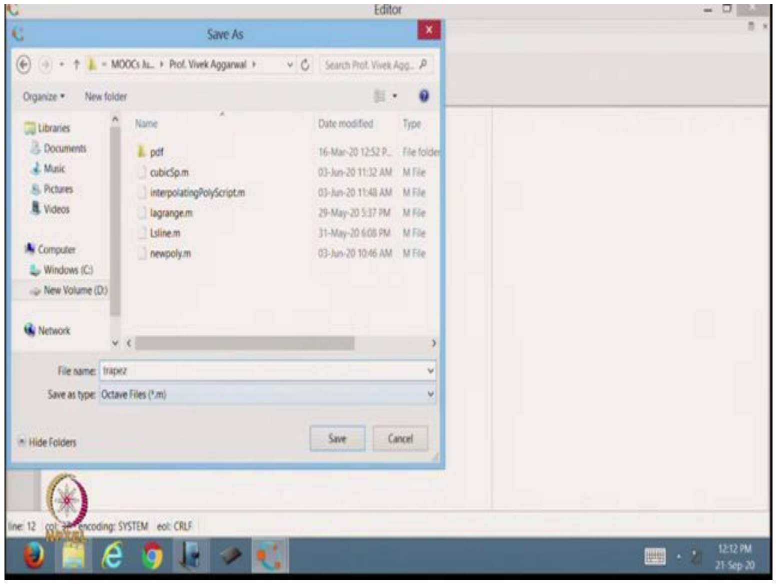
Scientific Computing using Matlab
Professor Vivek Aggarwal
Department of Mathematics
Indian Institute of Technology, Delhi
Professor Mani Mehra
Delhi Technological University
Lecture 60
Octave Code for Trapezoidal and Simpson's Rule

Hello viewers, welcome back to the course on Scientific Computing using Matlab. So, in the previous lectures we have discussed the numerical integration with Trapezoidal rule, Simpson one third rule, Simpson 3 by 8 rule and then we have also discussed the method of undetermined coefficient. So, today we will make the Matlab octave codes for that one. So let us go to the octave.

(Refer Slide Time: 1:20)





So, this is the cursor for the octave. Now, we can start a script file, so this is the script file. So, today we will make the code for Trapezoidal. So, I will write the function. So, let us make the function. So that function gives me the value area that it will determine, so I call it Trapezoidal. So, I just write the short form Trapez and then I will pass the value of the function, then the limits, lower limit, upper limit and the number of sub-intervals so that I am going to pass from here.

So, here I can write that input or f the function that is my f, then lower limit that is a and upper limit that is b. so this one I am going to input. And the output will be, so that output will be given by the area that is a and this is also here we have to, n is equal to the number of sub-intervals. So, this one we are going to define. Now I will find out the h, so what is the h, so h=(b-a)/N.

So, this one I am going to define. Then I define my area capital A=0. I start with this value is equal to 0. Then I will define the loop, for loop, so for I just define i is equal to from 1 to n-1, so in this case I am defining, the nodal values I will define, so $x = a + ih$, so that is my nodal value, because in this case if I put i=1 it will be a+h, i=2 it is a+2h.

Because this is uniform nodal values and then I will define capital A, this is the area,

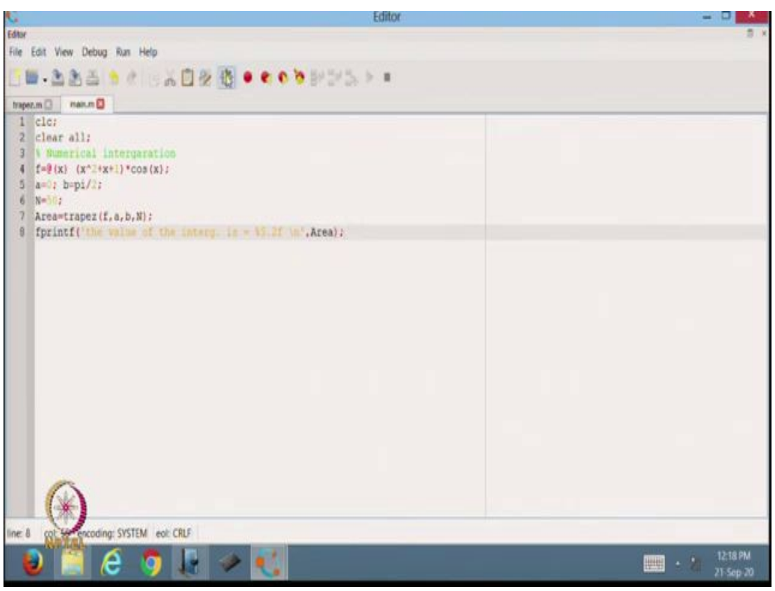
$A = A + feval(f, x)$, so this will be… So, from here I will get the value of x and this is the value of the a, I am getting because it is added, I started with the a 0. So initially it will be a plus, so this will be 0 plus value of the function at x that will be added, then the next will be added, so on, so for the for loop it will keep adding this value for the value of x and that is the value of a.

