|  |  |
| --- | --- |
| Code | output |
| clear  clc # clear all |  |
| clear  clc  #Use symbolic math tool  #exact value  syms f(x) # ‘syms’ used to load the symbolic tool box and  # declaring f(x) as a function  syms x # declaring x as a variable of that function.  f(x)=x^2; # equation  dfdt = diff (f, x, 1) # ‘diff’ differentiation (function, base,  # how many times derivatives) #’dffdt’ is a variable | Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.  dfdt(x) = (symfun) 2⋅x #result |
| clear  clc  #Use symbolic math tool  #exact value  syms f(x)  syms x  f(x)=sin(x)^2; # new equation  dfdt = diff(f,x,1) | Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.  dfdt(x) = (symfun) 2⋅sin(x)⋅cos(x) #result |
| Now put the value of x:  dfdt = diff(f,x,1) we are putting the differentiation value at a variable ‘dfdt’  so, to put value in the equation :    dfdt(2) # 2 is the value of x | dfdt(x) = (symfun) 2⋅sin(x)⋅cos(x)  ans = (sym) 2⋅sin(2)⋅cos(2) |
| Now, here it is not showing the exact value of  The equation. To see the ans we need to copy the ans and pest it like this:  Copy: 2⋅sin(2)⋅cos(2)  Code: 2\*sin(2)\*cos(2) | ans = -0.7568 |
| Now FDD:  Fdd= y1-y2/h  Here, h=delta x  Here, y2= f(x+h)  y1= f(x)  f’w |  |
| Example:  #given  x=2.12;  h=2;  x\_PLUS\_h=x+h;  #given end  # given equ = 3\*exp(2.5\*x)+2;  # or equ = 3\*e^(2.5\*x)+2;  y1=3\*exp(2.5\*x)+2; #y1= f(x)  y2=3\*exp(2.5\*x\_PLUS\_h)+2; #y2= f(x+h)  Fdd= (y2-y1)/h  disp (['The value of FDD is : ', num2str(Fdd)]) | The value of FDD is : 88898.3513 |
| Exponential can be written in 2 ways.  exp or e^  #equ = 3\*exp(2.5\*x)+2;  # equ = 3\*e^(2.5\*x)+2; |  |