VIDEO SCRIPT

MATLAB SYMBOLIC MATH TOOLBOX

TERM PROJECT CS177: Mehreen Asad

Due Date: 05/09/2011

I explored the Symbolic Math Toolbox in MATLAB for my Term Project. I found it pretty interesting and as I could not cover all of the features of the package due to time constraint, therefore I decided to start off with the basics of the package and then focus on Symbolic Integration since we didn't get a chance to discuss Integration in the class.

In the tutorial, first basic concepts of the package are introduced and then Symbolic Integration is discussed. For creating a symbolic object, we need to declare an object by using the "sym" or "syms" command. The difference between these two commands is that only a single symbolic object can be declared using the "sym" command whereas multiple symbolic objects can be created using "syms" command. Then a symbolic expression can be created by first declaring symbolic objects and then writing an expression just as done in case of numerical expressions. Then I have mentioned some basic Symbolic MATLAB Commands that are used in the examples that are illustrated in the tutorial regarding Symbolic Integration.

The symbolic MATLAB commands are:

- The "double(f)" command is used to transform a symbolic expression "f" written in exact form to numerical form.
- The "collect(f)" Command groups the terms in the expression "f" that have the variable with the same power.
- The "expand(f)" command expands expressions by carrying out products of terms that include summation and uses trigonometric identities and exponential and logarithmic laws to expand corresponding terms that include summation.
- The "factor(f)" command is the opposite of expand command and reduces the polynomial expression to be a product of polynomials of a lower degree.
- The "simplify(f)" and "simple(f)" are used for simplifying the form of an expression. The command [F how] = simple(f) gives the simplified expression as well as the method used for simplifying the expression by assigning it to variable "how"
- The "pretty(f)" command displays a symbolic expression in a format resembling the mathematical format.
- The "findsym(f)" command displays the names of all symbolic variables in "f" in alphabetical order.
- The "solve(eq)" and "solve(eq,var)" commands are used for solving the equation. The "solve(eq)" solves the equation for default variable "x" and the "solve(eq,var)" command can be used to solve the equation in terms of the

variable “var”. If there is only one variable, then the solution is numerical, otherwise the solution is in terms of other variables.

Now coming to Symbolic Integration. Here, we have both Indefinite and Definite Integration. “Int(f)” Command is used for Indefinite Integration if the expression contains only one variable. If the expression contains more than one variable, then the Integration is carried out w.r.t. the variable mentioned in “Int(f,var)” command. For Definite Integration, we have the commands: “int(f,a,b)” for a single variable expression and “int(f,var,a,b)” for multiple variables in an expression to be integrated while “a” and “b” are the limits of the integration.

Now, we will discuss all the examples in the tutorial. The project.m file with all these examples is attached on my google sites term project page. The first example is relatively simple just to give an idea of Indefinite Integration. First, we declare a symbolic variable “x” and then introduce a symbolic expression. Then, we simply use the “int” command to integrate it w.r.t. “x”. The pretty command writes the integrated expression in a mathematical form. Subplot command is used to plot both the function and its Integral.

The second example uses the concept of definite Integration. Here we again declare symbolic variables and an expression in terms of those symbolic variables and then integrate the expression with respect to the limits 0 and H.

In the third example, first symbolic variables are declared using “syms” command. Then v(t) and T are defined in terms of symbolic variables. Next, equation (1) as mentioned in the tutorial is solved in terms of V using the “int” command for symbolic integration. The limits of integration are set from 0 to T. The solution is then further simplified by “simple” command and [how] specifies the name of the simplification method.

In the fourth example, R=3 and L=8. Then w is written as a symbolic expression in terms of “R” and “y”. The Volume of the fuel at height “h” can be calculated by substituting “w” in the integral in the equation for the volume and carrying out the Integration using the definite integration. The result is an equation that gives the volume “V” as a function of “h” The value of “h” for a given “V” is obtained by solving the equation for “h” using the MATLAB command “solve” Command, which is introduced in the tutorial. In the current scenario, the values of “h” have to be determined for volumes of 40, 60, 80, 120 and 160 thousand litres.

This concludes the tutorial on Symbolic Math Toolbox with specific focus on symbolic Integration. All the project related files are attached on my term project page. Now I will run the project.m file just to give a brief demo of all the examples mentioned.