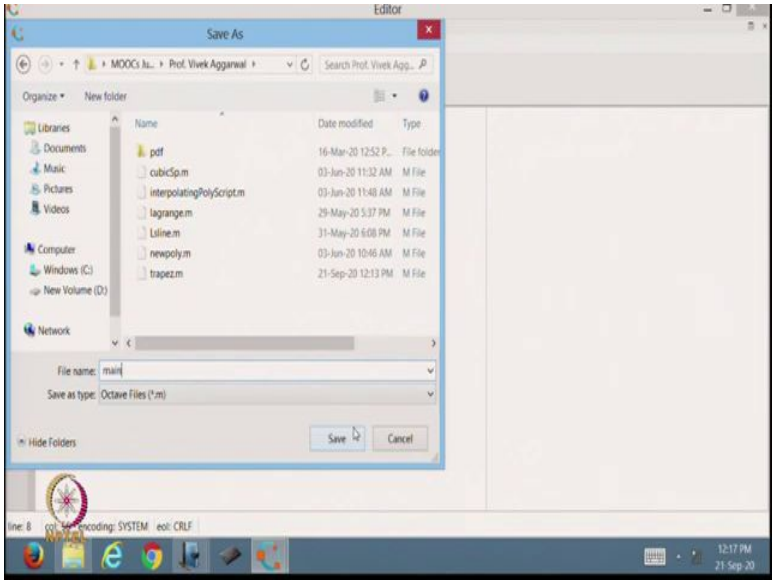
So, this is the end and after that I just pass a is equal to, because here I am using that composite rule so I know that this is equal to h multiplied by f the value of the function at the initial point plus f evaluate the function at the last point divided by, so this is we know that we have to take the average of this value plus and then I can define the, because it is h by 2, so h by 2 we are defining here plus h by 2, so this is h by 2 multiplied by the value of a.

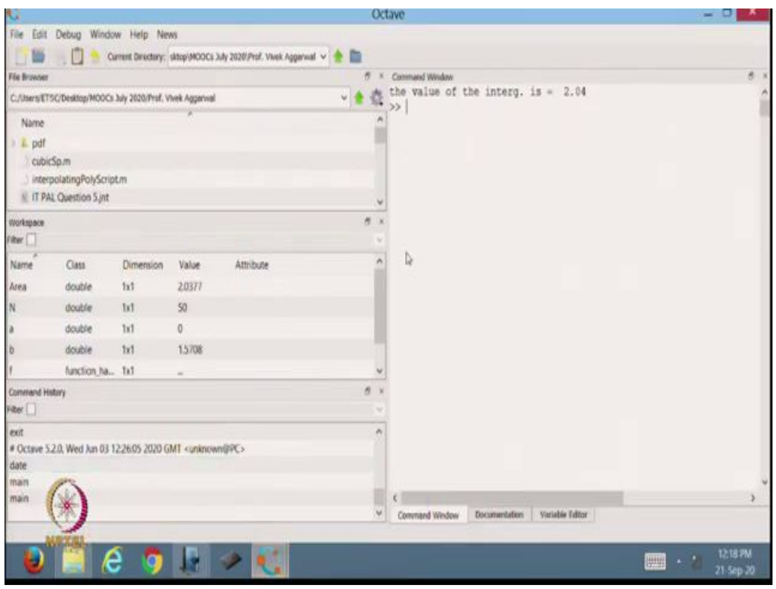
The 2 times of this one, so that will cancel out and this will be equal to a star, so this is the value we are going to get, so now I, from here I defined the function, this is the input values we are introducing here and that is the function here. So I can save this function, save file so let us save it on the desktop. So I just save it at the desktop yeah, so here I am saving. So, this will be saved as the name and the name is trapez.m. So, this will be saved.

So that is the t-r-a-p-e-z, so this is the function I have saved there. Now if I open again. So, let me cancel this one and then, I know that this one we cannot run because this is a function, so I have to give the input value here. So, for the input value what I do is that I define the, suppose I define another main, I should write the main script for this one.

(Refer Slide Time: 8:04)







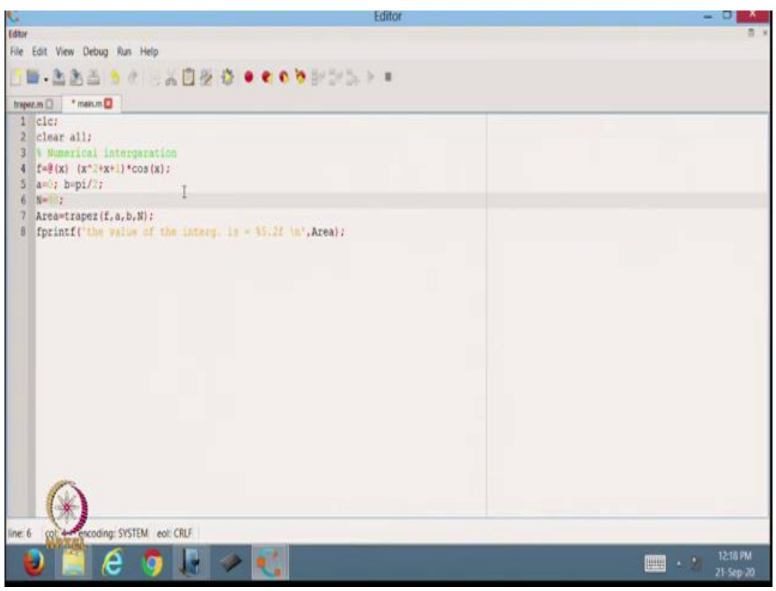
So, in this case I just write Clc and clear all, and in the end I will define that this one I am defining for numerical integration so this one I am doing for numerical integration. Now, first I have to define the f, what is my f? So, f I will define as a, because I have to pass this one so this is just the handler I am giving, the function of x. So, let us take the function. This is I am defining $f(x) = (x^2 + x + 1)cos(x)$.

