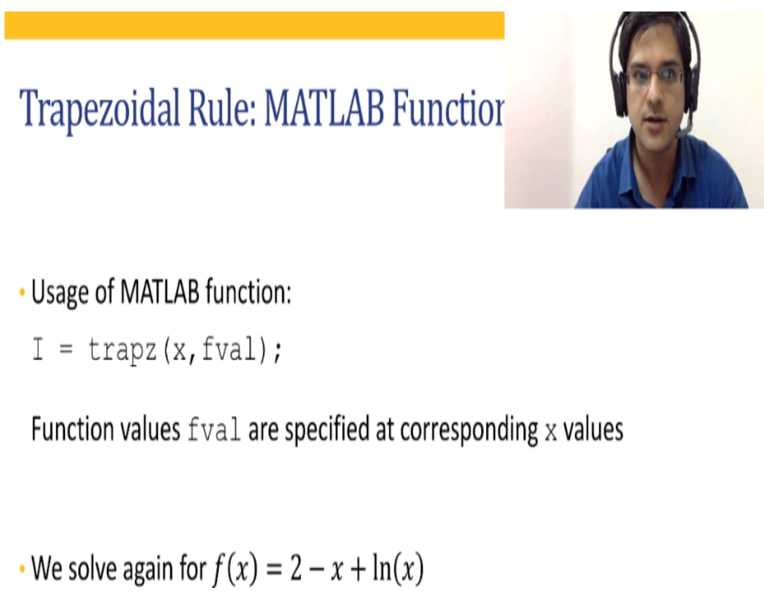
Matlab programming for numerical computation
Dr.Niket Kaisare
Department of Chemical Engineering
Indian institute of technology, Madras

Module No. #03
Lecture No. #3.6
MATLAB Functions and Application

Hello and welcome to MATLAB programming for numerical computations. We are in module 3. In the second part of this module we have been covering numerical integration. This is the last lecture in this module. In this lecture, we will consider MATLAB functions that can be used for numerical integration. Specifically, we will take 2 functions trapz and quad.

We will also look at solving an application problem using numerical integration. In the previous lecture, we have considered trapezoidal rule. We have considered both the single application of trapezoidal rule as well as multiple applications of the trapezoidal rule. We wrote our own code in order to compute the integral using trapezoidal.
(Refer Slide Time: 01:05)



Today we are going to use a MATLAB function called trapz. In order to compute that integral using the trapezoidal rule okay. The usage of this MATLAB function is, integral i is trapz x, fval where, x is the independent variable and f is f calculated at both various values of x.

x in this case is going to be an n+1 dimensional vector and fval also is an n+1 dimensional vector such that the function values fval are specified are the values that are specified to the corresponding x values. What that means is the first guy in fval is f(x1), second guy is f calculated at x2, the third guy is f calculated as x3 so on and so forth.

So let us look at the example that we took in the previous lecture and solve this using the MATLAB function trapz. (Video Starts: 01:58) So this was the script that we wrote in the previous lecture. What we did was we specify the problem statement over here. Then we used multiple applications of the trapezoidal rule and did that in 2 ways.

The first one was to calculate trapezoidal rule by summing from n intervals, the second way was we drive a direct formula for using the trapezoidal rule and use this as shown over here. The third way is to use the trapz function and I will display. I will show that today using MATLAB and trapz function okay.

Let us do help trapz okay. And if you look at the help for trapz, your integral is trapz of y. If you have your x as unit spacing, that means if x is 1, 2, 3, 4 and so on up to n+1. Then we can use z = trapz y. However, we know what our values of x are. And they are not given by unit spacing.

Therefore, we need to use the command z = trapz x, y where, x are the independent variables and y are the corresponding values of the function. (Video Ends: 03:31) So this is exactly the usage that I have shown over here. (Video Starts; 03:36) So what we can do is, we can just write that I_trap3= trapz xVec, fVec using trapz function disp I_trap3, err3 it will also display error 3.