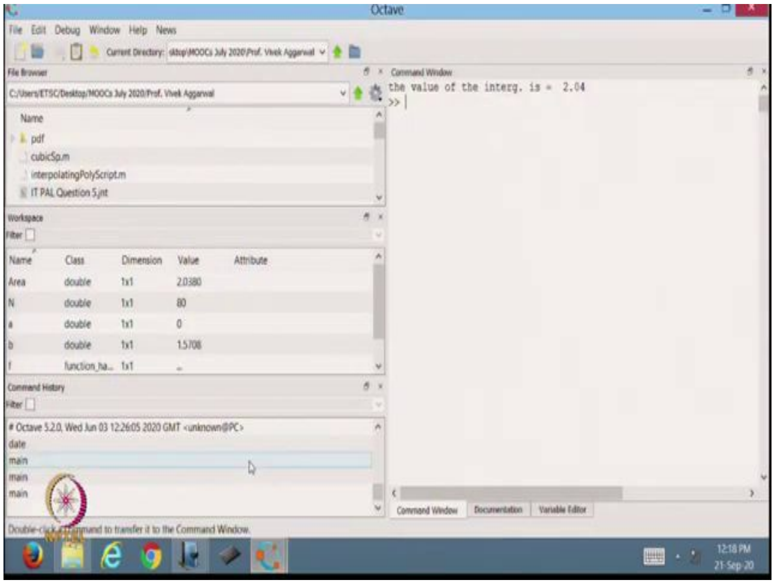
So, this is the function I am defining. Let us say take a=0 and b=pi/2. I define n as equal to because in this case no problem, I can define n as equal to maybe 50 Trapezoid rule, then I just call the function so I am calculating the area. So this is I am defining trapeze, because I am defining a trapeze passing f, a, b, and capital N, so this I am passing. So, whatever the value is coming I will just write fprintf.

So, this is the value I am writing fprintf, the value of the integration is, so this is just defining. So it is equal to, I am defining just 5 points, maybe 2 floating points, then a new line, this and that is the area. So this is the fprintf I am defining. So, I am saving it because we have to save it. So, let us save it where we want to save, so save the file and again I have to go to the previous file, MOOCs, then here, so I will call it ‘main’ that’s it and let us save this one.

Now, I just run this one. So let us run this one, change directory, yes, and after getting this value, so this is the error it is coming to. So we have defined the area wrong in the main file, so let us correct it a-r-e-a, area that fprintf here, so that is the area I am defining. Now we can just save and run. So, let us see. So, the value of the integration coming is 2.04, so that is the value it is coming with the trapezoidal rule.

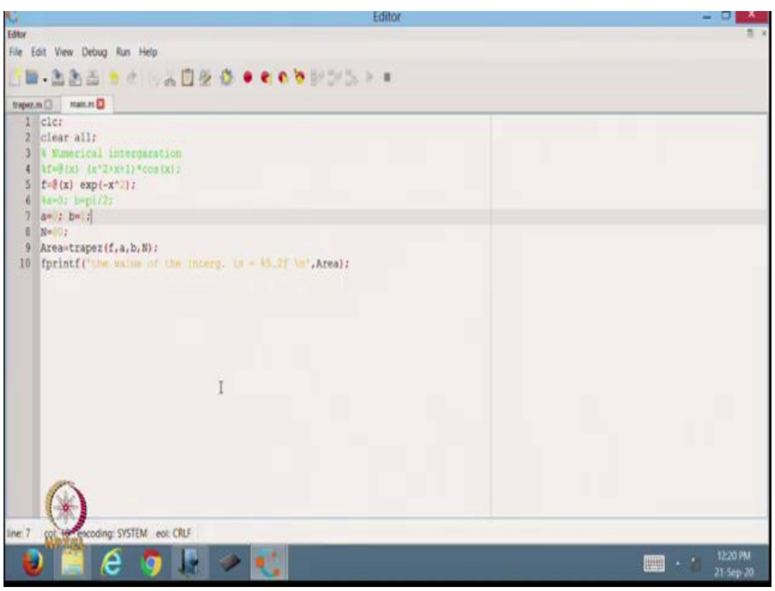
(Refer Slide Time: 12:24)

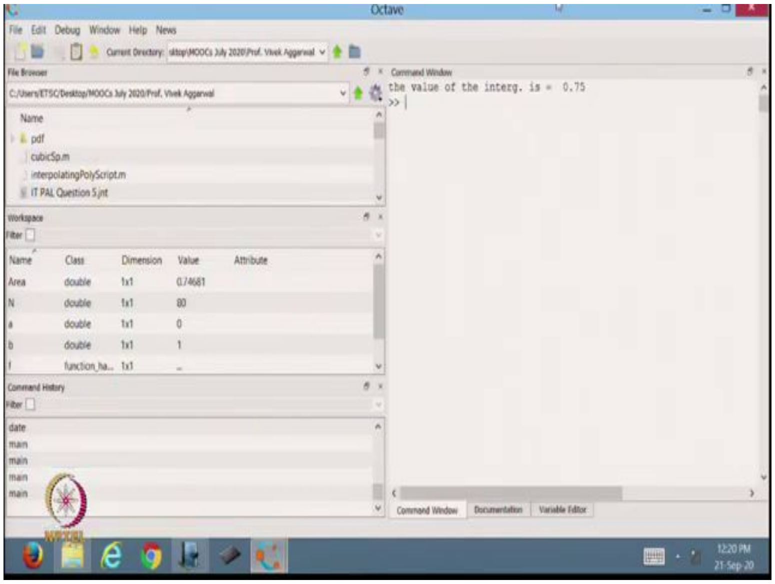




Maybe I can change this to n is equal to instead of 50, I can take n is equal to maybe 80, the number of points I will increase, so let us see what will happen in that case. So yeah, so it is again the 2.04, so it means that this is the exact value of the function, approximately the exact value of the function for this integration, 2.04. Maybe I can change this one and I can change the function here, so this function here maybe I can just define the next function.

(Refer Slide Time: 13:07)

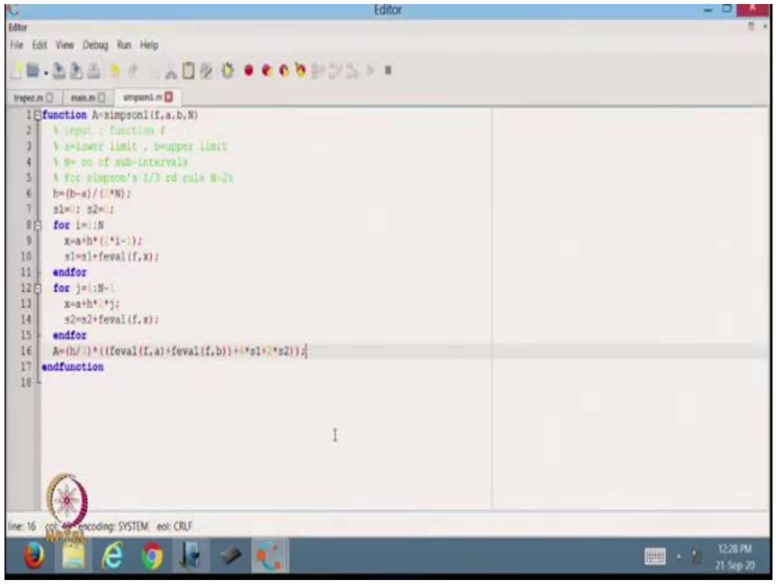


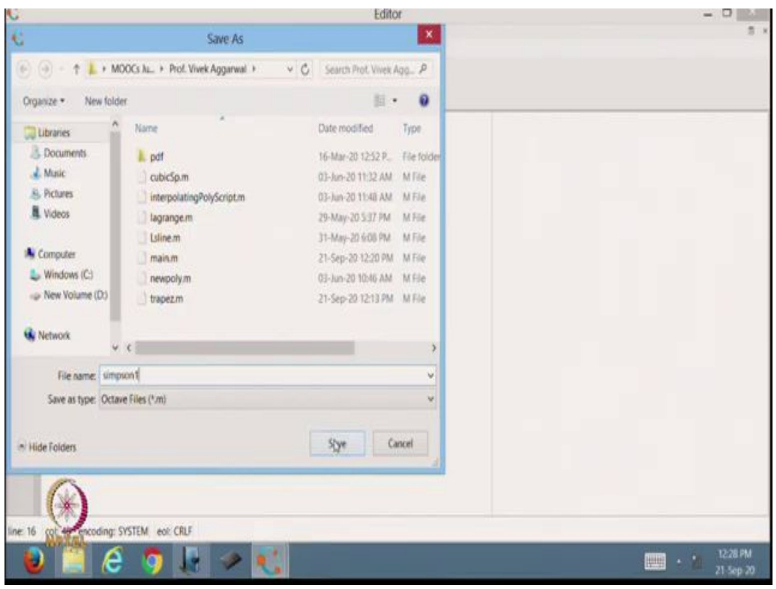


Maybe I can write f is equal to exponential function because that I know, that it is very difficult otherwise to find out the solution, so let us take this exponential function. So in this exponential function I am taking b=a=0 that is okay, but instead of taking here, I define my a is equal to, may be a=0 and b I just defined 1, so let us see what will happen and the number of mesh point I am taking 80, so no problem in this case.

So, let us see what will happen. So the value is coming 0.75. So this is basically the integration from 0 to 1 e raised to the power minus x square, the Gaussian function, and the value is coming 0.75. So, in this way we can define the different-different type of functions and that is the trapezoidal rule. So maybe we can define another function. So, let us define another function.

(Refer Slide Time: 14:33)





So, now we will do another function for Simpson maybe, so let us take another one. So I define the function, the area I just defined, the same as Simpson. So, I am defining Simpson 1 and again I am passing f, a, b, and N. So that we have to do, pass this value. Then I know that input is again, because you have to write as a comment that if you open this code after a long time, then you should be able to see what is written in this code and what is the use of this code.

So, in this case input I am just writing again, the input is f, the function f, a is lower limit, b is equal to upper limit, and then the n is the number of mesh points, number of maybe I can write sub-intervals, so these are number of sub-intervals. Now in this case I know that if I apply the

Simpson one third rule then I should write that for Simpson one third rule and should be multiple of 2, so that should be there, only then we are able to divide the given domain in the required number of match points.

So, in this case I know that the N should be equal to two k, so it should be even basically. So in this case what I am doing, I just apply, find the h, so h I am taking b minus a that is divided by 2 star N, so I just write it 2*N, so whatever the N we are passing I just multiply by 2 times, that is it. So maybe, then there is no problem. Suppose I just apply N is equal to 5, then I will get 10. So, 10 is an even number, no problem.

So I just do like this one, then I define the sum, so I define the sum s1=0, the initial sum, s2=0, because in this case we have need to find the two type of sum, one sum is multiplied by 4, if you remember the Simpson one third rule, we have to apply this, we have to find the 2 sums, 1 sum is multiplied by 4, and other sum is multiplied by 2. So, now we define the for loop for i is equal to 1 to N, so this one I have to define.

Now, $x = a + h * (2 * i - 1)$, this one I am defining. So when i=1, so this is i equal to 1, it will be a+h, when i=2 it will be a+3h so 2 into 2 is 4 minus 1, 3, so a+3h, then i=3, it will be a+5h, so basically you are getting the value at the odd values, a+h, a+3h, a+5h, like this one and then I define my sum $s1 = s1 + feval(f, x)$, so that we are calculating.