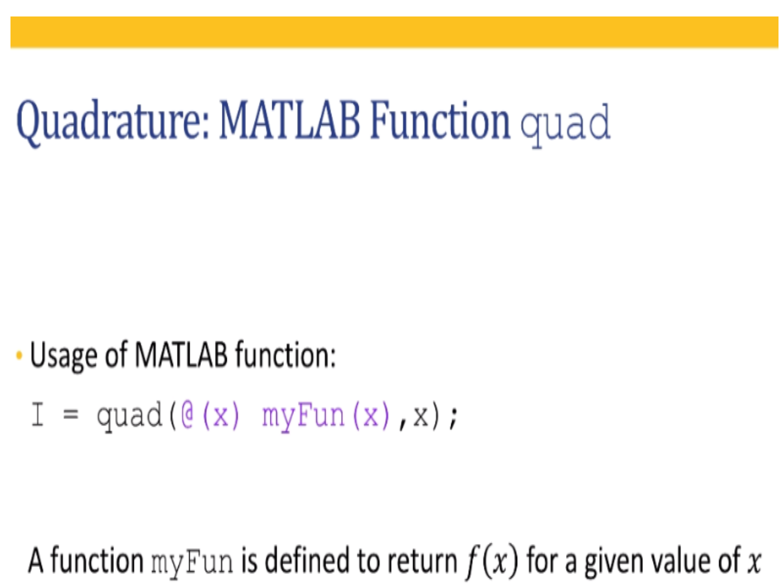
And let us calculate err3. err3 is abs (trueVal- I_trap3). Let us save this and let us run this function. Let us run this script rather okay. We have run this and we can see using the trapz function. We get the same integral value as we had gotten before and the error that we get that is trueVal- that itrapzval is also the same as before.

So as you can see it is very easy to implement or apply this trapz function okay. Let us, let us look at writing the same multistep integral from scratch, I will say edit multiStepIntegral2 okay. Redo the previous problem. I am doing it again just for sake of clarity, a was 1, b was 2, n was I believe 200.

Yeah n was 200 and our function was given by this okay. Our h was (b-a) divided by n, our xVec was a in steps of h to b, our fVec was myFunInt of xVec okay. And i was equal to trapz of xVec, fVec. We just save this and run this and we will see the results.

So i was 0.8863, abs (trueVal- i) you will see the error is the same okay. So what I have done is, I have shown you how quick it is to use trapz function in order to calculate the integral using the trapezoidal rule okay. Let us save this and let us close this function okay. (Video Ends: 07:04).

Now we have finished using the trapz function that was the first Matlab function that I wanted to cover in this lecture.
(Refer Slide Time: 07:11)



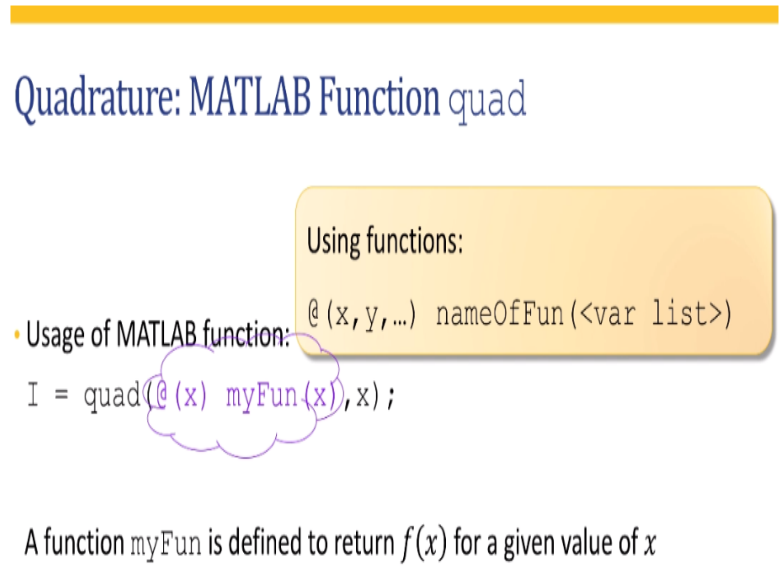
The second function that I want to cover is what is known as quad. In quad in the above in the method quad what we actually do is, we pass on our function file itself to the function to the MATLAB command quad. How we do that is, what is known as the anonymous functions okay. At x means x is the variable and myFun is the file that calculates the function f(x) okay. So the function myFun returns f(x) for any given value of x.

This x is specified in the brackets inside this at sign and the myfunction is the name of the file that we are going to use in order to pass on that function okay. This x is the values at which we are going to calculate our integrals okay. Now using the function this is going to be the common syntax of using any function that we are going to do you pass on as an argument to a MATLAB command.