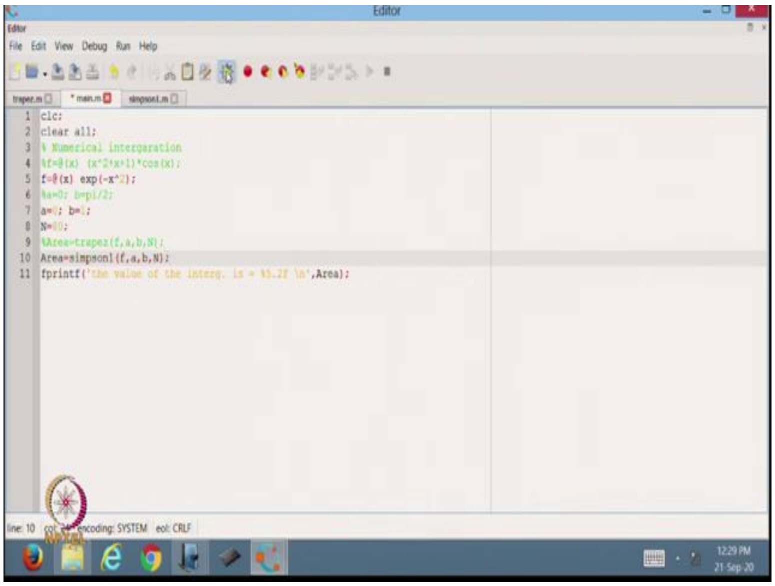
So, this is my sum giving the values at the odd number of nodal values, so basically it is a+h means x1, a plus 3 as x 5 like this one. Now I define the another loop i that is j from 1 to n-1. So this one I am defining, then $x = a + h * 2 * j$, because here I am defining when j=1 it will be a+2h, then j=2, it will be a+4h, so it is the even value of the nodal value.

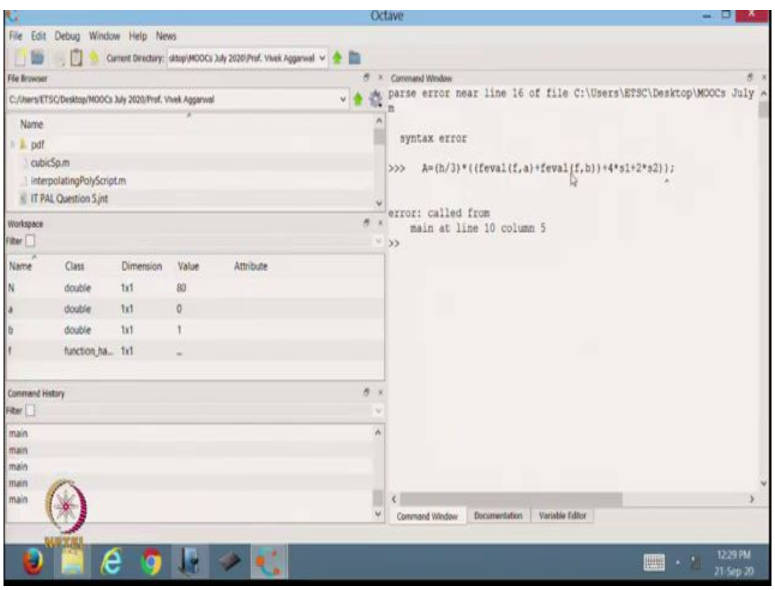
So, this I am defining a+2*h*j, and then s2=s2+feval(f,x), so this is the value we are defining and that is the end of this for loop and then I can pass my area, so A will be, now you can know that this is 3 by h 1 by 3 rule, so I will define

$A = (h/3) * (feval(f, a) + feval(f, b)) + 4 * s1 + 2 * s2$, this is the value I am defining.

So, it is a 4 times s 1 because we are getting the values at the odd values and 2 times the s 2, where the we have a overlapping nodes, so that is the even values, so we will get the value at x2, then x4, then x6, like this one. So, this is the way we are defining this function, so this is a Simpson one. Now, I just save it and this will be Simpson one that is it.

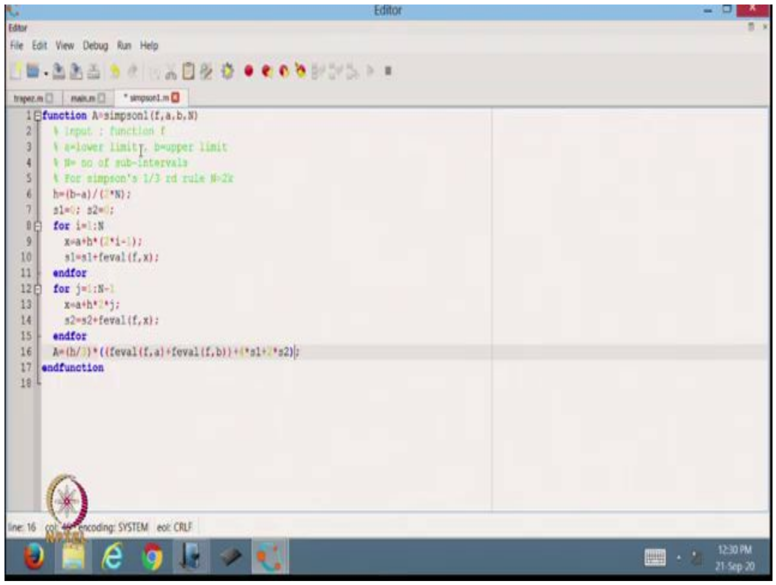
(Refer Slide Time: 22:36)