(Video Starts: 08: 25) Okay let us go to MATLAB and look at this now. I will do help function handle okay. Function handle is obtained by giving at function name or at arglist. That means list of arguments followed by that function name. What that means is this is, let us say we want to pass on a function.

What function we wanted to pass over here. We wanted pass on the function myFunInt which is a function of x the way we pass this on is at x myFunInt x okay. So this specifies the, that x is the variable that is required in the function, this is the name of the file containing the function and this is exactly the same as this particular guy.

If we have a function of 2 variables, say x and y the way we will write this is, myFunInt x, y okay. That is how we are going to pass on the functions to MATLAB commands.
(Video Ends: 09:48).
(Refer Slide Time: 09:49)



That are you going to use these functions okay. The command that we are going to use in order to calculate the integral is known quad. (Video Starts: 09:58) So let us check help quad okay. quad will you give the integral of a function specified in a file fun from a to b. We have our function f(x) that is specified in myFunInt.

So at x myFunInt x, that is how we are going to pass on the function handle to the command okay. So our i is going to be quad function name, initial, the lower bound, the upper bound. This is how we are going to run our quad command. Our lower bound was 1 and upper bound was 2 and we run this I, and we will get our integral.

If we recall from what we did just a couple of minutes back, the value of integral the true value of integral was indeed 0.8863. This is the way to use quad. (Video Ends: 11:15) What I am going to do next is, we will take up one more example and that is going to be a physical example that we take.

(Refer Slide Time: 11:22)]

Reactor Problem

- Let us consider the reactor design problem from Computational Techniques (Module 6, Part 4):

$$V = \int_0^{\text{conv}} \left(\frac{F}{k}\right) \frac{dX}{(1 - X)^{1.25}}$$

for $F = 10, k = 5$ and $conv = \{0.5\}$.

So what we are going to do is, find this integral from 0 to 0.5 of this particular function. Where, the function is f/ k * 1, divided by (1- x) ^ 1. 25. Where f is 10, k is 5 and we want to integrate from 0 to 0.5 okay.

(Vide Starts: 11:46) So let us go to MATLAB and see how we want to do that. We will create a function edit myReactorFunction okay, function fval = myReactorFun (x) okay. Capital f and small k are the values that we need to give F 10, k is 5 okay. And the function is f/k fval= f/ k, the whole thing divided by (1- x) ^ 1.25. That is going to be our function. Now how do we vectorizer this function?

So let us say we were expecting our x not to just the scalar x but to be vector x. So we want to make this into more vect allow x to be vector as well. How we are going to make this into a vectorizer function. We just make a dot slash over here because this guy is a vector and I will also got dot carried over here okay. And I saved this as myReactorFun.

How are you going to call this particular function or pass on this function to MATLAB commands using at x myReactorFun(x). This is going to be this entire, thing is going to be the function handle. That we are going to use. So what we do is v=quad. This particular guy which is basically at in bracket.

The variable x and followed by space and the file name with that same variable inside the bracket. So that is what is going to be and the limits. If integral from 0 to 0.5, so you go over here, so limits of integral are 0 to 0.5 okay. And if we press enter and this is going to be volume of the reactor that is required to obtain conversion of 0.5. (Video Ends: 14:46)

So with that I come to the end of this lecture. What we did in this lecture basically covered 2 different MATLAB commands in order to finding integral. The first 1 used the values of function at corresponding values of x in order to give the integral using trapezoidal rule that function was known as trapz. The second function that we used was quad, we use the function quad in order to find the integral of any function.

In using the function quad, we passed on the function handle using this particular method of passing the handle okay. So with that I come to the end of this lecture and indeed to the end of this module. In module 3, we covered numerical differentiation. In first 3 lectures, in that we covered first derivatives and higher derivatives f d, df / dx, d^2f / dx^2 using forward, central and backward difference formula.

We also saw that there is an optimal step size for which the error is minimum. We also looked at partial differentiation how to do. In the second part of this module. We covered numerical integration in lecture 3.4 be covered trapezoidal and Simpson's rule. Single application, we extended it to multiple application of trapezoidal rule.

In case we divide the overall rule domain into multiple intervals and applied trapezoidal rule. And finally, in today's lecture we covered 2 MATLAB functions called trapz and quad in order to compute the integral. With that I come to the end of this lecture and the end of this module. See you next week, thanks and bye.