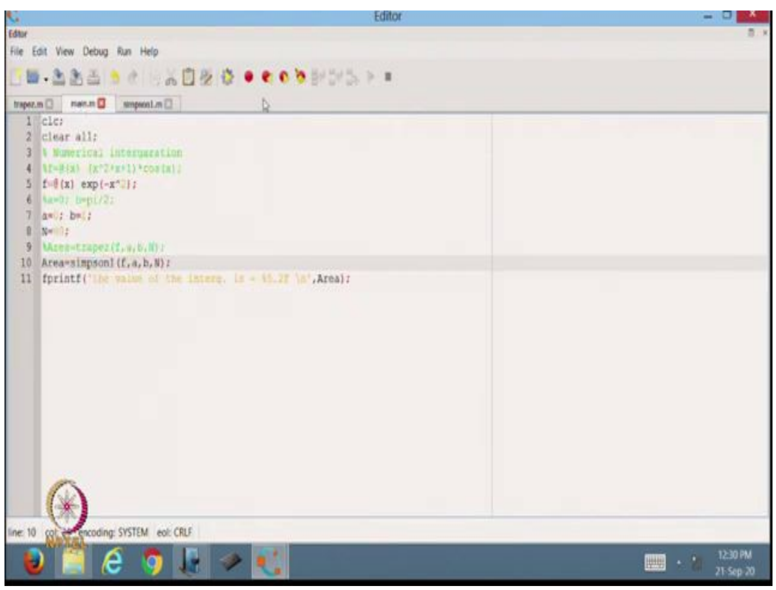


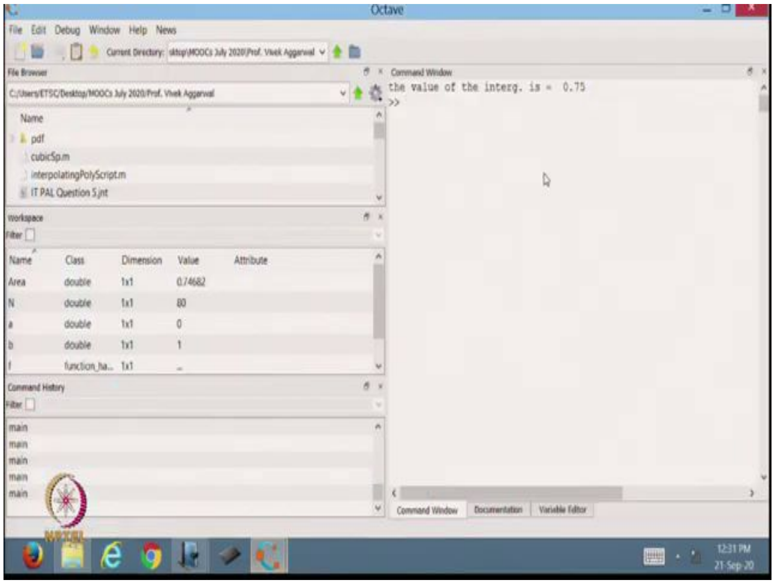


So, now I just use the same main file and let us see what will happen. So, the same file I am using, just I will change the name here, so in this case, this is my area. Now I define this value and define the area that is equal to Simpson 1, f at a at b and this is the number of points and so that is it. The only thing I have changed is the calling function that is it, so let us see what will happen. It is solving or giving me some errors. Yeah, so it is giving some error that some extra bracket is there. So, let us see.

(Refer Slide Time: 23:42)

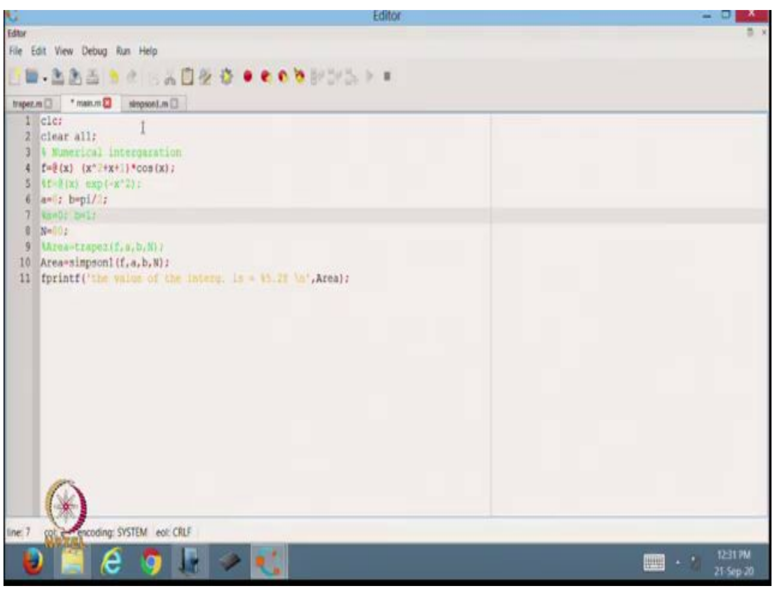


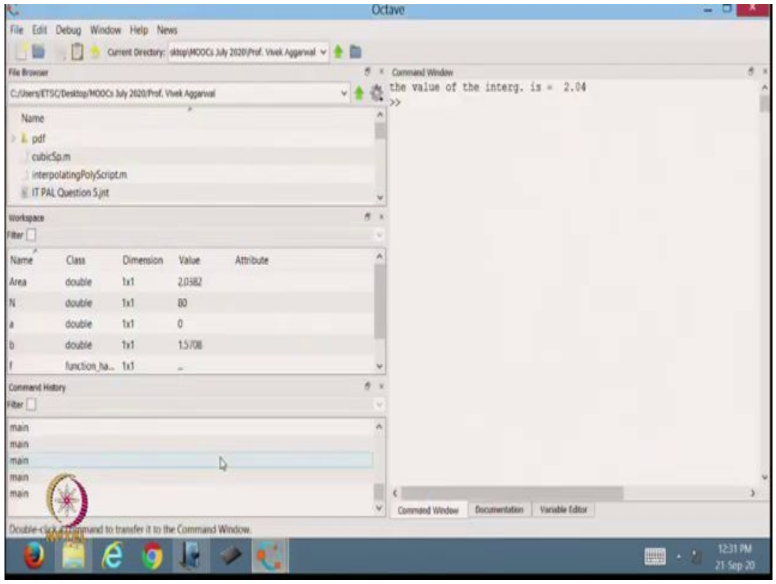




So, let us see Simpson 1, so in this case I am getting the value of the function at a plus the value of the function at b this is ok, plus 4 times s 1, plus 2 times s 2, this is the extra, so now it is okay. So, I will just save it. Now let us see and let us see whether we are getting the result. Yeah so we are getting the same result 0.75.

(Refer Slide Time: 24:38)

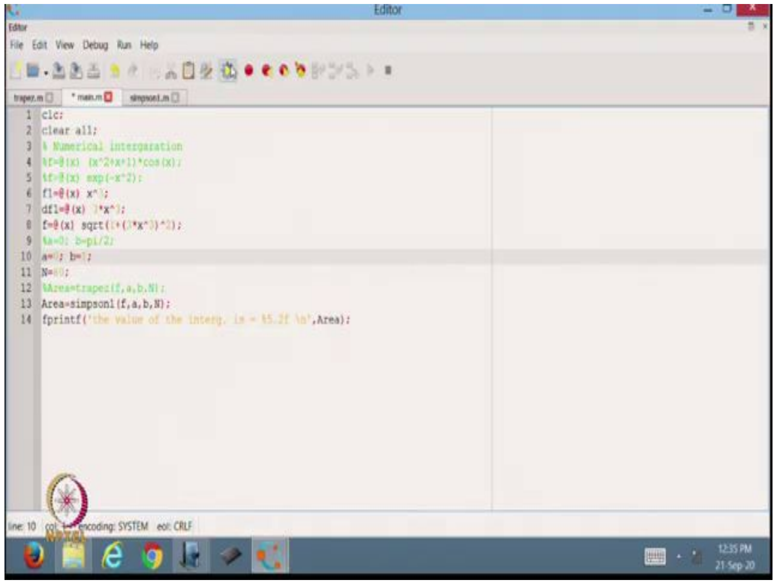


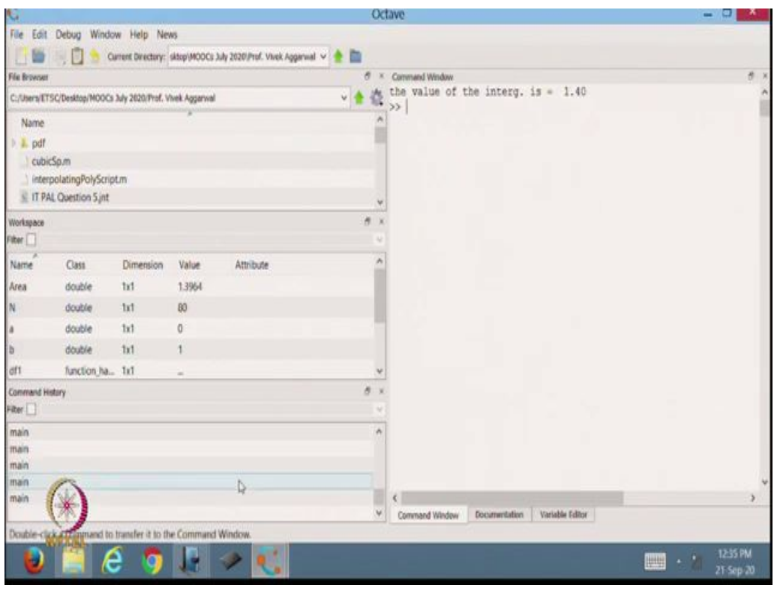


So now we can even let us see what will happen when we are getting the same result for the previous case. So let us take this value here. I take this f and comment this one and now I am so changing the value of a and commenting this one, so let us see what will happen. So in this case value should be 2 point something, yeah so it is 2.04. So from here also we are able to solve the (syst), this integral with both the method Simpson one third rule or trapezoidal rule.

So in this case and I also know the Simpson one third rule is the same as the Simpson 3 by 8, so this will also work for that. So maybe this is used a lot whenever we are finding the length of the curve. So in the case of the length of the curve we also need to find the integration and this will be very useful whenever we are dealing with finding the length of the curve or the surface area, so the numerical integration is essential in that case.

(Refer Slide Time: 26:05)



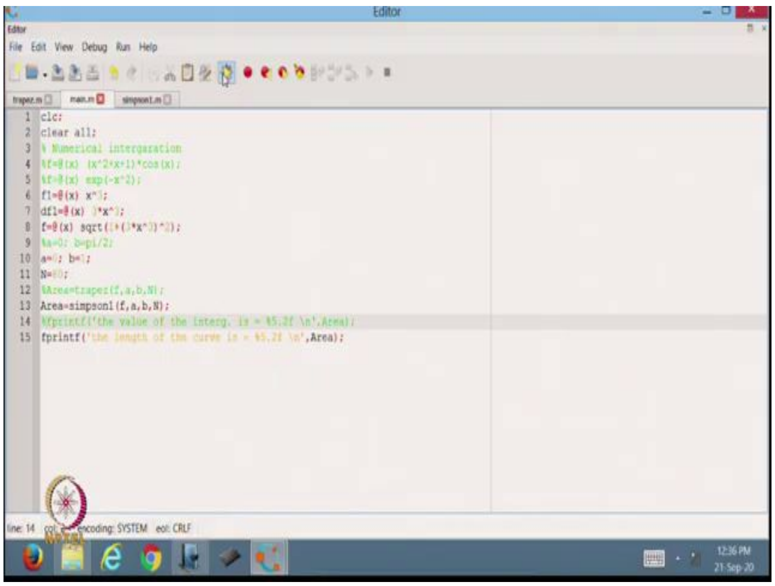


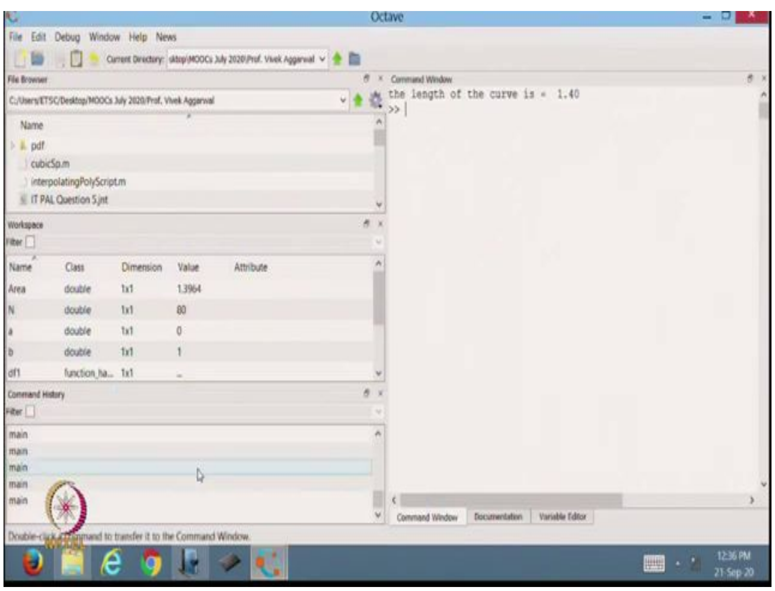
So, and everything we are doing is one dimensional, so maybe we can take another function instead of if I just take another function, f is equal to maybe I just take the simple function x raise to power, x raise to the power q, so this is the function I am defining. Then I define the function, so I just define it f 1 and then I define f is equal to the derivative of this one, so it will be again 3 multiplied by x cube. I am taking the derivative.

And then this is I just take note the f, I just write d f, the derivative of f 1, yeah so this is what I have defined. Now what I am going to do, I am finding, going to find out the length of the curve. So now I define f is equal to because, no, I know that the value of the function, so f will be the

$f(x) = 1 + sqrt(1 + (3 * x^3)^2)$, so that is my function 1 plus f dash square under the root, and I want to find its length. Maybe I should just change it from 0 to 1. And I am applying the Simpson one rule, so this will give me the length of the curve from 0 to 1 for the function x cube. So, let us see what will happen? So, it will give me the value of the, so that is 1.40, so that is the length of the curve from 0 to 1 for the function x cube.

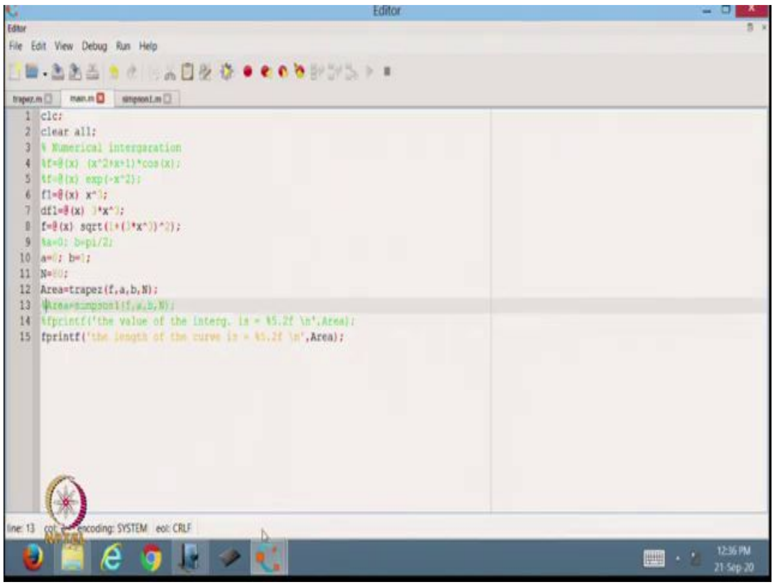
(Refer Slide Time: 29:06)

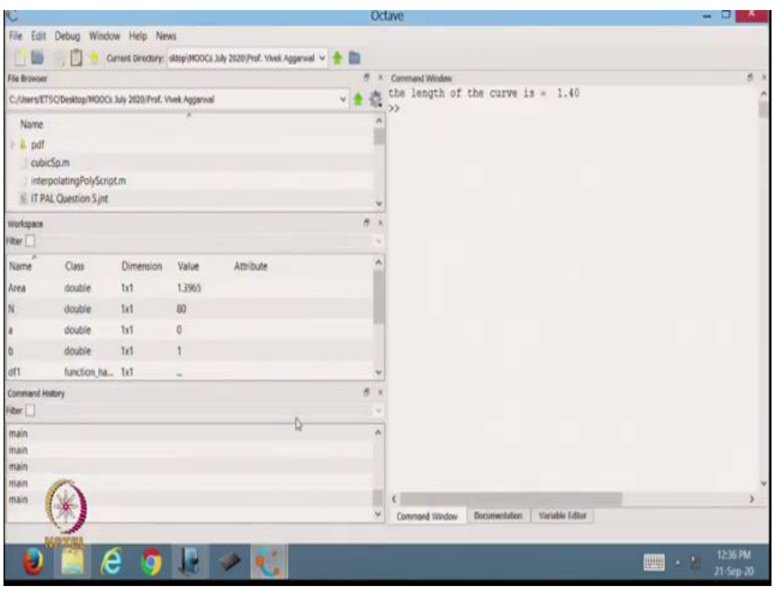




So, here I can write the value the integration so just instead of this, I just k, control c and then I control v and then I, from here I write the length of the curve is this one, so in this case it gave me the length of the curve that is it and if I run this one, I will get, so the length of the curve is 1.40.

(Refer Slide Time: 30:01)





I can do the same thing with Trapezoidal; let us see what will happen. So I should get the same value so it is 1.40. So it is giving me the length of the curve for both the methods trapezoidal or Simpson. So, I think now I should stop here. So today we have discussed how we can make the functions code for Trapezoidal and Simpson and we have been able to verify that it is giving the results for both the methods.

So, by this way we can find the, we can make the code for other numerical methods also. So this is just a primary way that we can use the Matlab code for numerical integration; so this is all about the method based on numerical integration, so that is all about the numerical integration. So thanks for watching and thanks very much.